Design Space Exploration Framework

A modular approach to flexibly explore large sets of design variants of parametric models within a single environment

Ekaterina Fuchkina¹, Sven Schneider², Sven Bertel³, Iuliia Osintseva⁴
¹,²M.Sc. ³Vertr. Prof. Dr. Ing. ⁴Prof. Dr.
¹,²,⁴{ekaterina.fuchkina|sven.schneider|iuliia.osintseva}@uni-weimar.de ³sven.bertel@hs-flensburg.de

Parametric modelling allows to relatively easily generate large sets of design variants (so called design space). Typically, a designer intuitively moves through this design space, resulting in one or several satisfying solutions. Due to the theoretically large number of variants that can be created with parametric models, obviously, there is a high probability that potentially good solutions could be missed, which is not at least because of human cognitive limitations. Consequently, it is necessary to develop a certain strategy to support designers in order to search for design solutions. Even though, various methods to systematically approach large data sets exist, the application of them in the design process is a special case, firstly, due to the existence of many non-specifiable and subjective dimensions (e.g. aesthetics) and secondly because of the multiple ways how designers actually search for solutions. This demands for a more flexible approach to design space exploration. This paper investigates how different methods can be combined to support the exploration of design spaces. Therefore, a conceptual framework with a modular architecture is proposed and its prototypical implementation is demonstrated.

Keywords: Design Space Exploration, Parametric design

INTRODUCTION

New methods for designing also affect the way designers think about a design object and consequently also the approaches for conceptualizing and defining this object. The parametric design approach, which we will focus on in this paper, pushes the designer to decompose the object into components liable to particular mathematical definitions and further to define an algorithm to construct links in-between them (Oxman et al. 2015). The result of this approach is a design model able to produce different objects (design instances) by changing the values of its defining parameters. The number of generateable objects could theoretically be close to infinity, especially when number of input parameters is also large, which is the case when looking at the multiple
parameters real buildings are composed of. At this point, several problems arise. The first is related to understanding the criteria by which one design could be compared to another in order to retrieve a superior solution. The second problem is how to do such a comparison in a situation with large sets of design objects. Then, obviously, not all objects are worth considering and a high probability exists that potentially good solutions could be missed, which is not at least due to human cognitive limitations (Woodbury and Burrow 2006). While the first problem is out of the scope of this paper, the second addresses the need of exploration strategies in order to exploit all benefits of parametric design. Methods addressing this issue will be discussed in this paper.

In the first section we will give an overview on the place of exploration in the design process. Then we continue briefly reviewing existing methods for design space exploration. In the third section we propose a conceptual framework for flexibly exploring design spaces. Lastly, we demonstrate the prototypical implementation of this framework and demonstrate its capabilities on a practical test case.

DESIGN EXPLORATION
The design process in general can be characterized by two activities (see figure 1). The first gradually reduces number of alternatives until the final single solution is found. This is possible by specifying restrictions while design process is developing. For example, designer introduces ranges for parameters, or defines the set and range of objective requirements (e.g. costs, energy demand, visual access) and finally he applies own subjective criteria to arrive at a solution. The second activity reflects the fact that design problems are usually ill-defined (Simon 1973), where, starting from an abstract blurred vision of a desired solution, the designer elaborates ideas, models, parameters, restrictions and at the end includes own knowledge to prefer one solution in front of the others. Altogether, it leads to the expansion of domain knowledge of initial design problem. Consequently, we can draw the following conclusion - the exploration process is a part the design process and, by this reason, as any other stage in the process, it should leave after itself reduced set of design instances and give some new information about them.

Shneiderman et al. (2006) summarize that the design cycle has 9 stages (Problem definition, Gather information, Generate ideas, Modeling, Feasibility analysis, Evaluation, Decision, Communication, Implementation). In our work we are defining exploration as a part (stage) of the design process that happens after the “modeling” and “analysis” stage, but before the “decision” stage (see figure 2). The following paragraphs briefly overview what data will arrive at the exploration stage from the predecessors in order to further understand what exactly designer will explore.

The “modeling stage” in the parametric approach, is aimed to transform one particular idea into a parametric model by describing it through parameters and geometric relationships. At this stage the set of all alternatives is reduced to the ones that the parametric model is able to generate. This set of generatable instances is called “design space”. Here each instance carries a set of defining parameters. These parameters define the “parametric space”.

