Stefano Converso

Università degli Studi Roma Tre
Dipartimento di Architettura
Largo G.B. Marzi 10, 00153
Rome, ITALY
stefano.converso@uniroma3.it

Keywords: interoperability, scripting, design, digital fabrication

Research

Mathematics for the Design of Variation: The "Nagashima!" Lamp Prototype

Abstract. This paper describes the process for design, development and fabrication of a lamp prototype that worked as an experiment in "digital bridges" between different software, using Mathematica as the main source for design geometry that is passed dynamically to platforms for documentation, lighting analysis and rendering, as well as CNC machining for final fabrication. The process will be used as key for discussing the dynamic relationship between variation, design option and analysis, in relation to the richness of our design culture.

Digital Bridges

This paper covers the design, development and fabrication process of a prototype for a lamp called "Nagashima!". The whole process worked as an experiment in "digital bridges" between different software.[1]

With the term "digital bridge" we mean a procedure that establishes a dialogue between two or more software programs, allowing the transfer of data between them in a fluid and bi-directional manner. This means that every change introduced in one of them is easily viewable and noticeable in the other one or ones. This link challenges the usual method that confines each research or work within its own domain, as well as, of course, the software package associated to it.

In the domain of architectural design this is even more important since every project is by definition a multidisciplinary experience, involving several different people at different levels with a great variety of knowledge and background. This complexity is commonly handled through a linear process going from conceptual design to design development and engineering, and ending with construction documentation, fabrication and assembly.

Several authors have highlighted the new opportunity introduced in this process by the widespread adoption of digital instruments. Their presence all along the process provides a new continuity: the boundaries between the different players and the different disciplines involved in the process tend to blur [Chaszar 2006]. The new continuity established by software is defined by Bernard Cache [2003] as "associativity". Cache describes associativity as the only factor that would introduce a real change in the design methods all along the building process. This means that digital models are dependent on each other, and that those relationships are maintained from the very beginning to the final built outcome of the process.

But the application of such a concept is problematic since it makes people involved in the process go out of their comfort zone. It's a challenge that goes from being technical and digital to becoming professional at a larger scale. This is now being widely investigated not only by software companies, but by engineering and fabrication

Nexus Netw J 15 (2013) 549–564

DOI 10.1007/s00004-013-0166-4; *published online* 21 September 2013

companies, institutions, associations of professionals. A specific domain has arisen together with new professional figures, dedicated to this know-how that stays at the edge between mathematical and architectural domains, with software giving a shared territory of experimentation [Lynn 2003]. At first, this phenomenon generated groups of young professionals called "computational groups" within practices of architecture and engineering: they were particularly dedicated to projects where complexity of building geometry for construction was considered high [Paoletti 2008], and therefore needed precise definition of all components of the structure, as early as possible in the process. During this pioneering period, authors dedicated big efforts to the formulation of the principles of such an architecturally focused use of geometry [Kilian et al. 2008]. Academia can be a great place to establish new links and experiment with new methods since it can involve in one place a mutidisciplinary team, going from engineers to architects to mathematicians, as required by these new forms of research, not always easy to achieve within the constraints of daily practice. One of the programs that has focused more on this issue since its beginning in 2004 is the "Product-Architecture Lab", a Master course started at Stevens Institute of Technology, in the United States. The program featured a class called "Interoperability" where students were highly exposed to the interaction between different software programs belonging to the often opposite scientific domains of engineering and architecture. This simple but powerful choice gave to the program great freshness and efficiency.[2]

Of course the premise for this opportunity to change is software. Software study is at the core of every contemporary research on design methods, but to become really effective it has to be considered a cultural rather than a technical issue. Software needs to be used not as a "tool" but as an "instrument".[3] In order for software to become an instrument, it needs to be shaped or adapted to external needs that are not exactly included in its original purpose. In that sense, a crucial role has been played by the so-called "scripting", a programming language linked or embedded in a software program that makes it possible to build a custom procedure within it. The point in our research is that these procedures do not have to remain confined to one single software application. That doens't radically change design habits. These procedures have to become the bridge between different software applications, which means different people, and different mindsets. This is exactly the point where the human factor comes in, and when many notions and habits of design are challenged.

In the School of Architecture at Università degli Studi Roma Tre we have set up a series of courses dealing with this issue from different sides (mathematics, structural analysis, environmental analysis, parametric modeling, architectural design). We founded a cluster of professors and researchers running the courses and interacting with each other named "formulas". This interaction involves first of all talking and setting up relationships with people, colleagues and possible consultants. It also involves open-mindedness and dedication to interdisciplinary research, not always easy to pursue in domains that tend to be very specialized. We are increasing the number and quality of interactions, and the experience of the "Nagashima!" lamp prototype is a good example of the procedure made possible by the collaboration and involvement of different people.

The procedure: a balance between coding and modeling

The starting point of the "Nagashima!" prototype is a code written using the API (Application Programming Interface) environment of the Revit Architecture 2010 platform, developed by Autodesk and commonly used in the architectural design field.[4] The interesting aspect of this software is that it is an object-based modeler, where the base objects of the program are customizable. Revit Architecture allows for the definition of objects as parametric entities, defined graphically as dependent on one or more parameters.

