**Proceedings of the ASME 2021
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE2021
August 17-2, 2021, Virtual Conference**

IDETC2021-72406

DESIGN EMBEDDING: REPRESENTATION LEARNING OF DESIGN THINKING TO CLUSTER DESIGN BEHAVIORS

Molla Hafizur Rahman
Department of Mechanical Engineering
University of Arkansas
Fayetteville, AR

Charles Xie
Institute for Future Intelligence
Natick, MA 01760, USA

Zhenghui Sha[1]
Department of Mechanical Engineering
University of Arkansas
Fayetteville, AR

ABSTRACT

Design thinking is essential to the success of a design process as it helps achieve the design goal by guiding design decision-making. Therefore, fundamentally understanding design thinking is vital for improving design methods, tools and theories. However, interpreting design thinking is challenging because it is a cognitive process that is hidden and intangible. In this paper, we represent design thinking as an intermediate layer between human designers' thought processes and their design behaviors. To do so, this paper first identifies five design behaviors based on the current design theories. These behaviors include design action preference, one-step sequential behavior, contextual behavior, long-term sequential behavior, and reflective thinking behavior. Next, we develop computational methods to characterize each of the design behaviors. Particularly, we use design action distribution, first-order Markov chain, Doc2Vec, bi-directional LSTM autoencoder, and time gap distribution to characterize the five design behaviors. The characterization of the design behaviors through embedding techniques is essentially a latent representation of the design thinking, and we refer to it as design embeddings. After obtaining the embedding, an X-mean clustering algorithm is adopted to each of the embeddings to cluster designers. The approach is applied to data collected from a high school solar system design challenge. The clustering results show that designers follow several design patterns according to the corresponding behavior, which corroborates the effectiveness of using design embedding for design behavior clustering. The extraction of design embedding based on the proposed approach can be useful in other design research, such as inferring design decisions, predicting design performance, and identifying design actions identification.

Keywords: Design thinking, design embedding, design cognition, deep learning.

1. INTRODUCTION

Design thinking guides how designers apply the design principles to generate, evaluate, and represent concepts to meet stated goals [1,2]. In the context of engineering design, design thinking refers primarily to the exploration (i.e., divergent thinking) and exploitation (i.e., convergent thinking) iterations in search of design solutions [3]. More generally speaking, design thinking is designers' cognitive activities during a design process. Their decision-making strategies in the design process are guided by their design thinking, and their corresponding actions are reflected through the design task. Therefore, design thinking works as a bridge that connects designers' knowledge space and design space [4], as shown in Figure 1. A deeper understanding of design thinking is vital for advancing design theories, methods, and tools.

However, understanding and interpreting design thinking are challenging because it is intangible and occurs in the human brain [5]. During a design task, different designers may adopt different design strategies. Thus, the design behaviors that reflect their design thinking are different too [2]. This is particularly true in complex systems design, where the problem often involves various design variables and constraints. For example, in one of our previous studies, several design patterns were identified in the same solar system design task by studying designers' one-step sequential decision-making behaviors [6]. In order to fundamentally understand design thinking, various empirical studies have been conducted based on different methodologies, such as protocol methods, controlled experiments, psychological tests, and neuroscientific measurement, such as functional magnetic resonance imaging (fMRI) [2]. While existing studies have leveraged the advancement in machine learning and data mining techniques in discovering behavioral patterns in design from which we draw insights and inferences about their design thinking [2], little research was done on understanding the latent

[1] Corresponding author: zsha@uark.edu

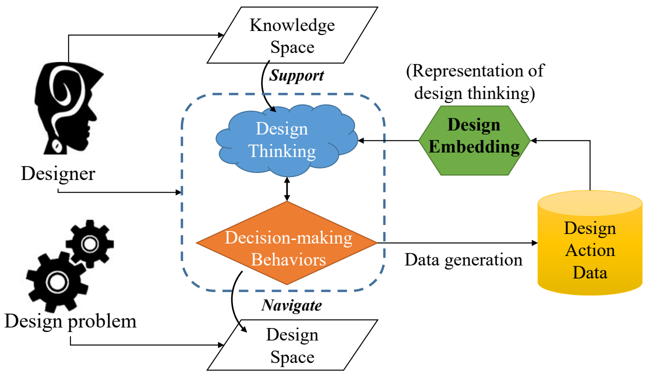

FIGURE 1: THE CONNECTION BETWEEN DESIGN THINKING AND DESIGN EMBEDDING

representations of design thinking. We define the representation of design thinking as an intermediate layer between human designer's mental processes (i.e., the thought process) and their behaviors (i.e., actions). Our hypothesis is that the design thinking representation is potentially an essential and effective pathway to the empirical studies of designers' thinking.

Now suppose design thinking is an abstraction and mapping of design behaviors at a high-dimensional space, then the understanding of design thinking must not be acquired from a single behavioral type. If multiple-dimensional design behaviors and the corresponding patterns are identifiable, then a series of questions are, would the representation of design thinking extracted from different design behaviors be different? How does the representation of design thinking obtained from each dimension of the design behavior look like? What are the ways, particularly computational methods, to extract such representations?

As the first attempt to answer these questions, on the one hand, we identify five different design behaviors, including one-step sequential behavior, long-term sequential behavior, contextual behavior, reflective thinking, and design action preference, based on current research on design theories. Each of these design behaviors is elaborated in Section 2. On the other hand, we explore the possibility of using embedding techniques from machine learning to transform high-dimensional design action data into low-dimensional embeddings, referred to as ***design embeddings***, for the latent representation of design thinking. In machine learning literature, an *embedding* is a low-dimensional vector representation of high-dimensional data [7]. Embedding maps discrete, categorical variables to a vector of continuous numbers. Figure 1 illustrates the connections among design thinking, design embeddings, and design behaviors.

In this study, our assumption is that design thinking is reflected by design behaviors in multiple dimensions. Therefore, by abstracting and extracting the latent representation of design behavioral data in a transformed dimension via embedding techniques, design thinking can be better characterized. Particularly, we develop an approach that applies different embedding techniques to learn design thinking representations from designers' action data. The scope of this study is focused on computer-aided design (CAD) for the ease of data collection. However, the approach is applicable in any design context as long as the design action data can be collected. This approach is demonstrated using the data collected from a high-school student CAD challenge where participants are asked to solarize their school with the required energy yield and payback period (see Section 4.2 for detail).

The remaining of the paper is as follows. In Section 2, we present the literature review on design thinking studies and summarize the common representations of design thinking in various data types. In Section 3, we present the overall research approach and discuss the technical background regarding the different embedding techniques adopted in this study. In Section 4, a case study on the solar system design challenge is presented. Also, we discuss the experiment details and data collection method in this section. The results are presented and discussed in Section 5. Finally, in Section 6, we end this paper by drawing conclusions and insights as well as a summary of limitations which opens up the opportunities for our future work on the topic of design thinking representation.

