The Death of the Author

Words Alex Wiltshire Art Daniel Brown

Google's Deepmind AI program has beaten human champions of the fiendish strategy game Go. Twitter bots flood timelines with often strange – sometimes plausible – posts that are read as closely as those of real friends. Computers are choosing to buy and sell financial stock, giving military drones targets to attack and fooling dating-service subscribers that they're prospective partners.

We've entered a world in which the divide between the digital and the physical is diminishing, where computers are an integral part of life and what they do is becoming hard to distinguish from what humans do. They're also designing things – buildings, art and video game worlds, all assembled from lines of code, sets of rules and mathematical functions. Design is being generated, not by human imagination and experience, but by central processing units. "The generated results are outside the norms of beauty and aesthetics," says Matias del Campo, a Viennese architect who creates buildings and structures by generating digital, seemingly organic systems. "It's a new aesthetic."

"It's incredibly exciting," agrees Daniel Brown, a digital artist and designer whose beautiful computer-generated forms accompany this article. "It can make things that humans have never made before."

This practice comes under many names. In the world of video games it's known as procedural generation; in architecture it's often computational design. Visual creators, such as Brown, might refer to it as generative art. Computational design is perhaps the best catch-all term, however, since it works in a similar way across all these disciplines. Applying mathematical rules to design problems creates buildings and shapes (often evoking organic processes) that evolve out of those rules. But because computational design often includes simulation as part of the process, it can also provide an integrated approach to design and testing. As such, it presents a new way of creating that transforms the very nature of being a designer. With the rapid growth and development of computer-based production processes such as 3D printing, computational design has greater potential than ever, placing it in the vanguard of the Fourth Industrial Revolution.[1]

Computational design's direct origins lie in the computer labs of the 1960s and 70s, but its roots

1 The Fourth Industrial Revolution is a development of the digital revolution (the so-called Third Industrial Revolution). It is somewhat vaguely defined, and has been proclaimed multiple times since the 1940s. It now typically refers to ways in which the Internet of Things, artificial intelligence and data science might be incorporated into physical manufacturing and cities. It came into particular focus during the World Economic Forum, which was hosted in Davos, Switzerland, in January 2016.

All pictures from the ongoing series Travelling by Numbers by Daniel Brown.

reach back into works such as Jean-Nicolas-Louis Durand's early-19th-century *Précis des Leçons d'Architecture*, a pre-modernist proposal to design buildings using repetitive, modular elements. Alternatively, you might look to Leonardo da Vinci's studies into representing centrally planned churches, involving plans defined by rules that configured circles and rectangles from which forms grew almost organically. And if you really want to keep going back, Vitruvius's tying of proportion to the human body in his 1st-century AD treatise *De Architectura* was also an attempt to codify designed forms.

Yet with the growth of computers, complete with their ability to tirelessly repeat simple instructions, the idea of calculating physical forms became possible. In the early 1970s, the computational theorists George Stiny and James Gips coined the term "shape grammars" for a process they developed that calculated 2D and 3D visual forms in order to explain them, applying

> "The generated results are outside the norms of beauty and aesthetics. It's a new aesthetic." —Matias del Campo

these grammars to painting, sculpture, architecture, urban design and ornamentation. "Really, shape grammars are for anyone who wants to design visually, whether in art, architecture, urban design, or in engineering and product design," says Stiny, who is now a professor of architecture at MIT. "They can be used to link form, function, and material in sundry ways at any scale, from nano to gigantic."

