INTRODUCTION

BRADY PETERS
TERRI PETERS

Smartgeometry (SG) was founded in 2001 as an informal network of designers interested in harnessing the powers of computation for architectural design. Friends and former colleagues Hugh Whitehead, Lars Hesselgren and J Parrish felt frustrated by the lack of resources and network surrounding computation and architecture and sought to redefine ways that architects could use digital tools. At first, the trio of architects drew on their network of friends and collaborators such as computer scientist Robert Aish, academics Robert Woodbury and Axel Kilian, and experimental practitioners architect Mark Burry and engineer Chris Williams to put together a few modest conferences and workshops. These began with a lecture and workshop in 2003 in Cambridge, UK, then in 2004 at the University of Waterloo, Ontario, Canada, where the focus was on software development, new tools for architects and engaging with ideas outside the boundaries of 'architecture'. These early workshops provided inspiration and a testing ground for the creation of new parametric software GenerativeComponents™ (GC) that was introduced to the group by Robert Aish and Bentley Systems. Rather than being concerned solely with software or form-making, SG focuses on the creation and application of digital tools and technologies, and in cross-disciplinary fertilisation of emerging ideas in practice. In workshop groups, designers are able to work on projects 'off the books', away from their offices or university settings, creating pure explorations of technique beyond the confines of the design project. SG embodied new ideas and new ways of thinking. The event now spans six days, with a four-day curated workshop and two-day public conference, and attracts more than 300 international participants and attendees each year.

WHY GEOMETRY?

Architectural design software at the time SG was founded was created by software developers using object-oriented programming that almost literally translated software 'objects' as building 'objects'. SG co-founder Lars Hesselgren has written that they wanted to build new design tools and founded SG as a rejection of these conservative influences that promoted computer-aided design (CAD) solely as the organisation of building components.[1] In order to be free of these predefined tools and have a higher-level discussion of building form in terms of first principles, this led to a discussion of geometry and mathematics. As this is a more generic approach, thinking of architecture and form in this way allowed them to share computational tools between disciplines. It allowed architects to design conceptually and create their own custom 'objects' rather than use the specified objects provided by their CAD software.

As Robert Aish explains in his chapter, SG explores the ideas of design computation, with the notion that there is a distinction

1 Design explorations at SG 2006, Cambridge, UK.
Conceptual geometric design explorations using GenerativeComponents™.

2 'Responsive Acoustic Surfaces' workshop cluster at SG 2011, Copenhagen, Denmark.
Participants engage in a design discussion.

Digital models from the 'Responsive Acoustic Surfaces' workshop cluster at SG 2011, Copenhagen, Denmark.
Participants work on digital models of hyperboloid geometry using a variety of software.

'Use the Force' workshop cluster at SG 2011, Copenhagen, Denmark.
Participants discuss design and computation.

between the generative description of the building, and the resulting generated model of the building. Therefore SG is more about the exploration of design intent and how this is inscribed in the design tools and the design environment, rather than specific technology for the integrated delivery of building projects. It is about designing a system, rather than working on a more detailed 3D model.

SG is an agile network. It is purposely structured to be able to react to and reflect ideas in contemporary practice; there is no overriding goal or charter. The idea is to engage with current issues and debates in a collaborative and non-competitive environment. Digital design leads logically to digital fabrication. Over several years, but culminating in 2010 where it was a central feature, the event embraced digital fabrication, interaction and simulation with 'workshops' that more equally split experimentation in digital and physical realms. As Xavier De Kestelier and Shane Burger explain in their chapter, the evolving workshop structure is due to shifts in participants and leadership. The earlier events attracted lower numbers of workshop participants and leaders and these were almost exclusively from professional practice. Recent SG events have had multi-day programs with larger audiences and an increased focus on academic and research questions. This shift is discussed in the chapter by CASE, where they identify the move away from the pragmatics of designing for construction of buildings, towards workshops based not only on research and experimentation, which does not necessarily rule out the practical building issues, but also on creative explorations using these same methods. The five current SG Directors are all from architectural practice, but each year attendees and workshop leaders come increasingly from research and academia.

