Form Follows Software

Pierluigi Serriano
University of California, Berkeley

Abstract

Software selection affects design outcome. Computer applications externalize in their graphical interface and in their internal logic a set of assumptions about how objects are constructed and space is represented. Accessibility of tools is in direct correlation with their rate of use. Depending on how user-friendly particular functions are, their use will appear with higher frequency than those foreign to the technological frames of the user groups for which software is designed. As each software is geared towards the needs of specific communities, it replicates in digital fashion those disciplinary practices already present in the analog world. However, modeling results are bracketed at its inception the very moment a particular 3D package is chosen from a diverse array of digital offerings. If the application adopted is designed to appeal to the computer animation industry, the modeling results will bear the imprint of those organic qualities: buildings will appear character-like. Since computer programs have built-in slant meant to aid disciplinary specific users, they yield families of designs with formal commonalities. Unquestionably, proficiency of software use also broadens inventiveness of design. Nevertheless some applications make particular transformations harder to achieve, and as a result will be likely to exclude those modeling options from architects’ imaginary world.

Keywords: modeling options, built-in slants, form-making, technology of orders.
Form Follows Software

Pierluigi Serriano
University of California, Berkeley

1 Introduction

In the bestseller The Tipping Point, Washington Post reporter Malcom Gladwell gives a provocative account of how ideas and trends snowball from being the prerogative of a narrow group of acolytes to becoming object of collective attention in global communities. He argues that there is a moment in the history of any macro scale social phenomenon when a ripple effect kicks in, its rate of diffusion escalating exponentially to peaks unimaginable in the early stage of its growth. The sudden rise of sales of Hush Puppies—fashionable shoes in the American market—and the inexplicable change of crime rates in New York City neighborhoods of Brownsville and East New York are two manifestations of tipping points. The author proposes to think of these rapid developments and spreads of social events as epidemics in order to understand processes of reciprocal influences and behavioral fluctuations in groups (Gladwell 2002).

Digital Design had its tipping point in the mid-nineties, but traces of its up-and-coming pervasiveness are distributed in the history of modernity. The twentieth century was the stage for the celebration of scientific knowledge. Disciplinary domains underwent a process of scientification that recast the expectations and hopes of what had then become the knowledge society (Böhme and Stehr 1986). Computer technology matured under the pressure of defense priorities, but spilled over to non-military purposes in the fifties. In the realm of computer visualization, Sketchpad made its entrance in 1962 giving birth to the field of computer graphics through the effort of Ivan Sutherland (Gere 2002). A computer-generated mesh, flying in an imaginary non-place, first appeared in the January 1965 issue of Architectural Record. The article titled “Will the Computer Change the Practice of Architecture?” reported on the future avenues of Computer Aided Design. A great many of these unprecedented resources were discussed at a conference at the Boston Architectural Center held on December 5th, 1964, in the presence, among others, of Serge Chermayeff and Walter Gropius. As the readership was being coaxed into the magic of fluid forms, machine-generated aesthetics, and tales of liberation from the dictatorship of the glass box, architecture was officially registering the introduction of computers in its horizon.

2 Methods

The argument of this paper is constructed using qualitative research methods to identify modes of use of computer applications in architectural practices, whose members have diverse hands-on expertise with the software. “If you want to understand what a science is . . . you should look at what practitioners do” (Geertz 1973). The implementation of qualitative methods through interviews and non-participant observation is designed to retrieve what the anthropologist Clifford Geertz calls “thick description.” This notion shapes a methodological approach that factors in the socio-cultural environment in which human actions take place. In the context of this paper, using computer technology is enmeshed in the subculture of the workplace and the personal trajectories of its community members (Forsythe 2001).

Two levels of analysis inform the paper: First, a comparative approach between three modeling programs adopted in diverse architectural practices and of appeal to signature architects. Literature review and usability testing will help to stake out the organizational properties of each graphical interface. This will cast light on ease of use and attitudes toward the extent to which computers can formally generate design outside of the broader set of concerns—social, cultural, technical, structural, and so on—in which architecture is customarily practiced.

Second, six case-studies are used to gain an insider’s perspective on the adoption of 3D Modeling packages in the design process. Unstructured interviews were conducted to retrieve data on behavioral patterns in sequencing computational functions to monitor the incremental refinements of design ideas. Although the focus of the paper is to establish a correlation—and possible causation—between commands and digital forms, narratives of team members interaction, and profiles of personal and professional backgrounds of the parties involved will be variably inserted into the text.

