00.01

Coding landscape

Contributors:

Bradley Cantrell
ASLA, FAAR, Professor and Chair of Landscape Architecture, University of Virginia School of Architecture

Adam Mekies
PLA, Associate, Design Workshop Inc, Aspen Colorado

Developing syntax

> **Coding:** To express in syntax a set of operations so as one or a system of computations may be made.[1]

The discussion and implementation of code is an undeveloped discourse in the profession of landscape architecture, yet that seemingly arcane world of computation may not differ from other disciplines or techniques, at their nascent origins, which ultimately extended the agency of human societies.

The act of coding is like authoring a well-written work of fiction. The organized collection of words, in a particular language, frames the story line and describes the characters, actions, and context.
The algorithm of a computer program is analogous in its reasoning. The relationship of variables, inputs, parameters, and results forms a story based on the sequence of events. Computing or "the computer" is the reader of this story, compiling the words and interpreting the code that forms the narrative. In this vein, change one key action or location, or replace a key character, and the story unfolds in an entirely different way.

Code and algorithms are discussed in the daily milieu of contemporary culture, remotely related to their original computer science definition. This discussion of code can take place in one of two types of languages, as Paul Coates describes "natural" and "artificial" languages.

> Natural languages have developed over the last 100,000 years or so and are, of course, based on the way we inhabit the world with other people. Natural languages have unknown syntax and the lexicon is subject to at least some natural drift and development. Artificial languages have an explicit syntax and well-defined lexicon.
>
> Using these artificial languages, one can define algorithms – one class of algorithms is those written in computer code. Computer code is a very particular kind of text. It is designed to be readable for humans, after training; in this it is much the same as natural languages – no one would expect to be able to read Proust in the original French without learning French beforehand.
>
> Coates[2]

It is therefore important that we establish a theoretical and practical underpinning to guide the shifting syntax of landscape architecture and computation: the discussion of "code" as a syntactical language and heuristic process that we push for computational design to become a subject of thought and common language in landscape architecture, to promote new ecological, social, economic, formal, and material design systems in the built environment.

Early history of computation

Computation and the quest for a machine to facilitate complex calculations can be traced to centuries before the Common Era. Various computational achievements can be attributed to the Chinese, Babylonians, and Greeks, among others.[3] The first modern mechanical computer, the so-called analytical engine, was developed in 1834 by Charles Babbage, a British mathematician, engineer, and inventor.[4] The analytical engine merited the term "computer" as it adhered (in retrospect) to the basic principles of today's mechanical computers.

The foundation of the modern computer was soundly established by a British mathematician and scientist, Alan Turing, in 1937 in his seminal paper on "Computable Numbers."[5] Apple's Steve Wozniak believed that Turing set the standards for modern computation: in his keynote address to the 2012 Turing Festival, Wozniak said that "Turing came up with what we know about computers today."[6]

FIGURE 0.1.1 Computers in 1942. For the first half of the twentieth century, finding their roots during the Second World War, "computer" was a job description, not a ubiquitous machine

Source: Public Release National Archive[9]

The term "computer" has been in use from the early seventeenth century (its first known written reference dates from 1613) and meant "one who computes,"[7] referring to a person executing calculations; this was of course prior to electronic computers becoming widely available. For Turing's contemporaries, computation, or computing, meant getting as many people as necessary to complete a task in as short a space of time as was possible. The use of a machine to complete human tasks was a new concept of the time, one society still struggles with in new ways in contemporary culture. Much of Turing's work investigated the potential of what could be computed by machines in place of their human counterparts.[8]

Around a hundred mechanical computers existed in the world in 1953, capable of making hundreds of calculations a day. We now take for granted that billions of calculations are made per second and that technology is further advancing in exponential strides as we now enter the world of quantum computing and qubits.[10] Gradually the world began to take advantage of inventions and replaced the "human calculator" with the mechanical machine. However, just as with electronic compilers (translators), in landscape architecture we still struggle with the translation and abstraction of thought processes to machine language.

The frustrations we still face to this day were just as salient at the earliest stages of military and top-secret computing. Similarly, these recurring frustrations were dealt with humor, even at the NSA (National Security Agency), as illustrated (below) in the monthly "Techniques and Standards" bulletin.[11]

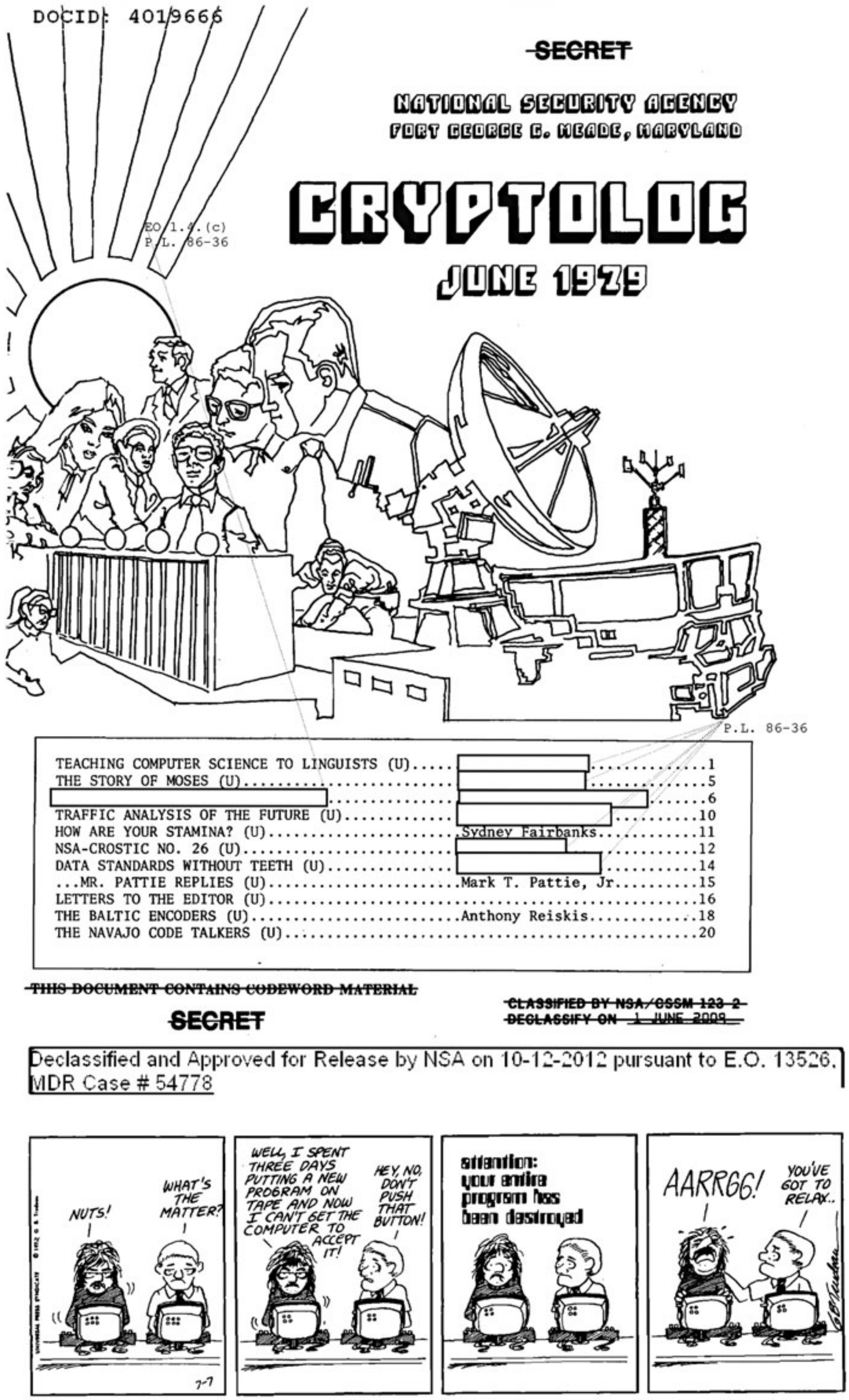
Emergence of computation in landscape architecture

The origins of computation, from our perspective as designers and planners, emerged first in the 1960s with new thought processes in analysis and environmental planning. This approach is perhaps best explained in 1967 in the seminal paper "Design with Nature," by Ian McHarg, an approach now referred to as "McHargian Analysis." Mcharg's explanation of an overlay system for land classification, coupled with much of the work done and courses taught by Carl Steinitz at the Harvard Graduate School of Design, established a basis for the development of modern GIS (geographic information systems).[12]

In 1965, Chicago architect and Harvard Graduate School of Design Architecture alum Howard T. Fisher, created the Harvard Laboratory for Computer Graphics and Spatial Analysis. There, supported by a major grant from the Ford Foundation, Fisher further developed GIS, which spun off a number of computer applications and integrated mapping systems, including tools such as SYMAP (Synagraphic Mapping and Analysis Program), with the ability to print contour maps on a line printer.[13]

These initial forays into GIS and related tools were initially speculations in computation and mapping but were eventually developed into commercial software applications and hardware implementations. Hardware purchasers would have access to free software, which the users could also develop for their own specific needs. Fisher's pioneering ideas, in turn, inspired Jack Dangermond, then research assistant at the lab from 1968 to 1969, to put these ideas to practical use. Dangermond's start-up company, ESRI (Environmental Systems Research Institute), was founded in 1969, focusing on software for land use analysis.[14]

In the early 1970s, computation in landscape architecture focused primarily on a two-dimensional understanding of data and mapping overlay. It was not until the late



DOCID: 4019666

~~SECRET~~

NATIONAL SECURITY AGENCY
FORT GEORGE G. MEADE, MARYLAND

CRYPTOLOG

JUNE 1979

EO 1.4.(c)
P.L. 86-36

P.L. 86-36

~~THIS DOCUMENT CONTAINS CODEWORD MATERIAL~~

~~SECRET~~

~~CLASSIFIED BY NSA/CSSM 123-2~~
~~DECLASSIFY ON 1 JUNE 2009~~

Declassified and Approved for Release by NSA on 10-12-2012 pursuant to E.O. 13526, MDR Case # 54778

FIGURE 0.1.2 ***Cryptolog*** **magazine, June 1979 – Cover and Cartoon**

Source: Public release, National Security Agency

1970s that three-dimensional computation expanded, gaining more traction not only in research institutions and government agencies but also in the entertainment industry.

