5-2018

An Investigation of Modeing Behaviors in Function Structure Modeling With Respect to Chaining Methods

Apurva Rajesh Patel
Clemson University, apurvap@g.clemson.edu

AN INVESTIGATION OF MODELING BEHAVIORS IN FUNCTION STRUCTURE MODELING WITH RESPECT TO CHAINING METHODS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science
Mechanical Engineering

---

by
Apurva Rajesh Patel
May 2018

Accepted by:
Joshua D. Summers, Committee Co-Chair
Marissa L. Shuffler, Co-Chair
Cameron J. Turner

ABSTRACT

The systematic engineering design process equips designers with tools and methods necessary to understand and solve a given design problem. Function decomposition is one such tool that allows designers to decompose the given problem into sub-problems which may be easier to address. Research on Function modeling, specifically Function Structure models, has focused on improving model construction techniques and using the Function Structure models to support concept generation. Additionally, Function Structure models have also been traditionally used as individual design tools; however, most other conceptual design tools are used in a collaborative setting (e.g. gallery sketching, method 3-6-5, etc.). This research investigates the use of Function Structure models as a collaborative tool by using seed models constructed using three different chaining methods (forward chaining, backward chaining, and nucleation) identified in a pilot protocol study. These seed models were intended to represent a partially completed model created by one designer, which was then delivered to the next designer for completion. A designer study and a protocol study were conducted to identify differences between the final Function Structure models generated using different seed models, based on the percent increase in the number of functions and flows, change in model complexity, and a rubric based evaluation of the model. Results show that using a nucleation seed model yield a higher increase in function and flows, as well as a larger change in model complexity. Analysis of the rubric based model evaluation shows that the presence of the seed model improves the evaluation scores, however, the type of chaining method used does not impact the final score. These results suggest that teaching of Function Structure models should include explicit identification of the different chaining methods, and recommends nucleation as the chaining method of choice. Moreover, future research areas are identified with respect to further comparison of chaining methods, as well as investigation of behavioral patterns in the modeling activity.

DEDICATION

This thesis is dedicated to my parents, Rajesh and Kalpana Patel, who have always been there for me, perhaps more often than I really wanted them to be. Their moral and financial support gave me the freedom to pursue intellectual endeavors. I would also like to dedicate the thesis to my brother, Trupal Patel, who has supported me in ways I can hardly expect anyone else to support me.

ACKNOWLEDGEMENT

I would like to thank my advisor Dr. Joshua D. Summers, and my committee members, Dr. Marissa L. Shuffler and Dr. Cameron J. Turner, for their support and guidance in completing this research. Their efforts to stimulate my curiosity, as well as to expand my boundaries have resulted in my improvement as a researcher, especially Dr. Summers. I would also like the CEDAR lab members, present and past, who have provided me with ample feedback on my work, joined me in research and publishing, and constantly pushed me to complete my thesis, while also being excellent distractions from work. CEDAR students are the ones who make the windowless labs a comforting and welcoming place to work. I would also like to acknowledge students from the psychology department, Billy and Michelle, who helped me with soul-crushing video coding as well as provided important insight on various topics. Finally, I would like to thank NSF for funding this research.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter One
FUNCTION MODELING: WHAT IS IT AND WHY STUDY IT?

The systematic engineering design process is an information exchange between the problem space and the solution space, guided by a series of decision making tasks which are supported by a variety of design tools [1–3]. These design tools can be used throughout the design process to help better understand the problem, and search of creative and efficient solutions. Ideation tools such as brainstorming, gallery method, Method 3-6-5, or collaborative sketching (C-Sketch) are used to help designers systematically generate solution concepts, whereas decision making tools such as Morph Charts, Pugh Matrices, decision matrices, or pairwise comparisons are used to help designers evaluate solutions against defined requirements [4–7].

1.1 Design Tools in Engineering

Engineering design tools can be classified into three board categories: tools that act in the problem space, the tools that act in the solution space, and tools that act in both the solution space and problem space simultaneously. The problem space is the part of engineering design that focuses on improving our understanding of the problem. The solution space, on the other hand, deals with generating solutions that address the identified problem. Figure 1 shows a non-exhaustive list of design tools, classified into the three categories. Tools such as requirements checklist, PDS, or objective tree work in the problem space, whereas, brainstorming, gallery method, and C-sketch support work in the solution space.

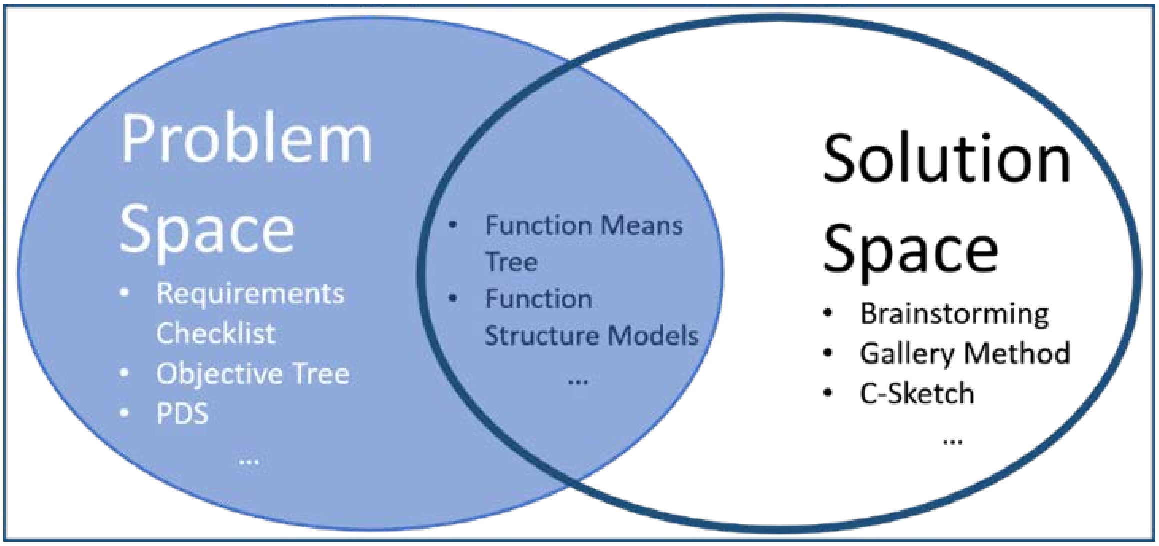


Figure 1: Classification of Engineering Design Tools

Certain tools such as the function means tree or function modeling, work simultaneously in both spaces. These function focused tools help the designer bridge the problem space to the solution space. Much of the research done to understand and enhance creativity with respect to the design tools has been focused on the design tools in the solution phase [8–10]. This research is focused on understanding function modeling, a design tool that works simultaneously in the problems space and the solution space.

1.2 Function Modeling as a Design Tool

As a design tool, function modeling works between the problem space and solution space, transferring information both ways, and improving the understanding of both. Function modeling can be used to objectively model a design problem by decomposing the problem and revealing its sub-functions [6]. Similarly, function models can also be used to model existing products for evaluating product similarity, identifying innovation opportunities, and other reverse engineering purposes [11–13]. Alternatively function models can also be used to generate solutions for a given problem [1,2,14]. In general, function models can be used for generative design purposes (problem definition, solution concepts) or reverse engineering purposes (product similarity, design evolution).

Function modeling is done through different representations, such as Function Structure model [2], Function Behavior Structure (FBS) model [12,15,16], Structure Behavior Function (SBF) model [17], Function Behavior State (FBSt) model [18,19], Contact and Channel model [20,21], and Function Interaction models [22]. These representations provide a more specialized approach to function modeling, ranging from developing computational tools for problem solving to identifying product similarities. This research will focus on the Function Structure model representation, a compound flowchart of anticipated system functions and flows [2]. Figure 2 shows an example Function Structure model describing a rice cooker.

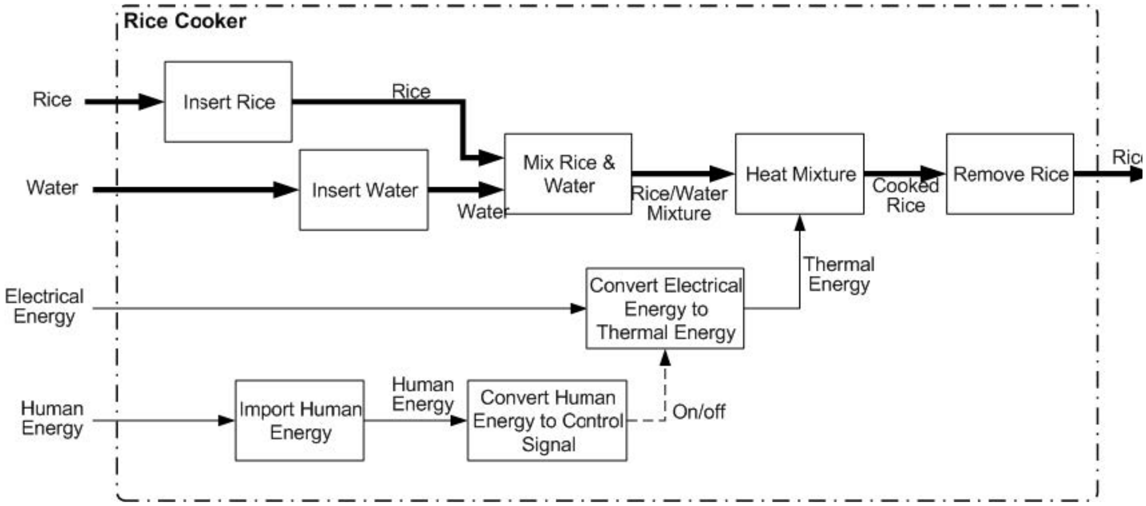


Figure 2: Function Structure Model of a Rice Cooker[1]

[1] From the design repository located at http://ftest.mime.oregonstate.edu/repo/browse/

As shown in Figure 2, a Function Structure model consists of function blocks and flow arrows. The dashed line is the system boundary; all the functionality of the system is contained within this boundary. Functions in the models are transformative actions on the energy, material, or signal flows passing through the system.

1.3 Research on Function Structure Models

The concept of function has always been a topic of research in engineering and design. Early research on function modeling can be found in field of computer science and artificial intelligence [23], whereas early description of function structure models in mechanical design can be found in the works of Pahl and Beitz [2]. More recent research efforts on function structure models have focused on strategies to construct the model, developing a standardized vocabulary, evaluation and interpretation of the model, using the model for concept generation, and using the model for comparison of existing products. However, limited research has been done targeting modeling behaviors related to function structure models. Understanding designers' cognitive activities while creating function structure models has been relatively unexplored.

Table 1: Summary of Function Modeling Research

No.	Citation	Year	Type of Study	Scope	Research Focus	Application Area
[24]	McAdams, Stone, and Wood 1999	1999	CS	S	PC	REV
[25]	Kurfman et al. 2000	2000	DS	L	MC	REV
[14]	Hirtz, Stone, and McAdams 2002	2002	LR	N/A	MC	REV
[26]	Bryant et al. 2005	2005	TS	S	CG	GEN
[27]	Sridharan and Campbell 2005	2005	DS	L	MC	GEN
[28]	Caldwell et al. 2008	2008	AS	M	MC	GEN/REV
[29]	Thomas et al. 2009	2009	DS	M	ME	GEN/REV
[30]	Schultz et al. 2010	2010	CS	S	MC/ME	REV
[22]	Ramachandran, Caldwell, and Mocko 2011	2011	DS	S	CG	GEN
[31]	Caldwell et al. 2012	2012	DS	M	ME	GEN/REV
[32]	Sen and Summers 2012	2012	PS	S	MB	GEN
[33]	Nagel, Bohm, and Linsey 2013	2013	DS	M	ME	GEN/REV
[34]	Tomiyama et al. 2013	2013	CS	S	IU	GEN/REV
[11]	Booth et al. 2015	2015	DS	M	MB	REV
[35]	Eisenbart et al. 2017	2017	CS	M	MC/ME	GEN
[36]	Gill, Summers, and Turner 2017	2017	AS	L	MC/ME	REV
[37]	Mokhtarian, Coatanéa, and Paris 2017	2017	CS	L	MC/ME	REV/GEN
[38]	Gericke and Eisenbart 2017	2017	CS	S	MC/ME	REV/GEN

Type of Study: CS – case study; PS – protocol study; DS – designer study; LR – literature review; TS – theoretical study, AS – analytical study

Research Focus: MC – model construction, ME – model evaluation and interpretation, CG – concept generation, PC – Product comparison, IU – Industry Use, MB – modeling behaviors

Scope: S – small, less than 12 functions; M – medium, 12 to 20 functions; L – large, more than 20 functions

Application Area: GEN – generative design, REV – reverse engineering

Table 1 presents a summary of recent research on function structure models, including the type of study, scope of research, research focus, and application area for each citation. The type of study categorizes the research methods used in any experiments conducted for the research, whereas the scope categorizes the Function Structure models involved in the research with respect to the number of functions. Moreover, the research focus categorizes the aspects of function modeling investigated. Finally, the application area states whether the research was geared towards generative design (solving new and novel problems) or reverse engineering.

As shown in Table 1, designer study (7 out of 18) and case study (6 out of 18) are widely used in research for function structures. Additionally, majority of the research on Function Structure models focuses on model construction (9 out of 18), and model evaluation and interpretation (8 out of 18). Some work has been done in the area of using Function Structure models for concept generation and product comparisons. Finally, work on modeling behaviors is relatively new, and mostly explorative.

1.4 Pilot Protocol Study

A protocol study was conducted previously to investigate modeling behaviors in function structure modeling [32], which examined two participants with varied backgrounds in design experience. The same design problem was given to both designers, and their activities were recorded, and the video was subsequently coded to analyze designer behavior. Findings from this protocol study suggested that patterns may exist in the construction of the model with respect to chaining methods. Three different chaining methods were identified: forward chaining, backward chaining, and nucleation.

1.4.1 Chaining methods in Function Structure Modeling

Chaining of a function structure model describes the directionality of the model as the designer constructs it. The example function model showed in Figure 2 can be used as a basis to describe the different chaining methods. In order to simplify the explanation of chaining methods, the text descriptions of the functions (block text) and flows (edge text) will be disregarded. A portion of the rice cooker Function Structure model was re-created and annotated, as shown in Figure 3.

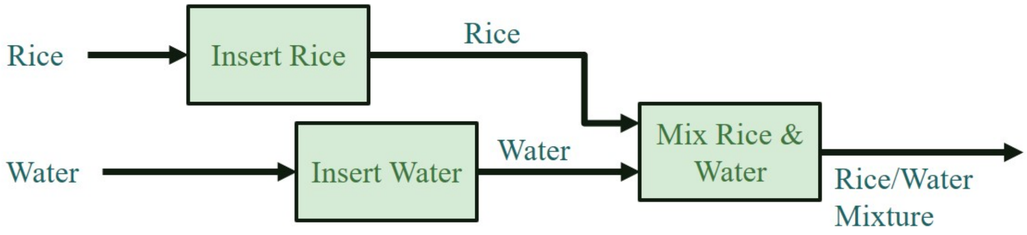


Figure 3 Partial Rice Cooker Function Structure

The partial function structure of the rice cooker has 8 elements. These elements can be added in a variety of sequences, which result in the three different chaining methods.

1.4.1.1 Forward Chaining

Forward chaining is a technique that generates the model from the known inputs to the desired outputs. For example, if the model in Figure 3 was generated using forward chaining, the elements would be from left to right, as shown in Figure 4.

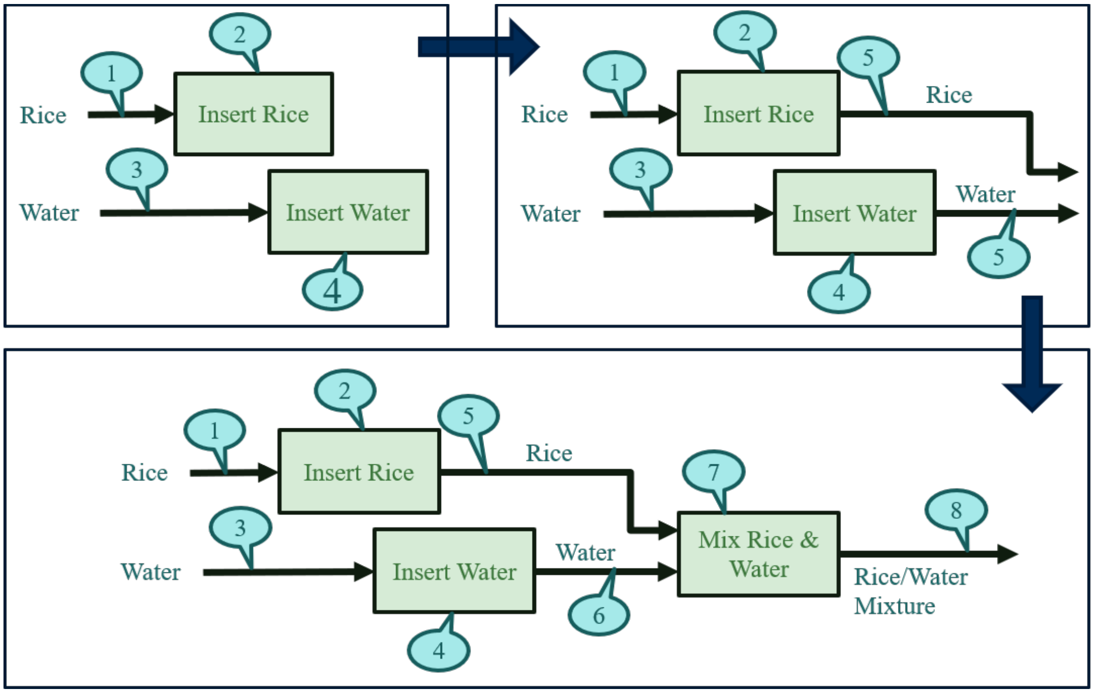


Figure 4: Partial rice cooker model created using forward chaining

As such, the source for a function is always added before the function itself, and the sink for the function always follows the function. A source is the element that serves as an input, whereas a sink is the element that acts as the output. This source-sink relationship can be graphically represented in a topology graph. An example of the topology graph for forward chaining is shown in Figure 5.

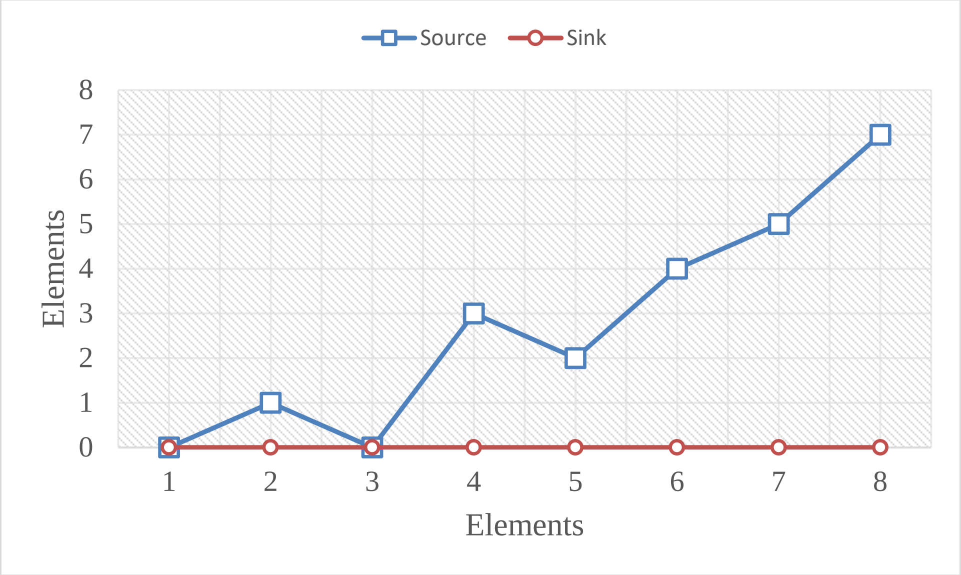


Figure 5: Graph representation of Forward Chaining

In Figure 5, the elements are numbered chronologically on both axes, and the source and sink identified for each element is graphed. For a forward chaining model, the line connecting the sources shows a positive slope, additionally, the source line is above the sink line, as shown in Figure 5.

1.4.1.2 Backward Chaining

Backward chaining is done in the reverse order compared to forward chaining, with the final output being added first and the initial input being added last. In backward chaining, the sink for the function is always added before the function, and the source for the function is always added after the function, as shown in Figure 6.

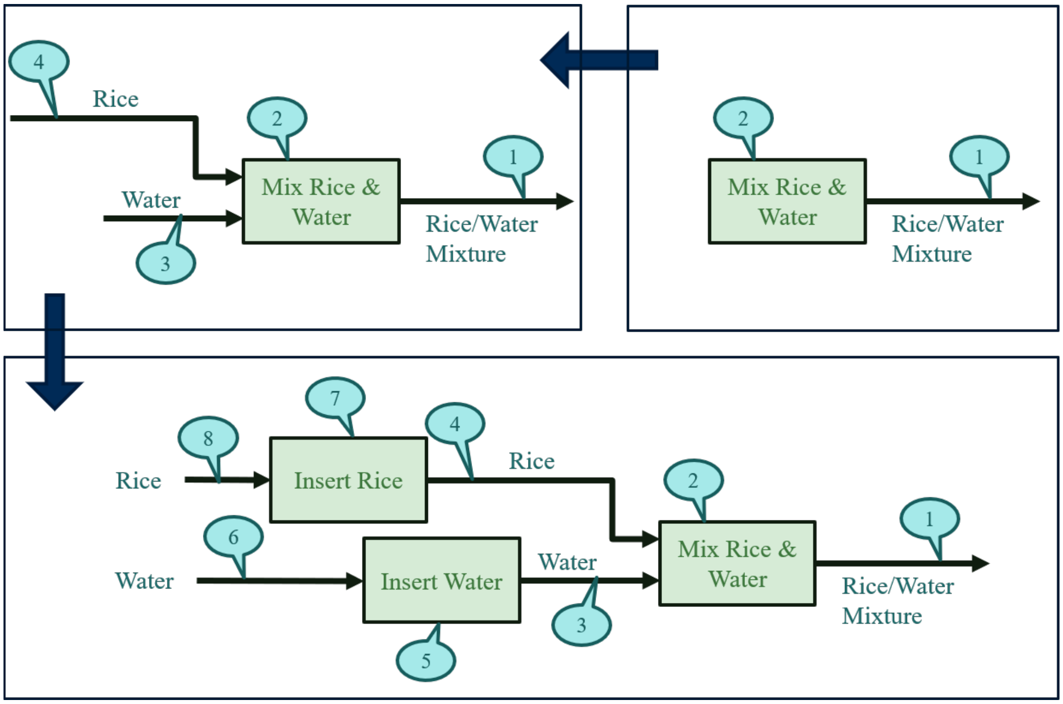


Figure 6: Partial rice cooker model using backward chaining

A general topology graph for back-ward chaining can be seen in Figure 7.

