INTEGRATING PARAMETRIC MODELING WITH BIM THROUGH GENERATIVE PROGRAMMING FOR THE PRODUCTION OF NURBS SURFACES AND STRUCTURES

LILIAN SILVA¹, NEANDER SILVA² and IGOR LACROIX³
¹²Universidade de Brasilia, Faculdade de Arquitetura e Urbanismo, Brasilia, Brazil
¹²{lilianmfs91|neander.furtado}@gmail.com
³Centro Universitário de Brasilia, Faculdade de Tecnologia e Ciências Sociais Aplicados, Brasilia, Brazil
³igorlacroix@gmail.com

Abstract. A workflow for integration of parametric modeling with BIM, using generative-programming, is described and tested in this research. The objective is to take advantage of these two distinctive design paradigms. This paper describes a design experiment that required a NURBS roof generated by sweeping profiles along a curved path. We assumed the use of multiple applications, using various file formats, are facts and are unlikely to disappear. Given that interoperability issues will certainly arise, we propose and test a design workflow using parametric modeling, generative programming, and building information modeling. Our major contribution was defining a workflow for designing NURBS surfaces and corresponding supporting structures enhancing interoperability among different applications through generative-programming.

Keywords. NURBS; Parametric; Programming; Interoperability; BIM.

1. Introduction

The process of designing and building curvilinear architectures is still challenging. We assumed that the use of multiple applications, based on distinctive design paradigms, are facts and are unlikely to disappear. We recognize the relevance of integration into standalone applications, but argue that the diverse nature of society and software development makes undesirable a unified platform. Therefore, our research focus on augmenting and optimizing interoperability.

Given that interoperability will give rise to a variety of communication and data transfer issues, we propose and test a design workflow based on the use of parametric modeling, generative programming, and building information modeling, BIM (Eastman et al, 2008). Although there have been numerous curvilinear architectures in the last decades (Krauel, 2010), these projects are still very challenging to design and build (Kolarevic, 2003, p. 6-7).
This paper describes a design experiment that required the development of a curvilinear roof based on Non-Uniform Rational B-Spline, NURBS, generated by sweeping curved profiles along a path also curved. Computer applications are based on a diverse range of design paradigms. Some authors have proposed at least two major categories of paradigms driving the development of applications in the field of architecture. The first one are applications oriented to the processes of generating forms that focus on operations such as extrusion, sweeping, lofting, subdivision, etc. The second category are construction function-oriented applications to model walls, windows, doors, columns, beams, slabs, etc. These applications also focus on the on building information (Mark et al, 2008).

We assumed that both paradigms are essential for the proper conception, development and construction of the forms described earlier. We believe the simultaneous use of parametric modeling, generative programming (Terzidis, 2009, p.19-22) and building information modeling (Eastman et al, 2008) may improve the design of these complex forms.

It is important to stress that the need for BIM applications does not arise only for producing construction drawings. One may argue that those may be produced using only non-BIM applications. However, it is much more time consuming and labor intensive to produce such representations in these applications.

The main point in using BIM is that the other applications do not recognize construction components such as walls, columns, beams, slabs, etc. A true design process must allow not only the production of construction drawings but also several other type of building information.

Our starting point was based initially on three applications already available in the market: FormZ (Autodesk), Rhinoceros-Grasshopper (Robert McNeel & Associates) and ArchiCAD (Graphisoft). Due to certain advantages of FormZ, regarding intuitiveness in direct graphical modeling, we adopted this application in the initial stages of design development.

Central to our research was an add-on to ArchiCAD developed by Graphisoft. A live connection was created between Rhinoceros-Grasshopper and ArchiCAD.

In our research the connection between FormZ and Rhinoceros is obviously not live. We do not consider this to be an ideal situation. However, we do consider this issue to be outside the scope of our present research. In the other hand, the connection Rhinoceros-Grasshopper-ArchiCAD is really bi-directional. The use IFC files was also left outside the scope our present research.

