



Title: **Software to Solve Structural Engineering Issues**

Authors: Alfonso Oliva, Director, Leslie E. Robertson Associates
Richard Garlock, Partner, Leslie E. Robertson Associates
Nidhi Sekhar, Computational Designer, Leslie E. Robertson Associates
Mia Tsiamis, BIM Specialist, Leslie E. Robertson Associates

Subject: IT/Computer Science/Software

Keywords: BIM
Design Process
Integrated Design
Parametric Design
Structural Engineering
Virtual Reality

Publication Date: 2017

Original Publication: CTBUH Journal 2017 Issue II

Paper Type:

1. Book chapter/Part chapter
2. **Journal paper**
3. Conference proceeding
4. Unpublished conference paper
5. Magazine article
6. Unpublished

Closing Gaps in Commercial Software To Solve Structural Engineering Issues



Alfonso Oliva



Richard B. Garlock



Nidhi Sekhar



Mia Tsiamis

Abstract

The ability of architectural graphical software to generate infinitely complex geometrical solutions for building design calls for an equally concerted effort to provide rational structural solutions. This paper presents the example of the design of a tower located in a dense urban environment to show how the evolution of modeling technologies is changing best practices in high-rise design. By engaging in the demonstrated workflow, designers and engineers can better respond to the needs of their clients and project sites, using research and development geared towards bridging the gap between incompatible software platforms and shortening the time between architectural vision and structural reality.

Keywords: BIM, Design Process, Integrated Design, Parametric Design, Structural Engineering, Virtual Reality

Introduction

As technology advances in both design and construction, so has our ability to build more monumental and unconventional structures. The realization of such structures necessitates a more modern, technological approach to the design and coordination process.

In the following pages, we will present a case study of the design of a 130-story tower in Southeast Asia, heretofore referred to as “the Tower,” whose main structure is composed of steel-framed floors supported by a central concrete core and composite perimeter columns (eight megacolumns and eight secondary columns). The building’s façade tapers at various degrees depending on the plan location and elevation, and the columns, in turn, slope in parallel (see Figure 1)

An algorithm-driven design approach, paired with a custom-developed interoperability platform, allowed for the rapid adjustment of complex geometries in the structural models and gave the architectural team the latitude to macro- and micro-adjust the geometry as needed.



Although the details of the design are not the focus of this paper, three design components are listed below to introduce the reader to the concepts of algorithmic-driven design and the necessity for and utility of an interoperability platform.

A parametric design is a design directed by an algorithm. The main advantage of this approach is that the user can build a dynamic 3D model, as opposed to the typical static 3D model. A dynamic model is capable of reacting to certain inputs in real time. A 3D model created through the algorithmic approach is controlled by input parameters that allow for real-time changes to its geometry.

In the specific case of the Tower, a study conducted on the column/ façade interaction of the structure required the development of an ad-hoc algorithm to optimize column locations based on the building façade’s slope rate. This algorithmic approach, implemented throughout the course of the project, proved useful for design and coordination purposes amongst the multiple disciplines of the design team.

Figure 1. “The Tower,” a confidential project in Southeast Asia, is the example subject.

Authors

Alfonso Oliva, Director

Richard B. Garlock, Partner

Nidhi Sekhar, Computational Designer

Mia Tsiamis, BIM Specialist

Leslie E. Robertson Associates

40 Wall Street, 23rd Floor

New York, NY 10005, United States

t: +1 212 750 9000

e: alfonso@alfonso-oliva.com;

richard.garlock@lera.com; nidhi.sekhar@lera.com;

mia.tsiamis@lera.com

www.lera.com

Alfonso Oliva is the Director of LERA+, LERA’s in-house computational research arm. Through LERA+, he collaborates with design teams on a variety of complex projects and provides instructional seminars to other firms. As a structural engineer, he has led the structural design of sculptural art installations, sports arenas, and supertall towers.

Richard B. Garlock has over 20 years of experience designing efficient structural solutions to facilitate architectural visions. His expertise ranges from university lab buildings to high-rise towers. Garlock was the Project Director for 4 World Trade Center, he was also a member of the WTC Recovery team following 9/11. In addition, Garlock has worked on a number of supertall mixed-use projects internationally, including the BDN Center in Indonesia.

