JOURNAL OF LEARNING ANALYTICS

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

Analyzing Engineering Design through the Lens of Computation

Marcelo Worsley and Paulo Blikstein
Stanford University

Abstract: Learning analytics and educational data mining are introducing a number of new techniques and frameworks for studying learning. The scalability and complexity of these novel techniques has afforded new ways for enacting education research and has helped scholars gain new insights into human cognition and learning. Nonetheless, there remain some domains for which pure computational analysis is currently infeasible. One such area, which is particularly important today, is open-ended, hands-on, engineering design tasks. These open-ended tasks are becoming increasingly prevalent in both K–12 and post-secondary learning institutions, as educators are adopting this approach in order to teach students real-world science and engineering skills (e.g., the "Maker Movement"). This paper highlights findings from a combined human–computer analysis of students as they complete a short engineering design task. The study uncovers novel insights and serves to advance the field's understanding of engineering design patterns. More specifically, this paper uses machine learning on hand-coded video data to identify general patterns in engineering design and develop a fine-grained representation of how experience relates to engineering practices. Finally, the paper concludes with ideas on how the specific findings from this study can be used to improve engineering education and the nascent field of "making" and digital fabrication in education. We also discuss how human–computer collaborative analyses can grow the learning analytics community and make learning analytics more central to education research.

Keywords: Engineering design, design thinking, machine learning analytics, expertise

1 INTRODUCTION

Over the past three decades, technology has had a significant impact on education (see Koedinger & Corbett, 2006; Lawler & Yazdani, 1987; Papert, 1980; Resnick, 2002; U.S. Department of Education, 2010; Wilensky & Riesman, 2006 for examples). From the observed transition from chalk and blackboard to whiteboards to overhead projectors to PowerPoint presentations to online videos to cognitive tutors to virtual learning communities. Through these developments, it is apparent that instructional approaches have gradually incorporated new technologies. But innovations were not only in the delivery of information: more recently, technology has clearly altered elements of teaching and learning. Technological innovations have also allowed us to capture and process much more extensive traces of how people learn in digitally monitored settings. Access to this expanse of data has been central to the development and proliferation of both the learning analytics and educational data mining communities (Baker & Yacef, 2009; Siemens & Baker, 2012; Bienkowski, Feng, & Means, 2012). Furthermore, the use of these technologies has enabled researchers to tackle and study educational challenges at scale and in novel ways. Despite all of the affordances, a number of challenges remain outside of the current capabilities of traditional learning analytics and educational data-mining approaches. As we consider learning analytics as a middle space, we would like to propose that computer-based analysis, by itself, is insufficient for answering many important research questions in education. Domains with a wide variety of possible solutions and learning pathways represent a challenge for purely automated analyses. For

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

example, fields where students are challenged to invent or create hardware or software solutions typically necessitate a level of human interpretation that can be difficult for a computer to infer. Similarly, situations where the design constraints may involve carefully weighing social and cultural concerns in conjunction with traditional engineering requirements may also require intensive and subtle human interpretation. While technological advances will undoubtedly expand the capabilities of pure computational analysis to a larger array of learning activities, we argue that we can address some of these challenges by combining analytics techniques with human coded data analyzed through qualitative approaches. This methodological intersection creates hybrid systems in which computer analysis is employed to study human labelled data. While we have been using these types of approaches in the nascent field of multimodal learning analytics (e.g. Blikstein, 2013; Worsley, 2012; Worsley & Blikstein, 2013), to date, we know of very few instances of Learning Analytics research that takes human-labelled data and exhibits how computational analysis can mirror, and extend, approaches and results achieved through traditional education research. However, a reality is that this type of qualitative research is what many education researchers are pursuing. By demonstrating the existence of robust computational methods that can be used to streamline traditional education research analyses, the field of Learning Analytics can more squarely enter the fold of the learning sciences. Such collaboration will serve to improve the quality and scalability of current education research, and increase the impact of Learning Analytics.

To help advance these goals and further the fields understanding of engineering design practices, we present two examples from an engineering task that demonstrate how combining elements of traditional qualitative analysis with machine learning can 1) help us identify patterns in engineering strategies and 2) allow us to garner a more fine-grained representation of how engineering practice varies by experience level.

2 LITERATURE REVIEW

This study is informed by prior research from engineering education and the learning sciences, and a largely distinct body of literature on artificial intelligence techniques that can potentially be used for studying open-ended learning environments. Our research bridges these communities by showing that each domain has a strong contribution to make in advancing the field's understanding of learning, especially in constructionist learning environments (Papert, 1980; Harel & Papert, 1991). In what follows, we highlight key studies from these paradigms and describe how their work informs the current study.

2.1 Engineering Education Research

The area of Engineering Education has received a great deal of attention recently. There have been various efforts to bring project-based learning to the forefront of engineering education and an equally strong call for curriculum to emphasize process instead of *product*. As an example of these changes, professors and researchers have been redesigning both first year and capstone design projects with the hope of helping students develop greater fluency with the tools and methods that they will need as practicing engineers confront. Traditionally, work in engineering education and project-based learning has involved developing new approaches for assessing learning and knowledge. Typically, studies from this body of research focus on qualitative analyses of student language (Atman & Bursic, 1998; Dym, 1999; Russ, Scherr, Hammer, & Mikeska, 2008), student artifacts (Dong & Agogino, 1997; Lau, Oehlberg, & Agogino, 2009), or the combination of language and artifacts (Atman & Bursic, 1998,; Worsley &

Blikstein, 2011) created in the process of designing, building and/or inventing. We look to contribute to the body of engineering education research by analyzing these practices at a very fine-grained scale.

2.2 Research on Expertise

Within the engineering education community, and beyond, considerable research has been undertaken in the study of expertise (for examples see Chi, Glaser, & Rees, 1981; Cross & Cross, 1998; Ericsson, Krampe, & Tesch-Römer, 1993). More specifically, a collection of researchers has investigated design patterns on engineering tasks through think-alouds (Atman & Bursic, 1998; Ericsson & Simon, 1980; Russ et al., 2008). When considering expertise in the engineering context, many of the constructs discussed have been cast under different names: computational thinking (Resnick et al., 1998; Wing, 2006; Guzdial, 2008), designing thinking (Dym, 1999; Dym, Agogino, Eris, Frey, & Leifer, 2005), and mechanistic reasoning (Russ et al., 2008). Because each of these constructs could easily be the subject of an entire review, we will only mention them in passing as to indicate that these ideas have contributed to the analyses in this paper we will focus on a single body of literature by Atman and her collaborators (Adams, Turns and Atman, 2003; Atman & Bursic, 1998; Atman, Chimka, Bursic, and Nachtmann, 1999; Atman et al., 2007; Atman, Kilgore, & McKenna, 2008) and which is representative of the state of the field, and is directly related to our analyses. Atman and Bursic (1998), Atman et al. (1999), and Adams, Turns, and Atman (2003) investigate engineering design language and practices by comparing engineering practices between freshmen and senior engineering students. In Atman et al. (2007), they compare expert engineers to college engineering students. The comparisons examined how the respective groups were rated in terms of time spent along several dimensions: Problem Scoping, Information Gathering, Project Realization, Total Design Time, and Considering Alternatives in the following activities: problem scoping, information gathering, project realization, total design time, considering alternatives, and solution quality. In conducting this comparison, the authors expected to find that experts (a) do a better job at gathering information (b), spend more time in the decision making process (c), spend more time in the project realization process (d), consider fewer design alternatives, and (e) spend more time transitioning between the different types of design activities. They employed basic quantitative measures to keep track of the number of times a given action was taken and the amount of time devoted to each action, and found that while a handful of their hypotheses were correct, the most insightful finding had little to do with the quantitative differences between the groups. Instead, the true findings had more to do with the overall pattern that different experts followed. While the group had previously identified iteration as an important component to engineering design, Atman et al. (2007) describe the expert design process as being like a cascade. These cascades were seldom present among novices. To identify cascades, Atman, Diebel, and Borgford-Parnell (2009) focused on three different representations of the students' design activities and stages. These representations include a timeline plot, which shows the presence or absence of a given action at each increment in time; a cumulative time plot, which captures the amount of time spent in each activity (y-axis) relative to the total amount of time (x-axis); and progress time plots, which is the same as the cumulative time plot, except that the x-axis is the percentage with respect to each individual action, as opposed to the overall time for all activities. Using the progress time plots, Atman, Diebel, and Borgford-Parnell (2009) define a cascade as being a design process typified by considerable time doing project scoping at the onset, and project realization at the end. Embedded within the ways that Atman et al. identified iterations and cascades is the importance of temporality. Simply looking at the number of times, or amount of time individuals spent in a given action was not predictive. Instead, the authors needed to look at the entire sequence of actions taken and the context in which each action appeared.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

In the same way, we are interested in studying the overall patterns of the engineering design process and doing so in a relatively automated fashion. However, traditional approaches for conducting this type of research are limited to human-based video analysis, which can be quite laborious and time-consuming. Other strategies, such as sequence mining techniques, tend to remove the individual or groups of segments from the context in which they appear. Nonetheless, we have identified a set of approaches from machine learning that inform the computational aspects of this study.

2.3 Machine Learning Analysis of Computer Programming Behaviour

Central to this paper is a desire to study human actions on relatively open-ended tasks. When considering automated analysis of open-ended tasks, much of the previous work relates to studying computer-programming behaviour. Blikstein (2011), Piech, Sahami, Koller, Cooper, and Blikstein (2012) Blikstein, Worsley, Piech, Sahami, Cooper, and Koller (in press) are examples of this work. The three papers describe a similar strategy of gathering snapshots of students' computer programs. Blikstein used the snapshots to examine the differences between expert and novice programmers. There he identified prototypical styles and patterns that students used over the course of a multi-week assignment. Piech (2012) and Blikstein et al. (in press) used the snapshots as the basis for identifying a set of generalizable states that students enter while completing a given assignment, or set of assignments. These states were determined through clustering and used to construct models of student learning pathways. In Piech et al., the authors build a Hidden Markov Model (HMM) of the different student paths. The transition probabilities from the HMM were used to compare individual students and ultimately cluster them into three groups. The clusters identified in their study aligned with final examination performance at a higher level of accuracy than could be achieved by using the midterm examination grade as a predictor. In Blikstein et al. (in press), the authors examine learning pathways across an entire computer science course, and show how progressions in students' tinkering and planning behaviour correlates with student grades.

From these three studies, it becomes apparent that the tools of computational analysis hold significant promise, especially when faced with large datasets. In the case of Piech et al. specifically, we find that using machine learning and probabilistic graphical models can be invaluable in developing representations of the students' data from which we can learn. In our analysis, we follow a similar approach but in the domain of hands-on engineering design tasks. Additionally, where Piech et al. computes student similarity based on the HMM transition probabilities, we chose not to make the Markov assumption which is only concerned with the immediately preceding state. This allows us to maintain the context for each student's action.

Other work by Berland et al, Martin, Benton, Ko, and Petrick-Smith (2013) uses clustering to study prototypical program states among novice computer programming students. They used these prototypical programming actions as the basis for studying how students transition between different actions. In so doing, they found that the data could be used to identify three general patterns: tinkering, exploring, and refining. These categories extend previous work on tinkering and planning behaviours in computer programming with a more complex representation of student programming practices (Turkle & Papert, 1991). Our Analysis 1 from this paper follows a similar paradigm by identifying general building patterns among our population of participants.

Taken together, a primary affordance of computational analysis is the high level of resolution one can achieve. While our analysis does not employ a big data science in the traditional sense of having thousands many participants, we do look at participant actions at a level of granularity that would be

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

hard to replicate by purely human analysis. Because of this, we are able to identify otherwise undetectable patterns in behaviour.

The remainder of this paper is divided into five sections. In the next section, we describe the dataset and the coding scheme. This is followed by an introduction to the basic machine learning techniques used in the two analyses that we present. We then transition into studying the first set of questions: 1) What are the prototypical building strategies used among engineers of varying levels of experience? 2) In what ways do these building practices relate to prior literature? 3) What new insights can we garner about engineering through these prototypical building practices? After addressing these questions, we move to the second set of questions that are specifically related to correlations between student actions and prior experience: 1) What building actions distinguish individuals of different levels of experience? 2) How do these building actions align with, contribute to, or differ from prior research in this field?

3 METHODOLOGY

3.1 Data

Data is drawn from thirteen participants. Each participant was given everyday materials, and asked to build a tower that could hold a small mass (< 1 kg). Participants were also challenged to make the mass sit as high off the ground as possible. The task was designed to examine how successfully students are able to take their intuitions about mechanics and physics and translate them into a stable, well-engineered structure. We expected students to use knowledge about forces, symmetry, and the affordances of different geometric shapes, to enable them to complete the task. The additional challenge of making the structure as tall as possible was introduced to push all students to the limits of their ability, regardless of prior experience.

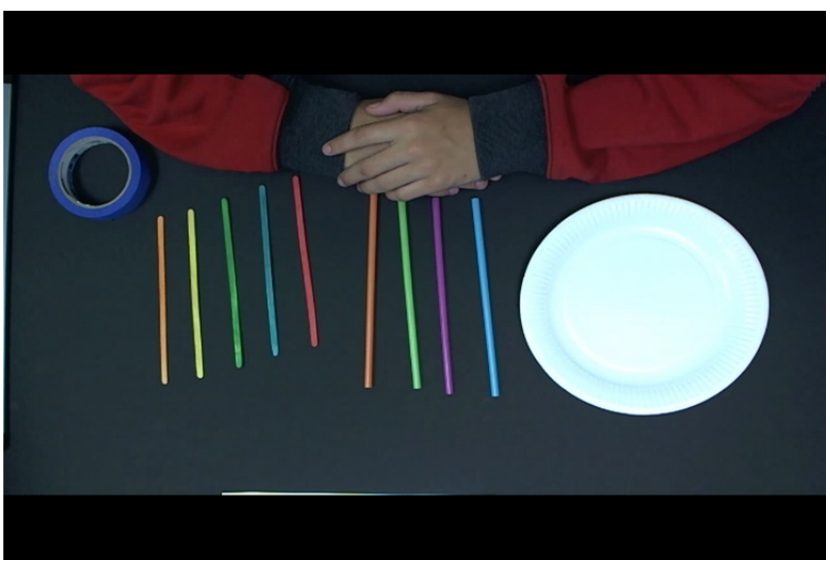

Figure 1. Initial Materials

Students were given four drinking straws, five wooden Popsicle sticks, a roll of tape and a paper plate (Figure 1) and were told that they would receive ten minutes to complete the activity. In actuality, they were permitted to work for as long as they wanted. Average participation time was approximately 25

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

minutes (SD=13minutes.) Three sample structures are depicted in Figures 2, 3, and 4 to give the reader a better idea of the task.

