MAKING USE OF 3D PROGRAMMING TO DOCUMENT AND ENHANCE AN ARCHITECTURAL VOCABULARY

Case study dealing with artefacts of the Canadian built heritage

NATHALIE CHARBONNEAU, PIERRE GRUSSENMEYER
UMR MAP 694, Équipe PAGE, INSA de Strasbourg, France

ABSTRACT: This paper examines the objectives and describes the methodology of an ongoing research project concerned with the built heritage of Montreal, Canada; more precisely, with the openings (doors, windows, bow-windows, dormers, etc.) in residential buildings. Within the framework of this project, we aim to implement digital environments to document and enhance the architectural vocabulary of openings. We adopt 3D programming to describe the way in which the artefacts are assembled from different components. The aim of this process is to enable the introduction of variations into the principles governing the composition of the artefacts, so as to generate new configurations.

KEYWORDS: Built heritage, openings, 3D programming, typology

RÉSUMÉ: Cet article énonce les objectifs et décrit la méthodologie d’un projet de recherche actuellement en cours de développement. Ce projet porte sur le patrimoine bâti de Montréal, Canada, plus précisément sur les ouvertures dans les bâtiments résidentiels (portes, fenêtres, orielles, lucarnes, etc.). Dans le cadre de ce projet, nous cherchons à implémenter un environnement numérique permettant de documenter et d’enrichir le vocabulaire architectural des ouvertures. Nous avons recours à la programmation 3D pour décrire la façon dont les artefacts sont assemblés à partir de différents composants. Le but de ce processus est de permettre l’introduction de variations dans les principes sur lesquels repose la composition des artefacts, de façon à générer de nouvelles configurations.

MOTS-CLÉS: Patrimoine bâti, ouvertures, programmation 3D, typologie
1. MERIT OF DIVERSITY

This research project deals with residential buildings of the built heritage of Montreal, Canada, with a focus on openings; more precisely, on windows, bow-windows, dormers, and doors in all their forms. Our main aim is to find ways of making use of 3D programming to document and enhance the architectural vocabulary of openings. In terms of openings in residential buildings, the Montreal built heritage combines various influences (from France, England, North America, etc.) and encompasses a wide range of solutions (see Figure 1). The worthiness of vernacular architecture is based in this tremendous diversity of architectural forms and details (Pinard 1987).

**FIGURE 1. EXAMPLES OF WINDOWS OF THE MONTREAL BUILT HERITAGE.**

Until the twentieth century, these windows, dormers and doors were pieces of craftsmanship; that is, new configurations were the “by-products” of previous ones. If we consider the architectural history of Montreal, we notice that during the twentieth century there occurred a breakdown of what Caniggia and
Maffei (2000) refer as the typological process. Over this time, the emergence of International Style has seen the materialization of other sources of inspiration. Accordingly, openings in new buildings do not necessarily refer to the Montreal built environment: architectural designers develop new solutions that are also likely be utilized in other built environments. As part of this trend, architectural details are sometimes reduced to their simplest expression, barren of all ornamentation. This new paradigm partly explains the clash between what was built in the residential districts of Montreal during, for example, the 1920s, and that which is built today.

Considering solely the geometry of artefacts, one may posit that the outcome of this evolution was an impoverishment of the architectural vocabulary. Indeed, there exists a trend toward fewer types of openings, and their forms are purified, so to speak. The infatuation with International Style, in combination with the industrialization of building components, induced a process of standardization and globalization. While contemporary design is valued, vernacular architecture reaches the point where it is considered obsolete: the less we know about architectural heritage, the less we care about it. In some cases, residential buildings dating from the beginning of the twentieth century are renovated without taking into account the elements that make up the building front. We believe that some examples of renovation work inconsistent with the original architecture arise from a lack of commitment and knowledge regarding the built heritage (see Figure 2).

**FIGURE 2. EXAMPLE OF INAPPROPRIATE RENOVATION WORK.**

We propose that it would be worthwhile to renew our awareness of the richness and diversity of our built heritage. Such ‘introspection’ could enable us to enhance the quality of the built environment and establish a better link
between past and present architectural forms. In any case, to obtain a better knowledge, understanding and appreciation of the heritage handed down by our predecessors would represent a meaningful outcome.

We believe that the architectural designer could draw inspiration from existing artefacts within a specific built environment. In doing so, computers could provide the designer with a useful tool. We are not suggesting the reproduction of existing artefacts or the repeated use of the same designs: it is a matter of enabling the architectural designer to gain access to a wide range of solutions and to profit from the richness and diversity of the built heritage.