Nidhi Sekhar focuses her efforts on developing and optimizing software that solves problems arising from the structural design workflow, for which she creates complex parametric models and simulations and devises solutions for automating various parts of the design process. She is the cocreator of Cuttlefish, an interoperability platform developed by LERA+.

Mia Tsiamis is a BIM Specialist at LERA. Having first joined LERA on a temporary basis to lead a research collaboration on architectural origami with Sapienza University, Mia Tsiamis rejoined LERA in 2016 and focuses on interoperability and various research endeavors.

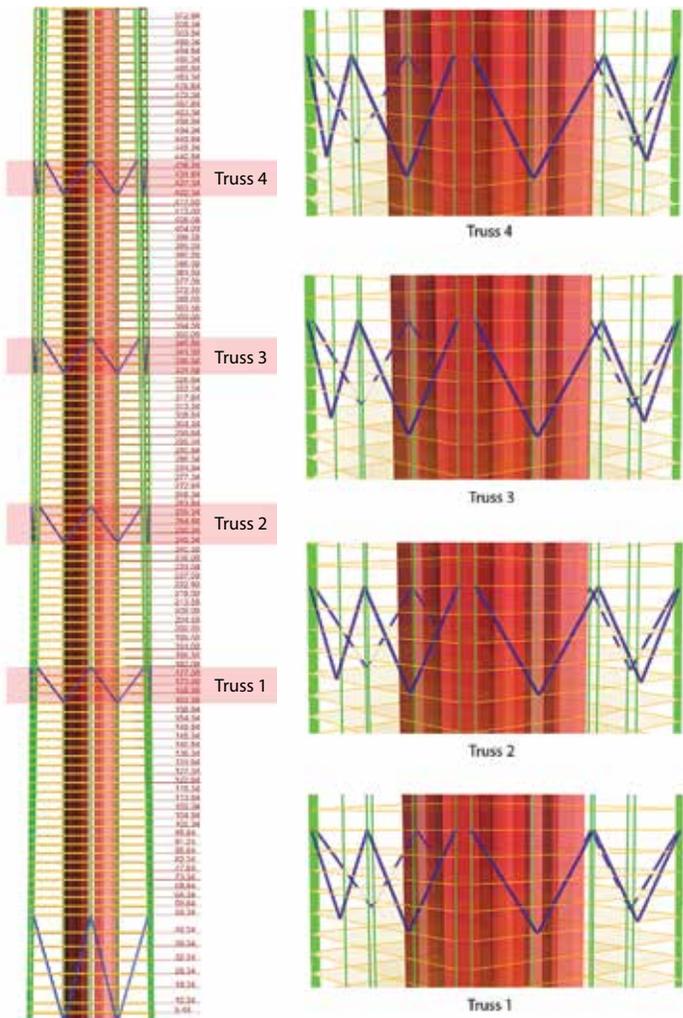


Figure 2. Outriggers optimization study.

Three different algorithms were developed to optimize the number and location of outriggers, their height, and their relative position in the building. This was achieved by linking the parametric definition to a Finite Element Analysis (FEA) software to measure the contribution of the above-mentioned elements in reducing lateral displacements of the Tower. This was studied at different stages of the project. For the algorithm to study approximately 10,000 unique configurations, the run time was about three minutes in each case.

Figure 2 shows screen captures of a typical iteration of the optimization algorithm at an early stage of design.

geometrical and analytical information between these platforms. As was the case for the sloping columns, the crown model was driven by an algorithmic definition in order to quickly readapt to architectural changes. Leveraging the power of algorithms and computation helped to streamline the overall process, as well as to aid exploration of different design options, all within a short period of time (see Figure 3).

Although the workflow presented herein was customized for this particular project, the fundamental framework can be extracted and applied to any high-rise design. The workflow incorporated several tools commonly used in architecture and engineering offices today.

“By reading the data stored in the cloud, the interoperability platform creates a corresponding set of points in the software and generates frames, rigid links, surfaces, and/or volumes based on those points and their associated information.”