The late 1970s saw government entities, such as the National Forest Service, take a deeper interest in nascent landscape architecture computational techniques of visualizing and documenting large landscapes and forest lands.[15] Built originally to monitor forest harvesting and annual forest fire behavior, these simulations encountered many of the same level of detail (LOD) challenges we face today in modeling and visualizing large expanses of vegetation.[16] The entertainment industry, both film and television,[17] explored the capabilities of computer graphics during that period of time. Early projects, however, struggled with budgets and especially story lines that did not expand beyond the "novelty" of computer graphics. Film director George Lucas pierced through those obstacles. In 1979, Lucas created a special computer graphics division for his company. It was in this environment that researchers had access to funding but more importantly guidance from a serious producer with "definitive goals."[18]

> A special effect is a tool, a means of telling a story. People have a tendency to confuse them as an end to themselves. A special effect without a story is a pretty boring thing.
>
> George Lucas, 1983[19]

Accessibility of computation in the private practice of landscape architecture

The first commercially accessible computers for the masses expanded rapidly in the 1980s, and with that hardware expansion software development would soon follow at an ever-increasing rate. In 1982, Autodesk, founded by John Walker, launched its first version of AutoCAD.[20] AutoCAD, to this day, is one of the most heavily used programs for detailed design and drafting in landscape architecture and other design and engineering fields. That same year, Dangermond's ESRI finally launched Arc/INFO, its first commercially available GIS platform. Arc/INFO remains the leader in large-scale planning and analysis work in landscape architecture.[21] Both of these tools, from their early creation, have been dominant in their use in the landscape architecture profession for the last 35 years.

Only recently have the detailed drafting and 3D world of CAD (computer-aided design) and the analysis and large-scale data platform of GIS truly started to merge in the software approach of "geo-design." Perhaps popularized through the first geo-design summit in January 2010, the ideas of geo-design codify the challenges in scale and complexity of landscape computing when shifting scales of models are required from regional ecologies, to civic spaces, to the visual presence of the virtual "wild."[22]

Early innovation in design computation often occurred in specialized studios that focused on the implementation of technology for specific scenarios or for moments within the design process. Overly specialized employees focusing on the development of experimental technology are a strain on the bottom line of traditional design practices when existing outside of two conditions: 1) the development group is increasing efficiency of specific billable tasks within a contract structure that allows for increased profit margins; and 2) the studio is developing marketable products or services that expand the existing scope of services that can be obtained by the design practice. Often the specialized studio exists as a marketing platform, a side hobby of specific design staff, and/or a "think tank" to explore opportunities within a traditional practice.

All are valid approaches in landscape architecture, but owing to the financial draw on firms, this specialization has historically lacked the resources to inflect great change in technological development.

As with the emergence of ESRI and Jack Dangermond from the work at Harvard Laboratories, or contemporary collaborations with labs such as the MIT MediaLAB with Sasaki Associates, applications of new technology often take the form of university collaboration or consulting partnership. Hence, with the concurrent need for both special skills and investment, a historical division in design and landscape architecture developed between the business models of toolmaker and that of tool users.

A case study interview in landscape architecture technology investment and the lessons to be learned

In the 1980s the young firm Design Workshop, led by co-founders Joe Porter and Don Ensign along with partners Richard Shaw and Kurt Culbertson, saw an opportunity for what would become one of the pioneering practices in early environmental simulation technology. These valiant but ultimately failed incursions into the technological market prompted the first crack in a schism between the landscape architect as a "creative" designer, and the landscape architect as a "technician."

> The real reason those efforts failed was the disconnected relationship of the individuals running the simulations. There was too much interest in simulation for simulations sake and not for solving the real-world problems.
>
> Culbertson, pers. comm.

The early 1980s was a time of need for new tools and new methods in landscape architecture. In 1982 Design Workshop reached out to Lucas Films in search of a consultant for assistance in visual simulation. The response included a plan far too expensive for the services required and stakeholders' budget. At the time, the stakeholders consisted primarily of western communities in sensitive ecological and visual contexts, large-scale land developers/resorts developers, and government entities such as the National Park Service or The Federal Bureau of Land Management, which were interested in these large-scale visual resource assessments. Many of these early efforts in natural and visual resource assessment built on the early work by Carl Steinitz and the Delphi process.[23] These early goals of open lands and "natural landscapes" simulations were in stark contrast to the majority of early simulation work being done in architectural or urban contexts, observed Joe Porter, retired principal and co-founder:

> I recall one of the few other groups we could see doing these simulations at the time was SOM, but these were all building focused, not the large-scale simulations we needed help with.
>
> Porter, pers. comm.

The ensuing conversation and recommendation from Lucas Films led Design Workshop to invest in an early Iris computer, which resulted in an investment of roughly $45,000; the Design Workshop Byte Cave was born. Beyond this initial investment, Design Workshop spent nearly $4,000/month on a consulting staff to operate and program the Iris computer system for the firm's ongoing projects. These are significant investments for a young landscape architecture firm, particularly considering that this was in the early 1980s.

> We didn't start the BYTE CAVE for marketing purposes at the time.

> We started it because we thought we would be better designers. We were designing large landscapes with complicated forms and relationships; we needed a way of seeing.
>
> Porter, pers. comm.

> At this time there were really three paths to invest in, and I recall discussing which one it should be. There was CAD and GIS that we basically could see were going to come anyways, and we would have to incorporate. It was the 3D simulation work that no one in the profession was doing at the time.
>
> Porter, pers. comm.

Complex three-dimensional simulations were integrated in the formation of numerous projects. Artists' renditions were given a new approach for the first time with an accurate geometric base to draw over. With the advent of the 3D computer wire frame, studies could be done combining the best of both worlds: creativity in design and efficiency and precision in visualization. Computational accuracy in digital simulation and artistic composition were now firmly in the hands of skilled landscape architects.

> The entire purpose at the outset was accurate representation. The world had changed to a point which demanded the honesty.
>
> Shaw, pers. comm.

Even with great successes in early efforts of simulation at the business level, at the survival level the balancing act between the needs of the landscape architecture studios and the prowess of the computer whizzes was difficult to merge with the need for efficient solutions. The critical nature of landscape architecture and the problems the discipline is addressing are often overshadowed by the computer graphics that gloss over the technical challenges posed by the public and private built environment.

> It was one day in the office someone called "Joe, Joe you need to see this." When I walked over the individual had a dinosaur standing in a street and you could see a car driving by reflecting in the dinosaur's eye, and I realized we were spending money on things we really didn't need. It was at this time (around 1993) that we sold the IRIS computer and began investing in putting individual computers on the designers' desks. It took time but eventually over the years there was a computer on everyone's desk in the office.
>
> Porter, pers. comm.

Implementing new technological avenues is always a challenge in established practice. To be "the first" is a common struggle[24] as explored further in the first essay, "Computation in Practice." The investments in technology we make can be a significant pull on a firm's revenue. Therefore, it is imperative that the technical support or innovation staff communicate effectively with the design team and stakeholders. The critical discussion of innovation for innovation's sake on the one hand, and creative design on the other hand, requires a careful balance in business practices according to Kurt Culbertson, CEO of Design Workshop: "This is the choice between being on the leading edge versus the bleeding edge, in practice."

To this day and anticipating the increasing prevalence of machine learning, the need for effective communication between the computational capability and defined human goals will remain paramount.[25] At Design Workshop, it was the Byte Cave "machine" that had a mind of its own; other firms faced similar struggles where the computer specialists and the designer, or seemingly the software itself, had a different understanding of the same challenge at hand.