In this context, the goal of the present research is to develop a digital environment that would serve two purposes: (i) enable the user to take cognizance of the multiplicity of existing artefacts and to better appraise them, and (ii) provide insight into the new configurations of openings that could be developed from existing types.

With this goal in mind, we undertook a three-stage procedure. The first step was to gather information regarding the artefacts under study and to organize these artefacts as a typology; this work has now been published (Charbonneau and Grussenmeyer 2008). Accordingly, in this study we address the two following steps: (i) how we are proceeding to identify adequate software and scripting language to implement the digital environment mentioned above, and (ii) how we are planning to evaluate the outcome of the interaction liable to take place between the user and computer, within the framework of the designer’s experience.

2. FINDING THE RIGHT TOOLS

In this work, we aim to further the architectural designer’s knowledge regarding openings and to stimulate his/her creativity. This cognitive process will be based on the interaction likely to take place between the user and a set of parametric objects. For each type of opening, these parametric objects can be employed to automatically generate 3D models illustrating multiple scenarios, in accordance with the user’s specifications.

To adequately develop these parametric objects, it is necessary on the one hand to identify those parameters that are characteristic of each type of artefact and to systemize the relationship between them (presence of different components, rules of proportions, etc.). On the other hand, it is also necessary to select tools that enable us to describe these relationships; namely, a geometric modeling kernel and a scripting language that enables us to efficiently communicate with the kernel. Using such tools, we should be able to describe the geometry of the various entities via a scripting language; that is, we will develop algorithms that establish, for example, which geometric primitives are involved, how they are modified, and how they are assembled. When interpreted by the geometric
modeling kernel, the entities will be generated and displayed so as to enable the user to manipulate them in a virtual space.

We initially planned to use freeware in developing the digital environment because we wanted the user to be able to freely download all the software required to develop new configurations of openings. A first attempt to develop the digital environment was made using the 3D graphics software Blender and the procedures were developed using Python; however, these tools did not enable the concise use of Boolean operations. This limitation is a major drawback because artefact geometry is described via algorithms in which operations such as the union, subtraction and intersection of volumes are important elements.

To overcome this problem, we switched to AutoCAD, a commercial software package. Visual Basic (VB) for AutoCAD provides a friendly user ‘tool box’ with which to develop the desired graphical user interface. Nevertheless, when tackling the task of describing artefact geometry, the scripting language requires a lot of information to generate 3D models. This wealth of detail results in rather cumbersome algorithms. This limitation hindered progress during this second attempt to develop the appropriate digital environment. For instance, to describe the geometry of a parallelepiped rectangle, VB requires the declaration of every dimension and coordinate, as shown in the following code:

```vbnet
Dim lengthRec As Double
Dim widthRec As Double
Dim heightRec As Double
Dim origin (0 To 2) As Double
lengthRec = 10
widthRec = 15
heightRec = 20
origin(0) = 0
origin(1) = 0
origin(2) = 0
Dim boxRec As Acad3DSolid
Set boxRec = ThisDrawing.ModelSpace.AddBox(origi, lengthRec, widthRec, heightRec)
```

Therefore, we changed tack again, and as a third attempt we worked with Maya and Maya Embedded Language (MEL). As with VB, Maya and MEL provide a set of user-friendly functionalities to implement a graphical user interface (see Figure 3). Using this approach, it is now possible to describe the geometry of various entities in a compact manner: the algorithm is optimized. Returning to the above example, the parallelepiped rectangle is now described as follows:
```csharp
float $larRec = 10
float $hautRec = 15
float $profRec = 20
polyCube -w $larChas -h $hautChas -d $profChas -n Nom;
```

Clearly, it is important to avoid ambiguities to ensure that the code is correctly interpreted by the computer; however, it is also important that we do not overcomplicate the process by which the programmer interprets the code. If the algorithmic solution is neat and concise, the programmer will be in a better position to identify patterns within the relationships that exist between the parameters and within the principles governing the composition of the artefacts. Conversely, a ‘messy solution’ has one or all of the following characteristics:

- **Obscure:** it doesn’t profit adequately from the advantages of symbolic programming in naming the entities and conveying the programmer’s meaning in a clear way.
- **Redundant:** the employed reasoning (or part of it) is unnecessarily duplicated.
- **Complicated:** the principle and logic that underlie the algorithm are partly or completely hidden.

Kalay (2004) proposed that there exists much in common between the development of computer programs and architectural design: both have a scientific basis, yet their successful accomplishment falls within the realm of art. In the present case, we found that MEL was the best language with which to ‘design’ the algorithmic solution in a concise manner. As noted by Springer and Friedman (1989:4), «In programming, perhaps more than in other arts, less is more. Simplicity is nowhere more practical than in programming, where the bane is complexity».