The complexity of the crown structure of the Tower necessitated numerous detailed geometrical and analytical studies across different software platforms. The interoperability platform used in the design workflow, described in more depth in subsequent sections of this paper, allowed for the transfer of

- **Rhino 3D** is a freeform software modeler that allows the user to accurately create designs for a variety of purposes, such as engineering, drafting, analysis, and rendering.
- **Grasshopper**, a plugin for Rhino, is widely used by architects, engineers, product designers, and other industry professionals to define parametric control over models. The plugin also enables design exploration by giving the user a graphical interface with which to develop generative design.
- **Autodesk Revit** is a BIM modeling tool widely used in the architectural, engineering, and construction industry for documentation and coordination purposes.
- **Dynamo**, a plugin for Revit, enables parametric control through its graphical interface.
- **SAP2000** and **ETABS** are FEA software used by engineers to study the effect of loads on structures.
- **Cuttlefish** is a custom-developed interoperability platform – composed of a Grasshopper and Dynamo component and a standalone desktop application – that links all of the above-mentioned software and directs the exchange of both geometrical and analytical data between them.

“Once the model was discretized [decomposed] in Grasshopper, relevant data about the Tower’s structure, such as the location in space of frames, slabs, and walls, as well as floor heights and section properties, was extracted using the Cuttlefish-Grasshopper component, a custom algorithm written in C#.”

Anchoring the Design Workflow Via Interoperability

The key to the success of the workflow implemented for the design of the Tower was the use of the interoperability platform, Cuttlefish, a tool developed with the goal of enabling designers and engineers to quickly translate models from one software environment to another. Using Cuttlefish, the geometry from Rhino 3D and Grasshopper could be translated into other software environments, such as CSI SAP and ETABS, and to Autodesk Revit through Dynamo, thus eliminating the task of having to redraw each model in each distinct environment. The conversion process can be summarized in three steps:

Step 1: Discretization of the model

The process began once the Rhino model of the Tower was sent by the architects. The first step was to discretize the model – in other words, to break it down into smaller, digestible components – using Grasshopper, a process that involves building parametric definitions and developing algorithms that recreate the model accurately in a non-native environment (see Figure 4).

The purpose of the discretization was to make the structure easier to analyze computationally by dividing it into simpler components. The simplification process was different for each geometrical element. Frames were broken down into straight-line

segments and floors into smaller three- or four-sided surfaces. Moreover, for the sake of the analytical model, floors were split further based on the intersections of frames and walls, and non-planar surfaces were simplified by converting them into meshes.

Step 2: Cuttlefish-Grasshopper component

Once the model was discretized [decomposed] in Grasshopper, relevant data about the Tower’s structure, such as the location in space of frames, slabs, and walls, as well as floor heights and section properties, was extracted using the Cuttlefish-Grasshopper component, a custom algorithm written in C#.

The component collects the above-mentioned data from the discretized model and stores it in a Structured Query Language (SQL) database located on a cloud server (Amazon Web Services in this specific case) in such a way that the information is easily retrievable and can thus be used to rebuild the model in a different software environment.

SQL is used for the creation and querying of databases. The SQL database consists of a set of tables that store information from different sets of geometry: a table for frames, a table for areas, and so forth. Each table is further subdivided into fields that store geometric information from the model, as well as unique information about the model location, version number, author, and date and time it was created.



Figure 3. Geometric model of building crown.

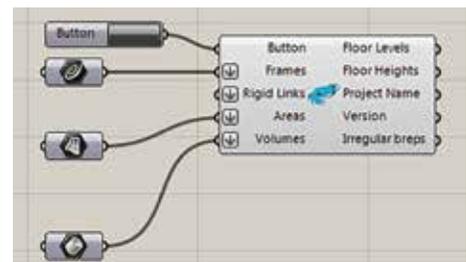
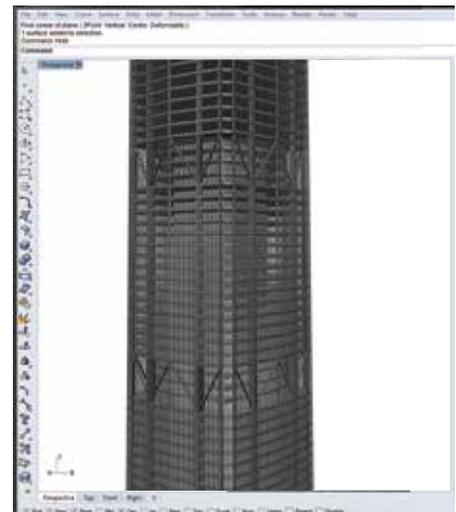


Figure 4. The Tower model created by Rhino (upper) then broken down to components using Grasshopper (lower).