2. LITERATURE REVIEW

2.1 Representation of design thinking

Extensive studies have been conducted to study design thinking. These studies adopted various ways to represent design thinking, such as by using cognitive study (e.g., protocol study, controlled experiment), physiological measurement (e.g., eye tracking, heart rate, electrocardiography(ECG)), neurological signals (e.g., electroencephalogram (EEG), functional magnetic resonance imaging (fMRI)) [8]. In protocol and controlled study, design data are encoded by ontological design model (i.e., function-behavior-design (FBS) design process model), which are collected from protocol study or controlled experiment [9]. These design data are typically designers' performed actions [10] and are further encoded to a deeper understanding of design thinking [11]. The encoded design data is analyzed by different computational methods in order to represent design thinking. For example, the first-order Markov chain model representing one-step sequential decision-making behavior is utilized to study design patterns [6,12]; the hidden Markov model is used to identify hidden design states [10]; and the long short-term memory (LSTM) unit model is used to predict future design process stages [13]. In some studies, sketch data are collected besides the verbal and design action data [14]. Sketching is further encoded using different sketch coding methods (e.g., C-sketch method [15]) to represent design thinking.

Design thinking is also studied using various physiological measures such as eye-tracking, ECG, and facial recognition. In the eye-tracking method, eye-tracking devices and software capture designers' eye movement and provide gaze points and heat maps of areas of interest [16]. Both the heat maps and gaze points, thereby, represent designers' thinking. This method mainly analyzes how much attention designers put on the area of a specific design object, and the data have been used to study design creativity [17] and how designers analyze the functionality of a design object [18]. Using ECG, heart rate variability (HRV) signals can be recorded and connected to mental stress [19]. HRV is measured during the different design segments, and the corresponding mental stress is measured. Different designers show different patterns of stress according to their design thinking.

Data collected from neurological studies try to connect design thinking and brain activity. The two most popular methods for neurological studies are EEG and fMRI. While EEG measures neural activity via the identification of electrical current, fMRI measures brain activity by the brain's blood flow using a magnetic field [8]. From the EEG data, the power spectral density of brain waves is also measured, and the correlation between design activity and brain waves is analyzed [20]. Data from fMRI are images of the brain at cross-sections that provide visual reasoning, such as brain activation patterns during design ideation [21]. Recent studies conducted by neurocognition scientists indicated that when designers engaged in divergent thinking, different cognitive domains were activated with the tasks that require analysis during the engineering concept generation [22]. Design neurocognition researchers also have successfully encapsulated the cognitive functioning behind engineering design [23]. This empirical research confirmed that design thinking is not merely an abstract construct. However, the external design behavior regulated by different cognitive processes involved during the search of design solutions requires further investigation through the study of design actions [24].

2.2 Behaviors in the design process

The design process involved various behaviors, among which sequential behavior is considered an integral part [25] and a natural feature of design competency [26]. Many types of research have been conducted to study designers' sequential behavior using the Markov chain model. Typically, the first-order Markov chain model is utilized to study designer transition behavior or one-step sequential behavior. This behavioral study is used to identify design patterns [8,27] and study designers' sequential learning process [28]. The Second-order and higher-order Markov chain model represents short-term sequential behavior. For example, to compare the design process between two design domains: architects and software designers, a second-order Markov chain has been implemented [29]. The higher-order Markov model is adopted in an agent-based modeling framework to study the effect of memory on sequential behaviors [30]. The hidden Markov model (HMM) is also used to understand designers' sequential design strategy. For example, HMM is used to extract design strategies to create a computer agent that can solve truss design problems [31]. Deep learning-based models are also utilized which are capable of capturing both long-term and short-term sequential behaviors. For example, in our previous study [6], by using the LSTM model, it is verified from the prediction that designers use both long-term and short-term memory effectively in a design process.

In addition to different sequential behaviors, studies have also been conducted on other types of behaviors, such as reflective thinking. Reflective design thinking is a conscious mental activity that examines designers' design actions, decisions, and inner selves throughout a design process [32]. Though the study of reflective thinking is a growing trend, very few studies have been conducted on design reflection [33]. Goldstein et al. [33] use designers' electronic notepad and pre-test and post-test to study designer reflective thinking and found that moderately reflective students understood design activities better than those with high or low reflectivity. Even though many studies on design behaviors have been conducted, most of them focus on a particular design behavior at a time. However, design thinking is not merely a particular design behavior; rather, it is an abstraction of design behaviors from multiple dimensions. Therefore, to a deeper understanding of design thinking, a study on different design behaviors is needed.

3. TECHNICAL BACKGROUND AND RESEARCH APPROACH

In this section, we first briefly introduce the research approach adopted in this study. Next, we present the technical background for different embedding techniques.

3.1 Theoretical background

One of the major contributions of this study is the identification of five design behaviors for studying design thinking representation. Therefore, before describing the overall research approach, we would like to present the rationale of how the five behaviors are identified. These behaviors include one-step sequential behavior, contextual behavior, long-term sequential behavior, reflective thinking behavior, and action preference.

The one-step sequential behavior, contextual behavior, and long-term sequential behavior are selected based on the mental iteration model [34]. Design is a goal-directed problem-solving process and can be modeled as an iterative and sequential decision-making process. Jin and Chuslip [34] proposed a cognitive model to describe the mental iteration during design. According to that model, in every design process, several cognitive activities occur, such as generate, compose, evaluate, etc. Also, different iteration loops are embedded in the design process. These loops collectively generate a global loop. Besides the global loop, each cognitive activity defines a local loop. In complex systems design problems, these loops frequently occur as designers go back and forth iteratively between different stages to search the design space and take different design actions to accomplish required design tasks. Therefore, in this study, we propose to use one-step sequential behavior and contextual behavior (short-term behaviors) to capture the local-loop behavioral patterns and use long-term sequential behaviors to capture the global-loop iterative patterns.

Next, we consider reflective thinking. The core of reflective thinking is metacognition and self-monitoring, which help designers to reflect experience and knowledge in their actions as well as provide feedback to improve the design process [35]. In a design process, designers may take various modes of reflective thinking. For example, some designers use a bigger picture (take a longer time to think) while others use a micro-scoping view (take a shorter time to think). Reflective thinking behavior enables designers to scrutinize their thinking, behavior, design process, thus produce higher quality designs [36,37]. Therefore, understanding and computationally modeling designer reflective thinking are important.

Lastly, we study designers' action preferences based on how frequently a designer uses different types of design actions (i.e., the distribution of design actions) during a design process. In total, five different design behaviors are adopted from three dimensions – mental iteration, reflective thinking, and design action preferences. We envision that modeling the design behaviors from multiple dimensions can help better understand design thinking.

3.2 Research approach

The overall approach (see Figure 2) starts with collecting the raw design action data from different sources, such as CAD loggers, design documents, etc. This raw design action data contains design actions, design-related artifacts, and the values of various design parameters. After collecting the design action data, to computationally model these five design behaviors, we adopted five different techniques. We use design action distribution to study design action preferences, the Markov chain model to study the one-step sequential behavior, the Doc2Vec to model contextual behavior, the bi-directional LSTM autoencoder to study the long-term sequential behavior, and the time-gap distribution to analyze reflective thinking. To explain the overall process, suppose a designer's sequence of design actions $[a_1, a_2, a_3, \dots, a_N]$ which has a timestamp associated with it $[t_1, t_2, t_3, \dots, t_N]$.