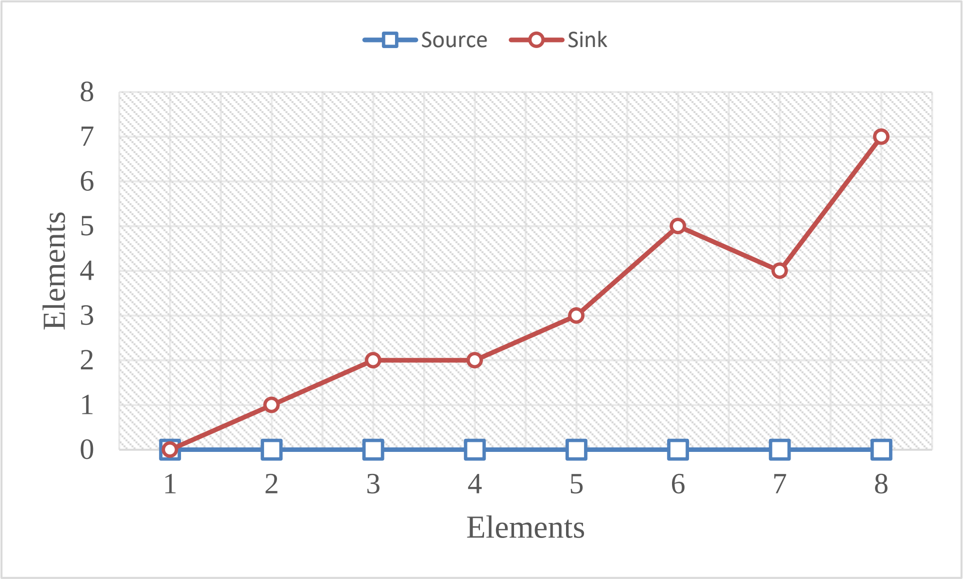


Figure 7: Graph Representation of Backward Chaining

As shown in Figure 7, the line connecting the sources is always zero, while the line connecting the sinks has a positive slope. This is indicative of backward chaining because it shows that the sinks to an element are added before the sources.

1.4.1.3 Nucleation

Nucleation of a model occurs when most central, or the element carrying the most information, is added to the model first and the rest of the model is nucleated from there, as shown in Figure 8.

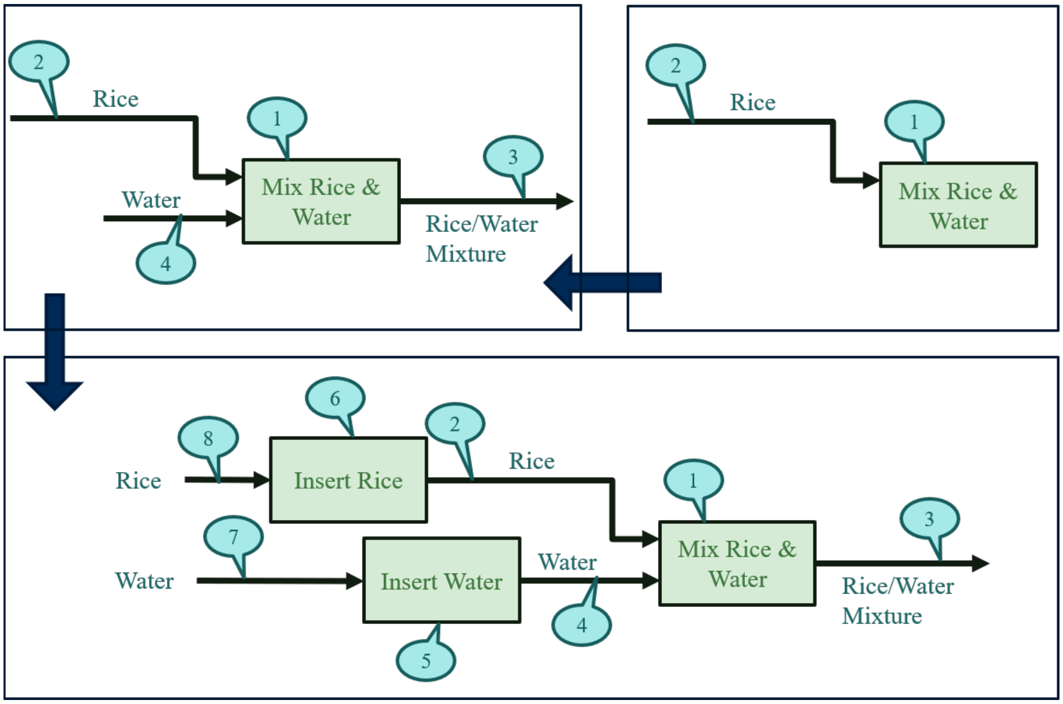


Figure 8: Rice cooker model created with nucleation

In other words, nucleation uses both forward and backward chaining as needed. A general topology graph for nucleation can be seen in Figure 9.

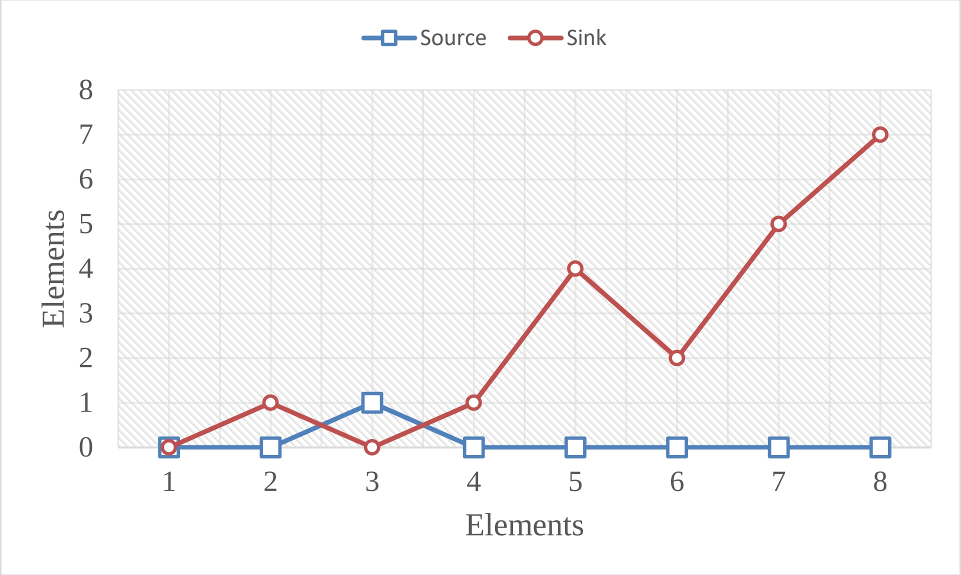
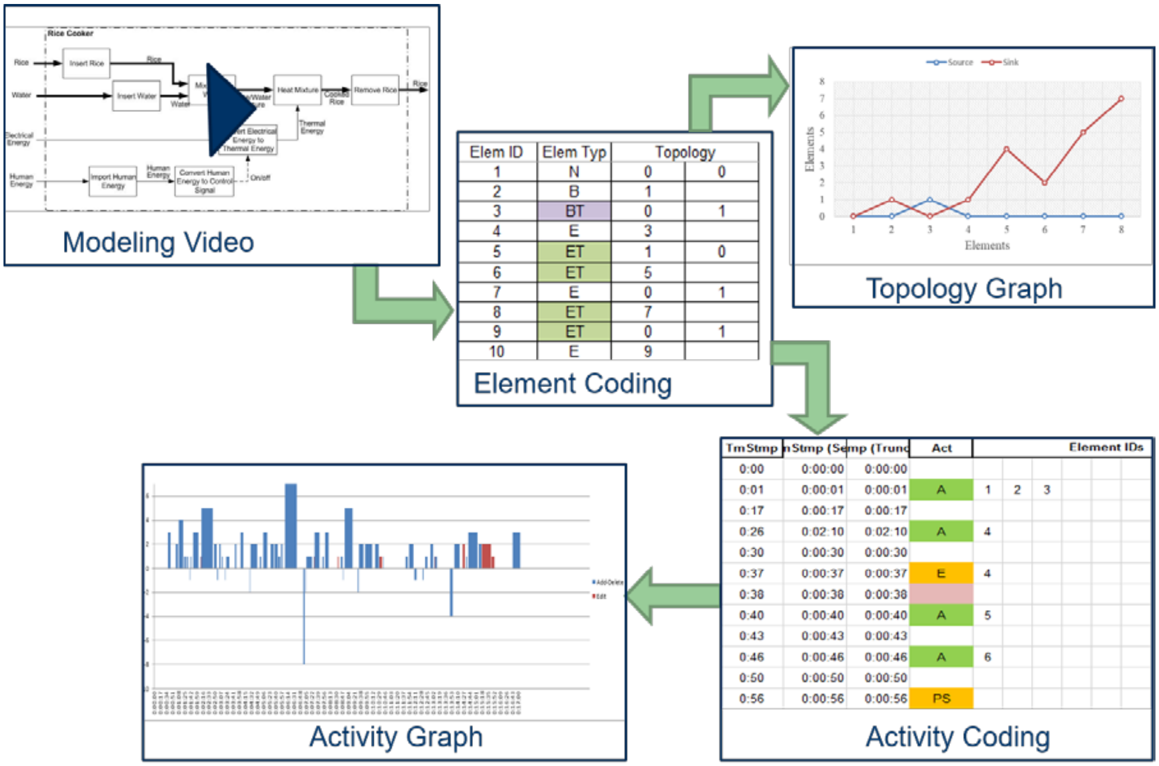


Figure 9: Graph representation of Nucleation

As shown in Figure 9, a topology graph for nucleation includes features from both forward chaining and backward chaining. Nucleation can be identified in a graph by observing instances of source and sink lines intersections. These are indications of change from forward chaining to backward chaining or vice versa.

1.4.2 Experimental Setup and Video Coding

In order to further understand chaining patterns in modeling behavior, a follow-up protocol study was conducted, where eight graduate student participants were given the task of creating a function structure model for a given design problem [39,40]. The videos collected for each participant were analyzed using a video coding procedure. The coding procedure consists of three types of coding: element coding, activity coding, and topology coding. Figure 10 shows an overview of the coding procedure.



Elem ID	Elem Typ	Topology	
1	N	0	0
2	B	1	
3	BT	0	1
4	E	3	
5	ET	1	0
6	ET	5	
7	E	0	1
8	ET	7	
9	ET	0	1
10	E	9	

Element Coding

Tm Stmp	n Stmp (Se	mp (Trunc	Act	Element IDs					
0:00	0:00:00	0:00:00							
0:01	0:00:01	0:00:01	A	1	2	3			
0:17	0:00:17	0:00:17							
0:26	0:02:10	0:02:10	A	4					
0:30	0:00:30	0:00:30							
0:37	0:00:37	0:00:37	E	4					
0:38	0:00:38	0:00:38							
0:40	0:00:40	0:00:40	A	5					
0:43	0:00:43	0:00:43							
0:46	0:00:46	0:00:46	A	6					
0:50	0:00:50	0:00:50							
0:56	0:00:56	0:00:56	PS						

Activity Coding

Figure 10: Overview of the Coding Procedure

As shown in the figure, the final results of coding a video are two graphs: an activity graph and a topology graph. As discussed earlier, the topology graph can be used to identify the chaining methods used in the modeling behavior. Alternatively, the activity graph is used to identify different types of activities on a time-scale. Observation and analysis of the graphs for all eight participants suggested that forward chaining was predominantly used by participants during the modeling activity (84.2% of total modeling time), with small amounts of nucleation (14%), and almost no backward chaining (1.8%) [39]. However, due to the small number of participants used in this study, a need was identified to conduct a different study to identify whether forward chaining was predominantly used in function modeling.

Chapter Two
MOTIVATION: WHY STUDY MODELING BEHAVIOR

As previously mentioned, research on investigating modeling behaviors in a function modeling activity is largely exploratory. Observation from the pilot study identify three chaining methods and finds that forward chaining was predominantly used; however, these findings are based on a small number of participants. Additionally, work on function modeling focuses largely on models generated by individual designers. Notably, none of the citations mentioned in Table 1 explicitly address group-based generation of Function Structure Models. However, other design tools used in the conceptual design stage such as brainstorming, gallery sketching, C-Sketch, and decision matrices are generally used in a group setting [41], resulting in an interdependence within the group using these tools.

2.1 Interdependence

Literature on functional modeling describes Function Structure models as a tool for problem decomposition, as well as a tool for facilitating ideation [39]. However, there is a lack of research focused on the cognitive processes within a designer when generating function models. Generally, when a designer reads a textual description of a product, two distinct representations are generated: a mental model, and a text-based reflection of the reviewed material [42]. Unlike the text-based reflection, a mental model is an object-based representation constructed in the working memory, consisting of the different components of the product [43]. The object based representation can take a variety of forms, including a mental map of the inputs and outputs of the product, a model of underlying mechanical reasoning, or the mental representation of the product working along with associated spatial manipulations [44]. In addition to the text based representation, the presence of a figure or a model that incorporates relevant product information has been found to improve the accuracy of formed mental models [45].

The process of generating a mental model is complex, involving a variety of variables, and associated sensitivity to those variables. However, this complexity is further intensified when considering the individual interactions that take place during a traditional group idea generation session [46]. For example, individual differences may lead designer to notice and elaborate upon different components of the same given design problem, resulting is differing mental models. When these mental models are communicated among the group members, individual members may modify and evolve their own mental models, resulting in different understanding of the given prompt. This difference in mental models stems from interdependence within the group members, and it can be helpful in identifying a variety of design solutions, as well as stimulating novel solutions [9,47,48].

In the following section, multiple idea generation methods will be detailed, where interdependence amongst individuals will vary depending on the design tool. Three different types of interdependence have been suggested in organizational literature: pooled, sequential, and reciprocal [49]. Pooled interdependence can be identified when individual members of the group work independently, but the end product is an accumulation of individual efforts. Alternatively, sequential interdependence happens when one individual completes a portion of the work, which is then transferred to another individual who continues that work and so on. Finally, reciprocal interdependence can be identified when individuals share work back and forth until completion, requiring high levels of coordination. Regardless of the type of interdependence involved, individual designers on a team will use both text and visual representations, as well as recollection of previous experiences with similar products or designs. This suggests that each individual designer can approach idea generation differently based on a variety of factors, including but not limited to, experience, mental model of the problem, and spatial ability [50]. Similar to understanding a given design problem, these individual differences in idea generation can lead to more varied and novel solutions, as suggested by the majority of ideation tools being collaborative in nature.

2.2 Progressive Ideation

Ideation, or idea generation, refers to the process of generating ideas, specifically in engineering, ideas directed towards solving a problem. Traditionally, ideation has been an individual task, where one designer takes the problem statement and produces solution concepts addressing the problem. Additionally, it is advised that the designer should avoid being influenced by existing solutions [51]. However, research has demonstrated that having knowledge of, or being able to review partial solutions or existing solution to similar problems can stimulate ideation. In order to take advantage of this, many group ideation methods have been developed [48]. Some examples of these are brainstorming, gallery sketching, the 6-3-5 method, and C-Sketch. These methods focus on taking advantage of multiple individuals being able to review each other's ideas while they are in progress. This is further supported by a meta-analysis of motivation gains that shows working in groups results in an increase in motivation for inferior group members when compared to individual work [52]. Additionally, research on virtual teams shows that feedback and discourse among team members results in overall positive effects on team member's motivation and satisfaction, as well as the overall team performance [53]. Conversely, there is also literature showing group ideation activities generally lead to inferior products compared to individual sessions [54,55], however, structured approaches to group ideation have shown to yield superior ideation results [9].

2.2.1 Gallery method

The gallery sketching method is described in engineering design texts as a design tool for ideation in the concept development phase of the design process [2]. The gallery sketching method is a multi-step process where the designers essentially take breaks between their sketching activities to discuss their ideas with other designers. The first stage of gallery sketching is a silent sketching phase where designers individually produce sketches that provide solution to the given problem. After a certain amount of time, the sketching activity is paused, and the designers are encouraged to review each other's solutions and discuss how the solutions address the problem. As the second stage ends, the designers resume sketching in the third stage to complete their incomplete solutions or improve upon them based on the discussions. Therefore, although there is individual work that occurs, the method of gallery sketching leverages reciprocal interdependence because the entire group of the designers share feedback and have direct input into each other's models.

2.2.2 Method 6-3-5

The 6-3-5 method was originally introduced by Rohrbach and extended by others as a concept generation technique in a group setting [56]. A group of six designers is used for this method. Each designer generated three solution ideas for the problem statement in a given amount of time. These ideas must be described only with words. Next, the ideas are passed to the next designer in a circular manner. Again, three new ideas are generated in the given amount of time, then passed on to the next designer. This is repeated five times, therefore on the past pass, the original ideas should return to the respective designers. During each pass, the designers are not allowed to verbally communicate. Moreover, duplicate ideas are not allowed on a list of ideas. As mentioned earlier, this method uses the ideas generated by one designer to stimulate more ideas from another designer. Therefore, this method reflects sequential interdependence because models and ideas are passed from one person to the next, with a final product emerging at the end. An additional challenge for the designers is also presented as they are forced to produce new ideas each turn. This method can be used by any number of designers by simply changing the number of passes. However, a group too large may run out of ideas, while a group too small may not provide the expected results.

2.2.3 C-Sketch

The C-Sketch, or collaborative sketching, was a method originally developed by the Design Automation Lab at Arizona State University [9]. This method is essentially a modification of the 6-3-5 method using graphical communication. C-Sketch can also be interpreted as a combination of the gallery method and the 6-3-5 method. The participants of C-Sketch are given a set amount of time to generate sketches for solution addressing the problem statement. After allotted time passes, the designers pass the sketches to the next designer in a circle. The designers are again given a set amount of time to additively modify the sketch they received. After the set amount of time, the sketches are passed to the next designers again and this process is repeated until the originator of the design receives the sketch. The designers are not allowed to communicate between the passes, and no text is allowed on the sketches. Similar to the 6-3-5 method, C-Sketch also reflects sequential interdependence, and can also be used by any number of designers, however, similar limitations exist.

2.2.4 Summary

Research on conceptual design and ideation tools has shown that using these tools in a collaborative setting is beneficial for concept generation. Function Structure models can also be considered a conceptual design tool, and as such, the concept of a collaborative approach can also be extended to Function Structure modeling. Therefore, research should be conducted to evaluate the viability of Function Structures as a collaborative design tool.

2.3 Intellectual Merit and Broader Impacts

Further research on chaining of Function Structure models can be useful in a variety of ways. Firstly, a better understanding of model chaining, and the differences within chaining methods can provide justification for teaching students to use specific modeling approaches based on the scenario. At present, instruction on Function Structure modeling is biased towards forward chaining; backward chaining and nucleation are either given less importance or completely ignored. Function modeling lectures from three different instructors were reviewed for the type of chaining methods discussed within the lecture [57–59]. It was found that instructors covered forward chaining predominantly, with two instructors implicitly showing traces of backward chaining during in-class examples. However, it should be noted that none of the instructors explicitly identify these modeling strategies.

This may be a result of existing literature on Function Structure models that discusses the creation of the model using forward chaining in the examples [2]. Additionally, pioneering work on standardization of function vocabulary define step-by-step instructions for creating Function Structure models, wherein forward chaining is implicitly used, without any use of backward chaining or nucleation [25]. This exclusive use of forward chaining forces students and designers to approach Function Structures from an input-to-output perspective, meaning that the model is constructed in the direction of the input flows. Alternatively, backward chaining proposes the opposite directionality, allowing designers to thinking about the problem from the opposite perspective, with desired output flows as the starting point. Similarly, nucleation allows students to use the knowledge of key functions, and develop the models with those as the starting point. As the different methods approach construction of the model in fundamentally different ways, the corresponding Function Structures are also expected to be different. As such, it is important to identify how the chaining methods differ, and whether the current reliance on forward chaining is advisable, or should a different chaining method be recommended for teaching Function Structure models.

2.4 Research Questions

The overall motivation for this research is to gain insight on how functions are mentally generated and perceived by designers using Functions Structure modeling as a tool. The understanding of designer cognition gained from this work will facilitate the development of new and more refined functional modeling tools, better suited for how designers use functions. Additionally, the information about how designers and engineers think about functions will allow for improvements in teaching methods related to functional modeling, and hopefully provide the students with a more useful understanding of function modeling that can be used for generative design.

The specific goal for this research is to understand the effect of chaining methods on the final function structure. Specifically, the following research question will be addressed by this research.

How do Function Structure models generated using different chaining methods differ from each other?

RQ1: How are the Function Structure models different based on increase in functions and flows?

RQ2: How are the Function Structure models different based on change in model complexity?

RQ3: How are the Function Structure models different based on rubric based evaluation of the model?

In addition to testing the patterns observed in the pilot protocol study, a secondary motivation to answer these questions is to be able to recommend a modeling strategy that can be used for teaching Function Structure modeling. For each research question, the performance for each chaining method will be measured and the overall best performing method will be recommended. It should be noted that in these test cases, the underlying assumption is that designers will receive a partially completed model, which they must grow to full completion. This will be further explained in the following chapter.

Chapter Three
EXPERIMENT DESIGN

In order to sufficiently address the research questions, a two-part experiment was designed, beginning with a controlled experiment, which was followed by a protocol study. The primary objective of the controlled experiment was to use replication logic to understand the effects of seeding function structure models with different chaining methods. With the knowledge gained from the controlled experiment, a subsequent protocol study was designed to further understand the effect of seeding function structure models, and identify patterns within the modeling behaviors of designers.