3 The Precedents

While in the ‘60s only a few architectural firms were already adopting CAD software for production aims, it took twenty more years for this technology to gain broader exposure and leave an imprint on the imagination of architects. Those familiar with the early stage of rendering algorithms will remember the iconic computer image of the Museum of Contemporary Art in Los Angeles by Arata Isozaki at the beginning of the eighties, maybe the first digital eye stopper in architectural memory. Reminiscent of the early paintings of the Italian painter Giorgio De Chirico, the solitude of that suspended landscape, with primary shapes arranged on an evenly lit ground plane, epitomized the start of a new era in architectural culture. Ever since, Computer Aided Design has transitioned from being the monopoly of competence of professional elites to being routine expertise in the contemporary design office. Twenty years later, Digital Design
is a collector of widespread media attention and an uncontested object of professional interests. The ever increasing level of computer literacy in the architectural workplace has swiftly permeated every aspect of the practice of architectural design. It is widely accepted to argue that the adoption of digital technology in professional practice streamlines the production process, empowers the design team, and multiplies exponentially the exploration of alternative schemes. As of 2003, the sight of a drafting board in an office is often interpreted either as the tangible remainder of an obsolete mode of production or a nostalgic holding on to a bygone approach to the making of architecture. In the analog age, a case among others, those conversant with Descriptive Geometry had obvious advantages in form-making compared to others who had less developed modeling skills. Comparatively speaking, to draw by hand a three point perspective was for many an intimidating, complicated, and labor intensive effort than crafting a single point perspective. It would take days and a much higher level of skills to deliver comfortably seductive bird-eye views: that was the prerogative of those few designers who were fully conversant with the complexities of descriptive geometry. The result was the abundance of single point perspectives in the representation of architectural designs, also as a byproduct of its being simpler to construct than multiple-point perspective. Nevertheless the rectilinear quality of many of those visions had its roots in the glorification of the rational and the efficient. The historically unprecedented possibility to represent the infinite through a dot (the vanishing point), gave rise to iconic images charged with highly symbolic value, among them The Last Supper by Leonardo and View of an Ideal City by unknown. But even more importantly the profile of those shapes was scripted in the very tools adopted to draw. Factual limits in the repertoire of forms were set by the very use of T-squares, the triangles, and compass to externalize arrangements existing in the mind.

4 The Hypothesis
It is my contention that the current 3D software packages follow the same logic of their analog counterpart. Morphological manipulations and their resulting forms can hardly be separated from the software that defined them. The construction of the graphical interface, the arrangement of the modeling and rendering icons, and the organization of the interrelated menus structure the task environment and the input of the topological data. As a result, their digital outcomes are bracketed in geometrically identifiable limits and bring about strongly patterned sets of architectural propositions. It follows that the choice of software is a primary, if not the utmost variable, bearing heavily on digital design. “Technological innovations open up new fields of possibility, which society either ignores completely or embraces without the assumption of any mechanical predetermination” (Lévy 1997). In the context of this paper the notion of form is extended to include not only shapes, outlines, or configuration of anything, but also the broader capabilities of software to generate formal outcomes in space. This entails looking beyond the focus on objecthood and assessing how mental processes find their digital correlatives in a modeling program.

If it is common sense today to use computers in the conception and execution of design, that common sense is not innocent. This paper presents a set of critical questions raised through a cursory analysis of three widely used modelers—Form*Z, 3DS Max, and Maya. These will be concisely analyzed in their graphical interfaces and compared in the modeling results that they yield. The thesis is that there exists a “Form*Zness,” a “3 DSMaxness,” and a “Mayanness”—if you will—built into the code that either enhances or hinders the exploration of three-dimensional alternatives in the design process other than those readily available in the defaults of each software. Tool is here defined as a combinatory procedure granting either interactively or through numeric input the generation of form. The ease—or difficulty—of use of each tool is in direct correlation with the frequency of use of that tool in the course of the modeling progression. If that procedure is scripted so that it can be grasped relatively easily by non-expert users, groupings of those shapes will populate the design. The number of iterations of that particular procedure will produce a distinct volumetric hallmark belonging to a family of identifiable forms; the more user-friendly the modeling tools, the higher the occurrence of those predetermined shapes will be in the formal outcome. A concrete and ubiquitous manifestation of this logic is the current proliferation of blob architecture.

5 Software Selection
Choosing a software is making a commitment to a world view. In basic programming practices, when the task to be accomplished is complex, pseudo code is used as the intermediary language to specify a sequence of actions to achieve a set of quantifiable goals (Grillmeyer 1998). In plain English, the software developer states a plan in linear fashion of how to accomplish the intended result. That pseudo code bounds the action domain, encapsulating the realm of possibilities end-users will have to come to terms with when implementing their design intentions. “The programmer designs the language that creates the world in which the user operates” (Winograd and Flores 1987).

It is widely known that computer applications in the private sectors target communities of potential users, and embed their practice into the code, in order to transform them into loyal buyers. Market demands are paramount forces in dictating the expansion of capabilities and the future directions of applications. Since a computer program stands for the computational arrangement of assumptions about how work is performed and how the process can be carried out, it follows that we shape software just as software shapes us. We could write a paper in Form*Z, and draw a plan in Microsoft Word, but we customarily choose not to do that, or at least this is not the first use that comes to mind. Form*Z is for modeling, whereas Microsoft Word is for writing. We operate according to our technological frames
Technologies are ways of building orders in the world” (Langdon, in MacKenzie and Wacjman 1995). The notion of value-free technology is untenable. For each technology, there is a conscious plan to structure human activity narrowing the array of effective courses of action available to the user. Technology is intentional just as design is. Today multiple metaphors guide the interpretative process of current digital design, particularly when it comes to arguing for what is known as blob design, the lure of the informal. Clay has now become digital to sculpt imaginary spaces and virtual worlds that we can see, yet can’t quite grasp in their tectons. The rhetoric of the polished renderings invariably touches the slick visions of positive architectural realities. Computer applications are partial accomplices in the fabrication of these fantasies. From our experience in the real world, we know that it is virtually impossible to have a perfectly flat surface using physical clay, and equally unattainable to draw organic forms using the straight edge, triangles and protractors. Each instrument embodies a range of possible uses, not all of them.