TALKING ABOUT COMPUTATION

'The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from them.'[2] This statement from computational pioneer Mark Weiser in 1991 is relevant to architectural practice today. Computation is everywhere; should it really be the medium and not the message? Architects desperately need to talk about computation, and over the past decade SG has provided the only experimental workshop-based discussion forum on this topic. It is not enough to say computation is ubiquitous in our field; it is not 'just' a tool – there can be no doubt that it is fundamentally changing architecture. Computation is not what architecture is, but if architecture can be understood as a practice, concerned with technique, then computation is a technique intricately connected to designing for meaning and experience in architecture. Even architecture as edifice, separated from any discussion of technique, reveals the tool of the maker. While meaning in architecture can come from symbols and symbolism in the building itself, it also comes from the experience of that building.[3] Therefore the better we can simulate the experience of architecture, the better we can design for it. The technologies explored and discussed at SG are still quite visible. However, one hopes they will be customised,

3

3 Prototypes and analysis from the 'Responsive Acoustic Surfaces' workshop cluster at SG 2011, Copenhagen, Denmark.
1:10 scale prototypes were tested for their acoustic performance and this data was used to inform the design of the full-scale prototype wall.

4

4 Workshops at SG 2011, Copenhagen, Denmark.
View of participants and workshop clusters.

tested and engaged with by architects even as they continue to be woven into the fabric of the design environment.

Of course, the mere fact of using a particular computational technique does not guarantee good architecture, the same way that using the same pen as Norman Foster will not guarantee a great building. But why shouldn't architects share techniques and tools? It would be pretty silly if architects each had to invent our own pens, drafting boards and drawing conventions. So while it is the building that matters most, rather than focusing on the process of design and making, in the context of design it is critical to acknowledge that design processes are changing, and SG is at the forefront of this change.

CREATING KNOWLEDGE AND TECHNIQUE

The ways that computation and architectural design are explored at SG are unique. The idea is to nourish a collaborative environment where participants feel as though anything can happen. The theme of the event is set in advance and then workshop leaders apply to lead a cluster based on their own research: for example, in 2012, 40 detailed applications were submitted for 10 positions. Participants also apply to join workshops, submitting portfolios and statements of interest, with only 100 selected. Carefully curated by the SG community and directors, the selected workshop themes are developed and in the four days, focused questions of design, digital technique and physical making can be explored. In contrast to many other design workshops with traditional student–teacher dynamics, workshop leaders do not bring work they have done earlier to get 'students' to build, participants do not come to learn something they know nothing about, and experts do not arrive with 'answers' to disseminate. Research and knowledge is created during the workshops. The challenge is not to construct a research question that can be 'answered' in four days but rather to construct a line of thinking that can be investigated intelligently and discussed through experimentation. The SG environment is part research and part professionally focused, which seems to inspire productivity, as participants work long into the night to actually *do* something as a group within the given time, to produce some results to share with the wider group by the end of the workshop, and to work together. In Robert Woodbury's chapter, he calls this the 'flow' of design computation. There is of course a healthy fear of failure and underlying pressure to make it perfect, or at least beautiful. This is architecture after all.

SG makes no claims to produce 'architecture'. It is not about form, it is about how we arrive at form. SG is about technique. There are, of course, many valid ways to design and SG celebrates this plurality of concept. It is not the place for design crits. In the four-day workshops, there simply is not time. Instead, techniques and tools are developed and tested. Participants find where a tool hits the wall, then how to mash it up with other tools and make it work better. It is like building a racing car – how fast and how hard can we push this machine – not how nicely can we drive it.

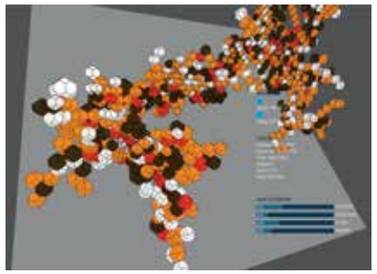
5 Aggregate module structure from the 'Agent Construction' workshop cluster at SG 2011, Copenhagen, Denmark.
The participants operate as 'agents', building and altering the structure without pre-made drawings or plans, instead being guided by rules responding to local conditions – 'here and now'.

