

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/352438663>

Capturing and Evaluating Parametric Design Exploration in a Collaborative Environment: A Study Case of Versioning for Parametric Design

Conference Paper · June 2021

CITATIONS

0

READS

65

3 authors:



[Verina Cristie](#)

Singapore University of Technology and Design

13 PUBLICATIONS 76 CITATIONS

[SEE PROFILE](#)



[Ibrahim Nazim](#)

Singapore University of Technology and Design

11 PUBLICATIONS 30 CITATIONS

[SEE PROFILE](#)



[Sam Conrad Joyce](#)

Singapore University of Technology and Design

30 PUBLICATIONS 143 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Cooler Calmer Singapore - initial phase [View project](#)

CAPTURING AND EVALUATING PARAMETRIC DESIGN EXPLORATION IN A COLLABORATIVE ENVIRONMENT

A study case of versioning for parametric design

VERINA CRISTIE¹, NAZIM IBRAHIM² and
SAM CONRAD JOYCE³

^{1,2,3}*Singapore University of Technology and Design*

¹*verina_cristie@mymail.sutd.edu.sg* ^{2,3}*{nazim_ibrahim|
sam_joyce}@sutd.edu.sg*

Abstract. Although parametric modelling and digital design tools have become ubiquitous in digital design, there is a limited understanding of how designers apply them in their design processes (Yu et al., 2014). This paper looks at the use of GHShot versioning tool developed by the authors (Cristie & Joyce, 2018; 2019) used to capture and track changes and progression of parametric models to understand early-stage design exploration and collaboration empirically. We introduce both development history graph-based metrics (macro-process) and parametric model and geometry change metric (micro-process) as frameworks to explore and understand the captured progression data. These metrics, applied to data collected from three cohorts of classroom collaborative design exercises, exhibited students' distinct modification patterns such as major and complex creation processes or minor parameter explorations. Finally, with the metrics' applicability as an objective language to describe the (collaborative) design process, we recommend using versioning for more data-driven insight into parametric design exploration processes.

Keywords. Design exploration; parametric design; history recording; version control; collaborative design.

1. Introduction

With the rise of the web in the 90s, the concept of the Virtual Design Studio (Wojtowicz, 1994) was born into the architectural pedagogy, where design projects were done over the network. During subsequent implementations of this concept, learning and collaboration was the focus, such as ETH Zurich's phase(x) (Hirschberg & Wenz, 1997), Harvard GSD's OpenD (Meagher, et al., 2005) or AA Design Research Lab's Collaborative Distributed Learning (Steele, 2006). Students were to exchange design works periodically, modify them creatively in a collective authorship scenario on a common web platform. However, such platforms often remained one-off technological proofs-of-concept, and lacked further investigation into the design process (Achten, 2009).

Independently, various design process studies have been manually performed to help better understand how designers from different fields (Lawson, 2006) or of different expertise (Eastman et al. 2001) think, including looking at the breadth and depth of the design exploration (Cross, 2004). The observants often had to perform think-aloud protocols while designing. With on-site observation, video and audio recording manually segmented and coded by experts in the field, scalability and generalisability issues were raised. Thus, in this paper, it is our aim to utilise data collected from such collaborative web platforms to understand the design process better, rather than relying on traditionally manual data collection and processing.

2. Data Collection



Figure 1. GHShot Design Versioning Grasshopper plugin.

In order to capture parametric design process data, GHShot (Cristie & Joyce, 2018; 2019), a design versioning plugin for Grasshopper previously developed by the authors, was utilised. GHShot works similar to the widely used code versioning tools (like Git) but on parametric model visual-coding rather than the usual software text-based coding. Critically for this work using GHShot, designers can send their design versions at any development point to the cloud. Consider the above sphere scenario in Fig. 1. The following data is captured upon sending:

Table 1. Data captured in each design version (based on simple sphere scenario).

Design Version Data	Meta-Data
<ul style="list-style-type: none">• Parametric model definition <i>Sphere and Slider Components and Link</i>• Parameter value <i>Sphere radius – slider value</i>• Geometry <i>Sphere mesh</i>	<ul style="list-style-type: none">• Design version ID <i>Auto-updated ID</i>• Time-stamp• Notes <i>If any, about the design</i>• Parental information <i>Previous version's ID</i>

2.1. EXPERIMENT SETTINGS

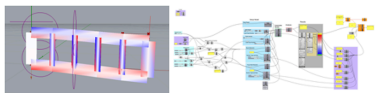


Figure 2. Base Truss Parametric Model given to the cohorts.

