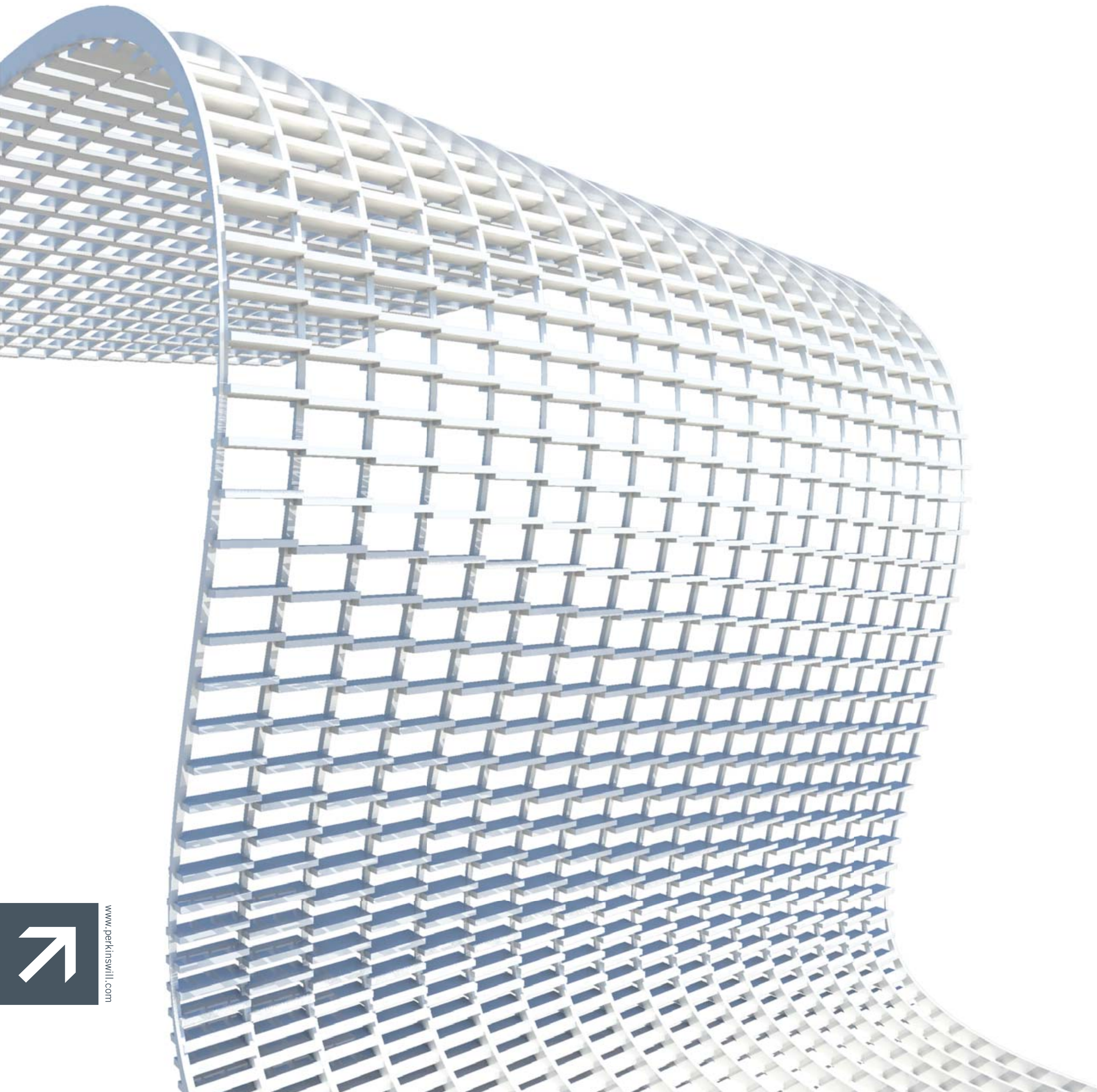
PERKINS+WILL

Research Journal

+7
2011 / VOL 03.01



www.perkinswill.com

03.
PARAMETRIC CONTROL OF BIM ELEMENTS FOR SUSTAINABLE DESIGN IN REVIT: *Linking Design and Analytical Software Applications through Customization*

Ajla Aksamija, PhD, LEED AP BD+C, CDT, *ajla.aksamija@perkinswill.com*
Mario Guttman, AIA, LEED AP, *mario.guttman@perkinswill.com*
Hari Priya Rangarajan, LEED AP BD+C, *haripriya.rangarajan@perkinswill.com*
Tim Meador, LEED AP, *tim.meador@perkinswill.com*

ABSTRACT
Using analytic data as a driver to control the geometry of BIM elements is currently a promising method for parametric creation of design elements such as sun shades, which respond to environmental constraints such as incident solar radiation or solar angles. This can be done qualitatively, but evaluating multiple options with many variables is time consuming. A preferred method is to use analytical data coming from applications such as Ecotect to parametrically control BIM families. This article reviews customization of the Autodesk Revit BIM authoring software to allow for data exchange between BIM and analytical applications (Revit and Ecotect), where analytic data is used to control the geometry of Revit families. The article first discusses concepts of solar radiation and relationships to optimum design of shading devices and previous parametric modeling work done in other software applications. Then, development of a custom plug-in for Revit that allows import of numeric data and parametric control of Revit families based on these values is discussed. Also, relationships to Ecotect and data exchange between these different software applications are discussed, followed by a case study.

KEYWORDS: BIM, parametric design, sustainability, analysis, software customization

1.0 INTRODUCTION
Recent developments in computational design tools are providing methods for improved design practices. Enhanced design representations, energy and thermal simulations and improved collaboration using digital media are some of the benefits of advanced computational tools. Building Information Modeling (BIM) is currently one of the major paradigm shifts in the building industry where the primary elements of change are:

- Representation of building elements as data-rich 3D objects, rather than as combination of 2D orthogonal views and written documents.
- Use of an interdisciplinary, comprehensive, building model as the source for derived "views", rather than a collection of "drawings" that is used to infer a 3D design.

A BIM provides a common database of information about a building including its geometry and attributes. It is an integrated, comprehensive building model that stores the information contained in traditional building documents such as drawings, specification and construction details as well as additional 3D information and metadata in a centralized or distributed database. The goal of BIM is to provide a common structure for information sharing that can be used by all agents in the design process and construction. It virtually simulates design and construction and provides groundwork for collaborative design since all the relevant information such as spatial organization, building components, building systems (mechanical, electrical, plumbing, HVAC) can be incorporated into building descriptions.

Visualization of design in three-dimensional space is one of the advantages of BIM; however, it is not the only capability and the integrative nature of contents must be emphasized. A BIM can also be used for simulations, building performance predictions and environmental analysis where the data contained in the BIM is used for daylight studies, energy analysis and solar access studies.

Typical workflow and data exchange between BIM and environmental analysis applications requires export of model geometry from BIM to analysis applications. Numerous examples of this process are available[1,2]. Best practices for data exchange between BIM and environmental analysis software depend on the analysis objectives and what type of information/data is needed. For example, for determination of building massing that minimizes solar exposure or incident solar exposure on the facade, data exchange through DXF file format is adequate. For these types of studies, geometric properties of the building massing or component under analysis (for example, part of the façade with shading devices) are sufficient. These basic parameters can be embedded in the model from the earliest stages of the design process and can be used for investigation of different design options through environmental analysis.

For other types of studies such as daylight or thermal analysis, enriched information about interior spatial organization, material properties and properties of shading surfaces is needed. Therefore, information stored in "design" BIM needs to be exported as "analysis" BIM. For example, Autodesk Ecotect® analysis software is designed to be used during the early stages of the design process and can be effectively used for a variety of analytical functions such as shadow analysis, shading, solar exposure studies, lighting and daylight studies. Data exchange between BIM and analytical software can be performed through Green Building XML (gbXML) schema, a computer language specifically developed to facilitate transfer of building properties stored in BIM to analysis tools.

