Energy and Buildings 68 (2014) 786–798

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

Parametric scripting for early design performance simulation

Julien Nembrini [a,*], Steffen Samberger [a], Guillaume Labelle [b]

[a] Structural design and Technology Chair, Hardenbergstrasse 33, Universität der Künste, Berlin, Germany
[b] Media and Design Lab, BC Building, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

ARTICLE INFO

Article history:

Julien Nembrini wishes to dedicate the article to Guillaume Labelle who passed away in January 2012.

Keywords:
Architectural design
Early design stage
Parametric modelling
Building Performance Simulation
Sensitivity analysis
Passive ventilation
User control

ABSTRACT

This paper discusses the advantages of using a coding interface both to describe form and run performance simulations in the context of architectural design. It outlines the relevance of combining recent interest in the design community for *parametric scripting* with available expert-level Building Performance Simulation. This approach enables designers to address performance-related design questions at the early design stage. Pitfalls when considering non-standard solutions and the potential of the approach to circumvent such difficulties are exemplified through a housing building case study, emphasizing the under-evaluated role of the analysis tool in steering design decisions. To circumvent the difficulty for non-expert users to interpret simulation results and transform them into design moves, the use of sensitivity analysis is proposed for its specific interplay with parametric scripting. Its potential for classifying parameters in order of importance is presented in the context of a design problem. The contribution outlines how providing designers with exploratory tools allows to consider sustainable construction in a systemic manner.

1. Introduction

Early on in the architectural design process, variants are typically explored in parallel until only one dominates and takes precedence. During this time, design elements such as context, overall form and typology are only partly defined and evolve and solidify as the design process unfolds. Unfortunately in practice, indoor comfort rarely plays a preponderant role in driving design, despite the dramatic impact on performance that decisions taken at that time have. As a result, indoor comfort is often only subsequently addressed, resulting in an overly strong dependence on building technology to guarantee comfort and performance needs [1].

Finer design details such as window openings or shading elements are typically left to later design stages. However, such details have great impact and interact strongly with form and context [2]. The aim for a better performing built form thus requires consideration of the entire building system and needs to include all relevant elements at all levels of detail. Typically, architects address such complexity by relying on experience and the use of design heuristics. However, the specificities of each context can undermine the relevance of such an approach.

In this regard, use of computer models for early design performance assessment by designers has demonstrated that higher levels of sustainability can be achieved [3]. Given interface and tool democratization, there are numerous examples of design approaches in interaction with performance tools, delineating the category of *performance-based design*. However, many such examples tend to mirror tool availability and capabilities whose relevance to the design question are crucial to producing meaningful results [4]. The difficulty in assessing the quality of results by designers, being non-experts in simulation, raises a conceptual question on the overall scheme [5], especially since the final building performance is tributary to choices made at early design stages.

In this sense, such non-expert usage of building simulation tools is controversial. There have been several proposals to overcome the problem, which can be grouped into two categories: simplified tools and simplified and reduced interfaces to expert-level tools [2]. Of these two categories, simplified tools tend to be more popular [6,7]. However, simplified tools over-simplify geometry and can use models considered too unreliable by building systems engineers [8]. In contrast, when expert-level tools are given simplified and reduced interfaces, usage is restricted to implemented functionality, hindering customization or mainly translating geometric information. What this second category lacks is the possibility to reach beyond the capabilities of the interface to address demanding design questions to explore the potential of non-standard solutions [8].

In parallel, in the architectural design practice and in research, there has been a growing use of *scripting* and *parametric design*. Scripting in the architectural sense refers to the use of computer

* Corresponding author. Tel.: +41 32 341 5968.
E-mail address: julien.nembrini@gmail.com (J. Nembrini).

0378-7788/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.enbuild.2013.09.044

(visual) code instructions to define architectural form. Parametric design enables the maintaining of dynamic links between parameters and their use in geometry definition for real-time, continuous modification. The conjunction of both is denoted here as *parametric scripting*. Example software include Grasshopper, which enables visual programming to steer the popular NURBS-based Rhinoceros CAD software [9]. Another example is Generative Components which although principally relying on a GUI interface, also allows scripting and has been largely used for this ability [10]. The recent DesignScript by Aish puts the emphasis on script-based geometric definition and very recently includes exporters to gbXML standard for energy performance analysis [11,12].

Despite being associated with ambitious form, parametric scripting has a strong potential for generating and exploring early design variants. Using such technique, designers are able to automate geometric description and modification of architectural form. By coupling it to Building Performance Simulation (BPS), non-experts can generate sufficient design details to run fully-featured BPS tests in the early design stages. Through parametric modifications, the exploration and comparison of alternative designs in achieving performance goals is considerably eased. However, existing examples mainly rely on visual programming tools and are therefore limited in the level of complexity tractable [9,13].

