AUTODESK FORGE RESEARCH PROJECT

A Web-based Parametric BIM Viewer for Online Collaborative Parametric Design Presentation and Optimization

Dr. Wei Yan, Professor of Architecture
Texas A&M University

1. Prior Work

Design presentation with Autodesk Design Review was created previously.

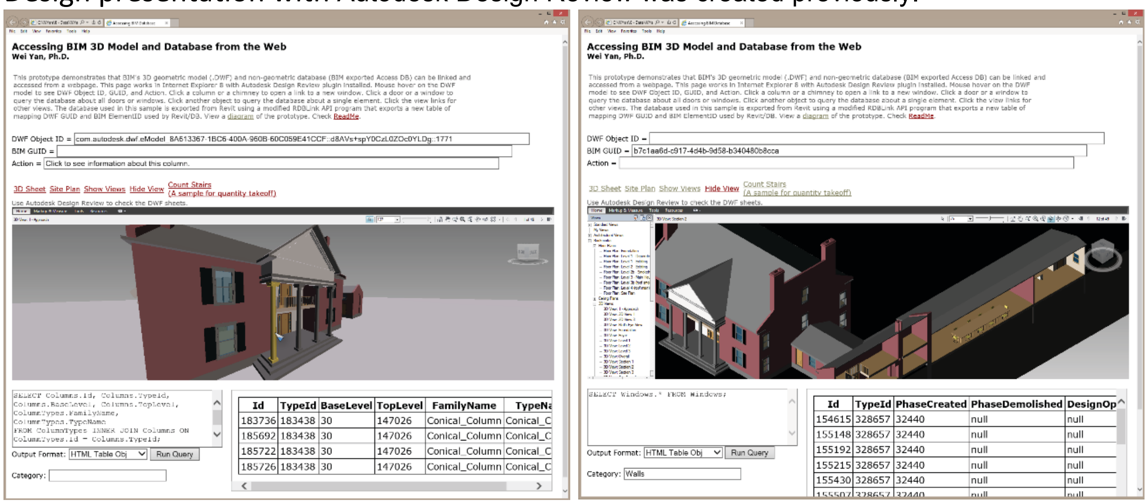

Figure 1. Prior work: integrated system allowing BIM model geometry and parameters to be viewed in Autodesk Design Review, a plugin of Internet Explorer. (The Revit model of the historic building was created by Leslie Leffke at Texas A&M University.)

2. Forge Project

Using Autodesk Forge Viewer API and Model Derivative API, the same BIM model was uploaded to Autodesk Forge cloud, translated into SVF format, and displayed in Autodesk Viewer online, using Javascript-based WebGL technology, without the need of a plugin.

2.1. Experimenting Autodesk Viewer Basics

To understand Autodesk Viewer, the author studied WebGL technology extensively, and learned about the APIs of Viewer and Model Derivative. As an outcome of the study, the historic building sample is displayed in Viewer as seen in Figure 2, 3 and 4.

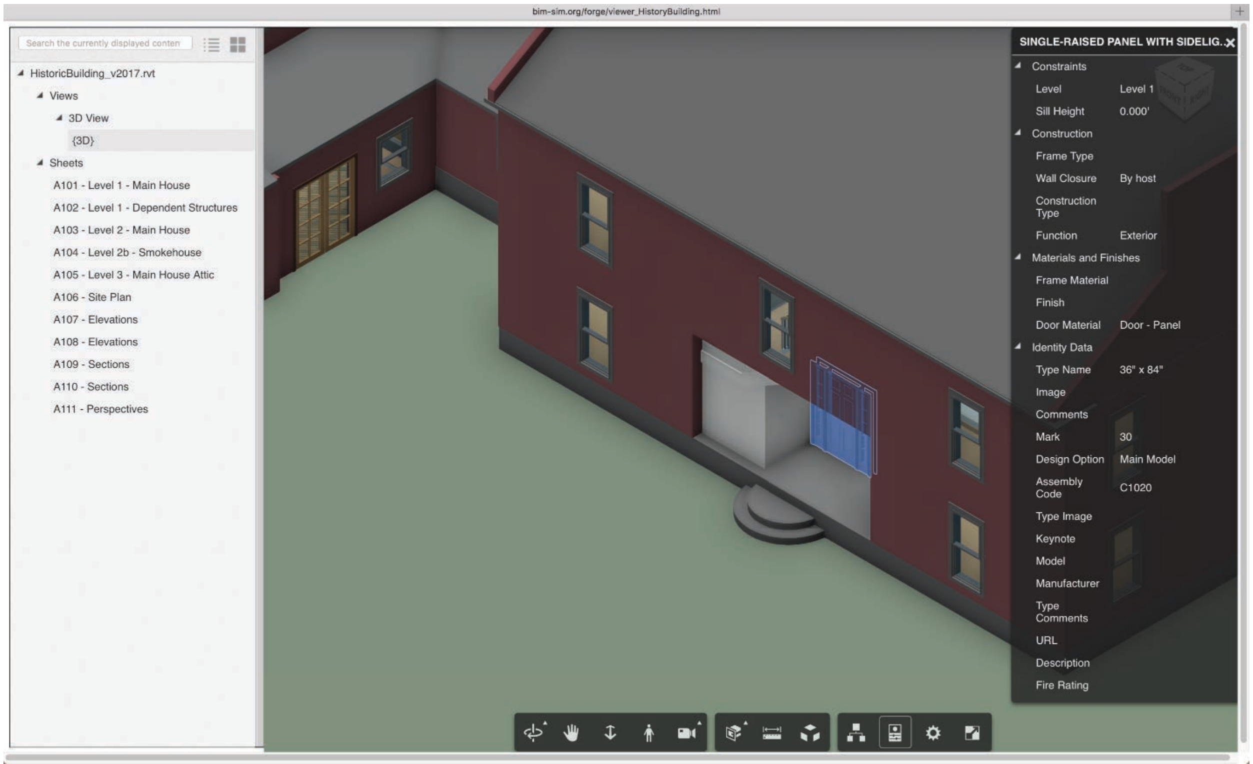

Figure 2. The historic building displayed in Autodesk Viewer with 3D, 2D videws and parameters.

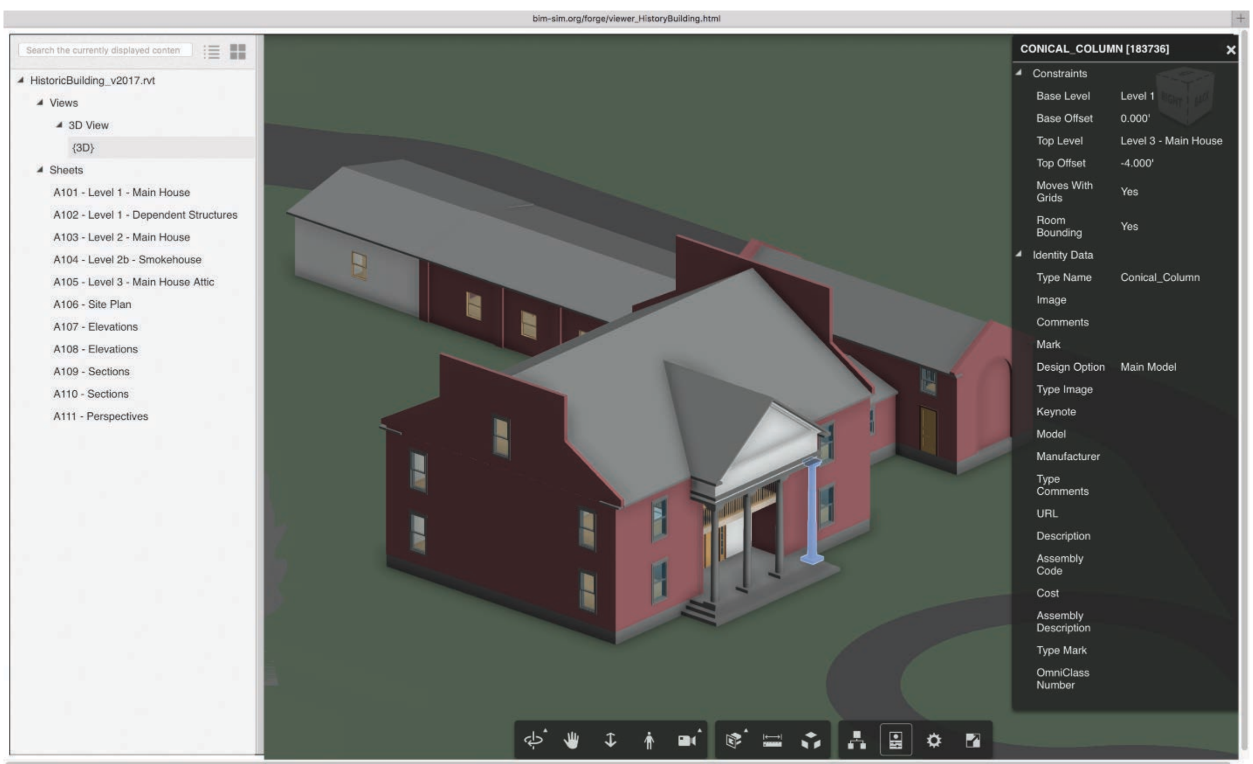

Figure 3. A different 3D view of the historic building displayed in Autodesk Viewer.