Currently, data exchange between BIM and analytical software is relatively easy to accomplish. It is very challenging to import analysis results back into the BIM and control geometry of its elements based on the results. The objective of this research was to investigate the functionalities of a custom-built plug-in for the Autodesk Revit® platform that allows import of analytical results such as solar radiation striking a surface, into the BIM model. It enables importing of data and parametric control of Revit families based on the numeric values contained in the imported data. It was tested in relation to building façade design, specifically focusing on optimizing design of shading devices using solar radiation data obtained from Ecotect analysis software. The underlying drive for the research was to ease the information exchange between BIM design and analysis applications, specifically focusing on effective use of real analytic data for parametrical control of model geometry in Revit.

In this article, first we define parametric design as a rule-based design method where design models can be manipulated based on certain constraints. We also discuss the concepts of insolation and design rules for the optimal design of shading elements since the driver for this research has been to use parametric tools for design of sustainable elements. Then, we focus on the description of earlier work in non-BIM software applications where we have used design rules for parametric design of shading devices, based on the building's location and latitude. Following this discussion, we focus on the customization of Revit and development of a plug-in that allows users to import analytic data into Revit and to parametrically control Revit families based on the numeric values. Several test cases are shown, illustrating this process in detail where solar angles or solar radiation data (coming from analytic applications) have been used to control positioning or geometry of shading devices and curtain wall panels.

2.0 PARAMETRIC DESIGN AND SUSTAINABLE ARCHITECTURE

What exactly constitutes parametric modeling? These processes and tools are relatively new to the architectural community and are based on the concept of rules, constraints, features and associations between parameters and objects in the model such as geometry. The rules and constraints, usually consisting of mathematical formulas, data values or numbers can be used to control the properties of the model or an object in a model such as geometry, shape or size. The underlying driver for parametric design is to be able to quickly adapt the characteristics of a model component based on a certain rule without having to recreate the entire model for each design iteration. The rules, or numeric values, may represent structural loads, environmental data (such as solar radiation, solar angles, wind velocity), or simply a change in dimensions.

The benefits of parametric tools in practice have been acclaimed while also acknowledged as increasing in complexity and time required for certain design tasks[3]. For example, there are case studies where parametric design methods have been used to determine building geometry and curvature of the cladding design for stadium buildings[4]. Other examples include parametric generation of tall building forms[5]. Computational tools such as Maya, Rhino and Grasshopper, CATIA, Solid Works, Inventor and Bentley's Generative Components are examples of platforms that allow parametric control of model geometry based on rules and constraints. There are also examples of short algorithms and code that can be used for parametric control of model geometry[6]. However, the purpose of this article is not to review capabilities of these different software platforms or different programming methods. Rather, the objective is to discuss parametric design in relation to sustainability, particularly focusing on building envelope design and reduction of solar radiation and the use of custom tools for parametric control of BIM elements.

2.1 Concepts of Insolation

Building energy consumption is highly dependent on location, climatic characteristics and orientation relative to solar exposure. Current trends are to design and construct building facades as highly glazed envelopes that offer great potential for daylight; however, solar heat gain must be controlled in order to create a habitable internal environment and reduce building energy consumption. Horizontal and vertical shading devices such as overhangs, fins and louvers can be used to reduce solar heat gain for the internal environment. Moreover, shading devices can include integrated photovoltaic systems creating relationships between desired daylight, energy consumption, available surface area and available solar radiation that need to be explored.

Solar *radiation* is the most significant contributor to heat gain associated with building facades. The prediction of average solar insolation for any day, month, season or year is needed in order to estimate the cooling load arising from radiation received on walls or transmission through windows. Solar *insolation* refers to the total amount of cumulative incident solar radiation on a point or surface over a specified period and is expressed in Btu/ft^2 (kWh/m^2) units. Understanding the intensity of solar insolation on different geometric shapes and orientations is important, especially in relation to building façade design. The position of the sun determines the intensity of solar radiation striking on various surfaces of a building. The amount of solar radiation striking a given surface of a building, wall or roof changes constantly as a result of the changing position of the sun. The diurnal and annual patterns of the sun's motion in the sky depend on the latitude of the location in question (distance from the equator) as well as the surface inclination seen in figure 1.

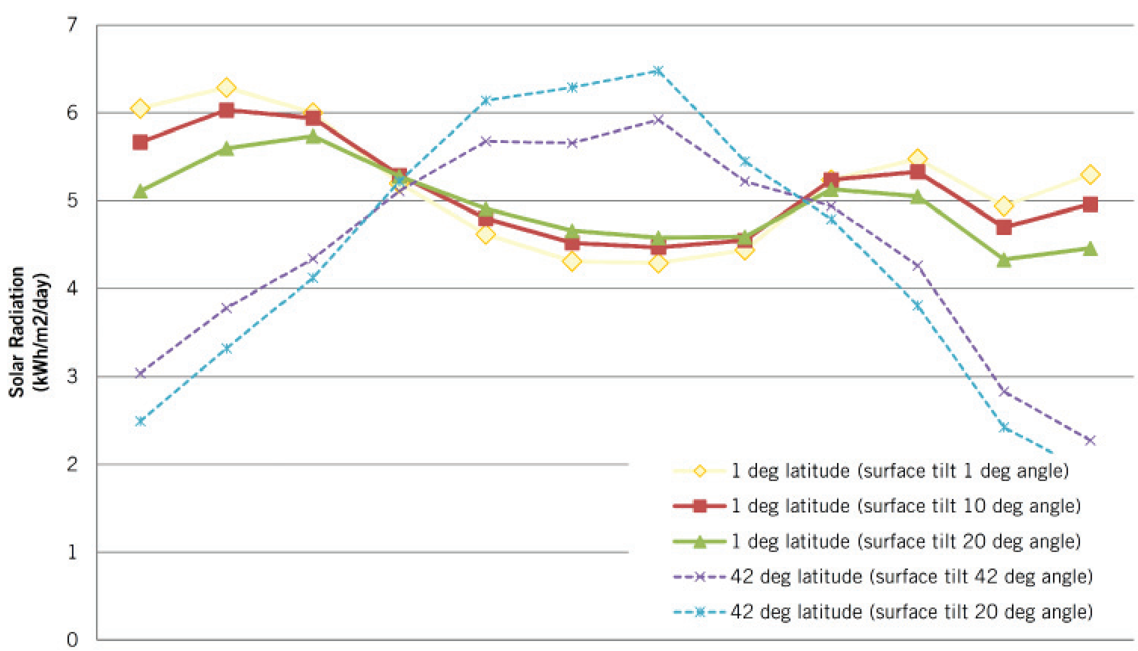


Figure 1: Annual average solar radiation in relation to latitude and angle tilt of the surface.

2.2 Selection of Shading Devices

Selection of shading devices depends on building orientation since each side of the building receives different amounts of solar radiation. Generally, horizontal devices should be used for south façades (north for southern hemisphere) since these types reduce solar heat gain throughout the year. Vertical devices such as fins should be used on east and west facades and preferably should be able to rotate depending on the daily sun path. Shading of the south facades should respond to seasonal, while east and west façades should respond to daily changes in the position of the sun.

If tilted shading devices are used, the optimal angle is generally equal to latitude for fixed horizontal elements. If horizontal shading devices can be rotated, seasonal changes can be accounted for by adjusting the angle depending on the location (latitude) and different seasons. For fixed devices, selecting preferred season or averaging values for different seasons can optimize shading efficiency.

Overhang design that incorporates horizontal shading devices can be sized according to the building location and data obtained from the sun path diagram (solar azimuth and altitude). Dimensions are dependent on horizontal and vertical shading angles.

Rotating angle of vertical shading devices should respond to daily and monthly sun path as well as building orientation angle. These can be expressed in relation to sun azimuth or horizontal shading angle. Deviation from true north can be accounted for by subtracting relative orientation angle. The methodology and steps to optimize design of shading elements include:

- Determination of the overheated period, based on building's location and climate, to select months and periods when shading is needed.
- Determination of the critical solar angles for the design.
- Determination of the physical properties of the shading devices such as type (horizontal or vertical) depending on the building's orientation, sun tracking capabilities and dimensions.