Figure 2. Sample Structure 1

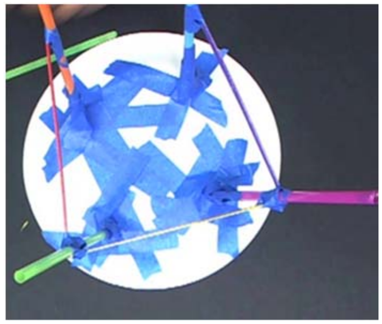

Figure 3. Sample Structure 2

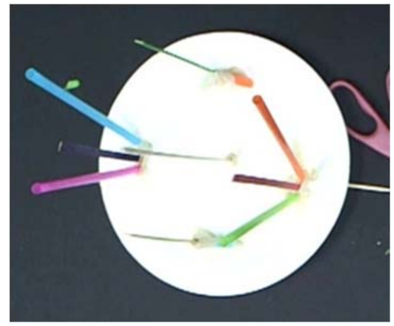

Figure 4. Sample Structure 3

Audio was used to capture meaningful utterances made by the participants, though students were not required to engage in think-alouds. Audio was also captured of each student's metacognitive analysis of his or her building approach. A video camera place above the students, pointing vertically down to the work area, captured the movement of objects as students progressed through the task (Figure 1). Gesture data, which consisted of twelve upper-body parts from a Kinect sensor, recorded the students' physical actions. While we only focus on the video data for this paper, Worsley and Blikstein (2013) contains a preliminary analysis of how the gesture data may provide an automatic channel for predicting expertise based on the frequency of two-handed actions.

3.2 Defining Experience

Prior to the study, students were classified based on their level of experience in the domain of engineering design based on two dimensions. The first dimension pertains to the amount of formal instruction students had received in engineering. Individuals who had completed undergraduate or graduate degrees in engineering were labelled as relative experts. Individuals who had not completed degree programs in engineering answered interview questions about their prior experience. These interviews, in conjunction with teacher-based ratings, were used to label the relative level of experience of each participant. To provide some additional context, the teachers worked with the students for more than two-hundred hours in an engineering and digital fabrication class, over four weeks. Student experience labels were assigned only when all researchers agreed. This labelling process resulted in a population of three experts, two high experience students, five medium experience students, and three low experience students.

3.3 Coding

In order to establish a basis for comparing students, we developed a set of action codes (Table 1). The process we followed in developing these codes mirrors that commonly undertaken in grounded theory-based research. An initial set of codes was identified through open coding of a sample of the videos. After individually developing a set of codes, the research team came together to discuss those codes and agree upon which ones to include in the final codebook. Once those codes had been defined and agreed upon, a graduate research assistant coded each video.

Table 1. Fine-Grain Object Manipulation Codes

Code	Description
BUILDING	Joining objects by tape or other relatively permanent means.
PROTOTYPING MECHANISM	Seeing if putting two (or more) objects together will work. This may include acting out a mechanism with the materials.
TESTING MECHANISM	Testing a subsection of the overall system.
UNDOING	Taking the structure apart to make a change to a previous build.
SINGLE OBJECT EXAMINATION	Pressing on or bending an object to explore its properties.
THINKING WITHOUT AN OBJECT IN HAND	Surveying the pieces but not touching anything or actively doing anything.
THINKING WITH AN OBJECT IN HAND	Holding one or more objects but not manipulating them.
SYSTEM TESTING	Putting force on a collection of relatively permanently affixed pieces to see if they will hold the mass.
ORGANIZING	Repositioning the raw materials but not actually building, examining, or prototyping.
BREAKING	Breaking apart sticks, bending straws, or ripping the plate.
ADJUSTING	Repositioning an object slightly, or applying more tape to reinforce or correct portion of the structure.

Similar to Atman's "Design Stages," we developed a scheme of higher-level object manipulation classes. These include *Realization, Planning, Evaluation, Modification*, and *Reverting.* The mapping between Object Manipulation Classes and Object Manipulation Codes can be found in Table 2. For the analyses presented in this paper, we will focus on examining patterns at the Object Manipulation Class level.

Table 2. General Object Manipulation Action Classes

Class	Codes
REALIZE	• Building and Breaking
PLAN	• Prototyping mechanism • Thinking with or without an object • Single object examination • Organizing and Selecting materials
EVALUATE	• Testing a mechanism • System testing
MODIFY	• Adjusting
REVERT	• Undoing

3.4 General Algorithm

3.4.1 Sequence Segmentation

The analytic technique begins by segmenting the sequence of action codes every time an EVALUATE action occurs. Our assumption is that we need to have a logical way for grouping sequences of user actions and each time a user completes an EVALUATE action, they are signalling that they expect their

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

previous set of actions to produce important, actionable information and feedback, which may be in the form of their current structure succeeding or failing.

3.4.2 Segment Characterization

Each segment is recorded based on the proportion of the five object manipulation classes (REALIZE, PLAN, EVALUATE, MODIFY, REVERT) that took place during that segment. Put differently, we now have a five dimensional feature vector for each segment, where each dimension corresponds to one of the object manipulation action classes. As an example, consider the following set of codes:

PLAN, REALIZE, EVALUATE, MODIFY, REVERT, REALIZE, EVALUATE

This sequence of eight codes would be partitioned into two segments. The first segment would be PLAN, PLAN, REALIZE, EVALUATE; the second would be MODIFY, REVERT, REALIZE, EVALUATE. These two segments would then be used to construct two feature vectors based on the proportion of each of the action classes. In the case of the first segment of the example sequence — PLAN, PLAN, REALIZE, EVALUATE — we see that there are two PLANs, one REALIZE, one EVALUATE, zero MODIFY, and zero REVERT. Thus, the proportion of the segment occupied by PLAN is one-half, or 0.50. The proportion of the segment occupied by REALIZE is one-fourth, or 0.25 and the proportion of the segment occupied by EVALUATE is also one-fourth. Following this same procedure for both of the segments yields the results in Table 3.

Table 3. Sample Segmented Feature Set

Segment	MODIFY	REALIZE	PLAN	EVALUATE	REVERT
1	0.00	0.33	0.33	0.33	0.00
2	0.25	0.00	0.00	0.25	0.25

3.4.3 Segment Standardization

After constructing all segments for all participants, each column (MODIFY, REALIZE, PLAN, EVALUATE, REVERT) of the feature set is standardized to have unit variance and zero mean. This step is taken in order to ensure no biases when we perform clustering in the next step.

Table 4. Sample Segmented Feature Set after Standardization

Segment	MODIFY	REALIZE	PLAN	EVALUATE	REVERT
1	-1	1	1	1	-1
2	1	-1	-1	-1	1

3.4.4 Segment Clustering

Following standardization, the segments are clustered into four or ten clusters using the k-means algorithm (the selection of four and ten as the number of clusters will be discussed in more detail later.) The clustering process uses all of the students' action segments in order to develop a set of generalizable action segments.

Each of the resultant clusters contains several of the segments, and can be characterized by the cluster centroid. This cluster centroid represents the cluster's average value along the five dimensions. As an example, if segment 1 and segment 2 defined a cluster, their cluster centroid would be zero along all dimensions (Table 5).

Table 5. Hypothetical Cluster Centroid from Sample Feature Set

Segment	MODIFY	REALIZE	PLAN	EVALUATE	REVERT
1	-1	1	1	1	-1
2	1	-1	-1	-1	1
Centroid	0	0	0	0	0

3.4.5 Segment Re-labelling

Each segment for each student is replaced with the generalizable action segment that it is most similar to (recall that an action segment is characterized by a five-dimensional vector that reports the proportion of that segment spent in each of the five Object Manipulation Classes: REALIZE, PLAN, EVALUATE, MODIFY, REVERT). Following from the example, above, the two segments would be re-labelled using the cluster centroid values and label (Table 6).

Table 6. Hypothetical Segment Re-labelling

Segment	Cluster
1	0
2	0

3.4.6 Dynamic Time Warping

Once each student's sequence has been re-labelled using the cluster centroids from above, "dynamic time warping" is used to compute the minimum distance between each pair of participants. Dynamic time warping minimum distance can be seen as a constrained form of the minimum edit distance or Levenshtein distance. It differs from edit distance in that when computing the distance between two sequences, a given sequence can only undergo item insertion. The inserted item must either be a repetition of the preceding item or the subsequent item. Item insertion is only used if it will reduce the overall distance between two students' sequences; otherwise, a simple numerical value is computed based on the Euclidean distance between the two vectors. As a very simple example, if we were computing the distance between two sequences: A) 1, 2, 0 and B) 1, 2, 2, 2, 1; we would extend sequence A to be 1,2, 2, 2, 0, such that the second value is repeated in order to produce the maximum alignment between sequences A and B. The reason for using dynamic time warping is that we are interested in looking at the overall design patterns that participants are using and are less interested in the amount of time spent in the respective stages. Dynamic time stretches the different students' vectors based on minimizing the differences between them and in no way alters the order in which actions appear. This computation yields an n-by-n matrix of minimum distances.

3.4.7 Participant Clustering

Finally, the n-by-n matrix from the dynamic time-warping calculation is standardized along each column, before being used to construct the final clustering, again with the k-means algorithm.

3.4.8 Algorithm Summary

In summary, this algorithm takes the full sequence of actions from each student and splits them in smaller segments every time a student explicitly evaluates, or elicits feedback from, his or her structure. The proportions of actions in the different segments are used to find representative clusters, which are subsequently used to re-label each user's sequence of segments. Finally, we compare sequences across participants and perform clustering on the pair-wise distances in order to find natural groupings of the participants. Figure 5 provides a visual representation of the overall process. In the following sections,

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

we show how this general algorithm can be used to 1) study prototypical building patterns as well as 2) identify those building patterns that differentiate individuals of differing levels of experience.

Figure 5. Summary of General Algorithm

4 ANALYSIS 1: PROTOTYPICAL BUILDING STRATEGIES

The goal of Analysis 1 is to identify prototypical building strategies among the research participants. More specifically, we answer the question of how building patterns can be used to understand engineering practices better. To address our question, we proceed by first discussing the types of patterns that we expected to see among our participants. We also describe the specific instantiation of the general algorithm that we applied for this portion of the analysis and why this approach is feasible. We then present three different representations of our findings: one quantitative, one based on video analysis, and one qualitative. Finally, we conclude the analysis with a discussion of how these findings can be used for studying the learning of basic engineering skills and used more broadly in the learning analytics community.

4.1 Hypotheses

Based on prior literature, one hypothesis is that students will use cascades (Atman et al., 2007; Atman Deibel, & Borgford-Parnell, 2009). During such cascades, students pay particular attention to PLAN at the beginning of the task and gradually decrease the proportion of time spent in PLAN as a greater proportion of time is spent in REALIZE. They are also constantly in the process of considering alternative designs. Another design process pattern to look for is iterative design. Atman and Bursic (1998) found that iterative design was important for creating effective solutions. As the individual begins to engage in the realization process, he or she is constantly updating the design, and perhaps even returning to PLAN actions in order to refine the product iteratively. In our building context, we expect to see cascades manifested as different amounts of iterative design. While Atman et al. typically attribute this to be being an expert-like behaviour, they do indicate that it is not limited to experts. Instead, they found that it merely occurred more frequently among their more experienced research participants.

Connected with the above hypothesis about design process is one of quality. Prior research found that as the amount of iterative design or cascading increased, so did the quality of the artifacts produced (Atman et al., 2007). Accordingly, an additional hypothesis is that the prototypical building strategies will have some correlation with the quality of the products.

4.2 Algorithm Implementation Specifics

We use the methodology described in the General Algorithm section and cluster the data into four clusters during the Segment Clustering step, as well as during the Participant Clustering step. The number of clusters was set to four during segment clustering based on the silhouette score (Rousseeuw, 1987). In the case of participant clustering, four clusters were used in order to ensure some variation between clusters, while also avoiding clusters with only one participant.

4.3 Object Manipulation Generalizable Segments

Recall that the approach we use involves k-means clustering at two different times. The first instantiation of clustering is intended to identify a set of generalizable action clusters. Each of these action clusters is defined by the percentage of time spent in Planning, Realizing, Modifying, Evaluating, and Reverting.

