SOFTBIM:
AN OPEN-ENDED BUILDING INFORMATION MODEL IN DESIGN PRACTICE

ABSTRACT

In this paper we propose softBIM, a working method developed through experiences from UNStudio's architectural practice. softBIM is the adherence to BIM processes through the use of code-based interfaces and bespoke—as opposed to standard—solutions, drawing on environments that may be thought of as open-ended, such as commonly accessible CAD platforms. We present examples from practice in which our definition of the softBIM method has been used to some extent. We discuss its advantages and disadvantages in relation to its use in early project phases. The premise for this study is twofold; it has been our intention both to revise the way in which our tools are used, arriving at the back-end programming level in an attempt to increase effectiveness, as well as to devise a specific open-source-type methodology drawing on the above philosophy in a specific way inherent to the UNStudio existing software culture. The goal of this study is to propose an integrative, schematic, and open-ended model for dealing with complex assemblies of geometric and nongeometric project data, aiming to remain nonreliant on specific software packages.

Gustav Fagerström
UNStudio

Marc Hoppermann
UNStudio

Nuno Almeida
UNStudio

Martin Zangerl
UNStudio

Stefano Rocchetti
UNStudio

Ben Van Berkel
UNStudio

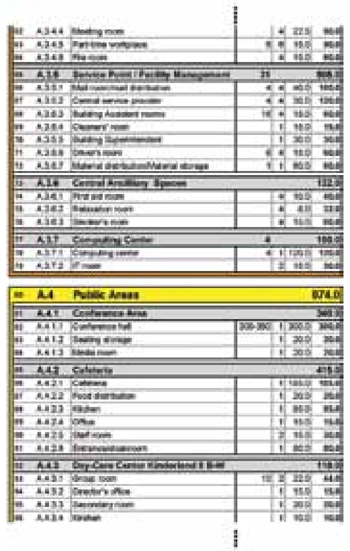

figure 1

figure 1

Programmatic categorization.

1 INTRODUCTION

1.1 Premise for BIM

BIM (building information modeling): a process revolving around virtual models embedded with data which, when shared among design team members, greatly reduces errors and improves facilities (Jernigan 2007). In this paper we focus on those subsets of BIM that we believe to be fundamental to this process:

-***The embedding of metadata within geometric information.***

-***Formalized nongeometric design drivers***; the ability to create and carry out changes to a 3D geometry model via textual input.

-***Formalized nongeometric and geometric design output***; the ability to derive text-based information from the 3D geometry model (e.g., quantity takeoffs); the ability to generate (as opposed to manually draw) two-dimensional information from a three-dimensional geometry model.

1.2 softBIM—hardBIM

Following the definition of softBIM above, its opposite is represented by off-the-shelf BIM packages commonly in use in architectural practice today, within the scope of this paper hereafter referred to as *hardBIM*. softBIM occupies an important role in our practice model, as it historically precedes the use of hardBIM and is regularly chosen as the preferred alternative to it for the reasons outlined below. Within the local setting of architectural practice and the prevailing commonly used software packages, the definitions can be seen as analogous to an open-source as opposed to a proprietary principle in the world of software development.

CAD standards can target a uniform output on a lowest common denominator basis—typically a 2D drawing (or even lower, a PDF). Current standardized BIM technologies—hardBIM—do offer possibilities to embed attributes within geometric entities, thus augmenting them with more abstract metrics unrelated to the geometry itself. These operations usually rely on a front-end approach where properties can be assigned through a series of ready-made GUI options.

Where this—for reasons of functional restrictions or otherwise—is deemed insufficient, a back-end alternative is possible. In this way the user may, via some form of code-based interface, gain control of customized attributes not covered by built-in functions.

2 METHODOLOGY—DESIGNDATA

DesignData: Through associating raw data with objects, meaning is given to the data, thereby turning it into useful information for the design process. With bidirectional links between the spreadsheet, the 3D model, and the representational 2D drawing, the softBIM approach aims to leverage this DesignData during phases where the design still remains fluid.