2.3 Shading Geometry, Architectural Components and Parametric Design

The principles of insolation and sun shading, described in the previous section, define an architectural problem as they are applied to a building design. The functional requirements of shading devices also offer aesthetic opportunities when combined with other design objectives creating interesting patterns and unique façade solutions. Projections and recesses, varied size and scale of louvers, and degrees of light and shade, can be combined in different permutations to create architectural interest. As seen in figure 2, horizontal fins are used on the south facade with a 45 degree angle maintained between the varying depths of the fins. Figure 3 illustrates vertical fins on the west and east façades, screening balconies and living quarters and offering visual privacy to the units facing each other. Figure 4a has a 14.7 ft (4.5 meter) modular screen that wraps the building on the southern façade and ties the different programmatic elements together. The fins are rotated based on the latitude. Figure 4b shows the inclined fins with different positions for summer, winter and spring.

In our previous work the use of these elements was studied in projects through the use of MEL (Maya Embedded Language) scripting in Maya software. These studies led to an understanding of the parameters that control an individual fin and how fins can be populated within different geometries. Exploration and positioning of vertical and horizontal shading devices using the

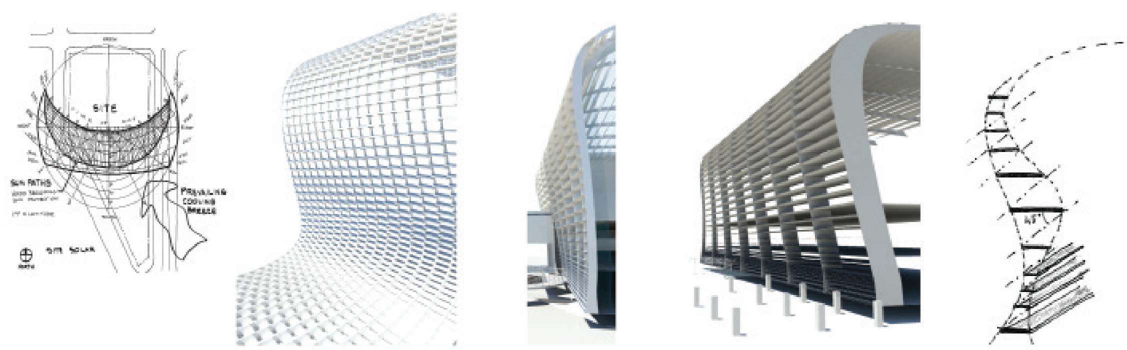


Figure 2: Shading elements as major architectural component (Community Center, Texas).

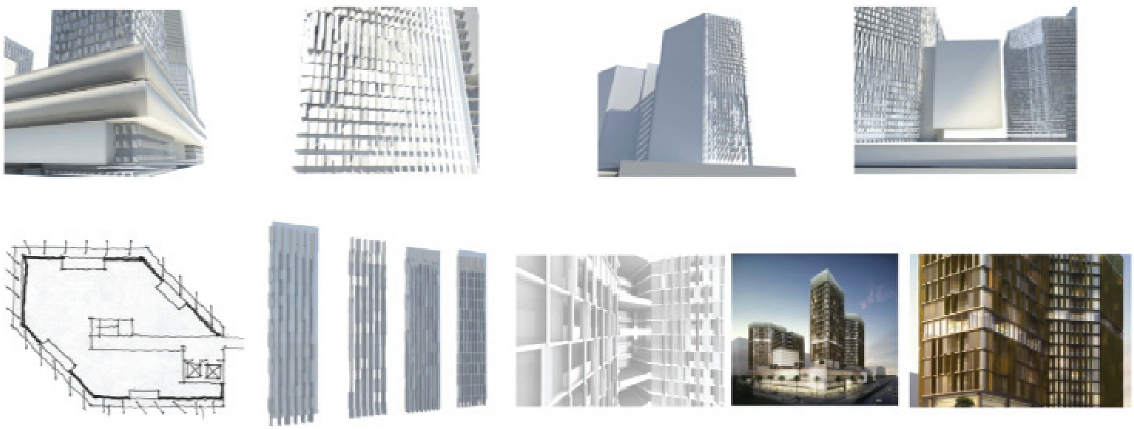


Figure 3: Vertical fins (Residential mixed-use tower, Abu Dhabi).

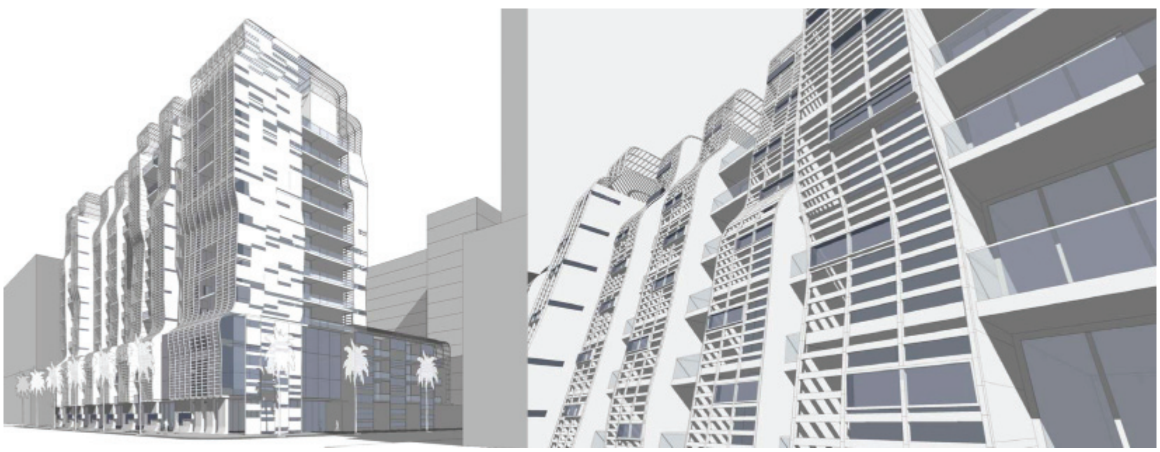


Figure 4a: Horizontal fins as screens (Residential mixed use building, Abu Dhabi).

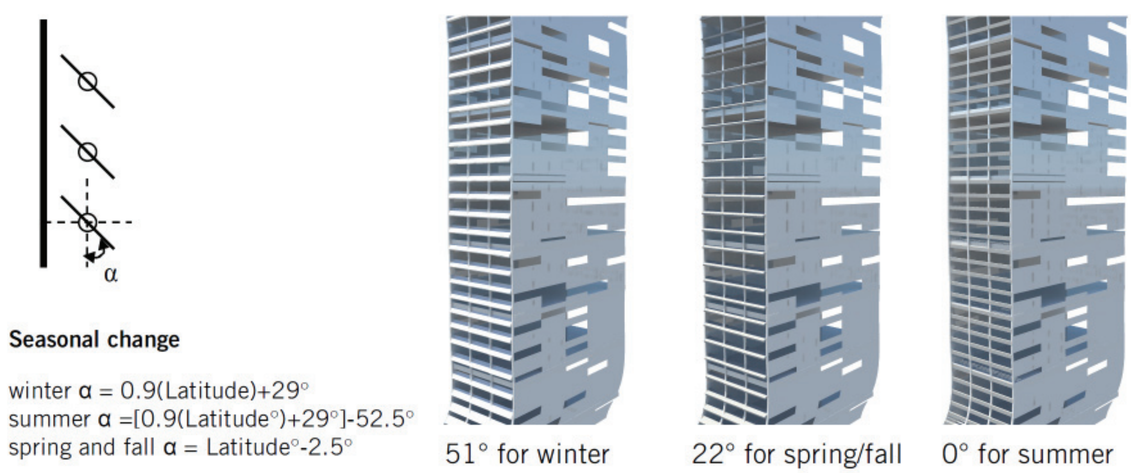
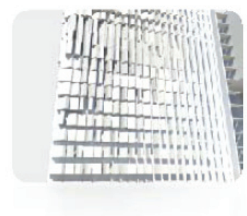


Figure 4b: Rotation of louvers that follow seasonal changes and parametric control.

script was possible on a variety of surfaces by simple parametric changes. We have used solar angles and design rules discussed in section 2.2 to size and position shading devices parametrically based on building location, orientation and solar angles. For example, figure 4b shows how parametric control of horizontal fins that respond to seasonal changes could be optimized where rotational angle is used in the script to change the angle of the fins. Figure 5 indicates the sequence of how the fins are positioned, sized, rotated and populated on a surface.