1.1. RESEARCH QUESTION AND HYPOTHESIS

In this scenario, what computational resources can be used and developed to generate these forms and related information while integrating applications based in those diverse paradigms? A workflow involving FormZ-Rhinoceros-Grasshopper-ArchiCAD interoperability was proposed allowing flexibility, while as well as specification. The completion of the process in BIM also made possible the extraction of additional information not available in applications like FormZ and Rhinoceros. Figure 1 illustrates the proposed workflow.
The interoperability between FormZ and Rhinoceros was achieved through a mutually recognizable. The interoperability in Rhinoceros-Grasshopper-Archicad connection helped in transforming abstract geometry into recognized BIM elements.

1.2. RESEARCH REVIEW

The articles we surveyed related to the purpose of our research, deal with different issues or have different objectives.

Ashour et al (ASHOUR and KOLAREVIC, 2015) test multiple solutions within a design workflow with the limitations of the softwares used in the process for exploring the best performance and quality of result. The idea is similar, but the softwares used are different and do not include BIM.

Plotnikov (PLOTNIKOV, 2016) aims to bridge the gap between the physical and digital design methodologies. It integrates Grasshopper with GIS, instead of a BIM system.

Khalili-Araghi et al (KHALILI-ARAGHI and KOLAREVIC, 2016) used a plug-in in the BIM software Autodesk Revit System. We point out that the only similarity here is a framework or workflow using a BIM system. However, their goal is a standalone platform rather than interoperability.

Sharah used programming in Rhino-Grasshopper using UV Mesh and Quadrangulation (SHARAH, 2017). However, its objective is mainly visualizing the modeled object in a virtual reality headset.

Therefore, these authors have not experimented with a workflow similar to the one we proposed above enabling the use of an external file imported into Rhinoceros and connected to Archicad via Grasshopper.

2. Methodology

We designed a surface for testing our hypothesis. The project was initially modeled in FormZ because of its easy to use graphical interface without resorting to programming. Initially, a NURBS surface was created to form the roof and
Firstly, a NURBS surface was created from six contours inspired on the arches of a bridge by using NURBS lofting tool. In the same way, the cladding was modeled with a 30cm offset below the roof. The axes contours of the modeled structural beams were implemented by using the contour tool to create an egg-crate type structure. Figures 2 illustrate part of this process.

Figure 2. The process of creating an Egg-Crate structure form NURBS Roof Surface in FormZ.

The project file was then exported through a file with the extension STEP for interoperability between FormZ and Rhinoceros. This format was chosen because it maintained the object’s original geometry properties. For example, what was a real curved surface in FormZ remained as a real curved surface when the file was imported and opened in Rhinoceros.

The main reason for choosing to define and model the curved surface earlier in FormZ was that if it was done only in Rhinoceros graphic window it would have taken more steps and time to create the surface. As an example, it took approximately four steps in FormZ to model it, while in Rhino it would have taken seven steps. Even if we considered exporting and importing from FormZ into Rhinoceros as two additional steps this would still represent six steps against seven. A key issue here would be to define precisely what is in a step. Due to the lack of space we will not go into much detail here. The fact is that a step may be considered from single actions, such as clicks and menu selections, to even a group of actions within an operation, such as the creation of a sweep along path.
In this research was adopted the second option, that is, the number of operations. Since the operations of exporting and importing are considerably much shorter than the graphical ones for creating curved lines and surfaces, we still have a strong case for adopting FormZ in the early stages of design. Again, we would like to stress that the use of multiple applications in the same design process is a very common situation in practice that has to be dealt with as a research problem. It is undeniable that designers have to and do resort to different applications taking into consideration the best tool for each design task at hand.

Secondly, the roof, cladding and structural axes imported into Rhinoceros from FormZ had to be further developed. We used the Grasshopper programming and the Rhino-Grasshopper-ArchiCAD Live Connection from Graphisoft for this purpose. This connection recognizes the Grasshopper generated objects directly inside ArchiCAD as real BIM elements without having to maneuver manually, reworking, or redoing objects or components. One example would be the steel beams adopted in this design experiment, which had an I profile shape, with the dimensions of 35 cm height x 21 cm width. Figure 3 illustrates one of these processes.

![Figure 3. NURBS Roof Surface Structure Beams in Rhino-Grasshopper to ArchiCAD.](image)

Thirdly, algorithms were elaborated in Grasshopper for generating steel beams from the imported axes. It was necessary to elaborate instructions to create the precise structural components by sweep the I profile with the aforementioned dimensions through the axes imported from FormZ. Figure 4 provides a general overview of the programming produced in Grasshopper for the purpose of generating the structural beams.
In the Rhino-Grasshopper interface, you have node 1 as shown in Figure 4 that represents structural contours/curves used as a path to generate the beams along this path in node 2. In node 2 a procedure creates a beam in ArchiCAD defined by curve from the selected structural contours/curves contained in node 1. Node 3 allows the user to define beams settings. This can be a rectangular or complex beam. In this case, was used a complex beam which the profile has the shape of an I beam. Node 4 is a subcategory of node 3 and allows the type of profile beam to be selected. As mentioned an I beam was selected. Node 5 allows the structural function of the beams to be specified. In this case, 0 is the value for load-bearing and 1 for non-load-bearing. In this research, a load-bearing structure was specified. Node 6 allows the user to define the position of the beam height structure in relation to its curve of reference stored in node 1. This height is defined by the value 0 for exterior (above) or 1 for interior (below). In this case, was specified 0 for exterior.

Fourthly, a paneling algorithm was applied to the imported surfaces to generate panels of approximately two by two meters with Grasshopper plug-in LunchBox and Rhino-Grasshopper-ArchiCAD Live Connection. The paneling is necessary for constructability reasons. In order to make feasible and more affordable the construction, it was necessary to subdivide the curved surface into subfractions of four side coplanar panels. This would make it possible to produce the panels by CNC 2D cutting metal sheets. The opaque areas of the roof were modeled with zinc-metallic sheets. The transparent areas were modeled with glass-reinforced sheets. Figures 5 and 6 illustrate in more detailed the programming involved in these processes.
In Rhinoceros-Grasshopper interface by default, you have the nodes/boxes that represent the surface (1 in the diagram shown in Figure 6) and the procedures (2, 3 and 4). The node 2 places points on the surface. The node 3 connects the points with lines forming polygons. The node 4 separate the polygons dividing it into a mesh UV mapping. This separates the surface, in this case, into smaller surfaces. Once installed the Rhino-Grasshopper-ArchiCAD Plug-in/Connection, in the Grasshopper interface appears a new set of nodes/boxes options within the tab: ARCHICAD.

These new nodes from Rhino-Grasshopper-ArchiCAD Plug-in/Connection, receive the information from the Mesh UV mapping algorithm of Grasshopper through node 6 in the diagram shown in Figure 6. The data received by node/box 6 in the form of a Mesh is passed on to node 7. This node allows the configuration of the Mesh settings by the user. Then it translates a geometry only object into a Mesh surface with BIM Components properties. In particular case of this research, this included the material properties of Zinc Roof panels. The nodes 8, 9 and 10 are subcategories of Mesh settings. Node 8 allows the building material to be specified. Node 9 allows the specification of in which layer the Mesh will be stored. Node 10 allows the specification of a load-bearing structure or non-load-bearing structure.

If the algorithm in this interface of Rhino-Grasshopper is executable, its nodes/boxes appear in GRAY. If there is a problem/error in executing the algorithm its nodes/boxes appear in RED. However, this will not be the case when you have already installed the Rhino-Grasshopper-ArchiCAD Plug-in/Connection.
If the Rhino-Grasshopper Algorithm is not connected to the nodes/boxes of the algorithm ArchiCAD receives, they will appear in red and orange because they did not receive the information. This will also be the case if the connection is not ON in ArchiCAD and Grasshopper simultaneously. A similar algorithm was used for generating the supporting structure of the glass cladding surface, as shown in Figure 6.