The purpose of the “analysis stage” is to compute features (“feature space”) that describe the performance of the generated instances. Performance thereby describes the degree to which certain design requirements are fulfilled (e.g. energy demand, floor area ratio, visibility). Reducing by such design requirements, the “solution space” is formed. Design instances in this space could have conflicting objectives, when one shows high performance by some criteria but low in the other. Therefore, to prefer one
instance to the other, intervention of the designer is necessary in a next stage.

At this moment, the “exploration stage” for an input will get the “parametric” and “feature” spaces together with a general information about design instances (e.g. geometry, layouts). The simplest definition of the exploration process is that it is related to an investigation of alternatives (Kang et al. 2010). Woodbury and Burrow (2006) extend this definition by the notion of working with alternatives which form a network, where it is possible to move towards previous design instances as well to find a way to the new ones. Bradner et al. (2014) put emphasis on the fact that exploration is a two-way interactive process whereby visual feedback plays an important role. This refers to an important design question “what if?” (Yamamoto and Nakakoji 2005) where, for example, changing values of parameters immediately changes graphical and analytical results. Summing up, the main goal of the “exploration stage” is to provide designers (or to allow to come up) with additional information (e.g. visual, statistical) to focus attention at solutions, which fulfil objective as well as subjective criteria (“subjective space”), and to help understand the root of decisions made. This activity could help in better understanding of the design space in general where, for example, if the designer come up with an understanding that some parameters constraints were set to tight, then he or she can go back to the “analysis stage” and relax them starting by this new iteration.

Finally, from the most promising solutions the “decision space” is formed and from these design instances a final decision will be made.

From overviewing the design stages important notions on exploration effectiveness can be concluded. First, it follows that effective exploration process includes a possibility to find the best compromises between objectives that are in conflict to each other. Second, we can judge the process by the designer’s willingness to prefer one design, among many (Bradner and Davis 2013). It is necessary to note that endeavor of designers will differ from one to another and, therefore, ensuring the greatest possible freedom of action is also essential.

RELATED WORK
Exploration systems exist in various fields, ranging from knowledge management tools to computer chip design (Boisot et al. 2007; Taghavi et al. 2009). These systems use various computational and visualisation methods, which are the topics of such disciplines as Big Data processing or Machine Learning. Essentially, they are aimed to find optimal performance omitting non-specifiable aspects (e.g. there is no interest for beauty in chip design).

Non-specifiable (i.e. subjective or non-computable, at least yet) aspects are individual for every designer. They reflect the creative, subjective nature of a design process, where final decisions which are made by the designer are based on their own view of the design problem and own image of
an ideal solution (Bradner and Davis 2013). Hence, there are many design space exploration applications oriented to not only applying data processing methods but also focused capabilities for individual design instance evaluation.

Mueller and Ochsendorf (2013) propose a framework based on evolutionary algorithms which allows to move fast to high performance instances by developing only promising branches of parameters values and avoid to stuck in a local optima. In each iteration the user can select instances relying on both visual and performance criteria for the next evolution step and perform direct manipulations on the design instance graphically and thereby obtaining live changes in the performance results.

Asl et al. (2014) suggest an integrated parametric (BIM-based) system that enables designers to explore design alternatives using a visual programming interface, while assessing the energy performance of the design instances in a graphical way. This system incorporates statistical methods to move to optima in case of contradictory performance criteria (Pareto optimality) combined with different visualisations (Scatterplot, Parallel Coordinates). These visualisations are enriched with object thumbnails superimposed on the plots in order to illustrate the link between the calculated optimal solutions and the appearance of the respective design instance. Similar approaches with implementing visualisation techniques were developed by Howes (2017) and Nagy et al. (2017).

Wortmann (2017) proposes a dimensionality reduction approach, which allows to represent a high-dimensional data in lower dimensions suitable for visual understanding by humans. Similar systems based on Self-Organizing Maps (Kohonen 1997) are described by Harding (2016) and Chen et al. (2013).