Such objects are called "families": they can be developed in a dedicated environment, where the available tools are pretty similar to more advanced and robust parametric modelers such as Catia, ProEngineer, SolidWorks, among others. Reference planes, constraints, levels, locked dimensions and parameters, together with geometry primitives and Boolean operations allow for the definition of parametric relationships between objects in the 3D scene. We encouraged the students to identify as early as possible those objects and their definition in the digital domain, as actual pieces that come out of a fabrication process, by showing them examples from the manufacturing industry. Once defined, an entity defined with its parameters can be instantiated into an assembly file, called in Revit a "project".[5] The project is the environment where scripts act with their procedures. The simple code we developed sets two iterative procedures: it copies an element (a family instance) several times and for each copy it also changes the value of one or more parameters of the element.

We developed the code as an "Advanced Replication" experiment. We wanted to build within Revit a procedure similar to what was available in Catia at the time as "Power Copy".

So we teamed up with a scripting professor of the "formulas" cluster to produce a "base script" to be given to students as a custom instrument in Revit, which they could use for their designs in the class called *Tecniche Parametriche di Progettazione* (Parametric Design Techniques).

We designed for students a procedure, a "playpool", which could also be interpreted as a design space.[6] We introduced them to the concept of conceiving not a simple object, but a system able to generate a series of objects, defining variants and invariants for its geometry.

Fig. 1. SHoP Architects: brackets for roof of a post-Katrina hurricane community center, New Orleans, USA. Each piece is an instance of a parametric object defined with the software Catia, using a "Power Copy" command

The first step for the students was to conceive and model a Revit family object as a constructive component whose geometry had to be parametric, that is inherently variable. Several examples were given of works of architecture based on this concept of variable components, such as New York City based SHoP Architects[7] design for the post-Katrina hurricane Community Center (fig. 1) and the Virgin Atlantic Clubhouse.

The students started to experiment with this instrument, which was challenging them at the technical level as well as at the design level: they had new possibilities but linked to new (possible) applications. Another new concept introduced by this experience is that process becomes central, while architectural design usually focuses more on the final object. Here, a design can grow from existing techniques, procedures and suggestions, with an attitude that is more similar to sampling (or in-venting) than to "creating" from scratch. After determining this first object and its possibility for variation, they had to introduce a rule for the instantiation in the code: the first example embedded all the variation into the definition of the family object. The code in this sense, is very basic: just copy the instances along the *z*-axis and then change a parameter of two different types of the two families (figs. 2-4).