Before analyzing the design action preference and the one-step sequential behavior, we apply an ontological design process model (e.g., the FBS model), which consists of several design stages to characterize the design process. By applying the design process model, we will obtain a sequence of design process stages $[p_1, p_2, p_3, \dots, p_N]$. With this operation, we can reduce the dimensionality of the design action data. This treatment is similar to an embedding (latent space representation), which can help interpret designers' thought processes. To elicit designers' action preferences, we count the total number of each design process stage that certain actions fall into and plot the resulting distribution for every designer. To understand designers' one-step sequential behavior, we apply the first-order Markov chain to every designer's design process stage sequence and compute the transition probability matrix. This transition probability matrix can be vectorized, which quantifies the features of the one-step sequential behavior. For example, given a design process model defining N design process stages, we can get an $N \times 1$ vector from action preference, and an $N \times N$ transition probability matrix from the Markov chain model for one designer. The transition probability matrix can be converted into an $N^2 \times 1$ vector. For n designers, two matrices in the dimension of $N \times n$ and $N^2 \times n$ can be formed, representing the aggregated action preference and the one-step sequential behavior, respectively.

To understand designers' contextual behavior and long-term sequential behaviors, we apply the Doc2Vec [38] and the bi-directional LSTM auto-encoder [39] on the design action sequence, respectively. Both Doc2Vec and bi-directional LSTM attempt to predict the next design action from the input sequence. Doc2Vec supports this process by training paragraph vectors as auxiliary information. We will get an embedding matrix from each of these methods. As the embedding matrix is already a representation of the relationship among design actions, the data transformation from design action to design process stage using an ontological design process model is not needed in these two methods. It is mention-worthy that the size of the embedding matrix is user-defined. For example, with the embedding size of M, and for n designers' sequences, an $M \times n$ dimensional matrix from each of the methods can be obtained.

To understand the designers' reflective thinking, we utilize the time-gap distribution analysis. Particularly, we consider the time gap between each design action performed by a designer. For example, for an action sequence, the time gaps are $[0, \{t_2 - t_1\}, \{t_3 - t_2\} \dots\dots \{t_n - t_{n-1}\}]$. The distribution of this time gap essentially carries the reflective behavior. From each of the designers' time gap distributions, we can get several features, such as the distribution type and its parameters. For a particular designer, we use these features to create a vector, $P =$

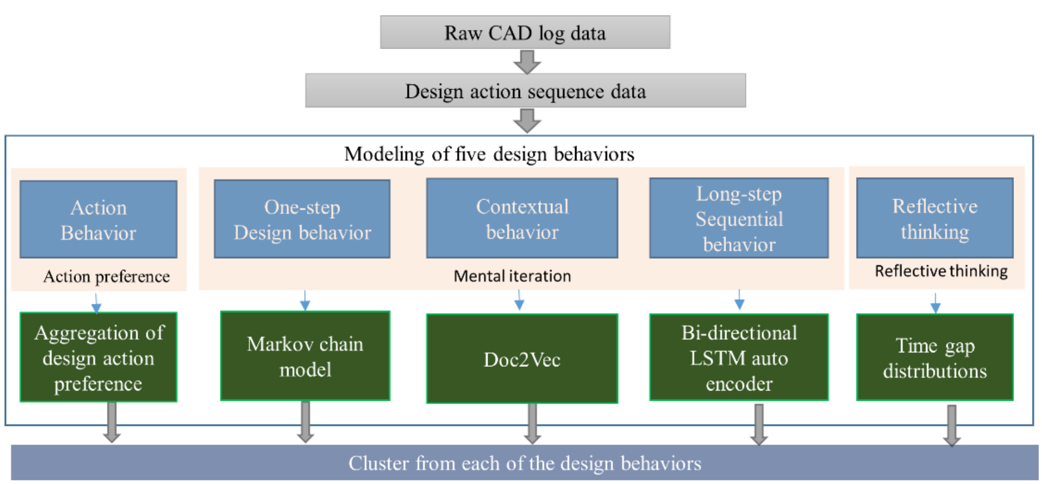

FIGURE 2: THE RESEARCH APPROACH FOR STUDYING DESIGN THINKING BASED ON FIVE DESIGN BEHAVIORS

$[Dist\ name, D_1, D_2, \ldots, D_n]$, where $Dist\ name$ indicates the distribution type (a categorical variable) and $D_1, D_2, \ldots, D_n$ are the distribution parameters. It is noted that the parameter number can be varied based on the type of the distribution. Assuming there are L parameters, for n designers, we obtain an $L \times n$ matrix. This matrix will be the feature representation of designers' reflective design thinking behaviors.

Based on these five models, we can obtain five behavioral matrices (i.e., the design embeddings) representing the five corresponding design thinking behaviors. Then, we implement a clustering method, i.e., X-mean cluster [40], on each behavioral matrix to group the designers who have similar design behavioral patterns in different behavioral dimensions. Figure 2 depicts a schematic diagram of the research approaches.

3.3 Doc2Vec

Doc2Vec uses a neural network approach to create a fixed-length vector representation of variable length sequences, such as sentences, paragraphs. In this study, since a design action sequence is a sequence of text data, it can be treated as a "sentence." Doc2Vec is based on Word2Vec, where it attempts to predict an element in a sequence from its surrounding or context element [38]. Given a sequence $w_1, w_2, w_3, \ldots, w_T$, to predict the context element w_t, the objective of the Word2vec is to maximize the average log probability.

$$\frac{1}{T}\sum_{t=k}^{T-k} log\ p(w_t | w_{t-k}, \ldots, w_{t+k}) \tag{1}$$

The prediction task is typically done by a neural network architecture with a multiclass classifier such as softmax [41]. This process can be expressed as follows:

$$p(w_t | w_{t-k}, \ldots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}} \tag{2}$$

$$\boldsymbol{y} = \boldsymbol{b} + \boldsymbol{U}\boldsymbol{h}(w_{t-k}, \ldots \ldots, w_{t+k}; \boldsymbol{W}) \tag{3}$$

, where Equation (2) outputs the predicated probability using the softmax function. y_i is the log probability for each output element i. Equation (3) represents the equation of feed-forward neural network where $\boldsymbol{U}, \boldsymbol{b}$ are the parameters of neural networks. $\boldsymbol{h}$ is constructed by a concatenation of vectors extracted from **W**.

In Doc2Vec, every sequence is associated with a unique vector, which is represented by a matrix **D** (for all sequences, it creates a matrix). Every element of the sequence is also mapped to a unique vector which is represented as **W** in Figure 3. The matrix **D** and **W** are concatenated and used in Equation (3) in place of $\boldsymbol{h}$.