“When a work of architecture departs from culturally established patterns, it always requires a collective effort of clarification” (Bonta 1979). Kostas Terzidis ventures in outlining a taxonomy of the clichés, morphological common places, and formal strategies defining classes of digital designs (Terzidis forthcoming 2003). Caricature Form (Deformation), (Un)Folding Form, Hybrid Form (Morphing), Kinetic Form (Motion), Warped Eye, and Algorithmic Form are the classificatory categories used to comprehend the singularities of forms unimaginarable a decade earlier. Drawing upon aspects of human perception problematized throughout cultural history, Terzidis offers a post-rationalized reading of geometrical bundles. But where is the software in all this? The following is a cursory review of the modeling programs, their menus, their defaults, and their digital underpinnings. Most of the remarks will attempt in a necessarily incomplete form to outline some of the basic principles of the software and its mode of use.

6 Form*Z

Form*Z is a general purpose modeler conceived and designed by architect Chris Yessios and a team, for his fellow architectural colleagues. Despite its initial focus on modeling for architects, Form*Z has progressively widened its scope to accommodate multi-disciplinary modeling needs. Officially released in 1991, this software has gained increased recognition among an architectural elite of signature designers such as Morphosis, and Coop Himmelblau, among the most notable. Yessios maintains that the underlying mathematical geometry technique of the program is Void Modeling (Serraino 2002). “Void modeling offers the means to deal with architectural design during its soft and dynamically tentative states, while a design solution is still unknown and under exploration” (Yessios 1987). Drawing from the notion that certain classes of objects are containers of fluids, a digital object in Void Modeling comprises of a container and a contained part. The former is an envelope of 0 width on which different attributes can be assigned; the latter is responsive to modeling transformations on objects so that they yield satisfactory results (Yiessios 1987).

As a default, the graphical interface displays a single viewport containing a ground plane in a 30x60 axonometric view. Multiple windows can be opened to model objects saved on the same file. Extrusion is a key modeling principle for object formation. Broadly speaking, massing is built by outlining an open or closed 2D shape on the ground plane and extruding the resulting shape either to an infinite vanishing point (box-like) or to a finite point (cone and pyramid-like). From these entities, multiple entities can be generated through the derivative tools, giving a wide spectrum of modeling capabilities. Meshes and C-Meshes (Controlled Meshes), among others tools, add further capabilities to modeling, but even in those cases extrusion of those outlines gives massing to otherwise two-dimensional objects. Coordinates can be accurately typed when needed, yet precision is at the discretion of its users. According to the case studies here introduced, Form*Z tends to be an excellent tool to take the temperature—to speak—of a design under formation.

Although rendering and animation are available, the strength of the program rests on its modeling capabilities. Within the architectural world, Form*Z is intuitive enough to be picked up by architects, as it replicates in the digital realm what they are accustomed doing the most: physical modeling. This has made the application very popular in the Anglo-Saxon world, with peaks of users in Asia, and Europe as well. Ever since the launch of Form*Z 4.0, a new modular structure has been introduced to give users the option to be more selective in the modeling tools they want readily available according to their disciplinary domains. Additional plug-ins and the possibility to run scripts have further broadened the modeling capacity of the program.

7 3DS max

Former 3D Studio Max, 3DS MAX 5 is a solid modeler with extensive use in design visualization, animation, and rendering for film, telecommunications, and game developments. Most of its acclaim in the digital community is based on its ever-increasing potential to produce quasi photographic images and its flexibility in generating accurate animations. This polygon-based software offers extensive capabilities to achieve unprecedented realism through a distinctive graphical interface. In order to accomplish these results, a steep learning curve faces users due to the layout of the menus and the logic behind the modeling procedures.