Three cohorts of second-year undergraduate architectural students (G1, G2, G3) in a structural design course were given a base truss parametric model (see Fig. 2) to collaborate on, to introduce them to different parametric structural typologies.

The two groups in each cohort were told to capture their design exploration process with GHShot and different instructions were given to see how it could affect the design process. While we tried to keep a similar timeframe over the cohorts, each year's ad-hoc scheduling often affected the experiment settings.

Table 2. The different experiment settings for each cohort.

	Basic Settings	Group A (user count)	Group B (user count)
G1	Given as a homework, continued in the class	No minimum design to submit (27)	Minimum 5 designs to submit (26)
G2	2 hour design exercise in the class, 3x30 minutes design block with design rating in between	Each design block, the students were free to continue their designs or their peers' (13)	Each design block, the students were to modify only the top 3 rated designs (12)
G3	1 week homework	Same instructions for both groups. Students were given extra points if they can make a cantilever model (14 and 13).	

3. Data Processing: Metrics Development

While in the past evaluating design processes required subject experts, recent developments has shown the ability to perform automated quantitative evaluation of parametric model's geometric diversity (Brown & Mueller, 2018), and flexibility (Davis 2013). In our context, we are interested in being able to meaningfully understand parametric design beyond its individual model level and probe further into its development process and collaboration, especially with the cohorts' design versions data. Both a macro (overall development) and micro (detailed, versions-based) approach are used for evaluation.

3.1. MACRO VIEW: EVALUATING DESIGN PROCESS THROUGH DESIGN TREE DEVELOPMENT

From helmet evolution in the centuries (Dean, 1915) to Latham's computer evolutionary art (Todd & Latham, 1992), to Tsunoda & Sakai's (2015) human-bot collaborative 3D house plan, traditionally design history has been represented as a tree. The diverging of design ideas and its development iteration is comparable to a tree's breadth and depth. To further characterise the design history tree captured, we use the following metrics:

1. Number of nodes/design versions (*nNode*). This number is an indication of the total amount of design options explored.
2. Branching complexities (*branchComplexity*). Being able to change the direction of one's thinking and generate more ideas is considered a characteristic of a creative thinker (Lawson, 2012). In a design version history tree, a node (design version) is considered branching if it has at least two child nodes. *branchComplexity* is calculated by aggregating all branching nodes in the tree. If there is no branching at all and the process is linear, *branchComplexity* is 0. Whereas, more branching will produce higher *branchComplexity* value.
3. Maximum/Average tree depth (*max_treeDepth*, *avg_treeDepth*). Although merging (convergence) operations is not currently captured; a linear continuation - the tree depth, could be a measure of how developed an idea is.
4. Collaboration Score (*collabScore*). This is measured by the ratio of the number

- of nodes derived from someone else's version / overall nodes ($nNode$).
5. Number of distinct design ideas explored ($nIdeas$). A design version is considered as having a distinct design idea if the geometry output is perceived as substantially different from other versions. To do this automatically, Keras (Chollet & others, 2015) deep learning model was used to extract features from the 2D image of the geometry model. Based on the features, elbow method (Thorndike, 1953) was used to determine the optimum number of clusters (k) and clustering is performed using K-means (Lloyd, 1982).

3.2. MICRO VIEW: METRICS OF A DESIGN VERSION AND ITS CHANGE

To quantify each design version, we use the Davis' (2013) existing metric for the parametric model (1 and 3) and Globa's (2016) for the geometry (4):

1. Number of Components in Parametric Model ($nComp$). This is the immediate proxy for the size of the parametric model.
2. Number of Unique Components ($nUniqueComp$). This is used as a proxy of how versatile a designer is and how unique a design is. We assume that expert designers would know how to use more component types to create more diverse designs.
3. Parametric Model Complexity ($graphComplexity$). Based on the number of link and components in the model, this is a measure of the amount of work to understand the parametric model.
4. Number of Meshes in the Geometry ($nMesh$). This is used as a proxy for the geometric complexity of the generated model.

Further, we look at design versions as time-series data to analyse the design change process, to see if we can learn any general or distinct patterns. Each design version is compared to its parent (previous) version. In every version data, there are three components: parametric model definition (*XML* string), parameters and performance values (as key-value pairs), and geometric output (3D objects). Upon observing that the participants were not taking performance values into account and were more interested in exploring visually unique geometrical shapes (topology), we decided to disregard performance differences in this experiment.