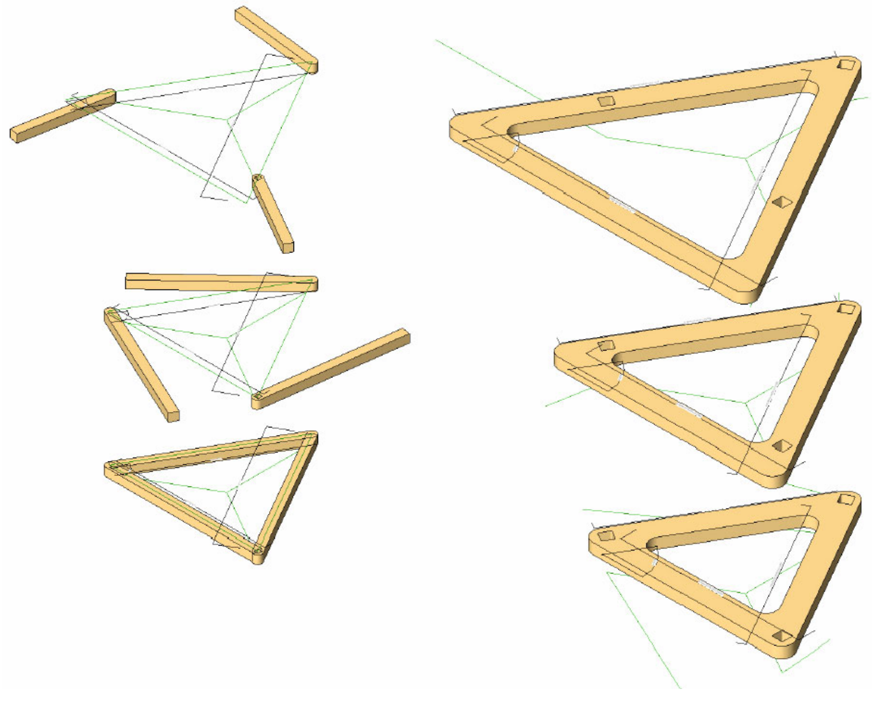


Fig. 2. Instances of the two parametric components of a student project ("Blossoming Flower Lamp"). As they are copied along the vertical axis, they change their shape according to the variation of internal parameters. Renderings by student Stefano Guarnieri, Tecniche Parametriche di Progettazione 2010

```
works with families: "triangolo.rfa" and "listelli.rfa"
using System;
using Autodesk.Revit;
using Autodesk.Revit.Symbols;
using Autodesk.Revit.Elements;
using Autodesk.Revit.Geometry;
using Autodesk.Revit.Structural.Enums;
using System.Windows.Forms;

public void BFLamp()
      {
         {
            // loading family "triangolo"
            Autodesk.Revit.Document doc = Application.ActiveDocument;
            string FileName = @"C:\Stefano\bfl\triangolo.rfa";
(omissis - exceptions for not loading the elements)

             // loading family "listelli"
            string FileName1 = @"C:\Stefano\bfl\listelli.rfa";
            FamilySymbol symbol 4= null;
            string NomeTipo4 = "listelli";
(omissis - exceptions for not loading the elements)

           //procedure
            int i;
            //n: max number f copied elements - i: visibility parameters into the family
            int n = 56;
            for (i = 0; i <= n; i = i + 1)
            {
               XYZ punto = Application.Create.NewXYZ(0.0, 0.0, 0.0);
              FamilyInstance famIst1 = doc.Create.NewFamilyInstance(punto, symbol1,
StructuralType.NonStructural);
              FamilyInstance famIst2 = doc.Create.NewFamilyInstance(punto, symbol2,
StructuralType.NonStructural);
              FamilyInstance famIst3 = doc.Create.NewFamilyInstance(punto, symbol3,
StructuralType.NonStructural);
              FamilyInstance famIst4 = doc.Create.NewFamilyInstance(punto, symbol4,
StructuralType.NonStructural);
              // calling the generating parameter
              Parameter param1 = famIst1.get_Parameter("PASSO");
              Parameter param2 = famIst2.get_Parameter("PASSO");
              Parameter param3 = famIst3.get_Parameter("PASSO");
              Parameter param4 = famIst4.get_Parameter("passo");
               param1.Set(i);
               param2.Set(i);
               param3.Set(i);
               param4.Set(i);
            }
         }
         catch (Exception e)

      }
   }
```

Fig. 3 - Revit code for the "Blossoming Flower Lamp"

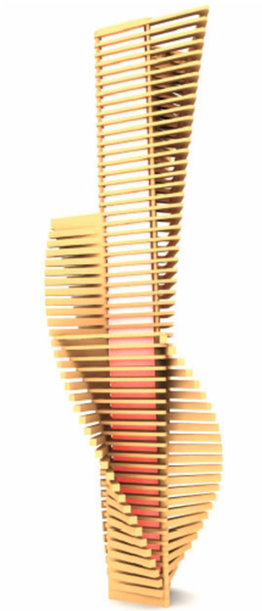


Fig. 4. Images of the student project for the "Blossoming Flower Lamp". The lamp is based on a continuity of vertical lines along the *z*-axis, due to constructive constraints: the lamp is assembled in a manner similar to spiral staircases, with steps attached and cantilevered from a core element

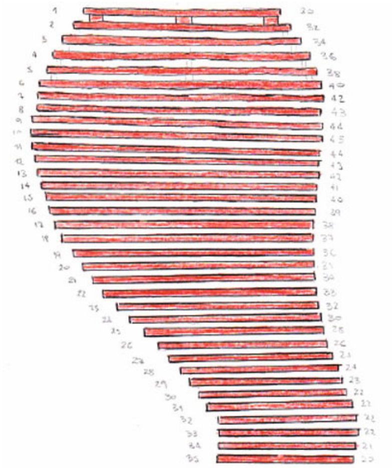
Definition of the "Nagashima!" prototype

The starting point of the "Nagashima!" lamp prototype was the above mentioned "design system": a) a parametric constructive component, modeled as a Revit family object; b) a rule for its instantiation, embedded into a script.

The variation introduced by the Nagashima example is in the balance between the two poles above: the parametric component becomes very simple, while the rule for its instantiation becomes more complex. The base element this time is a circle, variable only by its radius.

In the previous example the center point of the instance was always located along the *z*-axis, while in this lamp the intention is to locate the center points of the circles along a curved axis. Both the curved axis and the variation rule for the parameter become controlled by mathematical functions. The control over the functions driving the overall form of the lamp envelope is achieved by connecting the Revit script with a new file, developed with another software application: Mathematica. By changing the type and the parameters of the equations, Mathematica is used to visualize quickly the different hypotheses for the lamp, but at same time guarantee a precise control over it. The mathematical tool becomes a design instrument.

The first freehand sketch of the student (fig. 5a) shows this concept clearly: the drawing doesn't represent a form closer to the final one: the process then is not focused on "achieving a precise version of the starting form". What the sketch shows is the idea of a procedure generating a form. This is an important shift in how design is normally conceived: the final shape of the lamp was not 100% defined at the beginning of the process.