Five different areas construct the interface of the application. The 3DS MAX default layout displays at the center of the
interface a view port configuration with four distinct views which provide navigation in the virtual space. Similar to the orthogonal projections of Monge, from the top left going clockwise there is a top, front, perspective, and left view. Surrounding the edges of the view, a series of tabs gives access to tens of thousands of commands (Metossian 2001). Drop-down menus, context-sensitive menus, rollouts, flyouts, floaters, and floating and docking toolbars are some of the ways in which the user can access the software power. Although the interface appears tightly structured, way-finding can be a daunting task. Commands are layered to the extent that it requires context-based extensive practice to take full advantage of the digital resources of the application. Modeling allows for the construction of parametric as well as non-parametric objects on ground planes. Generally speaking, outlines are drawn on a ground plan, then extruded, and further modified. Foundation of object creation to generate basic geometric forms are the cone, the cube, the cylinder, the sphere (Metossian 2001). These premade geometry sets are the stepping stone for assembling more complicated scenes. Additional levels of sophistication in the modeling process are available through the Create tab with extra options to include extended primitives, meshes, NURBS, patches, and more. However, modeling transformations can take numerous steps what the results they are supposed to yield. To perform a Boolean subtraction on compound object, for instance, takes six steps, as opposed to the same operation done in Form*Z, which requires only two steps once digital objects overlap.

Rendering and animation are two main winning points of the software. As opposed to Form*Z, whose primary focus is modeling, 3DS MAX offers a full array of possibilities to create scenes and add life to them. Texture mapping, the available material library, and lighting options provide wide range of choices to detail the model and add a high degree of realism to its virtual worlds. Import and Export procedures are available to take advantage of the strengths of other programs making the software either the environment where the model is fully assembled and rendered or a transitional program for its rendering capabilities. In the course of the proposed case studies, it will show that 3DS MAX is precisely used in the latter fashion. MAX scripts are also available to supply users with additional modeling choices.

8 Maya

MAYA is a node based program (Riddel 2003). This means that objects are represented through numerous nodes each editable interactively or mathematically. It is a rather resource-intensive application requiring a substantial amount of RAM and processor power, and a three-button mouse. Some of its features are: polygon-modeling tools, advanced character-animation tools, dynamics (particle simulations to reproduce real-world forces following the laws of physics), effects, Sub-division surfaces, Matchmoving, and more. These digital capabilities endow users with great control over the behavior and look of the form generated and its subsequent editing. Similar to the 3DS MAX, the MAYA interface captures in a rather condensed form menu sets all amounting to countless commands. The default layout displays four views, called panes: Front, Side, Top, and Perspective. The menu sets are available on the top left corner of the interface in the Status line, which holds selection functions. Directly below is the Shelf Menu displaying the most often-used tools. In each pane, the user can access additional menus. Commands and menus, just like much software, can be reached in many different ways.

A powerful built-in feature of MAYA is the Construction History. In the progressive refinement of a model, the discrete steps are digitally memorized. Should the starting point for the modeling change, the software applies all the transformations entered to that point. NURBS, polygons, and subdivisions are the primary primitives of MAYA. NURBS (NonUniform Rational Basis Spline) are equations defining curves or surfaces. Instead of tracing a curve through a set of points, the B-spline defines a curve sensitive to the position of the defining vertex points. The software capability to create smooth surfaces is a point of appeal to those designers interested in fluid forms. Once objects are generated, they undergo surgical transformations to accommodate interdisciplinary design intent. In addition, MAYA allows to run scripts to further extend its capabilities.

The primary user groups are film and video artists, computer game developers, and design professionals who use it to construct sophisticated digital imagery, animation, and visual effects. The program is designed largely for character and organic modeling. In architecture, a younger generation of designers—with either formal or informal associations with the architecture program of Columbia University in New York City—is at the forefront of MAYA users to identify alternative approaches to form making. However, due to the system requirements and its associated costs this software is used by a rather restricted group when compared to Form*Z and 3DS MAX. Nonetheless, the introduction of MAYA in the kit of digital tools is central for architectural discourse. Designs engaging the capabilities of MAYA have triggered debates in the specialized community and the general audience about form and its ramifications in the tectonics of artifact as well as its yet to be determined civic signification.

9 Six Case Studies

The proposed examples present similarities in how the adoption of computer technology has informed and influenced design decisions. In all cases design ideas were already formed prior to implementation of digital technology, yet software was pivotal in giving final formal expression to those visions. Here architects capitalize on computer visualization and modeling capabilities with a broad spectrum of attitudes concerning the release of control to the machine for the generation of forms. Although these six professional practices are different in scale of projects undertaken, design attitude, and office organization, the adoption
of computer technology has broadened the opportunities to explore design themes which are the hallmark of their individual signatures. Three case studies are based on Form*Z, one is a combination of 3D Studio Max and Form*Z, one shows a project modeled with Maya, and the last one presents an unconventional use of AutoCAD.

1 Wes Jones

In the personal trajectory of Wes Jones, former founding partner of the now dissolved firm Holt, Hinshaw, Pfau, Jones, the transition from hand drawing to digital output occurred in 1994. As he was shopping around for CAD programs on display at MACWORLD in San Francisco, Jones picked Form*Z. “We detected a congruence between our practice and the software assumptions,” he recalls in a phone interview. The imaginary world informing Jones’s architectural visions was pre-digital, without the computer being necessarily part of its formation. His fascination with mechanical parts, joinery, and the aesthetic of the machine constructed the rhetoric of his drawings and renderings and concocted a distinguished mechanical formalism (Figure 1).

