Towards architect-aided computing design

Agnieszka Katarzyna Sowa
Towards architect-aided computing design

Agnieszka Katarzyna Sowa

In the design process of some recent, specific architectural projects the part elaborated by computers and machines significantly grows. They could generate, optimize and produce the most complicated and complex solutions, taking over some tasks which before were the domain of architects. This article presents a project carried out by postgraduate students at Eidgenossische Technische Hochschule in Zurich, Switzerland, where such a digital design process was implemented, with all its advantages and disadvantages. The observations and conclusions gained during the work allow the author to formulate the concept of Architect-Aided Computing Design, to define some challenges for architects created by such a working method, and to present an analysis about the potential new software for architectural production.

**Keywords:** structural optimization, rapid prototyping, CNC production, CAAD education
1. Introduction

1.1. Architect-aided computing design (AACD)

CAD and CAM technologies have opened up new opportunities for the creation of very complex architectural structures which until recently had been too difficult and expensive to design, produce and assemble. In such projects digital tools can assist designers in their thinking process from the very beginning [1] until the moment when the designed architectural form is built in reality, going through different computer-aided design methods, such as for example generation, optimisation, fabrication. Naturally, not every type of architectural tasks could take full advantage of a broad range of such methods. However, the application of numerous techniques may lead to changing the role of architects in such projects. The position occupied by the designer is that of a supervisor who “enter the stage” mostly to make design decisions between periods when only machines are recalculating the project, creating variations of possible solutions and optimising them. Still, it is architect who produces initial ideas which can also further evolve under the influence of computing results. It is the architect, too, who improves digital tools (software) basing on conclusions inspired by these results.

In the design process of specific architectural objects the computational part grows so significantly, that we can talk about “Architect-Aided Computing Design” – as an extension and developing phase of Computer-Aided Architectural Design.

The AACD schema could be implemented only in specific design conditions, for example:

“Genetic algorithms will only serve as useful visualisation tools if virtual evolution can be used to explore a space in which it is impossible for the designer to consider all potential configurations in advance, and only if what results shocks, or at least surprises.” [2]

While optimisation seems to be particularly useful when for example irregular structures are to be used in project, prototypes production can be employed in innovative and unique projects. But if a given project comprises all such features – the Architect-Aided Computing Design becomes a fact.

1.2. Challenges in architectural education and practice related to AACD

Apart from all its advantages, rapid development of CAD/CAM methods in architecture poses numerous challenges, both for academics and practitioners. Modern architects need more and more expertise if they want to take full advantage of all the possibilities offered by advanced digital tools. The educational challenge is to give architects the optimum amount of information which would at least allow them to communicate with IT
specialists and let them solve their most common problems by themselves. What is also very important is to make them realize what architectural forms they might construct and produce with state-of-the-art technologies. However, not all architects need to use a full spectrum of new technologies in every project due to considerations such as costs, needs or sustainability. The speed of software development is also very fast, so it does not make sense to teach only specific applications or technologies. Instead, it is better just to present the general idea of CAAD/CAAM, on examples of short project. It is exactly this approach that is adopted at the postgraduates studies at the CAAD Chair at the Architecture Faculty at Eidgenossische Technische Hochschule (henceforth ETH) in Zurich, Switzerland. At ETH students learn not only the general idea of programming sophisticated forms and graphics which would look good on the screen, but they also experiment with manufacturing their designs with CNC machines and rapid prototyping tools. Usually, the method employed in at this postgraduate course is so-called “digital chain” in architectural design.

“[…] digital interconnection of data within architectural design and production […], […] hypothetical lack of paper prints and human labour in production […]” [3]

Coming back to challenges posed by CAD/CAM development, it seems also that architects still have to eliminate their own deep-rooted prejudices. Ever since the computer appeared in architectural design for the first time, they have been afraid of the idea that machines will take over their role as form designers, artists, engineers in the sense of creative values. But architects need not worry – in the digital design process, it is still the designer who “tells” the machine what to do. The computer, however, allows architects to regain control over the entire design/building process. Sooner or later we will have to face the situation in which

“ […] there won’t be conventional drawings [in the design process]. There will be no paper, but fully documented three-dimensional models on CD … In the future we will have a digital model, then go directly to CADCAM manufacturing process.” [4]

To sum up, nowadays it seems to be obvious that the computer plays increasingly larger role in project coordination and planning, or information exchange, but it is sometimes possible to observe that the idea that the form could be generated by a computer program on the basis of different factors (for example environment features) still may result in controversies.