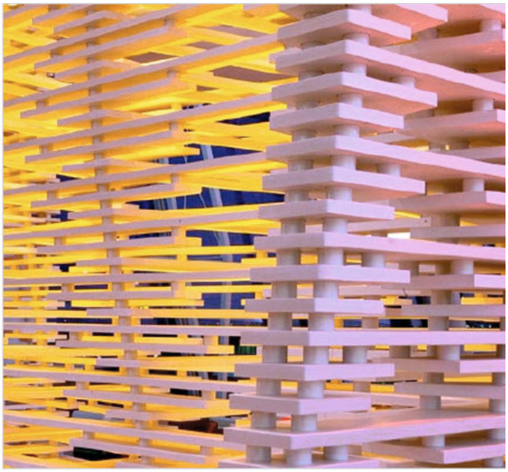


Fig. 5 (left). One of the first sketches of the lamp
Fig. 6 (right). The reference to an analogue exisiting system in SHoP Architects project for the Virgin Atlantic Clubhouse (JFK Airport, New York,Terminal 2)

The form was born through the interaction with other people, in this case a student of mathematics, who developed the part of the process in the program Mathematica. This interaction was highly encouraged by the course, where the issue of digital bridges was constantly recalled, by asking students to find external referrals for their modelling work. The introduction of "Mathematica" was easier since it is a tool known by students of architecture thanks to a parallel course taught by colleagues of mathematics of the "formulas" cluster open to the idea of academic collaboration.

Once a relationship is set up, the idea of digital bridges is then focused on achieving a constant feedback for the design development. Every idea of the designer in a digitally bridged system is elaborated in variable manner, and gets checked and negotiated throughout the process.

In the case of "Nagashima!" lamp the precise control over the shape in Mathematica is bridged with the script in Revit to generate the constructive components of the lamp (wooden circles) and imported into a rendering program in order to check the quality and intensity of light. Too many lamps today are generated with a focus on their form rather than on the light that they generate. In this case there was a clear concept for a light effect to be achieved: the generation of "light circles" on the walls.

The process moved on by generating several instances of the lamp. The first one, the simplest, was a variation of the radius over a vertical axis. In that case the script was a simplified version of the one used for the lamp project, the "Blossoming Flower" shown above in figs. 2-4.

```
Revit Script: C# - Revit API
(Parts common to previous one are not included)

//procedure
int 1;
int n = 33;
double f,g;

for (1 = 1; 1 < n; 1 = 1 + 1)
{

            XYZ location = Application.Create.NewXYZ(0.0, 0.0, 0.1*1);

FamilyInstance unico = Create.NewFamilyInstance(location, family,
StructuralType.NonStructural);
Parameter generico = unico.get_Parameter("raggio");

f = 0.7 * (Math.Sin(0.1*Math.PI * 1)-Math.Exp(-0.3*1))* 1 +20;
g = 0.0328*f;
generico.Set(g);
}
```

Fig. 7. Example of the base script

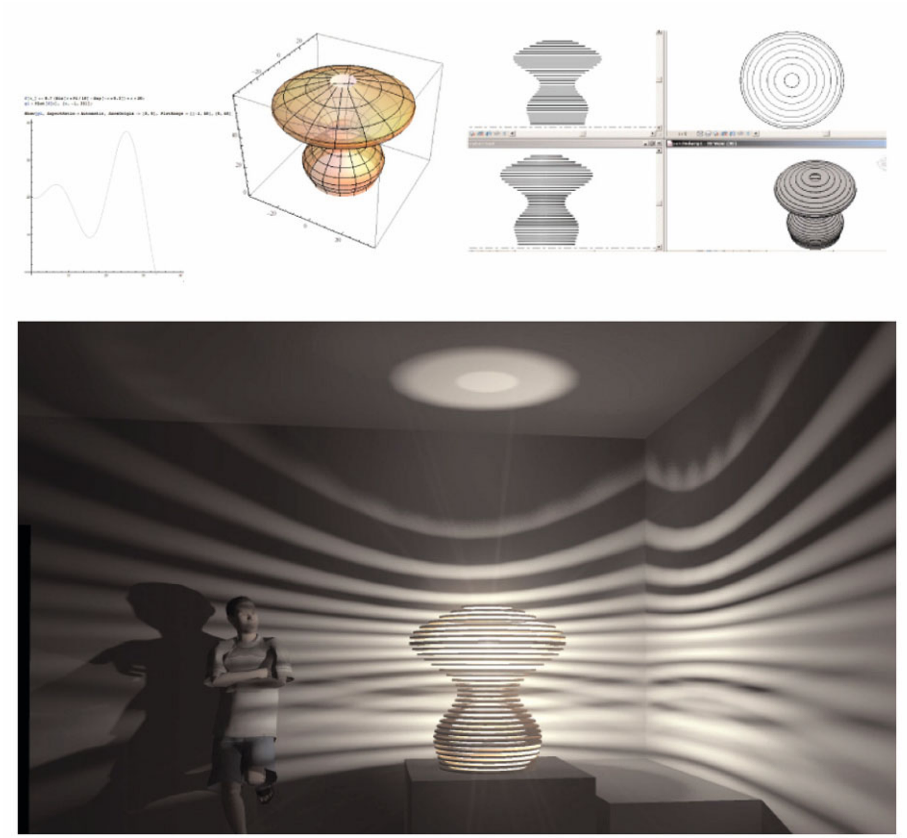


Fig. 8a-b-c. First run. a, above left) in Mathematica; b, above right) in Revit; c, below) the prototype

```
//procedure
int 1;
int n = 33;
double f,h;
for (1 = 1; 1 < n; 1 = 1 + 1)
{

                XYZ location =
Application.Create.NewXYZ(0.0328*Math.Cos(0.1*Math.PI*i), h),
0.0328*Math.Sin(0.1*Math.);
FamilyInstance unico = Create.NewFamilyInstance(location, family,
StructuralType.NonStructural);
Parameter generico = unico.get_Parameter("raggio");
f = 0.7 * (Math.Sin(0.1*Math.PI * 1)-Math.Exp(-0.3*1))* 1 +20;
g = 0.0328*f;
generico.Set(g);
}
```

