Discovering Design Change Pattern Through Versioning

○ Verina Cristie[*1], Jason Lim[*2], and Sam C. Joyce[*3]

*1 PhD Candidate, Architecture and Sustainable Design, Singapore University of Technology and Design, M. Sc.
*2 Adjunct Assistant Professor,, Architecture and Sustainable Design, Singapore University of Technology and Design, Dr.
*3 Assistant Professor, Architecture and Sustainable Design, Singapore University of Technology and Design, Eng. D.

Summary: Despite the plenty of data collected through collaborative design exercises in pedagogy settings, very few of these data were utilized for further studies. This occurrence is in contrast with the software development settings where software repositories are often mined to find insights on programmer's software building pattern. In this work, we implemented exploratory time-series data analysis through design versions data collected from a parametric design workshop of 44 students in groups of five. A framework to discover design change pattern through in-between change count was developed. The result revealed three different change patterns the group exhibit: premature fixation, constant change, and last-minute work. Finally, it is found that constant change pattern corresponds to higher instructor-given final design computation score, as students were encouraged to explore sufficient design ideas in the workshop.

Keywords: Classroom Cloud Collaborative Tool; Parametric Design; Data Analysis; Design Exploration; Version Control.

1. Introduction

Expert designers' ideation processes are often associated with breadth-first exploration as opposed to novice designer' depth-first exploration[1,2]. While not a definite formula for better design outcomes, it is often seen as a good design strategy: to explore more design ideas in the early stage and not committing too early to one design idea, with ideas converging in the later stage of design. To evaluate design progression in a pedagogy setting, instructors would often require students to submit their 'learning journals'[3], or 'process book'[4] where the development and exploration of ideas are contained.

The idea of evaluating design progression has turned to exploring digital evaluation with the birth of the web in the 90s. Design studio turned to virtual design studio[5], and learning journal became the design progression data (design versions) recorded online. Phase(x)[6] and OpenD[7] exemplified this by allowing students to upload their designs (in image and text format) as they progressed in their designs. In such virtual exercise, learning and collaboration were often the focus. Students were to download and modify each other's designs in the spirit of collective authorship. In a more recent example in parametric design, an interactive design gallery was developed to facilitate saving and retrieving of design alternatives exploration[8].

In this paper, we will focus on recording and evaluating the design process in parametric design. Parametric design is a way of representing design intent by establishing the relationships of its design elements[9]. Due to this being on graphic interfaces parametric modelling is often called visual programming, a counter to software design's predominantly text based programming. Jabi[10] further iterated that software design concepts such as versioning and iteration are fundamental themes in parametric design. It is our aim in this work to evaluate the parametric design process through the design versions captured, similar to how code versions were evaluated to understand programmer's software building pattern.

How the design progression data could be recorded online and how it has the potential to be evaluated similarly to code repository evaluation had been introduced in this first section. The rest of this paper continues as follows: in the second section, we review related works in design process evaluation through data analysis. Our case study and its result will be described in section three. Design change analysis framework and its implementation in the collected data will be discussed in section four. Section five deals with the development of design entropy framework and further discussion of the various measures of the design process. Finally, in the concluding section, we summarize our findings and discuss directions for future works.

2. Related Works & Scope

To understand the design process, design activity data had been used in numerous design protocol studies[11,12]. Typically, designers were observed directly or recorded while designing and they were asked to think aloud so that their cognitive process could be matched with their design action. We differentiated this study by using design progression data instead of design activity data. Design progression data contains design artefacts such as sketch or model at different points of time throughout the design process.

A parametric design's artefacts are a parametric model, its input parameters, and a resultant geometric output. To measure change and variance between two or more models, Brown & Mueller[13] have developed a diversity metric; i.e: how diverse geometric outputs from a parametric model is. Davis[14], developed complexity and flexibility metric; i.e: how easily understood and modified a parametric model is. Both metrics, however, are static measures of 'fixed' models; whereas the creation of a design including a parametric model involves changes and edits over time. To understand this design process better, a time-dependent analysis is critically needed. Prior work by authors[15,16,17] has demonstrated how this time-dependent design progression data can be captured. This paper aims to investigate and develop frameworks to understand this data better.

