International Journal of Engineering Education Vol. 37, No. 5, pp. 1398–1413, 2021
Printed in Great Britain

0949-149X/91 $3.00+0.00

Systems Thinking in Engineering Design: Differences in Expert vs. Novice*

YUZHEN LUO[1], KURT BECKER[1], JOHN GERO[2], IDALIS VILLANUEVA ALARCON[3] and OENARDI LAWANTO[1]

[1] Department of Engineering Education, Utah State University, Old Main Hill, Logan, Utah 84322, USA.

[2] Department of Computer Science and School of Architecture, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223-0001, USA.

[3] Engineering Education Department, University of Florida, Gainesville, FL 32611, USA. E-mail: yuzhen.luo@aggiemail.usu.edu

Systems thinking is a way of seeing and talking about the system so that we can understand and work with the system. It is both a cognitive ability and skill that is desired amongst engineers because of the complex problems that they are expected to solve in the workplace. Developing systems thinking capabilities of the engineering workforce is an industry endeavor as well as a desirable learning outcome for engineering education. This opens opportunities for research to better understand systems thinking of experts (professional engineers) in industry and novices (engineering students) in postsecondary education. The purpose of this study was to understand and compare the differences between expert and novice systems thinking in engineering design. Knowledge of experts and novices in their systems thinking can help inform engineering education on ways to bridge this gap in post-secondary settings. Using tools developed from Function-Behavior-Structure (FBS) Ontology, existing protocol data for 61 teams of 2 (18 professionals, 19 seniors, and 24 first-year students) underwent systems hierarchical coding and statistical analysis. Results show that systems thinking of professionals and senior engineering students are similar while first-year students were significantly different from their counterparts. This paper discusses several implications for systems thinking in engineering education and future research.

Keywords: systems thinking; expert-novice; problem decomposition and recomposition; engineering education

1. Introduction

Systems thinking is a cognitive ability [1–5] and a practical skill [6–8]. From a cognitive standpoint, systems thinking is a way of seeing and talking about the system so that the system is understood [9]. Systems thinking is also a set of tools because it offers techniques and devices that describe and communicate systematic behavior [9]. In this view, systems thinking is a skill that utilizes a unique set of tools to solve complex engineering problems [1], develop complex systems [10], and increase project success [11].

As systems become more complex and senior practitioners approach retirement, companies are faced with challenges to develop systems thinking capability of their workforce [11, 12]. Moreover, the importance of systems thinking is identified as a missing competency in engineering graduates [7, 8]. As such, researchers, engineers, employers, and government agencies look to engineering education as a solution to fill this void. This study investigates the differences of expert and novice engineering designers' systems thinking. Understanding how experts solve complex engineering design problems through systems thinking set a benchmark for future curriculum development. Sixty-one teams of two (18 professionals, 19 seniors and 24 first-year) of verbal protocol data were coded and analyzed quantitatively. Results show that systems thinking of professional engineers were significantly different from first-year students, but not seniors.

2. Relevant Literature Review

The definition of a system is more consistent than systems thinking in the literature. A system can be defined as the interaction of solution elements to achieve some higher order function or purpose [4]. The term solution element is synonymous with terms used by other authors including, entities and their relationships [3], integrated set of elements [13], and a set of physical parts that are part of a bigger whole [2]. Building on the definition of a system, systems thinking takes on a cognitive stance on how to solve a complex system [1–5]. It is considered a cognitive activity because it involves various modes of reasoning such as critical reasoning – evaluating the validity of claims, analytical reasoning – analysis from a set of laws or principles, and creative thinking – thinking outside of the box [3]. Furthermore, it is the mental capacity and ability of designers and engineers to treat problems as complex, and to see the system as a whole, rather than in part [1]. Good problem-solvers and designers should be associated with maintaining sight of the big picture by including systems thinking in engineering design [14]. Seeing the system as a

* Accepted 24 May 2021.

whole is synonymous with what other authors referred to as 'a big picture view' of the complex system [2] and 'holistic view' of the system [7, 14]. Systems thinking as a holistic view help designers and engineers focus on the relationships of the entities and the emergence of the desirable functions or outputs of these relationships [3]. More importantly, it allows one to comprehend the coherence and synergy of the system to produce the desired function [2].