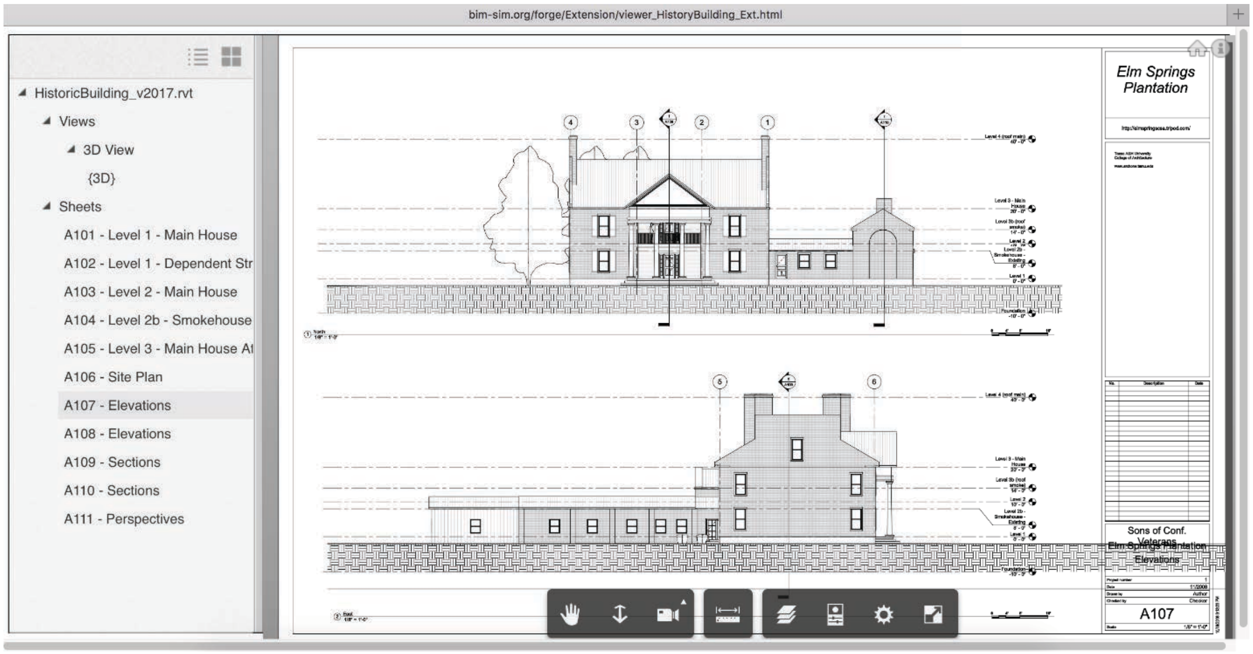

Figure 4. A 2D view of the historic building displayed in Autodesk Viewer.

A video showing the sample building's 3D and 2D documents in Viewer can be seen from the following link:
http://mediamatrix.tamu.edu/download_published_file.php?published_file_id=563991

Further, Review Viewer **Extensions** were studies for creating a prototype of Web-based Parametric BIM Viewer. A sample extension for editing the building model in Viewer is created. In the sample, the building objects, e.g. roofs, walls, can be moved away from their original locations to allow examining of the interior spaces. The move is done by user clicking the objects and moving mouse (Figure 5 and 6).

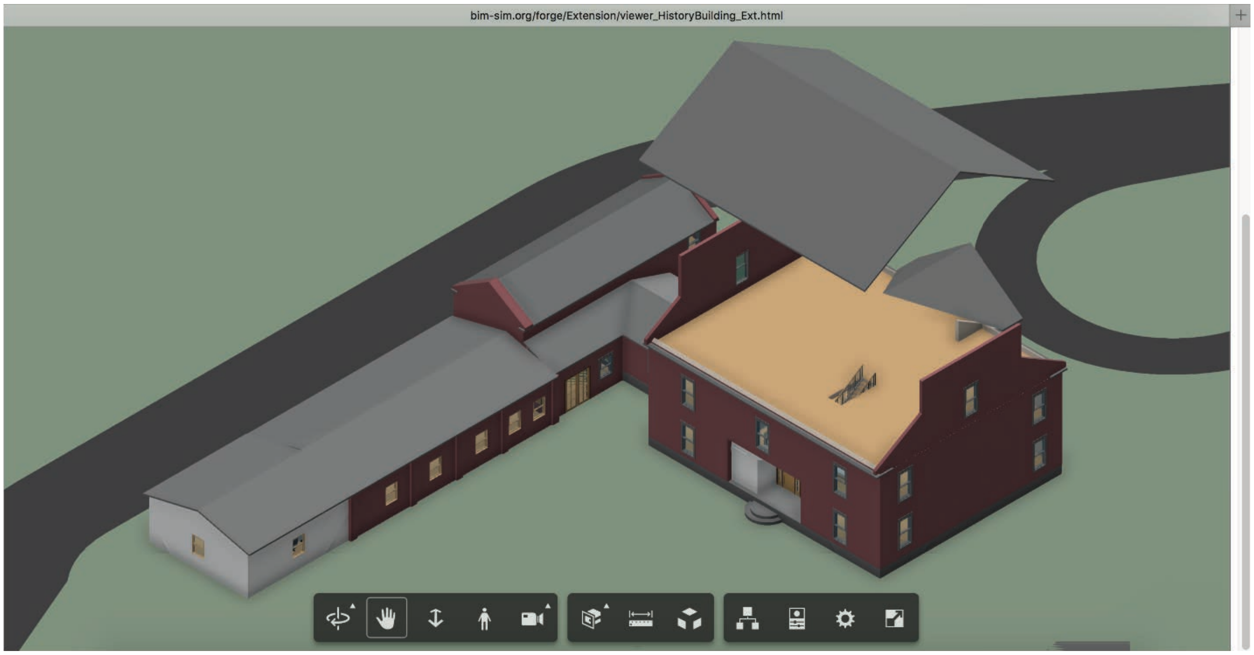

Figure 5. A roof is moved from the building, demonstrating the modeling editing capability in Viewer through using Viewer Extensions.

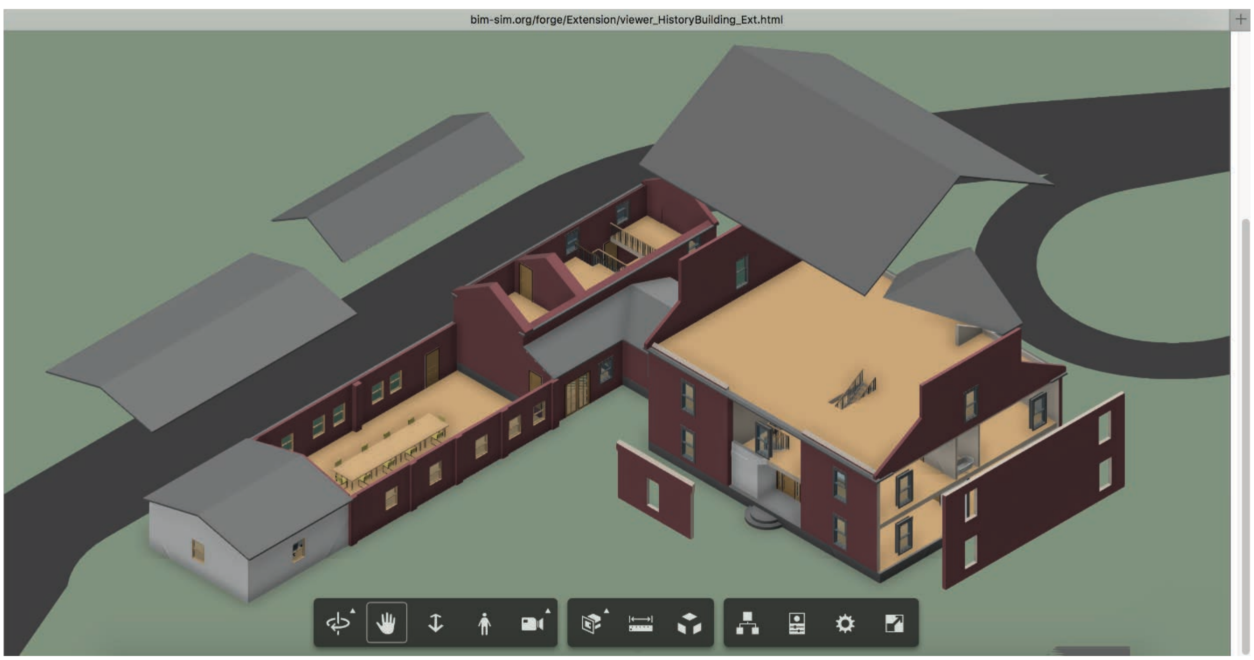

Figure 6. More roofs and walls are moved from the building, showing interior spaces.