Specifically, our research questions are:

- How do we detect and quantify the change in the collected time-series parametric design data?
- Can any pattern be found from the quantified change?
- What does the quantified change tell about the design's progression, and how does it relate to the final score?

We seek to answer these prescribed questions firstly by capturing the design progression data to then use them to develop the analysis framework, which will be described in section 4. In the next part, we will describe our data collection study case.

3. Design Workshop & Results

3.1. EXPERIMENT SETTINGS

Design progression data was collected in a 5-full-day workshop of undergraduate architectural computational design class. Sixty-four students were enrolled in the class, divided into 13 groups. Each group was tasked to design external façade based on the given design scenario, which was introduced on the first day. The final façade had to be aesthetically pleasing and at the same time adhere to site specific conditions such as sun direction and outside views. On the second day, a base parametric model file containing scripts to generate, modify, and evaluate façade surface was given. Students were to explore this model individually before discussing and continuing to develop the model as a group on the third though to the fifth day of the workshop. The GHShot Grasshopper plugin versioning tool[15,16], was used to record student's design progression throughout the workshop. At any point in the design development, students could send their current parametric model to cloud platform. By default, every model sent would be a continuation of its previous model. However, students could also specify if the current model sent was a variation/design alternative of the previous model sent. Establishing this continuation-or-variation was important to understand the overall design development (history) tree. At the end of the workshop, each group was to submit a design journal to summarize and reflect on their design journey. In addition, they would also need to explain important milestones in their design.

3.2. RESULTS

Out of the 13 groups, 9 groups were selected for further analysis (Figure 1). Four groups were not selected because there were not sufficient (less than four) parametric model versions sent to analyze their progression.

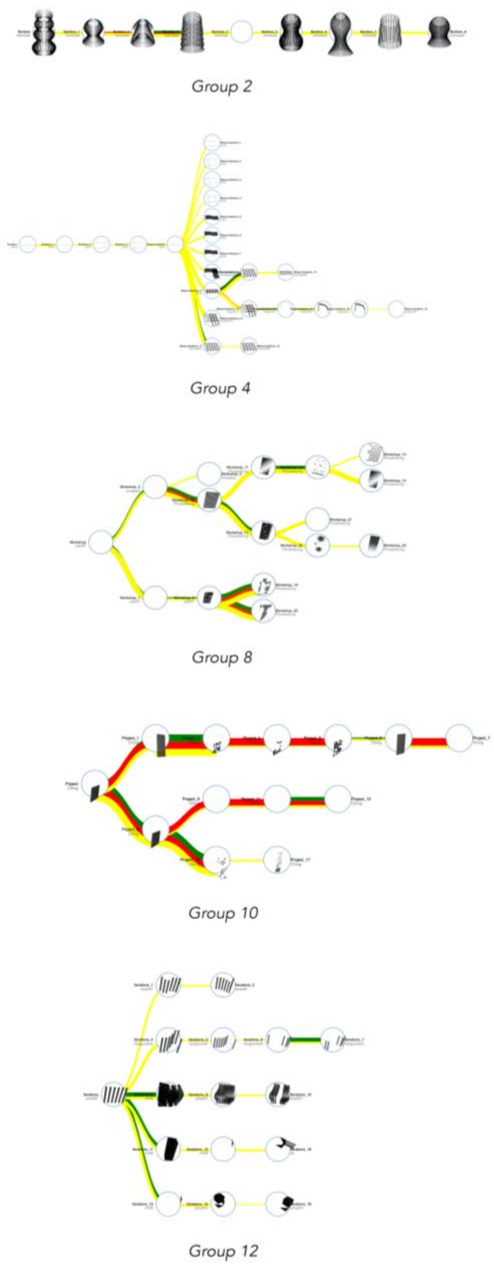

Figure 1. Selected Groups' Design Progression. Green, red, and yellow line represents the number of components added, deleted, and changed between versions.