2.1 Ordinary Data Management

2.1.1 User-Sorted Data

In everyday work on drawings or 3D models, designers intuitively organize drawn objects into layers, groups, blocks, or colors. This is done for reasons of representation but also in order to add logic to what is drawn. Essentially, the designer is augmenting the geometry with data that can later be queried and processed. A line drawn on a layer representing window frames will later be identifiable as such. Adding that line to a block with other lines allows for the creation of building elements (such as facade systems, etc.) that would later be recognizable as such in order to be reused in the drawing. This way, through sorting geometric objects, simple intelligence can be created and objects can be correlated to one another. As this technique relies only on front-end GUI options, it is very intuitive and flexible and is therefore the most commonly used way of adding data to objects. The obvious limitations are the lack of hierarchy, dependency, and extendibility of the embedded data (mostly limited to layer, color, ID, groups, or blocks). With the flexibility and ease of creation, the data likewise remains easily accessible, thereby allowing for errors, inconsistency, and loss of integrity. Any automation based on user-sorted data therefore needs to be handled with care.

2.1.2 Databases

One way to add data to models is through the use of external relational databases. These can be linked to object identifiers so that data can be added to these models in a consistent manner (through object tables which then relate to other tables representing the data required). This method is already well developed, for instance in facility management, where databases are used for maintenance, meeting room booking systems, or tracking equipment. While this is certainly a very effective, flexible, and upscalable solution, it is also a complicated one to use. The critical issue with this method is that in order to keep the database consistent, every change in the model has to be monitored. Event watchers have to be carefully written so that upon deletion of an object, the related entry in the database will also be deleted. Furthermore, the database is a separate entity that has to be maintained, copied, and backed up alongside the project.

2.2 softBIM Data Management

2.2.1 ObjectData

The method we suggest for the softBIM approach relies on the fact that most CAD software exposes via API a functionality to add *ObjectData* to objects or *DocumentData* to documents in a key/value-pair. (Examples of data stored on objects are XData in Autodesk AutoCad or UserText in McNeel Rhinoceros 3D. Examples of data stored on documents are XRecords in Autodesk AutoCad or DocumentData in McNeel Rhinoceros 3D. For the sake of consistency throughout this study these will be referred to as ObjectData and DocumentData, respectively.)

This mechanism of the drawing database is mostly used by third-party plug-in developers who utilize the data storage for additional information for their custom object types in the drawing. The benefit is that the data-object linkage lives on until the object itself is deleted but cannot otherwise be easily altered in error. As with most software, there are no standard GUI tools available to make use of this data; tools have to be created to exploit this capability. We suggest making use of this method in the early design stage and associating data for design purposes with 3D models through custom-made tools. This DesignData can be as simple as a program of demands but can also extend to more complex examples where the data is used—for example, in the structural information for columns and slabs or material information. For projects with a complex program (e.g., mixed use) the ObjectData could contain all the information of a program brief that is relevant for the early design phase. In a fast-paced submission, it is always a difficult task to organize a mixed-use program while keeping track of all the various spaces and areas. In these early phases, a flexible and lightweight system is needed in order to be able to adjust design ideas quickly, yet simultaneously obtain constant feedback concerning the data. Here the softBIM approach supports the design process by offering a system that integrates with the modeling environment without information-heavy, complex modeling that slows down the design process and impedes the flow of ideas.

2.2.2 Data Import

In a first step a spreadsheet (Figure 1) is created, which organizes the full program into categories.

Acting as a database, it feeds the custom tools with the data to be stored in the DocumentData of the 3D drawing. With all the data contained within the document, it can be easily queried at any moment in order to acquire information about the required spaces (Figure 2).

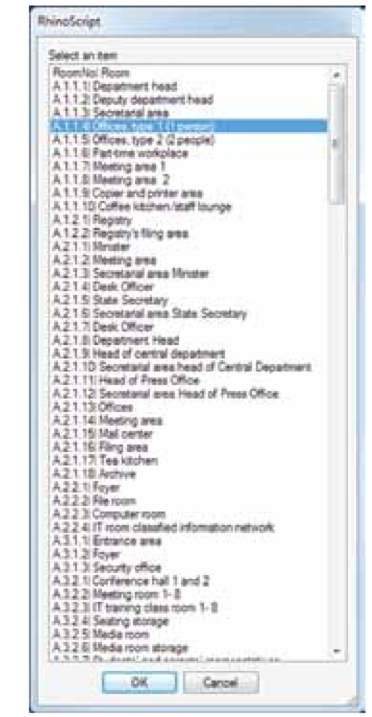

figure 2

figure 2

Query interface.