In the example of the Tower model, the data collected from Grasshopper was saved as a set of points, which represented the endpoints for frames and rigid links, as well as vertices for surfaces and volumes of the geometry.

The other advantage of storing such information on an online SQL database is the ability for multiple users to access the resulting translated models remotely and simultaneously.

Step 3: Standalone desktop application

After the SQL database was created, the Cuttlefish standalone desktop application was used to read the data stored in the cloud database and translate it to other software via their built-in application program interfaces (API). In a basic sense, an API is a set of instructions on how to interface with a certain software program. Interacting with the respective APIs of Revit/Dynamo and SAP2000/ETABS, the standalone application produces compatible models that can be moved fluidly between environments. By reading the data stored in the cloud, the interoperability platform creates a corresponding set of points in the software and generates frames, rigid links, surfaces, and/or volumes based on those points and their associated information. Figure 5 illustrates how the source model of the Tower in Rhino was converted into both geometrical (Revit) and analytical (ETABS) models.

Beyond the conversion of data across platforms, the interoperability platform also offered an automated way to track model versions based on models built using the Cuttlefish Grasshopper component. The model version data is stored in a different table of the same SQL database, along with information about the users and time of the build. This also means that all of the models translated using the platform were backed up to the cloud and made available at any time, giving users the ability to switch between different versions of the project without having to continuously save local copies, thus freeing up local drive space. The platform, therefore, served as both a version control and backup tool.

This Cuttlefish feature becomes key when used in coordination with other disciplines. As

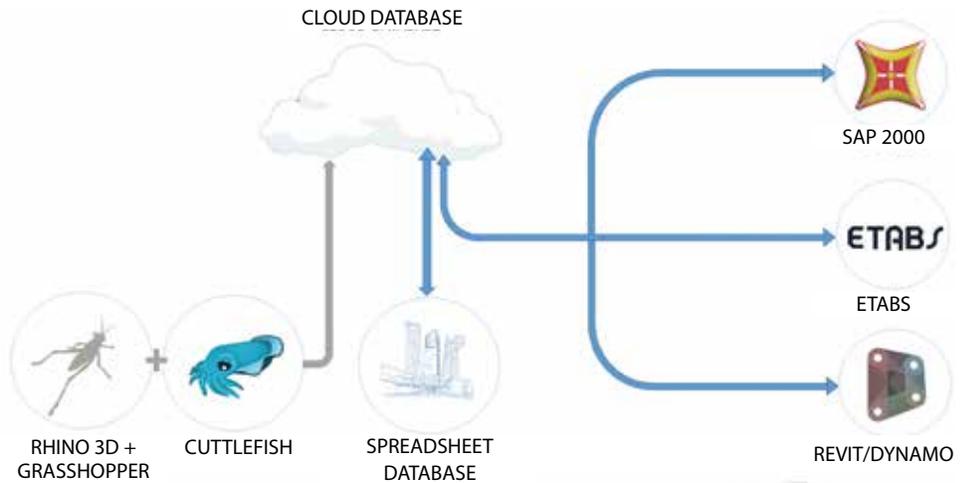


Figure 5. Diagram showing relationship between design platforms.