4. Design Change Analysis

To analyze design progression from versions collected, firstly the design change must be detected. Change in each version is compared against its previous version. Once detected, the number of changed elements are counted from the start to the end of the design to see if we could gain some insight from the parametric model change activity.

4.1. CHANGE DETECTION AND COUNT

Each version contains its parametric model definition and geometrical output at the time it is sent to the server. In Grasshopper, designers interact with their parametric model by the use of visual components representing encapsulated computation typically geometry processes that take multiple inputs and outputs, these are connected by wires representing where the data flows and how indexing works for the model which is effectively a computer program. This parametric model definition can be saved in text-based eXtensible Markup Language (XML) format. With XML, each component in Grasshopper is represented in a 'chunk' of text, and each chunk contains information of the component's ID, type, attributes, and the ID of other components that are connected to it.

4.1.1 Change in Parametric Model Components

The XML text were then parsed for a list of components and their ID and attributes. IDs appearing only in the newer/subsequent design version were detected as newly added components, while IDs appearing only in the older version were detected as deleted components. There were also IDs appearing in both versions. If their attributes were different in the newer and older versions, these were detected as changed components. Otherwise, they were counted as the same components in both versions.

Parametric change score of a design version was formulated as the sum of the changed components count, deleted components count, and newly added components count. We did not use change percentage against the overall number of components as even one component change could affect the parametric model entirely. By using change count, higher change score could be expected when new ideas were implemented (many components were being added, and the old ones were deleted), as compared to lower change score when typically only input parameters of the model were changed.

4.1.2 Change in Code Based Components

In the studied workshop, students were encouraged to use scripting as part of the design exploration. In Grasshopper, this is possible by the use of GhPython Script component allowing for custom logic in a component. To further investigate the code based changes students did, we used a popular text comparison algorithm called Diff[18]. Diff library used in our analysis is taken from Google's Diff Match Patch [1]. It allowed us to know the total lines of text in the script that were same, new, or deleted. If a line of code was changed, it was counted as deleting the old line and adding a new line. Code change score was formulated as the sum of deleted lines count and newly added lines count.

4.2. CHANGE COUNT SUMMARY

Both change count in both parametric model components and code-based components were summed up to reflect the overall change in a particular design version. In Figure 2, parametric change count (in blue), code change count (in orange), and total change count (in green) for the different groups were visualized. Plotting changes in line allowed us to see different design iterations students went through during the workshop. When there was major design development or new ideas explored, the number of changes often spiked. The y-axes of the graphs were not having the same upper limit, as each group produces different parametric models. For example, in G10's graph, we saw a pattern of changes and milestone marking (vertical line) repeatedly.

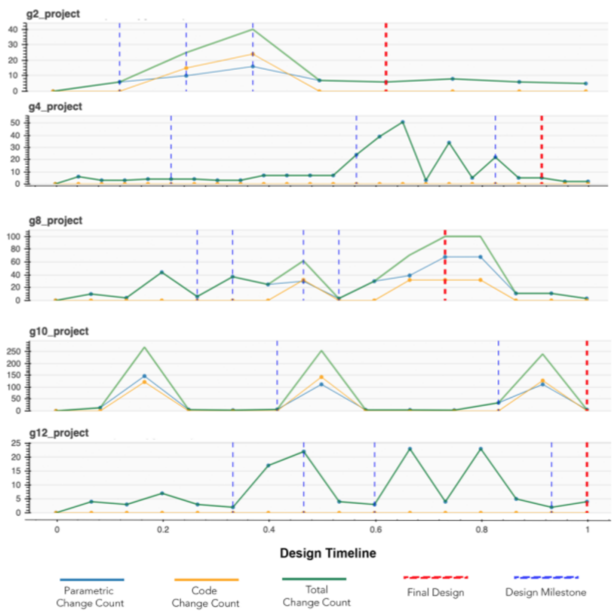

Figure 2. Change count plot across groups. Design timeline is represented from 0 to 1, 0 being the first and 1 being the last design version sent.