Looking at the materials of that world was the motivating source of his designs. A comparison between the pre and post-digital portfolio exhibits consistencies of tectonic references, yet with one distinct characteristic: increased detailing (Figure 2). “For us there has been little if any change that could be ascribed to the software, so maybe it is just a happy accident that we chose a software right away that so perfectly fit our needs. . . . Form*Z was designed as a solid modeler for architects, and this is how we had always thought about design, i.e. stuff that could be modeled with chipboard and styrene and plex, so Form*Z has always been obvious to us in the same way—only difference is the added possibility for detail” (Wes Jones, e-mail message. 04 May 2003) (Figures 3 and 4).

Jones is himself an accomplished Form*Z user. His first hand exposure to computers goes back to his involvement in the military, where he became quite conversant using Fortran. During his graduate days at Harvard—he used Modelshop, a CAD package which is no longer supported. As physical model making was where he invested time in design investigation, the logic of Form*Z turned out to be a good fit with his way of working out designs. However, none of the parametric capabilities available today in Form*Z feed Jones’s work. He admits that he uses the Mesh Tool, occasionally, but modeling options such as the Metaformz primitives and editing are never implemented. As far as modeling is concerned the commands available in the earlier versions of the application were sufficient enough to assemble his digital landscapes. It follows that although the application has grown to incorporate more and more organic modeling commands with the purpose of conquering wider market sectors, the architect shows no interest in that direction. His work builds primarily on the notion of extrusion. Objects are first built as 2D shapes on the reference planes and extruded according to the available Object Types. Then the Derivatives palette supplies the

architect with the necessary tools for the geometry he is keen on. Jones laments the limited benefits of the current ray tracing algorithm. As soon as the scene has some advanced level of details, he maintains, Form*Z fails to deliver in that capacity. For Jones, architecture is an activity that goes far beyond simple form-making. It certainly is disposition of objects in space, but this is only one side of the story. In his view, architects have to come to terms with the public nature of artifacts. Computers—therefore—are not generative, but powerful companion to an articulate, precise, and already fully pre-determined vision of the world. However, computers also brought new challenges: the fast-pace of computer development triggers anxiety in users, who are constantly panting behind the latest in software and hardware development, hardly ever fully investigating the power of those tools. To echo some of the most recent words of renowned New York Times critic, Herbert Muschamp, “Architecture is not just sculpture blown up to urban scale. It is a social art, and for many, it will operate as a symbol of unwanted change” (Muschamp May 18, 2003). That is why Jones expresses no interests in programs like MAYA, which he feels are tools for the production of self-indulgent forms.

2 Thom Faulders: Beige Design

Thom Faulders is the sole principal of a small architectural firm called Beige Design based in Berkeley, California. Recipient of numerous design awards, Faulders undertakes the question of digital form and its physical construction. The use of computers comes in after well defined initial concepts have been formed. Three-dimensional exploration of those ideas is carried out through physical and digital models. For Faulders Form*Z offers a visualization technique, a way of testing a notion that precedes the engagement of computer technology. Yet, although substantial portions of the design are formally done directly on the computer, the design forms remain pictorial. “It is a bit like designing in perspective,” he says (interview with Thom Faulders 25 June, 2003). The transition from the pictorial to working drawings is still intensely problematic. Once a form has been produced, being able to make exact determinations regarding the dimensions of individual pieces is rather complicated.

The designer has some first-hand knowledge of Form*Z, however he relies on those working with him to implement ideas through the application. Having limited expertise with the general purpose modeler, he believes, might actually help to dodge some of its most obvious uses. This means stretching and bending part of the assumptions on how to construct shapes in Form*Z, to the point of defying codified uses or modes of streamlining the modeling process. Often this comes with the undertaking of supremely long tasks to carve broader digital scenarios. Faulders’s main quest is to develop complexity out of regularity. To account for the need of the construction industry to standardize building components, the architect investigates the intersection of unconventional forms and their factual feasibility. Here the software is not generative of form per se, yet it influences form, because it allows intentions to be tested. How
Three small-scale schemes address the notion of establishing micro modeling units, which produce elaborate spatial proposition through incremental changes and multiplications. Monolithic in their material expressions, they all carry the idea of cloning form out of digital cells. Faulders operates surgically on the individual components and its points of connection to sculpt fluid massing and evocative movements. Of these three laboratory moments, the first is a competition entry for the redesign of a parking lot, the second is an installation, and the third is a volumetric sign for the entry of a subspace within the California College of Arts and Crafts. Although Form*Z was not the starting point, the final three-dimensional outcome in all instances comes from the formal negotiation between the initial design idea and the available modeling tools. Volumes could be distorted to create effects, Faulders adds, yet it was impossible to do it in a measurable way. The constant negotiation between extreme economy of means and the determination to build forms foreign to everyday architectural vocabulary without the costly technology of CAD/CAM brought about a hybrid design-build process.
Suspended Field is the proposed vision for a parking lot intervention in Cincinnati, Ohio. A veil of plotted points floats on a parking layout to map different conditions of use. Lightweight balls held through cables determine cognitive moments, which help to navigate the space and by virtue of their mutual position effect multiple visual experiences. Multiple sections were drawn to determine a mesh on whose vertexes the balls were individually placed (figures 5, 6, 7).