2. Project

2.1. Overview

There are two most discernible “outside” features of highly computer-aided architecture: free forms (called “blobs”) [1] or/and complex structures.

Agnieszka Katarzyna Sowa
complexity often comes from the structural irregularity, which seems to fascinate architects nowadays. However, in some projects the structure is regular, but applied to complex shapes (e.g. Bernard Franken’s BMW bubble pavilion, [5]); or the shape is simple, and structure irregular (e.g. Toyo Ito’s serpentine pavilion, [6]); or both shape and structure are complex and curvilinear (e.g. China Stadium of Herzog de Meuron, [6]). There are also projects where only parts of the design – e.g. ornaments – are prepared with CAD/CAM techniques. Such architectural designs are still not numerous, partly because of high costs. Nevertheless, some features and scenarios of advanced CAAD/AACD workflow could be implemented at architecture faculties as educational tasks, for example during the already mentioned postgraduate studies at ETH.

The students of 2003/2004 postgraduate course planned to design and build an architectural object in 1:1 scale, using all techniques and tools they have learnt throughout the studies. After four months of hard work, such as programming, production, experiments and discussions, the final result was achieved – the prototype of the designed structure was presented at an exhibition at ETH from 14 until 31 October 2004. Its design process is a good example of how architects could “cooperate” with computers/machines in order to achieve a result which would be impossible without them. The project has also some characteristics of the AACD approach.

The first idea was that the final form should not only be produced by machines, but also that the design process itself should be widely supported by computers. But the design did not proceed in the direction of free forms or “blobs”. This kind of architecture objects is easy to visualize on the computer screen but then difficult and expensive to build. Owing to tight deadlines and limited programming skills of the group members, it was decided to concentrate more on the complexity of joints and production, rather than the creation of trendy curvilinear forms.

The initial idea of the project was to create a 3-dimensional irregular straight-linear grid for a small scale architectural form. It would then be used as a basis to produce real construction elements for 1:1 prototype, which should be easy to re-assemble and transport. The grid appeared as an intersection of freely rotated planes and a cube – an outside form of the object (Fig. 1a). The inside space was also a cube, subtracted from the bigger one (Fig. 1c). The result of this scheme was a structure build by irregular frames (Fig. 1d,e,f) which have different widths and cross with one another at different 3D angles.

It was expected that the grid will be parametrically generated so that it could have different densities as well as different levels of irregularity depending on designer preferences. The size of the inside space was also parametrically defined and so was the thickness of the frames. These three features already offered quite a broad variety of aesthetic and structural results (Fig. 2, 3). After generation, the next step in the project was the...
Figure 1. General idea of the structure. (1a) freely rotated planes inside a cube; (1b) linear representation for calculation purposes; (1c) subtraction of an inside form; (1d,e) frame structure with wireframe visible; (1f) frame structure; (1g) mathematical description of a frame set (four coefficients for every frame).

Figure 2. Maya models of cubes generated by scripts with different parameters.
optimization of the structure, than production, all proceeded by computers. Architects were expected to create proper software for such design process and then to “cooperate” with machines – influence their calculations and being then influenced by them.

The choice of the programming tool was made already at the conceptual work stage – it was MEL (Alias Maya™ Embedded Language) that was chosen. First and foremost it was because this script language seemed to be easy to use for the challenges expected during the work and that it offered fast visualization tools. Secondly, because students were already familiar with its syntax and capabilities. It turned out later not to have been the best choice – MEL and Maya were not stable enough to carry complicated calculations needed for optimization. As for other purposes (such as generation, preparation of production drawings, etc.), the scripting language met the initial expectations.

2.2. Generation

Work on the project proceeded as follows: firstly, a simplified digital model of the structure, based on randomly generated input, was prepared in MAYA.
for visualization purposes (Fig. 1a). From the programming point of view, there was no need to visualize the data during and after the generation process. A visual analysis, however, was useful both for the programmers and the designers since they could evaluate the scripting progress and aesthetic results as well as draw conclusions concerning the structure.