3.1 Controlled Experiment

A mixed factorial experiment with both, within subject and between subjects, constructs was designed to investigate the effect of partially completed seed function structure models on the resultant function structure models created by participants. The participants completed both scenarios, however, depending on the manipulation, they were given distinct levels of model completion and distinct types of model chaining.

3.1.1 Variables Tested

Three independent variables were tested in this experiment: percent completion of the model, chaining of the model, and the design problem that was addressed by the model. These variables were of interest because they were expected to provide insight on how student designers think about functions in mechanical engineering via function modeling. Three levels of model completion were provided to students. Also, three different chaining methods were investigated. In addition to these groups, two different problem statements were used. Figure 11 shows a representation of the variables tested and the relational hierarchy between them.

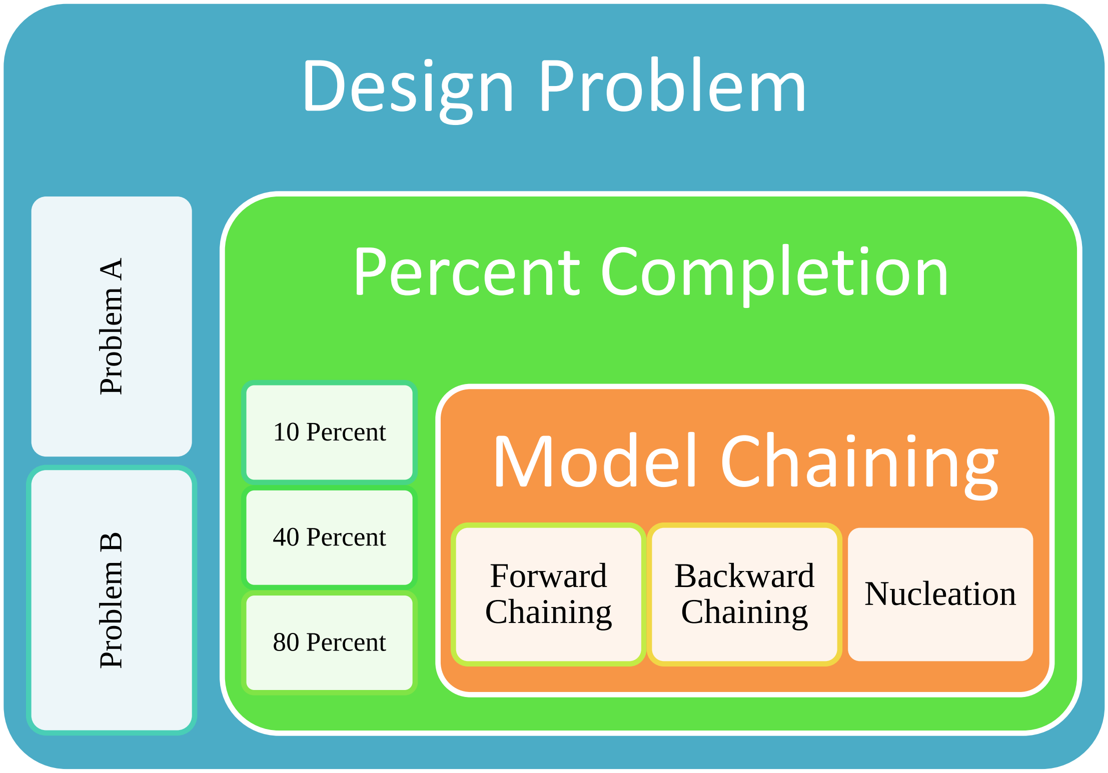


Figure 11: Relationship and Distribution of Experiment Variables

As shown in the Figure 11, the participants received two out of eighteen possible model combinations. The seed models were distributed such that each participant received one variant for both problems. The resulting function structures generated by the students were evaluated for the number of functions and flows added to the model and the quality of these models.

3.1.2 Design Problems

Since the participants were to be provided two models to complete, two different design problems were needed. These problems had to be similar enough to not generate vastly different function models, but different enough such that partially completed seed models generated from either problem would be differentiable. Additional aspects of the design problems are shown in Table 2.

Table 2: Properties of the Design Problem

Difficulty level	The problems must have material and energy flowing through the system and should require the participants to consider multiple inputs and outs.
Domain	The problem should be appropriate for mechanical engineering students.
Length	A senior level mechanical engineering student should be able to address the problem within 20 minutes.
Interest level	The problems must have real work significance and consistent across cultures.
Representation	The problems must be represented in text format.

In addition to the points presented in Table 2, the problems selected must not have an existing solution in the market, however, the solution should be conceivable. This allowed us to avoid cases where participants reverse engineer the known solution and create a function model that represents that product. The two problems developed are shown in Table 3. These problem statements were derived from existing problems that have previously been used in the field engineering design.

Table 3: Design Problems

Problem A: Automatic Clothes Ironing Device [32]
Design an automatic clothes-ironing machine for use in hotels. The purpose of the device is to press wrinkled clothes as obtained from clothes dryers and fold them suitably for the garment type. You are free to choose the degree of automation. At this stage of the project, there is no restriction on the types and quantity of resources consumed or emitted. However, an estimated 5 minutes per garment is desirable.
Problem B: Automatic Recycling Sorter[2]
Design an automatic recycling machine for household use. The device should sort plastic bottles, glass containers, aluminum cans, and tin cans. The sorted materials should be compressed and stored in separate containers. The amount of resources consumed by the device and the amount of space occupied are not limited. However, an estimated 15 seconds of recycling time per item is desirable.

As shown in Table 3, the two problems were selected and refined to be similar. Both problems statements contain three inherent functional requirements. In problem A, the device must be able to sort, iron, and fold clothes. Whereas, in problem B, the device must be able to sort, compress, and store recycling material. Similarly, both problems statements demand solutions that require sensory as well as actuation tasks, leading the solution towards an electromechanical device. Moreover, the problem statements are also written to be of similar lengths with problem A containing 70 words and problem B containing 61 words. The problem statements are structured similarly as well, with first stating the purpose of the device, followed by a description of the device functions, any resource requirements, and finally time constraints. Since the problem statements were intended to be similar, it is expected that the number of functions added by the participants should not be significantly different between problems, given other variables are held constant.

[2] https://www.asme.org/about-asme/news-media/press-releases/asme-announces-finalists-for-annual-student-design

3.1.3 Partial Function Structures

For the activity, the students were provided two partially complete function structure models. As such, this activity parallels an idea generation method exhibiting sequential interdependence because participants are provided with a partially complete model and asked to incorporate their own ideas. These partially complete models were generated from the following complete models. Figure 12 shows the complete model developed to address problem A and Figure 13 shows the complete model for problem B.

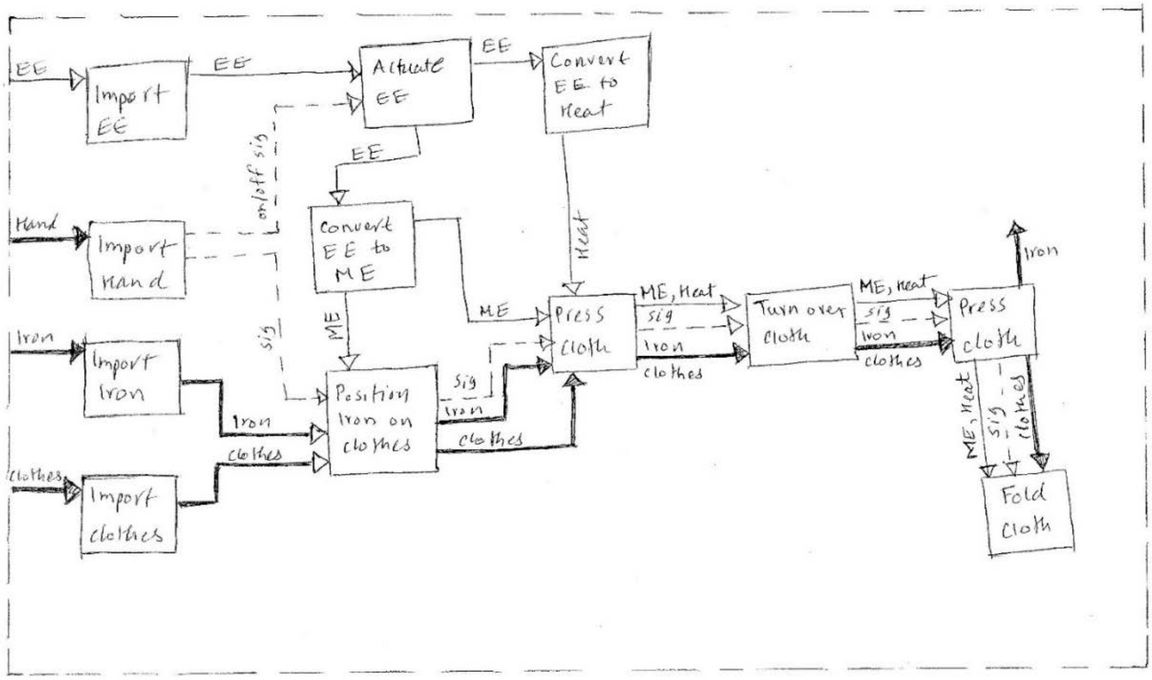


Figure 12: Complete Model of Clothes Ironing as Given in Experiment

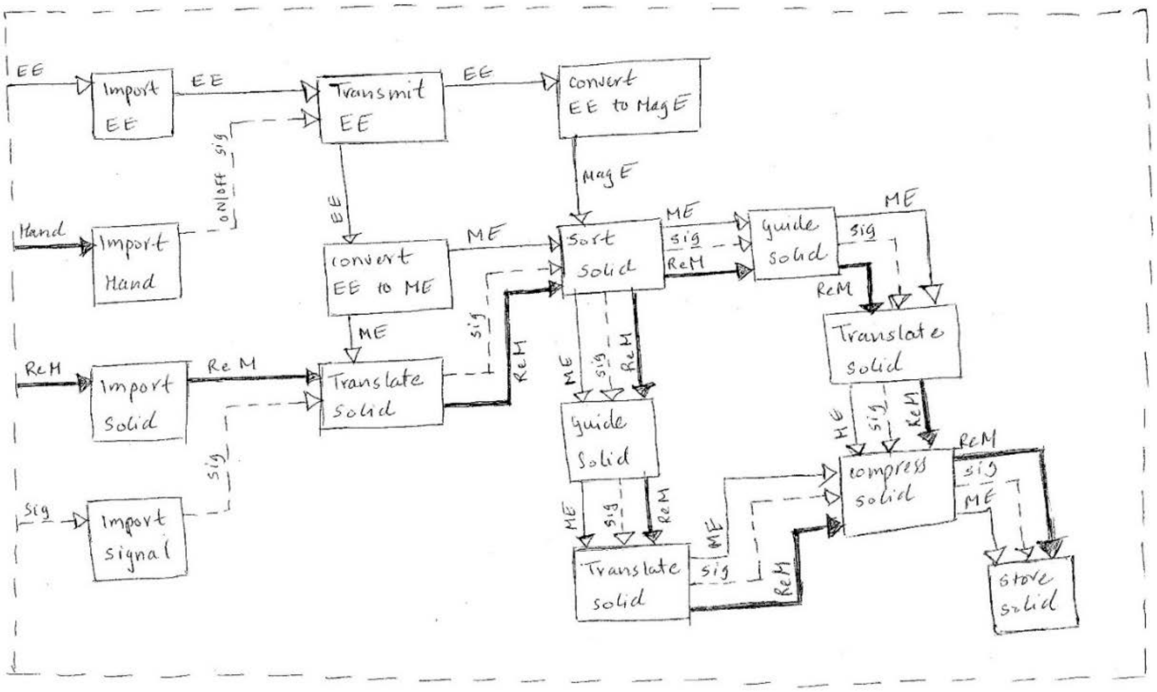


Figure 13: Complete Model of Automatic Recycling Machine as Given in Experiment

These completed function structure models were pruned as necessary to generate the partial models provided to the participants. The models were pruned to three different levels (10%, 40% and 80%), using three different methods (forward chaining, backward chaining, and nucleation). When pruning the model for a percent completion, the primary goal was to gradually increase the amount of information available in the model. As such, the 10% complete model only has a small amount of information, whereas the 80% complete model has nearly all the information that is presented in the complete models shown in Figure 12 and Figure 13. For chaining, the primary goal was to represent the core idea of the chaining method in the pruned models. Figure 14, Figure 15, and Figure 16 show model for problem A pruned to 10%, 40%, and 80% completion for nucleation, forward chaining and backward chaining respectively.

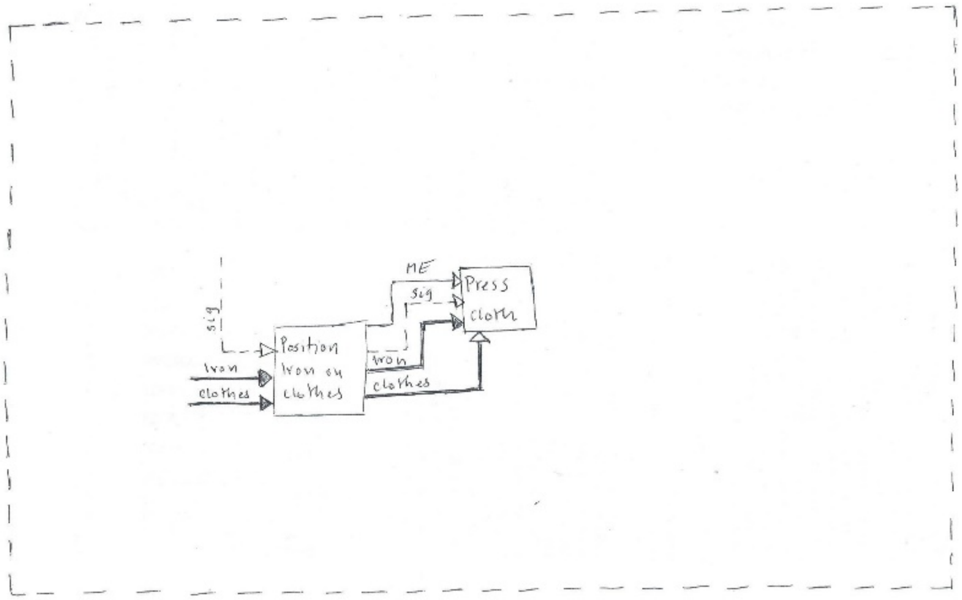


Figure 14: Nucleation model at 10% Completeion

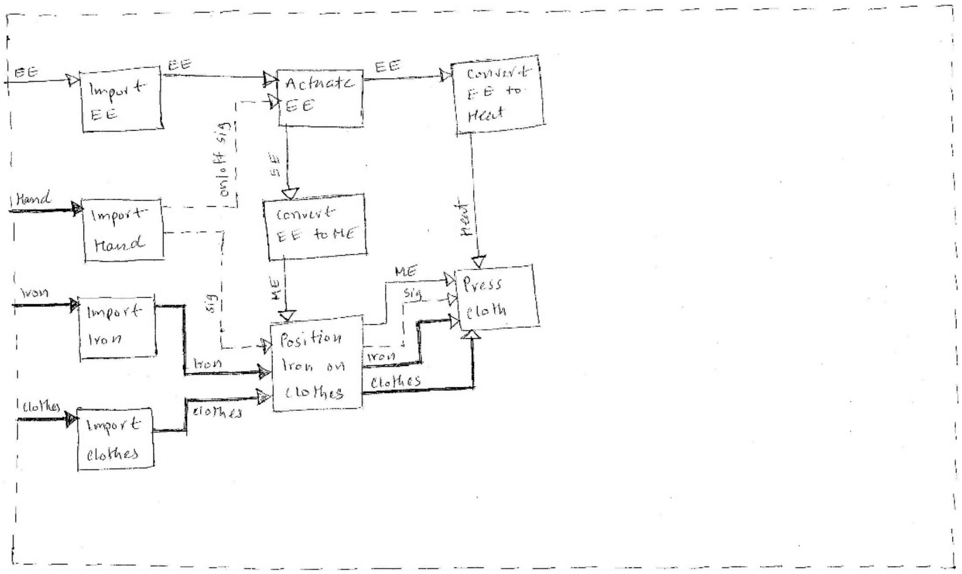


Figure 15: Forward Chaining model at 40% Completion

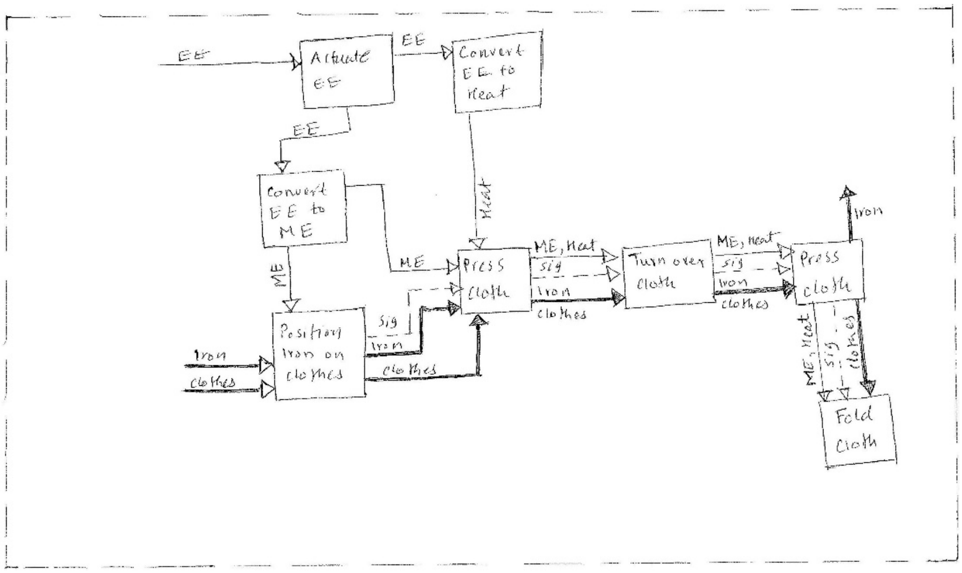


Figure 16: Backward Chaining Model at 80% Completion

As shown in Figure 14, the nucleation model starts at a nucleation point where an important or central function to the model lies. The forward chaining model shown in Figure 15 starts at the inputs of the model and functions are then chained along the direction of the flows. In this model, the outputs of a function are chained forward as the inputs to subsequent functions.

As opposed to the forward chaining model, the backward chaining model shown in Figure 16 starts with the known output function “fold clothes” and chains the inputs to this function as outputs to previous function “press cloth”, essentially chaining the model in reverse. The remaining function structure model variations were generated similarly.

3.1.4 Participants

The participants used in this experiment were senior level undergraduate mechanical engineering students enrolled in ME 4010 (mechanical design course) with knowledge of basic mechanical engineering concepts as well as function modeling with function structures. This group of students was selected because they are a reasonable representation of novice engineers who would be using design tools such as function structures for novel design [60].

A total of 86 participants were used in the experiment. The experiment design yielded each participant as a unique scenario; however, the individual elements of the study were replicated across participants. There were nine variations of the partially complete model provided to the participants for each of the design problems. Each participant was given two initial models to complete, one for each design problem, from a pool of eighteen model variations shown in Table 2. A minimum replication of seven participants is found for BB40.

Table 4: Partial Function Model Variants

Model	Participants	Model	Participants
AB10	11	BB10	10
AB40	9	BB40	7
AB80	11	BB80	9
AF10	10	BF10	8
AF40	8	BF40	10
AF80	11	BF80	12
AN10	9	BN10	10
AN40	9	BN40	10
AN80	8	BN80	10

In Table 4, a four-character code is used to describe the model variants. The code is constructed as follows.

- Digit 1 = Design problem (A or B)
- Digit 2 = Model Chaining (F, B, or N)
- Digit 3 and 4 = Percent Completion of the Model (10, 40, or 80)

For example, a model addressing design problem B, that is 40 percent complete with nucleation is coded as BN40. The distribution of the model to the 86 participants is also shown in Table 2.

3.1.5 Execution Procedure

The designer study was conducted during a regular class period. The students were informed about the activity on the day of the class and were introduced to the experiment as an activity to practice the material learned in the course curriculum. The students were awarded no extra credit for this task. Since the activity was performed as a normal in class activity, the students were seated on 8-person round tables. Figure 17 shows an example of the setup.

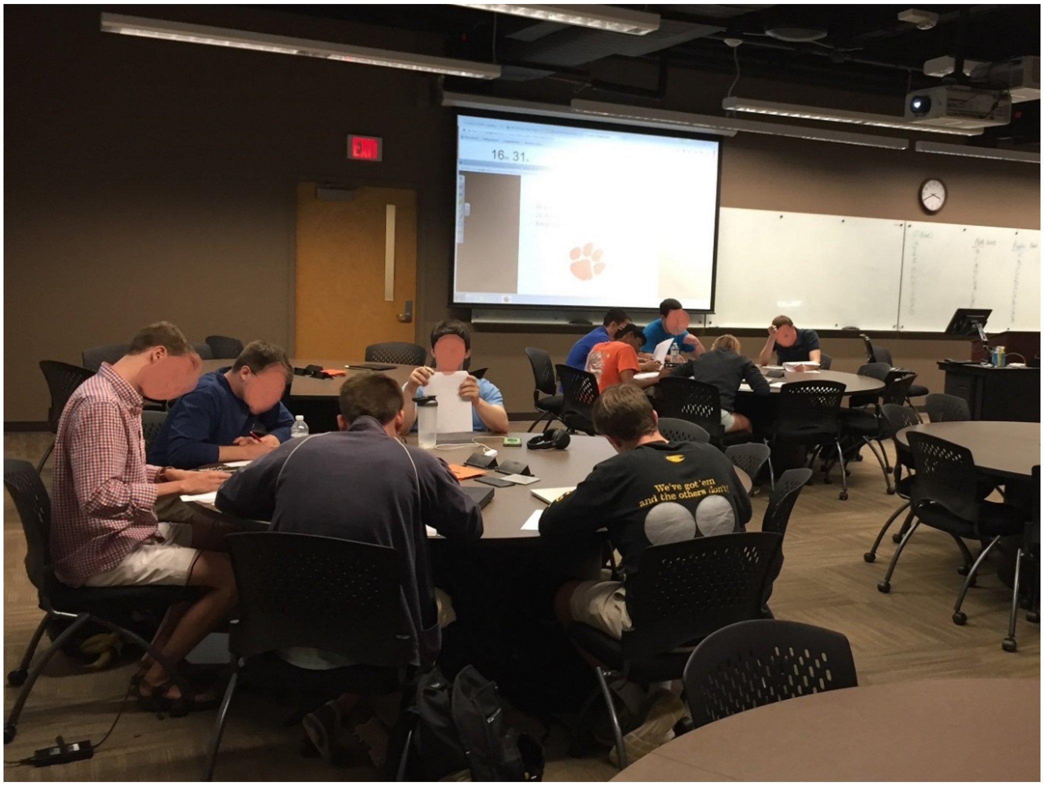


Figure 17: Experimental Setup for the Designer Study

The experiment packets were distributed to the students followed by a set of instructions where the students were informed that this was a function modeling activity, and they had the remainder of the class period to complete it. The students were allowed to ask questions before starting the activity, however, facilitators were present in the room during the activity to answer any individual questions.