In the context of parametric scripting, notions such as design alternatives comparisons, detail generation to populate simple massing studies, or data interoperability are considerably facilitated by automatic production, modification, or translation of model information with the help of code instructions. Parametric scripting thus bears the ability to provide a different point of view on the nature of an easy-to-use BPS interface, the debate about using simplified tools versus subsets of full-featured tools, or the difficulty to early define design physical information; topics identified as of importance for the relevance of BPS tools in the early design stage and their acceptance in a practice context [3,2,14].

Providing means to simulate building thermal behaviour merely improve understanding of parameters' influence on performance if the building is of relative complexity. To support design decision making, the presented research makes use of *sensitivity analysis* (SA) techniques which naturally combine with the scripting approach.

If there are numerous examples of SA applications to BPS, the majority are interested in assessing model sensitivity stemming from parameter uncertainties such as material properties [15], climatic parameters [16], usage schedules, etc. In standard practice, model geometry is fixed and the uncertainty in predicting the model behaviour is studied [17]. A recent contribution focuses in increasing the number of parameters included in the analysis by an order of magnitude [18]. Instead, the present research focuses on the high variability of design parameters at early design stages. Using the same technique in a different context and on a different set of parameters provides completely different information: the focus, initially on quantifying simulation accuracy, is shifted towards indicating performance influencing parameters to the designer. If this approach is briefly mentioned in [15], Struck et al. thoroughly study the design option space in different design settings: design studio, built references, and interviews with practitioners [19]. The present contribution argues that potential advantages of the use of SA for design support as outlined by Struck and coworkers are actually straightforwardly available when designing using parametric scripting.

The research presented here demonstrates, through the development of a parametric scripting performance framework, how this designers' interest to define form through code provides an innovative context to include complex systemic performance assessment at early design stage. In the paper, the authors present the approach by describing an experimental tool-set taking the form of a parametric scripting engine with a set of interfaces to validated expert-level BPS tools, and explain how the tool-set integrates into the design process. Throughout, the aim is to *assist* design instead of automating it [19], allowing the designer to make informed choices between differing design solutions.

The potential of the approach for performance assessment is demonstrated through a housing case study developed in two typology variants, addressing thermal gains and natural ventilation questions. These inherently conflicting aims can be tested for a given set of design constraints and compared against energy consumption or standard comfort definitions [20]. If these comfort requirements are now included in building regulations, their relevance to assess the success of passive ventilation strategies is debated [21] as field studies have demonstrated that users accept warmer temperatures when they are able to themselves control windows for venting [22]. Probabilistic models have been proposed to include this observation [23,24]. The present research uses this topic to illustrate the necessity to circumvent assumption buried in BPS engines to effectively support the design process. Assumption of automatic control in passive ventilation are here bypassed in defining a simplified window control user model through the help of the BCVTB interface to EnergyPlus [25].

The paper proceeds as follows: After introducing the methods and tools used (Section 2), a design case study is defined (Section 3). Results of using BPS are presented (Section 4) and their design implications discussed (Section 5). Finally, some conclusions are drawn from the results presented.

2. Methods

The research presented here proposes a specific approach in generating form exclusively through code instructions. Starting with hand sketches, designers write shape-describing code, abstractly creating and modifying objects through geometrical transformations. When compiling the code, a three-dimensional representation is created which can be parametrically modified in real-time. This process results in decoupling form definition from form representation and forces designers to "think before modelling". By applying changes on the code rather than on the representation, designers are engaged in a *reflexive process* in the sense of [26]. From experience gained in design studios, the approach encourages designers to question, make precise and simplify their intentions by themselves [27].

This context naturally lends itself to using code instructions to augment the geometrical information in defining material properties and detail input needed by expert-level BPS. This allows for the involvement of full-featured validated BPS at the onset and can provide performance feedback as soon as a form is defined. Although limited in absolute terms by users' lack of expertise, such testing provides early insight in key parameters influence over performance.

By using validated, expert-level simulation tools, the proposed method restricts the space of possible building designs only by their extended capabilities. The problem is transformed into exposing these capabilities to the user appropriately. To accomplish this, the primary feature is the possibility for fast simulation and analysis setup. Secondly, different levels of user involvement and interaction must be accommodated for and allow the corresponding learning curves to take place. Finally, it is important to provide access to extended – unexpected – capabilities. These three characteristics are believed to be crucial for usage in early design: when unexpected design proposals typically emerge and need to be rapidly compared to standard solutions.