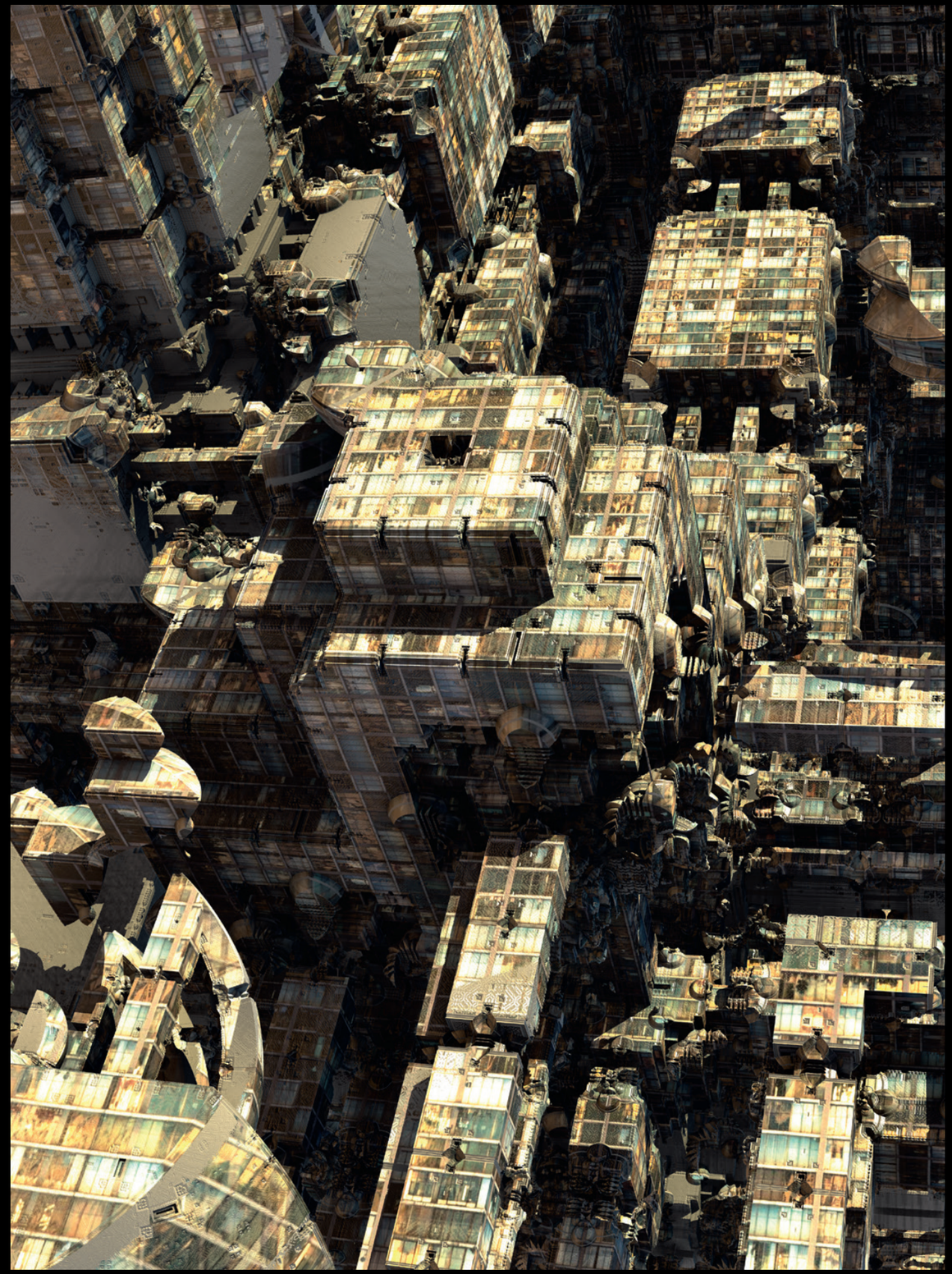
Stiny and Gips were inspired by the contemporary work of linguists like Noam Chomsky, who were exploring natural languages and trying to understand the rules that underpin them. Yet shape grammars threw up challenges not present in the grammar behind natural languages. While for Chomsky two letter As side-by-side are simply two As, for Stiny, interpretation of these forms lies in the eye of the beholder: "There may be an M, Vs in many sizes, snow-capped mountain peaks, cat's ears, and whatever my eye finds." At its most fundamental, a shape grammar is a set of rules that are applied, step-by-step in order to generate a language of design. Each rule comprises two pictures, a left-hand picture that shows a shape's starting form and a right-hand picture that shows how it changes into a new form, such as by moving position or rotating. These rules are then applied, one after another, to a main shape that's being calculated.