5

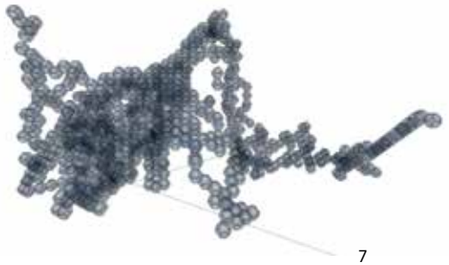
6 Virtual agent model from the 'Agent Construction' workshop cluster at SG 2011, Copenhagen, Denmark.
In the computer model, a swarm of virtual agents gradually and collectively build up a structure. Different agents are guided by different rule-sets, and the only communication between them is through the environment which they manipulate and which in turn affects their behaviour.

7 Visualisation of scan data from the 'Agent Construction' workshop cluster at SG 2011, Copenhagen, Denmark:.
The emerging physical structure is continuously scanned, and the data imported into virtual formats for analysis and further processing. In this format it can be directly analysed beside the virtual agent models, or used as an input in these.

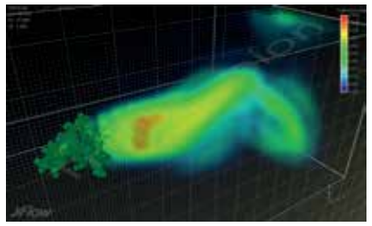
8 Airflow simulation from the 'Agent Construction' workshop cluster at SG 2011, Copenhagen, Denmark.
The scan data allows for simulation of the structure's performance, here through a fluid dynamic simulation in X-Flow.



6



7



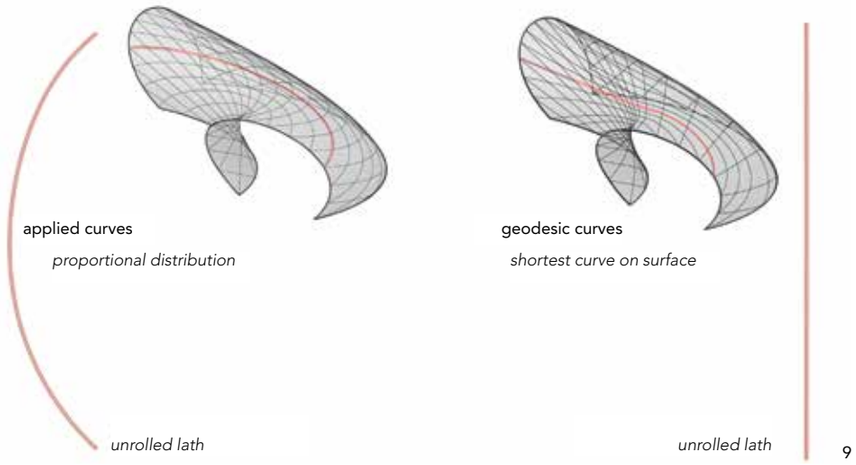
8

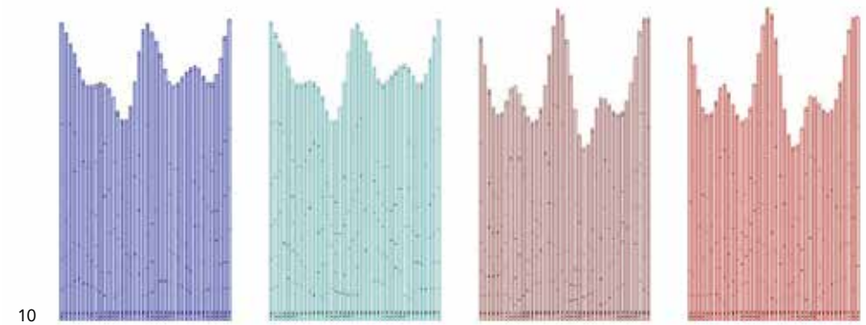
CODING

'The ability to 'read' a medium means you can access materials and tools generated by others. The ability to 'write' in a medium means you can generate materials and tools for others. You must have both to be literate. In print writing, the tools you generate are rhetorical; they demonstrate and convince. In computer writing, the tools you generate are processes; they simulate and decide.' Alan Kay[4]