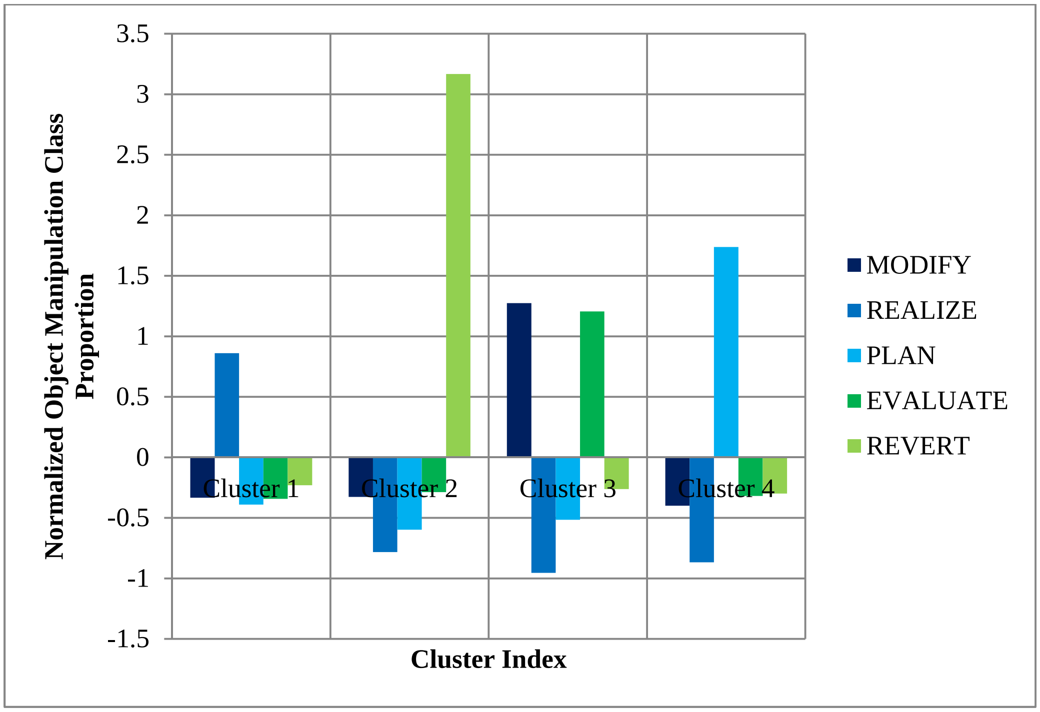

Figure 6. Object Manipulation Class Proportions by Cluster

In Figure 6, we report the Object Manipulation Class proportions for the different cluster centroids. From this, it is clear that Cluster 1 primarily aligns with REALIZE, Cluster 2 with REVERT, Cluster 3 with MODIFY and EVALUATE, and Cluster 4 with PLAN. In order to simplify discussion of these in the following sections, we will refer to the generalizable clusters as G-REALIZE, G-REVERT, G-MODIFY-EVALUATE, and G-PLAN. In this way, the user will not be confused between our discussion of the Object Manipulation Classes and the Generalizable Segment Labels.

4.4 Participant Cluster Centroids

The second stage of clustering occurs among the participants and is based on the similarity of their dynamically time warped Object Manipulation Sequences. From that clustering, four participants were assigned to Cluster A, three to Cluster B, two to Cluster C, and four to Cluster D. To simplify the naming, we will always refer to clusters of object codes (from Segment Clustering) using numbers, and clusters of participants (from Participant Clustering) using letters.

In order to better understand the nature of these different clusters and explore how their characteristics relate to prior research, we present three representations of the clusters. However, as a first indication that the clusters are differently, we treat each participant action segment as independent of all others. This obviously is not true, but provides a means for a quick comparison via Chi-Squared analysis. The

Chi-Squared analyses suggest that each cluster used the Generalizable Action Segments with markedly different frequencies (Table 7).

Table 7. Pair-wise Chi-Square Analysis of Generalizable Action Segment Usage

Group 1	Group 2	Chi-Square Statistic	Probability
A	B	57.6	0
A	C	42.48	0
A	D	33.35	0
B	C	69.03	0
B	D	44.7	0
C	D	64.78	0

4.5 Time-Based Graphical Representation of Participant Clusters

Having established that the four clusters are different, we now examine the nature of those differences. The first representation that we employ is a comparison of the time spent in the different Object Manipulation Classes.

Figure 7. Relative Time Spent in Each Object Manipulation Class by Cluster

As we compare the clusters in Figure 7, we clearly observe that Clusters A is characterized by having the most PLAN, and the least amount of EVALUATE. Cluster B has the least amount of REALIZE, and the most amount of REVERT. Cluster C (green) stands out for having the most time planning. On the other extreme is cluster A, which also spent considerable time in REALIZE and the most EVALUATE. It also has the least amount of PLAN, REVERT, and MODIFY as compared to its peers. Finally, Cluster D falls in the middle along all five of the dimensions.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

4.6 Proportion-Based Graphical Representation of Participant Clusters

One drawback of the normalized time plot (Figure 7) is that it does not take into account the total amount of time participants took on the task. Accordingly, we present a graph of the proportion of time spent in each of the Generalizable Action Segments by cluster (Figure 8).

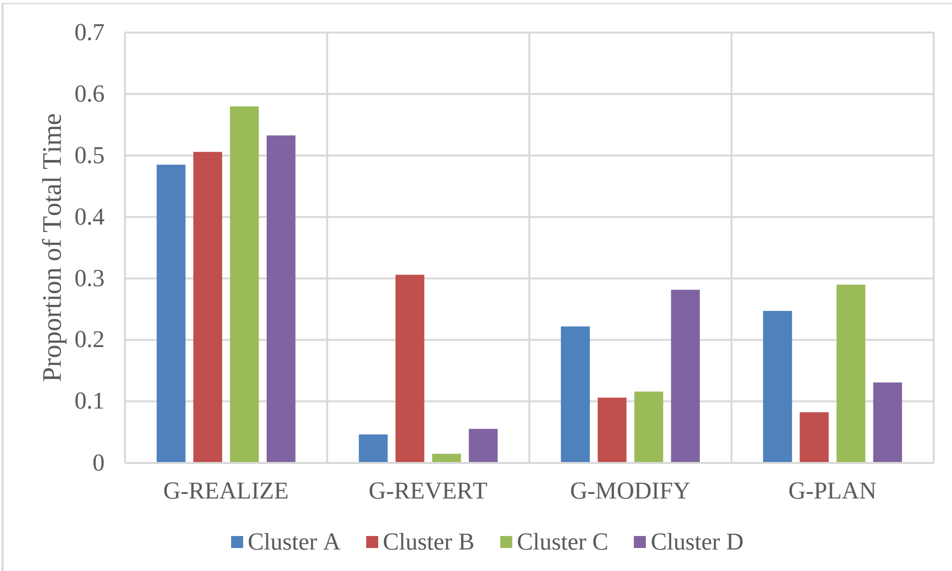

Figure 8. Proportion of Time Spent in Each of the Generalizable Action Segments by Cluster

When we move to this representation, we find that Cluster C, who spent the least amount of time planning in the previous graph, spends the largest proportion of time in G-PLAN. Hence, it is not that Cluster C participants did not plan; it is simply that their total time planning was less than their peers. Proportions for G-REVERT and G-MODIFY are at the lower end of the spectrum for the different clusters. Looking at the proportions also informs our understanding of Cluster A, which spends a large proportion of time in G-MODIFY despite spending little (absolute) time in MODIFY and EVALUATE (Figure 6). Apart from these, this representation appears to be analogous to what we observed in Figure 7. Thus using the proportion of time helps to better describe some of the nuances of each group's behaviour, while confirming many of the observations from the absolute time spent doing each object manipulation type. Furthermore, these two representations describe the noted differences within the Chi-Squared analyses.

4.7 State–Transition Representation of Participant Clusters

The first representation focuses on aggregate time spent in the different Object Manipulation Classes and Generalizable Action Segment types. These have been used within the literature as ways of studying engineering design patterns (e.g. Adams, Turns & Atman 2003; Atman et al., 1999, 2007, 2008). However, one goal of this paper is to go beyond this and look more closely at the patterns of actions that students take. The literature has suggested that examining the rate of transition among different actions can be informative for studying design patterns (Atman et al., 1999, 2007). To consider these, we construct a state machine of student actions within each cluster. Moreover, we can construct a transition probability table that examines the frequency with which individuals of a given cluster transitioned between different Generalizable Action Segments. Putting the data in this form deviates

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

from the dynamic time-warp analysis that we completed on the entire sequence of user actions, but still offers some insights into what characterizes each of the clusters.

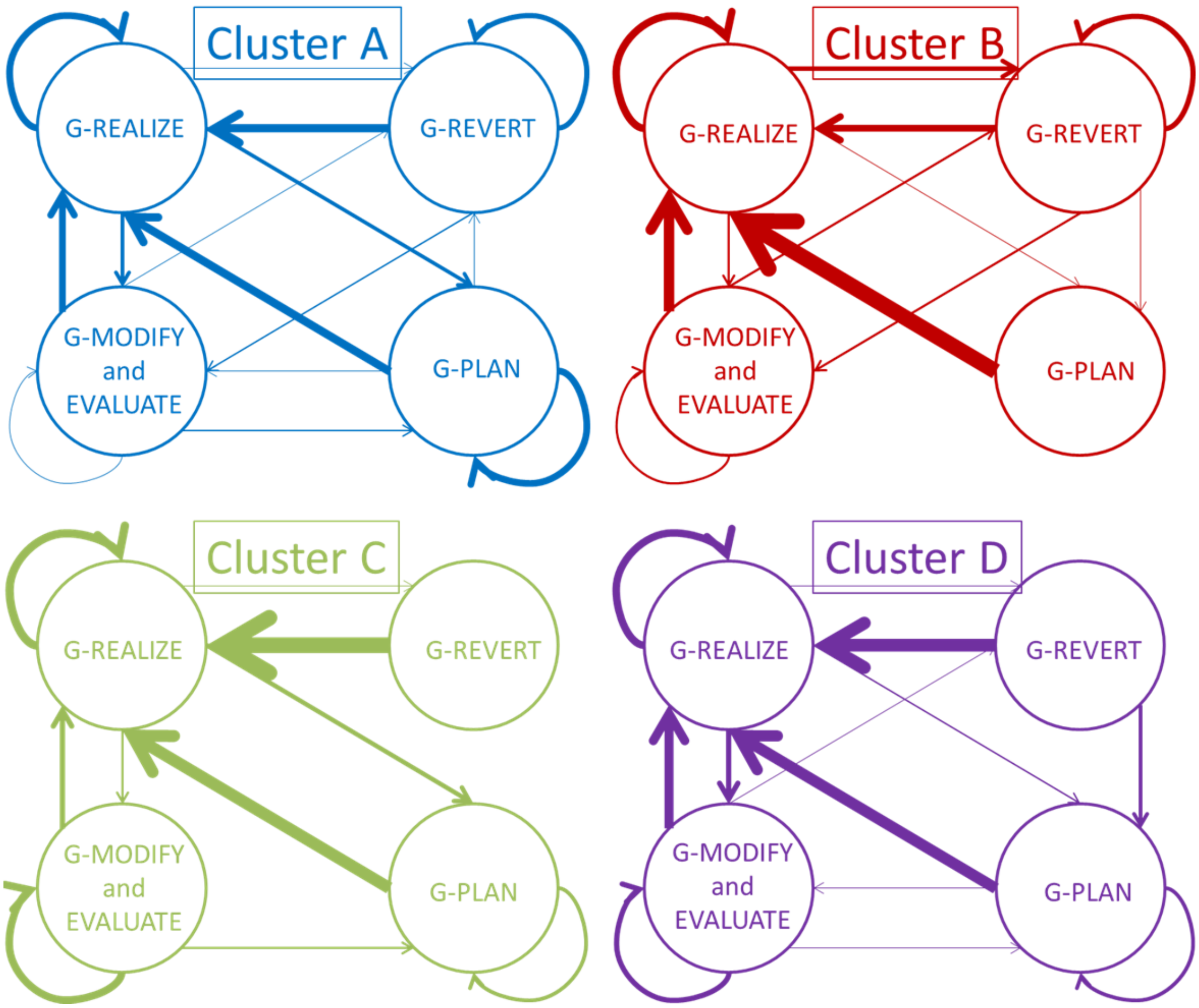

Figure 9. Transition Diagrams by Different Clusters. The size of each line corresponds to the probability of that transition with thicker lines indicating higher probability than thinner lines

Figure 9 shows the state machine diagram for all four clusters. Before diving into the specifics of each group's transition patterns, we present pair-wise Chi-Square analyses of the transition probabilities across all pairs of states. From

Table 8 we again see that all of the groups significantly differ from one another in their transition behaviour.

Table 8. Pair-wise Chi-Square Analysis of Transition Probabilities

Group 1	Group 2	Chi-Square Statistic	Probability
A	B	100.08	0
A	C	127.66	0
A	D	89.12	0
B	C	53.60	0
B	D	68.76	0
D	C	72.20	0

4.7.1 Cluster A

Cluster A is typified by planning behaviour, which appears to be sustained and frequent. Cluster A also records a relatively reduced transition probability for building. As a point of comparison Cluster A participants spend relatively less time transitioning to G-REALIZE than any other cluster. Approximately 50% of Cluster A's actions consist of transitions to G-REALIZE, whereas the value is roughly 65% for the other clusters. Instead of transitioning into G-REALIZE, Cluster A is frequently transitioning in and out of G-PLAN. Moreover, unlike many of the other clusters, Cluster A is more likely to engage in sustained planning, meaning that they will return to G-PLAN immediately after completing a G-PLAN segment.

4.7.2 Cluster B

Cluster B is typified by a lack of planning, and a prevalence of reverting. As evidence for this categorization, Cluster B seldom transition into G-PLAN. Furthermore, after completing a G-PLAN segment, the group always transition into G-REALIZE. Hence there are no instanced of sustained planning, as was the case for Cluster A. Apart from frequently transitioning to G-REALIZE and seldom transitioning to G-PLAN, Cluster B, differs from Clusters C and D in how they transition into G-REVERT. Namely, Cluster B is more likely to enter into a G-REVERT state than Clusters C and D.

4.7.3 Cluster C

From the transition probabilities, Cluster C appears to be largely focused on building. Of all of the clusters, Cluster C engaged in the most sustained G-REALIZE activity. The probability of staying in G-REALIZE was 0.55, whereas for the other clusters this valued ranged from 0.46 and 0.49. Additionally, Cluster C seldom transitioned into G-REVERT, and would always follow a G-REVERT segment with a G-REALIZE segment.