For many years, the research behind shape grammars remained chiefly academic. "When I started out," says Stiny, "the reaction [from designers] was immediate: 'You must be crazy.' But as shape grammars were shown to do more and more, the reaction changed to something like this: 'Well, whatever you can see and do with shape grammars, there's more to my designs than that.'" Even today, Stiny believes that architects and designers who use shape grammars in their work keep quiet about it: "The very idea of visual calculating is still far too controversial."

Some architects did, however, begin to explore the possibilities. One of the first was Christopher Alexander, who wrote the book *A Pattern Language: Towns, Buildings, Construction* in 1977, in which he proposed a system of rules – or a "pattern language" – for designing harmonious buildings and spaces. More famous is Greg Lynn, who edited a special issue of *Architectural Design* in 1993 called 'Folding in Architecture', in which a series of essays argued for the place for computers in the design process. Although Lynn later remarked that the examples the issue exhibited were designed at a time just before software and hardware became capable of realising their potential, 'Folding in Architecture' nevertheless identified calculus as the driver of a new field of design – one that could explore curvature, and a state where the monolithic could exist in the same terms of form, structure and detail as the minute.

Another key player was John Frazer, who wrote the book *An Evolutionary Architecture*. The book explores form-generating processes, which it puts forward as a form of artificial life governed by a DNA of rules and code. Frazer examined Tuscan columns and established algorithms that could generate various different forms of them, developing a system that could synthesise buildings as code. For example, the code for Le Corbusier's Maison Minimum is FF803F71180EFE033F and the 3D form of Mies van der Rohe's Seagram Building is 10083EFE0F00.

Recreating architectural forms as code has been the focus of some of the more practical



manifestations of computational design. In 2006 a set of researchers developed software called CityEngine that could generate cities and simulate systems such as CO_2 production and crowd dynamics as a visualisation tool for urban planning. One of the researchers, Pascal Mueller, founded a company called Procedural Inc to market CityEngine, finding that city planners loved its ability to demonstrate the effects of density zoning, while architects could use it to show how prospective designs might look in context. Filmmakers, meanwhile, could use it to generate cities as backdrops, such as the Metropolis of *Man of Steel* and the San Fransokyo of *Big Hero 6*.

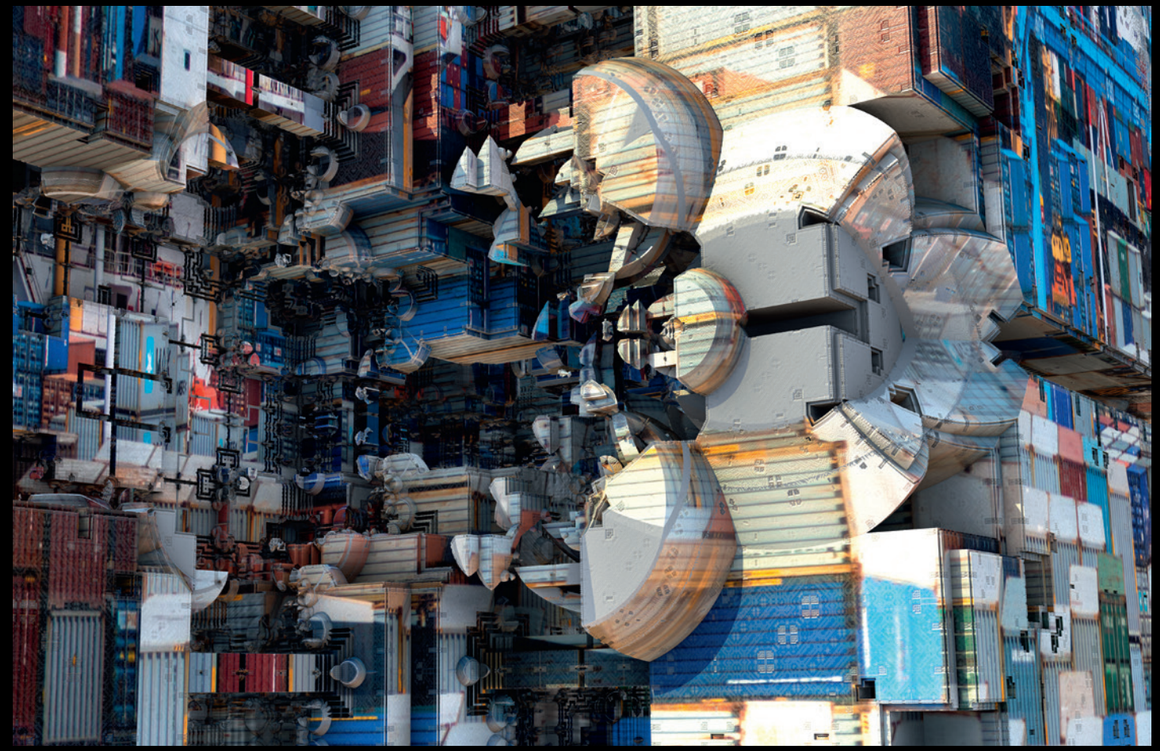
CityEngine works using rules that define urban forms, from building archetypes to street plans. When users paint these rules across an area of land, it can spool out acres of Haussmann-style Paris, or the cartoonish San Fransokyo, right down to the detail of a balustrade or doorknob. The software comes with hundreds of preset urban types, but designers can create their own too, programming the fundamental characteristics – the range of building heights, the numbers of windows, positions of doors, the width of streets. "We want people to have more time to figure out design choices," says Mueller. "It doesn't make a designer faster – it shouldn't – it should make the designer better, exploring more options, better quality, and seeing results."