Methods that focused on the particular design instance are discussed in the work of Erhan et al. (2010). These methods aim to make the effects of change within a parametric model controllable, measurable and obvious for designers. Thereby emphasis is put on the problem when in order to observe changes in a design the user needs several switching between different views, which adds additional cognitive load.

While all the approaches are exploring the advantages of a particular selected methodology, they are suitable for particular problems, as there is never one ideal approach. Therefore, every time designer should think about selection of an appropriate method. Considering the vast amount of different design approaches and performance criteria this seems no wonder. Hence, it appears appropriate not to limit the exploration process by predefining one particular set of methods, but rather allow the designer to customize the exploration procedure for his own particular needs. In the following, we propose a framework based on a modular architecture that allows to work with a variety of possible exploration methods in one single (web-based [1]) environment.

### EXPLORATION FRAMEWORK CONCEPT

Our proposed framework is built upon the idea that there exist many exploration methods, which in the right combination are more effective than in isolation. Therefore, we follow a modular approach, whereby each module represents a certain exploration method. On a structural level, we distinguish three conceptual levels (see figure 3), reflecting the different requirements that we derived from the theoretical conceptualization and related work, presented in the previous two sections.

1. “Strategy level” - where the designer creates sequences of modules and develops own strategies for the “solution space” exploration and understanding;
2. “Processing level” - is about gradually narrowing the “solution space”, where the designer interacts with the possibilities of one specific exploration method inside a processing module;
3. “Instance level” - focuses on the evaluation of a particular design instance, where the designer uses methods to evaluate and interact with one design instance.
In the following, each level will be described in terms of its underlying conceptual ideas as well as with implementation details and examples of test case. The test case stems from an industry funded research (DIPLANNER) that aims to generate building volumes and floorplans for the semi-automated design of residential buildings. For the test case different algorithms for generating building volumes were applied to create 600 variants for an inner-city block. For each of these variants different features were computed, such as site coverage and floor area ratio (GSI and FAR), Surface / Volume Ratio, Shadow analysis (how much of the courtyard never gets sunlight) and a solar analysis (how much of the façade area gets at least 30 sun hours).

**Strategy level**
Essentially, the idea of the “strategy level” is to offer the designer a possibility manipulating an order of various processing modules (see next section) and link them between each other in order to use the output results for further processing with other modules. Node-links visualisation is suitable to support the idea of data pipeline (Munzner 2014) and meets the requirement, where there is the need of preserving of particular states and context (Bradner et al. 2014).

**Processing level**
Generally, a processing module can be anything that takes a set of design instances as input and processes the data in such a way that the output will be either altered, partitioned or reduced. The current state of the implemented version of the framework has five available modules to process data: Parallel Coordinates, Self-Organizing Maps (SOM), K-Means Clustering, Filtering and Table View.

The “filtering module” is the simplest one and allows to get a subset of instances by defining conditions on particular features and logically combine them (e.g. all instances with an energy demand < 100 KWh). Thereby, one retrieves a subset for a further processing. Similar functionality exists in the “table module” where the user filters data in columns and
could immediately see results in a tabular form.

The “K-Means clustering” is the module where the user specifies the number of expected clusters and then explores the clustered subsets in a desired visualisation block (see figure 6). It is useful to get insight into distribution of design instances by their performance similarity or to pre-process the data for a further separated processing.

The “Parallel Coordinates” and the “Self-Organizing Maps” (SOM) modules allow to visually explore and manipulate data design instances set. The “Parallel Coordinates” allows to observe patterns in a dataset, perform visual filtering and selecting of design instances. This visualisation also helps to observe that e.g. instances that have a high values in one performance criteria exhibit low values in another (see figure 5).

In the “Self-Organizing Maps” module (see figure 6) design instances are visualised on a two-dimensional map according to the similarity of their computable features. Depending on the selected features, different color mappings are available, in order to display clusters as well as information about minimal and maximal values distribution for a certain performance criteria. The “SOM” module furthermore allows to visually select single or multiple design instances. In different zoom levels the instances are represented either by a single point or by an image displaying a certain visual representation of the instance (e.g. floor plan or isometry). Each selection can be stored separately and thus is available as output of this module. Additionally, input data is also stored with information about cluster or selection grouping from a predecessor module(s) and can be used for visualisation of the map.