The Viewer Extension sample for editing the building objects can be seen in the following video: http://mediamatrix.tamu.edu/download_published_file.php?published_file_id=563965

2.2 Developing integrated Visual Programming with Autodesk Viewer – Google Blockly as a simple sample

The aim of the project is to prototype a Web-based **Parametric** BIM tool. For parametric modeling, the **Visual Programming** method is very effective and widely used in the design community. Therefore, the project is to identify the possible Visual Programming tools that can be integrated into Autodesk Viewer, to enable parametric changes to the building models displayed in Viewer. The author considered approaches such as Autodesk Fractals, Flux, Dynamo, and Grasshopper, as well as Google Blockly.

Google Blockly is a Javascript-based visual programming tool. It's been used for online games, education, and other applications. The author experimented Blockly and it's capability to work with Autodesk Viewer API through Javascript programming.

In the sample below, Google Blockly blocks and Autodesk Forge Viewer are inserted into the same sample web page. A simple building sample is displayed in Viewer.

The blocks can be used to edit the model in Viewer. For example, adding a block of "number" allows a user to change the building model – move the roof up – by the specified number. A user can set value 10 in Google Blockly's number block. Autodesk Viewer can communicate with Blockly to obtain the value, and move the roof accordingly (Figure 7 and 8).

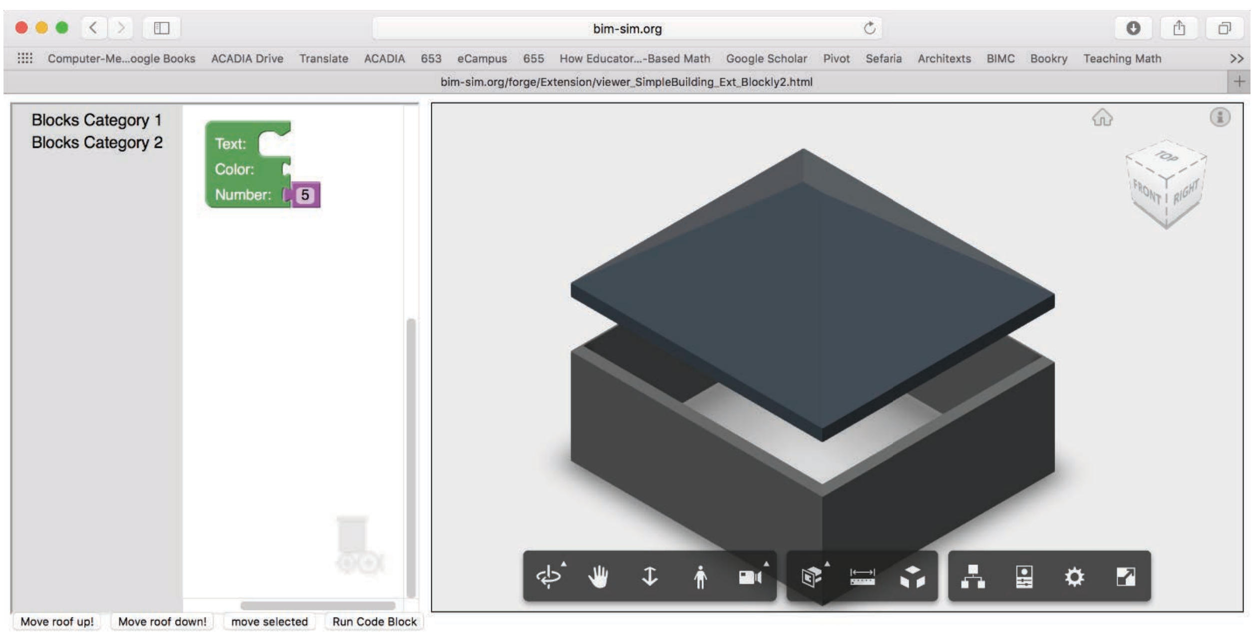

Figure 7. Roof moves up by 5.

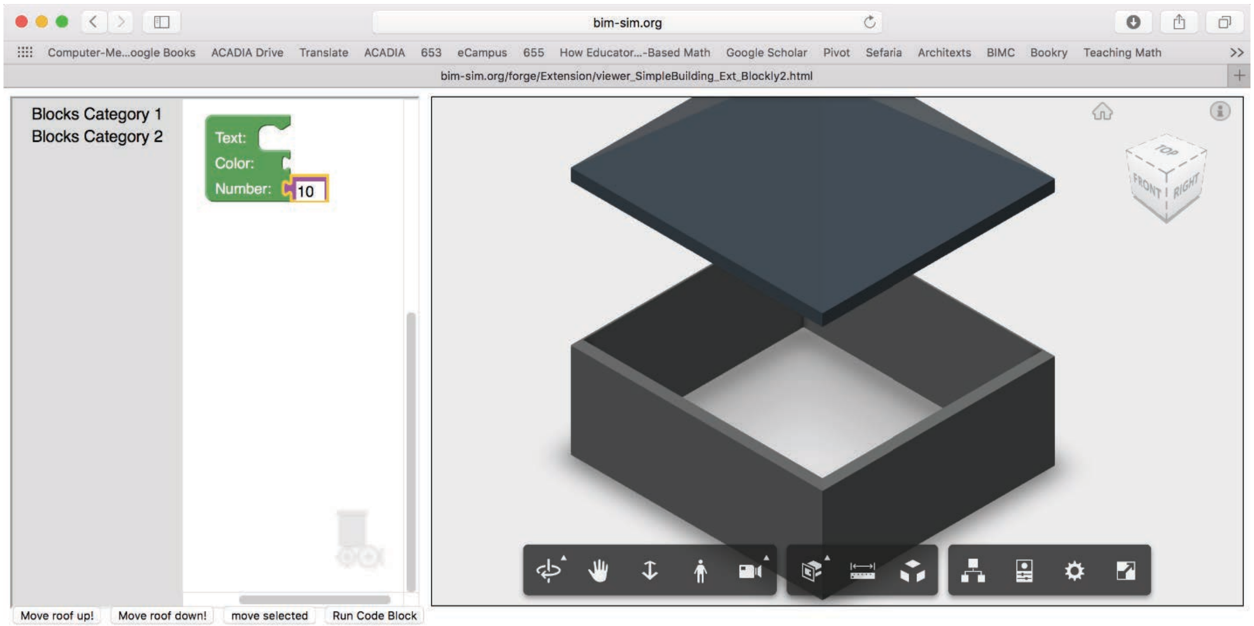

Figure 8. Roof moves up by 10.

The sample showing integrated Autodesk Viewer and Google Blockly for editing the building model can be seen from the video below:
http://mediamatrix.tamu.edu/download_published_file.php?published_file_id=563968

2.3 Developing integrated Dynamo and Grasshopper with Autodesk Viewer through Flux

The author researched on the use of Flux to facilitate the integration of major visual programming tools for design, including Dynamo and Grasshopper, with Autodesk Viewer, and developed a prototype for demonstrating the web-based parametric modeling method using visual programming and Autodesk Viewer. The prototype is created using JavaScript and Viewer API/Extension. A sample of running the prototype is described below.