4.7.4 Cluster D

Cluster D is typified by being at the middle of the pack along all four measures. Cluster D is very focused on building, but also makes frequent use of G-REVERT.

4.8 Qualitative Representation and Discussion

Thus far, we have focused on using quantitative data to study each cluster's characteristics. In what follows, we synthesize data from the two previous representations, and combine it with some qualitative analysis in order to solidify and summarize the four prototypical groups that we identified. During this section, we use progress time plots. For all of these plots, purple corresponds to G-PLAN, blue corresponds to G-REALIZE, red corresponds to G-REVERT, and green corresponds to G-MODIFY.

4.8.1 Cluster A — PLAN, REALIZE and MODIFY

In our estimation, Cluster A represents a group of students that is exhibiting a robust design process and high quality ideas. Through the two graphical representations of time spent in each Object Manipulation Class and Generalizable Action Segment, we saw that this group exhibited a large amount of planning behaviour. Furthermore, as we turned to the state machine representation we observed the sustained planning behaviour that this group followed, in which they would repeatedly undertake G-PLAN actions. This is corroborated by qualitative observations made from the dataset. All of the individuals in this cluster built in a modular and iterative fashion. They started by planning, and then got a portion of their structure to achieve stability.

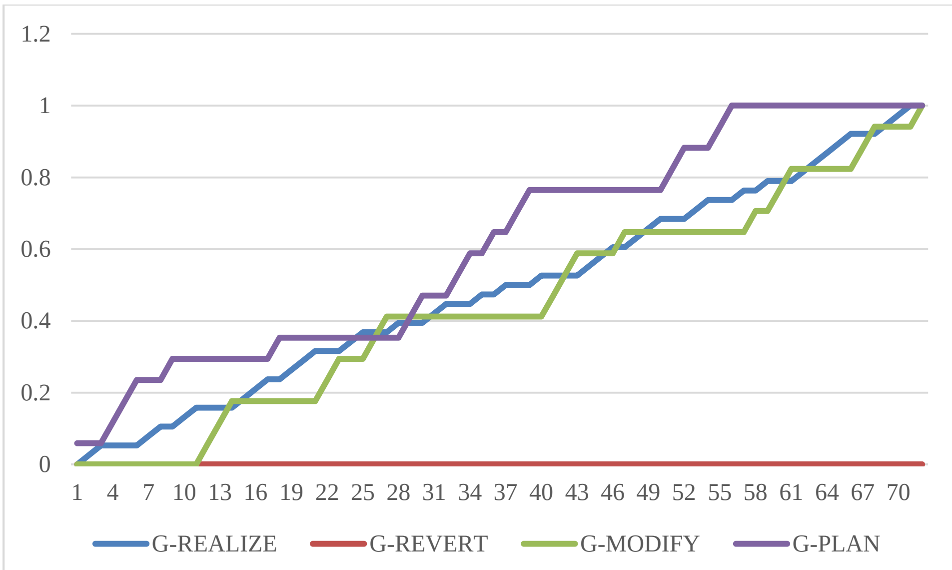

Figure 10. Cluster A Sample Progress Time Plot

After getting one part stable, they would return to planning and make another addition to their structure. This process would repeat itself until the participants were satisfied with their design or until all materials were used. Additionally, in their post-task metacognitive analysis, these students described their process as being iterative in nature, as well as involving some unexpected modifications. An example of this iterative approach can be seen in Figure 10, which depicts a progress time plot for one of the members of Cluster A. One can see from the plot that the blue and the purple lines are extensively intertwined. This is because the student alternated in using G-PLAN and G-REALIZE at different portions of the task. Finally, knowledge about engineering structures was evidenced in how the student talked about using triangles to reinforce the various supports in the structure.

4.8.2 Cluster B — REALIZE and REVERT

At the other end of the spectrum from Cluster A is Cluster B. From the aggregate time and state machine representations, we saw that Cluster B was characterized by G-REVERT actions and a lack of planning. In Figure 11, we see this represented in the purple and red lines with the purple line depicting G-PLAN. In this case, the line is flat, meaning that the student did all the planning at the beginning. The red line, indicating G-REVERT actions, steadily climbs throughout the process of the task and largely dominates all other activities. This was a common practice for this group. All of the individuals in this group had to undo their structures at one or more points during the task. Another key point of distinction that we observed qualitatively was that this cluster tended to use excessive amounts of tape in order to

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

reinforce connections or as the actual support mechanisms in their structure, which means that they were less likely to use a variety of engineering strategies.

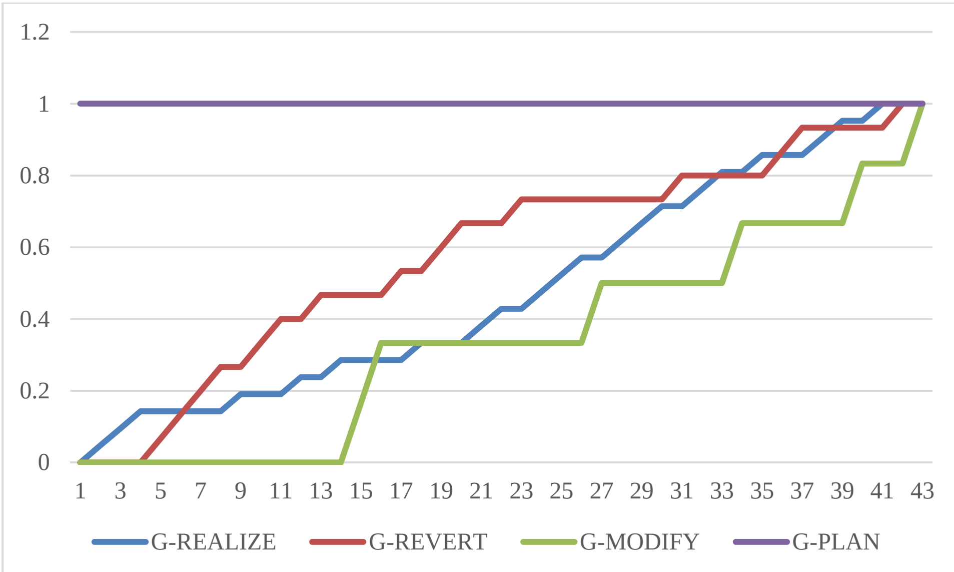

Figure 11. Cluster B Example Progress Time Plot.

4.8.3 Cluster C — REALIZE

In contrast to Cluster B, Cluster C consists of students who spent very little total time planning, relative to their peers, though they did spend a considerable proportion of their time planning. Interestingly, however, is that whereas Cluster A engaged in G-PLAN throughout the process, for Cluster C, planning was concentrated in just some moments. Comparing the rate of increase for planning instances in Figure 10 and Figure 12Figure 12, we see that the proportion of planning increases in much larger chunks for Cluster C, than for Cluster A.

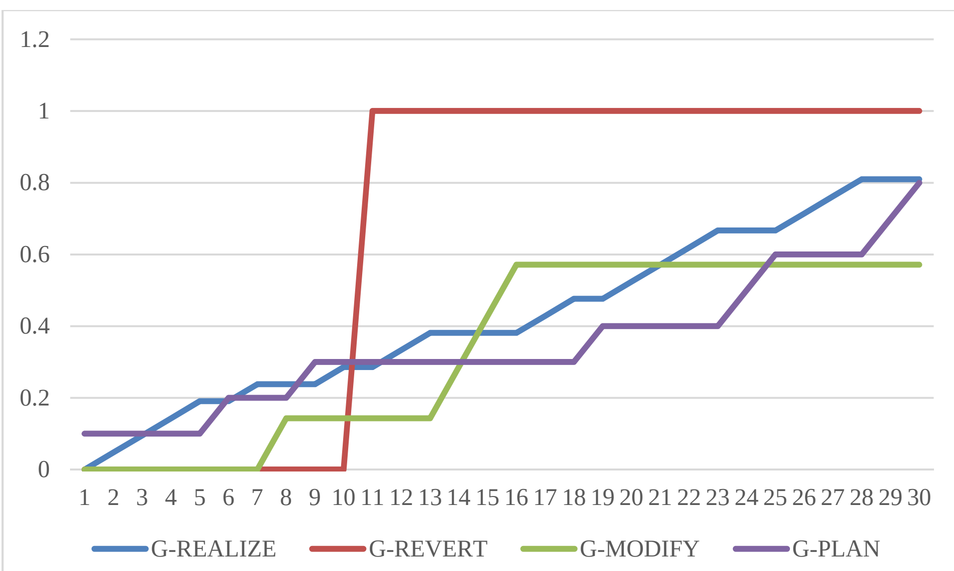

Figure 12. Cluster C Example Progress Time Plot. Purple is G-PLAN and blue is G-REALIZE

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

From the aggregate time plots and state machine representations, Cluster C appears to be typified by building process and a lack of undoing. This suggests that they generated ideas during the initial stages of the task that were sufficient to support the mass. As expected, in the qualitative analysis of these students, we saw a very streamlined process. Students would prototype a mechanism to make legs for the structure, test that prototype, and then repeat the process in order to make enough legs for the entire structure. One member of this cluster also found a way to use the roll of tape in the physical structure constructed (Figure 13).

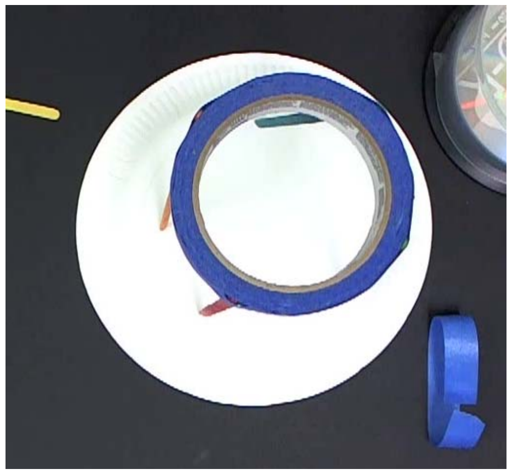

Figure 13. Clever use of roll of tape in design of structure

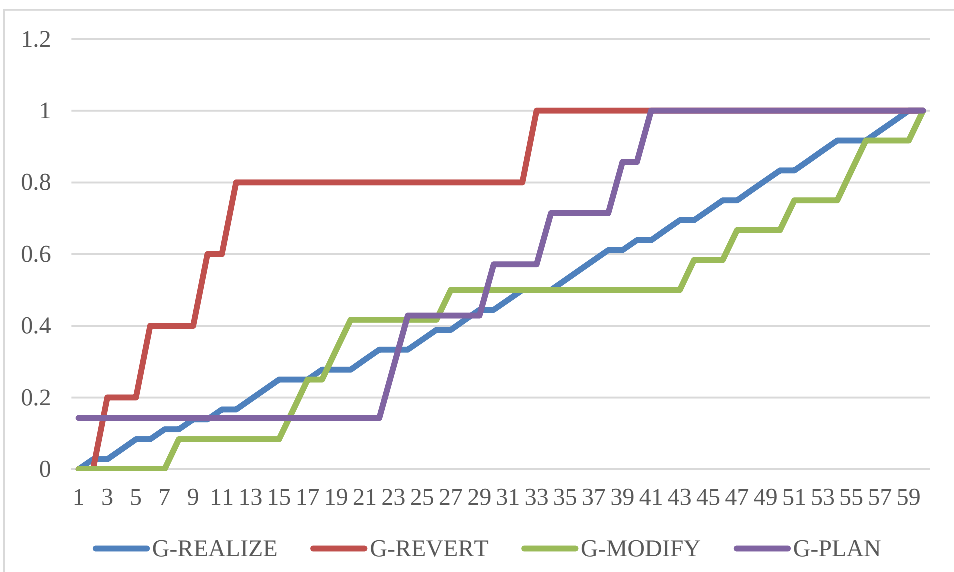

Figure 14. Cluster D Example Progress Time Plot. Red is G-REVERT, purple is G-PLAN and blue is G-REALIZE

4.8.4 Cluster D — PLAN, REALIZE, MODIFY and REVERT

Cluster D remained in the middle of the spectrum across all of the dimensions that we analyzed. Their distinctive characteristic is tied to starting in planning, and then subsequently iterating between G-REALIZE and G-PLAN, and later having that iterative process be disrupted by G-REVERT actions. For

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

example, they transition into G-REALIZE more frequently than Cluster A, but less so than the other clusters; and less frequently into G-REVERT than Cluster B, but more so than the other clusters. As can be seen in Figure 14 they begin by planning, and follow a process of iterating between G-REALIZE and G-PLAN. However, their pattern is also marked by frequent G-REVERT actions.

When we examine the behaviours of this cluster qualitatively, we confirm that it shares behavioural elements with each of the other clusters. For example, many of its members follow an iterative design process, in which they repeatedly prototype different aspects of their design and gradually test along the way. In this regard, they share the iterative building characteristic of Cluster A. However, they differ from Cluster A in the level of success of their ideas. Despite following a relatively sound design process, their structures lacked the appropriate engineering principles. For example, some of the students failed to reinforce the legs of their structures, causing them to fall over immediately.

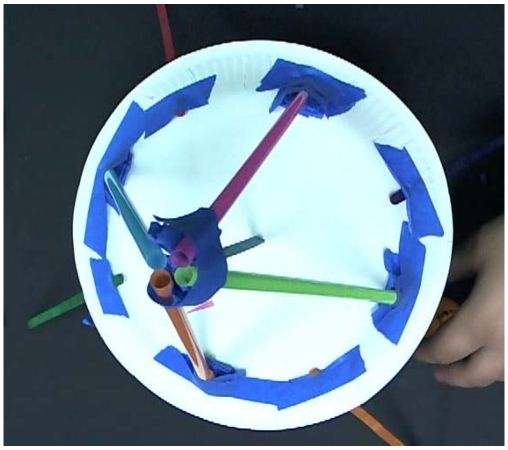

Figure 15. Cluster D Failed structure

Another student not only failed to reinforce the legs of the structure, but also assumed that the mass could balance on a circular surface without any reinforcement (Figure 15). Upon encountering such problems, students would often undo portions of the structure, not realizing the source of the structural failure. Thus, we would hypothesize that students who started with an iterative, systematic design had to resort to G-REVERT actions because of their lack of engineering knowledge. This difficulty in knowing how to debug their problems may have caused these students to share characteristics with the other clusters, despite being relatively systematic.