3.2.1. Parametric Change ($\Delta param$)

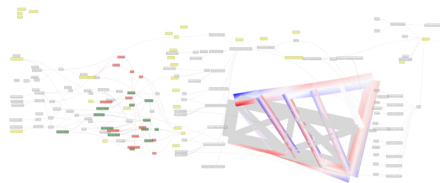


Figure 3. To change from original (grey shaded) to current geometry (coloured), parametric components were deleted (red), added (green), and changed (yellow).

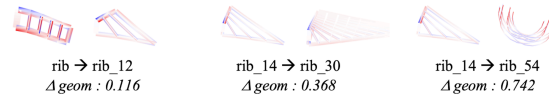
As a parametric model definition is XML text-based, we performed parametric model 'Diff'-ing, inspired by Diff (Myers, 1986), the standard software code

versioning's difference detection algorithm. As a parametric model's building block is its components, $\Delta param$ is defined as the total number of components added, deleted, or changed in its attributes.

3.2.2. Geometric Change ($\Delta geom$)

Geometric change can be the extent of design space explored (Brown & Mueller, 2019) and is often attributed as a measure of design creativity as it means designers are not fixated on a particular design (Nathan, 2015). To measure this, Hausdorff distance calculation from Meshlab library (Cignoni et al., 2008) was used. This distance ($\Delta geom$) is defined as the maximum distance of a set to the nearest point in another set. Hence, the higher the distance value is, the more dissimilar the two geometries are (see Table 3). The $nMesh$ metric mentioned previously was not used because the change in the mesh count does not necessarily represent the change geometry topology.

Table 3. $\Delta geom$ values for three level of geometric changes.



4. Results & Discussion

Below we present our findings in question and answer format for easier discussion.

4.1. HOW DO THE METRICS FARE ACROSS THE COHORTS?

- *nNode*: G1 has the largest nNode, as it had double the student size compared to G2 and G3. On average, the number of design versions submitted per student for all groups is between 3-5. G1-B had more design versions as compared to G1-A, as the students were required to submit minimum 5 design versions each.
- *branchingComplexity*: for all groups, it ranged between 5-6, except for G2-A where it is 8. This higher value is encouraged by the experiment settings where the students had to continue their own or other's design in the 30 minutes iteration, as compared to the other groups where there was no time limit. It is also higher than G2-B as G2-B can only choose the highest rated versions.
- *maxtree Depth* : The longer development time in G1 and G3 contributed to maxtreeDepth as high as 10 in G1-A and G1-B, and 8 in G3-A, as compared to G2-A's 6 or G2-B's 7. the *avg_treeDepth* of G1, G2, and G3 are 4.5, 3.3, and 4 respectively.
- *graphComplexity*: from the radar plot, it can be seen that G1 has a wider range of *graphComplexity* as compared to its initial value. As it has the lowest *nComp* and only 5 *nUniqueComp* related to its topology, it appears that its base model simplicity (and also the longer duration of exploration) gave room for more complex modification.