A clear and flexible scripting language is essential to develop the case study beyond simplistic scenarios. Clearly, modern computers have large processing capacity and are able to generate a large number of configurations based on complex geometrical operations. We argue that the only factor liable to hinder the developer’s work would be his/her inability to read over what he/she has written and difficulty in readily understanding the meaning of the code. If the developer is put off by the modifications required to include additional scenarios, the system is unlikely to evolve as desired.
3. EVALUATING THE OUTCOME OF INTERACTION BETWEEN THE ARCHITECTURAL DESIGNER AND DIGITAL ENVIRONMENT

Parametric objects are usually accessible via text boxes: the user enters numerical values that enable the regeneration of the 3D model. In the present case, parametric objects are accessible via sets of radio buttons that present different aspects or components of the artefact. The user can toggle between different configurations in an iterative way; by selecting different options, he/she can explore many different scenarios. For instance, the user can configure a new window by combining, in turn, different types of lintel, transom, mullion, etc. (see Figure 4).

As stated above, our aim is to enable the architectural designer to explore a wide range of solutions with regard to openings. We envisage that the exploration process would unfold as follows. The designer chooses a type of opening and selects different options to assess the resulting solution. Because we are seeking to establish a dialogue between the designer and the computer, the processing speed and display speed are of utmost importance. When the user selects an action, feedback is required almost immediately: it is important not to sever the connection between one experiment and the next. Exploring the
digital environment at a steady rhythm contributes to stimulating the user’s cognitive abilities.

After developing an interesting scenario, the user is able to access the functionalities available in the software’s user interface (scale, rotate, move, etc.) or to replace a geometrical form by another to modify the shape of the entity. In doing so, the user pursues the design process, giving free rein to his/her creativity. However, we presume that these ‘artistic creations’ (if inserted into the composition of a contemporary building) would not be disconnected from the Montreal built environment: because they were elaborated from traditional architectural elements, they would result from a typological-like process and would thereby convey the impression of being both contemporary and vernacular.

Furthermore, we believe that some designers might be interested in the startling effect of arbitrary variations. With this in mind, we are incorporating the option of random selection into the algorithms, whereby the artefact components are randomly selected by the system.

Figure 4 shows various scenarios combining different lintels, sills, mullions and transoms. The examples shown in this figure may seem somewhat simplis-
tic; however, while exploring the potential of the selected tools (software and language), we decided to tackle only the simplest scenario; accordingly, we began the implementation of the digital environment by working on sash widows. We intend to elaborate other parametric objects, describing other types of openings based on a more complex geometry, proceeding to sampling bow-windows, dormers and doors.

Although we are still developing algorithms and refining the user interface, we are now planning a consultation process with potential users. Once we have implemented parametric objects describing several types of windows, we will proceed with a consultation process involving architectural designers. In doing so, we will seek to answer the following questions: Is the user able to explore the digital environment with ease and at a steady rhythm? Does the user feel that his/her architectural vocabulary is enhanced in a meaningful way? To what extent is the practitioner inclined to insert the new configurations into his/her own compositions? And ultimately, to what extent is the digital environment an asset to the design process?

4. CONCLUSION

Our experiments with MEL (Maya Embedded Language) were successful as we were able to describe the geometry of a sash window in a concise and flexible manner. This result led us to presume that it would be possible to tackle a large number of interrelated scenarios. In this study, multiplicity is essential in order to answer the questions posed above. These questions are important because we believe that in the broad field of architecture there exists an alternative path to that of globalization. This path could encourage the architectural designer to enhance the specificity and originality of his/her own built environment.

In our endeavor to demonstrate that architectural details are not obsolete, we are driven by a fascination for the geometric diversity that surrounds us, rather than by nostalgia. When a built environment becomes overly familiar, we look at it without seeing, so to speak. It is hoped that the digital environment we are implementing will not only enable the user to renew his/her awareness of vernacular architecture but also to design new configurations of openings representative of their time while in harmony with the surroundings.

The digital environment implemented within the framework of this research project is intended exclusively for the architectural designer willing to contribute to the Montreal built environment. It goes without saying that, providing the present research project will prove to be conclusive, this approach could be transposed to other built environments.
ACKNOWLEDGEMENTS

We would like to thank the Fonds Québécois de Recherche sur la Société et la Culture (FQRSC), who funded this research project within the framework of the Bourse de Recherche Postdoctorale Progamme.

REFERENCES


Blender, < www.blender.org >

Autodesk AutoCAD,

< http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2704278 >

Autodesk Maya,

< http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=7635018 >