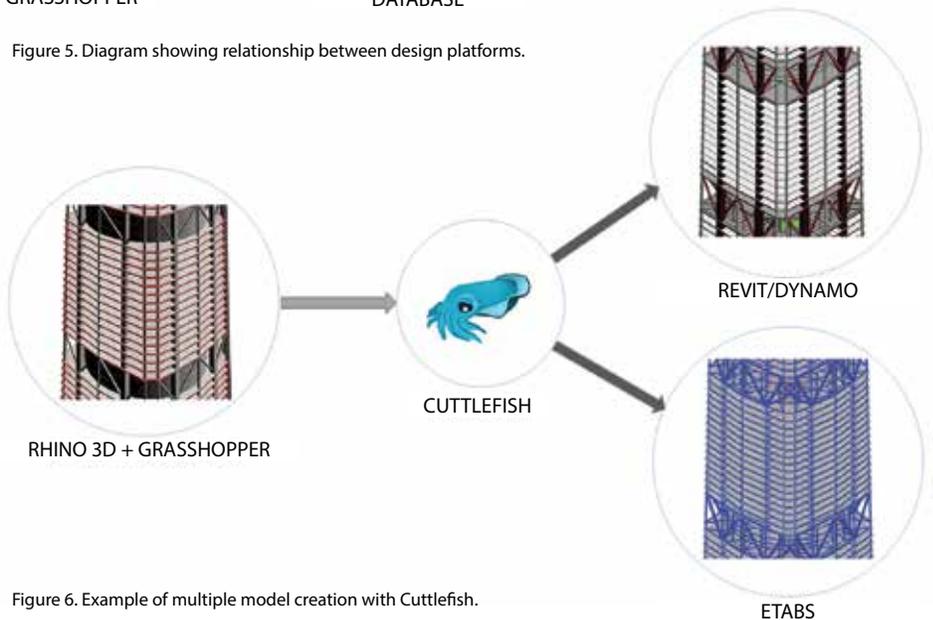


Figure 6. Example of multiple model creation with Cuttlefish.

access can be allowed and restricted at any time, the project could be shared, in part or in whole, with people outside of the team or company, without the need to share any file. The only requirement is the installation of the Cuttlefish application.

All of this was executed through a common user interface (see Figure 6). The overall flow of exchanges, from initial architectural model to translated structural/BIM models, is summarized by the diagram in Figure 7.

Using a Cloud-Based Database To Generate Analytical Models

Analytical models of the Tower generated with Cuttlefish for CSI SAP2000 and CSI

ETABS were used to perform structural analysis for the Tower and specific building components. The entire building structure was analyzed in CSI ETABS, while local studies on the crown were conducted in CSI SAP2000. CSI ETABS was used primarily to conduct analysis for gravity and lateral loads.

The images in Figure 8 depict iterations of the Tower and crown models, respectively. Apart from the difficulties inherent in building such complex models into FEA software with a high level of accuracy, the even greater challenge was to be able to react quickly to changes made by the architect. This is where the parametric modeling environment, combined with the ad-hoc platform used to transfer geometrical information across software,

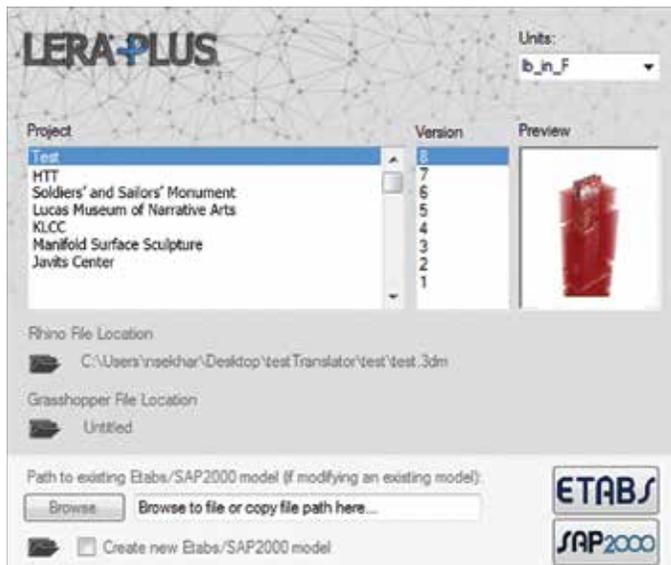


Figure 7. Cuttlefish user interface.