When a designer writes a script to solve a problem, the algorithm becomes part of the design and may then be explored in a creative way. But, as Fabian Scheurer explains in his chapter, algorithms are both a description of the problem and the solution. They define the solution space and they are built around the definition of the problem. He argues that design is all about decisions and that delegating these to an algorithm always means following predefined paths. Often the use of existing tools leads to existing solutions. Through the creation of new tools, new ways of thinking and new solutions can be found. Algorithmic thinking means taking on an interpretive role to understand the results of the generating code, and understand how to then modify the code to explore new options, and to speculate on further design potentials. As designers, we are influenced by the tools and techniques that allow us to realise our visions. It has been said that the tools determine the boundaries of art, and that it is the use of the right tools for the thing that one is making, and a deep relationship between the use of the tool and its formal results, that establishes the potentials of what can be made.[5] With computation, the boundaries of what can be made just got a lot bigger. Parametric systems and computational tools have enabled the realisation of projects that were previously inconceivable. Nicholas Negroponte introduced the concept of bits and atoms, arguing that atoms make up physical, tangible objects around us – the architecture that we inhabit – while our design environment and our digital models inhabit the space of the bits – the information that is contained within the computer that we use to design.[6] So, how does this relationship affect the architecture we design? This relationship between bits and atoms is becoming blurred. Not only do the experiments undertaken at SG work in between physical and digital realms, but design tools are increasingly used that simulate real-world performance and provide feedback on designs. Computational tools become co-creators in design, extending the intellect of the designer, and so the role of the designer becomes one of tool builder, of interpreter of results, and of a guide through solution spaces. In his chapter, practitioner Neil Katz explains that the technology needs to disappear: it is the design intent and process that is more important than the tool itself.

'Software modified by the designer through scripting, however, provides a range of possibilities for creative speculation that is simply not possible using the software only as the manufacturers intended it to be used. Because scripting is effectively a computing program

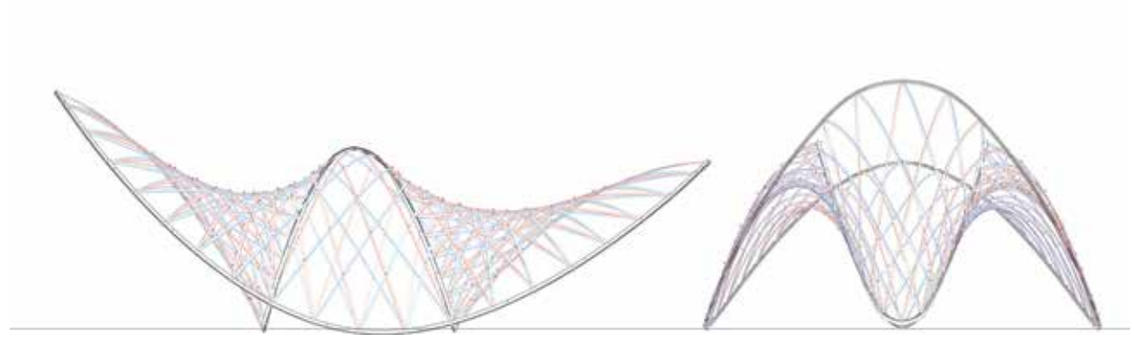




10

9 Geodesic curves from the 'Gridshell Digital Tectonics' workshop cluster at SG 2012, Rensselaer Polytechnic Institute, Troy, New York, USA.
Geodesic curves allow for complex curvature from straight segments.

10 Fabrication layout of laths from the 'Gridshell Digital Tectonics' workshop cluster at SG 2012, Rensselaer Polytechnic Institute, Troy, New York, USA.
Geodesic laths are unrolled with precise lengths and spacing of nodes for pin joints.