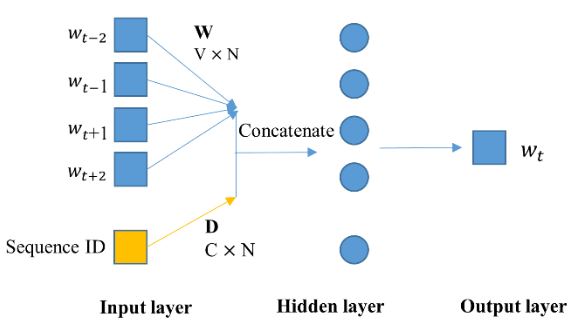

FIGURE 3: Doc2Vec

3.4 Bi-directional LSTM auto-encoder

The aim of using an auto-encoder (AE) is to learn a compressed, distributed representation of a data set. It is a neural network model that captures the most salient features of the input data [42]. The basic AE consists of only one hidden layer, and the target value is set equal to the input value. The training of the AE is done in two phases: encoding and decoding. In the encoding phase, input data are mapped into the hidden layer, and in the decoding process, the input data are reconstructed from the hidden layer representation. Given an input dataset $\boldsymbol{X} = \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \ldots, \boldsymbol{x_n}$, the two phases can be expressed as follows:

$$\boldsymbol{h}(\boldsymbol{x}) = f(\boldsymbol{W_1}\boldsymbol{X} + \boldsymbol{b_1}) \tag{4}$$
$$\widehat{\boldsymbol{X}} = g(\boldsymbol{W_2}\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{b_2}) \tag{5}$$

, where, $\boldsymbol{h}(\boldsymbol{x})$ represents the hidden representations of the input vector $\boldsymbol{X}$, and $\widehat{\boldsymbol{X}}$ is the decoder vector of the output layer. f is the encoding function, while g is the decoding function. $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ are the weight matrix of the encoder and decoder, respectively. $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are the bias vector in each phase, respectively. A schematic diagram of the auto-encoder is shown in figure 4 (b). LSTM is an upgraded variation of the recurrent neural network (RNN) [43], which is basically a recursive neural network used for sequential data. LSTM uses a gating mechanism that solves several flaws of the RNN (i.e., vanishing gradient problem, long-term dependency, etc.). A detail of the LSTM network is described in our previous work [13]. In this study, we leverage bidirectional LSTM in the auto-encoder architecture. Compared to the basic LSTM model, bidirectional LSTM consists of two groups of hidden layers. One layer for input sequence in the forward direction and the other layer for input sequence in the backward direction (see figure 4(a)). These two hidden layers do not interact with each other, and their output is concatenated to the final output layer. The mathematical equations for the

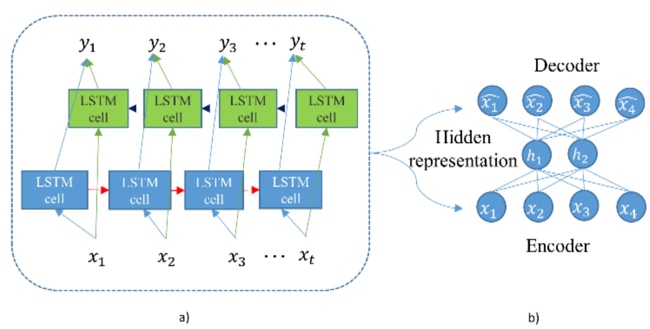

FIGURE 4 a): BI-DIRECTIONAL LSTM b) BASIC AUTO-ENCODER

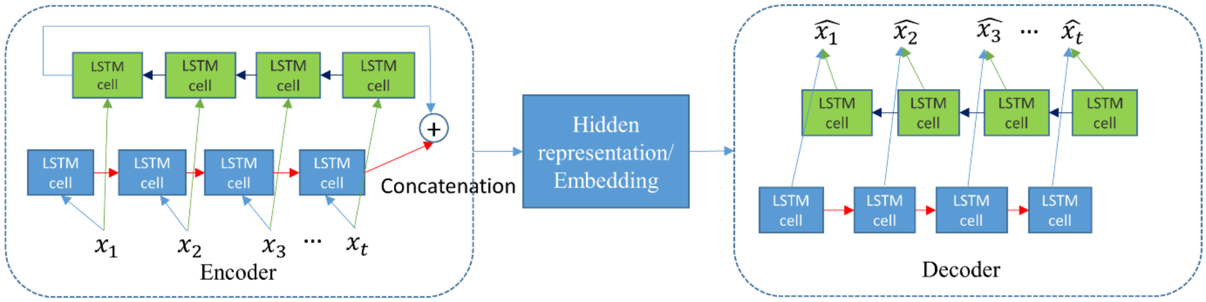

FIGURE 5: BI-DIRECTIONAL LSTM AUTO-ENCODER

bidirectional LSTM are the same as a basic LSTM, except that there are two hidden states at t^{th} time steps: $\vec{\boldsymbol{h}}$ (forward process) and $\overleftarrow{\boldsymbol{h}}$ (backward process). These two hidden states are concatenated for the final output

In the AE architecture, bi-directional LSTM is replaced with the feed-forward neural network. A schematic diagram of the bi-directional LSTM autoencoder is shown in Figure 5.

4. CLUSTERING DESIGN BEHAVIORS IN SOLAR ENERGY SYSTEM DESIGN - A CASE STUDY

This section provides an introduction to the design problem used in the case study and the data collection method.

4.1 Design procedure

The study was implemented in a suburban high school in the north-eastern US. The participants are 113 students from seven 9th grade classes of a course on the science of energy. These students barely had design experience before the project. During the six-day project, students worked on a design challenge with an open-source CAD software called Energy3D [44] individually and sought help from teachers if needed. Specifically, the project started with a day of Energy3D tutorial and followed by three days of conceptual learning, in which students interacted with simulations to understand five solar concepts and how these concepts affect solar-energy acceptance. Then students try to solve an authentic design challenge for two days to apply knowledge to practice and develop design skills.

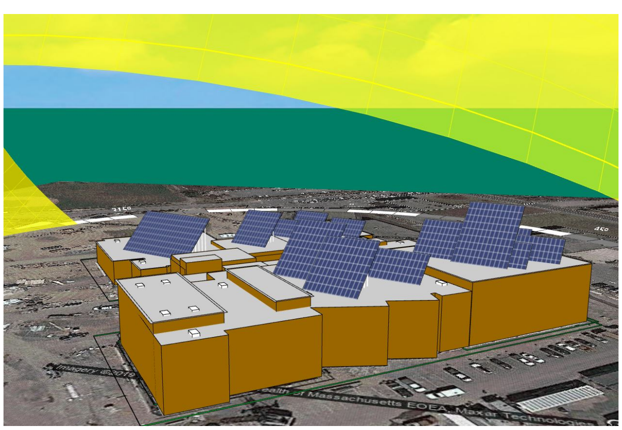

FIGURE 6: AN EXAMPLE OF THE SOLARIZE YOUR SCHOOL DESIGN

4.2 Design problem

The five solar concepts are the Sun's path, the projection effect, the effect of the air mass, the effect of weather, and solar radiation pathways. These concepts are tightly related to the design challenge and were selected by domain experts. Individual simulations and exercises were provided to students to learn each concept. The design task was customized to the students with their school as the context. The challenge was named Solarize Your School and set as asking for bids to power their school with green energy. Mainly, a 3D model of their school was provided. Students could install solar panels on the school building roof to generate no less than 400,000 kWh of electricity per year while the payback period was less than ten years. We provide three different solar panel models from which designers can choose any one of them for the design. This design challenge required students to balance several factors such as panel costs, solar panel orientation, tile angle, and avoiding shadows while aiming for the goal. Figure 6 shows an example of the Solarize Your School design.

4.3 Data collection and data processing

Energy3D collects the continuous flow of design logs, including design actions, time steps, design parameters, and simulation results. An example of a line of design action log is shown below.