Simultaneously with visualization tools, further MEL scripts were being designed to improve the generation process of the carrying structure, because already in this phase some limitations for planes were introduced (for example, planes which are parallel to cube walls were eliminated). The result of the generation process was a mathematical description of the created wireframe (Fig. 1e) as a set of planes’ coefficients. All other scripts were prepared in such a way that they needed only this very small amount of data (these four coefficients for every structure frame) to redraw the entire grid. Owing to this method information on very complex structure is stored in a relatively small text file (Fig. 1f). The structure description could be easily exchanged between different scripts (e.g. for optimization, production) and also easy visualised on every stage of generation/optimisation process. The created scripts offer a possibility to observe the same structure with different inside volumes.

2.3. Optimization

The grid generated by the script could not be used instantly as the carrying structure because it was created randomly and it did not satisfy all the conditions which have to be fulfilled by a buildable structure. A variety of these conditions could generally be quite large: the type and thickness of the material influence the structure by specifying maximum and minimum distances between joints, maximum size of elements available from a piece...
of material etc. Furthermore, the appearance of covering panels on the structure was considered during the design process, as irregular polygons filled the spaces between frames, parallel to the cube walls. If it would be decided to use such panels, their material would have also influenced the optimization process. The final shape was also dependent on the type of machines used to manufacture elements.

All things considered, the aim of the optimization was to improve the structure so that it could be possible to build with CNC machines; in other words, to translate an abstract mathematical model into a real form with all its limitations. It was planned in the optimization process that the genetic algorithm would be used; this would work on the “population” of cubes, analyze their features and exchange planes/frames between them with an aim to create a cube as close to the ideal as possible. Due to time constraints available for programming, a simplified version of the evolutionary algorithm was finally created. However, it met the original expectations only partially. Namely, only one structure was created whose planes were analyzed in the environment of other planes, in order to examine whether the joints it created with others had proper features for future production (e.g. the distance between joints, amount of connections with other frames, etc.). In every optimization cycle the plane with the worst features was detected and deleted, and new one was created instead. The problem was that the script was not learning during this process, so there was no guarantee that the newly generated random plane would have better features than the deleted one.
2.4. Prototypes and production

The optimized structure constituted the basis used to automatically produce fabrication drawings for CNC machines, also generated by MEL scripts. The progress of the design was all the time controlled by the evaluation of models, digital as well as physical, produced by rapid prototyping machines (3D printer, laser cutter) from different materials (Plexiglas, cardboard, wood, foam) (Fig. 6). At the same time with all scripting work the research on materials, joints and manufacturing was carried on. It was planned that the 1:1 prototype of the structure would be produced from wood by 3 axis milling machine. So that the main idea of flat, rotated planes could be preserved. But what is easy to generate and visualize with the help of computer is not always easy to manufacture and build. The size limitations of milling elements as well as assembly possibilities led to a situation when there was a need to divide every frame into smaller elements and then to join it together during the assembly. This kind of connection needed a special flat joint. Other connection was the one between frames – a slot joint connecting planes which cross in 3D space at different angles and depth. The width of such slots depends on the thickness of material used as well as on the angle between frames. Because the designed structure seems to be dense and rigid enough, the slot joints could be slightly less precise and there could be larger tolerance in dimensions than for example in the case of the flat connections mentioned above. Joints design – from the first sketches until the final outcome – was also constantly supported with the production of prototypes.

Because of chosen slot joining methods, the assembly rules also needed to be considered and calculated by the computer when manufacturing...
drawings were being prepared. The decisions about the direction of each slots were also done by computer.

As was mentioned before, the structure was expected to be easy and fast to produce and assemble. These features were achieved first, by making the elements ready to use already after milling (without post-production).

Figure 8. Assembly process for the final object – the slots on the frames are visible.
and second, by preparing the system of elements coding (every element was engraved with a description of its position inside the structure as well as information about elements it is connected with). The final effect was the successful production of wooden 1:1 prototype (Fig. 9).

All details related to the workflow process, some additional information and more pictures can be found in [7] and in the NDS group work booklet[11].