3.2 Protocol Study

Following the controlled experiment, a protocol study was conducted to further understand the effect of seeding function structure models with partially completed models. The design problems developed for the controlled experiment were also used for the protocol study. Additionally, the same variables were tested with, however, in this case, each participant completed one unseeded function structure model, and one seeded function structure model, where the partially completed function structure models developed in the controlled experiment were used as seed models.

3.2.1 Participants

Similar to the designer study, the participants used in the protocol study were senior level mechanical engineering students enrolled in ME 4010. However, the protocol study was conducted during a summer semester, resulting in fewer participants. The assignment of seed models for protocol study participants is shown in Table 5.

Table 5: Participant Distribution for Protocol Study

Model	Participants	Model	Participants	Model	Participants
AB40	3	AB80	2	AF10	3
AN10	3	BB10	3	BF40	2
BF80	2	BN40	2	BN80	2

As shown in Table 5, the total number and distribution of seed models in the protocol study is different compared to that in the designer study. In order to account for the small number of participants available for the protocol study, interesting seed models from the user study were identified (after a preliminary analysis of the results), and these were used to conduct the protocol study.

3.2.2 Execution Procedure

The approach used in this experiment was developed to capture the modeling behavior of designers while creating function structure models [32,39]. The protocol study was divided into three main sections: a pre-activity survey, the modeling activity, and a post-activity survey. The pre-activity survey is focused on capturing the participants understanding of function, their experience with function models, and their knowledge and experience regarding household electromechanical products. The post-activity survey asks the participant about their understanding of the given problem, their ability to perform the activity, and their level of satisfaction with the model generated. A rubric was provided for them to evaluate the function structure model based on completeness, usefulness and solution quality.

The modeling activity required participants to read and understand the problem statement, then create a function structure model addressing the problem statement. The experimental setup used is shown in Figure 18.

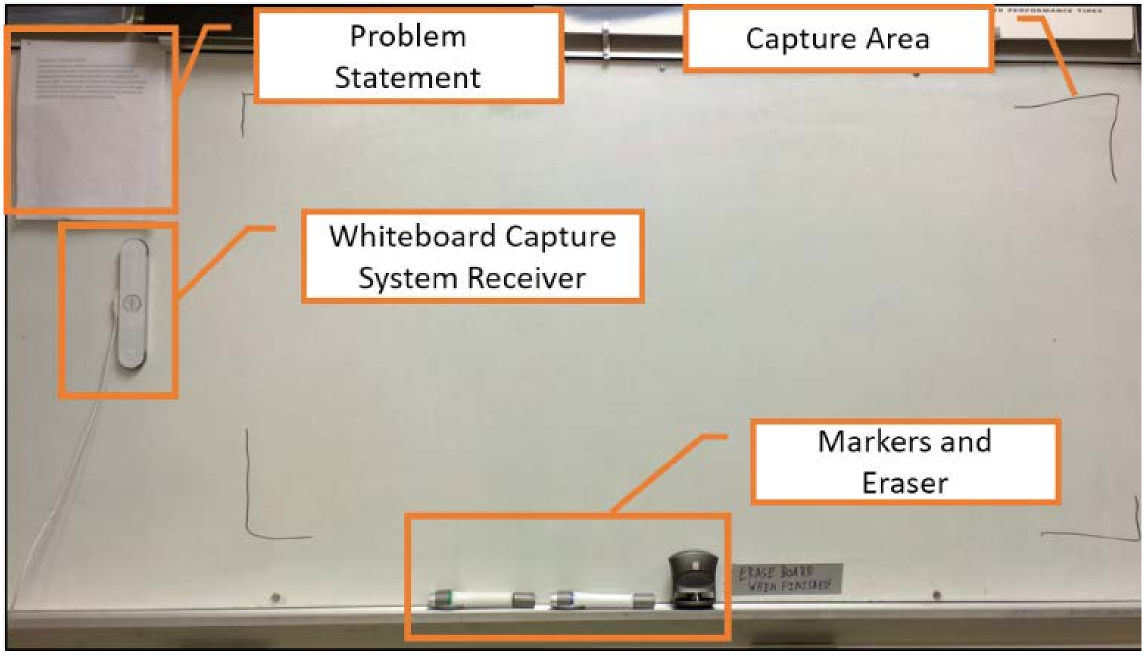

Problem Statement
Capture Area
Whiteboard Capture System Receiver
Markers and Eraser
ERASE BOARD WHEN FINISHED!

Figure 18: Experimental Setup for Protocol Study

The experimental setup consisted of a whiteboard, a whiteboard capture system (receiver, marker and eraser), a problem statement, and a video recording camera.

3.3 Data Collection and Analysis

Final function structure models were collected from both parts of the experiment, the controlled experiment and the protocol study. These were analyzed to identify the increase in number of functions and flows from the given model to the final model, the change in complexity between the given model and final model, and finally a rubric based evaluation of the models.

3.3.1 Percent Increase of Functions and Flows

As the goal of this experiment was to evaluate the effects of different seed models, metrics needed to be developed to compare the final function structures developed. The number of functions and flows added in the final model were counted, however, depending on the design problem, the level of completion, and the type of chaining used, the number of functions and flows present in the equivalent seed models may vary. For example, a 10% complete seed model developed using forward chaining for problem A has different number of functions and flows compare to the same seed model for problem B. This different persists as long as at least one variables is changed. Therefore, to avoid bias from type of seed model used, the number of functions and flows added in the final model were divided by those in the given seed model, and the metric used to measure the difference in output was the percent increase in the number of function and flows.

For instance, if a participant was given a seed model with 5 functions and 10 flows, and the final model generated contained 10 functions and 15 flows, the result is a 100% increase in the number of functions and a 50% increase in the number of flows. Since the seed models varied in the level of completion, the percent increase of functions and flows was expected to be highest for 10% complete models, and lowest for 80% complete models.

3.3.2 Change in Model Complexity

In addition to the increase in functions and flows to the given model, the change in model complexity was also evaluated. In order to generate a complexity vector for the function structures, a bipartite graph is necessary. The function structures were converted into bipartite graphs; which were then converted into an n x 2 matrix where n is the number of connections present in the function structure [61]. Figure 19 shows a function structure completed by a student participant, and corresponding bipartite graph generated for that function structure.

As shown in Figure 19, the bipartite graph generated for the function structure distils the compound flow chart into a 2-coulmn graph which shows the elements present in the graph and the relationships between the elements. This graph can then be converted into a matrix and processed in MATLAB to generate complexity metrics [62–64].

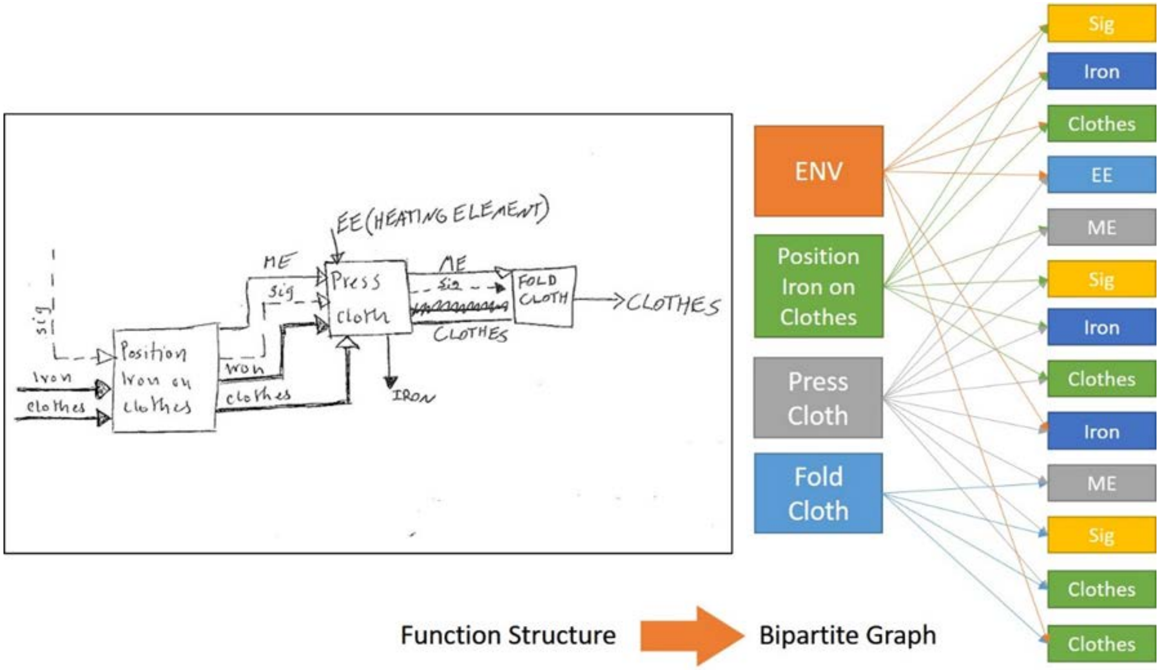


Figure 19: Converting Function Structure to Bipartite Graph

After creating bipartite graphs for all the function structures, the next step was to generate complexity metrics for the bipartite graphs. Complexity metrics have been used in the past to categorize product design, predict assembly time and market value for products [61,64,65], and estimate assembly defects [63]. All 29 of the complexity metrics present in the complexity vector are shown in Table 6.

Table 6: 29 Complexity Metrics in the Complexity Vector

Class	Type	Metric	Class	Type	Metric	Class	Type	Metric
Size	Dimension	Elements	Centrality	Between-ness	Sum	Decomposition	Ameri-Summers	
		Relations			Max		Core Numbers In	Sum
	Connection	DOF			Mean			Max
		Conn			Density			Mean
Inter-connection	Shortest Path	Sum		Clustering Coefficient	Sum			Density
		Max			Max		Core Numbers Out	Sum
		Mean			Mean			Max
		Density			Density			Mean
	Flow Rate	Sum						Density
		Max						
		Mean						
		Density						

After generating complexity vectors for each function structure, the complexity vectors of the student generated function structure needed to be compare to those of the initial function structures provided during the experiment. In order to do this comparison, four different pairwise distance comparison formulae were used: Euclidian distance, Hamming distance, cosine distance, and Chebychev distance.

3.3.2.1 Complexity Distance Metrics

The Euclidian distance between the complexity vectors can be obtained by calculating a 2-norm of the pairwise distance between each element of the vector. Equation 1 shows how the Euclidian distance is being calculated, however the MATLAB command ***pdist(X)*** was used to compute the distances.

$$D_{euc} = \sqrt{\sum (x_i - y_i)^2} \tag{1}$$

In Equation 1, x_i is the i^{th} element of the vector x (final model complexity vector) and y_i is the i^{th} element of the vector y (initial model complexity vector).

The Hamming distance between the complexity vectors examines the percentage of values between two vectors that differ. In other words, the Hamming distance between two vectors reflects the number of substitutions that need to be made in order to equate both vectors. Equation 2 shows how the Hamming distances are being calculated, however the MATLAB command ***pdist(X, 'hamming')*** was used to compute the distances.

$$D_{ham} = \frac{number\ of\ (x_i \neq y_i)}{n} \tag{2}$$

In Equation 2, x_i and y_i are the i^{th} elements of the final and initial complexity vectors respectively, and n is the number of elements in either vector. It should be noted that this calculation requires both vectors to be of the same length. Additionally, the Hamming distance is always a number between 0 and 1.

The cosine distance between the vectors examines the cosine of the included angles between the points in the vectors. Equation 3 shows how the distances are calculated, but similar to the hamming distance the MATLAB command ***pdist(X, 'cosine')*** was used to compute the actual distances.

$$D_{cos} = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\left(\sqrt{\sum_{i=1}^{n} x_i^2}\right)\left(\sqrt{\sum_{i=1}^{n} y_i^2}\right)} \tag{3}$$

In Equation 3, x_i and y_i are elements of initial and final complexity vectors. Similar to Hamming distance, cosine distance between the vectors also ranges from 0 to 1.

The Chebychev distance between two vectors measures the maximum magnitude difference between the coordinates of a pair of objects. Equation 4 can be used to calculate Chebychev distances, however the MATLAB command ***pdist(X, 'chebychev')*** was used in this case as well.

$$D_{chb} = max\{(x_i - y_i)\} \tag{4}$$

In Equation 4, x_i and y_i are elements of final and initial complexity vectors.

The four different distances were used to perform the same analyses to avoid any biases that may exist due to the distance calculation method used. The distance calculation was followed by a statistical analysis to determine the significance of the findings.

3.3.3 Evaluation of Function Models

The final function structures generated from the experiment were analyzed using an externally developed rubric [33,66]. In this rubric, the function structures are evaluated based on several aspects of the model such as blackbox models, flow conservation, product representation, and modeling conventions. The rubric consists of 18 questions that require the rater to provide a binary response of “0” or “1”, which correspond a response of “yes” or “no” respectively. Table 7 shows the questions that make up the rubric.

Table 7: Evaluation Rubric for Function Structure Models

1	Model contains a black box?	10	Do the function–flow pairs in the functional model take the general form of a verb/noun pair?
2	Black box contains input and output flows?	11	Is the functional model free of nonsensical functions?
3	Are the input and output flows in the black box appropriate?	12	Is the functional model free of nonsensical flows?
4	Does the black box represent flow conservation?	13	Is the model free of instances where the system acts on the system?
5	Do inputs from the black box match functional model inputs?	14	Is flow directionality consistent with the transformation in the functions?
6	Do outputs from the black box match functional model outputs?	15	Are flows conserved across function transformations?
7	Does the functional transformation described by the black box represent a plausible overall system functionality?	16	Are flow paths appropriate for product representation?
8	Does the black box function–flow pair take the general form a verb/noun pair?	17	Does the functional model represent flow conservation?
9	Do the function–flow pairs in the functional model overall represent a plausible view of the product?	18	Are the proper energy, material, and signal flow arrow conventions followed?

In the evaluation rubric, the first eight questions are targeted towards the black-box model, while the remaining questions are regarding the expanded Function Structure model. The final score of the model is a sum of the responses to all eighteen questions. It should be noted that this rubric does not measure the solution quality, rather it measures the quality or completeness of the Function Structure model itself. As such, a poorly articulated function structure may still receive a full score on the rubric if the function structure meets the rubric requirements. Therefore, the function structures developed for different seed models were not expected to have a significant difference in the assessment scores. Similarly, the designer study and protocol study responses were also not expected to be significantly different.

Chapter Four
RESULTS: COMPARISON OF EXPERIMENT TYPES

The final function structure models obtained from the designer study, and the protocol study were analyzed for the increase in function and flows, change in model complexity, and model evaluation. The models collected from the two different experiment types were compared to identify any significant differences that exist. The comparison was done based on the chaining method (forward, backward, or nucleation), and level of completeness (10 percent, 40 percent, or 80 percent) of the given seed model. The motivation for this comparison was primarily the intellectual merit of understanding whether the models generated in a protocol study were comparable to those generated in a designer study. However, a secondary motivation was to test if the general trends observed in one set of models was generalizable to the other set of Function Structure models. The reminder of this chapter will focus on various modes and level of comparison between designer study models and protocol study models.

4.1 Increase in Functions and Flows

The percent increase in the number of functions and flows was calculated for all the models generated by participants in each study. In order to identify any significant difference between results from the protocol study and the designer study, a two-sample t-test was performed assuming unequal variances. For this test, the null hypothesis stated that there was no difference between the two means. For this t-test and all subsequent t-tests, alpha value of 0.05 was used. Table 8 shows the result from the test comparing the percent increase in the number of functions.

Table 8: Comparison of Increase in Function

	Protocol	Designer
Mean	0.902	0.77473
Variance	0.671	0.61813
Observations	22	169
Hypothesized Mean Difference	0	
df	26	
t Stat	0.687	
P(T<=t) one-tail	0.249	
t Critical one-tail	1.706	
P(T<=t) two-tail	0.498	
t Critical two-tail	2.056	

As shown in Table 8, the test was unable to reject the null hypothesis, and therefore concluded that there was not enough evidence to show a significant difference (with a 95% confidence level) between the results from the user study and the protocol study. The t-test was repeated for percent increase in the number of flows with the same null hypothesis, and the results are shown in Table 9.

Table 9: Comparison of Increase in Flows

	Protocol	Designer
Mean	1.219	1.081
Variance	0.957	0.867
Observations	22	169
Hypothesized Mean Difference	0	
df	26	
t Stat	0.625	
P(T<=t) one-tail	0.269	
t Critical one-tail	1.706	
P(T<=t) two-tail	0.537	
t Critical two-tail	2.056	

Similar to increase in functions, the test was not able to show a significant difference in the percent increase of flows between the protocol study and user study results. However, it can be noted from the tables above that the variance of the sample data is high in both cases tested. This may be a result of the different levels of model completion provided in the seed models. In order to account for the high variance, the increase in functions and flows between user study and the protocol study is compared based on the model completion level given in the seed models. Figure 12 and Figure 13 show the comparison of increase in functions and increase in flows respectively.

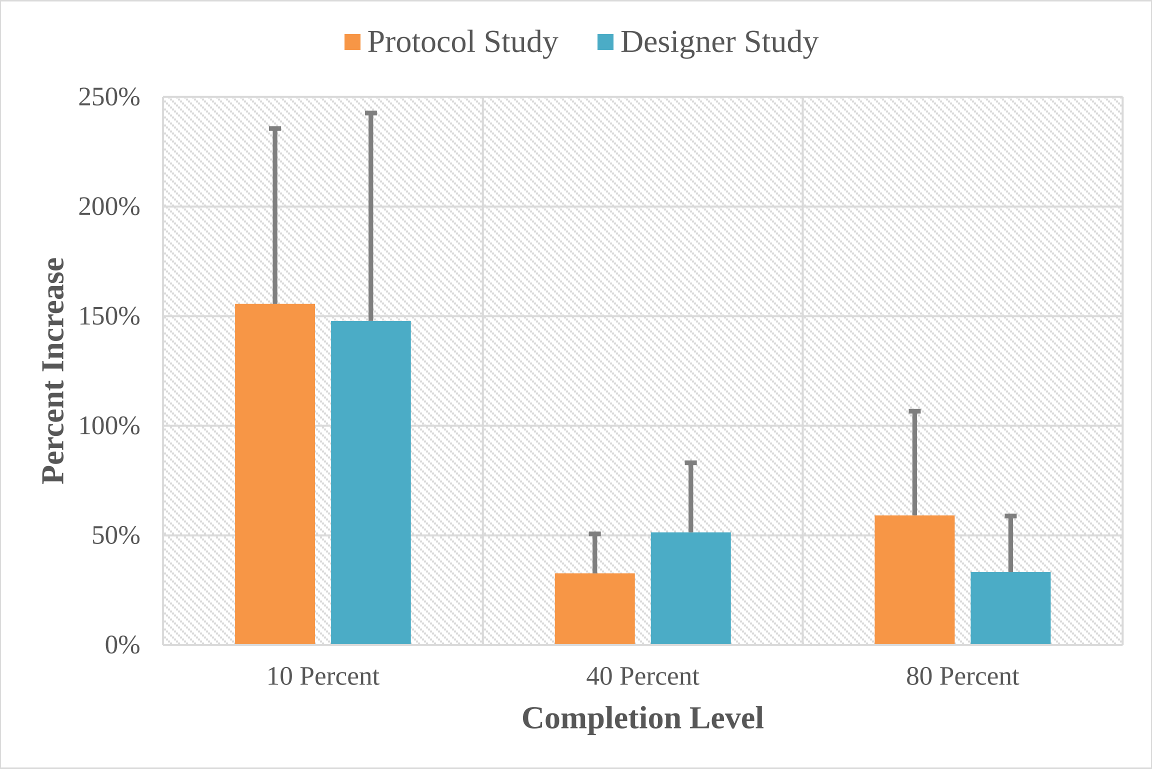


Figure 20: Comparison of Increase in Functions Based on Completion Level

As shown in Figure 20 and Figure 21, there is a relatively small difference in the sample mean of 10% complete models, whereas 40% and 80% complete models show higher differences between user study and protocol study. Additionally, 40% complete models show a higher increase in the user study, whereas 80% complete models show a higher increase in the protocol study. This is true for both functions and flows. However, these differences are only in the sample mean, and may not be significant.