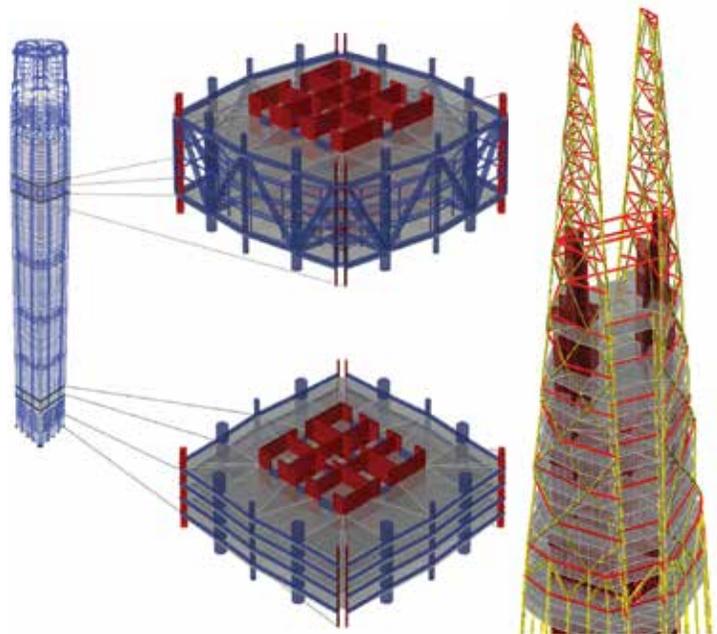


Figure 8. Building and crown analytical models.

enabled the design team to react to project changes with great agility.

A further advantage of utilizing an interoperability-based design workflow was the use of a native set of in-house design spreadsheets. Results from the structural analysis of the Tower, stored in the cloud database, were output upon request via the Cuttlefish user interface. With multiple users able to work on the same set of analytical data simultaneously, the possibility of introducing errors was decreased, and productivity was increased. In this instance, reading data from the cloud was reported by users to be considerably faster than extracting results from an FEA model.

Documenting the Tower in Revit Using an Algorithmic Approach

In order to document the design of the Tower, a Revit model was prepared and used to create drawings. The process of generating the Revit model was integrated into the workflow by creating a link between the architectural Rhino model, the structural analysis model, and Revit. This was done using Cuttlefish and Dynamo for Revit.

In order to translate the geometry from the Rhino model into Revit, the architectural volume was divided horizontally according to story height and vertically according to a column grid. At each level, the volume

cross-section was intersected with column centerlines to retrieve a set of points. Each column was defined as a line connecting a point on a given level to the associated point on the next level above. Slabs were designated as surface elements. The point coordinate data, as well as the column connectivity and surface data, was extracted from the Rhino model and stored in tables in the cloud.

The custom Dynamo Cuttlefish component read the point coordinate data and element connectivity from the cloud and translated it into Revit. Using the stored geometric data, the node generated a model of the structural system in Revit. The partial view in Figure 9 highlights the columns and belt truss systems. Once the structure had been analyzed in ETABS and the element section sizes had been calculated, the next step in the workflow was to assign section properties to the line elements and slab properties to the floor surfaces in Revit.

The resulting analysis and design yielded three types of columns: megacolumns, middle columns, and interior columns. The megacolumns have a non-standard cross-section, and this section tapers along the height of the structure. Using Grasshopper, points were generated at every floor level, representing the vertices of the

“Goals for development include making the interoperability platform smarter by enabling automatic updates of models across different software environments, so that whenever changes are made in one environment, the models in other environments are able to instantly absorb and react to the changes.”

megacolumn section perimeter. Working points were stored in the cloud and then translated into Revit through Dynamo. The Dynamo component used these points to draw a polycurve at each floor level, which was then lofted between adjacent floor levels to create a solid element representing each column. These solid elements were then imported into the Revit model. For all other columns and slabs, the Dynamo definition was used to assign standard section sizes and slab thicknesses according to the output from the analysis models, using standard Revit families (groups of elements sharing common properties).

Conclusion and Further Development

The speed and efficiency with which the design of the Tower was developed and optimized could not have been achieved without the use of an interoperability-based design workflow.