4.9 Dimensions of Analysis

As our initial hypothesis suggested, the approach that we used largely aligns with the quality of the design process and the quality of the engineering intuitions. Clusters A and D appear to be high on the quality of design process axis, as they follow an iterative design approach. Clusters B and C appear to be low on the scale of design process. Along the axis of engineering principles, clusters A and C appear to outpace clusters B and D. In this sense, the clustering has broken the participants into four quadrants of performance. Figure 16 shows the approximate placement of the clusters along these axes. One could posit that the clusters differ along a single dimension. However, from our qualitative analysis, quantitative analyses this does not seem to be the case. For example, based on the pair-wise Chi-Square analyses there is no way to reconcile the pair-wise Chi-Square statistics. However, the values can easily be reconciled when representing the clusters along two dimensions.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

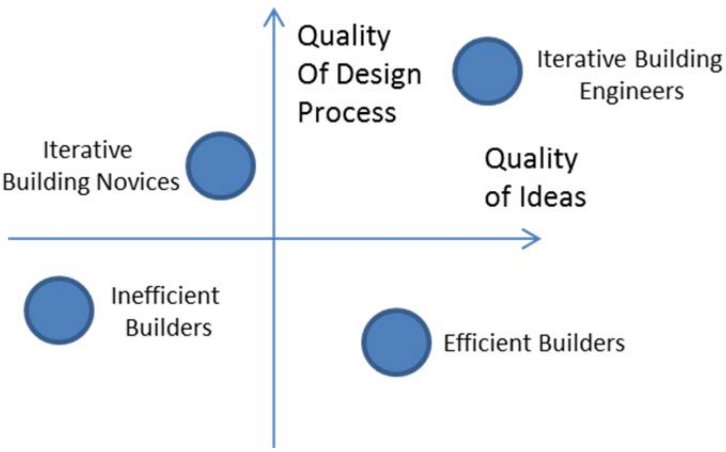

Figure 16. Quality of Design Process and Quality of Ideas Framework

4.10 Analysis Summary

From this analysis, we were able to generate prototypical building patterns that can be viewed as aligning to two dimensions. The two dimensions — design process and idea quality (Atman et al., 2007) — have been referred to in prior literature as being salient for engineering design tasks. However, unlike previous work, our analysis was completed using a computational analysis. Specifically, our result was achieved by using a dynamic time-warping algorithm, in conjunction with human coding and two iterations of clustering. The point of this analysis is not to suggest that these findings could not have been achieved through purely qualitative analysis, or that clustering will always produce these results. Qualitative analysis could have attained similar findings, but would have involved a much more labour-intensive effort. For example, it could have been quite challenging to 1) develop a systematic way for analyzing the data that would highlight these differences, 2) figure out an appropriate number of action segments to consider, and 3) determine a way to cluster that explicitly takes process into account (simply looking at the proportion of time each individual spent would overlook many of the nuances in the students' building patterns). These are all affordances that computational analysis can provide. At the same time, however, relying on purely computational approaches would not have been sufficient because of the challenges in coding the action segments. While our previous work has found some predictive power in using purely automated labelling of actions (Worsley & Blikstein 2013), here we are suggesting that combining qualitative analysis with data mining can enable education researchers to study complex learning environments more easily. Hence, as was the case with this study, they can provide novel frames for understanding the interaction between multiple dimensions.

In the second analysis, we move into answering more targeted questions about the nature of hands-on building practices. Instead of looking for an opportunity to explore the data and better understand the general patterns of behaviour, we enter with the goal of specifically identifying how to differentiate individuals of various levels of experience.

5 ANALYSIS 2: DISTINGUISHING BETWEEN DIFFERENT LEVELS OF EXPERIENCE

Under the previous analysis, we were primarily interested in identifying prototypical actions of students as they participated in an engineering design challenge. In this section, we are specifically interested in

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

what it means to have more experience. Moreover, since each student was classified based on his or her level of prior experience, we are interested in understanding how those differences in experience are manifested in the students' building practices. For this section, we again begin by considering what types of distinguishing practices we would expect to see among participants, and then discuss the specifics of the algorithm that we used. In discussing the algorithm, we also provide justification for using this approach over other alternatives. The discussion of the algorithm is followed by three representations of the findings. As before, we conclude with a discussion of how these findings relate to prior literature, and how this technique may be more widely applied.

5.1 Hypotheses

From prior literature, there are a number of hypotheses to consider about what will distinguish individuals with varying levels of experience. These hypotheses relate to the dimensions of planning, project realization, solution quality, and rate of transitioning. First, one would expect more experienced engineers to spend a greater proportion of time in project scoping or planning (Atman et al., 1999, 2007, 2009) and Adams et al. (2003). Furthermore, this additional planning behaviour should be evidenced both at the onset of the task, and throughout the activity, as the experts utilize a cascading approach. Secondly, one would expect more experienced engineers to spend more time in the project realization phase than those with less experience. Thirdly, the quality of the solutions may not differ very much among the different levels of experience, but this remains to be seen. This hypothesis is based on the divergent results presented in Adams et al. (2003) and Atman et al. (2007). Fourthly, in the terms of rate of transitioning between different activities, prior literature would suggest no significant difference between the different populations.

To the above, we also add the conjecture that experts will spend less time reverting and less time adjusting than their less experienced counterparts, but that they will test their structures more often as part of their iterative design process.

5.2 Algorithm Implementation Specifics

We use the algorithm described in the General Algorithm section, organizing our data into ten clusters in the Segment Clustering step and four during the Participant Clustering step. The number of clusters was set to ten during segment clustering on the basis that this provided the best result for distinguishing among individuals of varying levels of experience. More specifically, when we compared the accuracy of the results from different cluster counts, we found that ten clusters produced the best differentiation between experience levels. Because our objective is to develop a model that helps us understand the differences between the different populations, we are not concerned with over-fitting or confirmation bias. Put differently, our goal is not to create a classifier meant to apply to another set of students. Instead, it is to study this population of students and identify patterns or characteristics that vary by experience level. Furthermore, after we have identified these characteristics we will use qualitative analysis to validate the reliability of the approach. In the case of participant clustering, four clusters were used to align with the four different levels of experience present in our sample population. However, we again note that the objective of this approach was less about making a classifier to predict experience, as it was about understanding the nature of expert experience.

5.2.1 Justification for Approach

As mentioned in the prior literature section, others have employed different approaches for analyzing this type of process data (e.g. Adams 2003; Atman et al., 1999, 2007, 2008). However, most of these

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

approaches have not maintained the temporality of the data, and have instead looked at each student's process in chunks, or in aggregate. Because we are looking for iterative cycles, it seemed fitting to compare entire sequences of actions, as opposed to looking at subsequences, as would be done in sequence mining. Additionally, in previous work, when we explored using non-process-based metrics, we found them far less successful in describing the role of experience in our dataset (Worsley & Blikstein, 2013).

5.3 Object Manipulation Generalizable Segments

During the first clustering phase, we identified the ten Generalizable Action Segments. Figure 17 highlights these differences along the five General Object Manipulation classes. Looking at the figure, there is one cluster for EVALUATE, one cluster for MODIFY, five clusters that represent REVERT, and three clusters related to different combinations of PLAN, REALIZE and MODIFY. To make naming of the clusters easier to follow, each cluster will be given a title that characterizes its primary constituents.

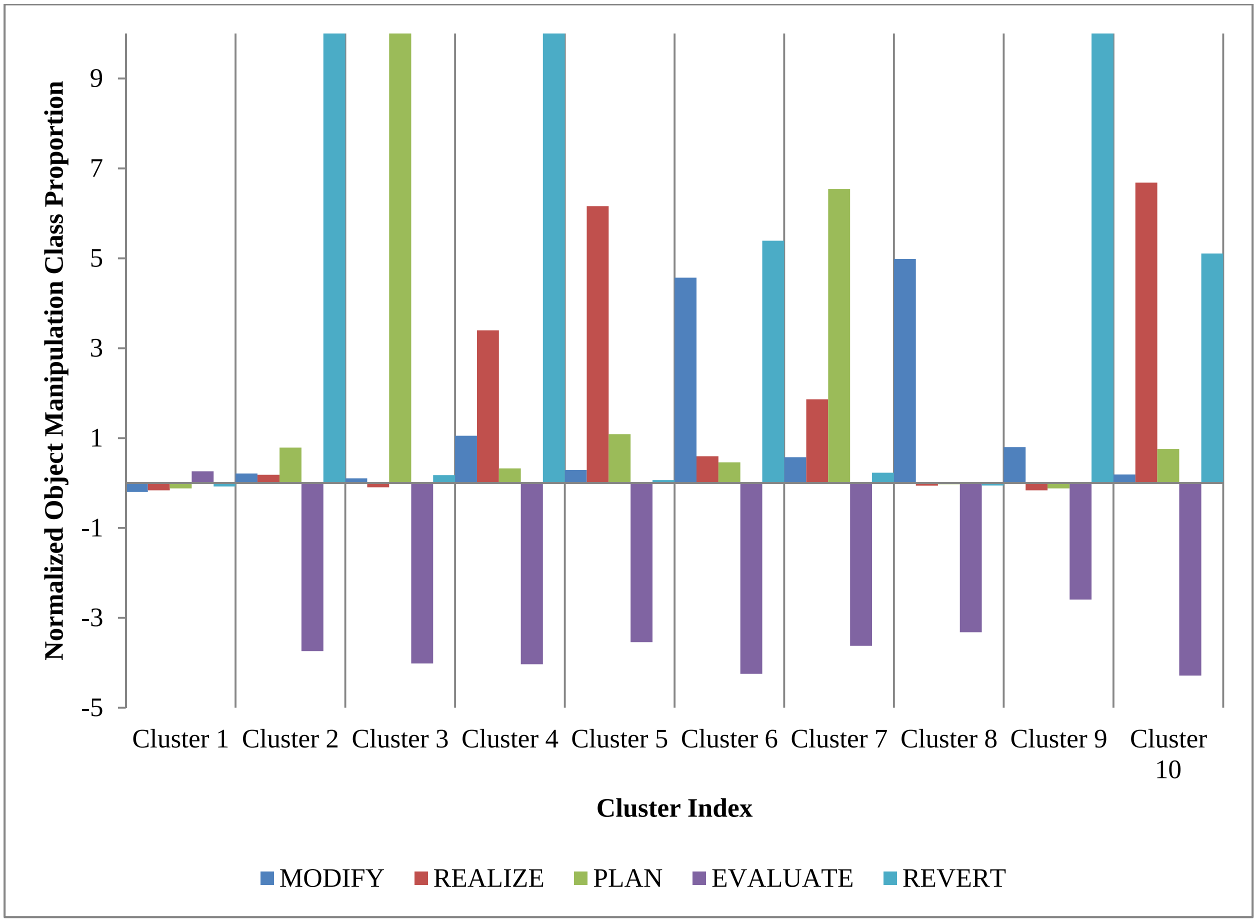

Figure 17. Object Manipulation Class Proportions by Cluster

5.3.1 EVALUATE Cluster

Cluster 1 represents the EVALUATE action, and was used for segmenting the sequence of actions. Accordingly, we expect this to be small in magnitude, and for all of the other clusters to include below average EVALUATE action proportions.

5.3.2 REVERT Clusters

Clusters 2, 4, 6, 9 and 10 (Figure 16) correspond to a large amount of REVERT. This suggests that undoing is an important behaviour to pay attention to when studying experience. However, simply looking at REVERT by itself is not sufficient. Instead, one needs to observe what other actions are taking place in the context of the REVERT action. In the case of cluster 2 (aka G-REVERT), the user is performing significant REVERT actions in the absence of any other action. This is in contrast to cluster 4 (G-REVERT-REALIZE), for example, where the user is completing a large number of REVERT actions, but is also doing several REALIZE actions. From this perspective, cluster 2 seems to correspond to doing a sustained REVERT, without any building. An example of this would be a student completely deconstructing the structure. Cluster 4, on the other hand, is more akin to undoing a few elements of one's structure with the intent of immediately modifying it. Put differently, cluster 4 may correspond to microscopic REVERT actions, whereas cluster 2 consists of more macroscopic REVERT actions. Clusters 6 (REVERT-MODIFY-REALIZE) and 9 (G-REVERT-MODIFY) appear to be characterized by a combination of REVERT and MODIFY actions. In this case, the user is undoing, not to make large structural changes to the design, but to make small adjustments. Clusters 6 and 9 are not identical, however. Cluster 6 also contains REALIZE actions. Cluster 10 (REALIZE-REVERT-PLAN) differs from the other REVERT clusters, in that REALIZE is the primary component, followed by REVERT and PLAN.

5.3.3 MODIFY Cluster

Cluster 8 (G-MODIFY) stands alone as a primarily MODIFY cluster, with below average values for all other actions.

5.3.4 PLAN, REALIZE, MODIFY Clusters

The remaining clusters — 3 (G-PLAN-MODIFY), 5 (G-MODIFY-REALIZE-PLAN), and 7 (G-PLAN-REALIZE-MODIFY) — can be characterized as different combinations of PLAN, REALIZE and MODIFY. Cluster 3 was dominated by PLAN actions, whereas clusters 5 and 7 include REALIZE and MODIFY actions.