Parametric and code modification count also often spiked at the same time, signifying that both coding and parametric changes were employed to achieve student's desired geometric outcome. Overall, we observed three distinct strategies for geometric manipulation from the graph:

- **Parametric component modification only**: a common occurrence found across the groups, where we saw yellow line fall often flat to 0 counts. This strategy is exemplified most in in group G4 and G12.
- **Both parametric and code modification, with dominant coding strategy**: this could be found when a version's code change count is bigger than its parametric change count. An example could be found at G2's 4th version.
- **Both parametric and code modification, with dominant parametric strategy**: this occured in many of the design versions; in general, students did less code modification than parametric model modification throughout the design process (orange line is typically located below the blue line). A clear example could be seen throughout G8's design versions.

4.3 DESIGN CHANGE PATTERN

Given the design timeline and its resulting change count, we came up with a cumulative design change graph to compare the journey of changes each group went through. We were interested to find further:

- Which group did more changes as compared to the rest? How did each group's change performance when compared with the rest?
- How did these changes happen? Were there more changes in the beginning or at the end?

The highest cumulative change count came from G9 with over more than 1500 change counts. G1, G11, and G10 has more than 500 change counts. G8 has around 500 changes, and the rest of the group has less than 300 changes. To understand if more works are done in the start, middle, or end of design timeline, the change counts were normalized from 0 to 1 (see Fig. 3). Further, it could also be observed how different groups have different line progression 'smoothness'. G11, G1, and G9 have sudden jumps in their progression, especially towards the end of the design timeline. This means more works were done at the end.

We identified three design change patterns from the normalized cumulative change count (see Fig. 4):

- **Premature Fixation (PF)**: more changes are done at the beginning of the design and there aren't much design developments after; such was shown by G2.
- **Last-minute work (LM)**: changes, or work to develop design seems slow as design progress and suddenly there is a drastic change at the end of the design timeline; such was shown by G1, G9, and G11.
- **Constant change (CC)**: continuous and consistent change throughout the design timeline; such was shown by the rest of the groups.

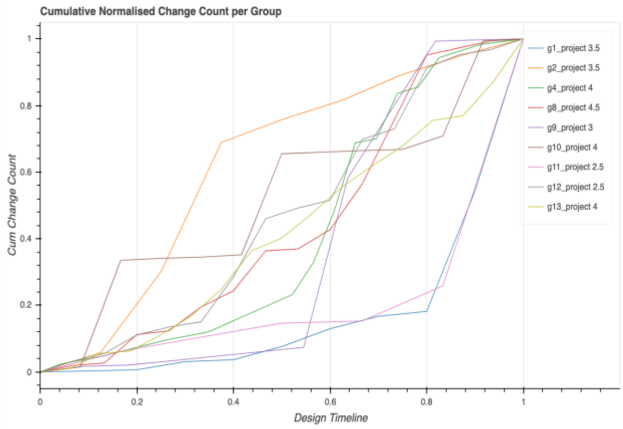

Figure 3. Groups' Normalized Change Count

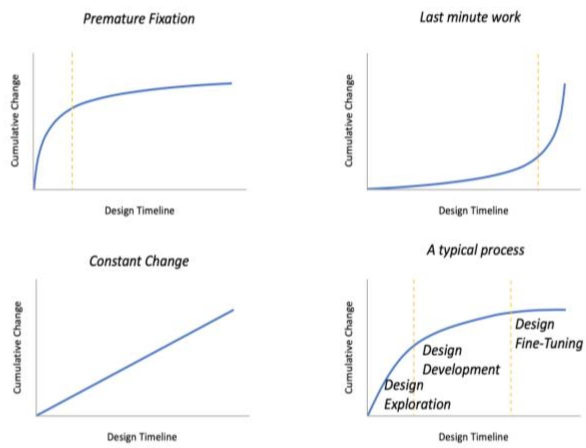

Figure 4. Design Change Activity Pattern

4.4 CONNECTION TO THE FINAL DESIGN COMPUTATION SCORE

At the end of the workshop, two score were given to each group, final design aesthetic score and final design computation score. For the purpose of the following discussion, we only take account of the design computation score, as aesthetic score is subjective and might not correspond to the parametric design process the students went through. The criteria for the computation score was sufficient design ideas and iterations were explored, especially as performed using the Grasshopper parametric software.