2.5. Final results of the project – successes and failures

As was stated before, the idea of architectural forms fully generated by the computer often gives rise to controversies, on the part of architects. Even among CAAD students who had chosen this particular course to broaden their knowledge in the digital design the opinions differed significantly already during the conceptual work on the presented project – the fascination with technology and opportunities it provides, with the idea about the architect as the form designer who has the influence on the appearance of every detail of the final project. Some interesting research ideas, such as the creation of software which would use environment features and information to create an architectural form fitting a given urban location, had to be abandoned because there were no possibility to persuade all the group members that “fully computer generated form” is an idea worth investigation.

So even if the workflow is a good example how to teach digital architectural design methods, the author as a student herself has one critical remark related to the coordination and leadership expected from the CAAD chair assistants at the initial phase of the work. It was expected that the final project would be a result of the work of all the members of the group (i.e. 13 people). It turned out at the very beginning that there is too many individuals to find fast a work idea which would be a compromise between “computer form generation supporters”, and believers in the statement that “it is the architect who is the one and only form creator”. Moreover, during the work it was also difficult to cooperate and coordinate work of so many people as well as to communicate in English, which was none of the students’ mother tongue. Naturally, postgraduate students are always expected to be independent and creative in their work. But perhaps
with some moderation of the discussion, the initial phase of the work would have been shorter and there would have been more time for “real work”.

2.6. Human interference in the digital design process

The visual results of computer generation and optimisation in the project were satisfying from the aesthetic point of view – because of the complexity and irregularity of the structure contrasted with simple, cubic outside form. In fact, out of three generated and optimised proposals for production the one which was selected was not more “beautiful” than others – what made it more interesting than others was a “natural” gap in the structure which made it possible to enter. So if the generation/optimisation script had been able to generate every structure with “door opening”, the decision could have been made purely on aesthetic grounds, which had been the goal from the beginning.

With the Architect-Aided Computing Design concept we assume that the architect appears only when there is a need for a “design decision” – connected with nonmeasurable project features – as aesthetic ones. During the work on the presented project, also other behaviours were present. The students agreed that visual appearance of the final form did not need any further manual improvement, in contrast with its structural features. All “human” architects who looked at the model claimed that they would be able to add a frame to the structure or change the position of the existing one immediately and easily so that the entire structure be easier to produce and construct. After a short ‘struggle’, however, almost everyone resigned because movements and adjustments in one joint made the situation worse on the others. That was the reason why the computer was needed during the work on the structure. It was simply too complex to be handled with without the help of machine. This fact seemed not to be obvious for some students involved with the project. The final structure, however, is a result of computer generation and optimization as well as human addition and adjustment. These were possible thanks to special tools created i.e. scripts which enabled additions and changes in the structure during and after calculations. The scripts could be used to add manually a construction frame or to change its position and then add its coordinates to the mathematical description of the entire structure.

2.7. Further research

Further work on the structure itself could proceed in the two main directions. Firstly, the programming tools could be developed so that they can handle other volumes from non-regular but also non-curvilinear forms to NURBS, with different optimization factors influenced by different methods of CNC production and materials used. The other way to continue the initial idea is to work on other construction systems using the same grid for different scales (patterns and ornaments vs. construction systems).
and purposes (for example kiosks, fair stands, etc.). It needs to be remembered that all the research was carried out on and all the scripts were written with one main aim in mind – to prove that it is possible to produce the designed structure in the time given. That was the reason why there was no user interface, no “user-friendliness”. According to the programmers, it was not needed during the workflow, where the aim was to build only one instance of such structures, but could be a next step to improve the results of the research.

3. Conclusions
3.1. Need for new software?

The important feature of AACD is that the architect can appear in the design process only when nonmeasurable – for example aesthetic – decisions are expected to be made. Between these moments computers should take over all the work.

These days such an approach is still discussed in more theoretical terms owing to inadequate software development (but some applications are already used e.g. plug-in for Alias' Wavefront's Maya – Genr8 [9,10]). For instance in the case of unique architectural idea usually dedicated, sophisticated software needs to be prepared at the generation/optimisation stage. However, when it comes to the production/fabrication stage, some universal applications helping architects to make prototypes of their ideas could be proposed.