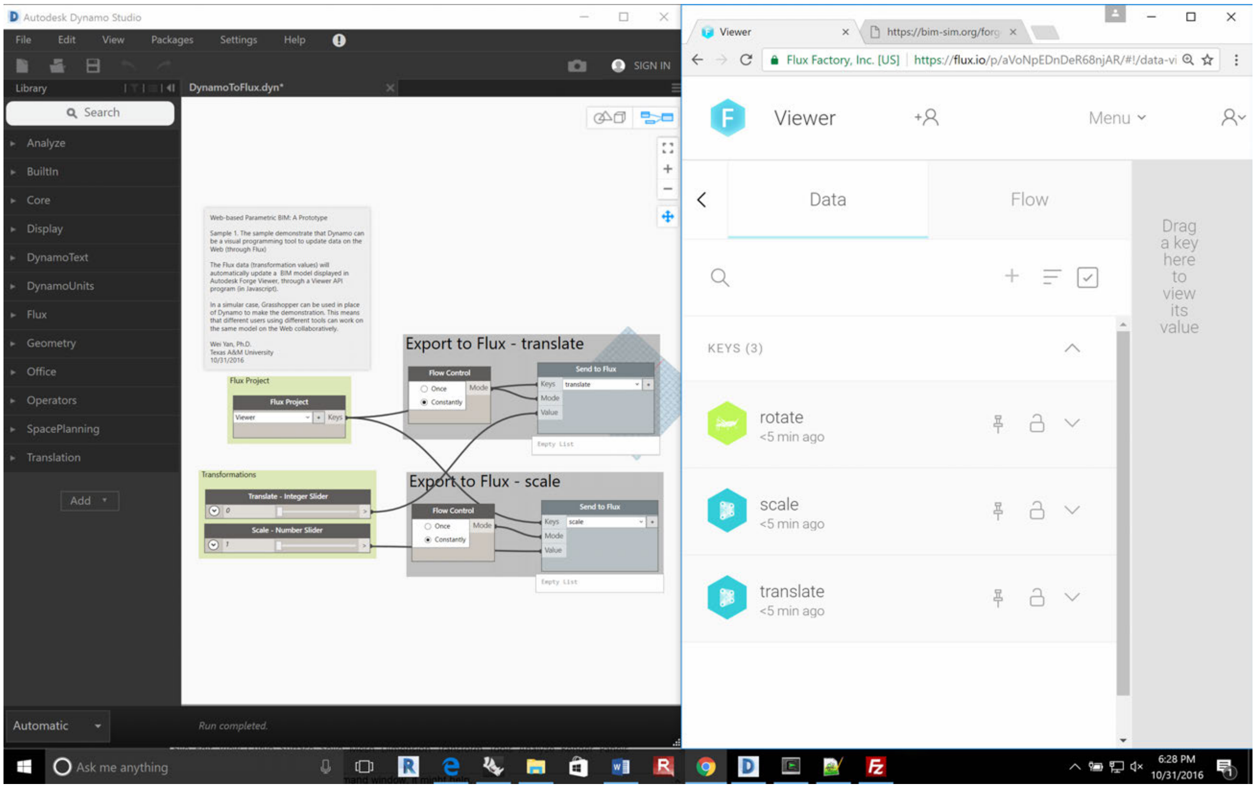

Figure 9. Integrating Dynamo through Flux – the Dynamo program and Flux project.

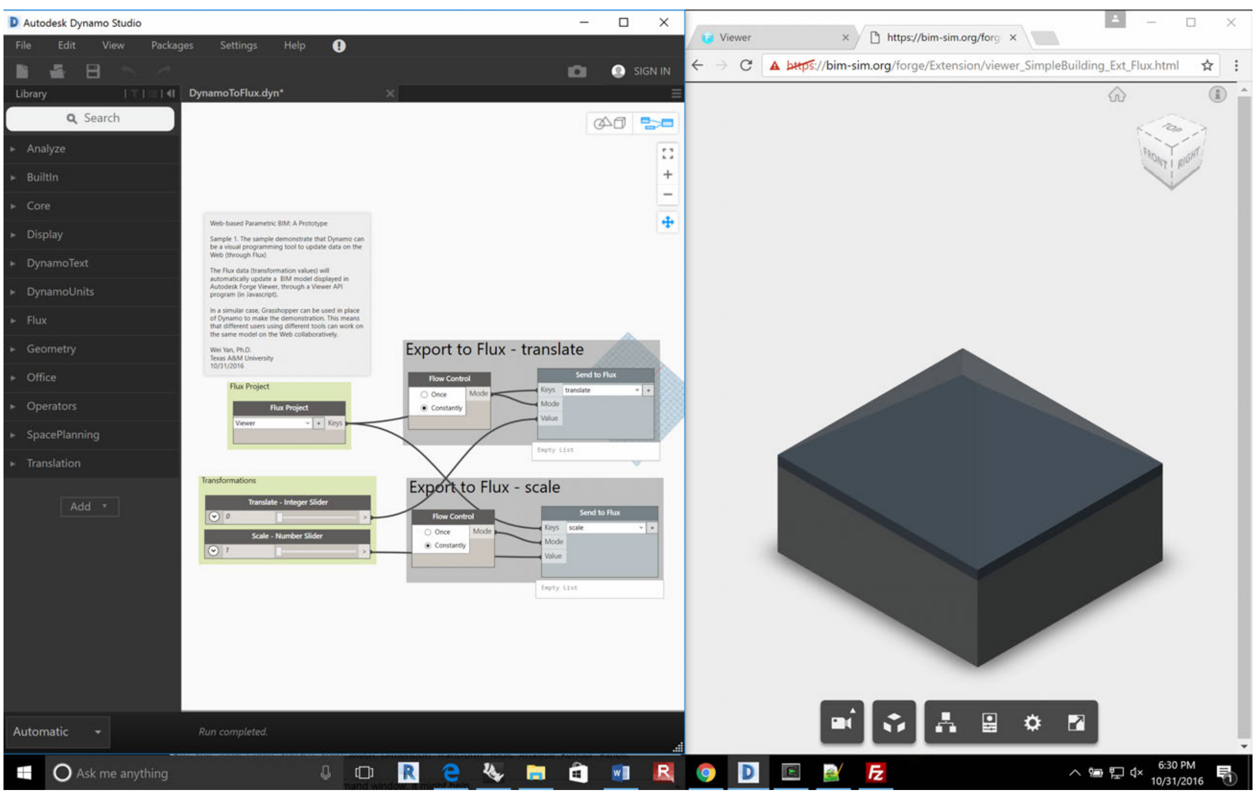

Figure 10. Integrating Dynamo through Flux – the Dynamo program and the Forge Viewer with the simple building sample.

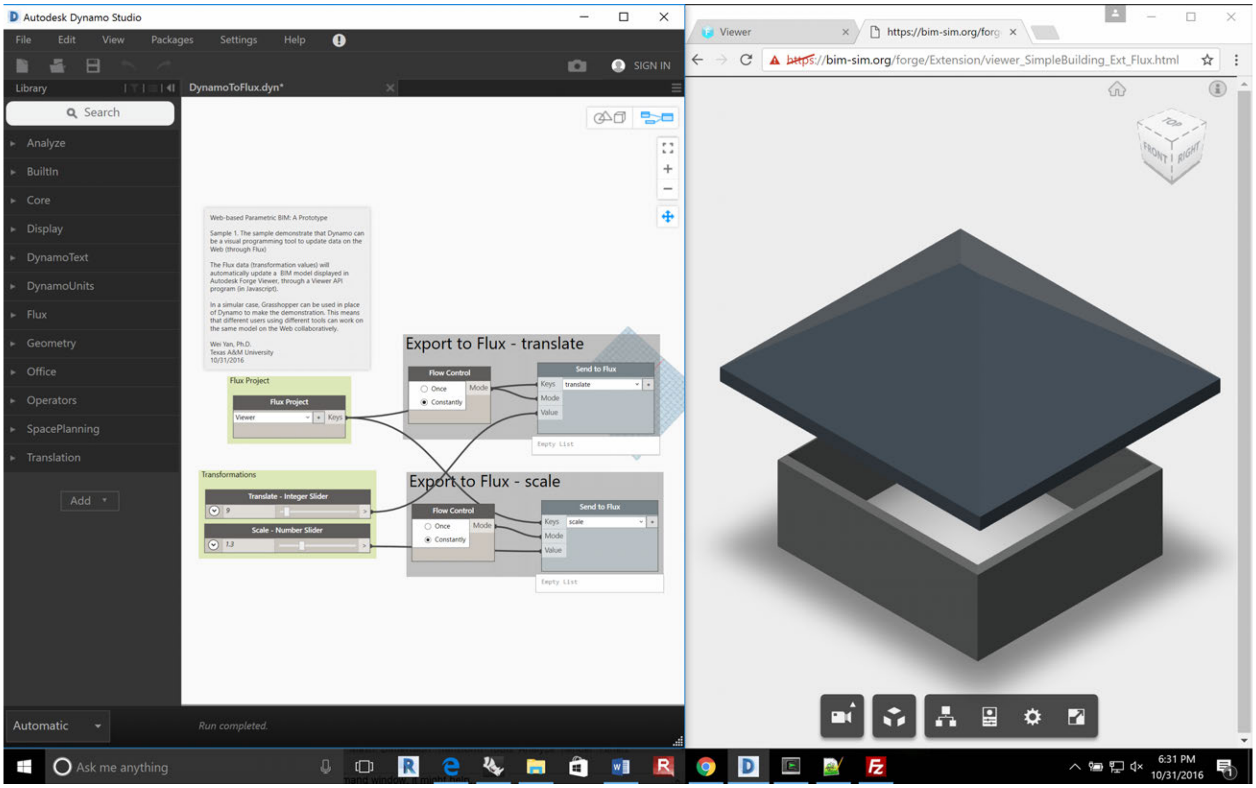

Figure 11. When the slider of Translate is changed, the translation data is sent to Flux, and the Viewer receives the data from Flux, and makes the translation of the roof in real time. Similarly, the slider of Scale is used to control the scaling of the roof.