At the base, the parametric definition in Grasshopper created at the beginning of the process allowed the design team to dynamically react to architectural changes. Similarly, the flexibility of the Dynamo

definition enabled the Revit model to be easily updated when any changes were made to the global geometry or structural section assignments.

The ease with which the geometry and structural element designations could be transferred between Rhino, SAP2000, ETABS, and Revit made it possible to readily perform iterations of the structural design and to evaluate alternatives, all while developing documentation models in parallel for communicating the design to all parties in the design team.

Many aspects of the workflow can still be further developed and improved. Goals for development include making the interoperability platform smarter by enabling automatic updates of models across different software environments, so that whenever changes are made in one environment, the models in other environments are able to instantly absorb and react to the changes.

The suite of compatible software can also be expanded in order to make the workflow even more effective. The addition of compatibility with virtual reality (VR) tools can be incorporated to aid in design

coordination and analysis. The immersive experience that VR offers holds powerful implications for the future of design, allowing the user to experience a model at real scale and to document, coordinate, and make changes in real time. Used in tandem with an interoperability platform, designers could achieve a greater level of efficiency and seamlessness than ever before. ■

Unless otherwise noted, all image credits in this paper are to Leslie E. Robertson Associates (LERA).

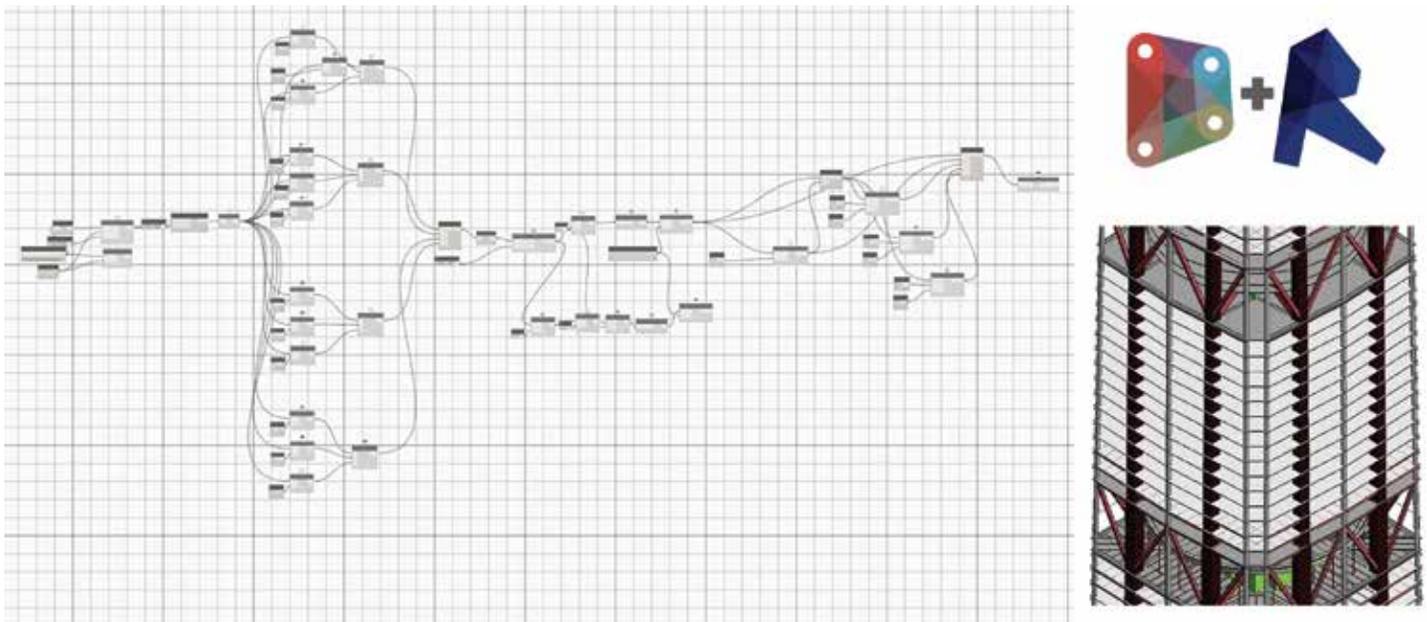


Figure 9. Dynamo definition and Revit model.