Particle Reflex is an installation for the San Francisco Museum of Modern Art commissioned in 2001. Here, Faulders confronted the constructability of a fluid form. Repeatable panels and adjusted increments were the ingredients of a strategy designed to make a commitment to the tectonics of this unique shape. Once the volumes have been modeled in Form*Z using meshes, the designer gets into the specific of each panel and adjusts it individually to assemble the conceptual surface (figure 8, 9, 10).

Wattis Entry reflects a further refinement in generating atypical planes through stitching construction units each customized differently from its adjacent ones. Form*Z proved to be ineffective in manipulating surfaces in a controlled fashion. After several attempts, Faulders drew the individual sections according to a predetermined design idea. He then transitioned to the digital by precisely drafting the specific outlines and transfer them in an orderly arrangement into the Form*Z environment. From there, he skinned the sections to have a smooth shell. This allowed him to determine the exact size of the discrete elements. In this case the application was stretched to do what was a rather specific mental process, yet not immediately feasible using the conventional tools of Form*Z (figures 11, 12, 13, 14).

3 Tom Leader Studio
A former partner of Peter Walker and Partners, landscape architect Tom Leader embarked in a new professional venture with Phillipppe Coignet in 2001. The distinguishing theme of their practice is the examination of site-specific resources and their stratification through time. Leader's biological reading of Rome in 1999 is the starting point for design ideas feeding his later work. In this provocative approach, the landscape architect looks holistically at the transformation of the ecosystem of Rome through the centuries in the course of 3300 years. By slicing time every 200 years and juxtaposing over a dozen of historical transparent plates, Leader generates an analytical artifact of unparalleled critical clarity to grasp in one moment centuries of earth development and land use.

Navigating space through time and mapping the site components via situated readings of the land underpin the current portfolio of the studio. Form*Z is the vehicle for this journey of historical investigation, where these temporal sections are modeled. Each element is assigned a surface style with distinctive color and transparency to make this information readable, a digital see-through to gain insights on its stratified formation. In plotting and superimposing the geological and human-made stratifications determining the current state of a site, Leader defines a new approach toward land use and
an instrument of civic awareness regarding the environment. Coignet admits that, like Faulders, it is rather hard to manipulate precisely the model in Form*Z to produce accurate and measured transformation, especially when the topological elements are being moved parallel to the reference planes. Vertical changes, instead, are a bit easier to implement.

The competition entry for the Fresh Kills Landfill in Staten Island, New York, makes manifest the notion of sectioning the site through time. The competition was launched to rethink the role of one of the greatest plots of land for the collection of garbage in the New York City area. As the competition was taking place, the terrorist attack of 9/11 struck the World Trade Center. The debris was collected in the Fresh Kills Landfill, giving the site a unique symbolic dimension in American history. Leader’s scheme brackets a 150 year-time period: 100 years to the present belonging to the past of Fresh Kills, the remaining projecting layered tendencies into the future. Rather than choosing a prescriptive strategy, the team conceived of a framework, where diverse possibilities of use can take place still within the identity of the site (figures 15, 16, 17, 18, 19, 20). Form*Z comes into play as a way of testing the spatial implications of overlapping sequential maps of the site, an unexplored form of cartography to bring to the fore the dynamics that make the place what it is today. Layering becomes a mechanism to achieve new consciousness of how functional transformations can unfold within a coherent development of
diversity. Mapping the local ecosystem, urban debris and its impact on the site’s surface, its transportation system, its physical edges, and the composition of the garbage come together in a composite image. Fly-thru animations grant the studio vantage points to assess the dynamic reading of the site, which through public participation would determine the future development of its destiny. This case study presents a rather uncommon approach to mainstream use of the software, yet it offers a critical attitude toward modeling as an opportunity to visualize the socio-cultural and geological forces that make any place the physical reality that it is.
Figure 11. Sequence 1

Figure 12. Sequence 2

Figure 13. Final Layout

Each panel is symmetrical about its vertical axis. This allows for the emergence of a 3-dimensional curve upon final assembly.
For Benjamin Ward, a Gensler associate, 3D Studio Max is a powerful rendering program. Modeling, however, presents procedural complexities that in his opinion are absent in Form*Z. “It takes more steps to perform a Boolean operation in 3DS Max than in Form*Z”. Despite his expertise in both, Ward deems 3D Studio cumbersome in its modeling capacities due to the organization of the interface, its hidden palettes, and how deep the user has to go to get tools than are critical since the inception of digital modeling. High detailing in 3DS MAX comes with large investment of modeling time. For the Changi Airport scheme in Singapore, Ward used a combination of Form*Z, 3D Studio, and AutoCAD to transition from modeling, to rendering, to measured drawings. Each view was modeled separately to endow the scene with high level of details without encumbering the entire model with unnecessary polygons. Quick studies were done in Form*Z since its internal logic allows designers to explore forms without any significant commitment of dimensions, yet in a rather efficient fashion. Ward initially attempted the same in 3DS MAX, but that proved to be harder than initially estimated.