Contrary to the holistic view is the reductionist view [15–17]. The reductionist view dissects the parts of the system to learn about the system [16]. The process of taking the system apart is recognized as a "Hierarchical Mappings" view of complex systems in engineering design [4]. In this view, the system is decomposed into the various levels or hierarchies of the system through a problem decomposition approach. Problem decomposition is the process of dividing the interacting elements in the system hierarchically into smaller manageable sub-problems. On the reverse end of this process, lies problem recomposition. This is where the interacting elements, also referred to as solution elements, are eventually synthesized to form the system behaviors, system functions, and outputs. A graphical representation of hierarchical mapping of a system by the International Council of Systems Engineering (INCOSE) is summarized in Fig. 1 [18]. Problem Decomposition, Recomposition, and Solution Elements are labeled for clarification. One may learn a lot by taking the system apart through the reductionist view. However, it is important to maintain sight of the relationships that bind the parts and subsystems to form the higher levels of hierarchies [16], which eventually merge to produce the system behavior and purpose. This does not mean that the systems (holistic) view is superior to the reductionist view. Both views are complementary and work together to build a more complete understanding of the system [16].

According to the Vee Model, "the top-down branch is done by successive levels of decomposition; each level corresponds to the physical architecture of the system and system elements. The bottom-up branch consists in following the opposite way of composition level by level" [19, p. 81] or recomposition. Although complex systems can be decomposed into parts recursively, "the emergent properties that we really care about disappear when we examine the part in isolation". Therefore, Systems Engineering requires "a balance of linear, procedural methods for sorting through complicatedness and holistic, non-linear, iterative methods for harnessing complexity" [19, p. 9]. During this process, systems thinking and analysis is always required. The tension between breaking things apart and binding them should be dynamically managed.

2.1 Systems Thinking and Engineering Education

The Accreditation Board for Engineering and Technology, Inc. (ABET) asserts that students should have "an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics" as the foremost student outcome for baccalaureate engineering programs in U.S. universities [20, p. 5]. However, systems thinking, a competency that contributes to such learning outcomes, is identified as a missing competency in engineering graduates [7, 8]. Consequently, researchers, engineers, and government agencies look to engineering education as a solution to fill

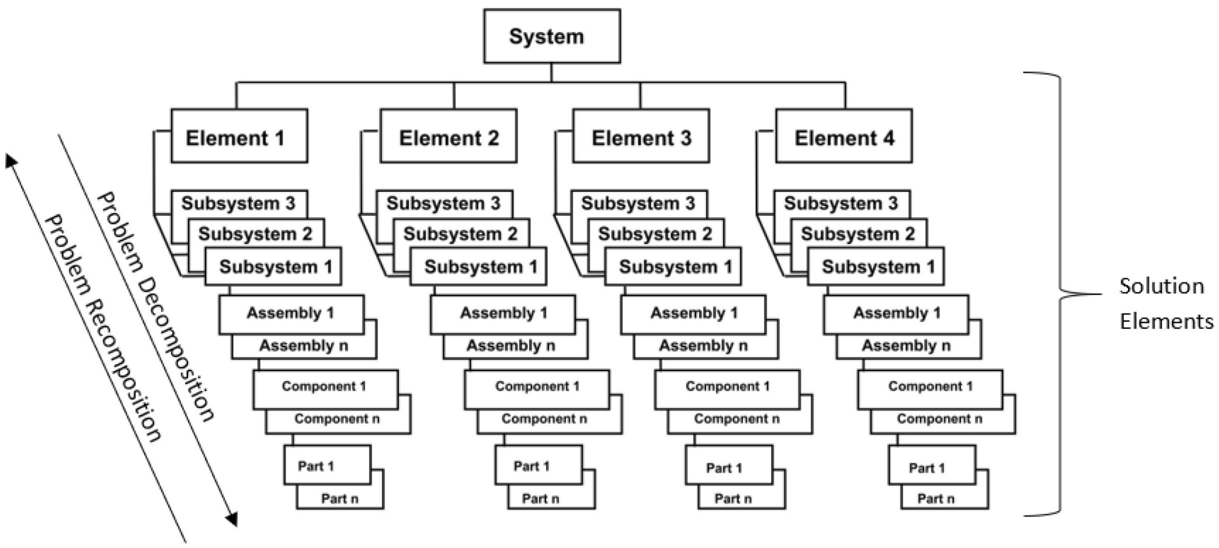

Fig. 1. System Hierarchy. Note. Figure adapted from INCOSE handbook [18].