After the integration between Viewer and Dynamo through Flux, it’s similar to integrate Grasshopper with Flux. Figure 12 demonstrates it.

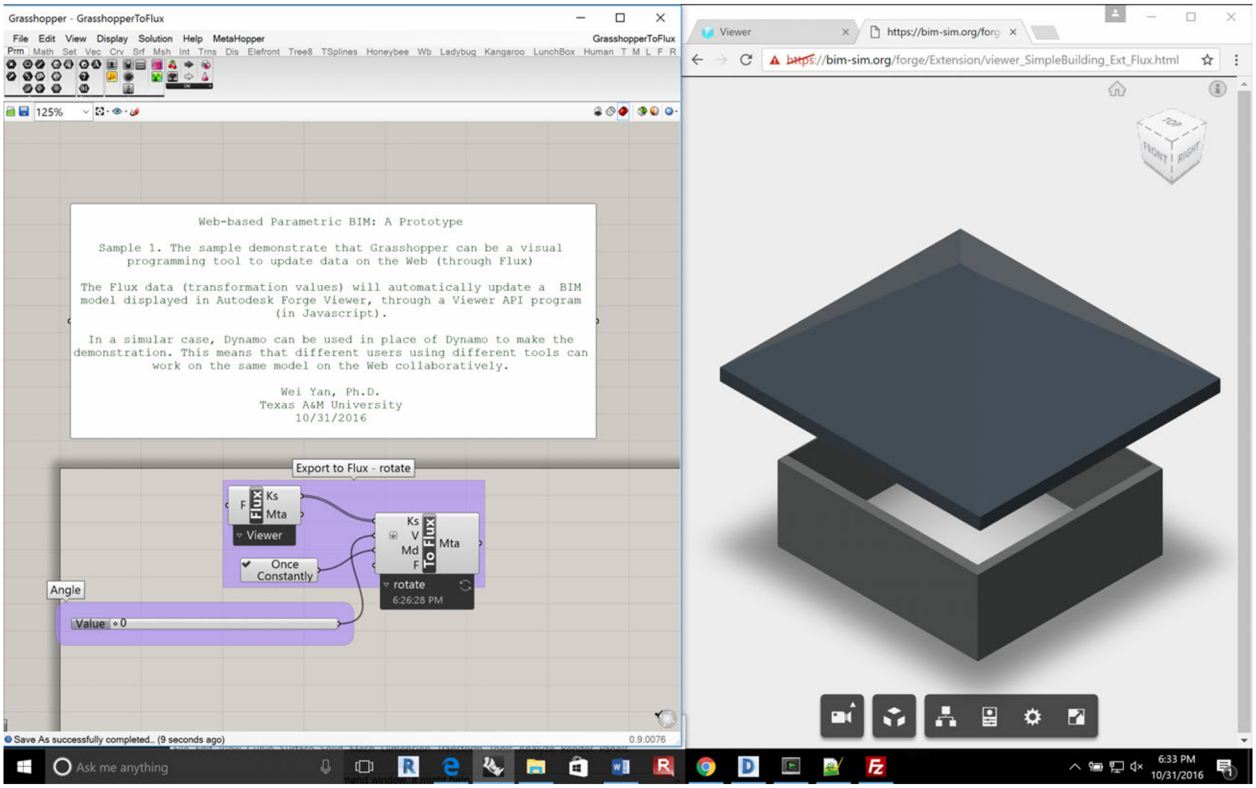

Figure 12. Integrating Viewer and Grasshopper through Flux.

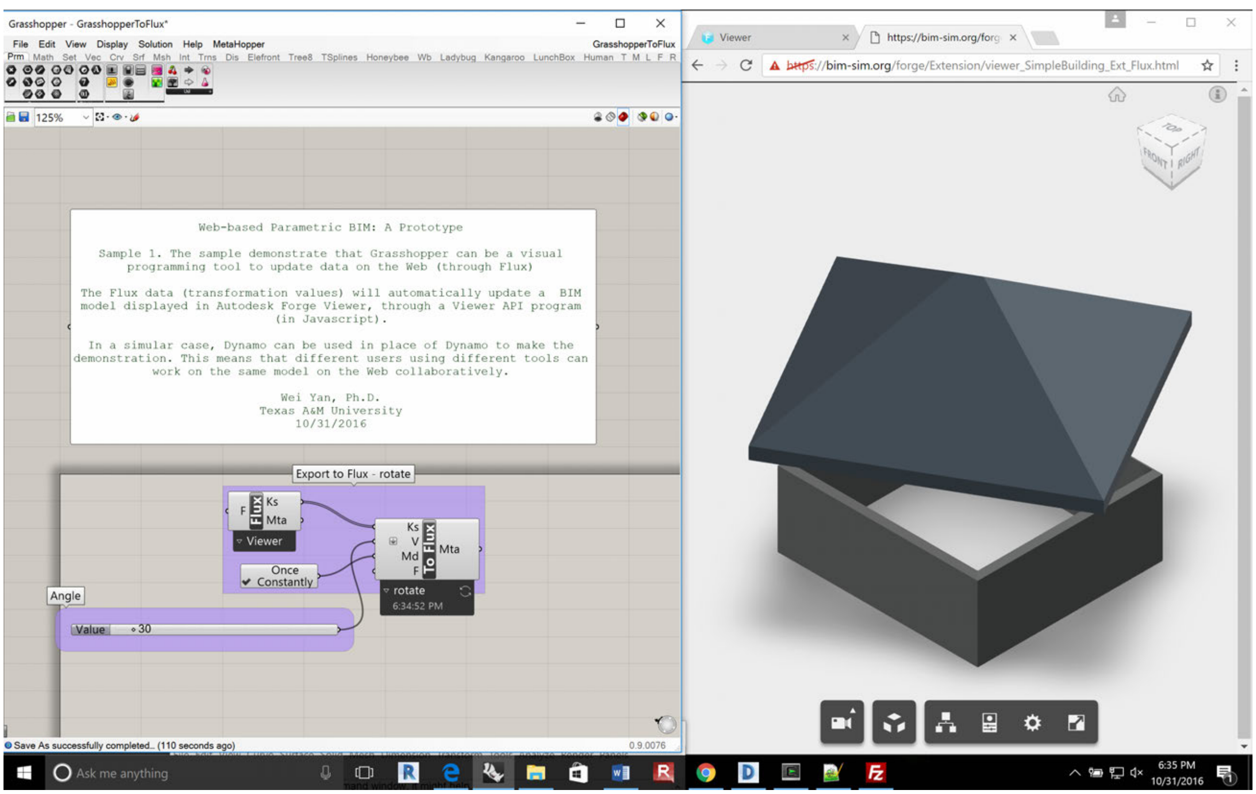

Figure 13. The Grasshopper slider of Angle is used to change the roof rotation.

More interestingly, the author developed a simulation of two users working on the same building model in Viewers: (1) User 1 controls the Dynamo program for roof translation and scaling with Viewer on Google Chrome; and (2) User 2 controls the Grasshopper program for roof rotation with Viewer on Microsoft Edge (Figure 14).