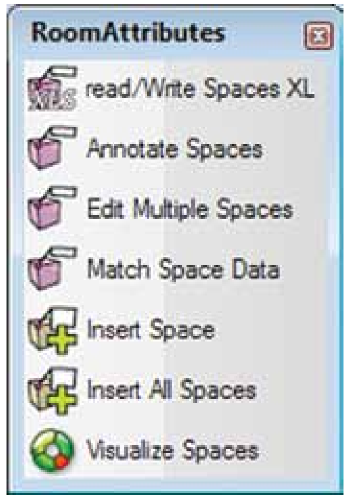

figure 3

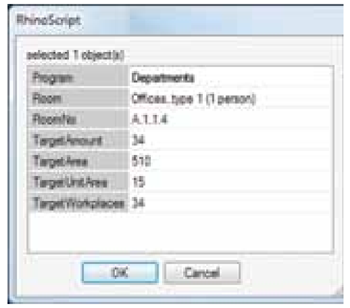

figure 4

figure 5

Several custom tools (Figure 3) allow for different ways to insert spaces into the model (e.g., by choosing from a drop-down list).

Upon insertion the space will get the correct measurements, and all data associated with the space is stored as ObjectData (Figure 4).

2.2.3 Feedback

By way of additional queries, the program brief is compared to the current design and information is fed back on which spaces and areas are still required (Figure 5).

Another way to support design decisions is the visual feedback of this data comparison (e.g., charts comparing the different categories of program to one another). The contained DesignData can be exported back to spreadsheet tables or used for automated drawing creation in which all DesignData is transferred and kept intact.

3 APPLICATION—CASE STUDY

This section explores the strategy and workflow devised by UNStudio for the project stages, from concept to design development, of a mixed-use program in a dense urban context. Specifically, we target the development of bespoke tools that combine and interrelate geometric and nongeometric data while remaining inside a CAD environment that is intelligible and accessible by the full design team.

The case in point is a large-scale development comprising two main programmatic components on a very compact city block. One component is an office building, the other a residential tower; the two are connected by a common plinth. Designing for a large-scale residential program in the local context of this project confronts the designer with a rigid set of conventions, obliging adherence to a strict rule set on efficiencies in order to maximize use and revenue of limited space, while at the same time giving equal qualities to apartment units of the same type. One possible consequence, in the case of a residential facade, is a monotonous building envelope consisting of a repeated stacked pattern of identical units. Having in the conceptual stage identified this as a likely consequence of planning law—exemplified by much of the existing local building stock—one main goal of the design brief became to develop the design of a building that has an intricate and rich texture that counteracts the surrounding monotony.

3.1 Design Intent / Rational Irregularity

"Let us then understand at once that change or variety is as much a necessity to the human heart and brain in buildings as in books; that there is no merit, though there is some occasional use, in monotony; and that we must no more expect to derive either pleasure or profit from an architecture whose ornaments are of one pattern, and whose pillars are of one proportion, than we should out of a universe in which the clouds were all of one shape, and the trees all of one size" (Ruskin 1874).

Figure 3

Custom tools for space insertion and editing.

Figure 4

ObjectData read-write interface.

figure 5

ObjectData vs. program brief feedback.

figure 6

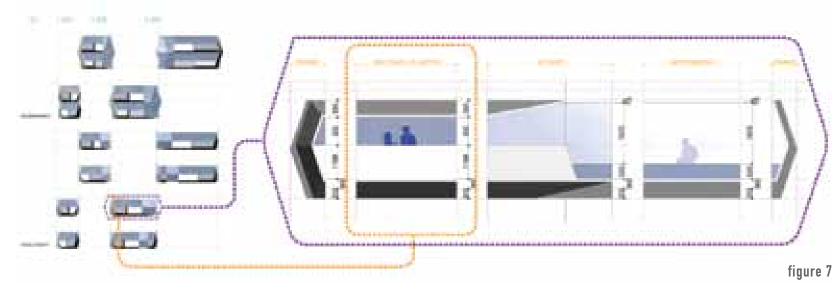

figure 7

figure 9

John Ruskin in his essay "*The Nature of Gothic*" (Ruskin 1874) talks of variety, irregularity, and intricacy. These variations, according to Ruskin, are what afford architecture the freedom and the delight that we humans need in order to be able to subjectively relate to buildings.

In the chosen case study and in line with Ruskin's tenet, it was desirable to achieve an effect of variety, irregularity, and intricacy throughout the whole facade by combining, in a simple fashion, a system of multifaceted, multitextured, and multiscaled elements. As the building is located in a dense urban area, it was further necessary to facilitate more privacy for units located in the lower parts of the tower.