11

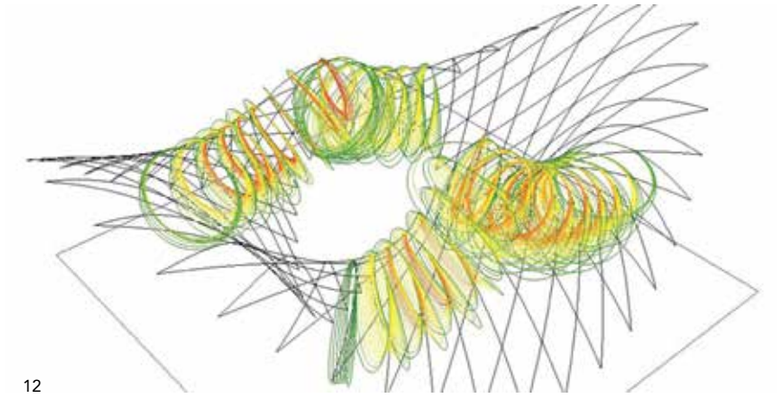
11 Elevation from the 'Gridshell Digital Tectonics' workshop cluster at SG 2012, Rensselaer Polytechnic Institute, Troy, New York, USA.
Diagrams of the lattice gridshell identified the location of each piece in the four-layer lath system.

overlay, the tool user (designer) becomes the new toolmaker (software engineer).' Mark Burry[7]

SIMULATING EXPERIENCE

Architecture can be thought of as drawing, but should be thought of as simulation. Architecture is the act of imagining a building at a remove from its construction, and then communicating this concept for others to build. To date, the imagining and communicating has been largely through drawing. However, it is not necessarily drawing that defines architecture, but this ability to create an abstraction of the building through some means. Through the drawing, the architect is able to imagine how light and space and material relate in the creation of architecture. Although largely within the mind of the architect, this simulation of effect and experience is a necessary part of architectural design. The pragmatic aspects of performance can be simulated as well. The digital design environment can be a design partner for this simulation of architecture. Through the adoption of new technologies, the creation of design techniques, the coding of custom design tools and the gaining of critical performance feedback, the abilities of the architect are extended.

SG was founded on the premise that a first-principles exploration of geometry in relation to design intent could benefit architectural design. The development, discussion and dissemination of these explorations of technique have been central to the SG workshops and conferences. The SG community explores these through parametric design, computer programming, digital fabrication, interactive design, simulation and optimisation. The scope of these approaches is enlarged at each yearly event. SG has been, and continues to be, a place where these concepts are not only discussed, tested and critically reflected upon, but critically, a place where this knowledge is created. A place for designing, coding and building.



12

12 Curvature analysis from the 'Gridshell Digital Tectonics' workshop cluster at SG 2012, Rensselaer Polytechnic Institute, Troy, New York, USA.
Curvature analysis determines radii of curvature to verify the minimum radius allowable for the bending of lath segments.

13 Completed gridshell from the 'Gridshell Digital Tectonics' workshop cluster at SG 2012, Rensselaer Polytechnic Institute, Troy, New York, USA.

13

REFERENCES
1 Lars Hesselgren, 'Smooth Is Not Enough', 18 January 2009, http://www.core.form-ula.com/2009/01/18/smooth-is-not-enough/ [accessed 12/09/2012].
2 Mark Weiser, 'The Computer for the 21st Century', in *Scientific American*, September 1991, pp 94–104.
3 Stan Allen, 'Introduction: Practice vs Project', in *Practice: Architecture, Technique and Representation*, G+B Arts International (Amsterdam), 2000, page XIV.
4 Alan Kay, 'User Interface: A Personal View', in Brenda Laurel (ed), *The Art of Human–Computer Interface Design*, Addison-Wesley (Reading), 1990, p 125.
5 Jose-Manuel Berenguer, 'Interview with Jose Manuel Berenguer' in *Mosaic*, 2005, http://mosaic.uoc.edu/2005/09/09/jose-manuel-berenguer/ [accessed 10/02/2010].
6 Nicholas Negroponte, *Being Digital*, Vintage Books (New York), 1995.
7 Mark Burry, *Scripting Cultures*, John Wiley & Sons (Chichester), 2011, p 9.