In order to provide additional degree of flexibility all modules can be opened in a separate browser windows to use brushing and linking techniques (Munzner 2014, p.267).

In our test scenario we start from a “K-Means clustering” module, where four cluster are created (see top in figure 6). Further, they are separately investigated in a “Table” module and three the most interesting clusters are selected to work with in the “Parallel Coordinates” and “SOM” modules. Then each cluster is explored in terms of tradeoffs between available features and their objectives. Figure 7 shows the maps where it is possible to find areas which fit objective criteria. At this step we selected design instances inside the cell and stored the selection for further processing.

Figure 5
The “Parallel Coordinates”module
**Instance level**

The idea behind the “instance level” is to offer the ability to work with a design object in detail (e.g. regarding its visual appearance) and to consolidate all introduced information during the design process (e.g. parameters values, performance features and user defined notes).

One of the concepts here, named “Annotation”, is aimed to provide abilities to introduce new information about an instance that the designer discovered during the exploration process. That is achieved by implementing such functionality as commenting to store designer thoughts (e.g. in a text form) and interpretation about particular design instance and its context, which could be used further in decision-making (Yamamoto and Nakakoji 2005). Codes, tags...
or labels could be used here for introducing user-defined classes and advance searching (Lösch et al. 2009). Further, connections to related instances can be created (e.g. for manually linking design instances where the designer thinks they somehow belong together). Additionally, visual notes and any other kind of information could be associated with an instance.

Another concept, “Comparison”, is about identifying similarities and differences in parametric and features space as well as in the visual appearance of design instances. Parametrical closeness could be determined through finding successors and predecessors during the generation process; closeness by performance features is the base for data processing modules; evaluating design instances based on their visual similarity is, for example, possible through image comparison (e.g. pixel wise comparison of building footprints) or volumetric analysis (Bustos et al. 2005). Right in figure 8 shows performing a search for similarly looking forms across the whole set of instances. In general, by this method a user can use any image e.g. hand drawing to perform such a query. Additionally, integration with modelling/analysis environment (and with stages conceptually) provides an instrument to interactively see the changes in the visualisation (e.g. on SOM map) by changing the value of parameters in a parametric module. That functionality extends the knowledge of designer about the way design and solution spaces are formed by performing sensitivity analysis.

The last concept, “Presentation”, serves for extracting and presenting information about a design instance, which can be important for decision-making. The functionality is to display available spatial visualisations, already introduced annotations, parametrical description and calculated feature information. In addition, various external references, like GIS data, simulations data and etc. can be added (see left in figure 8).

Right in figure 8 shows that the designer found an interesting for him shape in a low performant blue area but similarity search had discovered other similarly looking design instance in a higher performant red area. Left in figure 8 shows the design instance...
CONCLUSION AND FUTURE RESEARCH

In this paper we investigated ways to more flexibly integrate the design exploration process into the design process. We presented a conceptual framework prototype implementation with the modular approach, which provides the possibility to flexibly customize the exploration process, add new functionalities as well as gain advantages from their combination.

In the next phase of development, we are looking forward to extend the variety of available methods. Moreover, it is necessary to make the process of exploration inside the system easier by the even closer integration with the modelling/analysis environment. In the current state of development design instances are saved as a CSV file with computed features and separate image files which are loaded manually, that restricts sensitivity analysis by pre-generated design instances.

Another direction of a future development is a support of multistage iterative processes. Such a step will allow to track decisions and their relations from differently detailed models. For example, such functionality will support scenarios, where designer firstly iterates through a building form, then, based on a set of the most promising decisions, starts iteration on possible window layouts and so on.

Finally, the framework provides a sound basis for further research, such as studying the effectiveness of the exploration process. Therefore, user studies for exploring how designers (experts and novices) use this tool (which strategies do they apply, which methods are most useful, etc.) can be conducted.

REFERENCES


Boisot, M.H., MacMillan, I.C. and Han, K.S. 2007, *Explo- rations in information space: Knowledge, agents, and organization.*, Oxford University Press on Demand

Bradner, E. and Davis, M. 2013 ‘Design creativity: using pareto analysis and genetic algorithms to generate and evaluate design alternatives’, *CHI 2013*, Paris, France


Munzner, T. 2014, Visualization analysis and design., CRC Press