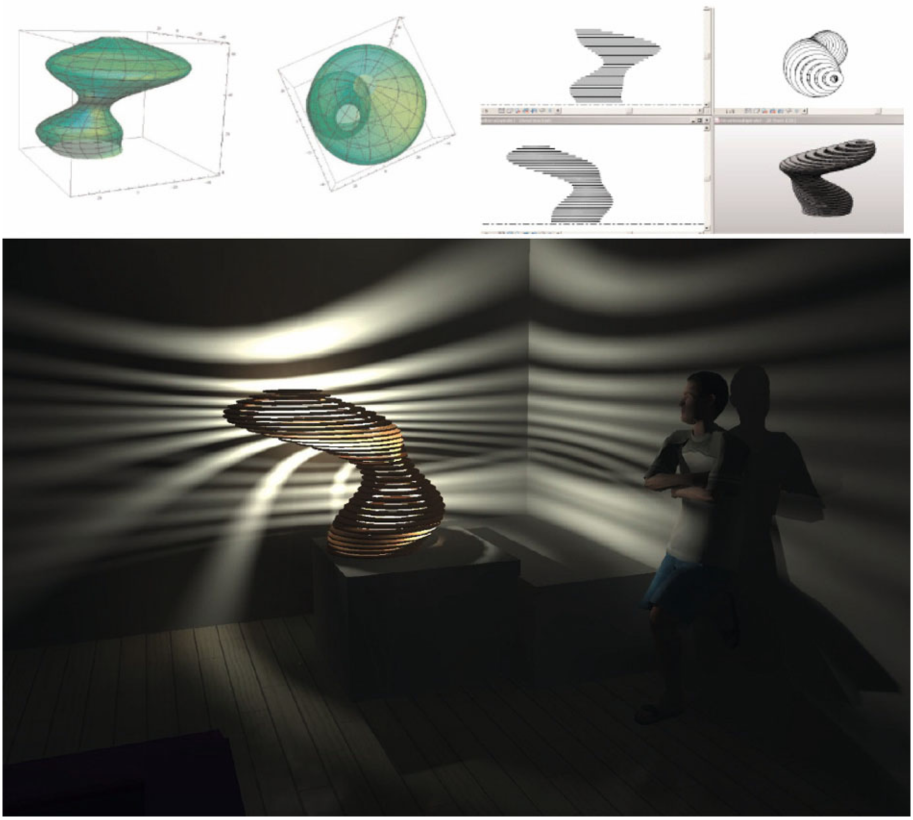
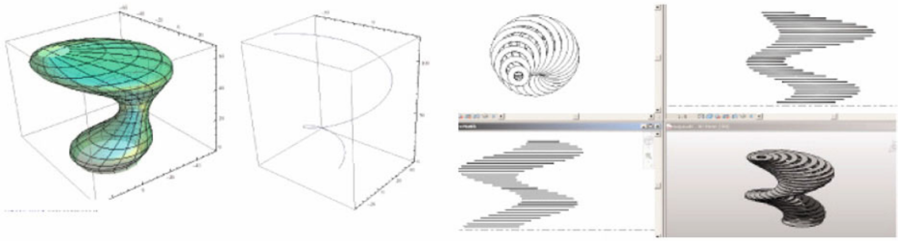


Fig. 8d-e-f. Second run. a, above left) in Mathematica; b, above right) in Revit; c, below) the prototype

```
//procedure
int 1;
int n = 33;
double f,g,h, S;
for (1 = 1; 1 < n; 1 = 1 + 1)
{
                    S = 0.1*i
          XYZ location =
Application.Create.NewXYZ(0.0328*Math.Cos(0.1*Math.PI*i),
0.0328*Math.Sin(0.1*Math.PI*i), S);
FamilyInstance unico = Create.NewFamilyInstance(location, family,
StructuralType.NonStructural);
Parameter generico = unico.get_Parameter("raggio");
f = 0.7 * (Math.Sin(0.1*Math.PI * 1)-Math.Exp(-0.3*1))* 1 +20;
g = 0.0328*f;
generico.Set(g);
    }
```