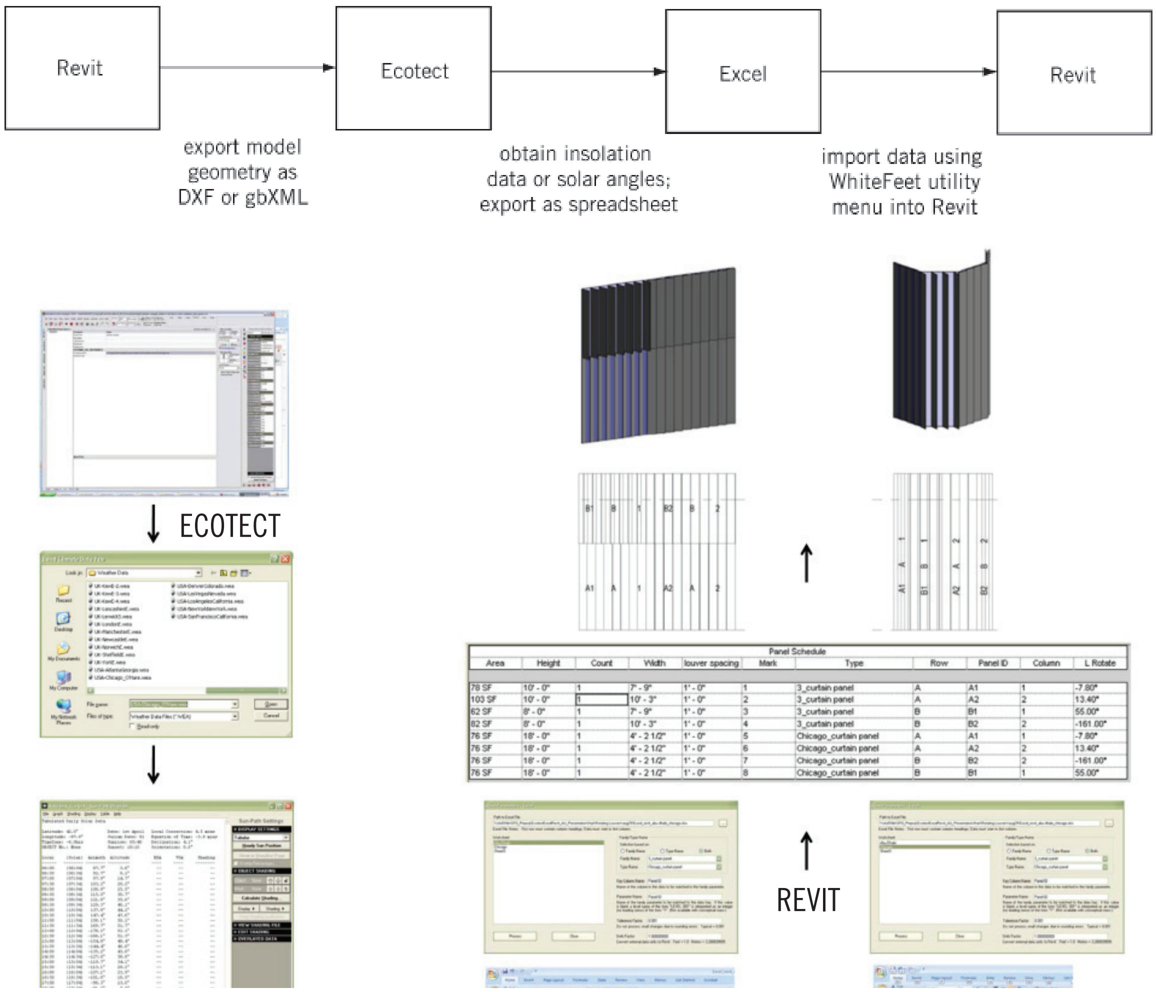
These explorations have indicated that parametric control of shading elements offers an improved design method for the design of sustainable facades, but the next step was to determine a similar approach for the Revit platform since it is the primary tool used for architectural documentation (i.e. its geometry is in context with other aspects of design) and to take the advantage of parametric functionality of Revit families. Through customization of Revit platform and testing and evaluation of different options, we were able to work out a method where custom plug-in can be used to import

```
////////////////////////////////////////////////////////////////////////////////
// horizontal fins
// winter- 51.5 degree
// spring- 21.5 degree
// summer- 10 degrees as 0 means- no light
////////////////////////////////////////////////////////////////////////////////
string $surf[] = `ls -sl -fl`;

int $divx = 75;

int $divy = 3;

for($i=0;$i<($divx);$i++){

for($j=0;$j<$divy;$j++){

float $u = float($i)/$divx;

float $v = float($j)/$divy ;

float $p[] = `pointOnSurface -u $u -v $v -p $surf`;

float $n[] = `pointOnSurface -u $u -v $v -no $surf`;

polyCube -w (.5)  -h (.1) -d (1.5) -sx 2 -sy 2 -sz 1 -n fin";

move $p[0] $p[1] $p[2];

rotate 0 (( -90)) 0 ;

currentTime 1 ;

}}
////////////////////////////////////////////////////////////////////////////////
```



```
////////////////////////////////////////////////////////////////////////////////
//Vertical fins
// Abu dhabi - longitude- 54deg 22 min
// Abu dhabi - lattitude - 24 deg  28 min

////////////////////////////////////////////////////////////////////////////////
string $surf[] = `ls -sl -fl`;

int $divx = 7;

int $divy = 8;

for($i=0;$i<($divx);$i++){

for($j=0;$j<$divy;$j++){

float $u = float($i)/$divx;

float $v = float($j)/$divy ;

float $p[] = `pointOnSurface -u $u -v $v -p $surf`;

float $n[] = `pointOnSurface -u $u -v $v -no $surf`;

polyCube -w (1.5)  -h (3) -d (.2) -sx 2 -sy 2 -sz 1 -n " fin;

move $p[0] $p[1] $p[2];

rotate 0 (-135.81) 0 ;

currentTime 1 ;

}}
////////////////////////////////////////////////////////////////////////////////
```

Figure 5: Parametric design and control of shading elements in Maya.



Panel Schedule										
Area	Height	Count	Width	louver spacing	Mark	Type	Row	Panel ID	Column	L Rotate
78 SF	10' - 0"	1	7' - 9"	1' - 0"	1	3_curtain panel	A	A1	1	-7.80°
103 SF	10' - 0"	1	10' - 3"	1' - 0"	2	3_curtain panel	A	A2	2	13.40°
62 SF	8' - 0"	1	7' - 9"	1' - 0"	3	3_curtain panel	B	B1	1	55.00°
82 SF	8' - 0"	1	10' - 3"	1' - 0"	4	3_curtain panel	B	B2	2	-161.00°
76 SF	18' - 0"	1	4' - 2 1/2"	1' - 0"	5	Chicago_curtain panel	A	A1	1	-7.80°
76 SF	18' - 0"	1	4' - 2 1/2"	1' - 0"	6	Chicago_curtain panel	A	A2	2	13.40°
76 SF	18' - 0"	1	4' - 2 1/2"	1' - 0"	7	Chicago_curtain panel	B	B2	2	-161.00°
76 SF	18' - 0"	1	4' - 2 1/2"	1' - 0"	8	Chicago_curtain panel	B	B1	1	55.00°

Figure 6: Process diagram showing the data transfer between different applications for parametric control of BIM elements.

analytical data into Revit to control the geometry of shading devices. We have used Excel spreadsheets to hold the analytic data coming from Ecotect analysis software and supply numeric values to Revit for parametric control through a custom tool. The process diagram is shown in figure 6, indicating how the data exchange between Revit, Ecotect and Excel is carried out. Components of the tool are discussed in detail in the next section as well as connectivity mechanisms between Revit and Excel.