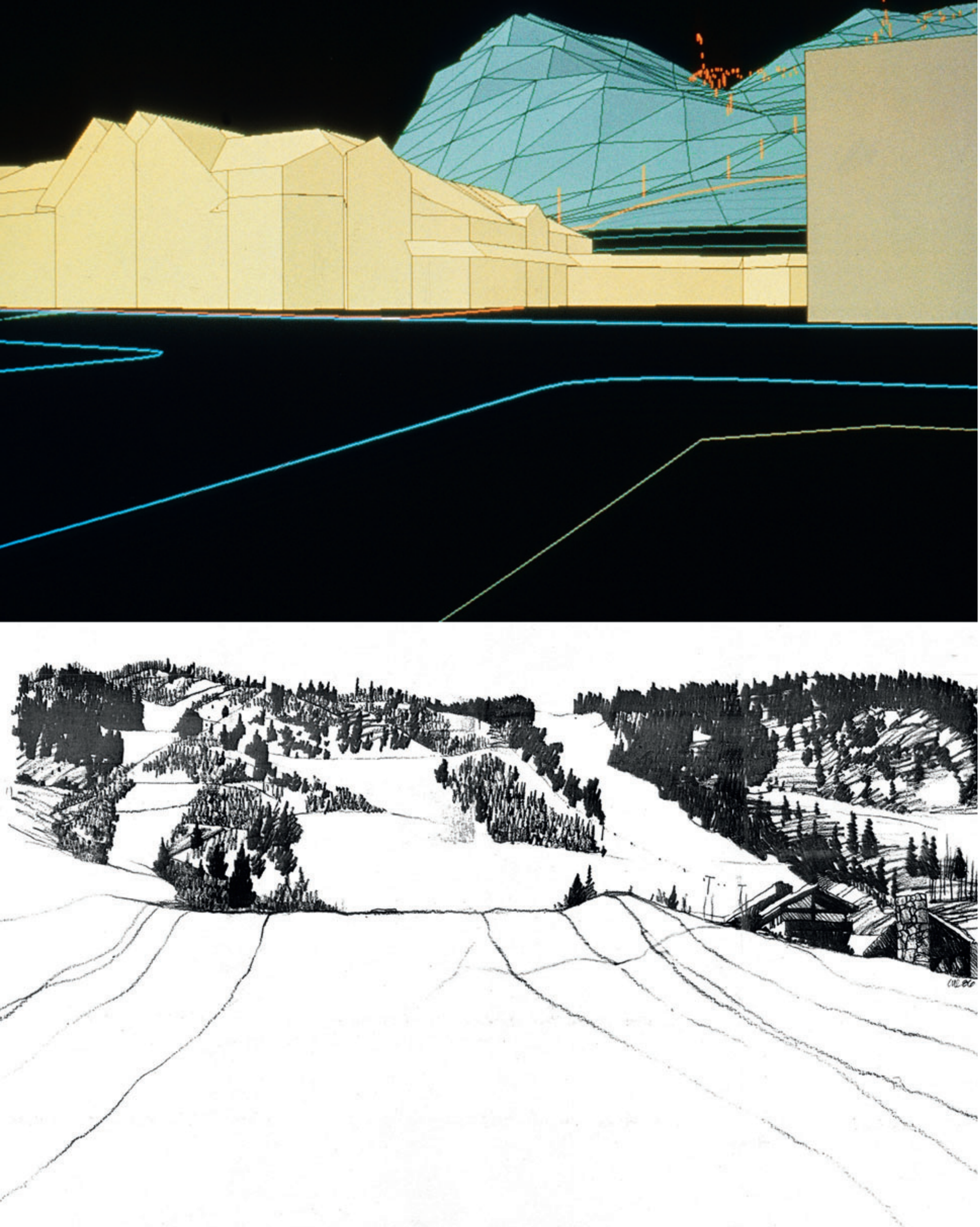
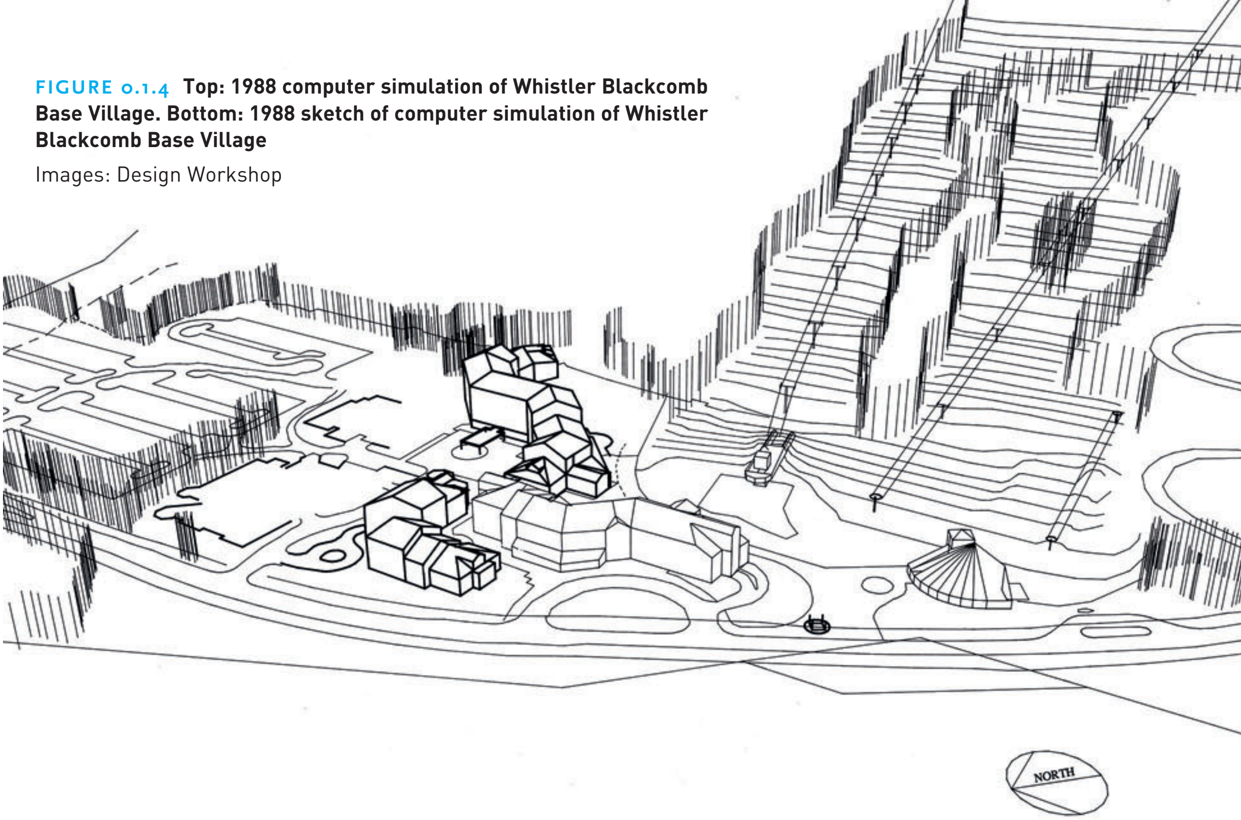


FIGURE 0.1.3 **Top: 1986 simulation of the Little Nell Hotel in Aspen, Colorado. Bottom: 1986 sketch of regrading – as viewed from the simulated or proposed hotel deck**

Images: Design Workshop

FIGURE 0.1.4 Top: 1988 computer simulation of Whistler Blackcomb Base Village. Bottom: 1988 sketch of computer simulation of Whistler Blackcomb Base Village

Images: Design Workshop



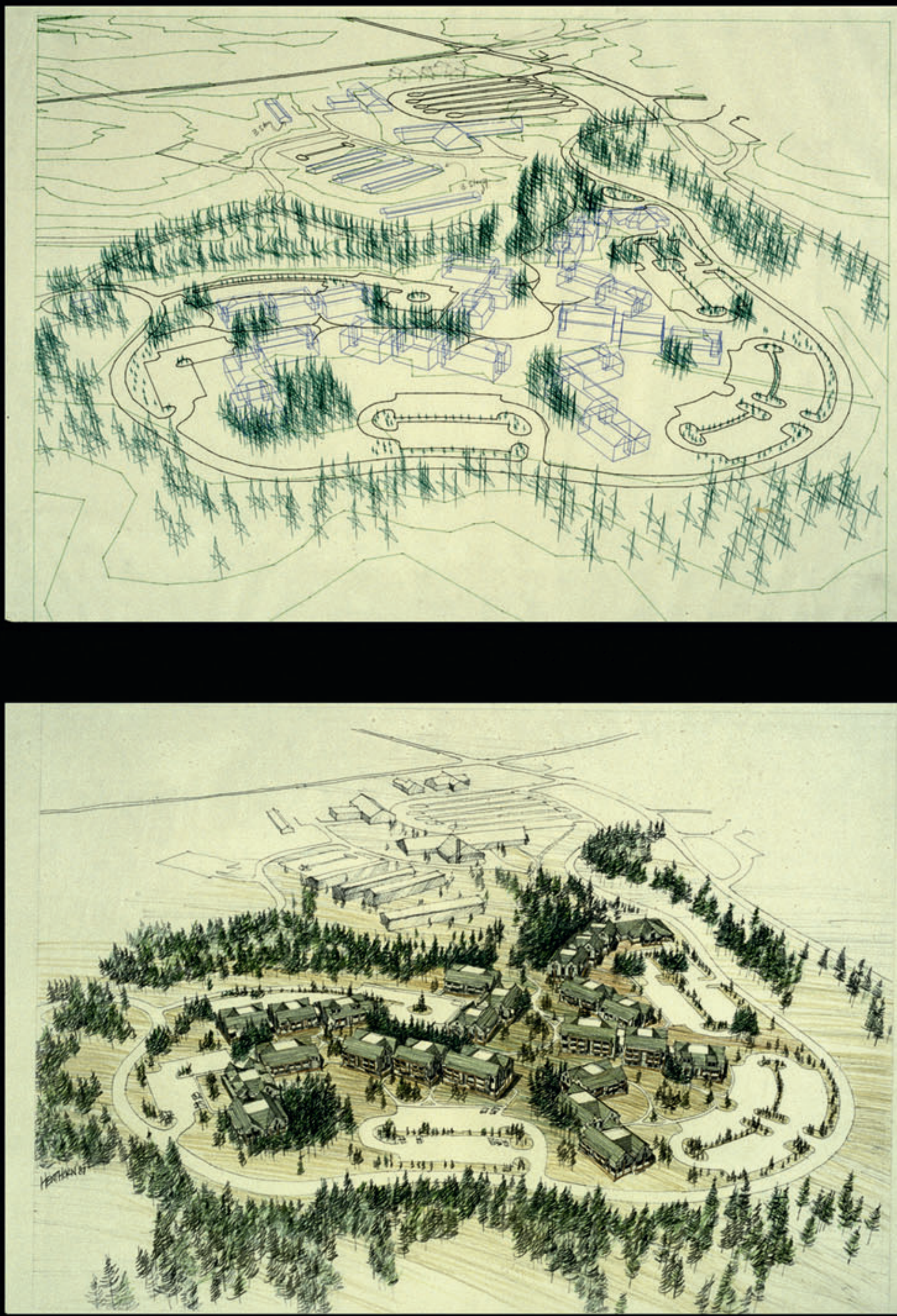


FIGURE 0.1.5 Top: 1989 wire frame computer simulation of Canyon Village (Yellow Stone). Bottom: 1989 sketch of top of wire frame of Canyon Village (Yellow Stone)