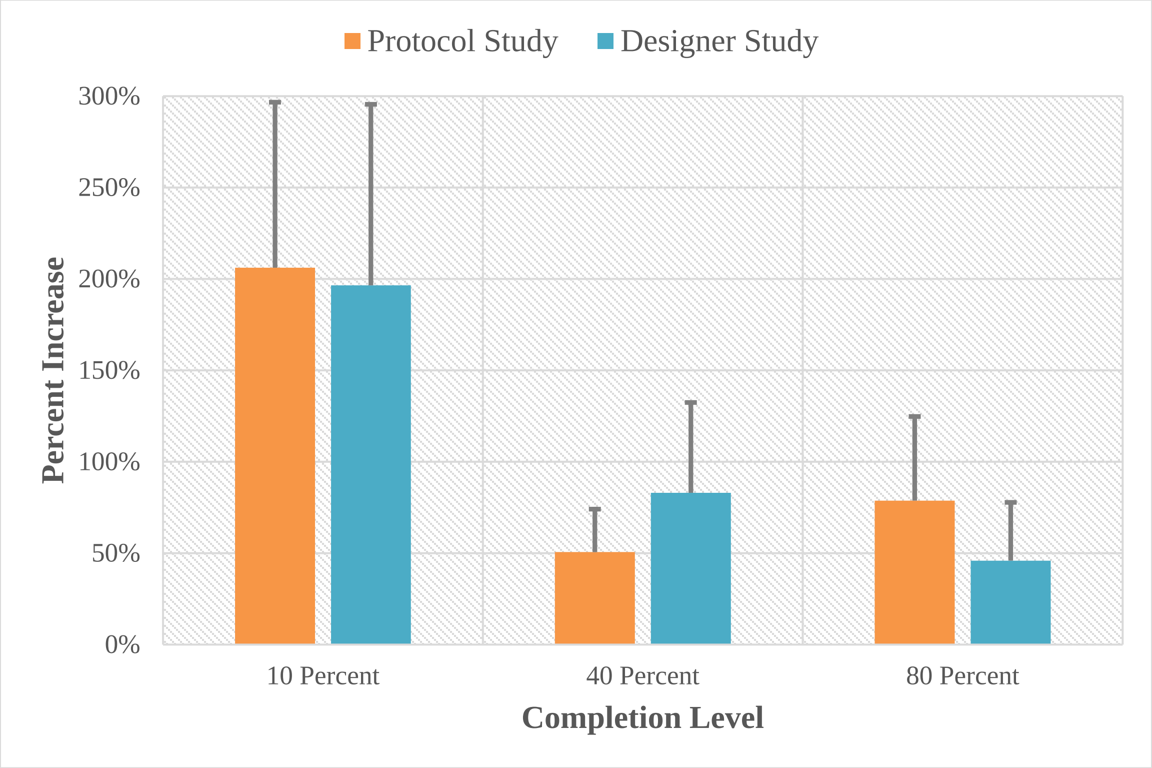


Figure 21: Comparison of Increase in Flows Based on Completion Level

In order to identify the significance of these differences, a set of two-sample t-tests are done assuming unequal variances, and the results are shown in Table 10. A hypothesis of zero mean difference was used in all t-tests done in this set, along with an alpha value of 0.05 to maintain a 95% confidence level.

Table 10: t-test Results for Comparison Based on Completion Levels

Case		Protocol Study		Designer Study		p-value
		Mean	Variance	Mean	Variance	
Functions	10 Percent	1.556	0.720	1.478	0.915	0.807
	40 Percent	0.327	0.038	0.514	0.103	0.052
	80 Percent	0.591	0.271	0.333	0.066	0.280
Flows	10 Percent	2.062	0.922	1.965	0.998	0.785
	40 Percent	0.506	0.064	0.830	0.249	0.016
	80 Percent	0.787	0.254	0.459	0.103	0.175

As shown in Table 10, five out of six cases for comparison between protocol study and user study results showed no significant difference. The significant difference between the increase in flows (for 40% complete models) in the two experiment methods is unexpected because the two experiments were conducted using the same design problems, and similar instruction for completion of the models. Additionally, the 10% and 80% complete models did not show significant differences between the protocol study and designer study. Moreover, the number of functions added at 40% completion level were not found to be different between the two experiments. Therefore, this difference in increase in flows at 40% completion level suggests a more targeted study should be conducted to reaffirm the significance of the difference.

Further investigation of the differences between protocol study results and the user study results was done by reviewing the percent increase in functions and flows for three different chaining methods. The results for functions and flows are shown in Figure 6 and Figure 7 respectively.

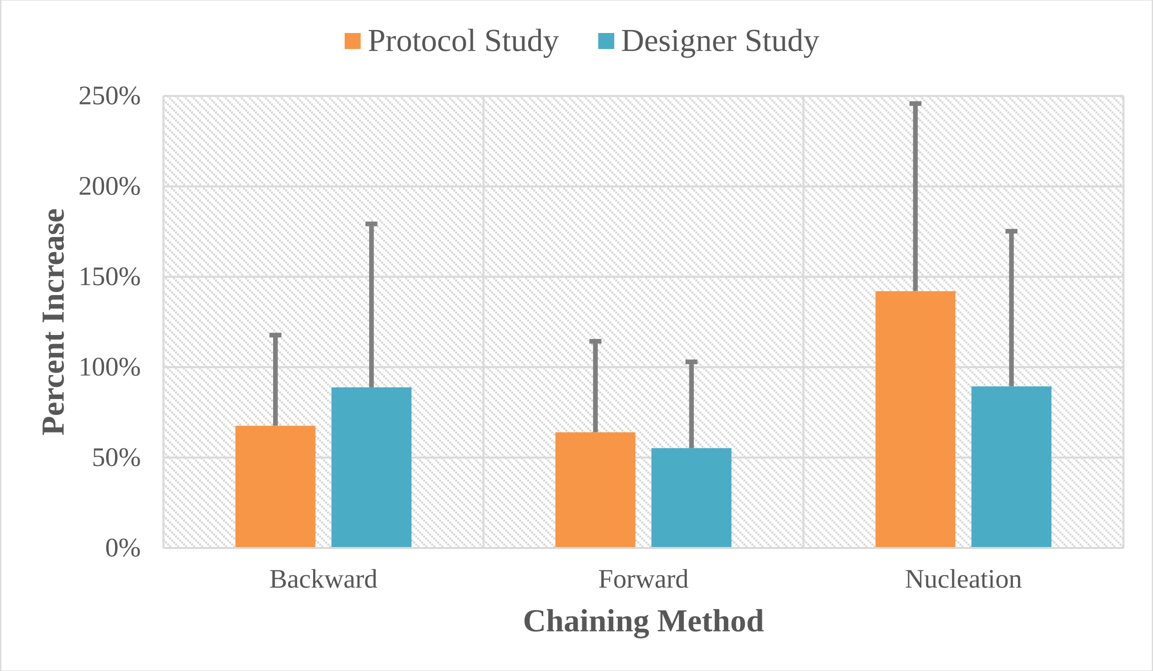


Figure 22: Comparison of Increase in Function Based on Chaining Methods

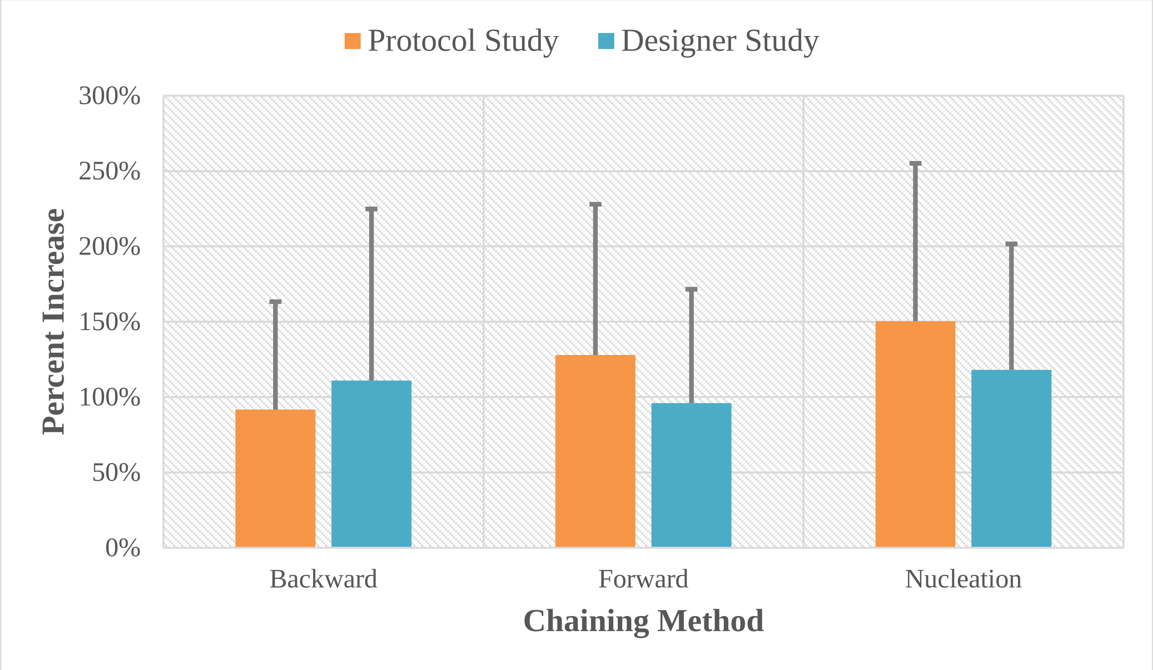


Figure 23: Comparison of Increase in Flows Based on Chaining Methods

Comparison of mean increase in functions shows that nucleation and forward chaining show a higher sample mean for protocol study, whereas backward chaining has a higher sample mean in the designer study. These trends are also observed for increase in flows, however, the forward chaining models show a higher difference for flows compared to functions. Conversely, in the case of nucleation models, a higher difference was found in the increase in functions compare to flows. However, similar to the comparison based on completion level, the differences in means do not necessarily show a statistically significant difference between the two studies. Therefore, a set of two-sample t-tests was conducted, assuming unequal variances, to determine the significance of the observed differences. Table 11 shows the mean, variance, and p-values for each case tested.

Table 11: t-test Results for Chaning Based Comparison

Case		Protocol Study		Designer Study		p-value
		Mean	Variance	Mean	Variance	
Functions	Backward	0.676	0.287	0.888	0.832	0.362
	Forward	0.640	0.295	0.552	0.232	0.695
	Nucleation	1.421	1.255	0.894	0.749	0.269
Flows	Backward	1.219	0.957	1.081	0.867	0.537
	Forward	0.918	0.585	1.110	1.320	0.548
	Nucleation	1.279	1.167	0.960	0.582	0.471

As shown in Table 11, none of the cases tested showed a significant difference between the user study and the protocol study. The lowest p-value was seen in the case of functions using nucleation, suggesting that the difference in mean for that case is more significant than other cases, while still being insignificant for 95% confidence level.

Following the comparisons based on completion level and model chaining, it can be concluded that Function Structure models generated in the protocol study were comparable to those in the designer study.

4.2 Change in Model Complexity

Next, the function structure models collected from the two experiments were compared based on the change in graph complexity. As discussed previously (chapter 3), the Function Structure models collected from both studies were converted to bipartite graphs, which were then used to calculate complexity vectors. Subsequently, the distance between the complexity vector of the given seed model and the final model were calculated using the four different distance metrics. Figure 24 shows a comparison between mean complexity distances for both studies.

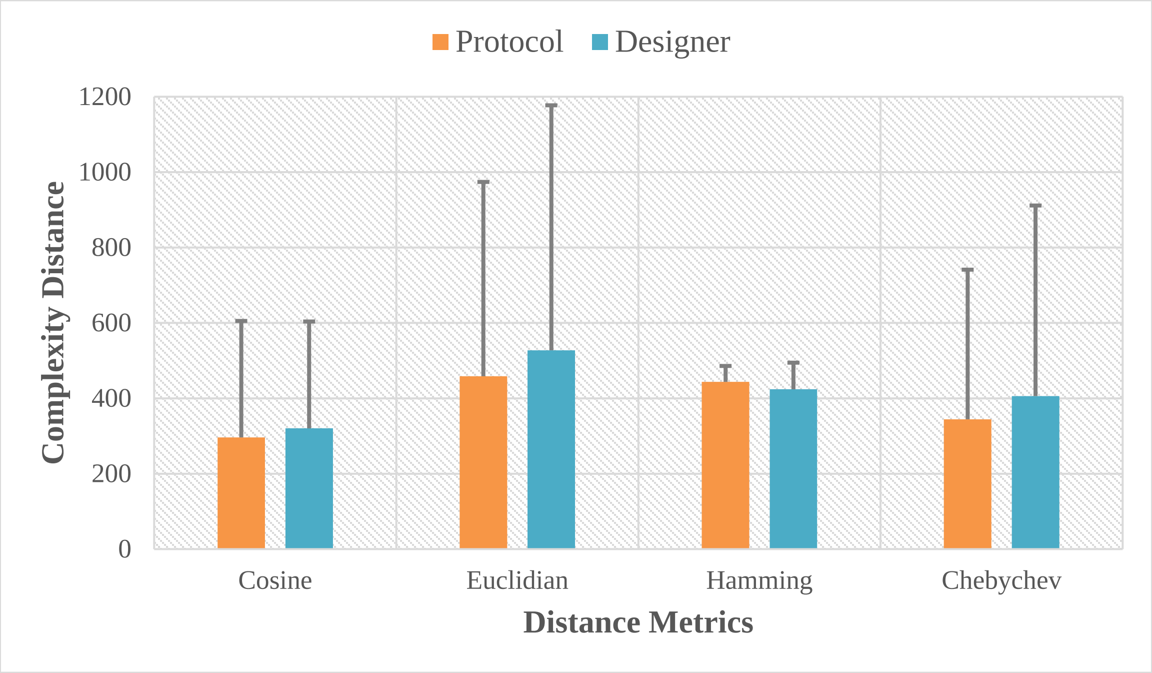


Figure 24: Comparison of Change in Model Complexity Between Experiment Types

It should be noted that the values shown in Figure 24 have been scaled from their original values show them on a single graph. The cosine distances were scaled up by a factor of 2000, whereas the hamming distances were scaled up by a factor of 500. However, the subsequent analysis has been performed with the original values.

As shown in Figure 24, cosine distance and Hamming distance showed negligible mean differences between designer study and protocol study, whereas Euclidean distance and Chebychev distance showed larger differences. However, none of the distances are greater than 20%, and therefore it is not clear if any of the differences are significant. In order to identify any statistically significant differences, a two-sample t-test was performed assuming unequal variances. The means from both populations were assumed to be equal. Table 12 shows the t-test results for Cosine distance and Euclidian distance.

Table 12: t-test Comparison With Cosine and Euclidian Distances

	Cosine		**Euclidian**	
	Designer	*Protocol*	*Designer*	*Protocol*
Mean	0.160	0.148	527.449	458.572
Variance	0.020	0.025	424766	278291
Observations	169	22	169	22
Hypothesized Mean Difference	0		0	
df	26		30	
t Stat	0.339		0.559	
P(T<=t) one-tail	0.369		0.290	
t Critical one-tail	1.706		1.697	
P(T<=t) two-tail	0.737		0.580	
t Critical two-tail	2.056		2.042	

As shown in Table 12, the t-tests were unable to reject the null hypothesis for Cosine distance as well as Euclidian distance. Similarly, Table 13 shows that the t-tests comparing protocol study and designer study based on Hamming distance and Chebychev distance also resulted in failure to reject the null hypothesis.

Table 13: t-test Comparison With Hamming and Chebychev Distances

	Hamming		**Chebychev**	
	Designer	*Protocol*	*Designer*	*Protocol*
Mean	0.848	0.887	405.874	344.409
Variance	0.020	0.007	257137	165014
Observations	169	22	169	22
Hypothesized Mean Difference	0		0	
df	38		30	
t Stat	-1.823		0.646	
P(T<=t) one-tail	0.038		0.261	
t Critical one-tail	1.686		1.697	
P(T<=t) two-tail	0.076		0.523	
t Critical two-tail	2.024		2.042	

As shown in Table 12 and Table 13, high variance was found in all the data sets compared, except the Hamming distance set. This is expected because the data set contains seed models with different levels of completion and chaining. In order to account for this, the data set was filtered based on chaining and completion, and the subsets were compared again to identify any significant differences that exist between protocol study and designer study models. Table 14 shows the results of t-tests comparing protocol study and designer study models.

Table 14: t-tests for Complexity Distances Based on Completion Level

Case		Protocol Study		Designer Study		p-value
		Mean	Variance	Mean	Variance	
Chebychev	10 Percent	231.7	64500	197.6	39600	0.710
	40 Percent	261.9	20300	394.2	126100	0.084
	80 Percent	609.8	448500	614.2	507900	0.988
Cosine	10 Percent	0.271	0.033	0.281	0.022	0.880
	40 Percent	0.090	0.002	0.122	0.008	0.158
	80 Percent	0.032	0.002	0.081	0.009	0.034
Euclidian	10 Percent	304.0	96900	263.2	61800	0.717
	40 Percent	332.7	32200	509.4	204300	0.069
	80 Percent	837.3	737900	794.8	848200	0.912
Hamming	10 Percent	0.900	0.009	0.896	0.006	0.895
	40 Percent	0.867	0.008	0.838	0.021	0.474
	80 Percent	0.891	0.007	0.812	0.030	0.086

As shown in Table 14, a significant difference was found between models from protocol study and designer study models when comparing the 80% complete seeded models using cosine distance. The corresponding p-value for this case was found to be 0.034, which satisfies the 95% confidence level requirement. Moreover, suggestive differences were found in 40% case for Chebychev distance, and 80% case for Hamming distance; both cases had p-values smaller than 0.1.

In addition to completion levels, models from the two studies were also compared based on the chaining method of the given seed model. Table 15 shows the t-test results for comparisons between designer study and protocol study using four different distance metrics, separated by chaining methods.

Table 15: T-tests Results for Complexity Distances Based on Chaining Method

Case		Protocol Study		Designer Study		p-value
		Mean	Variance	Mean	Variance	
Chebychev	Backward	175.3	18800	202.2	34300	0.630
	Forward	540.4	410300	584.9	425500	0.867
	Nucleation	341.7	62400	425.2	237400	0.483
Cosine	Backward	0.115	0.018	0.113	0.014	0.968
	Forward	0.214	0.038	0.188	0.021	0.744
	Nucleation	0.120	0.021	0.179	0.023	0.346
Euclidian	Backward	234.9	29400	271.3	55900	0.605
	Forward	693.7	671600	750.2	704800	0.868
	Nucleation	479.0	136500	554.3	397500	0.656
Hamming	Backward	0.922	0.006	0.852	0.036	0.066
	Forward	0.857	0.007	0.831	0.016	0.473
	Nucleation	0.877	0.010	0.863	0.008	0.742

According to Table 15, no significant differences were found in any of the twelve test cases. A suggestive difference was found in the case of Hamming distance, when comparing backward chaining models. Overall, from the 24 different tests conducted, one showed significant differences, and four showed suggestive differences. However, as majority of the cases showed no significant differences between the designer study and protocol study, it was concluded that models from the two studies were similar with respect to change in model complexity.

4.3 Model Evaluation

The function structure models collected from the user study as well as those developed in a protocol study were analyzed using a grading rubric developed to assess the quality of function structure models [33,66]. A two-sample t-test assuming unequal variances was conducted to test the hypothesis that the mean difference between quality of models was zero. The results are shown in Table 16.

Table 16: Comparison of Model Evaluation Between Experiment Methods

	Protocol	Designer
Mean	15.5	14.929
Variance	6.833	6.138
Observations	22	169
Hypothesized Mean Difference	0	
df	26	
t Stat	0.970	
P(T<=t) one-tail	0.171	
t Critical one-tail	1.706	
P(T<=t) two-tail	0.341	
t Critical two-tail	2.056	

The t-test was unable to show a significant difference between the two means. However, due to high variances in the data, the scores were filtered based on completion level and chaining methods. The differences were examined, and the results are shown in Figure 25.

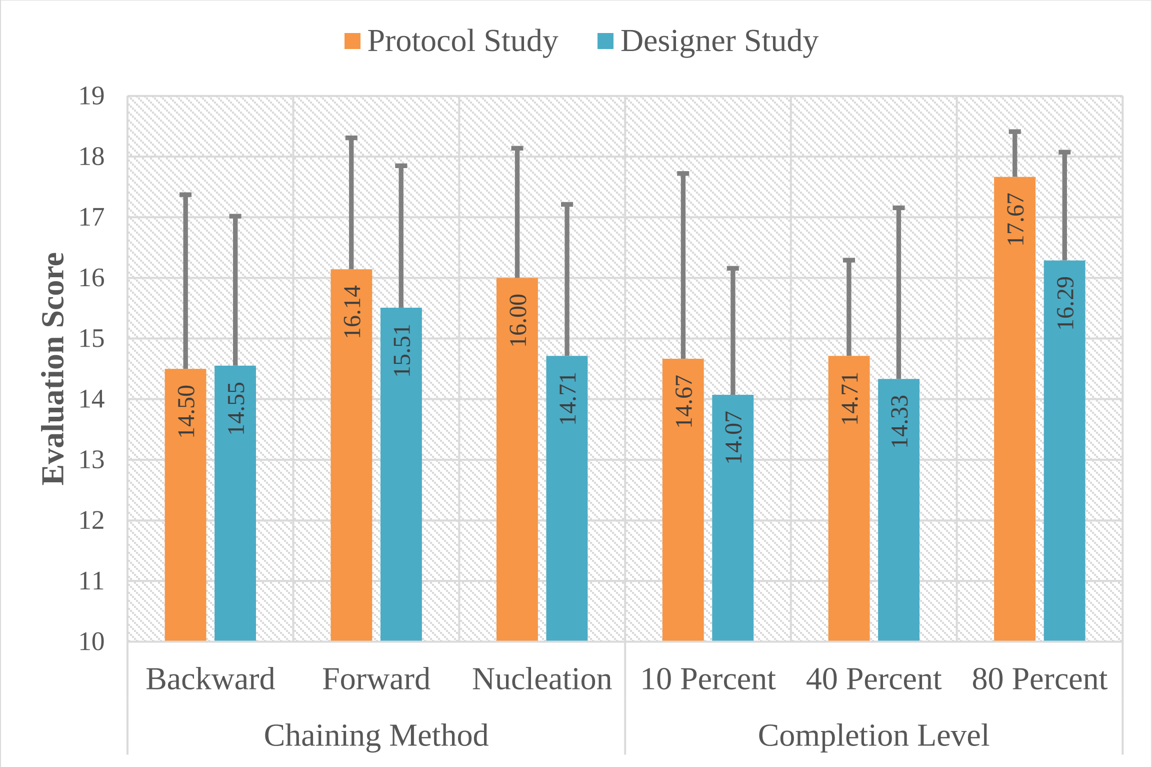


Figure 25: Comparison of Model Evalution Scores

As shown in Figure 25, varying levels of differences were found between the protocol study and designer study models. In order to identify any significant differences, a set of two sample t-tests were performed with no difference in hypothesized means, and a 95% confidence level. Results are shown in Table 17.