Great attention was paid in the modeling of the canopies, which was achieved through axial sweeps along one path. Derivatives and Boolean operations were critical to develop the imagery for the skylights (figures 21, 22, 23). But when it came to study the material quality of its surfaces, their reflections, and the lighting, 3DS MAX demonstrated its power. The Form*Z model was imported in 3DS MAX, materials were assigned, and different light sources were positioned. Lighting studies helped to make further determination on the overall architectural effects and materials to be used. Accuracy
Figure 23. Changi Airport. Reflected Ceiling Plan

Figure 24. Changi Airport. Canopies Detail using Vray

Figure 25. Changi Airport. Detail of Skylights

Figure 26. Changi Airport. Interior

Figure 27. Framework of lines for the lower level

Figure 28. Framework of lines for the lower level

Figure 29. Mapping the victims
in lighting simulations were pivotal in the final arrangement of the ceiling structure. In particular, Vray, a plug-in for 3DS MAX, yielded impressive results in the shading of the building components. Global illumination alone generated an image virtually indistinguishable from a physical model (figures 24, 25, 26).

Hopping from one application to the next is customary practice at Gensler, where software is used for what can bring the fastest results with the highest level of accuracy. The world of forms available in Form*Z seems enough to model some of the design choices at Gensler. However, only a few individuals are fully proficient with 3D-modeling. Once the massing is established, it is 3DS MAX that takes the lead in software use and informs the design choices of the office.

5 Cory Clarke and Jason Vigneri-Beane

Cory Clarke is a designer and an instructor of Advanced Digital Design courses at Columbia University in New York City. His proficiency in MAYA and Form*Z (he is co-author of one of the first popular Form*Z tutorials), is a critical vehicle for making informed decisions about software use and design scope. Being aware of the pros and cons of each application, Clarke and Vigneri-Beane adopted a combined approach in modeling by using both MAYA and Form*Z for different portions of their competition entry for the World Trade Center Memorial Competition held in May 2003 to honor the memory of those who perished in the 9/11 attack.

The choice of adopting both programs is directly correlated with the perceived capability to control object formation in each software. The central idea of the scheme is to map on the site the individual trajectories of the victims and their common fate. Personal information of each casualty provides the raw data for the construction of a continuous hardscape. For every life lost in that tragedy, an area with specific physical properties is identified and stitched to adjacent parcels representing other individuals. Once common rules have been established, the resulting landscape gives scale and identity to a global loss of human life. The memorial turns into a site for participatory mourning as well as a space for the experience of private pain. The proposal is articulated into two layers. Memorial markers standing for the silent symbols of the individual deaths dot the lower level.

Each marker’s position establishes a correlation between the victim and the location of his or her death. Directly above is the undulating public park, a place for gathering and spiritual renewal (figures 27, 28, 29, 30).

To give tangible form to this tragedy was the challenge that the designers undertook using tools that could precisely translate tables of personal data of the victims into a physical artifact. Cory Clarke states: “The reasons why we choose MAYA over any other software are because of (1) MEL, the MAYA embedded language, (2) it is a procedural modeler, and (3) it is a NURBs surface modeler” (Cory Clarke e-mail message, 2 July 2003). The first capability made it possible to convert the database of the victims’ profiles into a two-dimensional map through custom made scripts. The next step was the extrapolation of the floating landscape out of the map using NURBs. What makes MAYA so powerful compared to other programs offering NURBs is the creation of seamless fluid objects out of discrete pieces of information in a controlled, editable fashion. As a procedural modeler, the user enters a set of instructions generating previously undetermined form. As the commands are entered, they are remembered in the machine, building up the construction history palette. Once a sequence of commands has been processed, it is possible to go back into that linear path and change the specific instructions. All subsequent modifications would be traced down to its final form. This feature allows for an iterative process, trial and errors, to reach a satisfactory form for the design intent.

For the landscape portion of the project, the process was converting data into a map through MEL and then the map into a surface through NURBS. MAYA granted critical flexibility to achieve desired results about the character of the floating surface. The lower layer comprising the matrix of lines and the memorial markers was modeled in Form*Z. The section of the WTC Memorial designed in Form*Z was then exported in MAYA and then rendered to its final stage. For more basic shapes, ironically MAYA performs less competitively than the general purposed modeler. It follows that choosing MAYA as modeler brings about a set of forms influenced by NURBs. Fluid, pliable shapes are relatively easy to achieve, but modeling and editing primary shapes, according to Clarke, can be more problematic. Designs generated in MAYA will likely exhibit geometrical traits veering away from assembling known sets of Platonic shapes: it will feel more organic (figures 31, 32, 33, 34).