In accordance with the client brief, the residential part comprises five different dwelling types, ranging from studios to penthouses. Moreover, local building practice, regulations, and climatic conditions stipulate a series of constraints to the component library. Through an intricate relationship between GFA and NIA given by local code, a central role in the facade composition is given to balconies, AC ledges, planter boxes, and bay windows. Each of these elements becomes an individual component in the 3D library making up the building envelope.

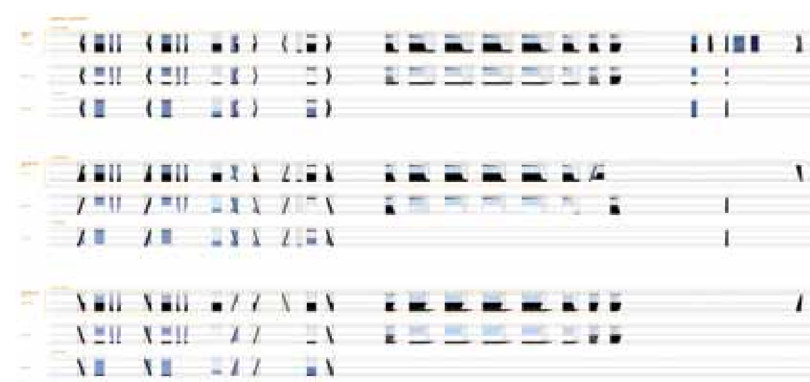

figure 8

figure 6

Library of hexagon components and small-scale differentiation.

figure 7

Basic facade components of an apartment.

figure 8

Facade component library.

figure 9

Block methods toolbar.

figure 10
3D to 2D / CAD elevation diagram.

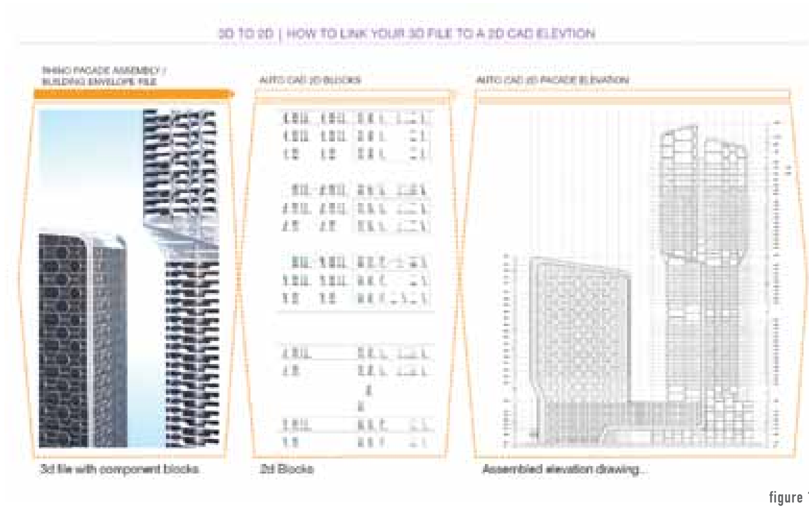

figure 10

The tower's facade is based on a combination of hexagon modules of one- to two-story height. While the hexagon modules act as a basic framework, the components are the elements we use to introduce variation and individuality into the facade. The components change in materiality (solid to open), geometric articulation (angling of glass to break reflections), and scale (changing the actual size of components).

It was evident that developing a facade texture based on the above ideas would bring with it great complexity. An irregular facade assembly poses challenges to version tracking and controlled update management, as well as to the diffusion of these within the design team. In order to confront these issues, a workflow was developed around the idea of an envelope consisting of a finite set of 3D components located in a central repository. In conjunction, a series of customized tools were created for the creation, management, and transfer of DesignData.

3.2 Design Output / 3D to 2D Geometry

With the numerous updates to massing and the count of apartment units that occurred especially in early stages of the present study, the drawing of elevations in a case such as this not only makes for time-consuming manual work but, more importantly, makes changes difficult to track; errors can occur.

For this reason the block logic is brought forward, also, when dealing with another of our stipulated BIM fundamentals: producing 2D information from 3D geometry. While 3D-based procurement is rapidly gaining legislative support (Hermund 2012) in more and more countries, the AEC industry as a whole has a long way to go before a full 3D process is feasible. Therefore reliable protocols will still be needed for the foreseeable future in order for the driving 3D geometry to convert into contractual 2D drawing. (Figure 10).