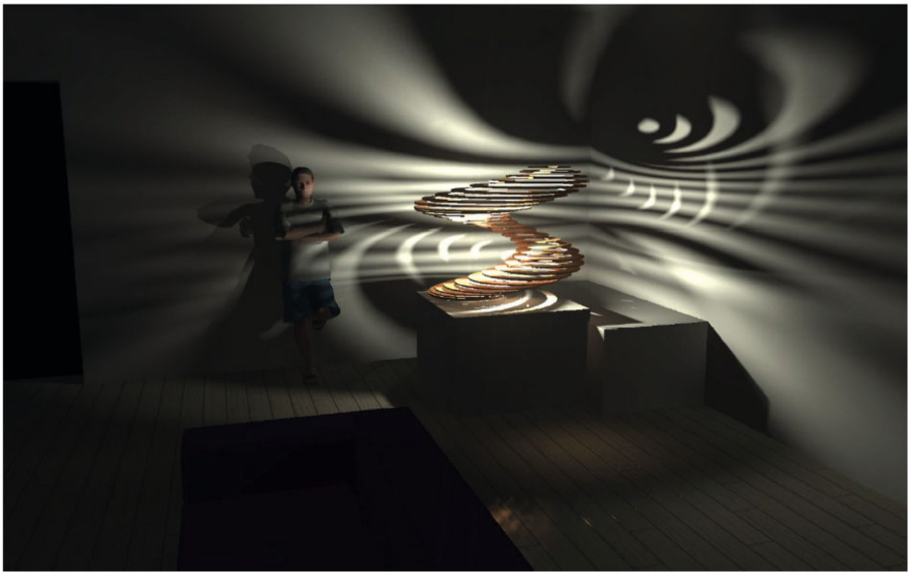
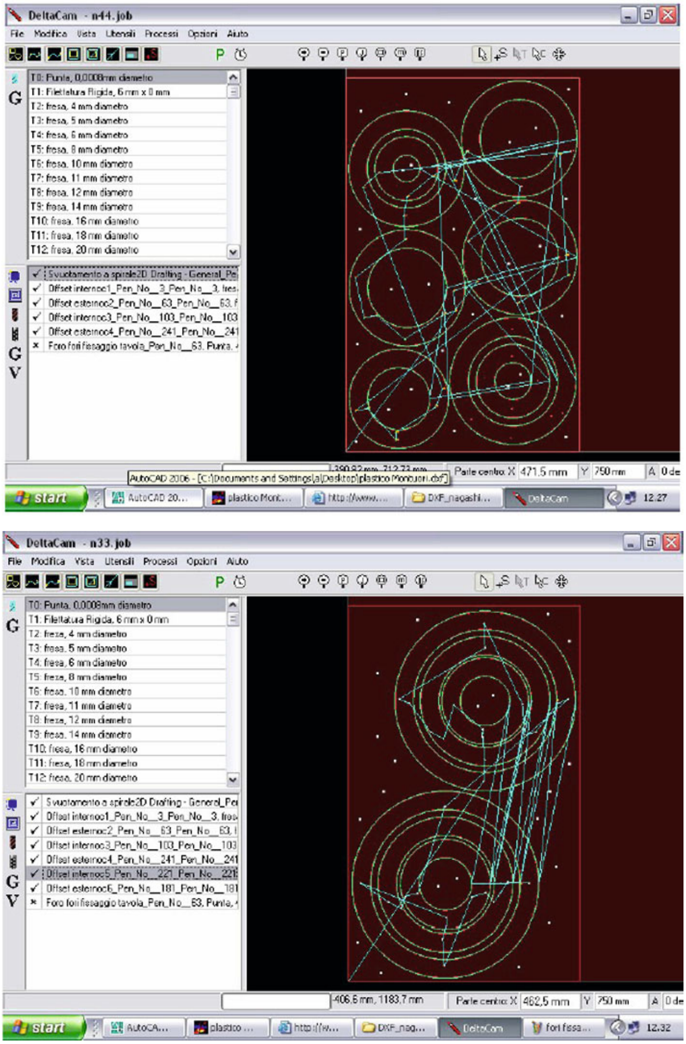


Fig. 8g-h-i. Third run. a, above left) in Mathematica; b,above right) in Revit; c, below) the prototype

The resulting light effect was not satisying, therefore successive runs were based on the attempt at adding dynamism to the light effect, by introducing a curved path for the location of center points of the circle, as previously mentioned. Challenges in these cases arose for construction and assembly, due to lack of vertical continuity for components along the total height of the lamp.



Figs. 9. The extraction of DXF data for fabrication, exported to DeltaCAM: toolpath generator software

The evaluation of the lighting effect introduced a new parameter for width and distance of the circles, which was optimized to a 1-2 cm dimension. The evaluation cycle included a bridge with the machining software, therefore a procedure was set up within Revit in order to extract from the assembly model the geometry of circles including information on joining. This was permitted by the rigor of mathematical definition that made it possibile to verify several design options, while always maintaining the rigor of the form definition, that gave an immediate link with fabrication. The design data of the single circles building the lamp were exported to the software for the Computer Numerically Controlled (CNC) machine as 2D information (dxf format). The single circles were nested into the wood boards and then milled out of them, including the holes prepared for the assembling phase (fig. 10).