6 Skidmore, Owings & Merrill LLP (SOM), New York

High-rise design is a forte in the design competence of SOM. From the heyday of the mid-century classic Lever House on Park Avenue in New York to the more recent accomplishments disseminated throughout the globe, SOM continues to explore this building type, adding to the skylines of the built environment. Skin and massing are two central concerns in the urban characters of a great many of these vertical statements and constant objects of design explorations. CAD programs were already implemented in the SOM practice of the early seventies.

---

Figure 30. Mapping rules
Specifically the firm had developed a proprietary drafting program called “Draft.” Later in a collaboration with IBM, the application changed its name to “AES” (Architecture and Engineering Series). In more recent years the office has switched to AutoCAD. Used often as production tools connected through intranets that allow file sharing by design teams and as marketing devices to attract increasingly demanding clientele, computers in the SOM working routine have recently come to cover a more central role in determining architectural images as well.

A case in point is the competition entry for the Central Bank of Kuwait (CBK) Tower in Kuwait City designed in May 2003. “Local environmental influences is a central principle of this design” states Anthony Feldman, SOM Associate Partner in charge of this project. The architectural interpretation of site identity...
and cultural context provide the design guidelines to establish the final form. The result is an inside-out skyscraper carrying a unique interpretation of a traditional Middle Eastern courtyard (Figure 40, 41).

On a square plot measuring 45m x 45m, the designers extrude a 185-meter tall tower with a 20m x 20m inner courtyard whose walls are plumb from the top to the lower most levels. The glazing system stands for only 10% of the outer wall to protect the environmental quality of the indoor space. A script run in AutoCAD 2002 generated the apparent randomness of the lighting pattern of the outer wall according to five rules:

- Each window could be placed at one of four elevations within the typical 5-foot by 14-foot “unit”: at floor-level, seated eye-level, standing eye-level, or ceiling-level. (All openings were placed to avoid conflict with the floor “sandwich”.)
- Two windows must occur in each “unit.”
- Six horizontal units are then grouped into 30-foot wide “panels,” constituting the structural pre-cast construction module; two out of six units within each panel must remain devoid of windows to allow for vertical transfer of the structural load.
- Five total panel types are repeated throughout the tower wall to create economy within a seemingly “random” system.
- No more than three same-elevation windows could be placed adjacent to one another for structural considerations (figure 37).

What distinguishes this project from mainstream high-rise building design is the unconventional use of AutoCAD to generate the fenestration pattern of the exterior envelope. The formal reference is the Mashrabiya, a distinctive type of wood carving particular to the Arab East mainly used for windows, which allowed cool breezes to enter homes in the heat of summer. To capture the mood of that regional motif, the designers turned to the computer for the iterative process...
Figure 38. Sequence 1
Figure 39. Sequence 2
Figure 40. Sequence 3
Figure 41. Sequence 4

Figure 42. Sequence 5
of finding a systematic approach in punching the windows on the buildings skins. Feldman came up with the idea of using computers to generate those iterations and established the formal rules for the arrangements of the windows. Neil Katz, an in-house expert in computer graphics within the SOM design department, wrote the code in AutoLISP.

Katz recalls: “Anthony approached me with a problem I found extremely interesting—developing an elevation based on certain criteria: a random appearance; relatively small openings (based on the climate in which the building is located); and modularity. Anthony and I worked together to develop the modules: squarish bays, with a band of window either horizontal or vertical, and located either at the center of the bay or at one of the edges: for the ‘vertical’ bays the window can be at the left side, in the center, or at the right side of the bay; for the ‘horizontal’ bays the window can be at the top, in the center, or at the bottom of the bay” (Neil Katz e-mail message, 25 June 2003).

The following is the pseudo-code use to determine computer-generated randomness:
-----

loop:for each floor
    loop:for each bay
        get-random(1 to 6)
        place-bay
    next bay
next floor
-----

Through progressive stages of refinement and post-rationalization for constructability purposes and economic considerations, the elevation achieved its final form (figures 38, 39, 40, 41, 42).

Directly mapped onto the outer walls, the computer generated distribution of these formal rules is converted into an architectural chart guiding the punching of the windows. Although software-driven, the resulting tapestry of lights is evocative of the glittering glamour of New York City skyscrapers, yet responsive to site specificity: a felicitous tangible intersection of ingrained design practices and computer aided design (figure 43).

10. Conclusion
Practicing architectural design with any three-dimensional modeling software is partaking in a technology of spatial ordering put into effect by the software developer. If these built-in slants are ignored, embracing technology for its own sake could lead to an uncritical stance both on the behalf of the users as well as on the critical assessment of those designs. The celebration of particular propositions in the press and in educational environments customarily groups propositions of very disparate architectural intentions under the general umbrella of digital design. But since architecture is a human activity that engages collectively many destinies propelled by hybrid interests, it might be worthwhile to go beyond the fact that whimsical forms can be built and raise the critical bar of what civic identity entails for the urban communities of the twenty-first century. As applications come and go, design obsolescence goes hand in hand with how fashionable morphological manipulations are. Additional case studies and quantitative tests are needed and quantitative tests to measure the rate of frequency of form elements yielded by specific software designed to further corroborate the statement that as of today form follows software.
References: