



**SBTIC  
2019**  
VIRTUALIZAÇÃO INTELIGENTE  
NO PROJETO E NA CONSTRUÇÃO  
2º Simpósio Brasileiro de Tecnologia  
da Informação e Comunicação na  
Construção  
UNICAMP | 19 a 21 de agosto

# MODELAGEM PARAMÉTRICA, PROGRAMAÇÃO ALGORÍTMICA E BIM: ESTUDO PARA GERAÇÃO DE VEDAÇÃO E ESTRUTURAS BASEADAS EM NURBS

## Parametric Modeling, Algorithmic Programming and BIM: A case study for the Generation of NURBS Cladding and Structures

**Lilian Maciel Furtado Silva**

Universidade de Brasília | Brasília, DF | lilianmfs91@gmail.com

**Neander Furtado Silva**

Universidade de Brasília | Brasília, DF | neander.furtado@gmail.com

**Igor Lacroix**

Centro Universitário de Brasília | Brasília, DF | igorlacroix@gmail.com

### RESUMO

Um fluxo de trabalho para integração de modelagem paramétrica com BIM, utilizando programação generativa, é descrito e testado nesta pesquisa. O objetivo é aproveitar esses dois paradigmas distintos de projeto. Este artigo descreve um experimento de projeto que exigiu uma cobertura NURBS gerada por perfis de varredura ao longo de um caminho curvo. Assumimos que o uso de vários aplicativos, usando vários formatos de arquivo, são fatos e provavelmente não desaparecerão. Uma vez que surgirão problemas de interoperabilidade, propomos e testamos um fluxo de trabalho de projeto usando modelagem paramétrica, programação gerativa e modelagem de informações de construção. Nossa principal contribuição foi a definição de um fluxo de trabalho para projetar superfícies NURBS e estruturas de suporte correspondentes, aprimorando a interoperabilidade entre diferentes aplicativos por meio de programação generativa.

**Palavras-chave:** NURBS; Parametric; Programação; Interoperabilidade; BIM.

### 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. The use of multiple applications with distinctive design paradigms are unlikely to disappear. We recognize the relevance of integration into standalone applications, but the diverse nature of society and software development makes undesirable a unified platform. Therefore, our research focus on augmenting interoperability.

Since interoperability will result in a variety of communication and data transfer issues, a design workflow based on the use of parametric modeling, generative programming, and building information modeling, BIM (Eastman et al, 2008) was proposed. Although there have been numerous curvilinear architectures in the last years (Krauel, 2010), these are still challenging to design and build (Kolarevic, 2003, p. 6-7).

We describe a design experiment requiring the development of a curvilinear roof based on Non-Uniform Rational B-Spline, NURBS, generated by sweeping curved profiles along a path also curved.

Some authors have proposed at least two categories of paradigms in developing applications in architecture. The first one contains applications oriented to the processes of generating forms through operations such as extrusion, sweeping, lofting, subdivision, etc. The second category are function-oriented applications to model walls, windows, doors, columns, beams, slabs, etc. (Mark et al, 2008).

Both paradigms are essential for designing and building 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 those complex forms.

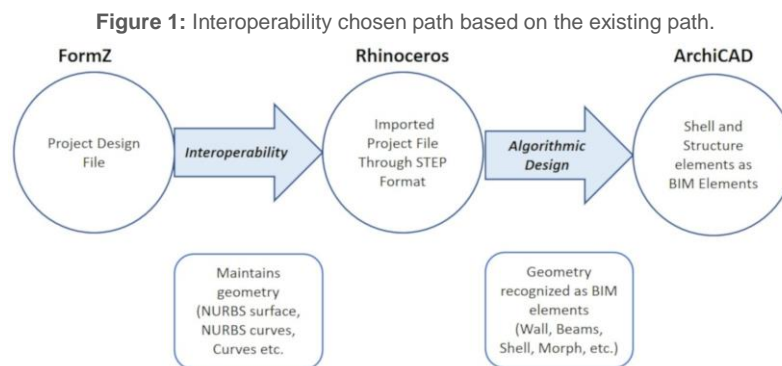
The importance of using BIM is that the other applications do not recognize construction components. A true design process must provide not only geometric descriptions, but also several other type of building information.

Our starting point were three commercial applications: FormZ (Autodesys), Rhinoceros-Grasshopper (Robert McNeel & Associates) and ArchiCAD (Graphisoft).

Central to our research was the Grasshopper-ArchiCAD Live Connection. The connection between FormZ and Rhinoceros is not live, but we considered this issue to be outside the scope of our research. However, the connection Rhinoceros-Grasshopper-ArchiCAD is bi-directional.

## 1.1 Research Question and Hypothesis

What computational resources can be developed to generate these forms and information while integrating applications based in those diverse paradigms? A workflow involving FormZ-Rhinoceros-Grasshopper-ArchiCAD interoperability was proposed (Figure 1).



The interoperability between FormZ and Rhinoceros was achieved through a mutually recognizable format. The interoperability in Rhinoceros-Grasshopper-ArchiCAD connection helped in transforming abstract geometry into recognized BIM elements.

## 1.2 Research Review

Related researches deal with different issues and objectives. Ashour et al (ASHOUR and KOLAREVIC, 2015) test multiple solutions within a design workflow, but the softwares used are different and do not include BIM.

Plotnikov (2016) integrates Grasshopper with GIS, instead of a BIM system. Khalili-Araghi et al (KHALILI-ARAGHI and KOLAREVIC, 2016) used a plug-in within Revit. However, their goal is a standalone platform rather than interoperability.

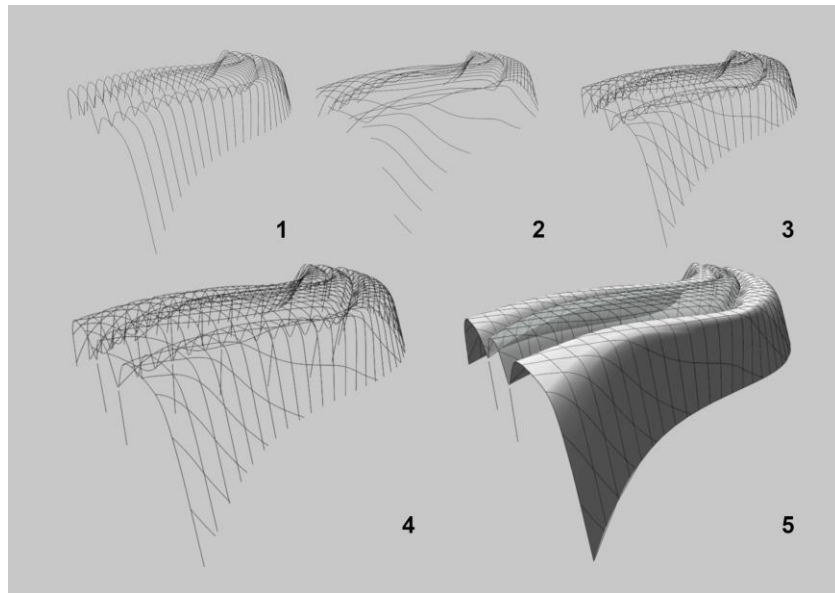
Sharah used programming in Rhino-Grasshopper using UV Mesh and Quadrangulation (SHARAH, 2017). However, the objective is visualizing the modeled object in a virtual reality headset.

## 2 METHODOLOGY

We designed a surface for testing our hypothesis. It was initially modeled in FormZ because of its easy to use graphical interface without resorting to programming. Due to certain shortcomings and lack of intuitiveness of Rhinoceros, particularly regarding direct graphical modeling, we adopted in the early stages of design development the application FormZ. A NURBS surface was created to form the roof and cladding.

Firstly, A surface was created from six contours with a NURBS lofting tool. The cladding was modeled with a 30 cm offset below the roof. The axes contours of the modeled structural beams were implemented by using the contour tool to create an egg-crate structure. Figure 2 illustrate part of this process.

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



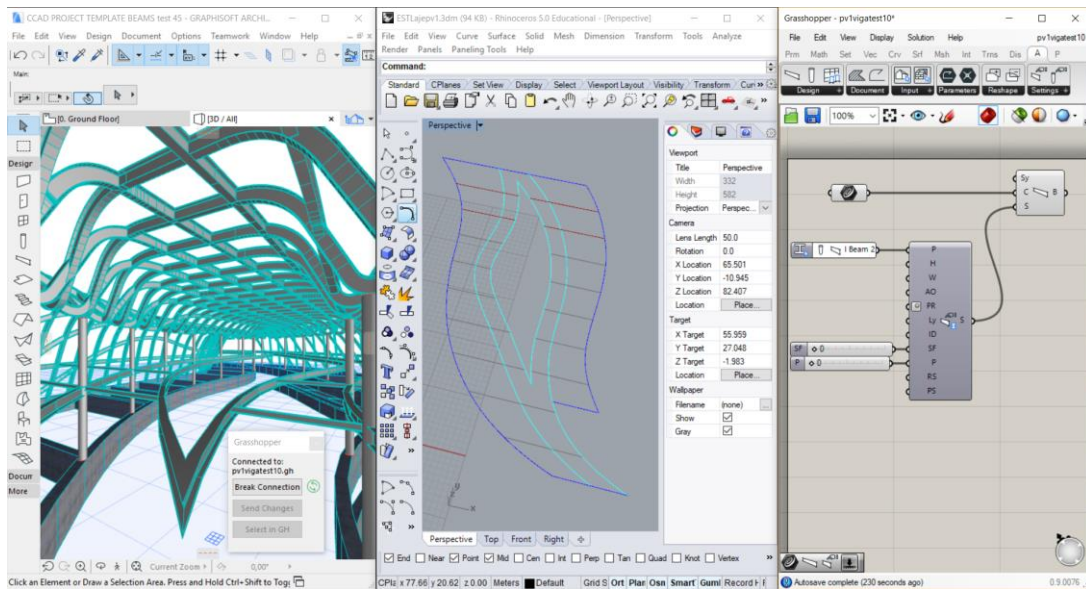
The project file was then exported to Rhinoceros through a STEP file. This format maintains the object's original geometry properties.

The main reason for choosing FormZ to define the surface was that 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.

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. Designers have to and do resort to different applications taking into consideration the best tool for each design task at hand.

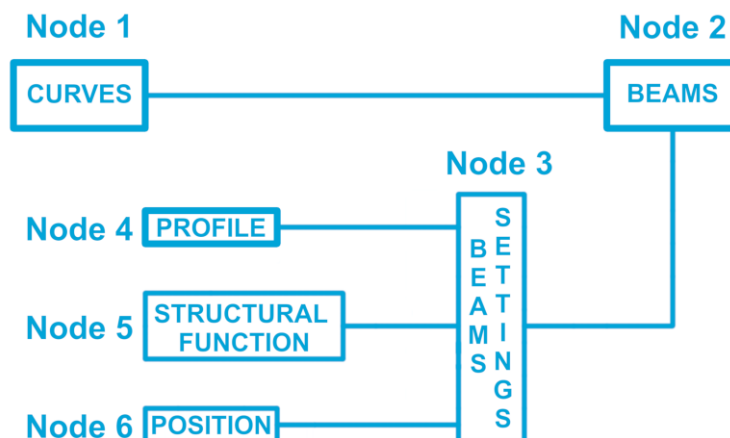
Secondly, the roof, cladding and structural axes imported from FormZ had to be further developed. The Grasshopper programming and the Rhino-Grasshopper-ArchiCAD Live Connection were used for this purpose. This connection recognizes the Grasshopper generated objects as real BIM elements in ArchiCAD. One example is 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.



Thirdly, algorithms were elaborated in Grasshopper for generating steel beams from the imported axes. Structural components were created by sweeping the I profile through the axes imported from FormZ. Figure 4 provides a general overview of the programming produced in Grasshopper for generating the structural beams.

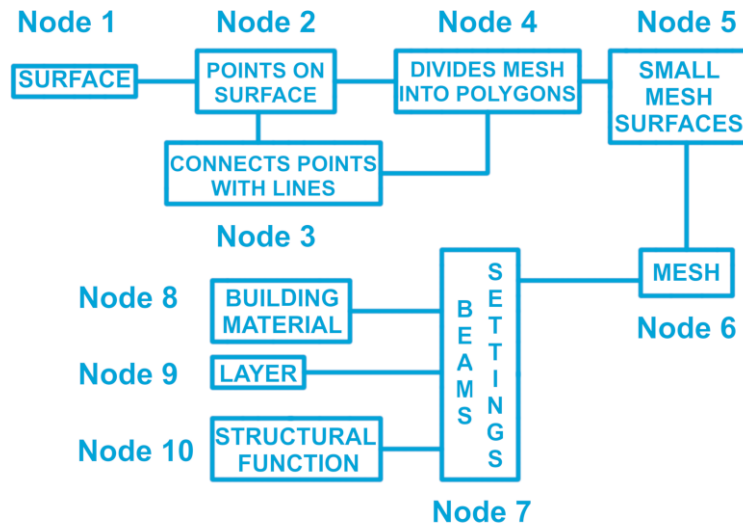
Figure 4: Algorithm for Structure in Rhino-Grasshopper to ArchiCAD.



Node 1, shown in Figure 4, represents structural contours/curves used as a path to generate the beams along them in node 2. A procedure in node 2 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. It was used a complex beam with the shape of an I. Node 4 is a subcategory of node 3 and allows the type of profile beam to be selected. Node 5 allows the structural function of the beams to be specified. 0 is the value for load-bearing and 1 for non-load-bearing. 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, an 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. In order to make feasible the construction, the curved surface was divided into subfractions of four-sided coplanar panels. Figures 5 and 6 illustrate in more detailed the programming involved in these processes.

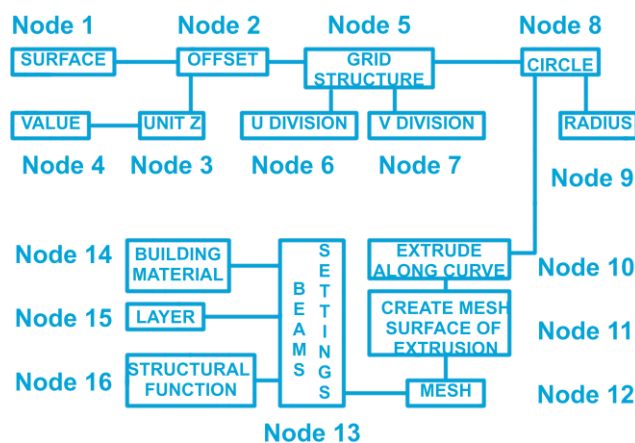
Figure 5: Algorithms for Paneling NURBS Roof Surface in Rhino-Grasshopper to ArchiCAD.



In Rhinoceros-Grasshopper interface the nodes/boxes represent the surface (1 in the diagram shown in Figure 6) and the procedures (2, 3 and 4). Node 2 places points on the surface. Node 3 connects the points with lines forming polygons. Node 4 separate the polygons dividing it into a mesh UV mapping. This separates the surface into smaller ones. Once installed the Rhino-Grasshopper-ArchiCAD Plugin/Connection, a new set of nodes/boxes options within the tab ARCHICAD appears.

These new nodes 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 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. A similar algorithm was used for generating the supporting structure of the glass cladding surface, as shown in Figure 6.

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. The use of a programming interface such as Grasshopper was essential to allow the creation of detailed structural beams and cladding panels and 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 CONCLUSION

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

The use of FormZ into the existing workflow during the conceptual design process helped to create more fluid forms within an easier interface. The interoperability provided by the Rhinoceros-Grasshopper-ArchiCAD connection helped in solving the issues with geometry not recognized as BIM elements such as the axis as a path for an I beam profile structure.

As future research, the algorithms developed in here could be further improved for their specific reuse in other situations for creating structural curved I beams and automating 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 on application to another.

### REFERÊNCIAS

- ASHOUR, Y. and KOLAREVIC, B. Heuristic Optimization in Design, Proceedings of ACAADIA 2015.
- EASTMAN, C., TEICHOLZ, P., SACKS, R. and LISTON, K. **BIM Handbook: A guide to Building Information Modeling for owners, managers, designers, engineers, and contractors**. John Wiley & Sons, Hoboken, New Jersey, 2008.
- KHALILI-ARAGHI, S. and KOLAREVIC, B. Captivity or Flexibility: Complexities in a Dimensional Customization System., Proceedings of ECAADE 2016.
- KOLAREVIC B. **Architecture in the Digital Age – Design and Manufacturing**. Taylor & Francis, New York, 2003.
- KRAUEL J., GEORGE W. and NODEN J. **Contemporary Digital Architecture – Design & Techniques**, Links Books, Barcelona, 2010.
- MARK, E. and GOLDSCHMIDT, G. A Perspective on Computer Aided Design after Four Decades, Proceedings of ECAADE 2008, Antwerpen.
- PLOTNIKOV, B., SCHUBERT, G. and PETZOLD, F. Tangible Grasshopper: A method to combine physical models with generative, parametric tools, Proceedings of ECCADE 2016.
- SHARAH, L., ESCALANTE, E., FABBRI, A., GUILLOT, R. and HAEUSLER, M. H. Streamlining the Modelling to Virtual Reality Process, Proceedings of CAADRIA 2017.
- TERZIDIS, K. **Algorithmic Architecture**. Elsevier/Architectural Press, Amsterdam, 2009.