Fig. 10. CNC milling of the circles

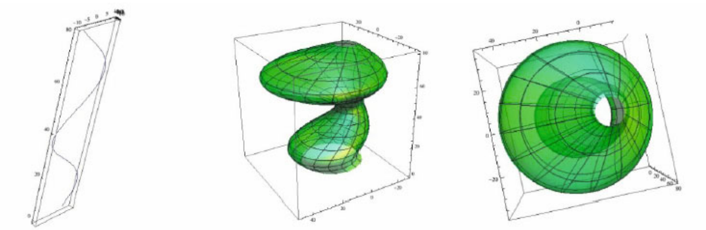


Fig. 11. The final design, called "Nagashima!" It represented the best solution according to lighting, constructive and architectural effects combined

The interesting part of the process, aside from its generation, is that every hypothesis could have been fabricated with the same ease of final design, since it was selected among alternatives of equal accuracy. So final design didn't need any further refinement after final choice. It was transferred to the CNC machine, and optimized for nesting into existing wood boards. A big part of the process, not explicitly mentioned in this paper, was of course a market search for all components to be used for the project. Market constraints became geometry constraints during the definition of the family. We explicitly looked for such an interplay between different factors.

The prototype was finally assembled as a "kit of parts", precisely defined during design. The idea was to have a compact package, according to guidelines followed by companies like Ikea. A great complexity in design methods is used to increase the ease of assembly.

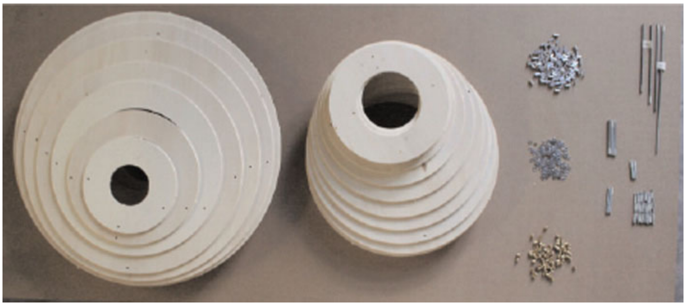


Fig. 12. The kit of parts for assembling the "Nagashima!" prototype

Fig. 13. The assembling procedure of the wooden circles with pre-cut holes

Conclusions and Challenges

Despite of a satisfying result in the built object, the process also showed some room for improvement. A first field of improvement is automation: the process still featured a few manual steps that could have been automated in a more robust and well engineered process from the computer science perspective, learning from this first one. The second field of improvement is of course construction: structural behavior was not fully satisfying, and assembly time was too high in case of just one person working on it. A third is lighting, which can feature a more advanced and integrated system, using for example LED strips with an hidden source for it embedded into the circles. Of course this will happen in case of industrial production, which we are currently evaluating with an interested company, since in fact there is a rigor behind the form generation that allows for its optimization and embedding of additional layers of complexity.

But this open process, featuring different platforms instead of just a single optimized one, also allowed for the inclusion in the process of several people, not all of whom were strictly specialized, which is a great achievement for a process such as the architectural one, where coordination is a main problem being faced constantly as building technology increases its variety and complexity. In this context, a new model for design, less centralized and more adaptive is urgently needed, but, as mentioned, the question is not merely technical: it involves a different structure for relationships. We believe this can happen in an integration between the two current fields of digital modelling and web networking, which needs to be fully researched. We believe the the "new desktop" of a contemporary designer will work with those two domains fully integrated, and working in a social domain, building up what we called a "Cloud Design", recalling the definition of cloud computing. The aim, however, is not to have a centralized system used to verify a design pre-conceived that needs optimization of a performance. The "Nagashima!"

example, despite its limits and small dimensions, shows that we need a different model in order to see the design come into being and grow in the cloud. We need different kinds of professionals who are capable of crossing boundaries between disciplines, but most importantly, are open to different mindsets, rather than remaining confined within a given field. The challenge that is at the time technical and human. Variation and adaptiveness need to become central in the design process, but they will be driven and expressed through software, which means they are deeply rooted in mathematical culture.

Fig. 14. The final "Nagashima!" lamp: light comes off the lamp as tested during the design process

Fig. 15. The final "Nagashima!" lamp: Global view showing how the lighting effect matches what tested during the design process

Fig. 16. Later studies for a new version embedding an inner surface, based on an helicoid

Acknowledgment

People who worked on the "Nagashima!" lamp project: Emanuele Trifiletti (student of architecture, Università degli Studi Roma Tre); mathematical modelling: Giampiero Dionisio (student of mathematics, Università degli Studi Roma Tre). "Blossoming Flower" lamp: Stefano Guarnieri (student of architecture, Università degli Studi Roma Tre); fabrication and construction definition: Massimiliano Pontani (Università degli Studi Roma Tre Fabrication Lab); scripting: Gian Marco Todesco (adjunct professor of scripting, School of Architeture, Università degli Studi Roma Tre).