Images: Design Workshop

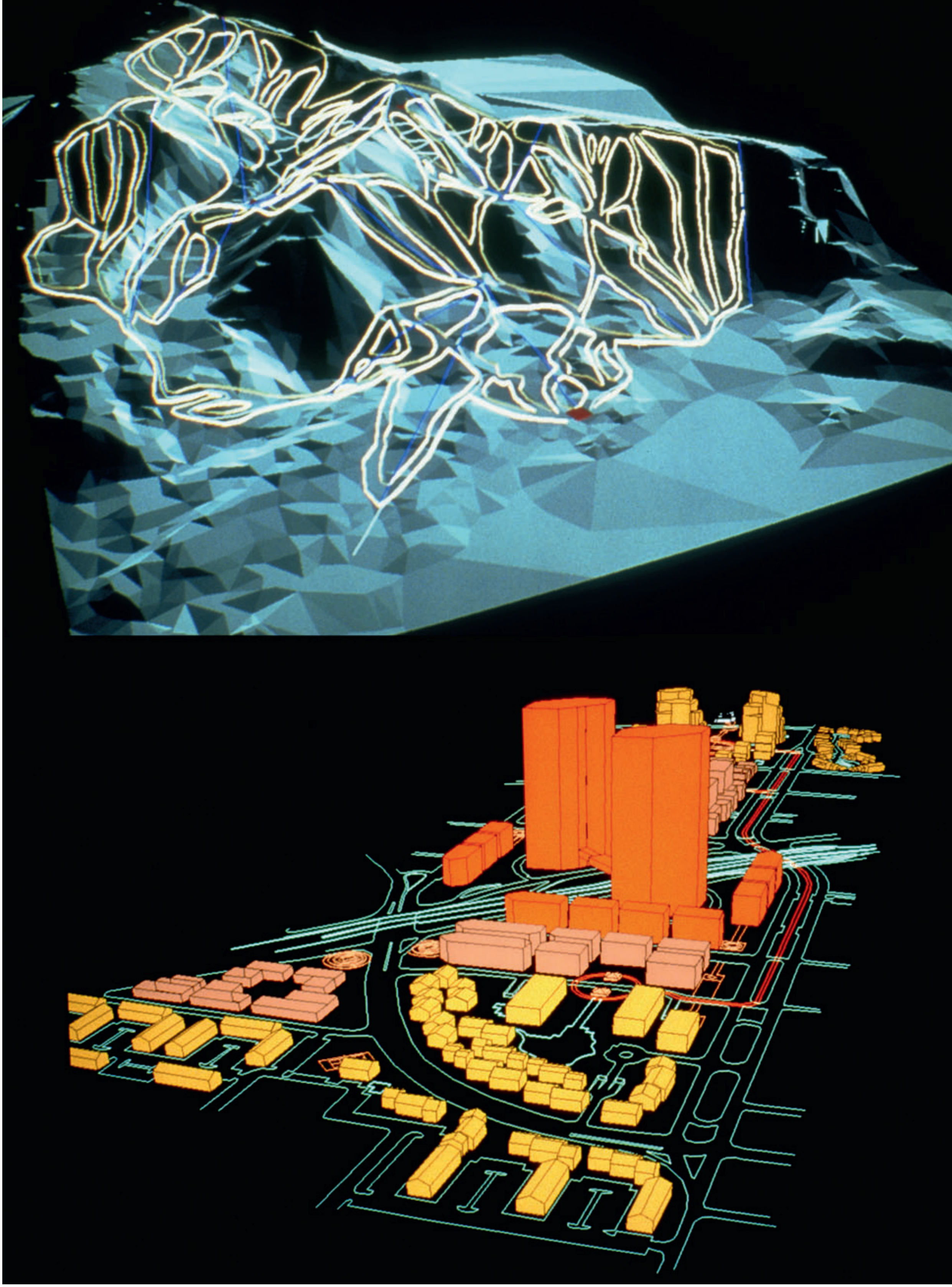


FIGURE 0.1.6 **Top: 1990 3D massing model of Canyon Village (Yellow Stone). Bottom: 1991 screen capture of 3D fly-through video created for HKS Architects of City Place Development Proposal (Dallas)**

Images: Design Workshop

In landscape architecture, the struggle remains one of translation between the tools of computation, with all their prowess, and the creativity of the designers, with all their imagination—whether that struggle implies two separate individuals or one with an inner struggle, the syntax of communication is the bridge to be crossed.

New paradigms

Computation inherently asks us to define elements of landscape architecture: associated characteristics, rules, actions, and relationships that form the model. Landscape architecture strives to understand the interrelationship of multiple, not inherently formal but synthetic models that define methods for designing living systems utilizing data, metrics, and speculation. The inherent relationships and rules that define landscape systems are apparent in contemporary proposals of urban form generation, or even economic regeneration. These relationships surface in practical terms through municipal codes or even climate resiliency planning, which are themselves inherently computational.[26]

The discipline of landscape architecture has been acutely concerned with the simulation of an established set of analog tools in computer software and hardware for the past two decades. Our tool set is comprised primarily of ways to paint, mark, draft, and model digitally in a way that mirrors physical materials that use paper, chipboard, pencils, pens, and markers. What we find is that, rather than claiming that there is a "digital" media, we have instead adopted computation as a simulator and optimizer of analog systems.[27] This state perpetuated a dialog in our discipline that has attempted to justify "computation" or defend "hand-drawing"; that discussion has been wholly unproductive over the past decade.[28] What is important to acknowledge is that neither methodology, analog or digital, defines the discipline of landscape architecture. To move forward in a contemporary design practice, or academic setting, requires a nimble understanding of how each tool may be deployed and the range of results it may produce.

Computation in landscape architecture provides more than opportunities to expand current design tools, workflows, and methodologies. Instead, the translation of landscape systems from modes of visual representation to relational, numerical, and temporal models provides a new lens that focuses the agency of landscape architects. Within the last decade, with the increase of computational efficiency, we find new models that more directly take advantage of the power of computation to build relationships and form new heuristic models in landscape architecture. This agency provides landscape architects with new territories of influence that thrust design into closer relationships with the exploration of iterative form for aesthetic and performative evaluation, as well as deeper connections with physical landscapes with new methods of construction that both collapse and expand current design processes.

New modes of thinking

Emerging coded environments

The term "code" is used in computation to refer to the instructions that drive software and hardware. Coding, the practice of creating code, implies an active state where software is created through a personal or collaborative syntactical process. Coders single out a group of individuals responsible for the creation of code while also implying an underlying culture built on these processes.

Code also provides a range of other definitions and can be used to describe rules or definitions that propose to delineate use or function. The coding of the environment implies a classification, the abstraction of physical and environmental phenomena to create a model that may be used for representation, analysis, or simulation. Design models, visual and/or numerical, describe the world and are the essential fodder through which designers develop design solutions. The continual construction, evolution, and maintenance of these models mediates and develops our relationships between the physical and virtual, underlying our assumptions of the physical world.

Our methods of abstracting the world, as landscape architects, are primarily computational and therefore generative and alterable in real time. The profession of landscape architecture is being reshaped and this is a calling to be aware of what is to come and how, ultimately, we may indeed reshape the profession in ways that may not currently be predictable. Computational design is woven into the surroundings of our daily lives. It controls the visuals we see on the ever-ubiquitous smart/mobile device, the time we wait at the mundane stop light, what we see first when we do a search on the Internet or send an email. Yet often we are intimidated by the prospects of computation, coding, or computer algorithms governing these daily tasks.

As our world becomes increasingly algorithmic, we must be aware that technological data usage does not simply become a reflection of privatized mobile/social media data mining, which, while a powerful tool and offering exciting new opportunities in urban planning, does have its limitations in data reliability or sample set.

As landscape architects have engaged in the previous decades with GIS, geo-design, and mobile data, we have garnished great rewards in being accumulators of some rather large data sets of physical topography, sea-level rise, and socioeconomic distribution. However, the gathering of data (the inventory) and understanding the algorithms controlling, sorting, or processing that information (the analysis) present the next stage of untold value for the potential of social, formal, materialistic, and environmental models that are more synthetic and controlled by the designer's intent. This is explored in essay 01.04, "Big Data for Small Places."

> The greatest value of building models is the disciplined way they require us to think about the impact of city policies and infrastructure on residents and visitors.
>
> Nick Chim and David Ory, Google Sidewalk Labs[29]

Doctoroff (Alphabet's Sidewalk Lab CEO) argues that the great leaps in economic growth and productivity have depended on the interaction and close proximity of the "physical environment," especially urban development and innovation.[30] The advent of a number of innovations and inventions such as electricity, the steam engine, and automobiles, for example, have drastically altered our way of life. However, in terms of architecture and landscape architecture, while a marked contrast exists between 1870 and 1940, Doctoroff asserts that "hardly anything" changed from 1940 to the present. The landscape architect's scope of work, contract structure, and client base remained largely the same in that span of time.

Whether we accept this hyperbolic proposition or not, we do begin to see the manifestation of that proposition. The information age is ushering in the high-tech "campus." The "campus" offers all the amenities of urban life that would appeal to a newer generation of professionals in the high-tech industry, in an attempt to attract the best talent and maximize or, better expressed, inspire productivity and creativity. The high-tech campus movement and the current early efforts, just as many of the pioneering efforts, have their quirks and, as economists are fond to say, have unintended consequences.

In the case of Apple, the quirks for their new "Spaceship" building meant housing problems, whether shortages of or high price of, and transportation problems. Google's "Googleplex" campus faced similar problems in addition to a hostile reaction from San Francisco as a result of the gentrification it brought about. Facebook's Frank Gehry-designed complex boasts of being the largest open-office workplace in the world.[31] However, questions remain about the "open space" concept and its effect on morale and productivity.

By far the most stinging of criticisms is about the role of the edifice in the urban setting, returning to the concept of the symbiotic relationship between innovations and the physical urban environment. All of these "campuses" have earned high marks not only by expert but also in the court of public opinion in the area of architecture, risk-taking, and bold and inspiring leadership in the workplace. While concurring, Louise Mozingo, landscape architect at the University of California, Berkeley, claims that Apple (and probably others) misses the point:

> You can't understand a building without looking at what's around it—its site, as the architects say. From that angle, Apple's new HQ is a retrograde, literally inward-looking building with contempt for the city where it lives and cities in general.[32]

As a landscape architect, Louise Mozingo, observes, in the 1950s and '60s corporate flagship offices fled the "dirty" inner city to the suburbs, building structures architecturally ahead of their time while nurturing and channeling their corporate culture toward productivity; but they did so deliberately. By contrast, Apple, Mozingo argues, fails: "Successful buildings engage with their surroundings."

Today's design culture seems to have a fascination in urban and architectural design with the electronically plastered and seemingly back-lit surfaces of reflectivity. The glorification of an apocalyptic aesthetic extends through our media-driven environment, bringing the worlds of reality and science fiction ever closer together.[33] Whether it be large billboard media or subtle sensor integration, the component makeup of urban landscapes is shifting from a static makeup to a more dynamic/responsive material composition.[34]

> One of the barriers to faster and wider change is a lack of dialogue between the people who live in today's cities and the folks who build tomorrow's technologies.
>
> Daniel L. Doctoroff, CEO, Sidewalk Labs[35]

The argument could be made that, in addition to those inhabiting the city and building the technology, our cities' future is largely influenced by a third group composed of landscape architects, architects, urban planners, and engineers. These "technocrats" are shaping the physical cities and environments within which future technologies and innovations must be integrated. They must anticipate and create "space" for a future that no one can define.

Our contemporary context romanticizes the technological, the "clean and simple," and the start-up culture, part of which is ephemeral. However, as landscape architects, the tie to urban form and "grit" of urbanity manifests itself on a much broader scale. By definition, the landscape architect must include all inhabitants, residents, and workers alike, in addition to purposefully capturing, or preserving, or creating, or even modifying the nontactile aspects of the city, well beyond the asphalt; those are aspects the technocrats are too often accused of missing.

The technological agenda we seek, as landscape architects, is that of a systemic socioenvironmental connection to technologies (known and unknown). Perhaps already within our reach lies not simply a cultural "Internet of Things" but an "Internet of ecologies," an "Internet of built environments." Presenting itself here is a model that influences not only the creation of day-one active spaces but the temporal dynamics of such environments throughout an evolving lifetime. Landscape architects are already "embracing digital media as a tool with analytic, performative, and representational possibilities." The computer is no longer the rival.[36] In a dramatic shift, the profession is rapidly moving beyond computation as a design representation medium; the tool is now influencing the thinking process of the landscape architect to shape dynamic models for adaptive and responsive landscapes.[37]

The aversion to computation

The human character may harbor an instinctive aversion to computation, coding, computer programing in the planning of our living environment and daily life. Understandably, we may express a fear of these media as manipulated by specialized individuals perceived to be somewhat distant from our world of daily social and physical existence, controlling our destiny.