Frazer's ideas, which originated from working on Cambridge University's Titan computer in the 1960s, were not just about recreating existing forms. He designed systems, modelled on nature, that could quickly evolve architectural concepts, test and evaluate them for suitability, and then generate new sets in response. In comparison to the traditional design process, these generated forms would often be unexpected, conjured by processes outside human imagination – sometimes seemingly mundane and sometimes beautiful sinuous forms. Sandra Manninger, who founded Span Architects in Vienna with Matias del Campo, sees this property as a core part of its appeal. "Best case, it's something you've never dreamed of designing," she says.

The promise of the unknown is one of the attractions of *No Man's Sky*, a forthcoming video game being developed by Hello Games, a small UK-based studio. Players start the game on the edge of a galaxy of 18,446,744,073,709,551,616 procedurally generated planets, with the aim of embarking on a journey to its centre. The terrain of each world is unique, comprising land, lakes and seas; barren deserts and ice, savannahs and lush jungles, all influenced by factors such as the location of the planet in relation to its sun. The creatures and plants are also procedurally generated, exhibiting different sizes, colours, and even vocal calls. It means that even *No Man's Sky*'s developers don't know exactly what's in the vast galaxy. It's a romantic notion, one that fits neatly with the game's sci-fi setting, promising that every player will encounter places, sights and sounds that no one has ever seen before.

Yet this heady idea also means that such generative forms are dislocated from the designer's hand, a property that Frazer was already well aware of in *An Evolutionary Architecture*. "When we 'design' [using this process], we are clear in our intentions, but perhaps 'blind' to the eventual outcome of the process that we are creating," he wrote. "This 'blindness' can cause concern to those with traditional design values who relish total control." The ramifications perhaps go beyond concern, however: the process has the potential to change the nature of being an designer. "The role of the architect here, I think, is not so much to design a building or city as to catalyse them: to act that they may evolve," wrote Gordon Pask, one of the forefathers of cybernetics, in his foreword to Frazer's book. But Frazer was careful to emphasise that the initial spark still came from human creativity, something with which Mueller agrees: "[CityEngine is] a tool, not the computer taking over. Identikit buildings are not a result of tools, they're the result of architects' decisions. It's the artist who decides what the tool's doing."