Table 17: t-test Results for Model Evaluation Scores

Case	Protocol Study		Designer Study		p-value
	Mean	Variance	Mean	Variance	
10 Percent	14.7	10.5	14.1	4.4	0.607
40 Percent	14.7	2.9	14.3	8.1	0.622
80 Percent	17.7	0.7	16.3	3.2	0.006
Backward	14.5	9.4	14.6	6.2	0.963
Forward	16.1	5.5	15.5	5.6	0.519
Nucleation	16.0	5.3	14.7	6.4	0.207

As shown in Table 17, 80% complete seed models were found to be significantly different between protocol study and designer study. However, it should be noted that the variance for function structure models generated from 80% complete seed models is significantly smaller compared to the remaining cases, likely resulting in the significance of difference from the designer study counterpart. Moreover, the remaining 5 out of 6 cases showed no significant differences between the studies. As a result, Function Structure models collected from the designer study and the protocol study in this research were considered to be comparable with respect to model evaluation.

Chapter Five
RESULTS: COMPARISON OF CHAINING METHODS

Following the comparison of models generated in two different experiments, the data collected from the experiments was also used to investigate similarities and differences between the chaining methods using the increase in functions and flows, change in model complexity, and evaluation scores of the final model. It should be noted that analysis from this point forward uses a combined data set containing models from both experiments.

5.1 Increase in Functions and Flows

The percent increase in the number of functions and flows was compared for the combined data set of the protocol study and the user study. Figure 26 shows the mean percent increase in the number of function based on the given seed model. It should be noted that the results for both design problems are consolidated as these results have been shown to have no significant difference [67].

The comparison of means shows that the average increase in function is different based on the level of completion of the given seed model. This result is expected as the potential for increase in functions is inversely related to the level of completion, meaning the closer the seed model is to full completion, the fewer additions are expected to the model. This is supported by 10% complete models showing a significantly larger increase compared to that of 40% models and 80% models. Notably, the difference between 40% models and 80% models seems negligible.

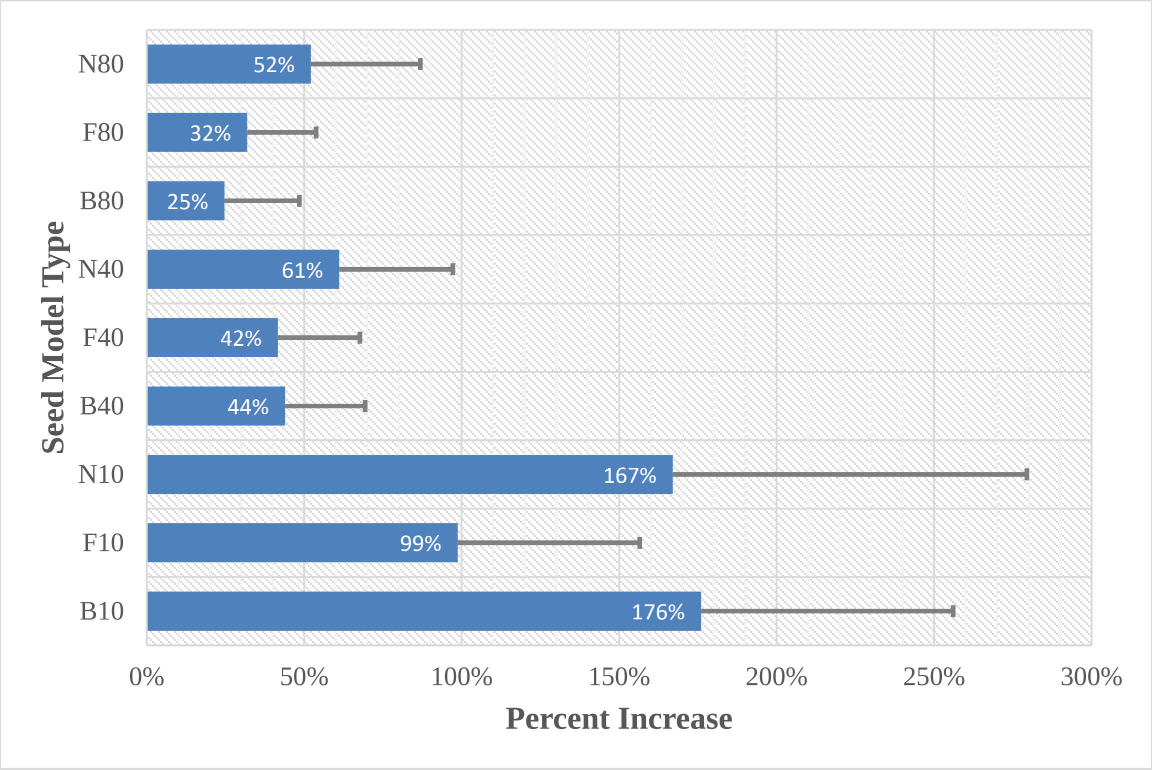


Figure 26: Increase in Functions By Model Type

With respect to chaining, at higher completion levels (40% and 80%), nucleation seed models showed a higher increase in functions compared to backward chaining models and forward chaining models. Incidentally, the latter two seem to be not significantly different at higher completion levels. However, at 10% complete level, the average increase in the forward chaining models (F10) is significantly lower compared to backward chaining and nucleation models at that level (B10 and N10 respectively). Overall, the analysis of increase in functions shows that, on average, nucleation yields more functions at higher completion levels; at 10% completion nucleation is comparable to backward chaining, and better than forward chaining.

Following the analysis of functions, the flows added to the models were analyzed. Figure 27 shows the mean increase in flows for each seed model type.

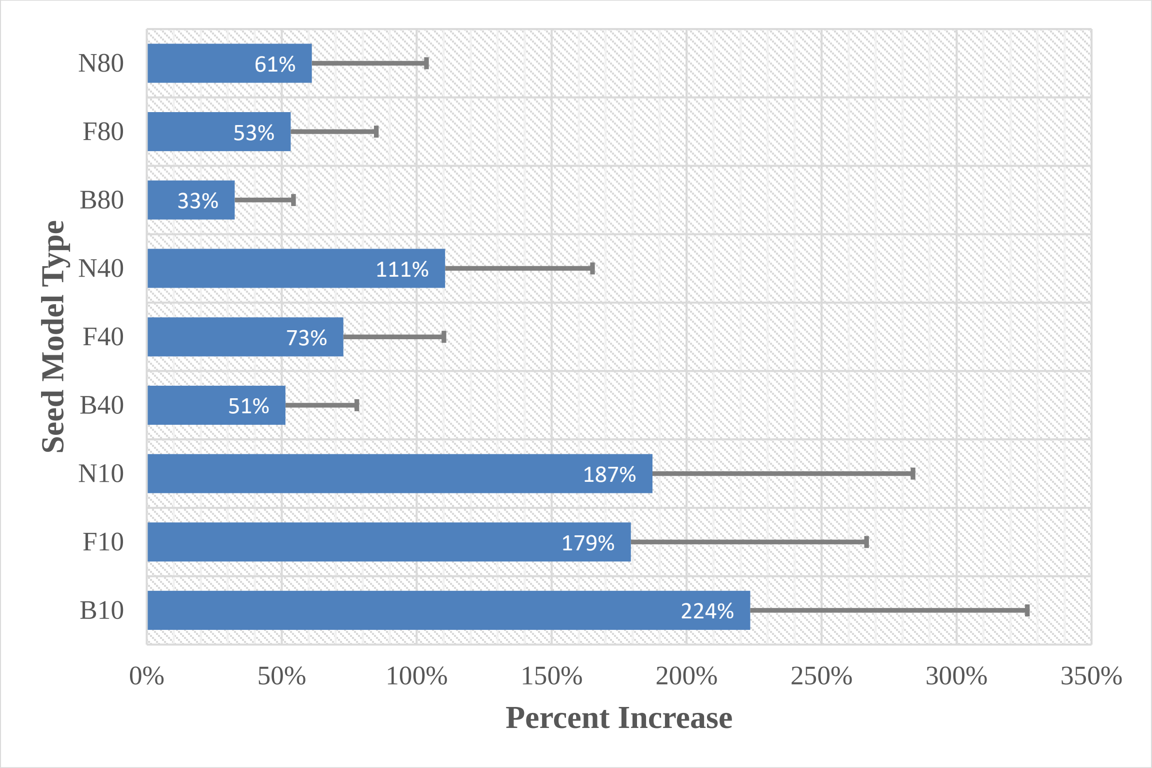


Figure 27: Increase in Flows by Model Type

Similar to functions, flows show a significantly higher percent increase in 10% complete models, when compared to the 40% complete and 80% complete models (shown in Figure 27). However, unlike functions, the 40% models and 80% models show a nonnegligible difference. Additionally, at the higher completion levels (40% and 80%), the difference between the three chaining methods is highlighted more in flows compared to functions. However, nucleation still yields the largest increase in these cases. At the 10% completion level, nucleation and forward chaining yielded similar increase in flows, with both being lower compared to backward chaining. Although, it should be noted that these differences between chaining methods are less drastic compared to those in functions.

The comparison of mean increase in function and flows suggests that in general, using a nucleation seed generated more functions and flows, if not the most. In order to determine statistical significance in these differences, a set of t-tests was conducted using an assumption of unequal variances. A mean difference of zero was used as the null hypothesis, and the results are summarized Table 18, where zeros indicate failure to reject the null hypothesis, and ones indicates successful rejection of null hypothesis.

Table 18: t-Test Results for Model Based Comparison

	Functions									Flows								
	B10	B40	B80	F10	F40	F80	N10	N40	N80	B10	B40	B80	F10	F40	F80	N10	N40	N80
B10	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1
B40	1	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	1	0
B80	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1	1	1
F10	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1
F40	1	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	1	0
F80	1	0	0	1	0	0	1	1	1	1	0	1	1	0	0	1	1	0
N10	0	1	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1
N40	1	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	0	1
N80	1	0	1	1	0	1	1	0	0	1	0	1	1	0	0	1	1	0

The increase in function and flows for 10% complete models are significantly different from those for 40% and 80% complete models, which was anticipated from the comparison of means shown in Figure 26 and Figure 27. Additionally, for functions at 80% completion level, nucleation models (N80) were found to have a significantly larger increase than backward chaining (B80) and forward chaining models (F80). However, at 40% completion level, no significant differences were found between the chaining methods; and at 10% completion levels, nucleation (N10) was found to be similar to backward chaining (B10). Interestingly, F10 was found to be significantly different from all other models.

As for flows, nucleation at 80% complete (N80) was found to yield significantly higher increases compared to backward chaining (B80), but similar to forward chaining (F80). In the case of 40% completion level, nucleation (N40) was found to have significantly higher increases than forward (F40) and backward chaining (B40). At 10% completion, no significantly differences were found between the chaining methods with respect to increase in flows. **Overall, nucleation was shown to have similar or significantly higher increases in function and flows.**

5.2 Change in Model Complexity

After comparing chaining methods based on increase in functions and flows, a follow up analysis was done comparing the chaining methods using the change in model complexity. First, the two design problems were compared based on change in model complexity.

5.2.1 Comparison of Design Problems

Change in model complexity from the two different design problems was compared to investigate any differences that exist between the design problems with respect to change in complexity between the given model and the final model. This test was conducted to ensure that the two design problems yield similar change in complexity. As previously mentioned, the design problems were proven to be similar based on increase in functions and flows, however, that conclusion is not applicable in the context of model complexity.

The complete set of results obtained for each problem was compared using a two-sample t-test with a null hypothesis assuming no difference in mean distances. Each of the distance metric was tested, and the results are shown in Table 19.

Table 19: Comparison of Design Problems Based on Change in Model Complexity

Case	**Problem A**		**Problem B**		**p-value**
	Mean	*Variance*	*Mean*	*Variance*	
Cosine	0.178	0.022	0.140	0.019	0.073
Euclidian	401.3	126600	636.3	660300	0.011
Hamming	0.871	0.008	0.835	0.029	0.073
Chebychev	310.3	80600	486.2	396500	0.014

The two problems were found to be different with more than 95% confidence level for Euclidian and Chebychev distances. Additionally, the Cosine and Hamming distance also showed a suggestive different with more than 90% confidence level. This result was unexpected as the design problem were created to be similar. However, the seed models of problem B, especially the 40% complete and 80% complete model, had more functions and flows compared to those for problem A, resulting in a higher potential for change in complexity. In order to further understand the difference between the design problems (with respect to change in model complexity), the complete set of models was divided into groups based on the chaining methods, and levels of completion. The two design problems were compared again to investigate the source of difference in complexity distances. Table 20 shows the t-test results for comparison based on level of completion.

Table 20: Comparison of Problems Based on Level of Completion of Seed Model

Case		**Problem A**		**Problem B**		**p-value**
		Mean	*Variance*	*Mean*	*Variance*	
Chebychev	*10 Percent*	205.6	32600	199.0	54000	0.901
	40 Percent	310.5	60800	445.3	162500	0.120
	80 Percent	421.5	131500	794.5	782100	0.031
Cosine	*10 Percent*	0.289	0.028	0.270	0.019	0.619
	40 Percent	0.150	0.008	0.088	0.004	0.003
	80 Percent	0.086	0.007	0.068	0.010	0.425
Euclidian	*10 Percent*	269.4	48600	268.5	85500	0.989
	40 Percent	397.9	96900	577.4	263900	0.104
	80 Percent	545.0	206300	1037.2	1311900	0.027
Hamming	*10 Percent*	0.892	0.007	0.901	0.006	0.669
	40 Percent	0.869	0.008	0.815	0.029	0.129
	80 Percent	0.849	0.008	0.791	0.046	0.163

Models generated from 80% complete seed models were found to have significant differences between the two problems, using Chebychev and Euclidian distances. This aligns with the expectations stated previously. Moreover, the 40% complete models were found to have a difference between the two problems using cosine distance, whereas those using Euclidian distance and Chebychev distance had p-values just above 0.1, suggesting that the difference between them is small. Additionally, Hamming distance found no significantly differences between the two problems. Next, Table 21 shows t-test results for comparison based on chaining methods.

Table 21: Comparison of Problems Based on Chaining Method of Seed Model

Case		**Problem A**		**Problem B**		**p-value**
		Mean	*Variance*	*Mean*	*Variance*	
Chebychev	*Backward*	174.8	11400	226.0	55300	0.279
	Forward	423.3	96300	732.3	693500	0.053
	Nucleation	345.7	113500	476.4	304500	0.267
Cosine	*Backward*	0.098	0.010	0.132	0.019	0.269
	Forward	0.227	0.029	0.156	0.013	0.055
	Nucleation	0.218	0.017	0.132	0.026	0.025
Euclidian	*Backward*	232.3	17300	305.7	90700	0.225
	Forward	541.2	150500	941.0	1155100	0.052
	Nucleation	446.6	180800	632.1	520400	0.224
Hamming	*Backward*	0.906	0.006	0.809	0.058	0.044
	Forward	0.827	0.006	0.840	0.025	0.658
	Nucleation	0.878	0.009	0.853	0.007	0.289

As shown in Table 21, the two design problems were found to be significantly different in the case of Cosine distance using Nucleation, and Hamming distance using Backward chaining. Additionally, a suggestive difference was found between problems in Chebychev distance with forward chaining, Cosine distance with forward chaining, and Euclidian distance with forward chaining. It should be noted that these three cases of forward chaining were close of 95% confidence level, suggesting that the forward chaining seed models are likely to result in different complexity distances between the two problems. Ultimately, only 5 out of 24 comparisons showed significant differences, and as a result, the following comparison of chaining methods was done using the combined set of problem A and problem B.

5.2.2 Comparison of Complexity Distances Based on Seed Models

Following the comparison of the design problems, the changes in model complexity, or complexity distances, were analyzed based on the different seed models provided to the participants. Figure 28 shows the mean complexity distances for each chaining method, and each level of completion.

5.2.2.1 Cosine Distance

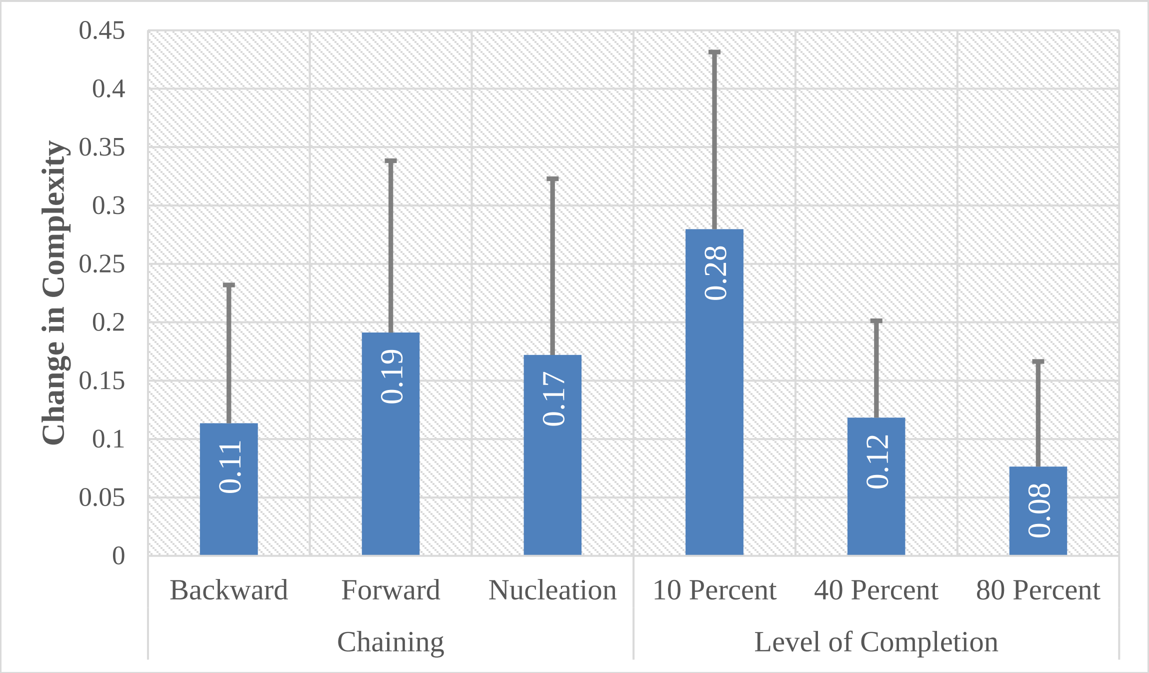


Figure 28: Cosine Distance Based on Chaining Method and Level of Completion

As expected, the complexity distances for 10% complete models were larger than those in higher levels of completeness (40% and 80%). Additionally, forward chaining and nucleation models show similar mean complexity distance, whereas, backward chaining had a smaller complexity distance in comparison. In order to identify any statistical significance, a set of t-tests were performed to do a pairwise comparison of the different model types using cosine distance to measure change in complexity. Results are shown in Table 22. Instances where the two-sample t-test was able to show significant difference are marked as “1” and cases where the null hypothesis could not be rejected are marked as “1.”

It should be noted that the table is diagonally symmetric so the lower triangle has been omitted. This is also the case for Table 23, Table 24, Table 25, and Table 27.

Table 22: Pairwise Comparison of Seed Models based on Cosine Distance

	B10	**B40**	**B80**	**F10**	**F40**	**F80**	**N10**	**N40**	**N80**
B10	X	1	1	1	1	1	0	1	1
B40		X	1	1	1	0	1	1	0
B80			X	1	1	1	1	1	0
F10				X	1	1	0	1	1
F40					X	1	1	0	0
F80						X	1	0	0
N10							X	1	1
N40								X	0
N80									X

From Table 22, we can observe that the B10 and N10 were found to be similar, and F10 and N10 were found to be similar, however no direct similarity was found between B10 and F10. As for the 40% complete models, B40 was found to be similar to F80 and N80; F40 was found to be similar to N40, and N80; finally, N40 was found to be similar to F40, F80, and N80. In comparison to the 10% complete models, the 40% complete models were shown to have similarities with 80% complete models. In case of the 80 percent complete models, B80 was found similar to N80; F80 was found similar to B40, N40, and N80; and finally, N80 was found similar to B40, B80, F40, F80, and N40. Overall, the backward chaining models tend to be different from forward chaining models and nucleation models at lower completions levels. This can be observed in Table 21 with a clustering of zeros in the bottom right corner.

5.2.2.2 Euclidian Distance

Similar to Cosine distance, the model variants were compared with respect to change in model complexity using Euclidian distance. Figure 29 shows the mean complexity distance for each chaining method and level of completeness.