This aversion may stem from our profession's attempt to reach far beyond the simply observable or gestural in nature. As designers, we struggle primarily with interface of the traditional syntax of code/computation. These traditional programming interfaces, such as coding in C# or Fortran, or even scripting in Python, have not yet operated at a level of abstraction designers are accustomed to thinking. Designers have had to rely on a team of computational experts attempting to translate the designer's language into computer code (scripting). Much can be lost in the translation. However, with developments in GUI (graphic user interfaces) such as Grasshopper software (plug-ins), a huge barrier has been crossed.

Doctoroff and the Sidewalk Labs team at Alphabet (Google) strive to close the gap between the residents of a city and the developers of future technology and their vision for urban planning: "We believe that when you put technologists and urbanists on the same team you have the potential to transform the urban environment."[38]
By bringing together landscape architects, architects, planners, technocrats, and of course city residents, who will live with the changes, we will all be closer to the ideals of urban planning. It is critical that landscape architecture engage with these conversations.

Not all landscape architects will become avid coders. However, it is imperative as a profession agitating for creativity, exploration, innovation, and substantial investment in form generation and alteration of the urban realm that we understand and communicate with those shaping the future components of the synthetic urban construct. The risks of not doing so are very high. There are risks as well for being in the leadership. The business sector is littered with companies that took risks and failed, but also others that did not innovate (or innovated too late or even too soon). The once-revered Eastman Kodak Company comes to mind.[39]

The interface barrier

With the conceptualization of the mouse in 1965 by Douglas Englebart and its popularization with the advent of Apple computers in 1983, we begin to see elements of the hand and first extensions/abstractions of analogue media emerge into the digital realm. However, it is only recently through the GUI of scripting that we observe a widespread use of computational means and methods in landscape architecture and our related disciplines. GUI-based scripting engines such as Grasshopper, Dynamo, Kismet, and Marionette have all become a contemporary phenomenon, opening up new computational vistas to designers who would simply not have bothered to cross the learning barrier to entry in text-based coding editors.

These coding and scripting abstraction/interface platforms have acted as a gateway for many designers, who expanded their reach to numerous problems and data sets in the emerging technological world. Grasshopper, for example, initially launched by McNeel and Associates and created by programmer David Rutten,[40] was built upon, after its creation, by numerous add-ons, plug-ins, or extensions (e.g., Rhinoceros). With the built-in script in the background, designers could now engage in parametric design, skipping over the tedious and discouraging scripting, undeterred by the computational demands of the past. Moreover, designers could now concentrate on their work, instead of spending time learning and acquiring computational tools to get to the task. The simplicity with which Grasshopper and Rhino could be utilized led to widespread use of the software across top architectural firms and eventually landscape architecture offices, opening the computational world.[41]

The success of the software was largely due to Robert McNeel's insight: "Writing code is not something designers really want to get their head into." His "business model" had a two-pronged approach: "designers set up sophisticated relationships between the parts of the design problem" and, in addition, the company would make the software available for free during the development process, benefitting from the input of users worldwide.[42] Although a small firm by comparison, without the deep pockets of a Dassault Systèmes or Autodesk, by 2009 McNeel reported having 250,000 Rhino users worldwide, among them 50,000 in the field of architecture. This number has since bourgeoned further, as Rhino became commonplace in architectural offices and urban design practices.[43]

This continuous improvement is a necessary survival mechanism in the marketplace. Competition, at times driven by monetary rewards and at times by self-motivation or satisfaction of rising to an intellectual challenge, leads to explorations and innovations. Progress in parametric design was not enough. In the field of architecture, the virtual wall beyond software and data-based design has already been pierced, as architect Rivka Oxman declared in 2008: ". . . novel directions for environments that support performance-based design are beginning to emerge."[44] Even five years after Oxman's remark, architect Michael Hensen showed more concern for the profession and the lack of progress, and ominously warned that "architecture is on the brink. It is a discipline in crisis."[45]

Admonitions such as Hensen's, although debatable in their severity, may also increase motivation to leverage technology for performance's sake. We know the field of innovation is littered with failures for a variety of reasons, at times for the better: "creative destructions" is what economists refer to when they mean that an old technology is naturally replaced by a new and better one (e.g., video tape cassettes replaced by CDs, DVDs, and flash drives).[46] The let-down may include at times the failure to keep up with innovation, lack of organizational skills, or lack of insight or any of these combinations.[47] The path to innovation does not follow a scientific course. At times risk-taking is necessary and at other times excessive risk (neither is quantifiable) leads to failure. The Eastman Kodak Company is a case in point. The iconic firm had failed to gauge the digital revolution, ironically a field in which the company was a leader; meanwhile, management decisions on how to cope with the threat from competition were sluggish or not appropriate. The company, listed on the illustrious Dow Jones since 1930, was dropped in 2004. It was founded in 1888 and filed for bankruptcy in 2012.[48]

This is not to say that the profession of landscape architecture as a whole lies at risk or is even impaired owing to the lack of a digital or computational engagement. However, the societal relevancy, particularly in urban contexts, raises a question over the role of technology (particularly start-up) companies in shaping the environments. Whether from direct infusion of investment from the technological sector or from simply a shift in technology available in urban futures, the influence with which computation and the computationally minded will shape our built environment is without question.

It may appear that the complexity of the world around us is increasing in the human ability to interact and control our surrounding everyday objects. In reality, we are seeing an increasing translation from mechanical to digital (coded) language within our daily lives. While perhaps it would be more difficult for the everyday individual or "traditional" car mechanic to work on a 2017 car than it might be to repair/restore a 1960s hot-rod, it might be the very opposite for robotics engineer or computer programmer, who may now be the most able to "work" on their own car over the weekend. It is the simple algorithms of our daily lives that we are not accustomed to engaging with to control our surroundings – unless we are given that control directly through pre-manufactured application. The barrier to "customization," "jerry rigging," or "fiddling with" lies in the cultural translation of the basic algorithms controlling our surroundings. Not understanding these algorithms, the language (codes) these instructions are written in makes the objects appear more complicated—when in reality they are simply more complicated in a digital sphere than in a physical or mechanical interaction. There are now fewer moving parts and more moving electrons.

Initiatives, risks, successes and failures

Designers' aversion to computational tools, especially for landscape architects, is no longer suitable.

Enormous risks are taken by those who profess to be in the avant-garde of their fields, some with tragic ends, but "the road not taken" may be just as calamitous. Landscape architects may end up as the quaint Norman Rockwell pharmacy pushed aside by the chain stores, in our case by related disciplines and unanticipated neo-disciplines eager to fill in the gap. Landscape architects, by their own design, yearn to be at the cusp of creativity and therefore need the computational tools to remain relevant, if not lead.

The rapidly growing technical and cultural interactions between humans and computers, whether they be touch screens, haptic devices, or virtual or augmented realities, are enabling our return to the gestural and observable interface process at the roots of our profession: free-hand spontaneous drawing and sketching—not coincidently, the word "digit" comes from the Latin word *digitus*, meaning finger or toe.

A new generation of digital natives have been brought up by the new interactive normalcy to live, work, and create abstractly through these virtual media. Machine learning and script definition of software are assisting to fill in the gaps of the executable details of our creative process. The executable interface is now rapidly evolving. It is accessible for designers to "code" problems at the highest levels of abstraction through gesture and real-time feedback, all while designers observe the instantaneous impact of their digital interaction on the built environment.[49]

The inspiration stemming from computation/parametric design manifests itself predominantly through the language of mathematics.[50] The relationship between form and environmental data is often more attuned to the architectonic or geometric nature of the architecture disciplines than those of landscape and the "wild" in nature. However, even though the more geometrically "simple" in design and process are easier to calculate through computational design, why do we see this technology being so often only used and advertised in the most abstract or biomorphic of projects? It is certainly refreshing to see these tools used as an enabler or inspiration for complex and new ways of design thinking. However, we must also take advantage of the day-to-day problem-solving capabilities and practical use of such computation engines.

Is it truly that these tools are used more commonly for esoteric competitions in architecture? Or, yet, that these tools are the "idolizable" graphics that we see published time and time again? Are the day-to-day applications of these tools being misconstrued as a tool for renderings over form generation? Are the forms of competition architecture and "deflated sea creature" starchitecture of our day reflections of computation for computation's sake? Or are these an interpretation of "computation" or "parametric" as an aesthetic even if made through more traditional modeling techniques?[51]

We describe, perhaps in a negative tone, the common perceptions and prevailing uses of Grasshopper and other parametric engines to hopefully draw the reader's attention to a new platform of thinking about computational design and technology in landscape architecture. Software, such as Grasshopper or Dynamo, must be recognized as problem-solving tools and engines of creativity. These tools are not simply engines of graphic communication that perhaps a new generation of design professionals may have mistakenly interpreted and represented as a means to an end in itself. Rather, parametric tools, such as Grasshopper, are practical instruments with the potential to address problems and find solutions while unleashing a vast source of creativity. For example, graduate students used:

> Rhino to create the model and Grasshopper to drive the dynamic inputs; i.e. sensors and their inputs that drive responsive actions. The model focused on the device's formal aesthetic and the transformations the device will make. The Grasshopper components were used to drive actual values within the digital model such as rotation (0–360 degrees), transformations (movement in feet/meters) and/or binary actions (off/on).[52]

Emerging from the most ancient of traditions in design and architecture, our obsession with geometry and form are driven by mathematical relationships that are both discreet and subliminal. Grasshopper, Dynamo, and Python are some of the first conversation openers in computational design today, yet it is specifically their abstraction of and thus accessibility to computation that have driven their remarkable success.

Abstraction and productivity

Abstraction improves productivity. You don't need to worry about the decisions made at the underlying levels. This is why designers are so well suited for large-scale thinking and master planning efforts, as we are trained to think abstractly and focus on the "big idea." The only thing you need to worry about is the interface to the next level. This is what we, as designers, need to begin understanding and interfacing with the systemic models available. Understanding that, we can influence that next level of realism in the tools of what we want them to perform. The higher up you go in this list, the more abstract you are getting with the execution of the problem. When creating your flowchart or algorithm you don't need to know the syntax of a specific computer algorithm. When translating that algorithm to a specific syntax you don't need to know the flow of logic gates or circuits of the hardware that will implement that code. These synthetic models of thinking across scales and variables are beginning to allow landscape architects to test and visualize in real time the implications of macro to micro decisions.