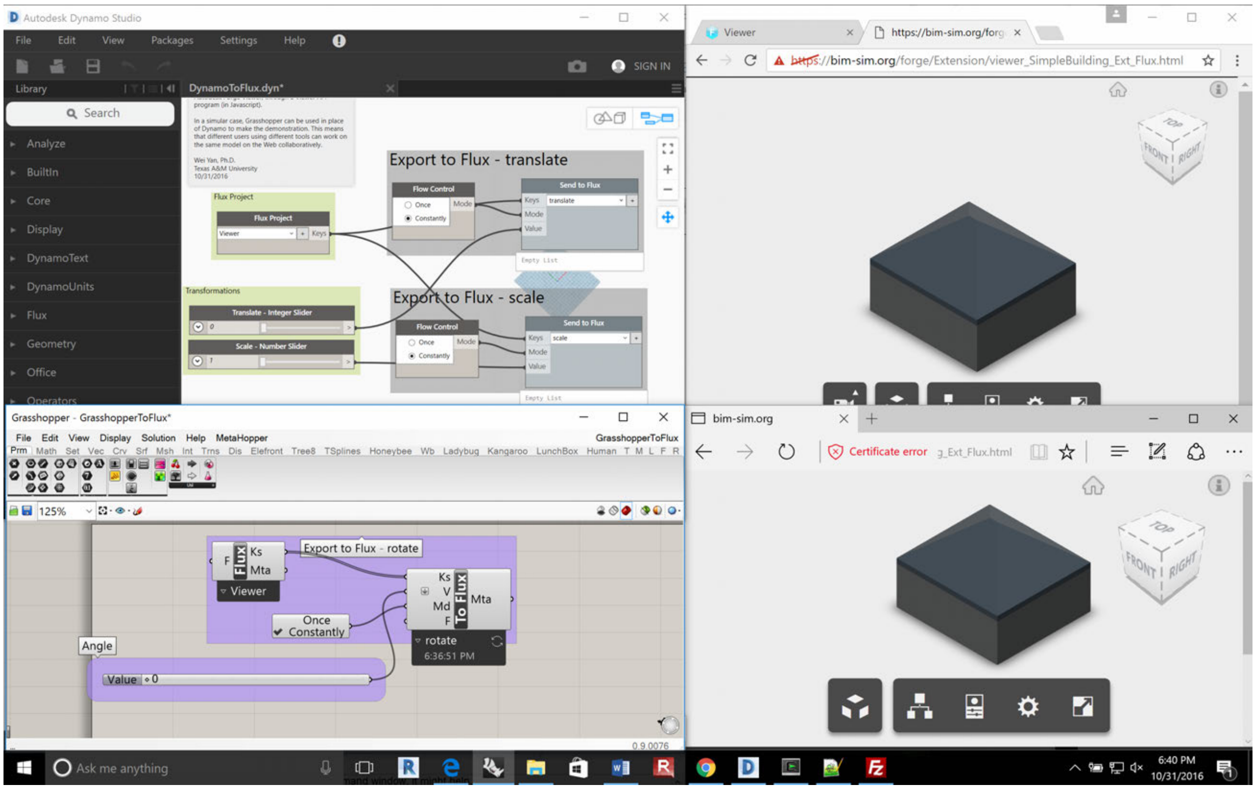

Figure 14. A simulation of two users (top and bottom) using different browsers (Google Chrome and Microsoft Edge), with different visual programming tools (Dynamo and Grasshopper), controlling different transformations (translation+scaling and rotation), through the use of Flux. The simulation is down on one computer.

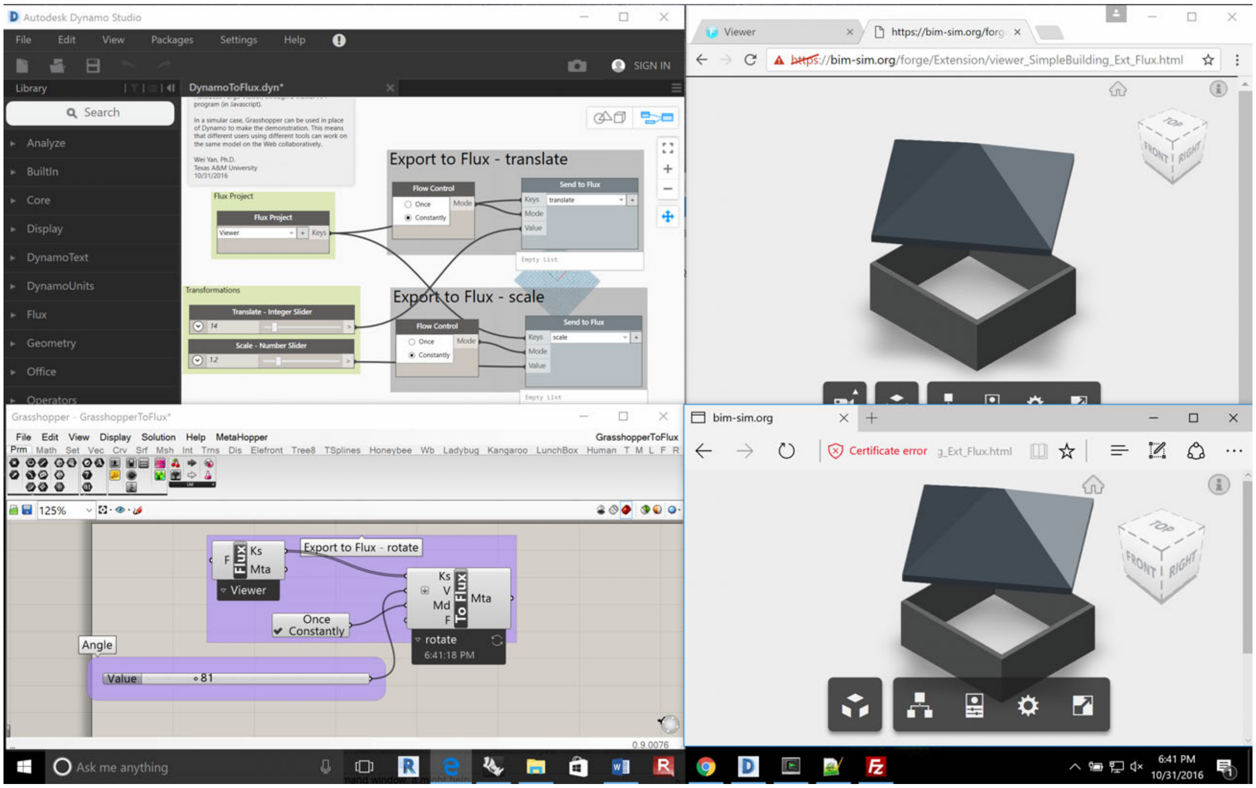

Figure 15. User 1 (top) controlling Dynamo and using Viewer in Chrome to translate and scale the roof. User 2 (bottom) controlling Grasshopper and using Viewer in Edge to rotate the roof. The transformations are synchronized for the two users. (The simulation was done on one computer).

The following video shows the above integrated Autodesk Viewer, Flux, Dynamo, and Grasshopper in action:
http://mediamatrix.tamu.edu/download_published_file.php?published_file_id=564315

Finally, the actual use of two computers to test the collaborative parametric modeling between two users was conducted. In the test, one computer runs Viewer and Dynamo to change the translation and scaling, and the other computer runs Viewer and Grasshopper to change the rotation. The transformations are synchronized for the two computers.

