Aedas R&D: Global Practices of Computational Design

Pablo Miranda, Christian Derix, Asmund Gamlesaeter, Lucy Helme
AedasR&D | Computational Design and Research

ABSTRACT
This paper gives an overview of the approach of working methods at the Aedas R&D Computational Design and Research [CDR] Group. It first contextualizes research in architectural practice and tries to propose an explanation for the difficulties in implementing it; then explains the evolution of the groups' computing approach from bespoke to heuristic sets of lightweight applications. It concludes with examples of the developed computational design approach.

1 RESEARCH WITHIN ARCHITECTURAL PRACTICE

1.1 HISTORICAL BACKGROUND
The origins of the systematic integration of experimental approaches in architectural practice can be found in the formation stages of modern architecture. It was at the end of the nineteenth century that the industrial research lab first took shape in places such as Thomas Alba Edison's Menlo Park, where Edison applied the principles of mass production to the process of invention (Friedrich Kittler 2002). The avant-gardes of the early twentieth century, in their effort to address the aesthetic, social, and new production forms of industrialism, integrated many aspects of the research lab into their practices. The Bauhaus or the Constructivist Vkhutemas can be seen, in this light, as research labs, functioning under similar principles as their industrial counterparts but with an agenda of aesthetic and social innovation rather than pure technological development. (Kenneth Frampton 1990).

The introduction of systematic experimentation to architecture remained, for the most part, limited to plastic research and bound to academia. Despite the importance of technological innovation in the modernist discourse, the economic difference of means of production between industry and architecture prevented the implementation of in-depth research in standard architectural practice, and limited its application to experimental architecture. Whereas research is substantial to most industries, architecture remains, in this sense, a conservative discipline that is strongly linked to a traditional form of production, that is, the tradition originated in the Renaissance of drawn representations.

1.2 RESEARCH AND ECONOMIC CONTEXT
In An Evolutionary Architecture (1995), John Frazer pointed out how architecture's form of production remains labor intensive rather than capital intensive (John Frazer 1995). Even without factoring in the construction or fabrication process itself, the development of an industrial product and a building differ substantially simply in the number of times the design will be reproduced. Architecture's products are, in most cases, singular (buildings), whereas in industry, a single design will be reproduced as many times as the market demands, often using largely automated processes. This obvious difference is essential in distinguishing the possibilities of the forms of research in architecture and industrial production. First, in an architectural design process, the demarcation between design, research, and production (construction) is unclear, and to identify the specific effects of investment on a particular part of the process, although still possible, become more difficult and, thus, harder to argue for. Second, the one-off character of buildings and the diversity of tasks and situations they may respond to (in terms of function, location, size, construction types) make research difficult to implement, if compared with industrial products. The areas architectural production where it is possible to argue for research are those involving industrial production, such as prefabrication or, more recently, numerically controlled fabrication.

Alternatively, in the labor-intensive architectural process, what is "reproduced" are the skills of the designers and the processes they use, and it is, perhaps, in regard to these that research can also take place.

The emergence of global practices and, in general, of large architectural firms, together with the ubiquity of digital technology, challenges this situation. Large firms invest in research to create a "competitive edge" by building digital know-how in the form of custom software, such as Gehry Technologies' Digital Project, or custom digital workflows.

2 COMPUTATIONAL DESIGN DEVELOPMENT STRATEGIES

2.1 TRANSITION

When we started out in 2004, we employed an "academic" approach to computational design, trying to implement computational heuristics that could create wholly novel workflows, often covering many design stages. While this tends to create bulky one-off programs with complicated interfaces that are hard to re-deploy and impossible to integrate with a non-computational architectural workflow, it allowed us to gauge the differences between traditional and computational heuristics.

Figure 1
Widgets for Access and Movement Design

As a direct result, we abandoned computational heuristics for some time and attempted two routes: building general utility programs for recurrent design situations and solving bespoke [custom-made?] design aspects like facades or mathematical shapes. Consequently, computation became more integrated into the architectural workflow as the risk of development asymmetry had been eliminated. Computing simply "replaced" or automated a stage of design rather than offering creative alternative processes or inputs through research.

2.2 WIDGET ASSEMBLIES

Compiling generic applications for recurrent design situations led us to collect small applications—widgets—with distinct, limited scope. The widgets' flexibility—supported by the use of Java and OpenGL—offers a modularity that allows for fast and easy assembly of algorithms into different workflows. Additionally, the creation of widgets promotes short development cycles and breaks the computational design workflow into short iterations, where several group members can develop versions simultaneously. While not pursuing an explicit methodology, this lightweight, short-cycle approach is reminiscent of the software development paradigm called Agile or Adaptive Development (James Highsmith 2000). Adaptive Development also draws from systems theory and distributed representation, where redundancy allows for adaptation and a component's defect is absorbed by alternatives, and is diagonally opposed to many current developments of large applications with unlimited scope.

This open framework of applications produces many hybrid architectural computation workflows, where the designers and code form an adaptive whole. Real-time simulation and interactivity foster a conversation as envisioned by Negroponte, who hoped for an "evolutionary dialogue" between user and machine (Nicolas Negroponte 1970).

Figure 2
Widgets for Visibility Analysis and Visual Formation

2.3 META-HEURISTIC DESIGN COMPUTATION

As the risk associated with workflow integration has diminished and widget assemblies afforded new architectural heuristics, the CDR group managed to re-introduce computational heuristics, such as evolutionary, annealing, or ant-colony search algorithms that were tested outside workflows in the early years. The modularity and limitation of scope of our widgets represent an ideal framework for the explicit encoding of otherwise implicit qualities and processes—key performance indicators—of a designer's rules of thumb and assumptions. Search and combinatorial algorithms can approximate good solutions for design situations with complex tasks but clear targets. The simplicity and real-time aspects of our simulations produce transparent results and empathy, encouraging their use. The synthesis between subjective architectural heuristics and search algorithms results in ever-novel meta-heuristic design computations like Negroponte's "evolutionary dialogue," forming new commercial workflows through research in large firms like Aedas.

Figure 3 Widgets for Land-Use Mix, Density and Massing Generation

CONCLUSION

The CDR group has collaborated with many partners, from urban designers and planners, to developers and architects, to interior and furniture designers, who lent us their "implicit knowledge," design drivers, and assumptions. Figures 1 through 3 show five widgets from larger collections used for Digital Masterplanning and Architecture. Figure 4 shows several applications built simultaneously through agile development for a specific design brief.

Large architectural practices have returned to conduct research in the interest of the industry and, thus, have turned themselves into public research agencies like Menlo Park was over a hundred years ago.

Figure 4 Widgets for Khalifa-bin-Zayed Mixed-Use Development Competition.

REFERENCES

FRAZER, J. (1995). EVOLUTIONARY ARCHITECTURE. LONDON: AA PUBLICATIONS.
HIGHSMITH, J. (2000). ADAPTIVE SOFTWARE DEVELOPMENT: A COLLABORATIVE APPROACH TO MANAGING COMPLEX SYSTEMS. NEW YORK: DORSET HOUSE PUBLICATIONS.
KITTLER, F. (2002). OPTISCHE MEDIEN, BERLINER VORLESUNG 1999. BERLIN: MERVE VERLAG.
NEGROPONTE, N. (1970). THE ARCHITECTURE MACHINE. CAMBRIDGE, MA: MIT PRESS.
FRAMPTON, K. (1980). MODERN ARCHITECTURE: A CRITICAL HISTORY. LONDON: THAMES&HUDSON.