Most CAD platforms offer the opportunity to predefine the topology and bundle together geometry packages (often referred to as blocks or cells) and metadata contained within them (e.g., ObjectData, see section 3.2.1). Drawing on this ability as well as that of maintaining and recognizing such bundles in a controlled way between two and three dimensions within the same CAD platform or between different platforms, we are able to set up an automated drawing extraction facility. In doing so it is even possible to modulate the level of detail to certain extents through the use of various dynamic block properties available in CAD.

figure 11
Input spreadsheet.

Figure 12
Units stacking matrix.

figure 13
Units module makeup.

figure 14
Unit-based facade makeup and materials.ObjectData interface.

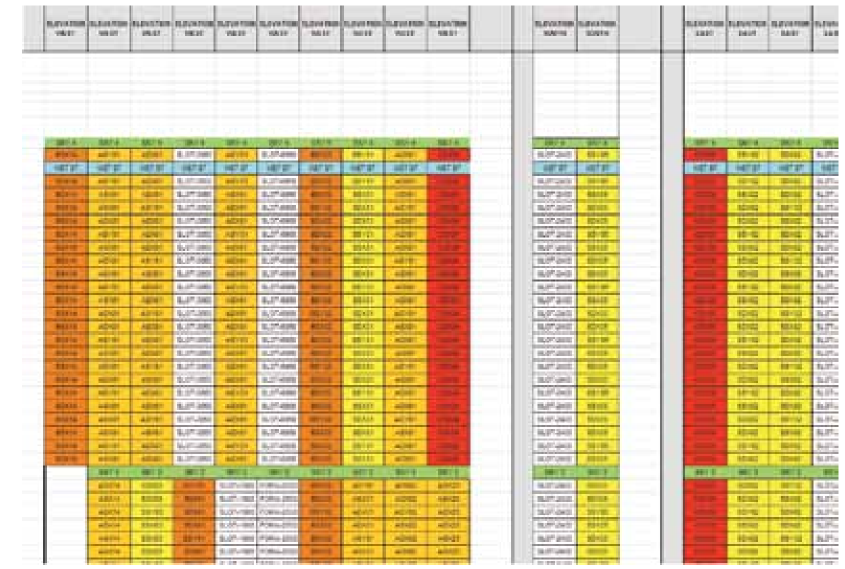
figure 11

figure 12

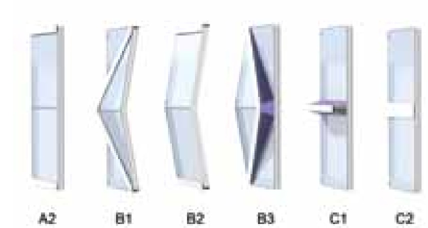

figure 13

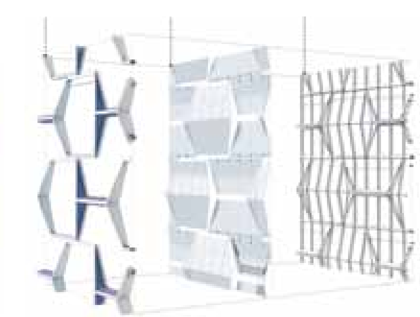
figure 14

figure 15
Entity metrics on output side.

figure 16
ObjectData interface.