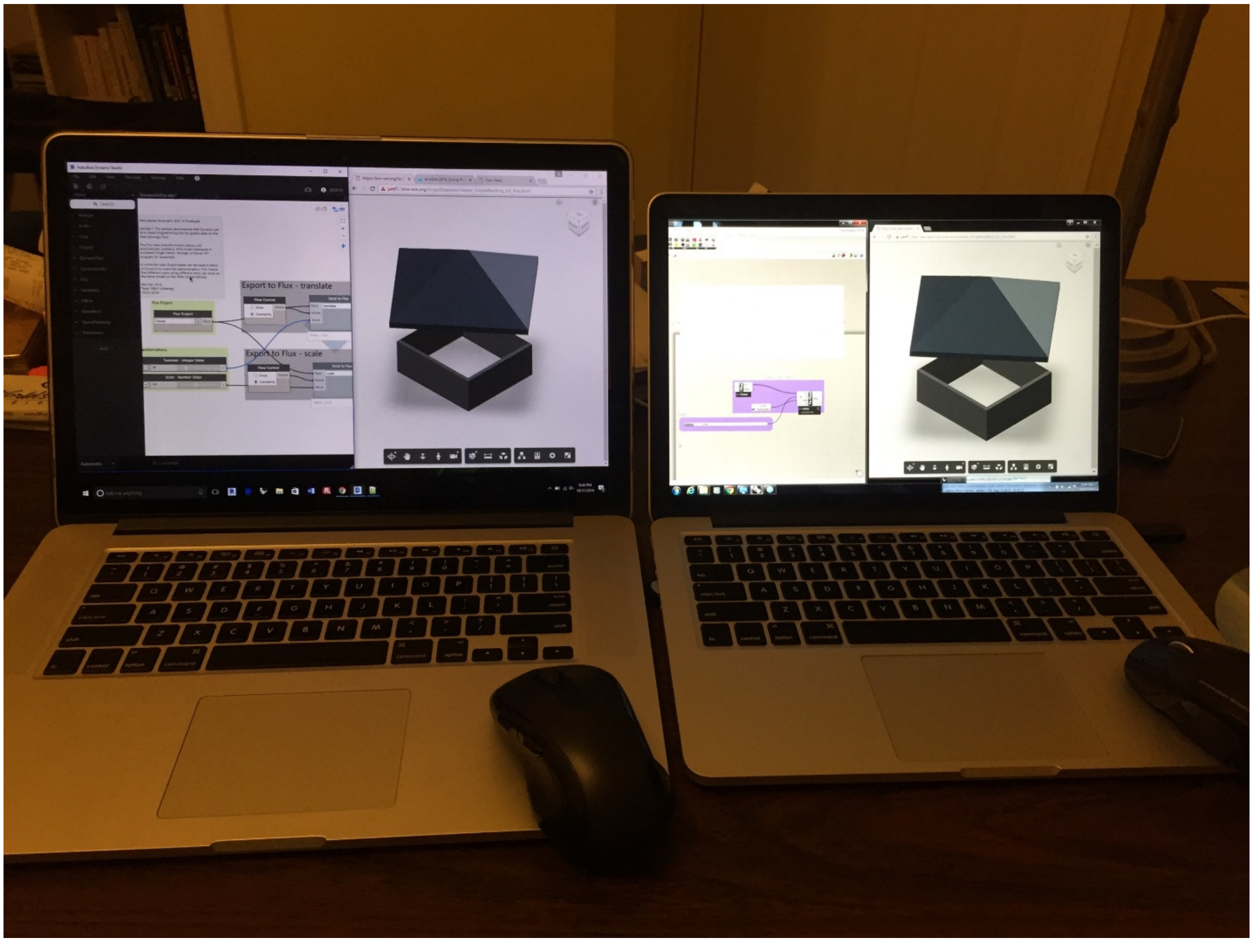

Figure 16. Two computers (representing two users collaborating on parametric modeling with visual programming tools). One computer controls the translation and scaling of the roof, and the other computer controls the rotation of the roof. The models in the Viewers on the two computers are synchronized.

The following video demonstrates the integrated Autodesk Viewer, Flux, Dynamo, and Grasshopper, running on actual two computers:
http://mediamatrix.tamu.edu/download_published_file.php?published_file_id=564317

2.4 Optimization with integrated Grasshopper, Autodesk Viewer, and Flux

The next goal of the project is to demonstrate a framework of Web-based parametric modeling supporting design optimization. A prototype is made to utilize the Genetic Algorithm tool (Galapagos) in Grasshopper to automatically generate design options and improve the options for sample design objectives. During the optimization process, Viewer is able to display the varying design options. In the sample as shown in Figure 17, the design objectives are: through changing the rotation and scale of the roof, make the roof:

1. Cover the whole building floorplan (50ft x 25ft)

2. Cover the front area of the door (i.e. a location point that is 12.5ft always and in front of the door needs to be covered by the roof. See Rhino plan drawing for the relation between the location point and the building floorplan.)
3. The area of the roof should be minimal (i.e. the scale factor of the roof should be as small as possible).

While the Dynamo program can still be used to change the translation and scaling of the roof, the Grasshopper program now contains a Galapagos (Genetic Algorithm) node, which uses the rotation angle and the scaling factor of the roof as parameters, and a fitness function that combines the above design objective.

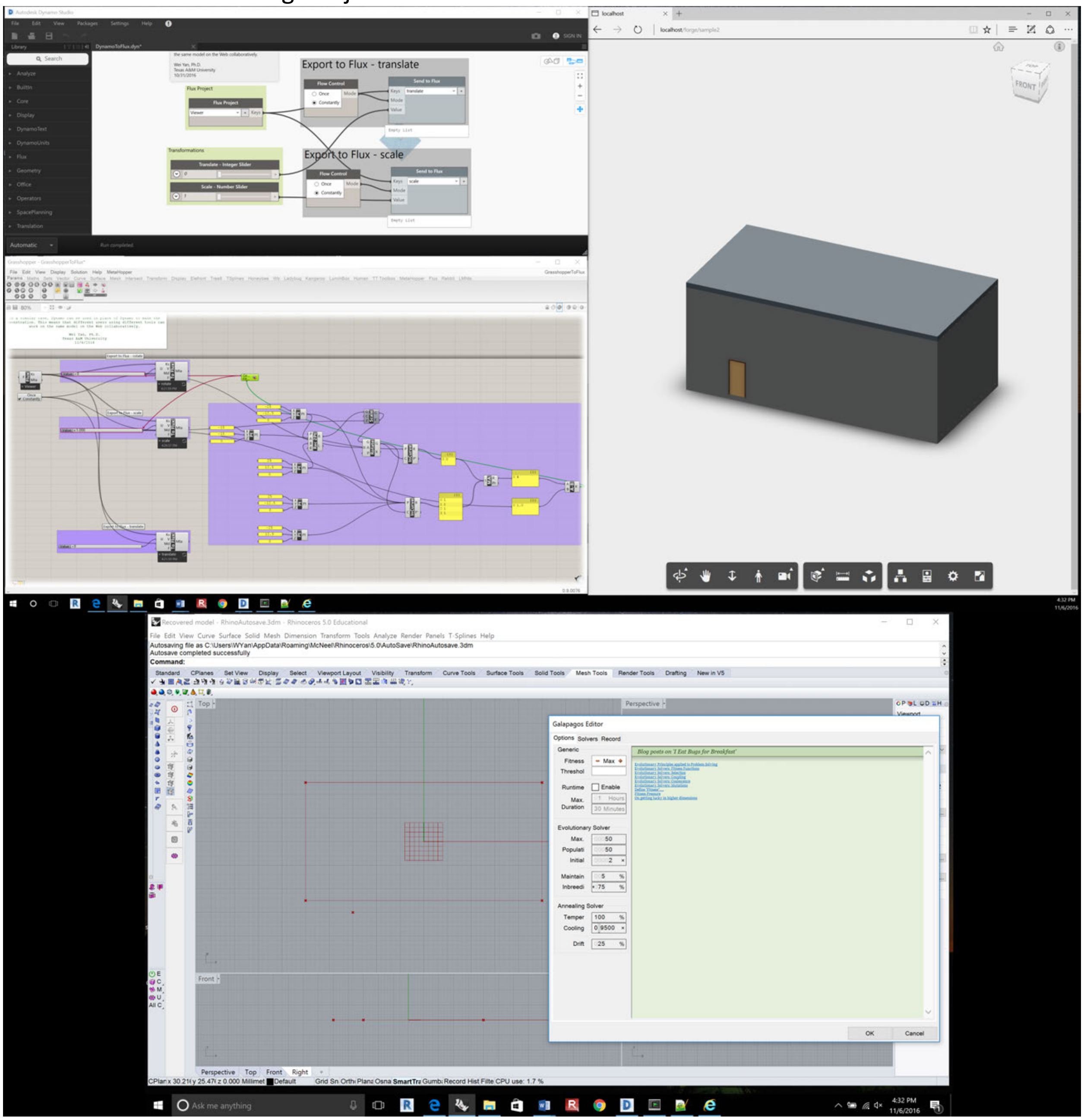

Figure 17. Sample Web-based parametric modeling and design optimization. Top: Dynamo program can change the translation and scale of the building roof. Grasshopper program contains a Genetic Algorithm node – Galapagos for optimization. Viewer displays the building model. Bottom: Rhinoceros program shows the building plan and the front location point of the door. A Galapagos editor is also displayed. Optimization settings are completed.

Running the Genetic Algorithm, multiple generations of design options are created automatically and evaluated against the fitness function. In the process, Viewer displays the multiple design options and the final optimal solution automatically (Figure 18 and 19).