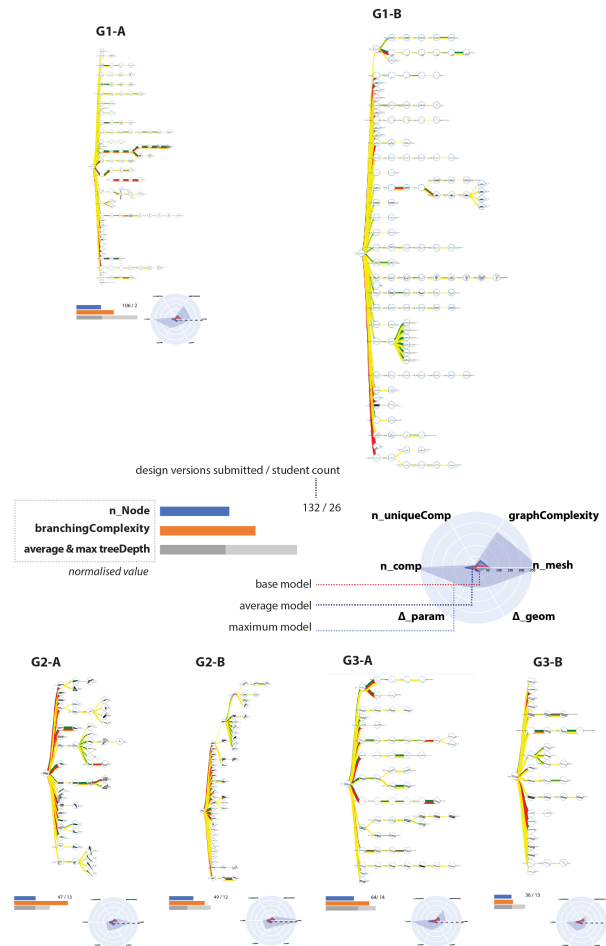


Figure 4. Design change tree and metrics visualisation from 3 cohorts. Radar areas are indicative of the initial metrics value and the extent of the students design space exploration.

4.2. WHAT ARE THE DESIGN CHANGE PATTERNS OBSERVED FROM THE DESIGN CHANGE TREE?

Yellow, green, and red lines and their thickness in Fig. 4's design change tree represents the count of modified, added, and deleted components respectively. Many versions submitted early in the tree has more yellow lines, signifying students started exploring by changing components' parameters. Some students also started adding and deleting components (creation modification) right away. Either they had an idea right off the bat, or the initial parametric changes were not uploaded. As the design progressed, we can observe two distinct continuous development paths: (1) refining (yellow lines only), or (2) idea exploration (thick red and green lines across). For many design paths, it is a combination of these.

4.3. ON N_IDEAS METRIC, HOW DOES AUTOMATIC DEEP LEARNING IMAGE CLUSTERING FARE COMPARE TO MANUAL REVIEWERS?

Figure 5 displays the resulting clusters taken as *nIdeas* metric on G2-B's dataset (based on the method explained in 3.1 above). There is no significant 'elbow' or turning point from the image features can be observed, implying that no optimum number of clusters were identified given the maximum of 20 clusters. It is possible that the images were hard to cluster - many individual unique design idea does not belong to any clusters. While all reviewers could agree that most design ideas can be found in G1, followed by G2 and G3, the number identified varied a lot depending on the reviewer (see Table 4), confirming the challenge in clustering even when performed manually.

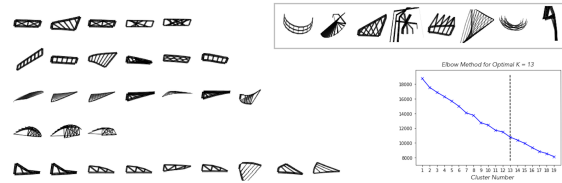


Figure 5. G2-B's 13 clusters of design ideas from the design versions.

Table 4. Number of design ideas found by reviewers and automatic image clustering.

	G1-A	G1-B	G2-A	G2-B	G3-A	G3-B
Reviewer 1	19	23	17	13	8	6
Reviewer 2	71	70	43	29	33	18
Reviewer 3	35	37	31	15	13	8
Image Clustering	13	14	13	11	6	8

4.4. ON COLLABORATION: ARE THERE ANY DIFFERENCES WHEN STUDENTS MODIFIED THEIR OWN DESIGNS VERSUS OTHERS'S ?

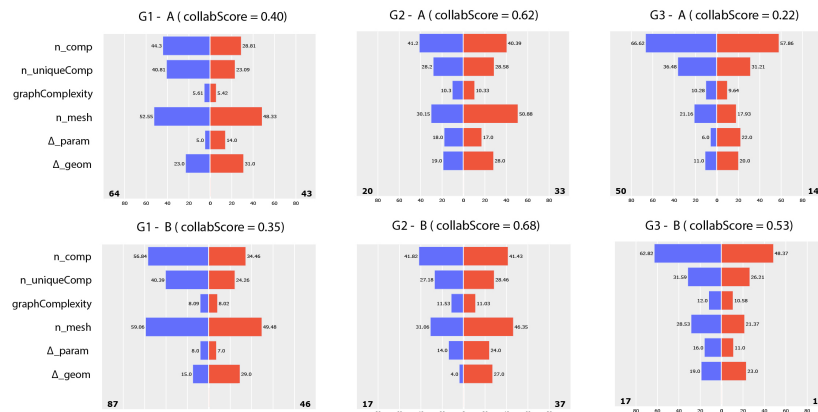


Figure 6. Metrics comparison of the cohorts. Blue: own design modification, red: other's.