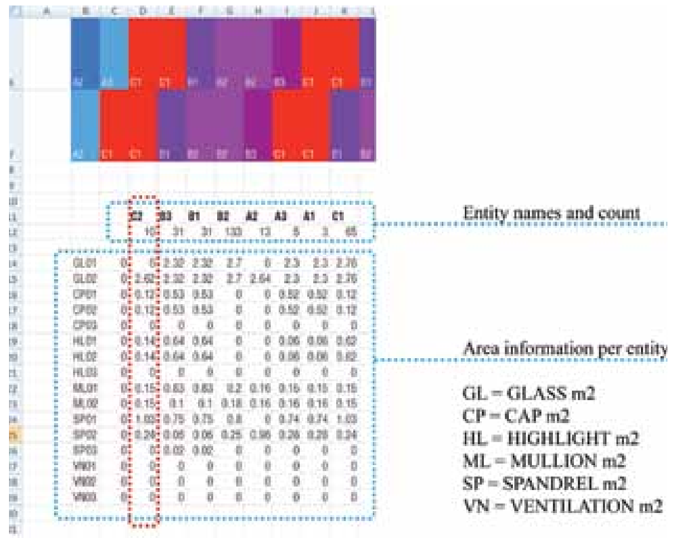

figure 15

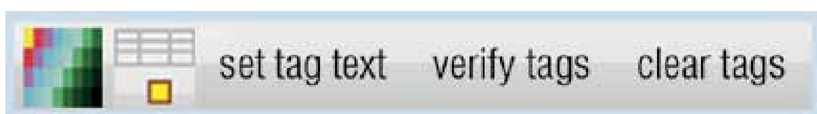

figure 16

3.3 Design Driver / Text to Geometry

In order to be able to work on the elevation with full overview and close numeric control over the unit distribution, an abstracted elevation is set up in a spreadsheet. Programmatic and unit-specific information are transferred in run-time to the geometry space (Figures 11 and 12).

3.4 Design Output / Geometry to Text

Both case study tower facades are based on a curtain wall module, consisting of a number of submodules that recombine to create two distinct yet thematically related signature patterns. Varying with the design intent, the different sides of the building recombine these same submodules into different assemblies. Via the API, we translate the geometric data into a spreadsheet, topologically remapping the sequence of the facade modules, while including information such as areas, materials, color, and entity names (Figures 13 and 14).

For purposes of topology verification as well as quantity takeoffs, it is useful to make such conversions between DesignData and geometric data. As mentioned in section 3, these operations rely on a common recognition of attributes between different data storage modes. In the present study DesignData describing material and technical makeup is appended to geometry as ObjectData, and in addition area or volume information is queried in run-time directly from the geometry object.

At the other end a topologically coherent model is rebuilt in a spreadsheet, displaying information on the ObjectData and relating it to its relative position in the architectural assembly (Figure 15). These DesignData entries are all appended to their geometry carriers through the use of ObjectData (Figure 16).

4 EVALUATION—FINDINGS

System flexibility: Present-day notions of associative-parametric design are often built upon the idea of the flexible and light-footed system, allowing for rapid testing, alteration, and design iteration almost literally on the fly. This is oftentimes in direct contrast to the hardBIM, in which the richness and layering of information can make for a relatively rigid and heavy-handed model. An open-ended approach can work toward bridging the divide between what we may think of as the design-intuitive and the executive-reliable.

System lifespan: Proponents of different methodologies, and by extension often specific software platforms, invariably need to collaborate and exchange information in design practice. In a globalized market where distributed design team constellations are increasingly common and where the design-build principle is rapidly gaining in popularity, certain models of procurement and project phasing may not retain the same designing party from the early stages through to the execution phase of a project. The significant investment in software licenses and personnel that the hardBIM platform requires may not eventually prove to be in proportion to the actual lifespan of the system as required by the project. Therefore, the ability to plug in to a softBIM methodology can potentially offer the most convenient option for a design practice.

System resilience: softBIM can provide multiresolution solutions and a level of detail that can be tailored to suit the specific project's needs. While this in itself is not unique to a customized model (commercially available standardized solutions also offer modulation in detail level), it provides an environment that allows for advanced parametric application while at the same time remaining totally inclusive at the front end. In other words, we are able to execute custom-made parametric routines as well as traditional, nonassociative CAD modeling, drafting, and drawing output all within the same GUI. Normally this can be made within the same GUI, or a combination of 2D/3D GUIs that are already part of the established workflow. In architectural practice this can present several benefits for interaction within the immediate as well as the extended project team.

5 CONCLUSIONS

The debate around the BIM is undermined by radically diverse understandings of its actual definition and purpose. Is BIM a methodology that enables and enhances collaboration and consistency throughout the designing cycle (and potentially beyond)? Is BIM a procurement procedure that reduces tendering/construction risk by making the access to the design data more transparent and complete? Is it a 3D model with all the necessary embedded information that renders the use of 2D documents unnecessary?

Answers to these questions differ according to your role within the AEC industry, and perhaps even depend upon one's profile within each profession. However, with each of these few questions, more are raised. How smoothly would a process run in which all parties make use of the same methodology (especially if by methodology a specific software is to be understood)? What level of transparency must be achieved in order for the risk to be effectively reduced? How do we define all the necessary information?