As summarized in Table 1 below, those with final instructor's given computation design score 4 and above are often found to have Constant Change pattern (CC), except G12. G8, which has the highest computation score (4.5), has rather a smooth linear progression with higher gradient. It can be seen evidently in Fig. 1 that G8 had plenty of design ideas explored. G2's Premature Fixation (PF) were given 3.5 score. Last-minute work (LM) were given score 3.5, 3, and 2.5 respectively, swinging towards the lower spectrum of the scores.

Table 1: Group's Design Progression Summary

group	total sender	total versions	design pattern	1st miles	fixation	final score
G1	1	11	LM	3	early	3.5
G2	1	9	PF	2	early	3.5
G4	4	24	CC	6	mid	4
G8	3	16	CC	5	mid	4.5
G9	2	12	LM	10	late	3
G10	1	13	CC	6	mid	4
G11	1	6	LM	3	early	2.5
G12	5	16	CC	6	mid	2.5
G13	4	17	CC	7	mid	4

Thus, well-performing groups in the workshop tended to have linear constant changes showing consistant and persistant effort. We speculate that this is partly due to the short duration of the design workshop (3 days for the group design), that constant change and iterations were encouraged so that more ideas could be explored and developed. In [19], it was mentioned that there are two different approaches good designers typically go through: either deliberating on a series of alternative solutions followed by a series of refinement and selection, or single idea with continuous revolution and evolution. It is important for designers to continue putting effort in the design process as it will guide the change direction. Such notion is consistent with Schon's[20] reflection in action, where professionals were found to receive feedback (reflection) to their thinking process as they perform problem-solving action in their current projects. Merely waiting for an idea to appear is unlikely to be successful.

The constant change appears to be reflective of the effort given in exploring the design. In a more conventional design scenario, we assume a plot where change is highest in the beginning during the design exploration phase, tapering to lower during design development, and eventually lowest during design fine-tuning as design change plateaued at the end (see Fig. 4, bottom right chart). Cross[2] mentioned that expert designers will do breadth-first exploration before going to dive into a depth-based exploration of a particular design. In other words, designs do diverge first before they converge. Hence, in a longer duration design development, we expect to see a different type of cumulative graph.

Lastly, to provide a deeper understanding of the design process, we also asked each group to specify which design version were their main design milestones, and which was the final design. This was visualized using blue (milestones) and red (final) vertical lines in Fig. 4 above. We identified at which nth design first milestone occurred and put it on the summary table under the first milestone order column. Based on this, we categorized if the design milestone was set (fixed) early, middle, or late. 1-4 is categorized as early, 5-8 as mid, and above 8 as late. We found an extreme case in G11 where its milestone was fixed early and final computation score was low (2.5). Three groups who set their milestone in the middle also had a rather high final computation score. Despite this, there isn't any conclusive relationship between a group's milestone fixation and its final score. Based on the literature, we would suggest having a milestone not too early, as more design exploration should typically be done before having a fixed idea.

In section three, we have introduced the design change framework, and in this section, we further elaborated how the change frameworks can be elaborated to discover design change pattern during the design process. Further, the changes were discussed in relation to the final design score and the design milestone. This change framework is novel in its implementation in the parametric design process to the best of our knowledge.

5. Conclusion and Future Work

In this paper, we have demonstrated how captured design progression data can be analyzed to interpret changes each group went through in a design workshop. Applying time-series analysis, a change metric was established, by firstly calculating change count of a design version against its previous version in both the parametric model and Python script lines in Grasshopper XML file. This quantified change count, when normalised, established each group's design change pattern, which can be categorised as premature fixation, constant change, and last-minute work. When connected to the groups' final design computational score, it can be observed that groups with constant change pattern tend to have a higher design computation score as compared to thos from last-minute work group. This means that score can be a proxy of amount of design explorations done by the groups. On the other flip side of the coin, it also means that generally the score was given objectively, rewarding groups with more designs explored with higher score. As a teaching aid, the versioning tool and the change framework can be combined as a method to highlight students whose design exploration might be sub-optimal and needing nudging to explore more designs.