- Without a specific experiment setting where they had to modify the highest rating designs - which might not be their own (G2-B case), students mostly tend to modify their own design versions (lower *collabScore*).
- As students work on their own designs, they tend to create and modify larger models (higher *nComp* and *nUniqueComp*). This is observed more significantly in G1 than G3. Perhaps when modifying other's models, it is easier to modify simpler models, such as the earlier developed models.
- *nComp* and *nUniqueComp* between self and others are comparable in G2. We suspect this is due to the half-hour time constraint which limits how much can be done.
- Despite this, *nMesh* is higher when other's models are modified in G2. It is possible that time constraint factor plays a part in a quicker idea exploration (White et al., 2010), and that students achieved higher $\Delta geom$ by simply modifying sliders or parameters (will be further discussed in 4.6).

4.5. ARE THERE ANY DIFFERENCES IN HOW EACH STUDENT MODIFIED THEIR DESIGNS ?

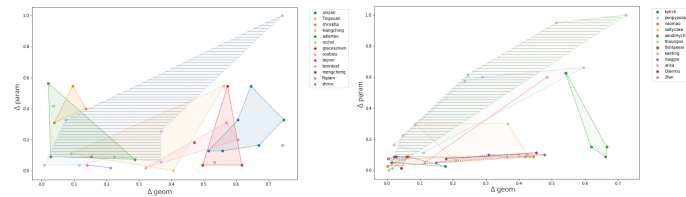


Figure 7. G2's design versions' $\Delta param$ and $\Delta geom$ plotted and connected per user. Each user occupies a 'zone'.

We plotted $\Delta param$ and $\Delta geom$ values from each user in G2 using a different colour (see Fig. 7). When each user's data points are connected, they occupied certain 'zones' in the plot. Some users' 'change zone' spanned to cover almost the entire plot, while some only covered small area in the plot. The data points signify specific parametric models modifications each user is most familiar with, resulting in various geometric change outcomes. We hypothesise that the bigger the zone spans represents users with higher parametric modelling proficiency and thus a wider range of design exploration; such as shown by G2-A's *Tingxua* (left) and G2-B's *tnaungo* (right) (shaded in Fig. 7 above).

4.6. HOW CAN WE CATEGORISE THE TYPES OF DESIGN CHANGES?

Based on $\Delta param$ and $\Delta geom$'s value, four categories were recognised (see Fig 8 below). This categorisation could potentially help design researchers to better understand users' design modification behavior and designers to be more structured/intentional in their modification. For example, if high geometry change value is a goal, an automatic design assistant can nudge them towards that.

1. High $\Delta param$ & low $\Delta geom$: Despite the high count in parametric changes, the geometry did not change significantly. It is possible that the design direction

was unclear, or it is intrinsic that the target design geometry does not vary much geometrically.

2. High $\Delta param$ & high $\Delta geom$: High count of parametric changes (often signified by a mixture of adding, deleting, and changing component modification) and the geometry drastically changes as well. This typically happens when a new design direction is explored.
3. Low $\Delta param$ & low $\Delta geom$: Only a few components change, typically this happens when a user wants to understand how geometry changes if a parameter attribute is modified.
4. Low $\Delta param$ & high $\Delta geom$: Despite the lower count of parametric changes, the geometry changes drastically, revealing parametric ‘surprises’ (Woodbury, 2010), and thus, a potentially interesting design direction.

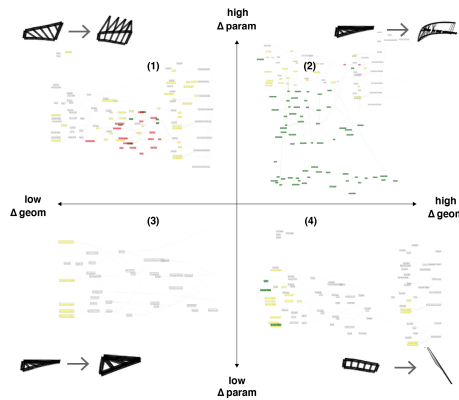


Figure 8. Four change clusters identified by $\Delta param$ and $\Delta geom$.