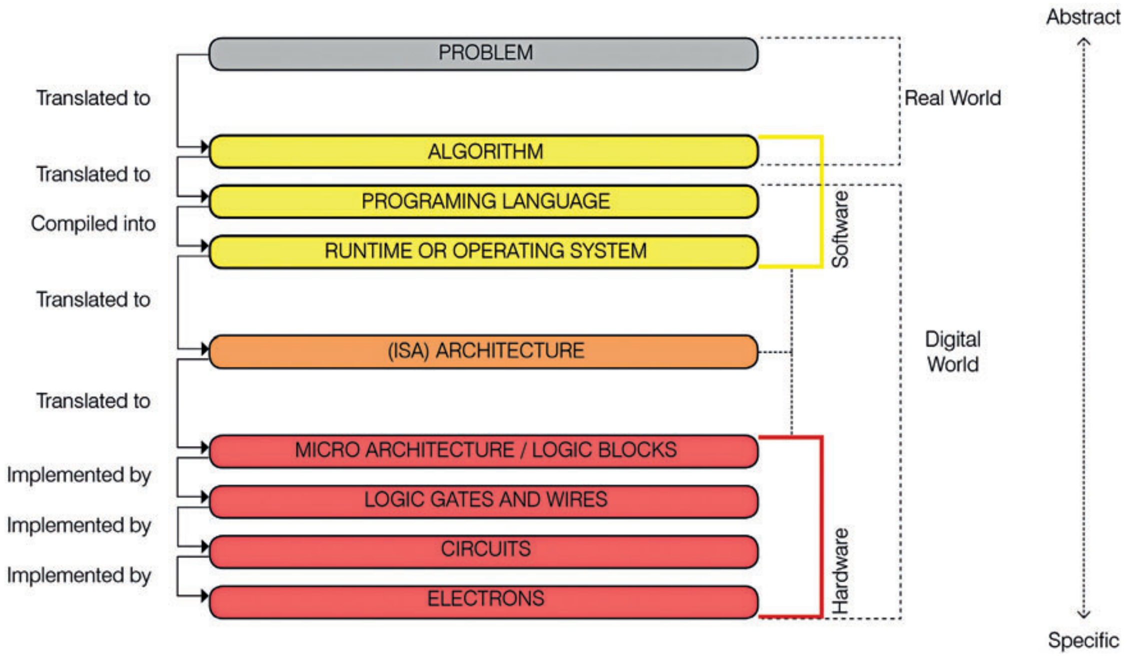


FIGURE 0.1.7 **We solve problems with computers through electrons. You can "program" at all of these levels; you can program at high levels of abstraction in flowchart and problem-solution discussion in everyday language; you can program all the way down so as to physically program the circuit gates or transistors to command a machine to perform a specific task. Some of the first computers required that level of involvement. It is helpful, however, to understand the logic of these lower levels so as to understand why something may not work, or may not work as anticipated**

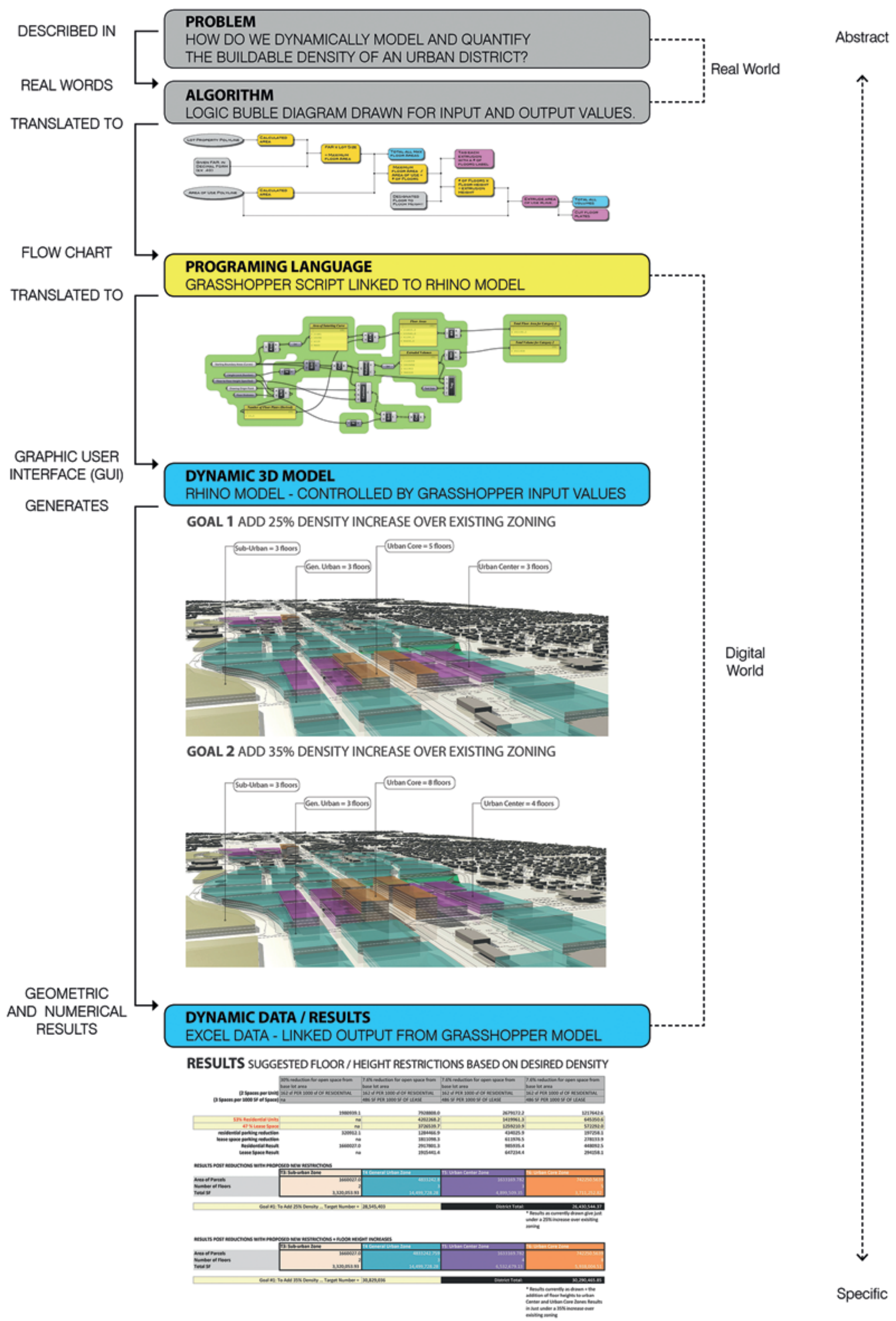


FIGURE 0.1.8 Example abstraction of computational model for Leawood KS. The FAR values of the properties were associated in Grasshopper as dynamic variables among others to test scenarios for the city of altering current zoning regulations to allow for increased urban density

Image: Design Workshop

Abstraction of code to scripting and the graphic user interface (GUI)

Many designers will not engage at the high level of syntactical knowledge necessary for scripting given time constraints as one of significant barriers. However, Grasshopper, Rhino, other GUI-based scripting allows designers to more readily connect the outcome of code with the formal representation without having to know how to write code.

The world-renowned architect Bjarke Ingels, in his 2013 interview, "Inside the Business of Design," described the impact of Grasshopper and visual scripting on architecture in these simple terms: "Grasshopper is to parametric scripting what Windows and Macintosh were to the graphical interface for personal computing." Ingels describes the essence of GUI-based parametric design as follows: "Scripting came from being this incredibly difficult thing in architecture to, at least, I can understand the principles. You basically construct incredibly complex formulas by graphically combining different variables with little wires almost like a switch board."[53]

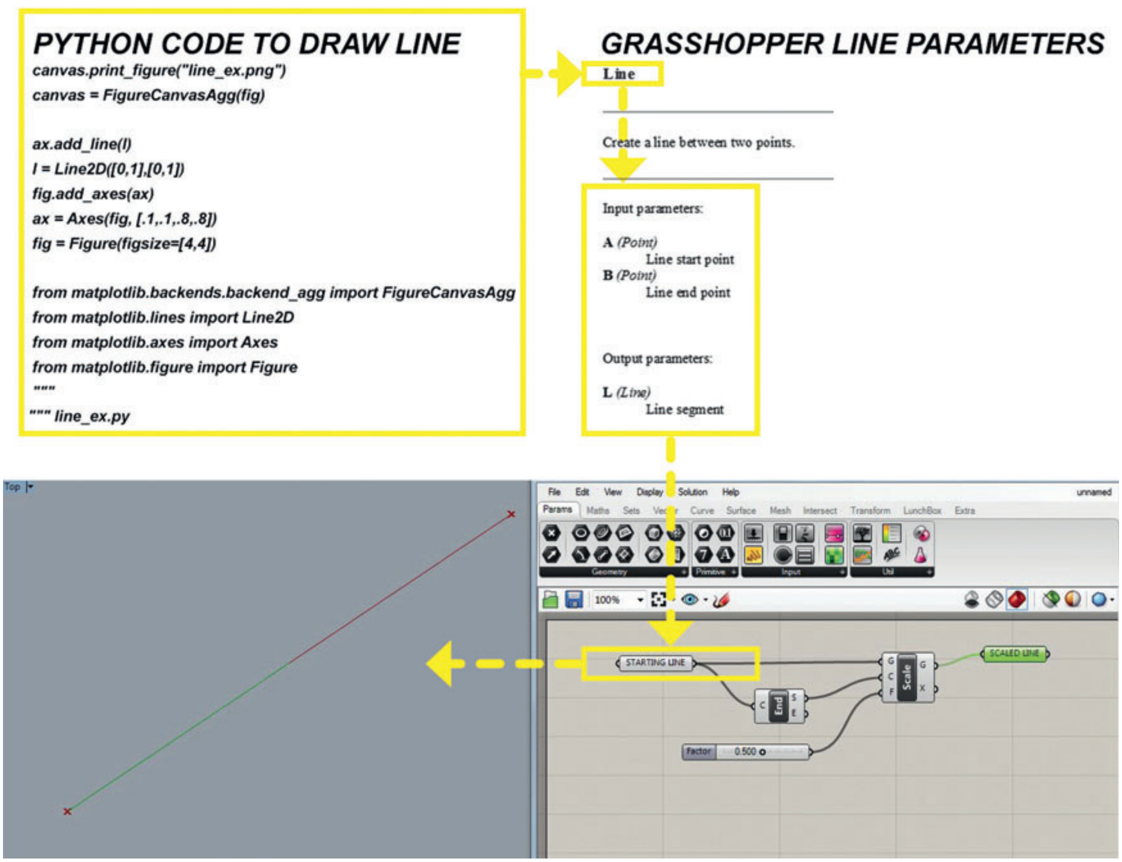


FIGURE 0.1.9 **With a line in Rhino/Grasshopper, the definition at the lower right can be seen as graphic icons representing the syntactical commands of the software. The line in question to be scaled by (.5) is represented by the node at the top left—with the process, variables, and resultant line shown as their own nodes. Within these nodes is a mass of "code"—this is what gives scripting, particularly GUI-based scripting, its efficiency and ease of use: in that background, the code extracts or translates the level of detail/syntax for the designer**

The roots of understanding computational and parametric design do not lie buried beneath complex mathematical formulas or coding syntax. Instead, they reside in the organization of thoughts and a design approach. When designers understand code and computation in this manner, it is possible to then frame design problems through this lens, opening up a dialogue between design intent and computational iteration and generation.