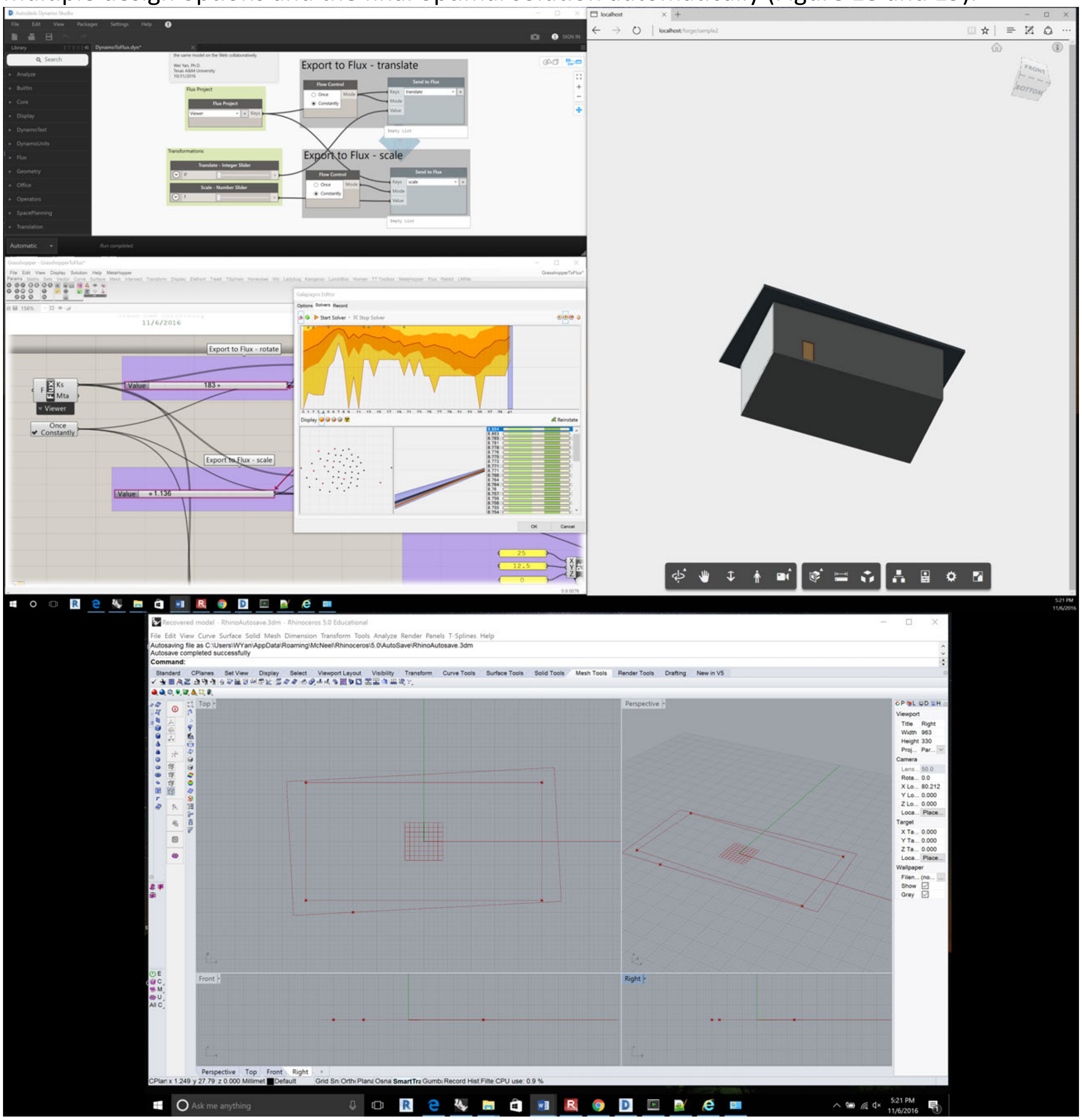

Figure 18. Running the optimization workflow. Top: the resulting optimal building design after running the process for 41 generations. The Viewer shows the resulting rotation angle and scale

of the roof. The roof can cover the front area of the door. Bottom: the optimal result is displayed in Rhinoceros: the building is covered by the roof, the door's front area (location point) is also covered by the roof, and the roof area (scale) is the smallest among all the valid solutions.

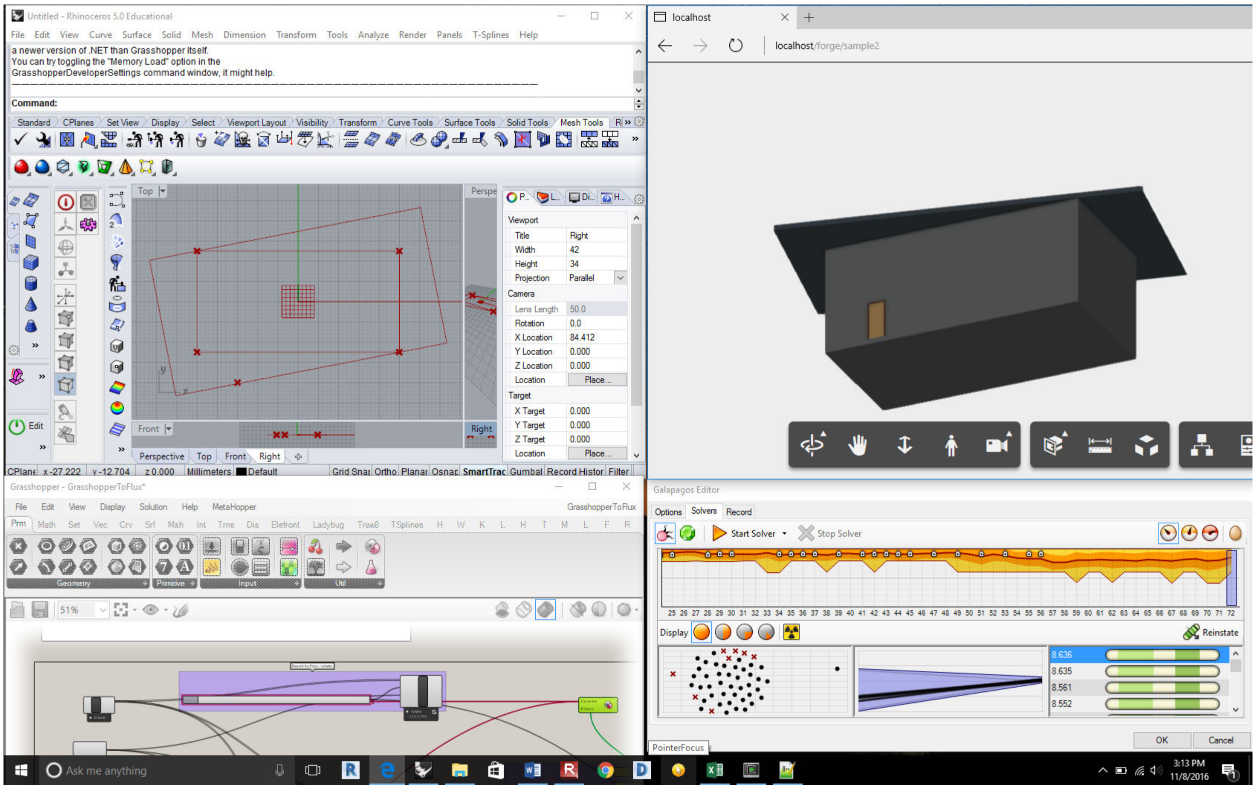

Figure 19. The door front point is set 5 ft farther away from the door. The results of the new optimization are shown in the figure.

A video demonstrating the running process of optimization can be seen from the following link: http://mediamatrix.tamu.edu/download_published_file.php?published_file_id=564452

Note that the changing of building's design options can be seen on the Web in real time by multiple users, who will be informed about the optimization process and may eventually contribute to the optimization process from different perspectives.

3. Conclusions and Future Work

The final prototype integrating Autodesk Viewer, visual programming tools Dynamo and Grasshopper, the data interoperability platform Flux, and Genetic Algorithm, was developed by the author using JavaScript programming with Viewer API and Extension as well as other mentioned tools. The prototype can run successfully demonstrating the capability of web-based collaborative parametric design and optimization with visual programming and BIM models. Future work includes: (1) extending the Viewer's model editing functions to enable more comprehensive parametric modeling capability; (2) testing the applications of web-based collaborative parametric design and optimization; (3) investigating distributed optimization processes utilizing the developed framework. The author expects that the web-based, collaborative parametric design method utilizing different modeling and visual programming tools will significantly enhance the design and optimization process towards participatory and sustainable design.

Acknowledgements:
This research project is funded by the Autodesk Forge Research Grant 2016. The support from the Autodesk Forge team, Ms. Natalia Polikarpova, Dr. Mikako Harada, Dr. Mohammad Rahamani Asl (project consultant), and Flux Factory, Inc. is greatly appreciated.