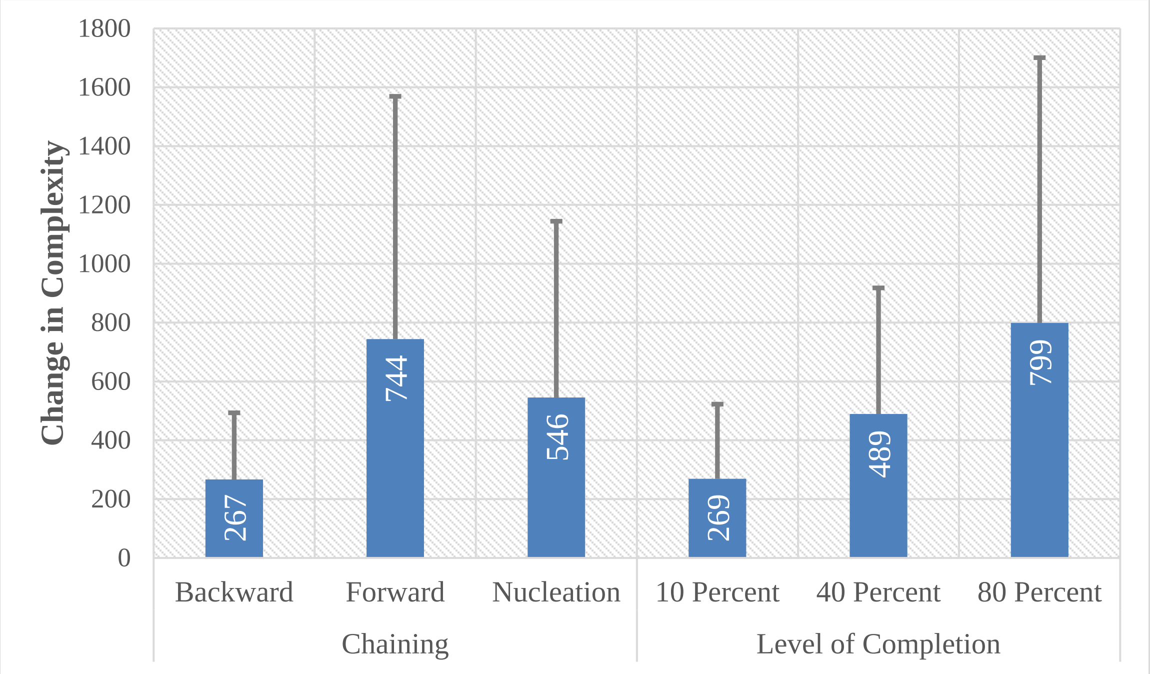


Figure 29: Euclidian Distance Based on Chaining Method and Level of Completion

Unlike cosine distance, the complexity distances of the levels of completeness were opposite to what was expected; 10% complete models showed the smallest complexity distance, followed by 40% models, and finally 80% models with the largest complexity distances. For chaining methods, forward chaining seems to show the highest complexity distance, followed by nucleation, and finally backward chaining. In order to determine significance of these differences, Table 23 shows the results of a set of two-sample t-tests conducted to compare each model variant to all other model variants.

Table 23: Pairwise Comparison of Seed Models Based on Euclidian Distance

	B10	**B40**	**B80**	**F10**	**F40**	**F80**	**N10**	**N40**	**N80**
B10	X	0	0	0	1	1	0	1	1
B40		X	0	0	1	1	0	1	1
B80			X	0	1	1	0	1	1
F10				X	1	1	0	0	1
F40					X	1	1	0	0
F80						X	1	1	0
N10							X	1	1
N40								X	0
N80									X

As shown in Table 23, the results for Euclidian distance show different trends compared to cosine distance. In this case, all of backward chaining models were found to be similar to each other, as well as F10 and N10. F40 was found to be similar to N40 and N80, whereas N40 was found similar to F10, F40, and N80. For the 80 percent complete models, F80 was only found similar to N80, however, N80 was found similar to F40, F80, and N40. Overall, the t-test results suggest that based on Euclidian distance, backward chaining, regardless of the level of completion, was within the same group as 10 percent complete models for forward chaining and nucleation. Additionally, forward chaining models at higher completion levels were found to be more disjoined, having only one or two similarities. Nucleation at higher levels of completion, on the other hand, shared more similarity with forward chaining, as well as other level of completions of nucleation.

5.2.2.3 Hamming Distance

Next, the seed models were compared based on Hamming distance between graph complexity of the given partially completed models, and final model. Figure 30 shows the change in complexity based on hamming distance, grouped by the chaining methods and levels of completion. It should be noted that the vertical axis scale in this graph does not start at zero; this was done to highlight the small differences between the bars.

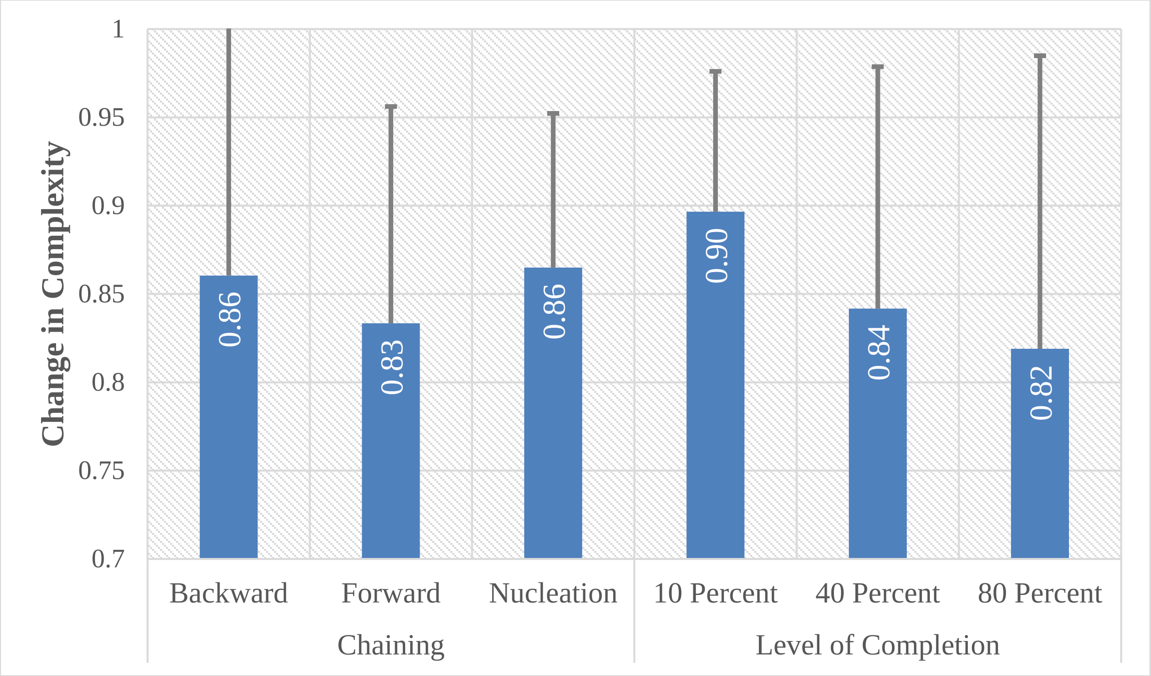


Figure 30: Hamming Distance Based on Chaining Method and Level of Completion

As seen in Figure 30, the Hamming distance from initial complexity vector to the final complexity vector does not highlight any large differences between the chaining methods, or levels of completion. This may, in part, be inherent to the Hamming distance metric as it compares vectors based on the percentage of values that are different. As functions and flows are added to the initial Function Structure model, most (if not all) elements of the complexity vector are expected to change regardless of the type of chaining method used or level of completion used. This results in the Hamming distance between vectors being closer in magnitude between the chaining methods, or levels of completion. However, small differences can still be identified, such as the descending order of distance from 10% complete models or 80% complete models. Additionally, backward chaining and nucleation were found to be similar in this case, with forward chaining showing smaller distances compared to the other two chaining methods.

With small differences in mean distance, it was necessary to conduct a set of t-test comparing each pair of seed models, and the results are shown in Table 24. Similar to previous t-tests, the null hypothesis assumed a mean difference of zero, and a 95% confidence level was required to show significant differences.

Table 24: Pairwise Comparison of Seed Models Based on Hamming Distance

	B10	**B40**	**B80**	**F10**	**F40**	**F80**	**N10**	**N40**	**N80**
B10	X	1	1	1	1	1	0	1	1
B40		X	0	0	0	0	0	0	0
B80			X	0	0	0	0	0	0
F10				X	0	1	0	0	0
F40					X	0	0	0	0
F80						X	1	0	0
N10							X	0	0
N40								X	0
N80									X

Model based comparison of Hamming distance showed that besides B10, all models were found to be part of the same group, with a few exceptions. B10 was only found similar to N10, while F10 and N10 were found similar to all models except F80. Tests comparing the remaining models failed to reject the null hypotheses. Therefore, when using the Hamming distance metric, all nucleation models were found to be similar to each other, and higher levels of completion of forward chaining and backward chaining were all found to be similar to one another, as well as nucleation (with the aforementioned exception of F80 and N10). In this case, hamming distance provides little insight about the differences that may exist between the chaining methods, however, it shows that starting a model with backward chaining or nucleation may increase the model complexity more so than forward chaining.

5.2.2.4 Chebychev Distance

Lastly, the seed models were compared based on Chebychev distance, and the results are shown in Figure 31.

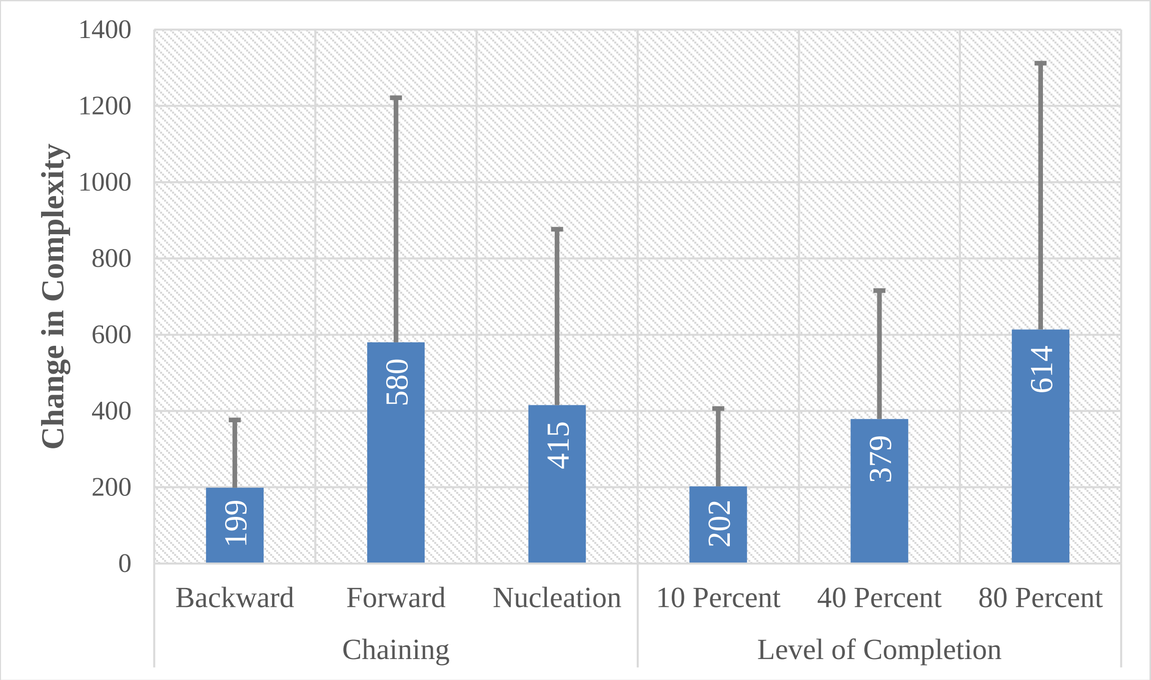


Figure 31: Chebychev Distance Based on Chaining Method and Level of Completion

The patterns for Chebychev distance are similar to those in Euclidian distance; with increase distance from 10% complete models to 80% complete models, and forward chaining showing the highest distance. Similar to previous distance metrics, a set of t-tests was conducted to investigate significance of differences between the different model types. The results of pairwise t-tests are shown in Table 25.

Table 25: Pairwise Comparison of Seed Models Based on Chebychev Distance

	B10	**B40**	**B80**	**F10**	**F40**	**F80**	**N10**	**N40**	**N80**
B10	X	0	0	0	1	1	0	1	1
B40		X	0	0	1	1	0	1	1
B80			X	0	1	1	0	1	1
F10				X	1	1	0	0	1
F40					X	1	1	0	0
F80						X	1	1	0
N10							X	1	1
N40								X	0
N80									X

Comparing Table 23 and Table 25, the t-tests results are identical, suggesting that comparison of chaining methods based on Euclidian distance and Chebychev distance are equivalent. Backward chaining is found to be different from forward chaining and nucleation at higher completion levels, with forward chaining and nucleation being similar to each other in all cases (10%, 40% and 80% completion levels).

The comparison of change in model complexity shows that in for 3 out of 4 distance metrics, the level of completion in the initial model has an impact on the change in model complexity. Additionally, backward chaining models were found to be significantly different from forward chaining and nucleation models. Finally, **forward chaining and nucleation models were found to generate larger changes in model complexity compared to backward chaining**.

5.3 Model Evaluation

In addition to the comparison of increased of function and flows in seed model, and the change in model complexity, the final model was also evaluated for quality of the model itself. The average evaluation scores for each seed model type are shown in Figure 32.

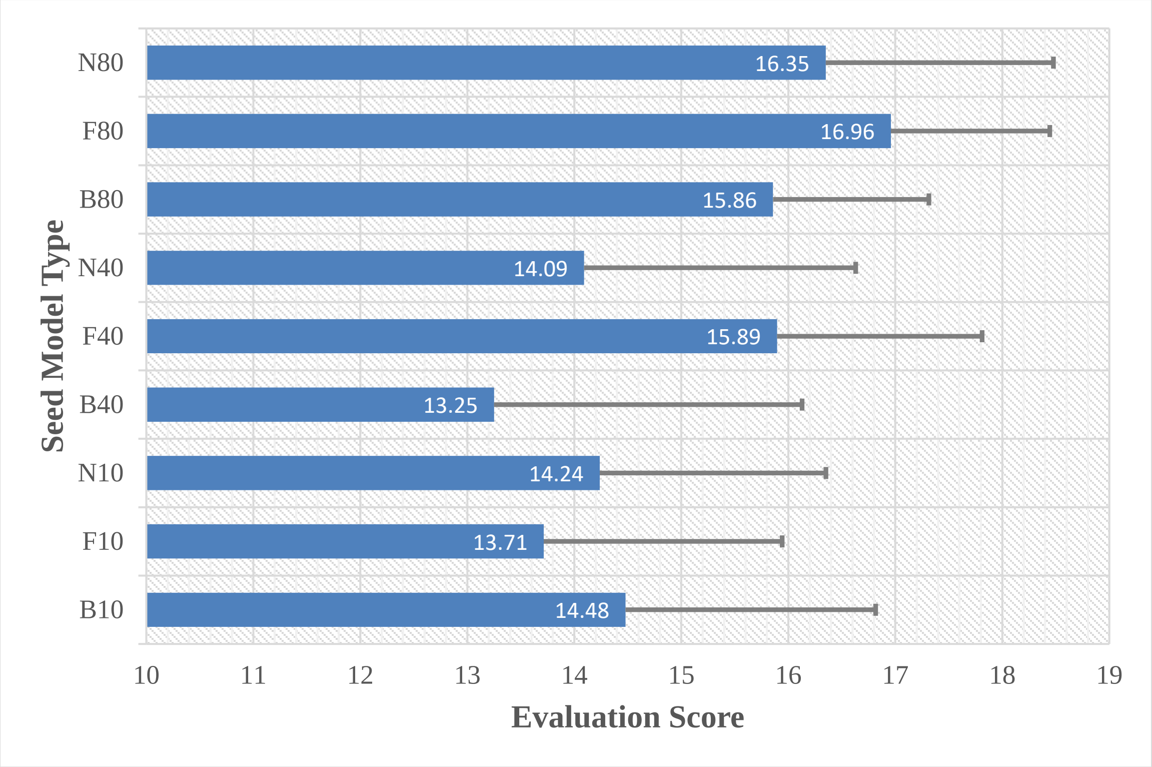


Figure 32: Evaluation Scores by Model Type

The evaluations of finals models were not expected to yield significant differences between the model types because of the nature of the scoring rubric. However, the mean scores for 10% complete models are lower than those of 80% complete models. In order to identify any significant differences between the nine groups, a single factor ANOVA was performed. Results for the ANOVA are shown in Table 26.

Table 26: Single Factor ANOVA Comparing Model Types Based on Model Evaluation Scores

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	291.72	8	36.465	7.463	1.37E-08	1.990
Within Groups	889.27	182	4.886			
Total	1180.99	190				

The ANOVA results indicate that the different model types tested are not all from the same probability distribution function (small P-value, and F-crit < F). Therefore, further analysis in the form of a set of two-sample t-tests was conducted to identify the model types that resulted in significantly different evaluation scores for the final models. An assumption of unequal variances was used, and a null hypothesis of zero mean difference was tested. Results for the complete set of t-tests are shown in Table 27, where the values are p-values associated with each of the t-tests.

Table 27: t-tests Comparing Models Types based on Evaluation Score

	B10	F10	N10	B40	F40	N40	B80	F80	N80
B10	X	0.285	0.728	0.146	0.042	0.606	0.026	0.000	0.010
F10		X	0.450	0.578	0.002	0.616	0.001	0.000	0.001
N10			X	0.232	0.016	0.841	0.008	0.000	0.004
B40				X	0.002	0.335	0.001	0.000	0.001
F40					X	0.016	0.947	0.061	0.498
N40						X	0.010	0.000	0.004
B80							X	0.019	0.407
F80								X	0.300
N80									X

The shaded cells in Table 27 highlight test cases where the null hypothesis was successfully rejected with a 95% confidence level. Comparison of model evaluation based on percent completion shows that a significant difference exists between 10% complete seed models and other seed models with the exception of backward chaining and nucleation models at 40% completion. This is suggested in the comparison of means shown in Figure 32. This is reiterated when reviewing the 40 percent complete models; forward chaining model (F40) was found to be significantly different from the 10 percent complete models, whereas the t-tests failed to prove the same for backward chaining and nucleation models. Conversely, backward chaining and nucleation are significantly different from 80 percent complete models; however, forward chaining model is similar to the 80 percent complete models. Finally, Table 27 shows that the 80% complete models for all chaining types are significantly different from the remaining models with the exception of forward chaining model at 40%. In general, a significant difference in model quality was observed between the 10% and 80% models. This suggests that the model elements added by the participants contributed to lower scores.

Subsequently, comparing the model evaluation based on the chaining method shows that at 10% and 80% completion levels, the chaining method used does not significantly impact the evaluation score. This suggests that when majority of the functions and flows are added by students, or when a small percentage are added by students, the model scores are not impacted by the chaining method used. Conversely, at 40% completion level, **forward chaining models tend to score significantly higher than backward chaining and nucleation.** This is expected because forward chaining is generally used when teaching function modeling, and student participants should be most comfortable using forward chaining. Ultimately, analysis of model evaluation shows that the chaining method used is generally not a significant factor in the final score of the model, instead the participants experience using function modeling plays a larger role.

In order to investigate whether the presence of any seed model had an impact, the scores from the seeded experiment were compared to those from the non-seeded experiment. Table 28 shows the summary of model evaluation scores.

Table 28: Change in Model Evaluation Scores After Seeding

	Unseeded Score	Seeded Score	Score Change	Score Change w/o Blackbox
Mean	7.24	15.52	8.29	0.90
Standard Deviation	2.24	2.61	3.19	2.39
Minimum	2	9	1	-3
Maximum	10	18	15	7

The presence of the seed model improved the evaluation scores by 8.29 points on average. However, it should be noted that this improvement, in part, is due to the black-box model which was given to them in the seeded experiments. Eight out of the 18 questions in the rubric are directly related to the black-box model, which majority of the students did not draw in the unseeded experiments. Additionally, six out of 22 participants showed a negative change in the score when using a seed model, eleven participants showed a positive change, and five participants showed no change. Interestingly, the participants that showed no change between seeded and unseeded models were those with above average scores in both cases.

In addition to the change in scores after introducing a seeding model, the scores for three lowest scoring rubric questions were examined to identify whether a pattern related to chaining methods could be identified. Figure 33 shows the distribution of participant scores, based on chaining method, where "zeros" and "ones" refer to the scores allotted for each rubric question.

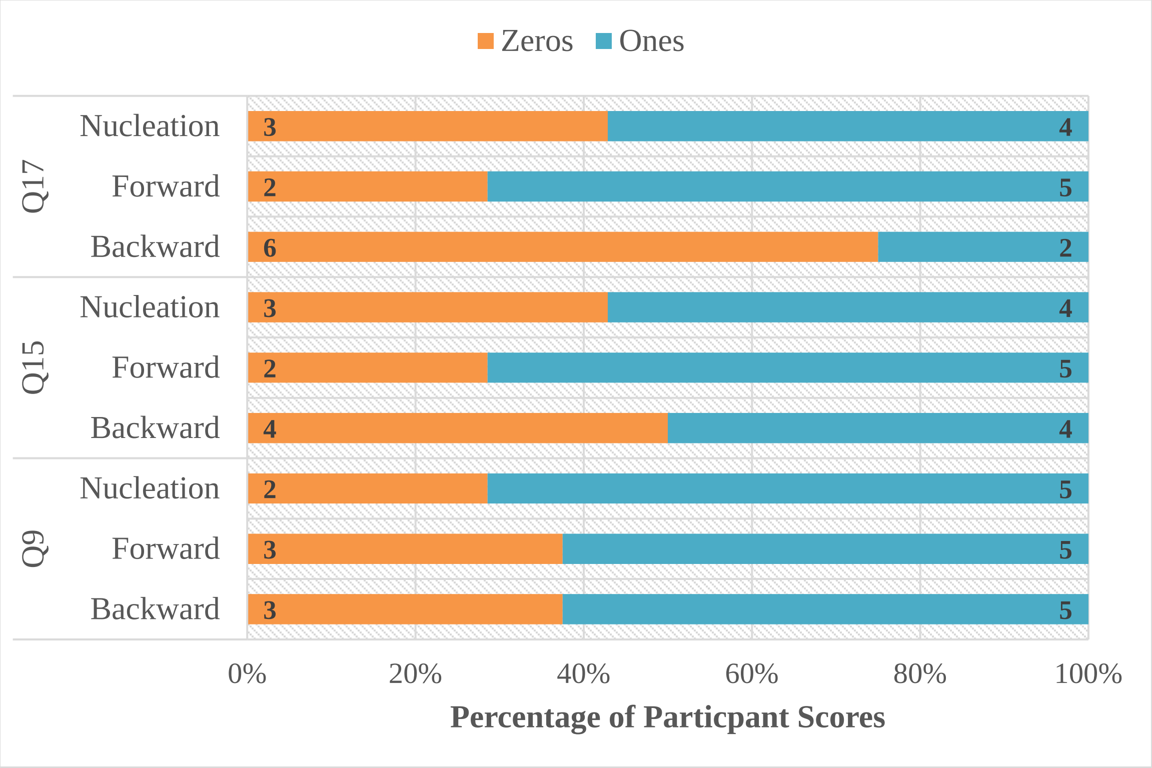


Figure 33: Score Distribution of Select Rubric Questions

As shown in Figure 33, the rubric evaluations for Q9, Q15 and Q17 (see Table 7 in Section 3.3.3 for details) show that backward chaining tends to perform worse on these questions (more zeros compared to ones). Nucleation performs better than forward chaining for Q9, and worse for Q15 and Q17. This result is within expectations as Q9 refers to overall plausibility of the product, and Q15 and Q17 focus on flow conservation. A nucleation seed model provides the most information to participants about product functionality, whereas forward chaining and backward chaining provide information about the input flows and output flows respectively. Therefore, with nucleation, participants are better able to maintain the overall product plausibility; whereas, with forward chaining, participants are better able to follow the input flows and ensure conservation across functions. However, it should be noted that, as previously mentioned, these questions were found to be the lowest scoring, suggesting these aspects of the function model are not well understood or materialized by the participants. A targeted study should be conducted with a larger participant pool to more certainly understand the effects of chaining methods or model plausibility and flow conversation.