3.0 CUSTOMIZATION OF REVIT

3.1 Use of a Custom Programming Approach

The team considered alternative approaches for providing the parametric functionality. Although commercial applications that provide much of the required capability are available and other programming environments could have been used to develop a solution, an application based on the Revit Application Programming Interface (API) linked to Microsoft Excel was selected.

The custom programming strategy has several advantages over so-called "user friendly" applications such as Grasshopper, that provide a graphical user interface and generate code in the background.

Although the graphical interfaces are "easy" to use for the production of quick, dramatic results, they are not as explicit about the imbedded decision process that lead to the final form. Computer programming code de-

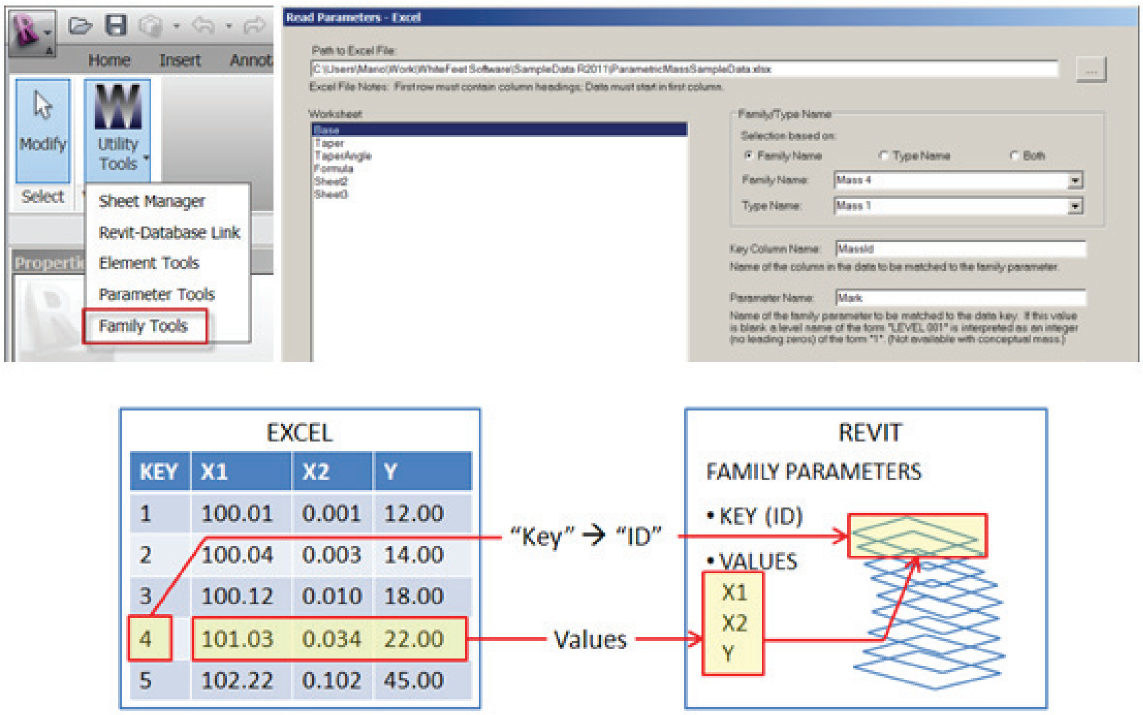


Figure 7: Relationship between Excel data and Revit family parameters.

veloped in a text-based language such as C# (which was used in this project) is self-documenting. In other words, a programmer can read the code and exactly understand that rational basis of the results and manipulate them to precise values. This is especially significant where the parametric objectives are complex and based on precise requirements such as the relationship between the sun shading elements and the solar calculations.

The specific software selections were based not only on the project's needs, but in accordance with Perkins+Will's preferred development platforms. By using Revit as the BIM component, Excel for data manipulation and C# in Visual Studio as a programming environment, the team is advancing the skills and developing code modules that will have broader applications.

The evolving nature of the teams' design solutions necessitated a software programming strategy that was flexible and supported experimentation. A broadly applicable toolset (given the working name "WhiteFeet") that includes a standardized Revit menu system and modular code blocks was developed. This solution extends significantly beyond the needs of the current project, but was easily extensible to include all of the project needs.

It was especially important to manage the user input settings that were needed for each study. These include arcane values such as family names, parameter names and file paths. The solution includes a system of initialization files that restore the user interface to named configurations in conjunction with settings that were stored as data in the Excel worksheets (tabs), so that each experiment is precisely defined and can be reproduced.

3.2 Data Strategy for Manipulating Revit Families

The basic framework of the solution is based on three subcomponents:

- The Revit geometry is created by using a small number of Revit families that are placed in many instances. The families include instance-level parametric dimensions so that individual variations among many instances result in an overall form change.

Table 1: Data connectivity options between Revit and Excel.

Option	Description	Reasons to Use / Not use
MS Jet engine	Driver that can read Excel and Access files directly.	• Very straightforward code. • Cannot be used with 64-bit OS.
MS Interop	Starts a session of Excel or Access that runs at the same time as Revit.	• Complex programming to start/stop. • Fragile at run time.
MS SQL Server	True "industry strength" database.	• Very robust/scalable/powerful. • Difficult for users to create instances.
Delimited Text	Plain ASCII file with a <tab> or other delimiter to separate fields.	• Messy code. • Users cannot use database/spreadsheet.

- The analytical and computational basis for the model geometry is stored in a database system external to Revit.

- The correspondence between the family instances and their database records is maintained by a one-to-one "key" relationship as seen in figure 7. This requires conventions in the way the families are created and the quantitative data is stored, so that they can be maintained and enforced by the software.

Several database options were considered. Although a true database such as Microsoft Access or Microsoft SQL Server would have been preferable from a programming standpoint, Microsoft Excel was selected because of its familiarity and ease of usability to the entire team. This necessitated some special programming and imposed some strict requirements on how the Excel worksheets were formatted, but did not pose any significant problems.

The protocol used for the communication between Revit and Excel is somewhat problematic since Microsoft (who writes drivers and other tools for working with Excel) has provided a changing and incomplete set of options. Microsoft would like to see SQL Server used for this kind of activity, but this is not how SQL Server and Excel are used within Perkins+Will. In particular, the "Jet/ACE" drivers that would be best suited cannot be used with the combinations of 32-bit and 64-bit software that is the Perkins+Will standard. For these reasons, Excel was run as a parallel application to Revit and accessed using COM through the "Interop" interface. This solution is no longer usable with the 2010 version of Excel so an alternative strategy has been developed for work since this project.

The "key" relationships are based on required conventions:

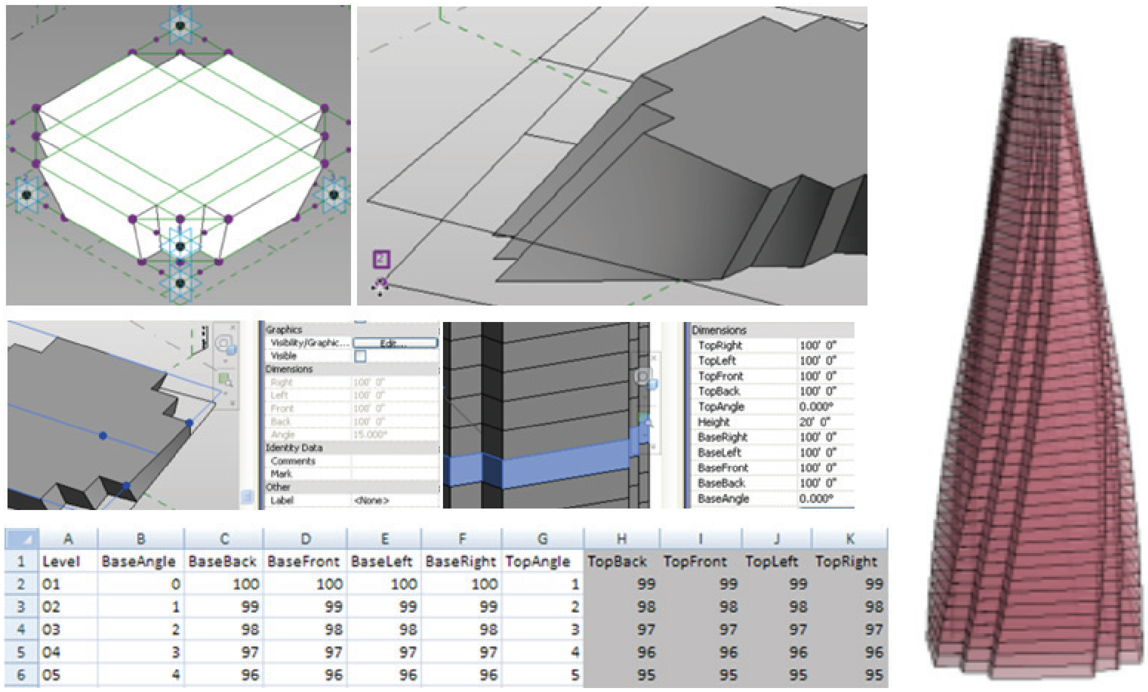
- Each family instance includes an instance parameter that identifies it. This value (string or integer) must be present and must be unique; however, there are no other limitations on the name of the parameter or the values.

- Each geometric study (model) is associated with a single worksheet (tab) in Excel. Multiple models (versions of the geometry) are maintained by collecting several such worksheets into a single Excel workbook (an Excel file.) Each worksheet must have a column designated as the key value with data such that there is an exact one-to-one correspondence between the rows of the worksheet and the instances of the family.

The instance parameters of the Revit family, which control its geometry, each correspond to a column of the Excel worksheet. These values must be present, but there is no requirement for uniqueness. Two general strategies emerged for doing this:

- Option 1: Using several parameters in the Revit family, where each parameter directly controls a single dimension and updating all of them for each family instance.

- Option 2: Using a single parameter (often named "factor") that was used as a factor in several calculated parameters within the family, which then controlled the dimensions. Typically this factor parameter was normalized so that it held a value between zero and one. This served to allow work in Revit and in the data to proceed independently without having to define the allowable range of values.



	A	B	C	D	E	F	G	H	I	J	K
1	Level	BaseAngle	BaseBack	BaseFront	BaseLeft	BaseRight	TopAngle	TopBack	TopFront	TopLeft	TopRight
2	01	0	100	100	100	100	1	99	99	99	99
3	02	1	99	99	99	99	2	98	98	98	98
4	03	2	98	98	98	98	3	97	97	97	97
5	04	3	97	97	97	97	4	96	96	96	96
6	05	4	96	96	96	96	5	95	95	95	95

Figure 8: Example of Revit family instance (tower floor) and Excel data used to derive of the twisting geometry of the tower.

3.3 Solution Workflows

The basic workflow occurred in three steps:

- Families were developed and placed in Revit.

- An Excel file was created. In some cases the Excel data was imported from another program, specifically Ecotect in this project for obtaining solar radiation values. In other cases, all of the data manipulation was done in Excel.

- The WhiteFeet program was launched from the Revit Add-ins menu. The name of the key parameter, the path to the Excel file and the name of worksheet and other settings were filled in. The synchronization process was launched from a command button.

The effect of these steps was to update the parameter values of all of the family instances. After the WhiteFeet program exited, Revit then regenerated the families, resulting in the new form.

The initial development of the program was for use in creating complex building forms. These typically were tall towers that included changes in dimension and twisting of the plan geometry at each floor as seen in figure 8. In these studies, a family instance represented a floor. The geometry of the instance was independently controlled at the top and the bottom of the floor to give an overall smooth effect. Generic solids, mass objects and adaptive components were used successfully in this way.

Beginning with the sun shading explorations, the program was adapted for use with families that constituted repeated instances of wall elements. In some cases these were simply arrayed families and in others they were panels imbedded in a Revit curtain wall family. The behavior was the same in both cases because Revit allows the addressing of the imbedded families as if they were placed separately.

To make it easier to manage the families, a convention for defining their identifiers was used. In this, the rows were assigned letter names, the columns integer names and the cells were named by combining these two values. This was easily accomplished by creating three separate parameters, naming the rows and columns in elevation views where they could be selected as groups and using another WhiteFeet tool to concatenate the values to the ID parameter.

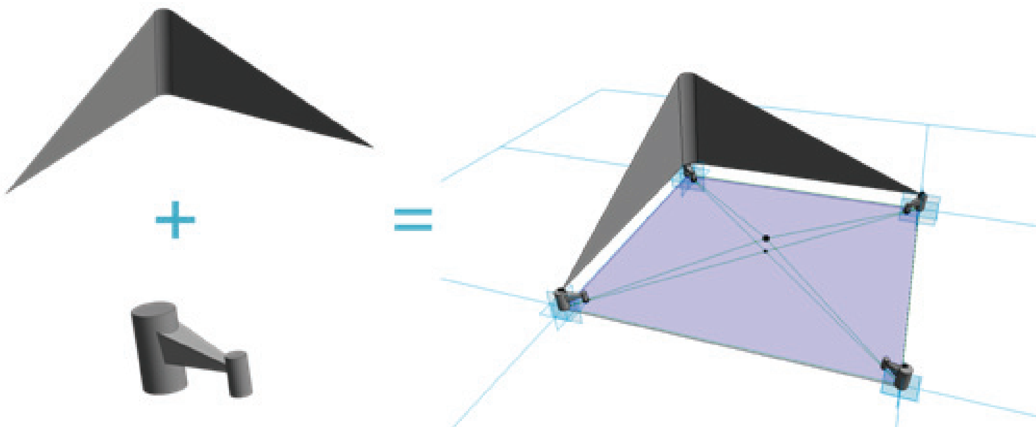


Figure 9: Example of nested families.

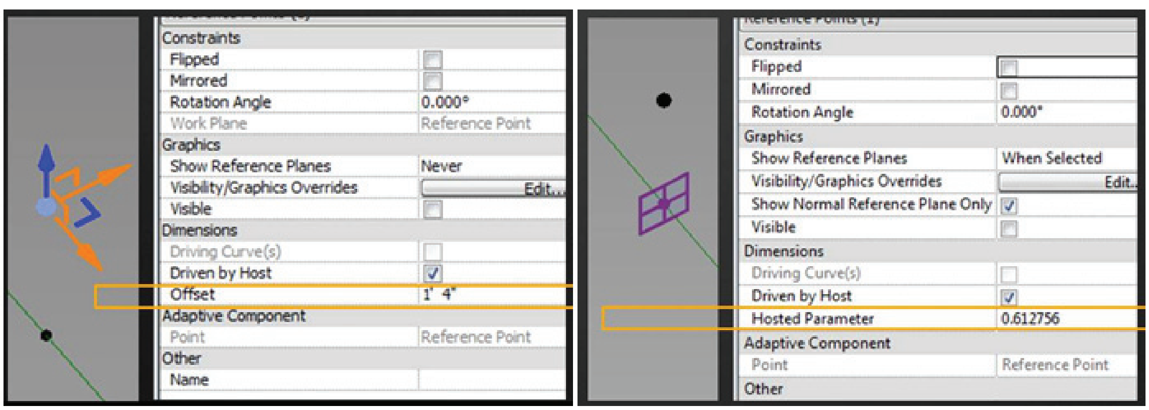


Figure 10: "Offset" and "Hosted" parameters in Revit.

As more complex surfaces were developed, the same process was applied to mesh surfaces containing adaptive component cells in the Revit Conceptual Design Environment. Although the naming of the cells followed a similar strategy to the planar curtain walls, the irregularities in the patterns necessitated a refined strategy for identifying the separate rows and columns. A special WhiteFeet tool was developed for this purpose. In some cases these needed to be assigned arbitrarily.