{"Timestamp": "2019-10-22 08:34:26", "Project": "Stoughton High School", "File": "stoughton-high-school-ma.ng3", "Change Tilt Angle for All Racks": {"New Value": -1.0}}

Although initially, we collect 113 designers' data, after analysing their design, we realize that several students did not follow the design requirements (e.g., failed to choose one of the provided solar panels). For a fair comparison, we only consider the designs that met the design constraints, and this leads to 39 valid designs.

In this study, we only collect the design action data, such as the "change Tilt Angle for All Racks" in the above example. By extracting the design action from every row of the log file, a design action sequence can be generated. It's worth noting that we ignore the camera-related action such as "zoom in," "zoom out," and "camera" because it does not affect the design performance per se. After removing the irrelevant design actions,

TABLE 1: THE CODING SCHEME BASED ON THE FBS DESIGN PROCESS MODEL

Design process	Design action
Formulation	Add any component
Analysis	Analysis of annual net energy
Synthesis	Edit any component
Evaluation	Cost analysis
Reformulation 1	Remove structure
Reformulation 2	Remove solar device
Reformulation 3	Remove other components

60 unique design actions are identified. Then, for action behavior and one-step sequential behavior, we develop a coding scheme based on the FBS model to transcribe the design action data into a sequence of design processes. The coding scheme shown in Table 1 is used to categorize each design actions into one of the seven design process stages, including Formulation (F), Analysis (A), Synthesis (S), Evaluation (E), Reformulation 1 (R1), Reformulation 2 (R2) and Reformulation 3 (R3). The detail of the transformation process is described in our prior work [12].

5. RESULT AND DISCUSSION

5.1 Result

In this section, we present the result obtained from different design behaviors, particularly action preference, one-step sequential behavior, contextual behavior, long-term sequential behavior, and reflective thinking. The behaviors are represented as embedding and clustered using the X-mean clustering method. To compare the final design performance from the designers in each cluster, we developed a metric to quantify a student's final design quality (DQ). This metric is as follows:

$$DQ = \frac{P_R \times B \times E_0}{P_o \times C \times E_R}$$

where,
P_R = required payback period
B = budget
E_0 = Obtained energy output
P_0 = Obtained payback period

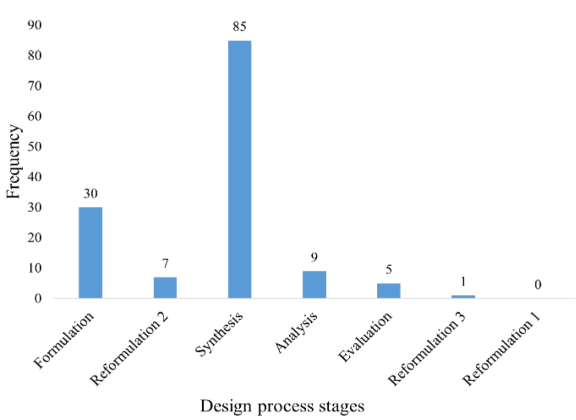

FIGURE 7: ACTION PREFERENCES OF DESIGNER P4L25

C = Cost
E_R = required energy output

A student's action preference is represented by the distribution of the design process stages that the student was in during the entire design process. By following the coding scheme in Table 1, we get a 7×1 vector for each designer; and for 39 designers, we get a 7×39 action behavior matrix. Figure 7 shows one designer's action preference distribution. By applying X-mean clustering on the action behavior matrix, three clusters are found. Cluster 3 includes ten designers who achieve the highest mean DQ of 1.325 with a standard deviation of 0.40, while Cluster 1 achieves the lowest DQ with 1.208 (standard deviation 0.408). Cluster 2 contains 13 designers with a mean DQ of 1.25 (standard deviation 0.64). Analysis of variance (ANOVA) indicates the difference between the cluster's DQ is not significant (p-value is 0.708).

We quantify the one-step sequential behavior using the first-order Markov chain model. Particularly, the transition probability matrix obtained from the first-order Markov model is characterized as the one-step sequential behavior. Like the previous method, before applying the model, the FBS design process model transforms the design actions into the sequence

TABLE 2: CLUSTER OF ONE-STEP SEQUENTIAL BEHAVIOR

	Cluster 1	Cluster 2
0	P1L10	P1L12
1	P1L14	P1L13
2	P1L17	P1L20
3	P1L18	P1L3
4	P2L10	P1L5
5	P2L12	P2L11
6	P2L13	P2L2
7	P2L14	P4L1
8	P2L16	P4L10
9	P2L17	P4L25
10	P2L7	P4L28
11	P3L3	P4L32
12	P4L11	P4L5
13	P4L26	P6L12
14	P4L27	P6L17
15	P4L9	P6L18
16	P6L1	P6L3
17	P6L14	
18	P6L15	
19	P6L19	
20	P6L4	
21	P6L6	
Mean of design quality	1.26	1.25
STD of design performance	0.278	0.648

design process stages. We obtain a 7×7 transition probability matrix for seven design process stages and then flatten the matrix to get a 49×1 vector. After obtaining 39 designers' transition probability matrices, they are converted to a 39×49 matrix that captures the one-step design behavior, from which the X-mean clustering is applied. By clustering one-step sequential behavior, we identify two clusters. In this method, the DQ obtained from both clusters is similar. Cluster 1 contains 22 designers with a mean DQ of 1.26 (standard deviation 0.278), while Cluster 2 achieves a mean DQ of 1.25 with a standard deviation of 0.648. The t-test indicates no significant differences between the DQs of the two clusters (p-value 0.27). Table 2 shows the results of one-step sequential behavior clustering. Here, the designers are indicated with the class number and the laptop number. For example, "P1L10" means that the designer is from Class 1 and used laptop number 10.

Using Doc2Vec, we obtain design embedding that represents the designers' contextual behavior or short-term behavior. Several hyper` parameters need to be tuned and selected for the Doc2Vec model. For example, in this study, we choose the embedding size for Doc2Vec as 100. Additionally, we choose the context window size as 5. With these settings, for 39 designers, we obtain a 39×100 embedding matrix. We apply the X-mean clustering method on the obtained embedding matrix and get two clusters. The first cluster contains 30 designers with a mean DQ of 1.22 and a standard deviation of 0.483. The second cluster contains nine students with a mean DQ of 1.34 and a standard deviation of 0.432. However, the t-test again indicates the difference between the DQs of the two clusters is not statistically significant (p-value 0.27).