In summary, we see that five of the cluster centroids emphasize REVERT actions, and the context in which they appear, while the remaining five are aligned with different proportions of EVALUATE, PLAN, REALIZE, and MODIFY actions. We can anticipate that there are distinguishing factors about how each of these are used that will help us as we examine the impact of experience on engineering practice.

5.4 Participant Clusters

Clustering students based on their pair-wise dynamic time-warped distances results in the precision and recall values presented in Table 9. Precision refers to the proportion of items identified for a certain class that actually belong to that class. Precision of 0.5 means that half the students placed into the low experience cluster were actually of low experience. Recall refers to the proportion of items belonging to a certain class that are correctly identified. Recall of one, means that all of the students of low experience were included in the low experience cluster.

Table 9. Precision and Recall for Cluster to Experience Alignment

Experience	Precision	Recall
Low	0.50	1.00
Medium	1.00	0.60
High	0.67	1.00
Expert	1.00	1.00

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

From Table 9, we see that the algorithm worked best at uniquely clustering Expert behaviour. It also attained recall of 1 for individuals of Low experience. Again, a recall of 1 means that all Low experience individuals were properly assigned to a single cluster. For individuals of intermediate experience, the algorithm was less accurate. Nonetheless, we reiterate that our primary objective is to understand better the patterns that distinguish "relative" experts from "relative" novices. We refer to the students as "relative" experts and novices because we did not employ a universal standard of expertise, but instead based their expertise on the amount of expert experience that they had. Hence, the majority of this analysis will be on examining how this representation of student actions was able to delineate between different levels of experience.

As a first indication that student behaviour differs by experience, we performed pair-wise Chi-Squared analyses (Table 10). Once again, in order to use Chi-Squared we treat every action of each participant individually. Making this simplification has limitations, provides a quick means for comparing across levels of experiences. From Table 10 it appears that all groups differed from one another in terms of usage of the different Generalizable Action Segments. Additionally, based on the Chi-Square statistics, we see that the pair-wise relationships follow the expected trend, with the most similar pairs (Expert-High, High-Medium, Medium-Low) having lower Chi-Square statistics than more dissimilar pairs (Expert-Medium, Expert-Low, High-Low). In order to pinpoint the nature of these differences we return to the three representations used in Analysis 1.

Table 10. Pair-wise Chi-Squared Analysis of Generalizable Action Segment Usage

Group 1	Group 2	Chi-Square Statistic	Probability
Expert	High	20.49911	0
Expert	Medium	47.26303	0
Expert	Low	260.6675	0
High	Medium	43.7318	0
High	Low	81.84317	0
Medium	Low	26.14816	0

5.5 Proportion-Based Graphical Representation of Participant Clusters

As before, we begin with a graphical representation of time spent in different activities. Among this first set of Generalizable Object Manipulation Segments (Figure 18) we see that G-REVERT is only used by individuals of Low experience, and G-REVERT-REALIZE is used more frequently among lower levels of experience. G-REVERT-MODIFY-REALIZE is only observed among individuals of Low and Expert experience. Finally, G-REVERT-MODIFY is relatively high for Medium experience individuals.

From Figure 19, we see that G-MODIFY is used extensively by individuals of all experience levels. We also observe that G-MODIFY-REALIZE-PLAN accounts for a larger proportion of user actions as experience level increases.

From Figure 20, we see that G-PLAN-MODIFY accounts for a larger proportion of time for High experience and Expert individuals than for Low and Medium experience individuals. G-PLAN-REALIZE-MODIFY also appears to follow this trend among individuals of Low, Medium, and High experience.

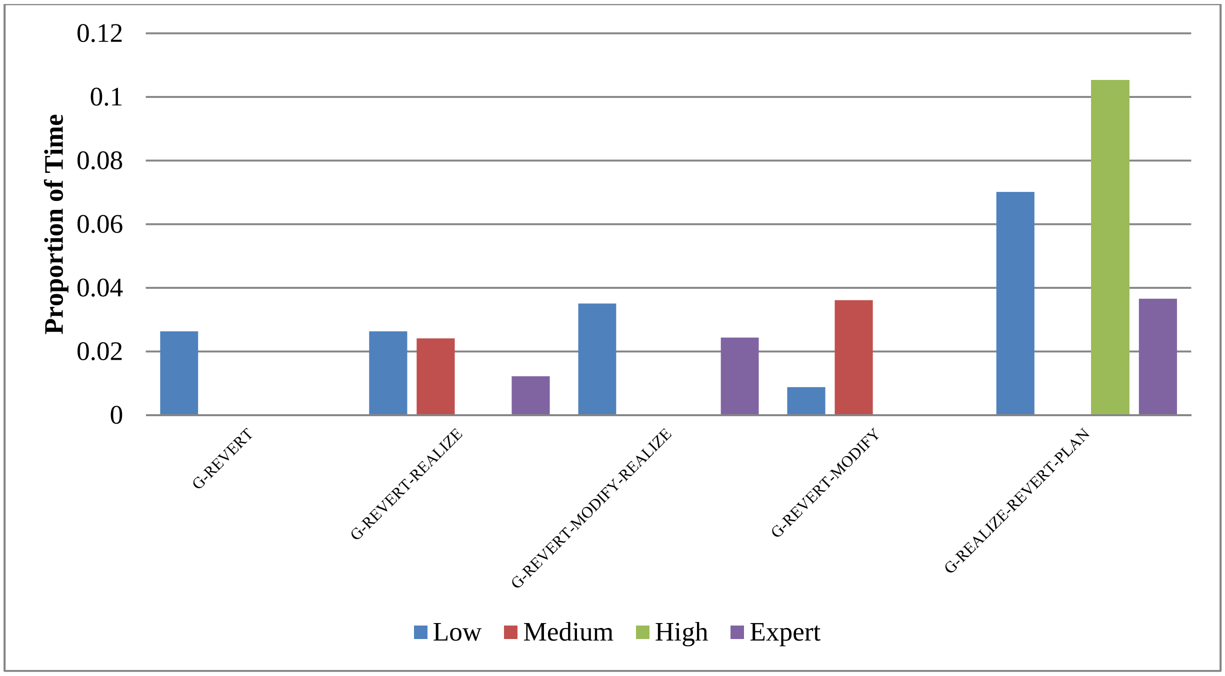

Figure 18. Proportion of Time Spent in each REVERT Action Segment by Experience

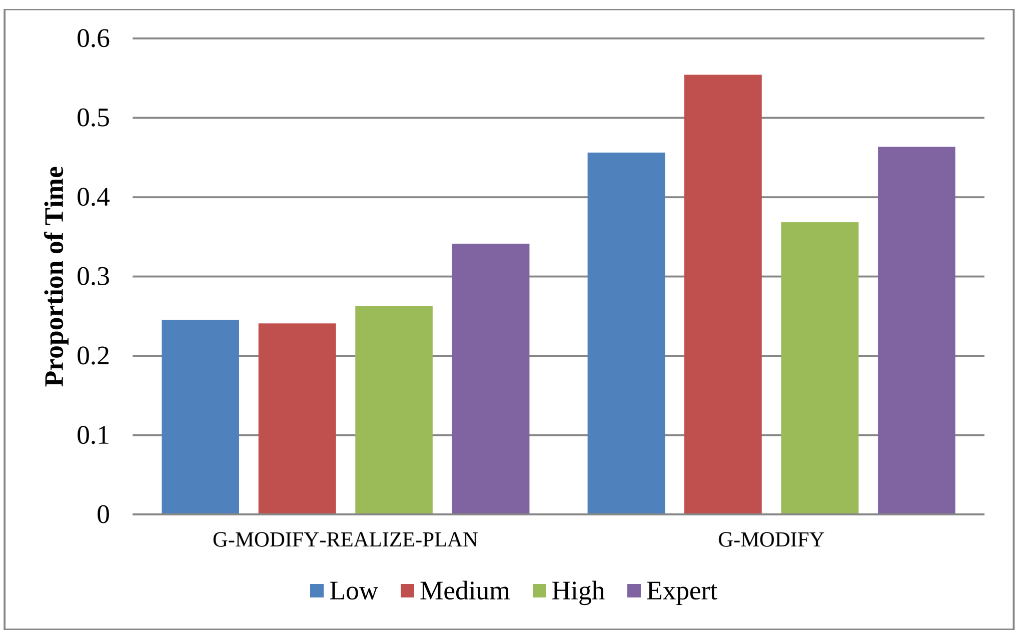

Figure 19. Proportion of Time Spent in each MODIFY Action Segment by Experience

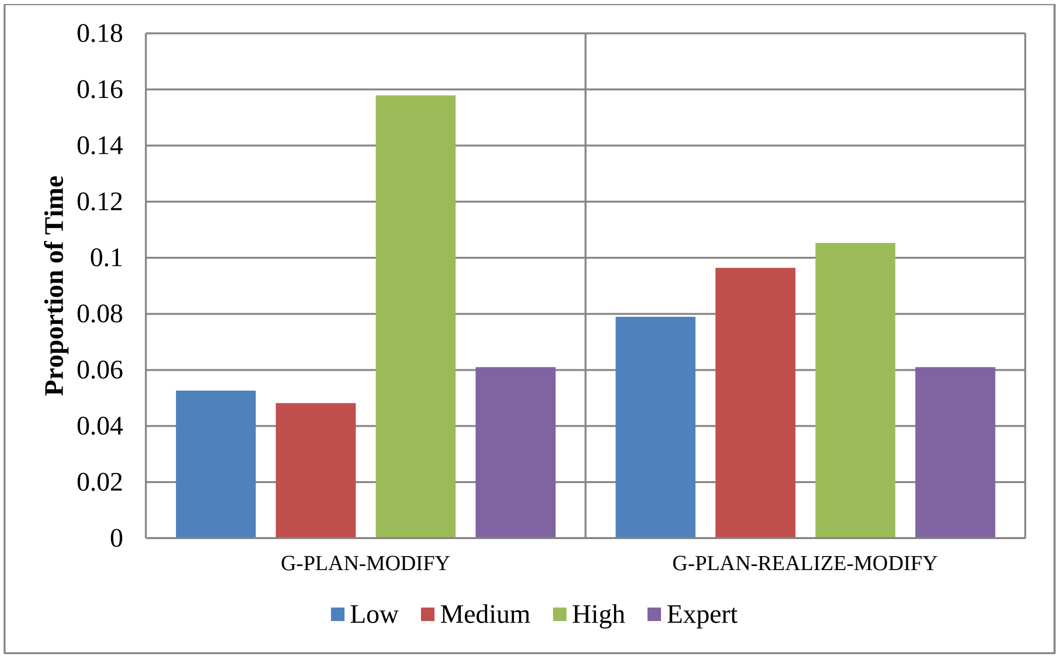

Figure 20. Proportion of Time Spent in each PLAN Action Segment by Experience

As we step back from the specifics of each generalizable segment, we observe that the Low experience population uses all five G-REVERT segments, Medium uses two of the five, High uses one of the five, and Expert uses three of the five. High experience and Expert individuals only used G-REVERT segments that also included REALIZE, with those of High experience having the additional constraint of only using G-REVERT segments that included both REALIZE and PLAN. This provides some initial indication that as experience increases, both PLAN and REALIZE become more central to the building process, which aligns with two of the hypotheses.

5.6 State–Transition Representation of Participant Clusters

In order to study how experience relates to behaviour more closely, we will again turn to a state–transition probability representation (Figure 21). As before, we compute the frequency of transitions between the different generalized segments and examine differences among our population of research participants. Through a pair-wise Chi-Square analysis of transition probabilities (Table 11) we see that all the transition behaviour of Expert does not significantly differ from that of High or Medium, but that significant differences exist among all other pairs. The lack of significant differences between Expert-High and Expert-Medium may initially seem odd, but when one considers that significant differences remain among the overall usage of individual behaviours, this becomes less problematic and may offer some meaningful insights into how experience impacts behaviour. However, we withhold the remainder of this discussion for a later section.