Notes

1. The "Nagashima!" project is currently under development with an industrial partner, no commercial use whatsoever of what is presented in this paper is allowed.
2. The program started on this premise: "The program overcomes longstanding deficiencies in design education by creating a distinctive fusion of design culture and technology through the disciplines of computation, analysis, and advanced production methodologies". It is directed by its founder, John Nastasi, who is also teaching Advanced Manufacturing at Harvard University. To read more on the program, see [Converso 2010]; see also [Nastasi 2005].
3. The distinction between tool and instrument and the shift towards accuracy of measurements in science are described [Koyré 1957].
4. It is interesting to notice how even the same software package has the tendency to be "specialized" for the different fields: Revit Architecture, Revit Structure, Revit MEP.
5. In Catia, for example, the same relationship is possible between "part" and "assembly", with some differences between what it is possible to do in the two environments.
6. This kind of link illustrates a connection between research and teaching activity in software for design: a good example are the classes thaught at Zurich ETH University by Fabio Gramazio and Matthias Kohler, where students played with simple scripts in a "playpool" determined by professors. In that case, the procedure was the link between design and machining with an industrial robot. See [Bonwetsch et al. 2007] and [Bonwetsch et al. 2006].
7. On the work of SHoP architects, see [Converso 2008].

References

BONWETSCH, Tobias, D. KOBEL, Fabio GRAMAZIO and Matthias KOHLER. 2006. The informed Wall: applying additive digital fabrication techniques on architecture. Pp. 489-495 in *Synthetic Landscapes*. Proceedings of the 25th Annual Conference of the Association for Computer-Aided Design in Architecture.

BONWETSCH, Tobias, Fabio GRAMAZIO and Matthias KOHLER. 2007. Digitally Fabricating Non-Standardised Brick Walls. Pp. 191-196 in *ManuBuild*. Proceedings of the First International Conference, 25-26 April 2007, Rotterdam. M. Sharp, ed. London: CIRIA.

CACHE, B. 2003. Towards a Non-Standard Mode of Production. In *Architectures Non Standard*, Fréderic Migayrou and Mennan Zeynep, eds. Paris: Editions du Centre Pompidou.

CHASZAR, André, ed. 2006. *Blurring the lines*. Chichester, UK and Hoboken, NJ: John Wiley and Sons.

CONVERSO, Stefano. 2008. *SHoP Works, IT Revolution in Architecture*. Rome: Edilstampa.

———. 2010. The Form Beyond Software. In *Digital Design for Construction*. Milan: Maggioli Editore.

———. 2012. *SHoP Works, IT Revolution in Architecture*. Rome: Edistampa (Eng. ed. of [Converso 2008]).

FRAZER, John. 2006. The generation of virtual prototypes for performance optimization. Pp. 208-212 in *Game Set and Match 2. The Architecture Co-Laboratory*, Lukas Feireiss and Kas Oosterhuis, eds. Rotterdam: Episode Publishers.

KILIAN, Axel, Helmut POTTMANN, Andreas ASPERL, Michael HOFER and Daril BENTLEY, eds. 2008. *Architectural Geometry*. Bentley Press Institute.

KOYRÉ, A. 1957. *From Closed World to Infinite Universe* (Hideyo Noguchi Lecture). Baltimore: John Hopkins University Press.

LYNN, Greg. 2003. Variations calculées. P. 91 in *Architectures Non Standard*, Fréderic Migayrou and Mennan Zeynep, eds. Paris: Editions du Centre Pompidou.

———. 2004. Constellations in practice. *PRAXIS* 6: 8-17.

NASTASI, John. 2005. Retooling the Architecture Machine. *Museum Quarterly Journal*, Innovation of Digitally-Driven Architecture. XXIII (3)

PAOLETTI, Ingrid. 2006. *Costruire le forme complesse. Innovazione, industrializzazione e trasferimento per il progetto di architettura (Building Complex Shapes)*. "Tecnologia e progetto" Series. Milan: Editrice Clup.

About the author

Stefano Converso, and architect and design professional, focuses on the relationship between advanced digital technologies and design culture. Since February 2010 he has been a Research Fellow at DIPSA Department of Università degli Studi Roma Tre, where he researches software customization in contemporary design, analysis and manufacturing practice. Since 2008 he has been a professor of parametric design at the School of Architecture at Università Roma Tre. He received a Ph.D. in the European Program 'Villard d'Honnecourt', coordinated by IUAV Institute for Architecture, in Venice, Italy, for a research program on 'digital spreading'': the diffusion of advanced design technology on a large scale. He has participated in international events and conferences and has collaborated with Institutions such as Product Architecture Lab (Stevens Institute of Technology), Politecnico di Milano, TU Delft, EAAE-ENHSA Architectural Design Teachers' Network. He was project manager for the Italian team that developed a sustainable, highly efficient housing prototype and took part in the international competition Solar Decathlon Europe 2012, aimed at developing twenty protoypes of innovative houses powered solely by solar energy. He will be involved again in the July 2014 edition of the same competition. He is the author of books and articles in journals and magazines such as *De Architect*, *Modulo*, *Arch.IT* and *Esempi di Architettura*. Since 2003, he has been a member of the editorial committee of the magazine *Il Giornale dell'Architettura*, where he is the coordinator of the Digital Science for Design section. He is author of the entry on 'digital architecture' in the latest edition of the *Dizionario dell'Architettura*, published by Einaudi.