4.0 CASE STUDY: COMPUTATIONAL REVIT FAMILIES AND PARAMETRIC CONTROL

Revit and the WhiteFeet utility menu can be used to create parametric dynamic shading systems driven by Ecotect solar incidence data. A number of curtain wall and different shading method case studies were generated. As an example, in the following case study we illustrate how a curtain panel pattern family can be used to panelize free-form surfaces with dynamic geometry. For modifying curtain wall pattern families, the suggested method is to nest complex geometries in separate families as seen in figure 9. This allows each component of the curtain panel to be tested independently and properly "flexed" or tested in Revit with a range of possible values that would come from the solar radiation data.

There are two different types of behavior in the hosted geometry within the conceptual design environment. When hosting points on other points, the created point will have a parameter named "Offset" shown in figure 10. This specifies the point's offset in the "Z" direction from the host point. The second hosted behavior is the "Hosted" parameter. This results when a point is hosted on a reference line. This parameter can range from 0 to 1 and controls where on the hosted line the point falls. We call it a "Factor" parameter as seen in figure 11. This benefits the Revit family in several aspects. First, it

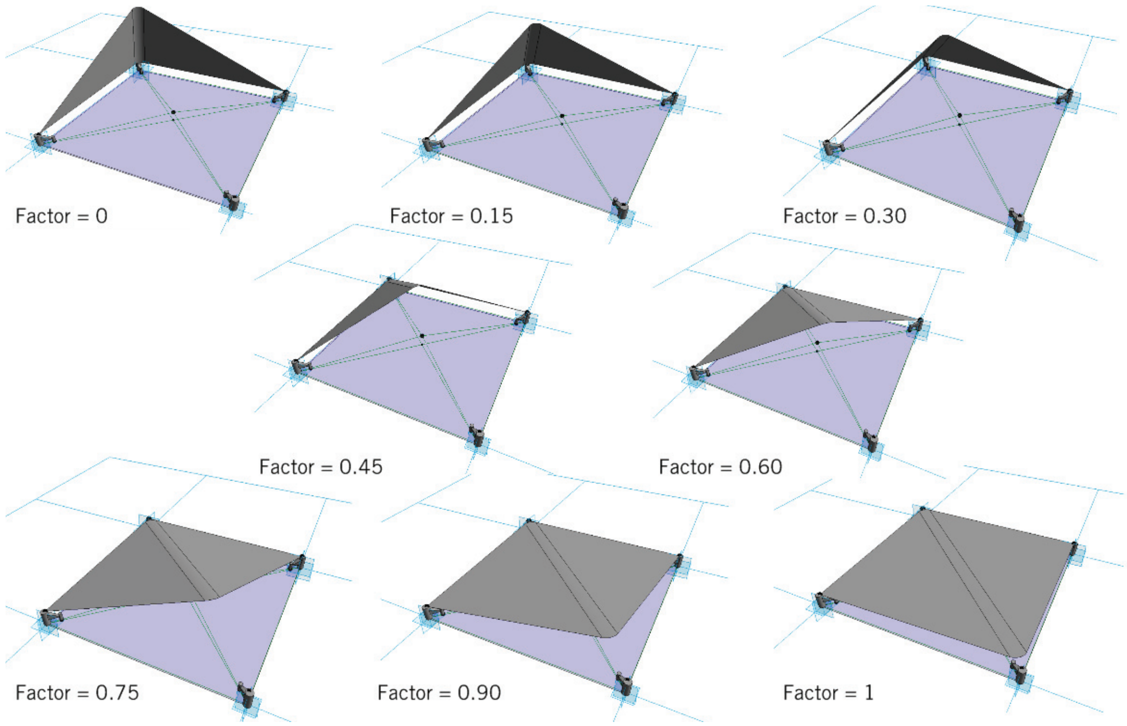
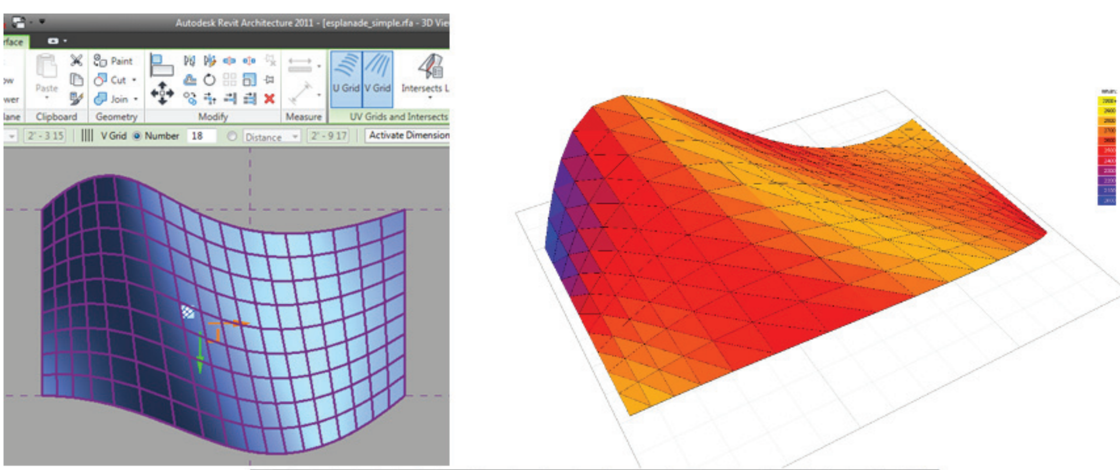
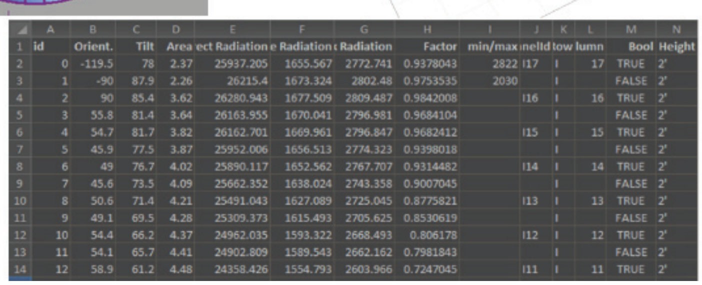


Figure 11: “Factor” parameter and adaptation of Revit family based on the value of the parameter.





	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	Orient.	Tilt	Area	ect Radiation	e Radiation	Radiation	Factor	min/max	nelId	tow	lumn	Bool	Height
2	0	-119.5	78	2.37	25937.205	1655.567	2772.741	0.9378043	2822	I17	I	17	TRUE	2'
3	1	-90	87.9	2.26	26215.4	1673.324	2802.48	0.9753535	2030		I		FALSE	2'
4	2	90	85.4	3.62	26280.943	1677.509	2809.487	0.9842008		I16	I	16	TRUE	2'
5	3	55.8	81.4	3.64	26163.955	1670.041	2796.981	0.9684104			I		FALSE	2'
6	4	54.7	81.7	3.82	26162.701	1669.961	2796.847	0.9682412		I15	I	15	TRUE	2'
7	5	45.9	77.5	3.87	25952.006	1656.513	2774.323	0.9398018			I		FALSE	2'
8	6	49	76.7	4.02	25890.117	1652.562	2767.707	0.9314482		I14	I	14	TRUE	2'
9	7	45.6	73.5	4.09	25662.352	1638.024	2743.358	0.9007045			I		FALSE	2'
10	8	50.6	71.4	4.21	25491.043	1627.089	2725.045	0.8775821		I13	I	13	TRUE	2'
11	9	49.1	69.5	4.28	25309.373	1615.493	2705.625	0.8530619			I		FALSE	2'
12	10	54.4	66.2	4.37	24962.035	1593.322	2668.493	0.806178		I12	I	12	TRUE	2'
13	11	54.1	65.7	4.41	24902.809	1589.543	2662.162	0.7981843			I		FALSE	2'
14	12	58.9	61.2	4.48	24358.426	1554.793	2603.966	0.7247045		I11	I	11	TRUE	2'

Figure 12: Example of curved surface in Revit, solar radiation analytic data in Ecotect, and data values in Excel.