Language of change

The design profession is beginning to see fascinating examples of these new computational approaches and applications. However, the vehicle by which these applications are brought to life remains mysterious. What is not as evident is the logic, the thought process, and the utilization of parametric design that have been applied to bring about the complex execution. Years of efforts dating back to 1967 at the MIT Media Lab succeeded in "civilizing" or "taming" design and computer code. Starting from "Scratch," so aptly named, in 2003, the program began to use graphics interface rather than the cumbersome coding string.[54] At MIT Media Lab, computer scientist Mitch Resnick directs the "Lifelong Kindergarten," where, at a very young age, children learn to program and design. As Resnick explains, "When you learn to read, you can then read to learn."[55]

One entry barrier to that vehicle, the aversion to understanding the potential of computational media and its syntactical interface, has been widely broken down in recent years by young designers through the GUI syntax of scripting. How can we leverage this newly acquired foothold and understand better what we are gaining from parametric modeling/visual programming/coding as a design process and conceptual generator?

Coding is a common language of creation, iteration, logic, communication, exploration, and innovation for the twenty-first century. Just as the tried-and-true graphic conventions of landscape architecture (the heyday of the "EDSA," Mike Lin-style penmanship) have become an international standard for landscape communication, so too have the various languages of coding to the world of technology development.

Computation and parametric design are grounded in the field of mathematics. As such they are by-products that the field of mathematics, in its pioneering age, had not envisioned. Edward Frankel, a mathematician, expresses the practical use of the discipline in much the same way we might broach the subject of computational design:

> One of the key function of mathematics is the ordering of information. This is what distinguishes the brush strokes of Van Gogh from the mere blob of paint. With the advent of 3D printing, the reality we are used to is undergoing a radical transformation: everything is migrating from the sphere of physical objects to the sphere of information and data. We will soon be able to convert information into matter on demand by using 3D printers just as easily as we now convert a PDF file into a book or an MP3 file into a piece of music.[56]

A computer program is not a task that someone who knows how to code goes right into and writes simply because they know the language. The program is dependent on a problem to be solved. A programmer must know the logic and sequence of commands intended to be developed. The code is simply the wording telling the computer what to do. That communication ability is vital.

Similarly, in the design professions the knowledge of how to use a software media is not the same as the knowledge of creating built environments. Our landscape architecture profession understands the language of design, drawing, and planning. We would look at someone rather wearily if they assumed that the ability to use CAD alone is a license to create a master plan for a community.

The development of a computer program is much like the comprehensive master plan for an urban design at different scales. The larger the design challenge, the more complex and comprehensive the design must be, just as the more demands asked of a software program, the more complex and comprehensive the algorithm development must be for that software.

The design itself and the creative development of the key algorithm make the software run and create the city. The letter keys typed into the computer, or the lines drafted onto the plan sheets, are mere translations of the critical thinking that went into the original creation. The vision is what counts. Ideas carry most of the weight.

Computerization vs. computation

One of the greatest struggles we face, as a design profession, is our attempt to overcome what we perceive to be the limitations of technology and computation. That perception is that computation is "only" a tool kit, only a set of operations. We must understand computation as a way of thinking, as a way of linking our thought process and dynamic environments. This is very different from "computerization."

We must make a critical differentiation between the contemporary computerization and the vast potential of computation. The most common mode of using mechanical computers in contemporary landscape architecture is just that: computerization. We input preconceived, and often predrawn, solutions into their digital format for safekeeping, printing at various scales, or enhancing their graphic presentation. The ideas themselves often do not grow by this means of digitalization but many times lose their clarity of communication in the cumbersome translation of media.

> One increases the amount and specificity of information, while the other only contains as much information as is initially supplied. A computer-aided approach assumes an object-based strategy for encapsulating information into symbolic representations – method of organizing information. In contrast, a computational approach enables specific data to be realized out of initial abstraction – in the form of codes which encapsulate values and actions.
>
> Sean Ahlquist and Achim Menges[57]

Computerization is a tool kit of prefabricated software that we accept or use within the bounds of what it allows our landscape to be. What we yearn for as a profession is computation. That concept goes far beyond the tool kit. Computational design is the systematic method for critical thinking that emphasizes thought process and iteration over memorization and duplication. It stresses the linking of ideas, and interaction between the parts of the problem and the solution.

Computational thinking combines the powerful orderly process of algorithmic organization with the equally powerful, but more chaotic, process of iterative design. Computational design is a way of approaching all the challenges in the world around you in a more visionary, creative, far-reaching, and organized way that is more likely to succeed. We engage with computational decisions each day whether we realize it or not. In the design field the passage beyond computational skills, and tools, albeit influenced by computer thinking, is a paradigm shift: "Steps away from 'form making' and toward 'form finding.'"[58]

Of course, the danger with any innovation or innovative techniques is that the ultimate practical goal and problem-solving may be lost. Entertainment and dazzle at times supplant substance. No discipline, however, is immune from such temptations. The entertainment aspect, and even the ostentatious, are an integral part of the creative and inventive mind:

> At present scripters tend to be of the "lone gun" mentality and are justifiably proud of their firepower, usually developed through many late nights of obsessive concentration. There is a danger that if celebration of skills is allowed to obscure and divert from the real design objectives, then scripting degenerates to become an isolated craft rather than developing into an integrated art form.
>
> Hugh Whitehead former head of the Foster + Partners Specialist Modeling Group[59]

It is perhaps this lone gun mentality that has shaped the professional misconception of coding or scripting as a distant task related to but not a part of the design process.

Models for landscape architecture: computation as transitional tool set

A fundamental shift in the design tool set for landscape architects is required to address the hyperscaled issues of climate change, global extraction economies, and megacities, to enable the profession to develop viable synthetic design proposals into the future. The landscape architect must embrace a tool set that is real-time and more fully augments and extends the capabilities of the human mind. This goes beyond simulating analogue media in virtualized environments and beckons for design and construction techniques that are more directly connected to material and biologic systems through responsive technologies.[60]

> Creating new models is difficult and hence the tendency is to work with existing models of thought. In computation this is even more likely due to the reusability of algorithms in the form of code. The path of least resistance has led to a limited set of computational models for design being used over and over again.
>
> Gengnagel, Kilian, Palz, and Scheurer[61]

The transition will be a complex evolution from "static" built/urban environments to "dynamic" self-constructing, living, breathing, and even artificially intelligent (thinking) environments.

These cities are not in such a distant future; our urban environments are rapidly becoming "responsive" habitats, not simply static constructs. For how long will a "set" of static drawings help us to create working and living environments for a dynamic and mobile populace, ecology, and culture?

Computation and technology become ways of testing/experimenting with not only more complex physical but also social systems in the built environment. The scale of our built environment, based on the practicality of contemporary physics, will be built through systems of subcomponents/assemblies. While the vast potential of media such as 3D printing are contemporary "shock and awe"

FIGURE 0.1.10 Ecopods
Source: Squared Design Lab/Höweler + Yoon Architecture

FIGURE 0.1.11 **3D Boulders scanned and assembled in the computer for contractor construction**
Image: Design Workshop

examples forming themselves at life scale/inhabitable environmental scale, these methods are already predominantly integrated as means of component assembly alongside traditional construction trades and typologies.[62] Whether the applicable "codes" involved are in syntax of jurisdictional zoning requirements, mandated decree by an ancient ruler, or Python scripting running a 3D printer, any deviation from the variables at hand is governed by the associated impacts to the project. What our new computational media and new methods of construction allow is a fundamental bridge between design idea and physical reality. The connection of the virtual and physical model in space can now exist in direct mirror or alternate reality of each other, as opposed to the cumbersome two-dimensional abstraction of orthographic drawings and measured scales.

At the core of the essays in this volume is an attempt to place landscape architects at the forefront of discussions and solutions to the future demands of the built environment that are often dominated by the technology industry. Technology companies will continue to extend their reach into the city, providing us with a glimpse of the future workplace and living place as seen in corporate campuses of Facebook, Apple, and Google. However, maintaining the involvement of the designers in that plan for expansion is critical, not only for those outside the industry affected by the design but also for the enterprise itself (employees and technological elites), which may not see "the whole picture." Landscape architects and designers must understand the potential and collaborative nature of these movements. They must insert themselves, not for the sake of it but to help avoid the pitfalls of the past, contribute from the hard-fought lessons of the past, and offer the intellectual capital earned through research and practice.[63]

Our emerging design methodologies mean nothing without the profession understanding its potential and investing in its development. Currently in landscape architecture the profession has achieved for design what Ford achieved in 1913 for his Model T.[64] The profession has created a fundamental industry shift in process and efficiency, with less impact on creativity or new service development, producing a system of creating the same things faster and more cost-effectively rather than utilizing the potential of a new paradigm to think, generate, and analyze.

In 1915, a survey revealed that at Highland Park, in the factories of the Model T, laborers spoke more than 50 languages, very few speaking English.[65] How was it that execution remained so efficient? The assembly line labor of the engine did not join the paint department two days out of the week—they knew their component execution and it did not matter what was to their right or left. It is this seemingly streamlined process that allows specialists within firms to execute a task more rapidly such as 3D rendering, paving details, or specification writing in contemporary landscape architecture. However, it is the massive precut libraries of two-dimensional people copied and pasted from one proposal to another, or Grasshopper definitions of white hex grids over green terrain, which represent the globalization of a design "aesthetic" in contemporary computational models of practice, rather than a complex and adaptable model of shifting local variables.

> There is an economy of shapes, anyone who has ever worked all night to create design drawings or models on a deadline is vividly aware of that. Some shapes are quick and easy to construct with available tools, but others are slow and laborious. Since architects (and landscape architects) must always produce designs

FIGURE 0.1.12
Responsive topography for fluvial landscapes exhibition at Harvard Graduate School of Design, Bradley Cantrell, Leif Estrada, Jeremy Hartley, Tyler Mohr, Andrew Boyd, Cambridge, MA
Image: Keith Scott

within finite time periods, and with limited expenditure of resources, they are always constrained by the current shape economy.