These hardBIM solutions mostly require specialist input, have slow learning curves, and are inherently rigid. Furthermore, increasingly large-scale projects and large teams of consultants put pressure on coordination among disciplines, for which the answer from hardBIM is to increase the level of detail. All these factors discourage the use of BIM in the early design stages while the design process is still fluid. However, during these stages a degree of control, coordination, and ability to access information from the models is as important as in later phases. In fact, it could be argued that a rigorous access to and assessment of certain parameters is critical throughout these fluid stages, when ideas can be backed with accurate data that may make or break the design.

The use of three-dimensional modeling in UNStudio's practice has been central to the active development of ideas and concepts, along with creating complex and challenging geometries and organizations. The coordination between three-dimensional concepts and models and two-dimensional drawings and other documents traditionally used in the industry has forced the development of a workflow that accurately translates information from one format to the other and has provoked reflection concerning the best ways to use (and add to) the original three-dimensional geometry to its fullest. In our practice these techniques for extracting information from the available models are customized to the specific purposes for which the information will eventually be used. A purposeful BIM is built according to the level required for each phase or status of the design, in addition to an exact adherence to the contractual requirements, client expectations, and workflow conditions. By putting forward softBIM as a method, we do not aim to claim that it can, or should, replace in its entirety the highly sophisticated hardBIM packages that are currently available commercially. On the contrary, the rapid spread of hardBIM is presently helping to question stagnated positions and boundaries within the AEC industry, something we believe will ultimately lead to more cost-effective and less wasteful processes and overall better buildings.

Instead, our experience suggests that the workflow imposed by hardBIM presents challenges to geometrically complex projects, specifically during early phases. Furthermore, the systematic linking of abstract metrics to geometrical containers in the early phases offers not only the possibility of informing the decision-making process with quantitative results, but also—and perhaps more importantly—the potential extension of the decision-making period during which design exploration occurs. Therefore we advocate a flexible and targeted access to the data, which avoids the impediments to workflow, transparency, and information exchange that hardBIM may impose on a project team.

Such an approach becomes both effective and strong, as it streamlines the effort of the design development according to the essential issues of the relevant project phase. The added benefit to the design process is that it affords more time for the development of ideas, while at the same time testing and acquiring instant validation of critical parameters—which in turn enables informed decision making on early stage concepts.

REFERENCES

Dritsas, S., and M. Becker. (2007). Research & Design in Shifting from Analog to Digital. Proceedings of ACADIA 27: Expanding Bodies: Art, Cities, Environment. Halifax, Nova Scotia: Association for Computer Aided Design in Architecture.

Hermund, A. (2011). Applied 3D Modeling and Parametric Design. PhD Dissertation, Royal Danish Academy of Art School of Architecture.

Jernigan, F. (2007). BIG BIM little bim: The Practical Approach to Building Information Modeling. Salisbury, MD: 4Site Press.

Ruskin, J. (1874). The Stones of Venice, Volume II: The Sea Stories. London: Smith, Elder & Co.

Saggio, A. (2007). The IT Revolution in Architecture: Thoughts on a Paradigm Shift. Rome: Carocci.

PROGRESS TOWARD MULTI-CRITERIA DESIGN OPTIMIZATION USING DESIGNSCRIPT WITH SMART FORM, ROBOT STRUCTURAL ANALYSIS, AND ECOTECT BUILDING PERFORMANCE ANALYSIS

Robert Aish
Autodesk

Al Fisher
Buro Happold

Sam Joyce
University of Bath

Andrew Marsh
Autodesk

ABSTRACT

Important progress toward the development of a system that enables multi-criteria design optimization has recently been demonstrated during a research collaboration between Autodesk's DesignScript development team, the University of Bath, and the engineering consultancy Buro Happold. This involved integrating aspects of the Robot Structural Analysis application, aspects of the Ecotect building performance application, and a specialist form-finding solver called SMART Form (developed by Buro Happold) with DesignScript to create a single computation environment. This environment is intended for the generation and evaluation of building designs against both structural and building performance criteria, with the aim of expediently supporting computational optimization and decision-making processes that integrate across multiple design and engineering disciplines.

A framework was developed to enable the integration of modeling environments with analysis and process control, based on the authors' case studies and experience of applied performance-driven design in practice. This more generalized approach (implemented in DesignScript) enables different designers and engineers to selectively configure geometry definition, form finding, analysis, and simulation tools in an open-ended system without enforcing any predefined workflows or anticipating specific design strategies and allows for a full range of optimization and decision-making processes to be explored.

This system was demonstrated to practitioners during the Design Modeling Symposium, Berlin, in 2011, and feedback from this demonstration has suggested further development.