Because of the large number of states, we will only construct the diagrams for the six states associated with significant differences when comparing individuals of Expert experience to people of lower experience. These include the five REVERT states and G-PLAN-REALIZE-MODIFY. This observation in itself corroborates the idea that the frequency and context of REVERT actions is important when studying the role of experience on engineering practice.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

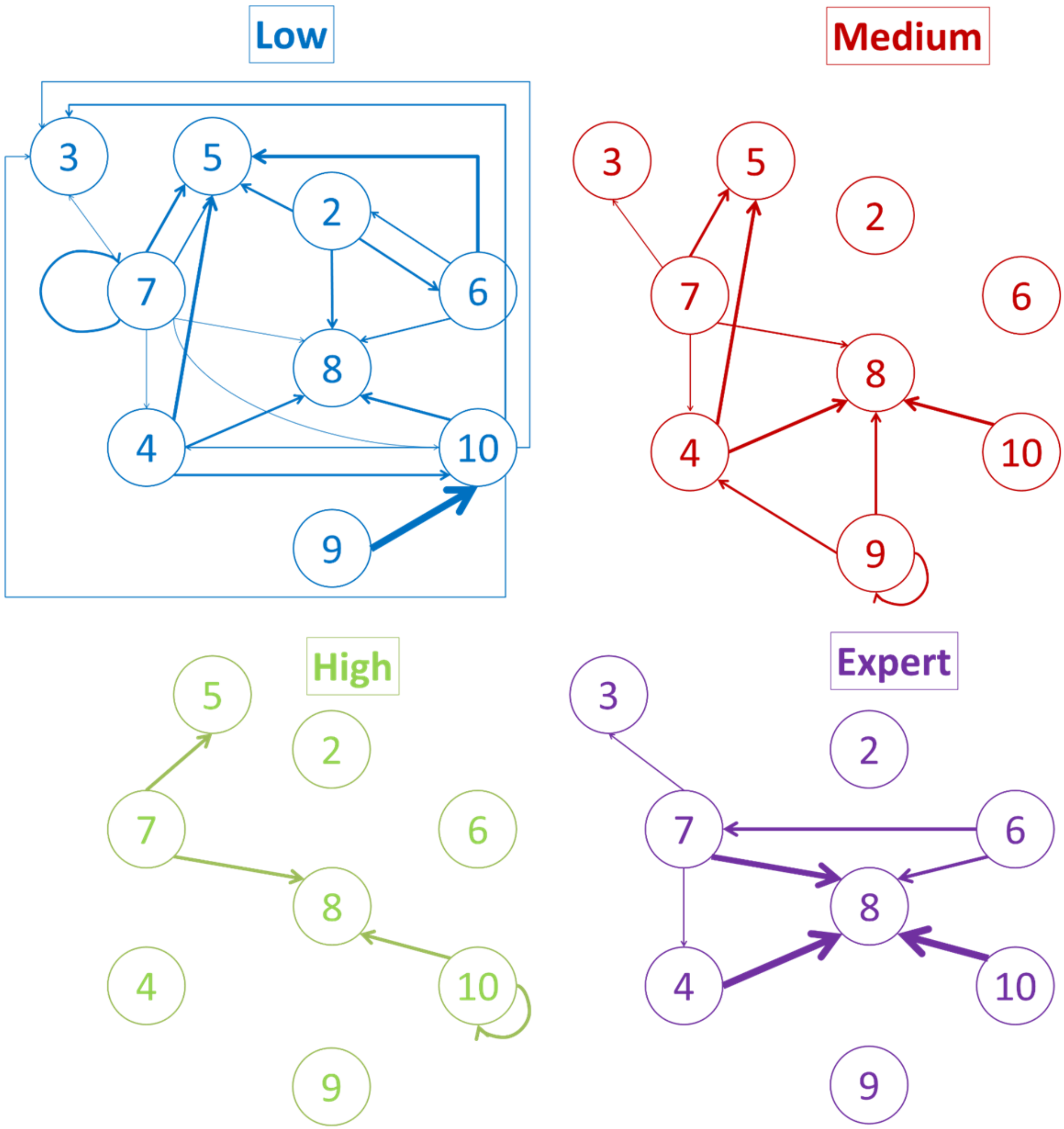

Figure 21. State–Transition Diagrams for Clusters of Different Experience Levels

5.6.1 Expert-Low Comparison

Expert and Low demonstrated differences in the nature and extent of structural undoing, as well as in what prompted them to adjust their structures. The Expert experience group typically engaged in modifications only after REALIZE actions. The Low experience group resorted to modifications through a much larger variety of previous actions.

Table 11. Pair-wise Chi-Square Analysis of Transition Probabilities

Group 1	Group 2	Chi-Square Statistic	Probability
Expert	High	90.25028	0.723537
Expert	Medium	108.9014	0.233152
Expert	Low	626.605	0
High	Medium	354.1774	0
High	Low	1071.301	0
Medium	Low	190.4553	0

These two groups differed in how they transitioned into six of the ten generalized segment: G-REVERT, G-REVERT-MODIFY, G-MODIFY, G-REALIZE-REVERT-PLAN, G-REVERT-REALIZE, and G-REVERT-MODIFY-REALIZE. From the Figure17, we observed that the Expert group never used the G-REVERT or G-REVERT-MODIFY states. They were less likely to transition into G-REALIZE-REVERT-PLAN and less likely to transition into G-REVERT-REALIZE. The Expert group would only transition into this state from G-MODIFY, whereas the Low group would transition into G-REVERT-REALIZE from three different states. This pair also differed in how they transitioned into G-MODIFY with Expert more likely to transition into G-MODIFY from previous states that included REALIZE actions.

5.6.2 Expert-Medium Comparison

Where Expert and Low demonstrated differences in the sequencing of building and modifying, Expert and Medium demonstrated differences in the context in which undoing actions were used. Specifically, the Expert group typically used more complex REVERT actions, meaning that the REVERT was used amidst several other actions.

The Expert group demonstrated differences from the Medium group. These differences were recorded in transitions into G-REVERT-REALIZE, G-REVERT-MODIFY, G-REVERT-MODIFY-REALIZE, and G-REALIZE-REVERT-PLAN. The Medium group never used the G-REVERT-MODIFY-REALIZE or G-REALIZE-REVERT-PLAN actions. On the other hand, the Expert group never used the G-REVERT-MODIFY action. Finally, for the G-REVERT-REALIZE state, the Expert group is more selective in its use, and only does so from G-MODIFY, whereas the Medium group only does so from G-PLAN-REALIZE-MODIFY and G-REVERT-MODIFY.

5.6.3 Expert-High Comparison

High and Expert groups differ in the nature of their planning behaviour. The Expert group is more likely to engage in planning behaviour that is in conjunction with project realization, whereas the High experience group was more likely to enter explicit and dedicated planning sessions.

Here we see differences in four classed: G-REVERT-REALIZE, G-REVERT-MODIFY-REALIZE, G-PLAN-REALIZE-MODIFY, and G-REALIZE-REVERT-PLAN. The High experience group never uses G-REVERT-REALIZE, or G-REVERT-MODIFY-REALIZE. They are less likely to transition into G-PLAN-REALIZE-MODIFY and do so from a smaller number of prior states. Finally, the High experience group is more likely to transition into G-REALIZE-REVERT-PLAN. In addition to these statistically significant differences, we also observed a trending difference on G-PLAN-MODIFY, which mirrors the observation from Figure 20. High experience individuals were more likely to transition into this state than Experts were.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

In summary, then, the distinguishing factors of the Expert group include:

- Their REVERT actions are always exercised in the context of REALIZE actions. This means that they would be more likely to undo a structure while completing a larger objective that might involve adding a new part of modifying an existing component.
- They use an iterative strategy that involves returning to planning, in the midst of building. For example, students in this group would complete a portion of their design and then enter another stage of planning. In contrast, students from other groups might engage in planning, but then simply move forward towards realizing their design without ever going back to planning.

5.7 Discussion

As we move into the qualitative analysis portion, we will see how many of the differences observed quantitatively are corroborated through video analysis.

5.7.1 Revert Action Context

A key observation made from the quantitative analysis is that Expert individuals complete REVERT actions in the context of REALIZE actions. In order to make this more evident, consider the structure in Figure 22. The user has just added two pieces of green tape to support the structure, and is about to test the strength of her structure.

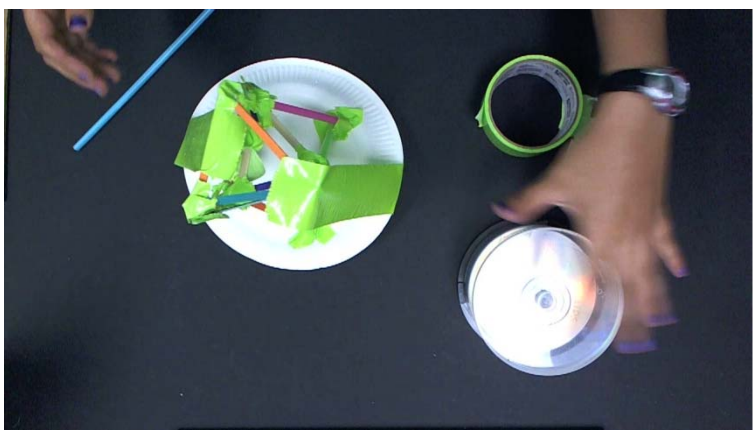

Figure 22. Example of Structure *Before* Undo

After testing the structure, however, she finds that it is not sufficiently stable and requires additional support. She therefore removes the two pieces of tape and employs the light blue straw instead (Figure 23). In this way, we see that while she did revert her design, it was not a matter of completely undoing the structure. Instead, she needed to find a better way to distribute the mass across the structure and correct for weaknesses.

5.7.2 Interspersing PLAN and REALIZE actions

The second observation is that the Expert individuals return to PLAN activities throughout the design process. One way to express this is through a timeline (Figure 24). The nodes on the graph correspond to different Generalizable Action Segments. For the sake of readability, this has been simplified by merging two of the segments that contain PLAN actions into segment number 1. Segment number 2 on the Y-axis corresponds with generalizable states associated with building and adjusting. Thus, we see that this expert began in a planning stage and then transitioned into building and modifying. After completing

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

that cycle of planning, building, and adjusting, the individual returns to planning again around time step 60 on the x-axis, and repeats the process. If we examine the video content at this point (Figure 25), we see that the individual has managed to complete the base of their structure and is now considering what to do next. In fact, one observes in the image that the user is testing the material again while reasoning about the next steps.

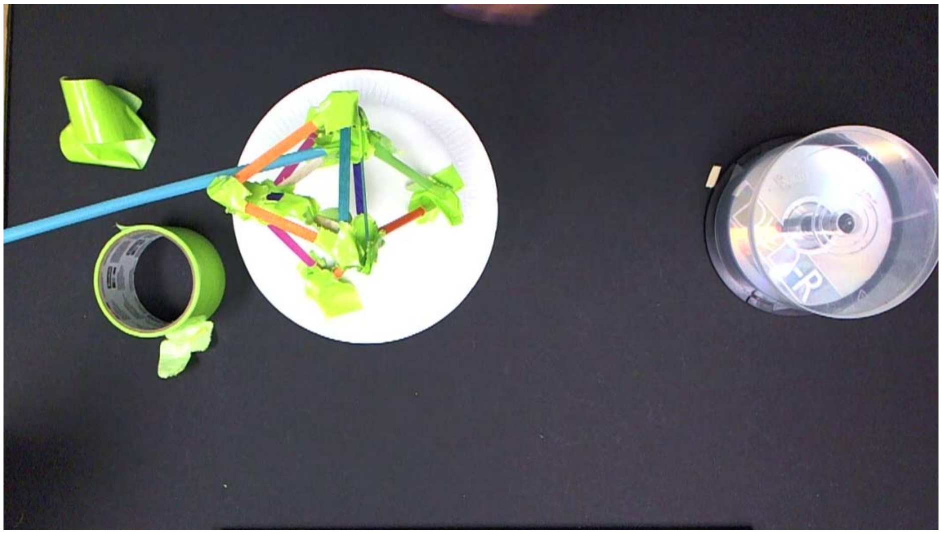

Figure 23. Example of Structure *After* Undo

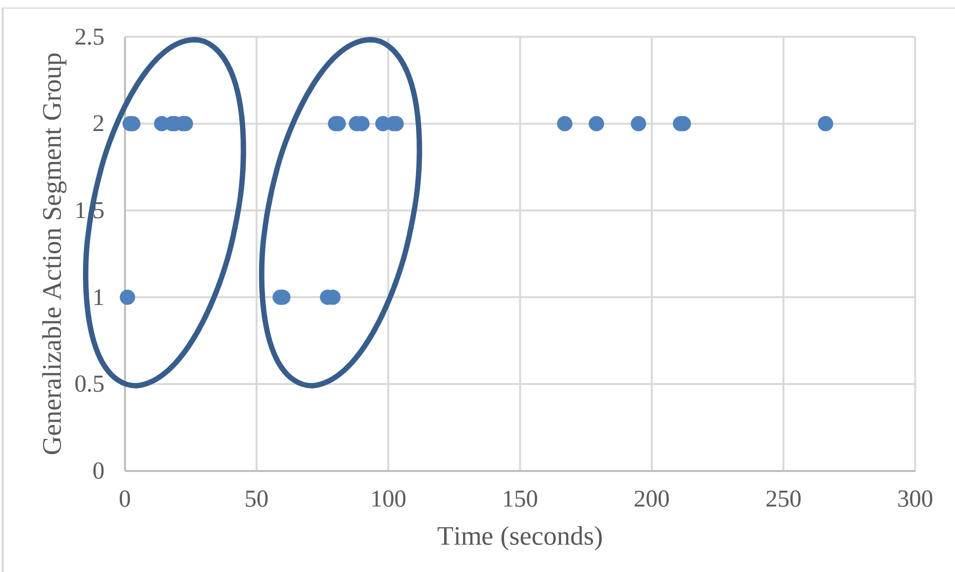

Figure 24. Sample Expert Timeline (1 = PLANNING, 2 = BUILDING and ADJUSTING). Typical Plan and then Build and Adjust cycles are enclosed within the ovals.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

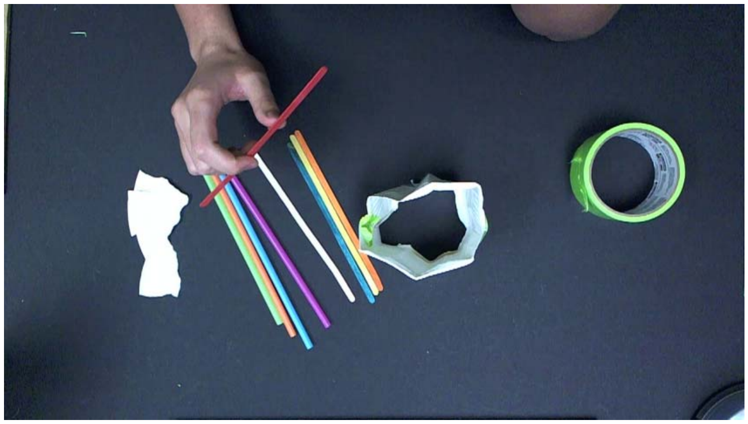

Figure 25. Expert Structure after first iteration of planning, realizing, and adjusting

Later on, we see that the student has made several new additions to the structure (Figure 26), but these additions were only conceptualized during the second iteration of planning and building.