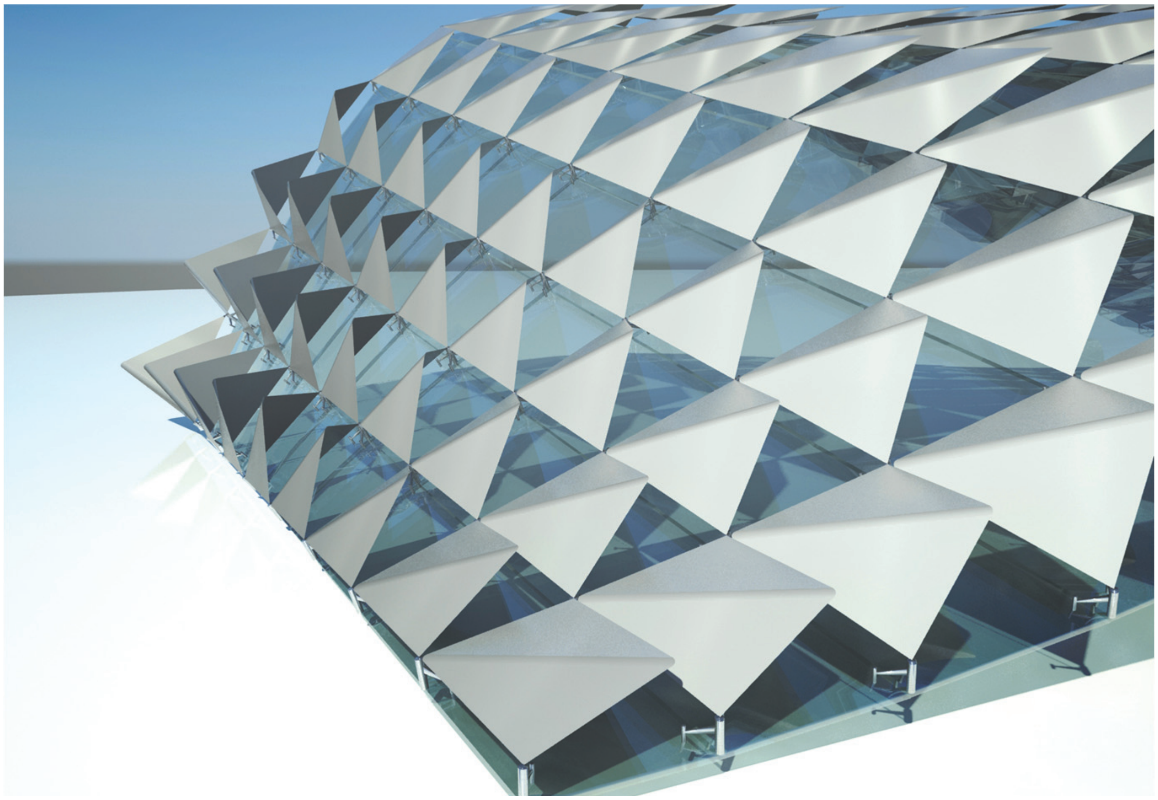


Figure 13: Example of parametric control of shading elements in Revit.

allows the family to be modular and able to be plugged into other curtain panel pattern families as an adaptive component. Secondly, it accepts normalized data from Ecotect, which allows the family to respond to varying solar incident values depending on time of day or location.

In order to align the Ecotect data with individual instances of Revit panel families, several instance parameters can be created within the family. This allows the subdivision of families to be logically ordered in order to align them with Ecotect. After creating a surface in the conceptual design environment, the surface can be subdivided into a desired number of divisions, which can then be exported into a DXF file. This geometry can be imported into Ecotect to analyze incident solar radiation and obtain solar radiation values based on building location and specific orientation of the panel. These values can be exported from Ecotect into an Excel spreadsheet as seen in figure 12. The obtained solar radiation data needs to be normalized in order to align it with the Revit panel families to fit the 0 to 1 “Factor” parameter. The method for matching values obtained from Ecotect to “PanelId” Revit parameters is by concatenating rows and columns in Excel spreadsheet. This normalized, concatenated data is imported into Revit using WhiteFeet utility menu and used to control the geometry of Revit panel families. The resultant is shown in figure 13, showing a surface where the shading elements for the curtain wall panels respond to solar radiation striking this curved surface.

5.0 CONCLUSION

Parametric design offers some advantages over traditional modeling methods, since it allows adaptation of an object through the use of rules and constraints or “parameters” to influence the object’s properties. These processes as well as parametric computational tools, are relatively new in architectural design. They enable the adaptation of model geometry based on rules or data values, eliminating the need to recreate the model for every design change. In essence, the benefits of parametric design are:

- Parametric modeling uses manipulation and adaptation of object’s properties based on rules and data values.

- Multiple design options and design iterations can be created by modifying object attributes and properties (such as dimensions or shape) without recreating the entire model.
- Analytical data developed in response to environmental constraints, or other types of logic-based control, can be used to derive geometry.

There are also limitations to parametric model design, which are:

- Parametric modeling requires use of advanced computational tools, which require investment and time.
- In some instances, this design method also requires customization of software applications for implementation.
- The logic behind the architectural design process must be understood in order to be implemented in the parametric design and it can be difficult to express in quantifiable terms in some instances.

In this work we presented relationships between parametric design and sustainability, particularly focusing on building facade design and methods to reduce solar radiation. We have discussed the concepts of insolation and the dependency of the actual values on latitude and orientation of the surface in relation to the sun's position. We have also discussed basic rules for the design of shading devices that are based on a building's orientation as well as steps for optimizing performance of shading devices. These rules have been implemented and parametrically tested on surface geometries using Maya modeling software and custom algorithm (MEL script), where positioning, sizing and typologies of shading devices are easily controlled and manipulated. These explorations showed that the parametric control of shading elements offers improved design method for the design of sustainable facades, but it was also necessary to investigate whether a similar method can be applied to Revit since this BIM platform is currently used for architectural design and documentation. We have tested a custom tool for Revit that can be used to import analytic data such as solar radiation values to control geometry of Revit families.

This process proved beneficial for determination of building forms and parametric design of elements that respond to environmental constraints and data such as insolation. Future implementation and testing of this tool and process should focus on other applications and parameters, testing values such as wind velocity and response of the building form design.

Additional Notes

A previous version of this work has been presented at the Autodesk University 2010 Conference. A recording of the presentation and associated handout material is available at: http://au.autodesk.com/?nd=class&session_id=6854.

The custom software may be downloaded from: http://www.whitefeet.com/License.htm.

Acknowledgments

Authors would like to acknowledge Richard Adams and Jeffrey Erath (Community Center, Texas); Shubhankar Sanyal, Steve Kunin and Wieren Chen (Abu Dhabi projects).

REFERENCES

[1] Aksamija, A., (2010). "Analysis and Computation: Sustainable Design in Practice", *Design Principles and Practices: An International Journal,* Vol. 4, No. 4, pp. 291-314.

[2] Aksamija, A., and Mallasi, Z., (2010). "Building Performance Predictions: How Simulations Can Improve Design Decisions", *Perkins+Will Research Journal,* Vol. 2, No. 2, pp. 7-32.

[3] Aish, R., and Woodbury, R., (2005). "Multi-Level Interaction in Parametric Design", *Lecture Notes in Computer Science,* Vol. 3638, pp. 151-162.

[4] Hudson, R., (2008). "Frameworks for Practical Parametric Design in Architecture", *Proceedings of the 26th eCAADe Conference,* Antwerpen, Belgium, pp. 847-854.

[5] Park, S. M., Elnimeiri, M., Sharpe, D. C., and Krawczyk, R. J., (2004). "Tall Building Form Generation by Parametric Design Process", *Proceedings of the CTBUH 2004 Conference,* Seoul, Korea.

[6] Woodbury, R., (2010). Elements of Parametric Design, Retrieved on 3/15/11 from http://www.designpatterns.ca/.