It's important to note that generative processes are not random. They don't create forms without sense or reason to them. *No Man's Sky*'s galaxy is deterministic, mathematically generating from a seed number (the mobile phone number of a member of the development team, as it happens), which when applied to the player's location in the galaxy generates the world around you through a set of interrelating rules. It's completely consistent every time you visit the same place, down to the scatterings of rocks at your feet, the shoreline



in the distance and the stars in the sky. Hello Games's creativity lies in defining those rules, which in *No Man's Sky* generate worlds that evoke the sci-fi book covers that inspired the game. *No Man's Sky* is just one of many games and related platforms that use procedural generation. Another, *Dwarf Fortress*, uses it to make a world and its history, writing stories of the societies that have lived there and left stories and ruins behind them before the player arrives. *Shadow of Mordor*, based on *The Lord of the Rings*, uses the technology to model the social dynamics of an orc army. SpeedTree is a system that generates realistic trees so artists don't have to make video-game forests by hand. Or there's Angelina, made by the computational creativity researcher Michael Cook, which can generate entire games, picking graphics from the web, and devising rules and goals. These implementations are allowing big and encompassing games to come from very small teams of people, getting round the serious problem in the game industry of simply making enough things for players to see and space for them to explore. Small studios can't afford to employ the hundreds of artists that made, for example, the blockbuster game *Assassin's Creed Syndicate's* detailed Victorian London setting and it's becoming increasingly difficult for the biggest developers too. Computational design presents a potential way out of the problem.

Daniel Brown's work, which often uses the endlessly recursive nature of fractals as its base material, is a fine example of this. Brown is best known for *On Growth and Form*, a series of videos of gracefully growing artificial flowers that was featured at London's Design Museum (2003-2004), as well as for creating images from 3D fractals that look like buildings. Using the free software Mandelbulb, he flies around a fractal form to find interesting places and then renders them, applying textures from photographs over their surfaces and lighting them in a way that mimics natural illumination. The images are bizarre and surprising, instantly familiar yet alien, and always beautiful in their uniformity.

Computational design naturally involves a very different way of working to traditional areas of the discipline. Brown doesn't start with sketches; he starts with generating and then observes the results. "I typically start with a form in mind; there are essentially off-the-shelf mathematical formulae for particular shapes. You can easily combine these to get an approximate model, and then add or subtract complexity and noise to vary and evolve the form," he says. "But there are tools now, such as Mandelbulb, that let you do this visually. It's getting easier every day." Manninger and del Campo describe their process as a "constant cloud" of research and testing, quickly creating genealogies of designs to build up a set of possibilities, with the aim of finding rigorous and appropriate solutions on which to focus. For a 2011 exhibition at MAK, the Museum for Applied Arts in Vienna, Span explored 3D recursive geometries and fractals in a similar fashion to Brown. "So what happens is that we change a little bit of the algorithm and every time it generates a different result, which is always surprising. So you can steer it but you can't define it," says del Campo.

Manninger and del Campo use programming languages like Python and 3D-design applications like Maya, which can process Python scripts and then export the results to other applications, like the 3D printing-based Rhino. But each project demands different tools and methodologies. "We haven't got a protocol," says Manninger. "We never use the same software." Instead of grandly inscribing lines on a sketchpad to be realised in massive physical form, practices like Span are more like curators of ideas and authors of processes. "There's a shift from architect to programmer, to tool-builder," says del Campo. Brown agrees: "You create systems, not strict designs. That means all designers have to think systemically rather than in terms of pretty pictures." But Brown is adamant that what he practices isn't necessarily arcane and only accessible to savants. It's quite possible to start experimenting using Processing, a language developed to explore computational generation for which a large, supportive community is already in place.

Meanwhile, the Design Research and Computational Science departments of the Autodesk corporation have collaborated on software called Project Dreamcatcher, which automates some of the tool-making that computational design requires. Dreamcatcher is a CAD system that can automatically generate many forms, from bicycle frames to bridges, by meeting a designer's goals and constraints, including function, material and cost. A designer might, therefore, ask for a chair that can support 100kg, be made from carbon fibre, and which weighs as little as possible. When reviewing results, they can adapt the goals and constraints, thereby generating new sets of results. It's a process in which the designer takes the role of decision maker rather than originator.

Yet the prevalence of tools in all discussions of computational design gives other designers pause. Marcelo Spina, director of the Los Angeles-based interdisciplinary design studio Patterns, says, "I can't let a tool become a burden for what I want to do." Spina is also concerned by the idea that one could rationalise design as far as to be able to create an app that generates your own house. "Architecture is too complex. People like Frank Gehry figured out how to rationalise part of the process, but there's another part that's completely human and based on artistry." Similar criticisms are made of procedural

Instead of grandly inscribing lines on a sketchpad to be realised in massive physical form, practices like Span are more like curators of ideas and authors of processes.

generation in video games, where generated worlds and levels can feel formulaic, lacking the wit of human design.