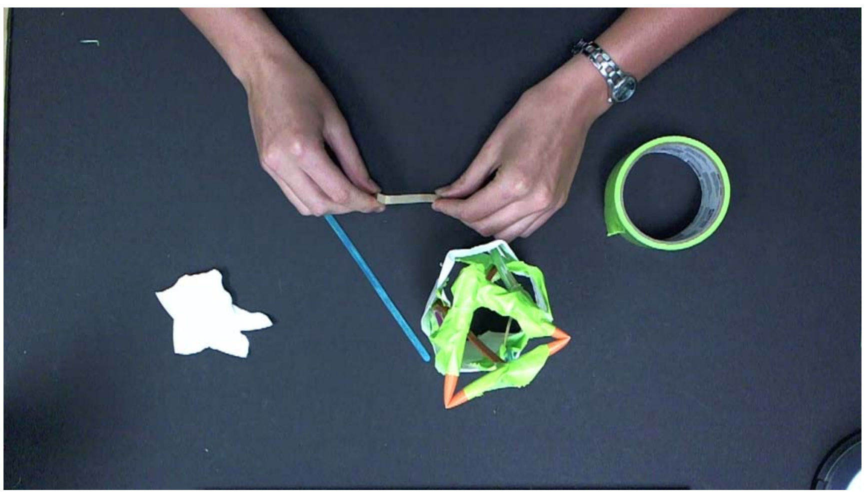

Figure 26. Expert Structure after Second Iteration of Planning, Realizing and Modifying

To corroborate this approach further, the participant described the process as being iterative and requiring flexible planning.

> "I thought that I was going to make this my base [pointing at the folded paper plate in [Figure 25]. And then when I realized that I had extra materials left over, I decided to try and add more height underneath, so that was unexpected. I thought I was going to have this be the base, plus struts coming up."

Once again, there are elements of this analysis that could have been completed using purely qualitative analysis. However, garnering these results, and at a level of granularity that explicates the different types of REVERT actions and the context in which REALIZE and PLAN actions are completed, would have been quite challenging. Through the affordances of computational analysis, we are able to focus the human analysis component on the output of the different algorithms being used.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

6 DISCUSSION

In the first analysis, we used machine learning in conjunction with human coding to cluster students based on their action sequences. Those sequences were first transformed into generalized sets of action segments as determined through k-means clustering. The transformed sequences were then used to compute pair-wise distances between all study participants. This process-based analysis generated four clusters of students that differed along two dimensions. The first dimension relates to how students engaged in the design process. Consistent with prior literature, we identified a group of students that employed iterative design practices (e.g. Atman & Bursic, 1998) and others who tended to follow less systematic approaches (failing to iterate, or use incremental development strategies). While we did not focus on students' level of experience in that analysis, we did observe that the majority of individuals in the iterative design group had Expert-level experience with engineering. At the same time, being among the iterative designers did not necessarily mean that an individual had Expert-like experience. This is consistent with the prior literature in this domain, which states that novices can also use iterative design strategies.

We also observed that the clustering analysis differentiated students on the axis of idea quality. We saw that students who spent considerable time undoing previous actions oftentimes overlooked key engineering intuitions that would have made their structures more stable. Without knowledge of how to correct their problems, students employed noticeably different building patterns. What this means for engineering education, especially at the K–12 level, is that we cannot focus only on the engineering design process. Instead, we have to ensure that students also find ways to develop their knowledge about deep engineering intuitions. Examining the extent to which employing iterative design helps students investigate engineering principles in order to develop more accurate intuitions about how their structures will behave. In particular, activities based on pure "trial and error" are likely to be inefficient at helping novices effectively decipher sound engineering principles. This is significant because in many engineering and "maker" programs at the K–12 level, there is a popular belief that letting students tinker will eventually generate more advanced knowledge about engineering and computer programming. However, it is quite possible that many of these students are learning about the engineering design process, without truly gaining insights into engineering principles.

Additionally, using this type of process-based analysis can help educators gain a deeper understanding of a student's conceptual challenges. For example, students who are unsuccessful at an assigned task may be dealing with challenges in the design process, in engineering intuitions, or in basic engineering principles, or any combination of those. Furthermore, even for those students who are successful in completing a given task, identifying where the student falls on these two dimensions can help instructors streamline their learning interventions, and reduce the likelihood that teachers provide students with non-applicable suggestions.

In the second analysis, we performed a more targeted comparison of how more experienced students tackle engineering design challenges. Using the same dataset and the same overall approach as the first analysis, we again identified a set of generalizable action segments. These generalized segments are more fine-grained than those from the first analysis, and consisted of five variants of undoing (e.g. G-REVERT-MODIFY), a variant of modifying (G-MODIFY), and three variants of the combination of planning, realizing, and modifying (e.g. G-PLAN-MODIFY). The level of specificity in these segments helped us to realize the different ways that the five basic actions (EVALUATE, REVERT, MODIFY, REALIZE, PLAN) are used. For example, some REVERT actions were completed in the absence of any other actions, while

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

others were completed in conjunction with different proportions of REALIZE, MODIFY and PLAN. At a higher level of comparison, the second analysis also demonstrated how individuals with different levels of experience use the five basic actions. Even when the same action is used, it often may be employed in a different context. For example, Experts had a tendency to return to planning midway through the process, while other groups would do batch planning. Hence the scope of a given PLAN action differed by experience, with Experts using the PLAN step to address shorter-term goals and objectives. In this way, the second analysis helped disambiguate among similar actions, and suggested ways that experience impacts how students approach engineering design and how it evolves over time and practice.

7 LIMITATIONS AND FUTURE WORK

A key portion of this analysis was the human labelled video data. This provided a coarse-grain sequencing of each participant's actions. One future direction for this research is to leverage the hand-labelled data, in conjunction with computer vision and the gesture data, to label student actions automatically. However, to date, the open-ended nature of the task means that individuals go about enacting each of the possible actions in very different ways. This has been the primary hindrance to training a classifier to detect the different actions accurately. Nonetheless, as the field continues to improve the sophistication of multimodal analysis techniques, automatically extracting data from open-ended tasks will become increasingly feasible.

Another area for future work is to examine the extent to which employing iterative design helps students investigate engineering principles. Analysis 1 presented quality of design and quality of idea as orthogonal dimensions. However, it may be that the two dimensions are related to one another.

8 CONCLUSION

With the expansion of "making" in education, complex hands on learning environments are receiving a lot of attention without having a significant research base. The analyses reported in this paper were motivated by a desire to study complex hands on learning. Furthermore, the goal was to identify some keys insights into understanding how study develop and demonstrate proficiency within the hands-on learning context. Traditionally, analyzing video from hands-on learning has been extremely difficult, labour intensive, and hard to describe in discrete quantitative terms. It is also challenging to see the evolution of subtle patterns that might not reveal themselves using traditional statistical approaches. However, contrary to prior research in this area, our primary data source was not student speech but student actions. Because of the lack of computational approaches for extracting this data automatically, we relied on human coding to provide time-stamps of when students started and stopped different object manipulation actions. We then took this data and performed a sequence of machine learning algorithms in order to study general building practices and to highlight manifestations of relative expertise in engineering design actions. The first analysis showed how we can garner similar results to prior qualitative research, but by using computational clustering techniques and dynamic time warping. It also highlighted that idea quality and design process are two prominent dimensions through which to compare and contrast student development. In the second analysis, we showed how a similar approach could also be used to better identify the characteristic behaviour differences between individuals with various levels of prior experience. Specifically, we showed that both the intent and context of engineering practices is closely related to students' level of experience. For both studies, we validated our findings through visual and qualitative representations that helped us distill what the different clusters of action segments and different clusters of participants might tell us about engineering design patterns.

As we close, we want to take a step back and consider the larger implications of this work, beyond improving our understanding of engineering practices. There is a tremendous opportunity for Learning Analytics to intersect with qualitative research methods to tackle questions that do not lend themselves to easy data extraction. Embarking on work that lies at this intersection will continue to help bridge the learning community and the analytics community. To date, we know of very few instances of Learning Analytics research that takes human labelled data and exhibits how computational analysis can mirror, and extend, approaches and results achieved through traditional education research. If we can show them robust methods for streamlining their analyses, while also permitting them to remain in their current areas of specialization, we have the potential to bring Learning Analytics more squarely into the fold of education research. This will serve to improve the quality and scalability of current education research, and thus increase the impact of Learning Analytics.

REFERENCES

Adams, R. S., Turns, J., & Atman, C. J. (2003). Educating effective engineering designers: The role of reflective practice. *Design Studies, 24*(3), 275–294.

Atman, C. J., Adams, R. S., Cardella, M. E., Turns, J., Mosborg, S., & Saleem, J. (2007). Engineering design processes: A comparison of students and expert practitioners. *Journal of Engineering Education, 96*(4), 359–379.

Atman, C. J., & Bursic, K. M. (1998). Verbal protocol analysis as a method to document engineering student design processes. *Journal of Engineering Education, 87*(2), 121–132.

Atman, C. J., Chimka, J. R., Bursic, K. M., & Nachtmann, H. L. (1999). A comparison of freshman and senior engineering design processes. *Design Studies, 20*(2), 131–152.

Atman, C. J., Deibel, K., & Borgford-Parnell, J. (2009). The process of engineering design: A comparison of three representations. *Proceedings of the 17th International Conference on Engineering Design (ICED '09).* August 24–28, Palo Alto, CA, USA (pp. 483–494).

Atman, C. J., Kilgore, D., & McKenna, A. (2008). Characterizing design learning: A Mixed-Methods study of engineering designers' use of language. *Journal of Engineering Education, 97*(3), 309–326.

Baker, R., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining, 1*(1), 3–17.

Berland, M., Martin, T., Benton, T., Ko, P., & Petrick-Smith, C. (2013). Using learning analytics to understand the learning pathways of novice programmers. *Journal of the Learning Sciences*, 22(4), 564–599.

Bienkowski, M., Feng, M., & Means, B. (2012). Enhancing teaching and learning through educational data mining and learning analytics: An issue brief. *Washington, DC: SRI International*.

Blikstein, P. (2011). Using learning analytics to assess students' behavior in open-ended programming tasks. *Proceedings of the First International Conference on Learning Analytics and Knowledge* (LAK '11). February 27–March 1, Banff, AB, Canada (pp. 110–116). New York: ACM.

Blikstein, P. (2013). Multimodal learning analytics. In Dan Suthers, Katrien Verbert, Erik Duval, and Xavier Ochoa (Eds.), *Proceedings of the Third International Conference on Learning Analytics and Knowledge (LAK '13)* (pp. 102–106). New York: ACM.

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S. & Koller, D. (in press). Programming pluralism: using learning analytics to detect patterns in novices' learning of computer programming. *Journal of the Learning Sciences.*

Chi, M. T., Glaser, R., & Rees, E. (1981). *Expertise in problem solving.* Learning Research and Development Center, University of Pittsburgh.

Cross, N., & Cross, A. C. (1998). Expertise in engineering design. *Research in Engineering Design, 10*(3), 141–149.

Dong, A., & Agogino, A. M. (1997). Text analysis for constructing design representations. *Artificial Intelligence in Engineering, 11*(2), 65–75.

Dym, C. L. (1999). Learning engineering: Design, languages, and experiences. *Journal of Engineering Education, 88*(2), 145–148. doi:10.1002/j.2168-9830.1999.tb00425.x

Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. *Journal of Engineering Education, 94*(1), 103–120.

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, *100*(3), 363.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological review*, *87*(3), 215.

Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM, 51*(8), 25–27.

Koedinger, K. R., & Corbett, A. T. (2006). Cognitive tutors: Technology bringing learning science to the classroom. *The Cambridge Handbook of the Learning Sciences* (pp. 61–78)*.* Cambridge: Cambridge University Press.

Lau, K., Oehlberg, L., & Agogino, A. (2009). Sketching in design journals: An analysis of visual representations in the product design process. *Engineering Design Graphics Journal, 73*(3), 23–29.

Lawler, R. W., & Yazdani, M. (1987). *Artificial Intelligence and Education: Learning environments and tutoring systems* (Vol. 1). Bristol, UK: Intellect Books.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012). Modeling how students learn to program. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education,* (SIGCSE '12) (pp. 153–160). New York: ACM.

Resnick, M. (2002). Rethinking Learning in the Digital Age. In *The Global Information Technology Report: Readiness for the Networked World*. Edited by G. Kirkman. Oxford University Press.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., & Silverman, B. (1998, January). Digital manipulatives: New toys to think with. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 281–287). New York: ACM Press/Addison-Wesley.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

(2014). Analyzing Engineering Design through the Lens of Computation. *Journal of Learning Analytics*, *1*(2), 151-186.

Russ, R. S., Scherr, R. E., Hammer, D., & Mikeska, J. (2008). Recognizing mechanistic reasoning in student scientific inquiry: A framework for discourse analysis developed from philosophy of science. *Science Education, 92*(3), 499–525.

Siemens, G., & Baker, R. S. D. (2012). Learning analytics and educational data mining: Towards communication and collaboration. *Proceedings of the Second International Conference on Learning Analytics and Knowledge* (pp. 252–254). New York: ACM.

Turkle, S., & Papert, S. (1991). Epistemological pluralism and the revaluation of the concrete. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 161–191). New York: Ablex Publishing Corporation.

U.S. Department of Education, Office of Educational Technology (2010). Transforming American Education: Learning Powered by Technology. National Education Technology Plan 2010. Washington, DC.

Wilensky, U., & Reisman, K. (2006). Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology through Constructing and Testing Computational Theories—An Embodied Modeling Approach. *Cognition and Instruction*, *24*(2), 171–209.

Worsley, M. (2012). Multimodal learning analytics: enabling the future of learning through multimodal data analysis and interfaces. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction* (pp. 353–356). New York: ACM.

Worsley, M., & Blikstein, P. (2011). What's an expert? Using learning analytics to identify emergent markers of expertise through automated speech, sentiment and sketch analysis. *Proceedings of the Fourth Annual Conference on Educational Data Mining (EDM 2011).* Eindhoven, the Netherlands (pp. 235–240).

Worsley, M., & Blikstein, P. (2013). Towards the development of multimodal action based assessment. *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (LAK '13), 8–12 April 2013, Leuven, Belgium (pp. 94–101). New York: ACM.