![Image of a diagram showing the algorithm process for generating the supporting structure of the glass surface in Rhino-Grasshopper to ArchiCAD.]

Figure 6. Algorithms for generating the supporting structure of glass surface in Rhino-Grasshopper to ArchiCAD.

Node 1 stores the glass surface selected by the user. Node 2 generates an offset surface. The subcategory node 3 defines the surface offset Z axis direction. The subcategory node 4 defines the distance it offsets from the original surface. Node 5 generates a grid structure of the offset surface. Nodes 6 and 7 are subcategories of node 5, these specify the U and V grid division respectively. The supporting structure for the glass surface was composed of tubular parts. Therefore, node 8 defines this circle profile. The subcategory Node 9 specifies the width of the profile’s radius. Node 10 extrudes the circular profile along structural grid lines. Node 11 generates the tubular Mesh surface based on the extrusion produced in Node 10. Node 12 receives Mesh information from Node 11 and sends it to ArchiCAD. Node 12 receives geometry information from Node 11 and send it to Node 13. Node 13, called Mesh settings, contains three subcategories that allows the user to specify some properties: building material (Node 14), layer for information to be stored (Node 15) and structural function (Node 16).
3. Results and Observations

The beams were generated in ArchiCAD directly as BIM elements from the profile axis imported into Rhino from FormZ. The total time spent to generate the elements, between finishing programming in Grasshopper and seeing the results in ArchiCAD was 12 minutes and 48 seconds. The proposed workflow, that is, FormZ-Rhinoceros-Grasshopper-ArchiCAD connection, made it possible the desired flexibility at the beginning of the design process, as well as precision and speed in the detailed specification of the final designed building. It would not be viable to model these curvilinear beams, roof, and cladding by hand, obviously. The number of variables and the amount of data involves making it unnecessary to argue for that. The initial direct modeling stage developed in FormZ was necessary due to the dependency, at the conceptual design stage, of the designer decisions about the shapes of the curved sources to generate the surface as well the profile axis. The use of a programming interface such as Grasshopper was essential to allow the creation of detailed structural beams and cladding panels. The use of Rhino-Grasshopper-ArchiCAD connection was essential for translation the elements generated in Rhinoceros as BIM recognized components in ArchiCAD. The completion of the process in a BIM system also made possible the precise extraction of construction and fabrication drawings, including 2D cutting plans.

4. Conclusions

To the best of our knowledge, our major contribution resides in defining a workflow for designing NURBS surfaces and corresponding supporting structures by enhancing the interoperability among different applications and by using generative programming.

The use of FormZ into the existing workflow for more flexible and creative geometry during the conceptual design process helped to create more fluid forms like NURBS surface within an easier interface. It also allowed the easier extraction of structure axis through the contour tool than in would be done inside Rhinoceros alone with less steps. Its interoperability with Rhinoceros through STEP file format retains its original geometry properties, particularly, when dealing with real curves and curved surfaces. Once imported into Rhinoceros its geometry was recognized as surface which is easier to work with than with the mesh type geometry. If we had imported via 3DMax format (.3ds), the surface would have been reduced to a mesh, when implemented with Grasshopper and the connection with ArchiCAD. The interoperability provided by the Rhinoceros-Grasshopper-ArchiCAD connection helped in solving the issues with geometry not recognized as BIM elements such as generating the structure axis as a path for an I beam profile structure. It also helps in the decision process by testing a variety of solutions that would best fit the project.

As future research, the algorithms developed in this research could be further improved for the specific reuse of them in other situations of creating structural curved I beams and to automate paneling of curved surfaces. These could result in specific plugins within Grasshopper. Future research could also involve new algorithms to improve interoperability with Rhino-Grasshopper starting with a file
from FormZ in order to increase automation in the process of importing from one application to another.

References