On the one hand, this idea sounds promising to anyone who has already fought few times with not so friendly machines. Moreover, each of them needs special “treatment”, different software, special tips and tricks. They have limitations, too, but only on the computer screen no material borders for the creation of architectural form exist. Some of these limitations could be overcome by using more than one fabrication method per model (for example, 3D printer for joints with complex geometry and laser cutter for covering structure panels [11]).

On the other hand, there are also “additional values” when working with these tools. The best example here is working with CNC milling machine [10]. During the fabrication milling tools can create additional patterns and ornaments, not programmed, but emerging from the method this machine removes the material to obtain the form. If it is the designer who plans the whole process from the 3D model to the final prototype, he/she could also anticipate which tool is the best to use to get this kind of “additional values”.

To sum up, when the need for the new software is considered, the specification of the tasks and expectation for such “software for production” could look as follows:

1. input – 3D model of a form, prepared in any modelling software by architect or by configuration/generation/optimization software)
2. form analysis
   • decision about what machine/machines should be used
     (depending on model size, time limitations, etc.) and which
     material (if it is not specified by the designer)
3. preparation for production
   • preparation of the form, for example for 2D fabrication:
     contouring, triangulation, unfolding etc.
   • mixed method – decision which parts will be produced with
     which machines
   • optimization of material used (2D arranging on material sheets
     for laser cutter, 3D optimisation for milling machine, addition of
     some connections between models and material pieces if the
     material has to be rotated during fabrication, etc.)
   • generation of code for the machine (for example G-code for
     milling)
4. managing data and machine usage (anticipation of time needed for
   production, generating optimised timeframes for the working time of
   every machine, etc.)

3.2. AADC perspectives

Generally speaking, if it was not for computers, the creation and execution
of an idea described in this article would be impossible, at least in such a
short time. What is more, it can be said that in the presented project not
only do digital tools had an influence on the appearance of the final form,
but it was also the designer who influenced digital tools during the
workflow.

The final effect of such process should be regarded only as an instance
of an endless variety of possible solutions offered by the computers. Most
tasks were processed by them, which made the work faster and easier.
Naturally, it was only possible after appropriate software tools were
designed.

Unfortunately, due to communication problems and time limitations, it
was not possible to elaborate the results which can be used instantly in the
real building process. However, proper educational and implemental
solutions for CAAD/AADC are still to be found, that is why all experiences
are important and useful, even the ones not ended in spectacular success.
Nevertheless, the presented digital design work schema seems to be
satisfying. So is the fact that the group of architects – neither qualified
programmers, nor constructors – was able to design, program, produce and
build such a structure in quite a short time seems to be significant.

It cannot be claimed that Architect-Aided Computing Design is the one
and only future perspective for the architect – machine “cooperation”,
because of limited range of design tasks where it could be applied. Even if
the computer seems to be able to take over most architect’s activities in
such projects, the knowledge and ideas of the “human” are indispensable. However, as AACD comprises numerous computer technologies, it seems to be very suitable for education purposes, because it gives students a good possibility to familiarize themselves with a broad range of computer-aided design methods used by the most open-minded contemporary architects.

Acknowledgements

The project presented in this paper was a group work of postgraduates studies at ETH in 2003/2004 at the CAAD Chair by professor Ludger Hovestadt. The students were: If Ebnöther, Jörg Grabfelder, Anna Jach, Jae Hwan Jung, Alexandre Kapellos, Irena Logara, Michelangelo Ribaudo, Hanne Sommer, Agnieszka Sowa, Detlef Wingerath, Thomax Wirinsg, Li-Hsuen Yeh. The author would like to thank them all for all their work, help, cooperation and company during this year. She would also like to express her gratitude to the CAAD chair staff (especially Karsten Droste, Markus Braach, Russell Loveridge and the NDS coordinator Philipp Schaerer) and Martijn Stellingwerff from Delft University of Technology (for an exchange of some interesting ideas about “software for fabrication” during eCAADe conference in Lisbon).

References


Agnieszka Katarzyna Sowa, Bialystok Technical University, Department of Architecture, Grunwaldzka 11/15, 15-893 Bialystok, Poland
agnieszka.sowa@gmail.com

Towards architect-aided computing design