For future experiments, we are aiming for a more accurate design version timeline. This could be achieved by either allowing design version timestamp change or enforcing a stricter rule to immediately sent version after each design exploration. Such practice will allow a better analysis of how each group reached the design milestone and differentiating versions that took longer or shorter time to achieve. In addition, we could also establish a change rate metric, where change count can be measured against the time taken to establish a particular design; and thus could potentially reveal greater depth in designers' working pattern.

Lastly, other than using change metric as a proxy of design space explored, we speculate that such automatic change measurement could probably be beneficial for assisting both automatic[21] and user-directed parametric design exploration[22]. Change metric could serve as an internal threshold for the parametric component arrangement or informing designers of the parametric similarity of their design versions.

Endnotes

1. https://github.com/google/diff-match-patch

References

1) Eastman, C., Newstetter, W.C., and McCracken, M. W.: 2001, Design knowing and learning. Elsevier.
2) Cross, N.: 2004, Expertise in design: an overview. Design Studies, 25(5), 427–441.
3) Roberts, A, and Yoell, H.: 2009, Reflectors, converts and the disengaged. Journal for Education in the Built Environment 4.2, 74-93.
4) Brunner, L. A.: 2009, A record of the design process. Art and Design Conference Proceedings: 192-1.
5) Wojtowicz, J.: 1994, Virtual Design Studio. Hongkong University Press.
6) Hirschberg, U. and Wenz. F.: 2000. Phase (x)—memetic engineering for architecture. Automation in construction, 9(4), 387-392.
7) Mark M., Bielaczyc K., and Huang, J.: 2005, OpenD: supporting parallel development of digital designs. User eXperience, 25-es.
8) Mohiuddin, A., Woodbury, R., Narges, A., Mark, C., and Völker, M.: 2017 A Design Gallery System: Prototype and Evaluation. 2017, ACADIA77, Cambridge, MA, 414- 425
9) Woodbury, R.: 2010, Elements of Parametric Design. Routledge.
10) Jabi, W.: 2013, Parametric design for architecture. Laurence King Publishing.
11) Suwa, M., Purcell, T., and Gero, J.: 1998, Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions. Design studies, 19(4), 455-483.
12) Yu, R., Gu, N., Ostwald, M., and Gero, J.: 2015, Empirical support for problem–solution coevolution in a parametric design environment. AI EDAM, 29(1), 33-44.
13) Brown, N. C. and Mueller, C. T.: 2019, Quantifying diversity in parametric design: a comparison of possible metrics. AI for Engineering Design, Analysis and Manufacturing 33(1):40–53.
14) Davis, D.: 2013, "Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture." PhD dissertation, RMIT University.
15) Cristie, V. and Joyce, S. C.: 2019, 'GHShot': a collaborative and distributed visual version control for Grasshopper parametric programming. 37th eCAADe and 23rd SIGraDi, Porto, (3)35-44.
16) Cristie, V. and Joyce, S. C.: 2018, GHShot: 3D Design Versioning for Learning and Collaboration in the Web. Extended Abstracts of the 2018 CHI Conference. ACM.
17) Cristie, V. and Joyce, S.C., 2017, September. Capturing And Visualising Parametric Design Flow Through Interactive Web Versioning Snapshots. International Association for Shell and Spatial Structures Annual Symposia, No. 5, 1-8.
18) Myers, E. W.: 1986, An O (ND) difference algorithm and its variations. Algorithmica 1, no. 1-4: 251-266.
19) Lawson, B.: 2006. How designers think: The design process demystified. Routledge.
20) Schön, D. A.: 1983, The Reflective Practitioner: How Professionals Think in Action. Ashgate Publishing.
21) Harding, J. E. and Shepherd, P.: 2017, Meta-parametric design. In Design Studies 52, 73-95.
22) Nazim, I. and Joyce, S. C.: 2019, User Directed Parametric Design for Option Exploration. 39th ACADIA