TABLE 3: CLUSTER OF LONG-TERM SEQUENTIAL BEHAVIOR

	Cluster 1	Cluster 2	Cluster 3
0	P1L10	P1L12	P1L3
1	P1L13	P2L14	P6L1
2	P1L14	P2L16	P6L18
3	P1L17	P2L2	
4	P1L18	P2L7	
5	P1L20	P3L3	
6	P1L5	P4L28	
7	P2L10	P4L32	
8	P2L11	P4L9	
9	P2L12	P6L12	
10	P2L13	P6L6	
11	P2L17		
12	P4L1		
13	P4L10		
14	P4L11		
15	P4L25		
16	P4L26		
17	P4L27		
18	P4L5		
19	P6L14		
20	P6L15		
21	P6L17		
22	P6L19		
23	P6L3		
24	P6L4		
Mean of design quality	1.20	1.34	1.35
STD of design performance	0.356	0.637	0.588

We obtain design embedding for the long-term sequential behavior by utilizing the bi-directional LSTM autoencoder. In this architecture, in both the encoder and decoder parts, we use a bi-directional LSTM layer with a size of 128. Therefore, the embedding size from the LSTM autoencoder is 256, and with all the designers, we obtain a 39×256 matrix. By clustering the embedding matrix, we get three clusters. Table 3 shows the clustering results of the long-term sequential behavior. Cluster 1 contains 24 students with a mean DQ of 1.20 (standard deviation 0.356), while Cluster 3 has only three designers with a mean DQ of 1.35 (standard deviation 0.588). Cluster 2 contains 12 designers with a mean DQ of 1.34 (standard deviation 0.637). According to the ANOVA test, the difference among the clusters is not significant (p-value 0.7).

Finally, to obtain the embedding from reflective thinking, we get the parameters of the designers' time gap distribution. We only consider the time gap between 0s to 300s. The time gap between two actions taken exceeding 300s indicates the student likely stopped the design process. So, we omit the time gaps of more than 300s. In order to understand what distribution fits these time gap distributions, we use Kolmogorov–Smirnov test [45], where different distributions, including Normal, Exponential, Gamma, Generalized extreme value (GEV) distribution, and Weibull distribution, are compared against. The test indicates that GEV distribution has the best fit for majority of the designers' time gaps. Figure 8 shows designer P1L10's empirical time gap distribution and the fitted GEV distribution. From the distribution, we identify three parameters, including shape, location, and scale. With these three parameters from 39 designers, we obtain a 3×39 embedding matrix. This matrix represents the designers' reflective thinking. After applying the X-mean clustering method, we obtain four clusters. Figure 9 shows the four clustering results. The results of the clustering,

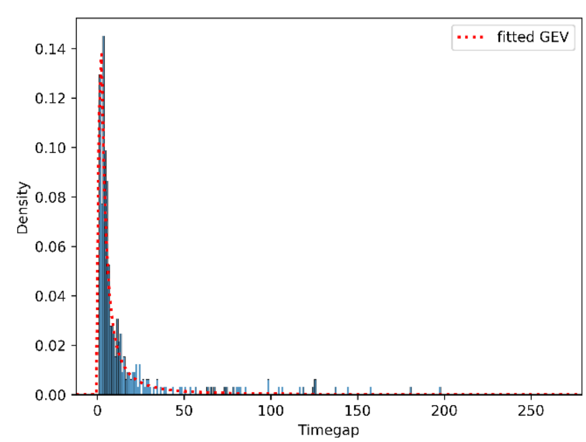

FIGURE 8: TIME GAP DISTRIBUTION OF DESIGNER P1L10

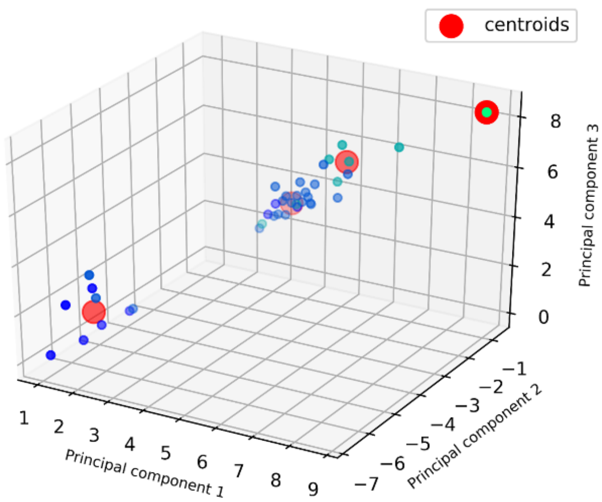

FIGURE 9: CLUSTER OBTAINED FROM EFLECTIVE THINKING BEHAVIOR

shown in Table 5, indicate that Cluster 1 contains nine designers with a mean DQ of 1.199 and a standard deviation of 0.32. Cluster 2 contains 22 designers with a mean DQ of 1.31 and a standard deviation of 0.55, while Cluster 3 contains seven designers with a mean DQ of 1.14 and a standard deviation of 0.39. Cluster 4 has only one designer with a DQ of 1.10. The Anova test indicates that the difference in the DQs among the clusters is not significant (p-value is 0.83).

5.2 Discussion

This study aims to understand design thinking behaviors from different behavioral dimensions by characterizing them through design embedding. After obtaining the embedding, we apply the X-mean clustering method to each of the embedding matrices to cluster designers. The clustering results indicate that the designers are clustered not according to their final design quality but instead based on their behavioral patterns. Different design patterns in a design process can lead to similar quality of the final design. For example, in the clustering based on designers' action behavior embedding, the designers of Cluster 3 use a high number of *Synthesis* on average compared to the designers in other clusters. Cluster 3 uses on average 500 Synthesis, while Cluster 1 and Cluster 2 use on average 150 and 233 *Synthesis,* respectively. This indicates that designers of Cluster 3 are involved in editing design components more frequently than the other designers during the design process. Additionally, we observe a higher number of usage of *Formulation* among the designers in Cluster 3 than those in the other clusters. The average number of the *Formulation* used by Cluster 3 is 62, while the average frequencies in Cluster 1 and Cluster 2 are 35 and 40, respectively. Figure 10 shows the design process stage preference of Cluster 3.

For the clustering based on designers' reflective thinking behavior, designers in each cluster also follow specific design thinking patterns. For example, the designers of Cluster 1 often wonder 1s-3s in between every two actions. This behavior may

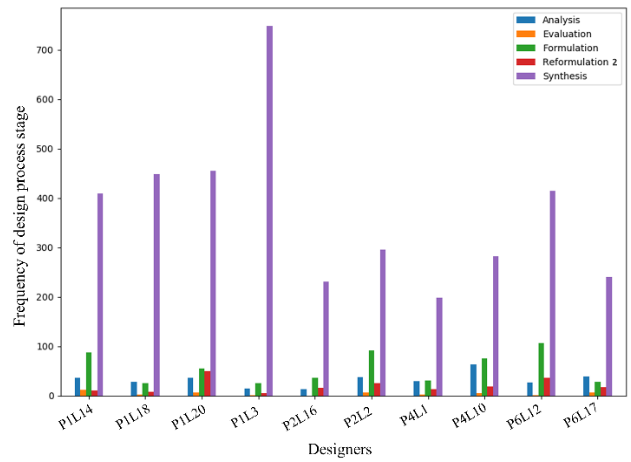

FIGURE 10: PREFERENCE OF DESIGN PROCESS STAGES OF CLUSTER 3

indicate that the designers in this cluster prefer trial-and-error, thus quickly clicking different design action buttons in the CAD software to explore the design space. In Cluster 2 and Cluster 3, designers follow a similar distribution of time gaps. However, unlike Cluster 1, designers of Clusters 2 and 3 has a relatively lower number of 1-3s time gaps. Rather in these clusters, 4-10s time gaps are prominent. This indicates that the designers in these clusters tend to ponder a little bit before taking the next design action. There is only one designer in Cluster 4. In different from the other students, this student has a uniform distribution of the time gaps during the entire design process.