Even today, it's hard not to see Frazer's 20-year-old ideas about organic architecture in terms of science fiction. Greg Lynn and his studio, Form, became celebrated for biomorphic architecture, or blobitecture, the creation of organic shapes through computational design. But it was also questioned: was it form-making for its own sake? Was it too concerned with the tools and not with the building itself? And was it before its time? Frazer's longer term goals were to connect his generative systems to building processes, such that they were self-generating and self-constructing, perhaps using molecular engineering, nanotechnology,

or the genetic engineering of plants to achieve this. He even wondered if it could produce the start of artificial life.

Today's technology finally shows glimpses of Frazer's sci-fi future, however. MX3D is a steel canal bridge in Amsterdam designed by Joris Laarman using Project Dreamcatcher that is due to open in 2017. The bridge will be 3D printed in a factory by robotic arms, which also print their own track across the bridge as they produce it; a slight deviation from the original plan to print the bridge in situ over the canal, which was deemed to breach health-and-safety regulations. In fact, the development of 3D printing has helped to give new significance to computational design. In tying the act of production to the act of designing, and in opening up the ability to make unique, one-off objects, 3D printing complements the individualistic, adaptive, nature of computational design. "I'm convinced that will continue growing in the construction industry," says del Campo. "There will be a direct connection between the computational model and the physical reality of the building."

Even in 1987, the architecture theorist Charles Jencks saw the potential of this in his book *The Language of Post-Modern Architecture*. "The new technologies stemming from the computer have made possible a new facility for production," he wrote. "The emergent type is much more geared to change and individuality than the relatively stereotyped productive processes of the First Industrial Revolution[…] computer modelling, automated production and the sophisticated techniques of market research and prediction now allow us to mass produce a variety of styles and almost personalised products. The results are closer to the 19th-century handicraft than the regimented superblocks of 1965."

But if we can make unique things, there's a resulting pressure to design them too. Through services like NikeiD, customers can customise purchases, creating an expectation of similar customisability in other goods. Yet, as Brown says, "You come into the paradox that people want customisation and unique products, but most people aren't designers, and have no inclination to become one. How do you manage everyone wanting a custom product? You can't have as many designers as there are consumers, so we're going to have to look at semi-intelligent processes to guide and help and advise them." The Italian architect Carlo Ratti, who directs MIT's Senseable City Lab, sees all this as part of a grander movement in design. "A solitary, top-down, Promethean attitude has characterised the 20th-century architect," he says. "Today, I believe that more collaborative approaches are coming back, rooted in internet culture and in the new paradigms of online collaboration."

The days of the designer as ultimate author are over; perhaps it's time to become the originator of ideas instead. Computational design reframes the architect as an enabler of new forms, rather than their dictator. It recasts the designer as a toolmaker. CityEngine's Pascal Mueller sees the rapid growth of cities as one important way in which computational design can be used for good: "By 2050, the population of cities is due to double, and cities will grow in both height, area and overall density. There are almost no tools for that, and the only way to deal with this is with tools and communication."

"Tools can help us make better design decisions and be aware we're making decisions that have an impact on people's lives," adds Manninger. "We will communicate through the tools better, and communicate with experts from fields outside our own. I dream of tools that develop and help us to have expertise to engage with one another and the things we create in more exciting ways. To engage with materials in more exciting ways, to create more intricate, sensual objects."

In other words, computational design isn't necessarily about happy blob-making or rationalising creativity anymore. It's about process and communication, and working better together. But how will it grow beyond small and focused design practices? "Some things are already in motion," says del Campo. "The construction industry has to change, and that is already happening, slowly. Awareness of these possibilities has to be bigger in the public's eye, and decision makers in politics and industry have to be aware. A mistake a lot of architects make is to think they are the decision makers, but we're not! We can make proposals and generate ideas, but it's up to the decision makers." And if all this should snap into place? "We would like to go as far as hardware will take us." END