Chapter Six
CONCLUSIONS

The goal of this research was to investigate the impact of chaining methods on Function Structure model generated by designers. This was done by prompting designers with a seed model generated using one of three chaining methods, up to one of three completion levels. A protocol study and a designer study were conducted where participants were asked to complete a given (partially completed) Function Structure models. The collected models were then analyzed based on increased in functions and flows, change in model complexity, and model evaluation.

- **Models from designer study and protocol study are comparable.**

A review of the Function Structures from both experiments showed that models collected from the designer study and protocol study are comparable. The increase in function and flows between designer study and protocol study were comparable. No significant differences were found when the experiments were compared based on the chaining methods. Additionally, increase in functions was found similar between designer study and protocol study when compared based on level of completion; increase in flows was also found similar expect 40% completion level, where designer study showed a higher increase.

The complete set of models obtained from the two studies showed no statistical differences based on complexity distances (using cosine, Euclidian, hamming, and Chebychev distances). When comparing by chaining method, no significant differences were found, however, comparison by level of completion showed that 80% complete models showed a significant difference in cosine distance. Finally, statistical analysis showed no significant differences in the evaluation score of models between the two experiment types when compared by chaining method. However, significant differences were found when comparing 80% complete models. Overall, the analysis suggests that trends observed in the protocol study can be ascribed to the designer study.

- **Nucleation showed comparable or higher increase in functions and flows compared to forward chaining and backward chaining.**

Function Structure models were compared based on the type of seed model in order to identify similarities and differences between the three chaining methods. Analysis of increase in functions and flows showed that, in general, nucleation yields similar or significantly higher increases in function and flows. At lower completion levels, nucleation had comparable increase in function and flows, compared to backward chaining and forward chaining; however, at higher levels of completion, nucleation was found to have significantly higher increase in functions and flows.

- **Nucleation showed comparable change in model complexity compared to forward chaining and backward chaining.**

Moreover, a review of change in model complexity using cosine distance, Euclidian distance, and Chebychev distance shows that nucleation and forward chaining showed similar change in model complexity, which was significantly higher than backward chaining. When using hamming distance, backward chaining and nucleation showed significantly higher change in model complexity compared to forward chaining. Therefore, in all cases, nucleation was shown to have a similar or higher increase in model complexity.

- **Seed models generally improve scores; however, the effects of chaining methods need to be further studied.**

Finally, comparison of model evaluation scores showed that at 40% completion level, forward chaining scored significantly higher than other chaining methods, however, at 10% and 80% completion level, the chaining method has no significant impact on the model scores. Additionally, the comparison between seeded and unseeded models showed that existence of the seed model improves the evaluation scores in general. Moreover, analysis of individual rubric questions suggests that nucleation assists designers in ensuring that the Function Structure model represents a plausible product, whereas forward chaining and backward chaining are more useful for maintaining flow conservation.

In summary, the observations and conclusions from this research suggest that construction of Function Structure models using nucleation may be more effective when compared to the traditional method, forward chaining. Nucleation was showed to generate more functions and flows, as well as more complex model topologies.

Chapter Seven
FUTURE WORK: POTENTIAL NEXT STEPS

As identified in Chapter 2, work performed in this research is a narrow slice of the larger research topic. Expectedly, various new research avenues were identified as a result of this research. New research questions can be identified in the areas explored in this thesis, specifically with respect to textual analysis of the final models, with respect to change in model complexity, and with respect to alternative model evaluation methods. Additionally, the videos collected from the protocol study can also be analyzed to gain a deeper understanding about modeling activities, specifically pause patterns.

7.1 Analysis of Function and Flow Labels

The Function Structure models collected in the experiments contained functions (blocks), function labels (block text), flows (edges), and flow labels (edge text). Analysis of increase in functions and flows presented in chapter 5 only refers to the blocks and edges. As such, this analysis was focused on topological changes in the models. However, that is only one part of the information contained in the Function Structure models. The collected models can be analyzed based on the function and flow labels to understand the increase in function and flows from a different perspective. The following research questions can be investigated.

- How similar are the function labels between different chaining methods?
- How similar are the flow labels between different chaining methods?

- Can the Functional Basis Vocabulary be used to compare the chaining methods?
- How different are the chaining methods with respect to the number of distinct function labels and flow labels?

7.2 Analysis of Change in Model Complexity

As previously mentioned, the complexity vector generated for each Function Structure model consists of 29 different complexity metrics. The analysis of model complexity discussed in this thesis use vector distance metrics to calculate distances between the complexity vectors to investigate change in model complexity. However, this does not account for the nature of the complexity vectors, more specifically the scale of individual complexity metrics within the complexity vector. Certain complexity metrics are expected to have values in the scale of 1000, whereas some complexity metrics are expected to have values significantly smaller than 1. As a result, distance metrics comparing entire complexity vectors are susceptible to the effects of enormous differences between the values of constituent metrics. Therefore, the complexity data can be analyzed differently by comparing the change in each individual complexity metric instead of the change in entire vector. The following research questions can be investigated in that regard.

- How do individual complexity metrics change from given model to final model?
- How does the change in individual complexity metrics differ between the three levels of completion?

- How are the individual complexity metrics different between the chaining methods?
- How does the change in individual complexity metrics differ between the chaining methods?

Additionally, it can also be investigated whether the complexity vectors are additive; does the sum of given complexity and added complexity add to the final complexity of the model? This is a relevant discussion because the basis of comparing the chaining methods in this research has been the change in model complexity. Although this a plausible comparison metric for model complexity, the underlying behavior of model complexity is not understood from a mathematical operations perspective. For example, let's assume two bipartite graphs exist: A and B. Complexity vectors for each graph can be calculated separately. Next, a third bipartite graph, C, is defined as the union of A and B. In this scenario, does the complexity vector of C equal to the sum of complexity vectors for A and B?

7.3 Evaluation of Function Structure models

The models in this research were evaluated using an externally developed rubric, intended for scoring student generated Function Structure models [33,66]. The comparisons between level of completions, and those between chaining methods, used only the final score for each model. These models can be further analyzed by comparing the performance of the models on each question of the rubric. Additionally, alternative model evaluation methods can also be used to compare the models.

7.4 Evaluation of the Modeling Activity

In addition to three aspects investigated in this research, the modeling activity can be analyzed to identify any patterns that may exist. The video coding (shown in Section 2.1.2) of the designers' modeling activity can be reviewed and analyzed for pause patterns, element sequences, addition and deletion chunks, edit frequency, etc. The following research questions can be potentially investigated.

- What patterns exist in modeling activity with respect to pause length and frequency?
- What patterns exist in modeling activity with respect to the type of activity followed by a pause?
- What patterns exist in modeling activity with respect to the number of elements added, deleted, or edited after a pause?
- What patterns exist in modeling activity with respect to the clustering of elements between pauses?
- How does presence of a seed model affect pause patterns?
- How are pause patterns different based on the type of chaining method used for the seed model?

REFERENCES

[1] Dieter, G. E., and Schmidt, L. C., 2013, *Engineering Design*, McGraw Hill, New York.

[2] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H., 2007, *Engineering Design*, Springer London, London.

[3] Ullman, D. G., 1992, *The Mechanical Design Process*, McGraw-Hill, New York, NY.

[4] Dym, C. L., and Little, P., 1999, *Engineering Design: A Project-Based Introduction*, John Wiley and Sons, New York, NY.

[5] Hyman, B., 1998, *Fundamentals of Engineering Design*, Prentice Hall, Upper Saddle River, NJ.

[6] Otto, K., and Wood, K., 2001, *Product Design Techniques in Reverse Engineering and New Product Development*, Prentice Hall, Upper Saddle River, NJ.

[7] Ulrich, K. T., and Eppinger, S. D., 2008, *Product Design and Development*, McGraw-Hill, New York, USA.

[8] Yang, M. C., 2003, “Concept Generation And Sketching: Correlations With Design Outcome,” Proc. DETC 2003, p. 6.

[9] SHAH, J. J., VARGAS-HERNANDEZ, N., 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.

[10] Shah, J., Vargas-Hernandez, N., and Smith, S., 2003, “Metrics for Measuring Ideation Effectiveness,” Des. Stud., **24**(2), pp. 111–134.

[11] Booth, J. W., Reid, T. N., Eckert, C., and Ramani, K., 2015, “Comparing Functional Analysis Methods for Product Dissection Tasks,” J. Mech. Des., **137**(8), p. 81101.

[12] Takeda, H., Yoshioka, M., and Shimomura, Y., 1994, “Analysis of Design Processes by Function , Behavior and Structure,” Delft Protoc. Work. Conf. Proc.

[13] Umeda, Y., Kondoh, S., Shimomura, Y., and Tomiyama, T., 2005, “Development of Design Methodology for Upgradable Products Based on Function–behavior–state Modeling,” AI EDAM (Artificial Intell. Eng. Des. Anal. Manuf., **19**(3), pp. 161–182.

[14] Hirtz, J., Stone, R. B., and McAdams, D. a., 2002, “A Functional Basis for Engineering Deisgn : Reconciling and Evolving Previous Efforts,” Res. Eng. Des., **13**, pp. 65–82.

[15] Gero, J. S., and Kannengiesser, U., 2007, “A Function–behavior–structure Ontology of Processes,” AI EDAM (Artificial Intell. Eng. Des. Anal. Manuf., **21**(4), pp. 1–34.

[16] Qian, L., and Gero, J. S., 1996, “Function–behavior–structure Paths and Their Role in Analogy-Based Design,” Artif. Intell. Eng. Des. Anal. Manuf., **10**(4), p. 289.

[17] Goel, A. K., Rugaber, S., and Vattam, S., 2009, “Structure, Behavior, and Function of Complex Systems: The Structure, Behavior, and Function Modeling Language,” AI EDAM (Artificial Intell. Eng. Des. Anal. Manuf., **23**(1), p. 23.

[18] Lin, Y., and Zhang, W. J., 2004, “Towards a Novel Interface Design Framework: Function-Behavior-State Paradigm,” Int. J. Hum. Comput. Stud., **61**(3), pp. 259–297.

[19] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., 1996, “Supporting Conceptual Design Based on the Function-Behavior-State Modeler,” Artif. Intell. Eng. Des. Anal. Manuf., **10**(4), p. 275.

[20] Albers, A., Alink, T., Matthiesen, S., and Thau, S., 2008, “Support of System Analyses and Improvement in Industrial Design Trough the Contact & Channel Model,” 10th Int. Des. Conf. Des. 2008, **Dubrovnik**, pp. 245–252.

[21] Albers, A., Ohmer, M., and Eckert, C., 2004, “Engineering Design in a Different Way: Cognitive Perspective on the Contact and Channel Model Approach,” *Visual and Spatial Reasoning in Design*, J.S. Gero, B. Tversky, and T. Knight, eds., Key Centre of Design Computing and Cognition, Cambridge, MA, USA.

[22] Ramachandran, R., Caldwell, B. W., and Mocko, G. M., 2011, “A User Study to Evaluate the Function Model and Function Interaction Model for Concept Generation,” *International Design Engineering Technical Conferences*, pp. 273–284.

[23] Freeman, P., and Newell, A., 1971, “A Model for Functional Reasoning in Design,” *Second International Joint Conference on Artificial Intelligence*.

[24] McAdams, D. a., Stone, R. B., and Wood, K. L., 1999, “Functional Interdependence and Product Similarity Based on Customer Needs,” Res. Eng. Des. - Theory, Appl. Concurr. Eng., **11**(1), pp. 1–19.

[25] Kurfman, M. a, Stone, R. B., VanWie, M., Wood, K. L., Otto, K. N., Wie, M. Van, Wood, K. L., and Otto, K. N., 2000, "Theoretical Underpinnings of Functional Modeling: Preliminary Experimental Studies," 12th Int. Conf. Des. Theory Methodol., p. DETC2000/DTM-14563.

[26] Bryant, C. R., Stone, R. B., McAdams, D. A., Kurtoglu, T., and Campbell, M. I., 2005, "Concept Generation From The Functional Basis of Design," *International Conference on Engineering Design*, Melbourne, Austrailia.

[27] SRIDHARAN, P., and CAMPBELL, M. I., 2005, "A Study on the Grammatical Construction of Function Structures," Artif. Intell. Eng. Des. Anal. Manuf., **19**(3), pp. 139–160.

[28] Caldwell, B. W., Sen, C., Mocko, G. M., Summers, J. D., and Fadel, G. M., 2008, "Empirical Examination of the Functional Basis and Design Repository," *Design Computing and Cognition '08*, Springer Netherlands, Dordrecht, pp. 261–280.

[29] Thomas, J., Summers, J. D., Sen, C., Mocko, G. M., Fadel, G. M., Summers, J. D., and Fadel, G. M., 2009, "Investigation of the Interpretability of Three Function Structure Representations: A User Study," Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf., pp. 1–14.

[30] Schultz, J., Mathieson, J. L., Sen, C., Summers, J. D., Caldwell, B. W., Mocko, G., Mathieson, J. L., Summers, J. D., and Mocko, G., 2010, "Limitations to Function Structures: A Case Study in Morphing Airfoil Design," *ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, ASME, Montreal, Canada, p. DETC2010-28559.

[31] Caldwell, B. W., Thomas, J., Sen, C., Mocko, G. M., and Summers, J. D., 2012, "The Effects of Language and Pruning on Function Structure Interpretability," J. Mech. Des., **134**(6), p. 61001.

[32] Sen, C., and Summers, J. D., 2012, "A Pilot Protocol Study on How Designers Construct Function Structures in Novel Design," 5th Int. Conf. Des. Comput. Cogn., p. No. 37.

[33] Nagel, R. L., Bohm, M. R., and Linsey, J. S., 2013, "An Investigation Into the Effectiveness of an Algorithmic Approach to Teaching Function Modeling," Proc. ASME 2013 Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf., pp. 1–17.

[34] Tomiyama, T., van Beek, T. J., Komoto, H., and D'Amelio, V., 2013, "Making Function Modeling Practically Usable," AI EDAM (Artificial Intell. Eng. Des. Anal. Manuf., **27**(3), pp. 301–309.

[35] Eisenbart, B., Gericke, K., Blessing, L. T. M., and McAloone, T. C., 2017, "A

DSM-Based Framework for Integrated Function Modelling: Concept, Application and Evaluation," Res. Eng. Des., **28**(1), pp. 25–51.

[36] Gill, A. S., Summers, J. D., and Turner, C. J., 2017, "Comparing 'Function-Structures' and 'Pruned Function-Structures' for Market Price Prediction: An Approach to Benchmarking Representation Inferencing Value," Artif. Intell. Eng. Des. Anal. Manuf., **in press**, pp. 550–566.

[37] Mokhtarian, H., Coatanéa, E., and Paris, H., 2017, "Function Modeling Combined with Physics-Based Reasoning for Assessing Design Options and Supporting Innovative Ideation," Artif. Intell. Eng. Des. Anal. Manuf., **31**(4), pp. 476–500.

[38] Gericke, K., and Eisenbart, B., 2017, "The Integrated Function Modelling Framework and Its Relation to Function Structures," Artif. Intell. Eng. Des. Anal. Manuf., **31**(4), p. in press.

[39] Thiagarajan, A., Patel, A., O'Shields, S., and Summers, J. D., 2017, "Functional Thinking: A Protocol Study to Map Modeling Behavior of Designers," *Design Computing and Cognition '16*, pp. 339–357.

[40] Thiagarajan, A., 2015, "Function Thinking: A Protocol Study," All Thesis Clemson Univ., **1**(August).

[41] O'Shields, S. T., 2016, "Design Collaboration in Industry: WHen, Why, and How," Clemson University.

[42] Hegarty, M., and Just, M. A., 1993, "Constructing Mental Models of Machines from Text and Diagrams," J. Mem. Lang., **32**(6), pp. 717–742.

[43] Mani, K., and Johnson-Laird, P. N., 1982, "The Mental Representation of Spatial Descriptions.," Mem. Cognit., **10**(2), pp. 181–187.

[44] Forbus, K. D., Gentner, D., and Law, K., 1995, "MAC/FAC: A Model of Similarity-Based Retrieval," Cogn. Sci., **19**(2), pp. 141–205.

[45] Glenberg, A. M., and Langston, W. E., 1992, "Comprehension of Illustrated Text: Pictures Help to Build Mental Models," J. Mem. Lang., **31**(2), pp. 129–151.

[46] Linsey, J. S., Clauss, E. F., Kurtoglu, T., Murphy, J. T., Wood, K. L., and Markman, a. B., 2011, "An Experimental Study of Group Idea Generation Techniques: Understanding the Roles of Idea Representation and Viewing Methods," J. Mech. Des., **133**(3), p. 31008.

[47] Kohn, N. W., and Smith, S. M., 2011, "Collaborative Fixation: Effects of Others' Ideas on Brainstorming," Appl. Cogn. Psychol., **25**(3), pp. 359–371.

[48] Shah, J., 1998, “Experimental Investigation of Progressive Idea Generation Techniques in Engineering Design,” *Design Engineering Technical Conference,* ASME, Atlanta, GA, pp. 1–15.

[49] Puranam, P., Singh, H., and Chaudhuri, S., 2009, “Integrating Acquired Capabilities: When Structural Integration Is (Un)necessary,” Organ. Sci., **20**(2), pp. 313–328.

[50] Hegarty, M., Kriz, S., and Cate, C., 2003, “The Roles of Mental Animations and External Animations in Understanding Mechanical Systems,” Cogn. Instr., **21**(4), pp. 209–249.

[51] Jansson, D. G., and Smith, S. M., 1991, “Design Fixation,” Des. Stud., **12**(1), pp. 3–11.

[52] Weber, B., and Hertel, G., 2007, “Motivation Gains of Inferior Group Members: A Meta-Analytical Review.,” J. Pers. Soc. Psychol., **93**(6), pp. 973–993.

[53] Geister, S., Konradt, U., and Hertel, G., 2006, “Effects of Process Feedback on Motivation, Satisfaction, and Performance in Virtual Teams,” Small Gr. Res., **37**(5), pp. 459–489.

[54] Mullen, B., Salas, C., and Johnson, E., 1991, “Productivity Loss in Brainstorming Groups: A Meta-Analytical Integration,” Basic Appl. Soc. Psych., **12**(1), pp. 3–23.

[55] Stroebe, W., and Diehl, M., 1994, “Why Groups Are Less Effective than Their Members: On Productivity Losses in Idea-Generating Groups,” Eur. Rev. Soc. Psychol., **5**(1), pp. 271–303.

[56] Wodehouse, A., and Ion, W., 2012, “Augmenting the 6-3-5 Method with Design Information,” Res. Eng. Des., **23**(1), pp. 5–15.

[57] Summers, J. D., 2015, “2015-08-ME4010 Complete Lectures [Powerpoint Slides],” pp. 90–115.

[58] Mocko, G. M., 2015, “2015-09-15 ME8700 Lecture [Audio Recoding].”

[59] Zanker, W., 2016, “2016-06 ME4010 Functions Lecture [Video Recording].”

[60] Borrego, M., Karlin, J., Mcnair, L. D., and Beddoes, K., 2013, “Team Effectiveness Theory from Industrial and Organizational Psychology Applied to Engineering Student Project Teams: A Research Review,” J. Eng. Educ., **102**(4), pp. 472–512.

[61] Mathieson, J. L., Arlitt, R., Summers, J. D., Stone, R., Shanthakumar, A., Sen, C., Arlitt, R., Summers, J. D., and Stone, R., 2011, “COMPLEXITY AS A

SURROGATE MAPPING BETWEEN FUNCTION MODELS AND MARKET VALUE," Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf., pp. 1–10.

[62] Patel, A., Gill, A. S., Kramer, W., Summers, J., and Shuffler-Porter, M., 2016, "Graph Complexity Analysis of Function Models Expanded from Partially Completed Models," *4th International Conference on Design Creativity*, J. Linsey, M. Yang, and Y. Nagai, eds., Atlanta, GA.

[63] Patel, A., Andrews, P., and Summers, J. D., 2016, "Evaluating the Use of Artificial Neural Networks, Graph Theory, and Complexity Theory to Predict Automotive Assembly Defects," *Volume 4: 21st Design for Manufacturing and the Life Cycle Conference; 10th International Conference on Micro- and Nanosystems*, ASME, Charlotte, p. V004T05A003.

[64] Sri Ram Mohinder Cherkadu Vasala, Amaninder Singh Gill, and Joshua D Summers, 2016, "Using Graph Complexity Connectivity Method to Predict Information from Design Representations: A Comparative Study," *Design Computing and Cognition'16*, pp. 703–720.

[65] Mathieson, J. L., Wallace, B. A., and Summers, J. D., 2013, "Assembly Time Modelling through Connective Complexity Metrics," Int. J. Comput. Integr. Manuf., **26**(10), pp. 955–67.

[66] Nagel, R. L., Bohm, M. R., Linsey, J. S., and Riggs, M. K., 2015, "Improving Students' Functional Modeling Skills: A Modeling Approach and a Scoring Rubric," J. Mech. Des., **137**(5), p. 51102.

[67] Patel, A., Kramer, W., Summers, J. D., and Shuffler-Porter, M., 2016, "Function Modeling: A Study of Sequential Model Completion Based on Count and Chaining of Functions," *Volume 7: 28th International Conference on Design Theory and Methodology*, ASME, Charlotte, United States, p. V007T06A036.