The clustering of the design embedding obtained from the one-step sequential behavior indicates that designers follow several design patterns. For example, we observed that designers in two clusters use *Synthesis→Synthesis* and *Formulation→Synthesis* very frequently. *Synthesis→Synthesis* action pair indicates that designers sequentially edit the parameters of design components. For example, after changing the solar panel's tilt angle, designers continue changing the azimuth of it. *Formulation→Synthesis* action pair indicates that after adding a component, a designer starts to edit its parameters. For example, after adding a solar panel, a designer starts changing the solar panels' base height. There are some design patterns that are distinct from each cluster. For example, the designers in Cluster 2 use *Evaluation→ Analysis* design patterns a lot during their design processes, while this pattern is used very rarely among the designers in Cluster 1. This pattern indicates that after doing cost evaluation (compare the current cost with the given budget), the designer then analyzes the system's energy output. Figure 11 shows a heat map of the transition probability of the design patterns found by the designers of Cluster 2. The bright square indicates a high transition probability of the corresponding design patterns, where the dark square indicates no or very low transition probability.

6. CONCLUSION

In this study, we develop an approach to represent design thinking by characterizing design behaviors from multiple

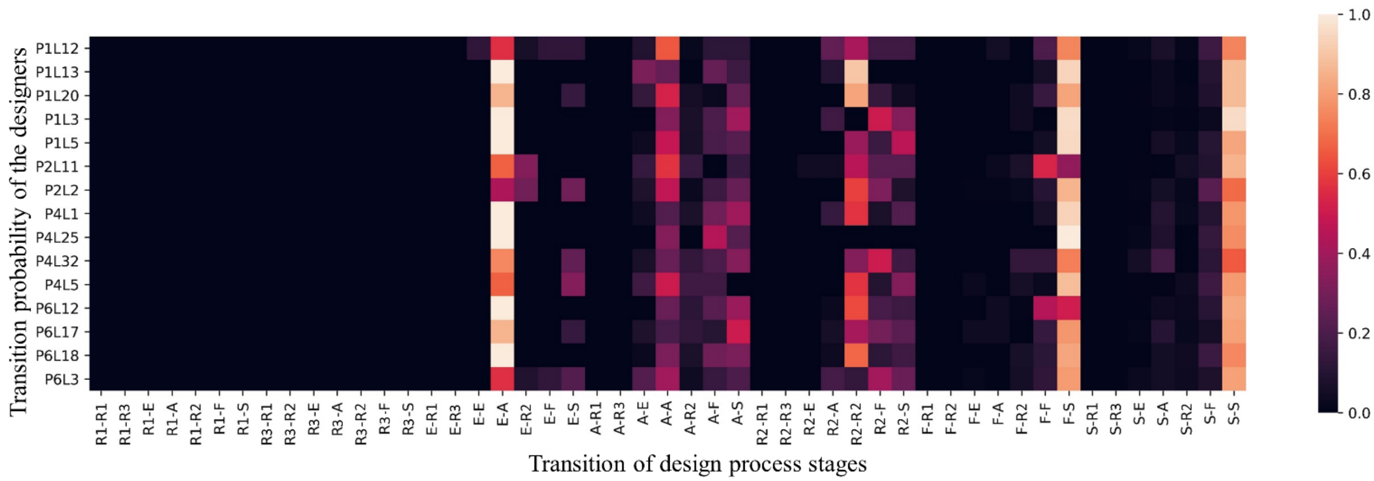

FIGURE 11: HEAT MAP OF THE TRANSITION PROBABILITY OF THE DESIGN PATTERNS OF CLUSTER 2

dimensions. We identified five different design behaviors, including design action preference, one-step sequential behavior, contextual behavior, long-term sequential behavior, and reflective design thinking. The design behaviors are characterized by different machine learning and statistical methods, and the design thinking is represented through a latent representation referred to as design embedding. We use the distribution of design actions to characterize designers' action preferences. The First-order Markov model is utilized for characterizing designers' one-step sequential behavior. To model designers' short-term sequential behaviors, the Doc2Vec sequence learning technique is adopted, while a bi-directional LSTM autoencoder is used to characterize the long-term sequential behavior. Finally, we use time gap distribution to represent reflective design thinking. After identifying the design embedding from each design behavior, the X-mean method is applied to cluster each embedding to identify similar behavioral patterns. The result indicates that the behavioral patterns characterized in different dimensions do not necessarily categorize designers in the same cluster. Also, while designers are clustered based on their design behavioral patterns, different design patterns could lead to similar design quality.

The major contribution of this paper is the identification of latent representation (i.e., design embedding) of design thinking through design behaviors from multiple dimensions. The implementation of design embedding can be useful in design research in different ways. For example, design embedding can be used to identify designers with similar behavioral patterns and discover beneficial design strategies. Furthermore, as a design process is typically a combination of different design behaviors, different forms of design embeddings can be integrated to develop predictive models that could yield better accuracy for design performance forecasting. This is one of the future studies we plan to work on. Particularly, we plan to develop a machine learning model based on the identified design embeddings and their correlations with the final design quality.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the financial support from NSF-CMMI-1842588 and NSF-DRL-1503196.

REFERENCES

[1] Crismond, D., and Adams, R., 2012, "The Informed Design Teaching and Learning Matrix," J. Eng. Educ., **101**, pp. 738–797.

[2] Dinar, M., Shah, J. J., Cagan, J., Leifer, L., Linsey, J., Smith, S. M., and Hernandez, N. V., 2015, "Empirical Studies of Designer Thinking: Past, Present, and Future," J. Mech. Des., **137**(2).

[3] Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., and Leifer, L. J., 2005, "Engineering Design Thinking, Teaching, and Learning," J. Eng. Educ., **94**(1), pp. 103–120.

[4] Rahman, M. H., Schimpf, C., Xie, C., and Sha, Z., 2019, "A Computer-Aided Design Based Research Platform for Design Thinking Studies," J. Mech. Des., **141**(12).

[5] Brockmann, E. N., and Anthony, W. P., 1998, "The Influence of Tacit Knowledge and Collective Mind on Strategic Planning," J. Manag. issues, pp. 204–222.

[6] Rahman, M. H., Gashler, M., Xie, C., and Sha, Z., 2018, "Automatic Clustering of Sequential Design Behaviors," *Proceedings of the ASME Design Engineering Technical Conference*.

[7] Yoav, G., and Graeme, H., 2017, "Neural Network Methods in Natural Language Processing," Morgan Claypool San Rafael, SR, USA, pp. 104–113.

[8] Gero, J. S., and Milovanovic, J., 2020, "A Framework for Studying Design Thinking through Measuring Designers' Minds, Bodies and Brains," Des. Sci., **6**, p. e19.

[9] Kan, J. W. T., Gero, J. S., and Press, C. R. C., 2009, "Using the FBS Ontology to Capture Semantic Design Information in Design Protocol Studies," *About: Designing. Analysing Design Meetings*, pp. 213–229.