William J. Mitchell, Cambridge[66]

We must look to the new economies surrounding our profession to reshape our own. What the smartphone/mobile device network created as a new market for mobile app development, or what the Wright brothers created for the transportation industry in disruptive technologies, is what our profession must look forward to, not shy away from. When cars fly (as Airbus has already demonstrated) or become automated pods at half their current size, what will happen to our scopes of urban streetscape design? Will there be increases in green space development, or will terrestrial urban land become an undesirable back of house, with high-rise green walls and fifty-story or more green roofs the gentrified public spaces? The opportunities of a future design economy lie not only in new materials and levels of formal complexity in architectural and environmental fabrication but also in opportunities to tackle greater challenges of execution, and influence over the direct relationship between design, ecological systems, and fabrication.

Notes

1. Coates, P. *Programming Architecture*, Routledge, 2010.

2. Coates, P. *Programing Architecture*.

3. Randell, B. "The Origins of Computer Programming," *Annals of the History of Computing IEEE*, 16, 6–14, 1994.

4. Bromley, G. "Charles Babbage's Analytical Engine, 1838," *Annals of the History of Computing*, 4(3), 196–217, July–September 1982. doi:10.1109/MAHC.1982.10028, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4640697&isnumber=4640695

5. Turing, A.M. "On Computable Numbers, with an Application to the Entscheidungs problem," *Proceedings of the London Mathematical Society*, s2–42, 230–265, 1937. doi:10.1112/plms/s2–42.1.230

6. Trueman, C.N. "Alan Turing," *The History Learning Site*, April 21, 2015, historylearning site.co.uk, accessed November 24, 2017.

7. "Computer." Oxford English Dictionary (3rd ed.). Oxford University Press.

8. http://european-section-with-lepoutre.blogspot.com/2013/09/memory-wwii-alan-turing.html#!/2013/09/memory-wwii-alan-turing.html

9. www.nsa.gov/about/cryptologic-heritage/historical-figures-publications/women/honorees/caracristi.shtml

10. "Quantum Internet Is One Step Closer," *Wall Street Journal*, June 15, 2017, www.wsj.com/articles/quantum-internet-is-one-step-closer-1497550005

11. NSA, *Cryptolog* #54, published June 1979. Released Public Declassified by NSA October 12, 2012, accessed June 25, 2017.

12. Weller, R. "Planning by Design Landscape Architectural Scenarios for a Rapidly Growing City," *Journal of Landscape Architecture*, 3(2), 2008.

13. https://cga-download.hmdc.harvard.edu/publish_web/Annual_Spring_Workshops/2006_HRG/Harvard_Chrisman.pdf

14. Chrisman, N. *Charting the Unknown: How Computer Mapping at Harvard Became GIS*, ESRI Press, 2006.

15. Ervin, S. and Hasbrouck, H. *Landscape Modeling: Digital Techniques for Landscape Visualization*, McGraw-Hill, 2001, p. 34.

16. www.foresthistory.org/ASPNET/Publications/multiple_use/chap5.htm

17. www.danielsevo.com/hocg/hocg_1970.htm

18. Morrison, M. *Becoming a Computer Animator*, Sams, 1994.

19. MLA. *The Phantom Menace Review Quotes*. Quotes.net, www.quotes.net/mquote/1110673, accessed June 27, 2017.

20. http://fourmilab.ch (John Walker's personal website), accessed June 25, 2017.

21. Walliss, J. and Rahmann, H. "Introduction," in *Landscape Architecture and Digital Technologies*, Routledge, 2015.

22. Steinitz, C. *A Framework for Geodesign*, ESRI, 2012.

23. Vargas-Moreno, J.C. *SPATIAL DELPHI: Geo-Collaboration and Participatory GIS in Design and Planning*, Department of Urban Studies and Planning Massachusetts Institute of Technology, 2008.

24. Christensen, C.M. *The Innovators Dilemma*, Harvard Business School Press, 1997.

25. Brodie, M.L., Mylopoulos, J., and Schmidt, J.W. (eds.), "Preface," in *On Conceptual Modelling. Perspectives from Artificial Intelligence, Databases, and Programming Languages*, Springer-Verlag, 1984.

26. Ahlquist, S. and Menges, A. "Computational Design Thinking," in Menges, A. and Ahlquist, S. (eds.), *Computational Design Thinking*, Wiley, 2011.

27. Manovich, L. *Software Takes Command*, unpublished, http://softwarestudies.com/softbook/manovich_softbook_11_20_2008.pdf, accessed October 10, 2012.

28. Treib, M. *Representing Landscape*, Taylor & Francis, 2008.

29. www.sidewalklabs.com/blog/a-key-to-democratizing-urban-solutions-is-building-better-models

30. https://medium.com/sidewalk-talk/reimagining-cities-from-the-internet-up-5923d6be63ba

31. Frankel, T.C. "What These Photos of Facebook's New Headquarters Say About the Future of Work," *The Washington Post*, November 30, 2015; Boorstin, J. "Inside Facebook's Futuristic New Headquarters," ETCNBC.com, May 22, 2015.

32. Rogers, A. "If You Care About Cities, Apple's New Campus Sucks," *Wired*, www.wired.com/story/apple-campus, June 8, 2017.

33. Rappaport, M. "States of Distraction Media Art Strategies Within Public Conditions," in Aneesh, A., Hall, L., and Petro, P. (eds.), *Beyond Globalization. Making New Worlds in Media, Art, and Social Practices*, Rutgers University Press, 2012.

34. Srinivasan, R. "Digital Public Spaces: Implications Toward Cultural Memory and Participation," in the proceedings of CATAC (Cultural Attitudes toward Technology and Communication), 2008.

35. Doctoroff, D.L. *It's Time for Urbanists and Technologists to Start Talking*, https://medium.com/sidewalk-talk/it-s-time-for-urbanists-and-technologists-to-start-talking-df1b57abfbd1

36. Cantrell, B. and Michaels, W. *Digital Drawing for Landscape Architecture: Contemporary Techniques and Tools for Digital Representation in Site Design*, John Wiley & Sons, 2010.

37. Oxman, R. "Thinking Difference: Theories and Models of Parametric Design Thinking." *Design Studies*, 52, 4–39, 2017.

38. www.sidewalklabs.com/about

39. Deschamps, J.-P. "Classic Root Causes of Innovation Failures—Things We All Know but Sometimes Forget," *Strategy and Communication for Innovation*, 41–60, 2017.

40. Tedeschi, A. *Parametric Architecture with Grasshopper*, Le Penseur, 2011.

41. *AEC Magazine* "Review: Grasshopper ArchiCAD Connection," December 8, 2015.

42. Martyn, D. "Rhino Grasshopper," *AEC Magazine*, 42 (May/June), 14–15, 2009.

43. Martyn, D., "Rhino Grasshopper," *AEC Magazine*, July 1, 2008, and May/June, 2009.

44. Oxman, R. "Performance-Based Design: Current Practices and Research Issues," *International Journal of Architectural Computing*, 6(1), 1–17, 2008.

45. Hensel, M. *Performance-Oriented Architecture: Rethinking Architectural Design and the Built Environment*. John Wiley & Sons, 2013.

46. Schumpeter, J.A. *Capitalism Socialism, and Democracy*. Harper & Row, 1976.

47. Jensen, M.C. "The Modern Industrial Revolution, Exit, and the Failure of Internal Control Systems," *The Journal of Finance*, 48, 831–880, 1993. doi:10.1111/j.1540–6261.1993.tb04022.x

48. Deschamps, J.P. "Classic Root Causes of Innovation Failures—Things We All Know but Sometimes Forget," *Strategy and Communication for Innovation*, 41–60, 2017.

49. Kayser, M., Laucks, J., Duro-Royo, J., Gonzales Uribe, C.D., and Oxman, N. Silk Pavillion, 2013, CNC Deposited Silk Fiber & Silkworm Construction MIT Media Lab; Kayser, M., Duro-Royo, J., Sharma, S., Bader, C., Kolb, D., and Oxman, N. A Perpetual Spring Environment for Bees and Humans, 2016, MIT Media Lab; S.J. Keating, Lelad, J.C., Cai, L., and Oxman, N. Towards Site-Specific and Self-

Sufficient Robotic Fabrication on Architectural Scales, *Science Robotics*, 2, 2017.

50. Burry, J. and Burry, M., *The New Mathematics of Architecture*, Thames and Hudson, 2010.

51. Schumacher, P. *Parametricism as Style – Parametricist Manifesto*, London, 2008.

52. Cantrell, B. and Melendez, F. *Responsive Systems*, Dissertation, Louisiana State University, 2012.

53. Ingels, B. 2013 interview, "Inside the Business of Design," www.youtube.com/watch?v=otsmpjaeHXc&list=PL1IM4xtSxXPPHPYEoR6f-87ccWfOWGfqo

54. Nagle, J. *Getting to Know Scratch*, Rosen, 2014.

55. *Mitch Resnick TED Talk*, November 2012, www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code

56. Frankel, E. "Love and Math, The Heart of Hidden Reality," *The New York Times*, November 18, 2013.

57. Ahlquist, S. and Menges, A. "Computational Design Thinking," in Menges, A. and Ahlquist, S. (eds.), *Computational Design Thinking*, Wiley, 2011, pp. 10–11.

58. Oxman, R. "Performance-Based Design: Current Practices and Research Issues," *International Journal of Architectural Computing*, 6(1), 1–17, 2008.

59. Burry, M. *Scripting Cultures*, Wiley, 2011, p. 252.

60. Cantrell, B. and Holzman, J. *Responsive Landscapes*, Routledge, 2015.

61. Gengnagel, C., Kilian, A., Palz, N., and Scheurer, F. "Computational Design Modeling," in *Proceedings of the Design Modeling Symposium*, Berlin, p. VII, 2011.

62. Loosemore, M. *Innovation, Strategy and Risk in Construction. Turning Serendipity into Capability*, Routledge, 2014.

63. Sheppard, S.R.J. "Participatory Decision Support for Sustainable Forest Management: A Framework for Planning with Local Communities at the Landscape Level in Canada," *Canadian Journal of Forest Research*, 35(7), 1515–1526, 2005.

64. Womack, J.P., Jones, D.T., and Rood, D. *The Machine That Changed the World*, Harper-Perennial, 1991.

65. Cossons, N. "From Manufacturer to Prosumer in Two Hundred and Fifty Years," in *Transactions of the Newcomen Society*, 74(1), 2004.

66. Terzidis, K. "Foreword," in *Expressive Form: A Conceptual Approach to Computational Design*, Taylor & Francis, 2003.