5. Summary and Conclusion

In this paper, we have demonstrated how collaborative parametric design exploration process can be (1) captured through the practice of versioning and (2) interpreted through the proposed metrics. At the macro level, history tree-related metrics such as the number of nodes, branches, and depth of the tree are used as proxies to describe the design space explored in terms of its size, diversity, and development of ideas. At the micro-level, design change metrics for both parametric and geometry models were established to evaluate and categorise the modifications performed. For example, the change metrics were useful to see how some students are more versatile in their designs through the wider range of metrics extracted from their design versions. In the collaboration context, time appears to be a crucial factor: little difference was observed in the change metrics when one was modifying his/her personal design vs others' in the shorter given time. In contrast, larger and more complex parametric model modifications were observed with own models when longer time is allowed.

Finally, while in this paper we have started utilising data to develop metrics towards understanding the design process better, the availability of such design

progression data opens the door to wider possibilities for digital and parametric design. In software engineering, where versioning was initially used, version data is also used to identify which programmers introduce more bugs (Kim et al., 2006). A similar approach can be used to determine which designers contribute significantly to design development. For example, in a pedagogy context, this can be used to support students who are lagging and support them. Data collected from multiple years can inform teachers of unique or perhaps shared struggles students face in learning parametric design. Additionally, as the field of architecture continues to adopt various AI technologies, we believe that capturing design process data can play a significant part in bridging the cognitive gap towards building autonomous AI design assistants.

References

- Achten, H. and Beetz, J.: 2009, What happened to collaborative design?, *eCAADe*.
- Brown, N.C. and Mueller, C.T.: 2019, Quantifying diversity in parametric design: a comparison of possible metrics., *AI EDAM*, **33**(1), 40-53.
- Chollet, F.: 2015, “Keras” . Available from <<https://github.com/fchollet/keras>>.
- Cignoni, P. and al, initials missing: 2008, “Meshlab: an open-source mesh processing tool” . Available from <<https://github.com/cnr-isti-vclab/meshlab>>.
- Crilly, N.: 2015, Fixation and creativity in concept development: The attitudes and practices of expert designers., *Design Studies*, **38**, 54-91.
- 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*.
- Cristie, V. and Joyce, S.C.: 2019, ‘GHShot’: a collaborative and distributed visual version control for Grasshopper, *37th eCAADe and 23rd SIGraDi*.
- Cross, N.: 2004, Expertise in design: an overview, *Design Studies*, **25**(5), 427-441.
- Davis, D.: 2013, *Modelled on software engineering: Flexible parametric models in the practice of architecture*, Ph.D. Thesis, RMIT.
- Dean, B.: 1915, An Explanatory Label for Helmets, *The Metropolitan Museum of Art Bulletin*, **10**(8), 173-177.
- Eastman, C., Newstetter, W. and McCracken, M.: 2001, *Design knowing and learning: Cognition in design education*, Elsevier.
- Globa, A.A., Donn, M. and Ulchitskiy, O.A.: 2016, Metrics for measuring complexity of geometric models, *Scientific visualization*, **8**, 74-82.
- Kim, S., Zimmermann, T. and Whitehead, E.: 2006, Automatic Identification of Bug-Introducing Changes, *21st IEEE on Automated Software Engineering*.
- Lawson, B.: 2006, *How designers think: The design process demystified*, Routledge.
- Lloyd, S.P.: 1982, Least squares quantization in PCM, *Information Theory*, **28.2**, 129-137.
- Meagher, M., Bielaczyc, K. and Huang, J.: 2005, OpenD: supporting parallel development of digital designs, *Proceedings of Designing for User eXperience*.
- Myers, E.W.: 1986, AnO(ND) difference algorithm and its variations, *Algorithmica*, **1**, 251-266.
- Sakai, Y. and Tsunoda, D.: 2015, Decentralized Version Control and Mass Collective Collaboration in design, *Proceedings of eCAADe 33*, 207-214.
- Steele, B.: 2006, The AADRL: Design, Collaboration and Convergence., *AD*, **76**(5), 58-63.
- Thorndike, R.L.: 1953, Who Belongs in the Family?, *Psychometrika*, **18**(4), 267-276.
- Todd, S.P. and Latham, W.: 1992, *Evolutionary art and computers*, Academic Press, Inc..
- Wenz, F. and Hirschberg, U.: 1997, Phase (x) Memetic Engineering for Architecture, *Proceedings of the 15th ECAADe Conference*, Vienna.
- Wojtowicz, J.: 1995, *Virtual Design Studio*, Hong Kong University Press.
- Yu, R., Gero, J. and Gu, N.: 2014, Cognitive effects of using parametric modeling by practicing architects: a preliminary study., *Proceedings of CAADRIA 2014*, 677-686.