[10] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Mining Process Heuristics from Designer Action Data via Hidden Markov Models," J. Mech. Des., **139**(11), p. 111412.
[11] Rahman, M. H., Xie, C., and Sha, Z., 2019, "A Deep Learning Based Approach to Predict Sequential Design Decisions," *Proceedings of the ASME Design Engineering Technical Conference*.
[12] Rahman, M. H., Xie, C., and Sha, Z., 2021, "Predicting Sequential Design Decisions Using the Function-Behavior-Structure Design Process Model and Recurrent Neural Networks," J. Mech. Des., pp. 1–46.
[13] Rahman, M. H., Yuan, S., Xie, C., and Sha, Z., 2020, "Predicting Human Design Decisions with Deep Recurrent Neural Network Combining Static and Dynamic Data," Des. Sci., **6**, p. e15.
[14] Suwa, M., and Tversky, B., 1997, "What Do Architects and Students Perceive in Their Design Sketches? A Protocol Analysis," Des. Stud., **18**(4), pp. 385–403.
[15] SHAH, J. J., VARGAS-HERNANDEZ, N. O. E., SUMMERS, J. D., and KULKARNI, S., 2001, "Collaborative Sketching (C-Sketch) — An Idea Generation Technique for Engineering Design," J. Creat. Behav., **35**(3), pp. 168–198.
[16] Self, J. A., 2019, "Communication through Design Sketches: Implications for Stakeholder Interpretation during Concept Design," Des. Stud., **63**, pp. 1–36.
[17] Sun, L., Xiang, W., Chai, C., Yang, Z., and Zhang, K., 2014, "Designers' Perception during Sketching: An Examination of Creative Segment Theory Using Eye Movements," Des. Stud., **35**(6), pp. 593–613.
[18] Matthiesen, S., Meboldt, M., Ruckpaul, A., and Mussgnug, M., 2013, "Eye Tracking, a Method for Engineering Design Research on Engineers' Behavior While Analyzing Technical Systems."
[19] Nguyen, T. A., and Zeng, Y., 2014, "A Physiological Study of Relationship between Designer's Mental Effort and Mental Stress during Conceptual Design," Comput. Des., **54**, pp. 3–18.
[20] Liu, L., Nguyen, T. A., Zeng, Y., and Hamza, A. Ben, 2016, "Identification of Relationships Between Electroencephalography (EEG) Bands and Design Activities."
[21] Goucher-Lambert, K., Moss, J., and Cagan, J., 2019, "A Neuroimaging Investigation of Design Ideation with and without Inspirational Stimuli—Understanding the Meaning of near and Far Stimuli," Des. Stud., **60**, pp. 1–38.
[22] Shealy, T., and Gero, J., 2019, "The Neurocognition of Three Engineering Concept Generation Techniques," Proc. Des. Soc. Int. Conf. Eng. Des., **1**(1), pp. 1833–1842.
[23] Vieira, S., Gero, J., Delmoral, J., Gattol, V., Fernandes, C., Parente, M., and Fernandes, A., 2019, "Understanding the Design Neurocognition of Mechanical Engineers When Designing and Problem-Solving."
[24] Pachidi, S., Spruit, M., and van de Weerd, I., 2014, "Understanding Users' Behavior with Software Operation Data Mining," Comput. Human Behav., **30**, pp. 583–594.
[25] Adams, R. S., and Atman, C. J., 1999, "Cognitive Processes in Iterative Design Behavior," *FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings (IEEE Cat. No.99CH37011*, pp. 11A6/13-11A6/18 vol.1.
[26] Bucciarelli, L. L., and Bucciarelli, L. L., 1994, *Designing Engineers*, MIT press.
[27] Yu, R. O. N. G. R. O. N. G., Gero, J. O. H. N. S., Ikeda, Y., Herr, C. M., Holzer, D., Kaijima, S., Kim, M. J., Schnabel, A., and Citeseer, 2015, "An Empirical Foundation for Design Patterns in Parametric Design," *20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Daegu, South Korea, May*, pp. 20–23.
[28] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Capturing Human Sequence-Learning Abilities in Configuration Design Tasks through Markov Chains," J. Mech. Des., **139**(9), p. 91101.
[29] Gero, J. S., Kan, J. W. T., and Pourmohamadi, M., 2011, "Analysing Design Protocols: Development of Methods and Tools."
[30] Gero, J. S., and Peng, W., 2009, "Understanding Behaviors of a Constructive Memory Agent: A Markov Chain Analysis," Knowledge-Based Syst., **22**(8), pp. 610–621.
[31] Raina, A., McComb, C., and Cagan, J., 2018, "Design Strategy Transfer in Cognitively-Inspired Agents," *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, p. V02AT03A018-V02AT03A018.
[32] Hong, Y.-C., and Choi, I., 2011, "Three Dimensions of Reflective Thinking in Solving Design Problems: A Conceptual Model," Educ. Technol. Res. Dev., **59**(5), pp. 687–710.
[33] Goldstein, M. H., Purzer, Ş., Adams, R. S., Chao, J., and Xie, C., 2019, "The Relationship between Design Reflectivity and Conceptions of Informed Design among High School Students," Eur. J. Eng. Educ., **44**(1–2), pp. 123–136.
[34] Jin, Y., and Chusilp, P., 2006, "Study of Mental Iteration in Different Design Situations," Des. Stud., **27**(1), pp. 25–55.
[35] Boud, D., Keogh, R., and Walker, D., 2013, *Reflection: Turning Experience into Learning*, Routledge.
[36] Valkenburg, R., and Dorst, K., 1998, "The Reflective Practice of Design Teams," Des. Stud., **19**(3), pp. 249–271.

[37] Prudhomme, G., Boujut, J. F., and Brissaud, D., 2003, "Toward Reflective Practice in Engineering Design Education," Int. J. Eng. Educ., **19**(2), pp. 328–337.

[38] Le, Q., and Mikolov, T., 2014, "Distributed Representations of Sentences and Documents," *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, JMLR.org, pp. II–1188–II–1196.

[39] Marchi, E., Vesperini, F., Eyben, F., Squartini, S., and Schuller, B., 2015, "A Novel Approach for Automatic Acoustic Novelty Detection Using a Denoising Autoencoder with Bidirectional LSTM Neural Networks," *Proceedings 40th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2015*, pp. 5-pages.

[40] Pelleg, D., and Moore, A. W., 2000, "X-Means: Extending k-Means with Efficient Estimation of the Number of Clusters.," *Icml*, pp. 727–734.

[41] Goodfellow, I., Bengio, Y., and Courville, A., 2016, *Deep Learning*, MIT press.

[42] Wang, Y., Yao, H., and Zhao, S., 2016, "Auto-Encoder Based Dimensionality Reduction," Neurocomputing, **184**, pp. 232–242.

[43] Hochreiter, S., and Schmidhuber, J., 1997, "Long Short-Term Memory," Neural Comput., **9**(8), pp. 1735–1780.

[44] Xie, C., Schimpf, C., Chao, J., Nourian, S., and Massicotte, J., 2018, "Learning and Teaching Engineering Design through Modeling and Simulation on a CAD Platform," Comput. Appl. Eng. Educ., **26**(4), pp. 824–840.

[45] 2008, "Kolmogorov–Smirnov Test BT - The Concise Encyclopedia of Statistics," Springer New York, New York, NY, pp. 283–287.