this void. Although some universities tackled this endeavor from a systems engineering and engineering design perspective, they encountered many challenges [21]. Systems engineering programs found it challenging to integrate systems thinking into their engineering curriculum, given their already overwhelming amount of important materials to cover [8]. Furthermore, engineering design education, which span various engineering disciplines, has proven to be difficult for students learning about design thinking, and for faculty seeking to teach these skills [14]. Despite evidence in support of project-based learning as a successful design pedagogy to improve student learning, resource allocation (e.g., faculties and facilities) towards design pedagogy remain low on priority [14]. Moreover, critics argue that systems thinking and systems design require engineering education that achieves competence rather than specialization in subject knowledge [22]. This requires an ability to learn and progress through an open-ended, formative, and dynamic learning process rather than the traditional 'rote' application of pre-defined knowledge [22]. Others assert that traditional engineering programs lack formal engineering education to help students understand the holistic implications of ill-structured problems, which students are likely to encounter after graduation [7-8]. Despite the challenges to incorporate systems thinking in engineering education, research and implementation of systems thinking has become active in the past decade, and some studies have shown evidence of success [7, 14, 23–25].

These successes to improve engineering education have led to opportunities. One opportunity is to better understand expert knowledge of systems thinking, then compare it to that of novices to identify the differences between experts and novices [26–29]. However, expert time is a challenge for researchers to obtain, and as a result, few studies in engineering education have expert data on systems thinking. Consistent with the literature, experts are professional engineers that acquired at least 10 years of work experience, or 10,000 hours, and novices are students enrolled in an undergraduate engineering program [27–32].

The transition from novice to expert is continuous, and intermediate stages of expertise exist between novices on the one end, and experts on the other end. When studies compare experts and novices, it is recognized as the relative approach [33]. The assumption here is that experts are more knowledgeable and are more experienced *relative* to the novices. Therefore, novices should seek to achieve the level of proficiency of experts. The alternative research approach to studying experts is an absolute approach, where truly exceptional people in their area of expertise are studied [33]. Since this study compares experts and novices, the relative approach assumption is accepted.

3. Research Design

The purpose of this study was to investigate systems thinking of teams of expert engineers and teams of novice engineering students. It is guided by the research question: *What are the differences in systems thinking between professional engineers and engineering students when solving engineering design problems*? This research synergistically brings together methodologies from different disciplines to characterize and model the effects of education and experience on engineering students' and expert designers' design cognition. The methodologies are drawn from:

- Design theory: design ontologies.
- Cognitive science: protocol analysis and cognitive style.
- Statistical modeling: correspondence analysis, standard statistical analysis and Markov modeling.

Using tools developed from Function-Behavior-Structure (FBS) [34], and existing FBS coded protocol data for 61 teams, systems level codes were generated through a systems hierarchy coding scheme [35]. These system level codes formed the database for statistical analysis.

3.1 Participants

The participants consisted of professional engineers and undergraduate engineering students (seniors and first-year) that previously took part in a National Science Foundation (NSF) funded research project. There were 61 teams, where each team consisted of two members, therefore, a total of 122 participants were involved in this study. There were 43 teams of undergraduate engineering students (19 seniors and 24 first-year students) from a large public university in western United States, who majored in mechanical, civil and environmental, and biological engineering. First-year students were engineering students enrolled in first-year engineering undergraduate courses, and seniors were engineering students enrolled in senior engineering courses. Eighteen teams of professional engineers were recruited from companies in western United States. Professional engineers had at least 10 years of work experience, or 10,000 hours, and a Professional Engineering license was not required.

The teams were invited to design a window opening device within 1-hour. See Appendix A for details of the design task. Participants were encouraged to think-aloud and their design sessions were audio and video recorded. Verbal protocol data for the 61 design sessions underwent FBS coding in the aforementioned NSF research project. The FBS coding scheme was used to quantify the verbal protocol data in a uniform way [36]. More importantly, the design sessions were segmented to ensure that each segment contained a single design idea. See Fig. 3, column D, for an example of FBS segmentation and coding.

3.2 Function-Behavior-Structure (FBS) Ontology and Relation to Systems Thinking

FBS is an instrument used to measure design. The FBS ontology describes all designed issues, or artefacts, irrespective of the specific discipline [36]. It is a uniform way to characterize and measure designing in three fundamental constructs – *Function, Behavior, and Structure*. The goal of designing is to transform a set of functions, driven by client *Requirements* (R) into a set of descriptive *Documentations* (D). *Function* (F) is the intended teleology or "what the artefact is for". *Behavior* is "what the artefact does" and provides measurable criteria for comparison. Designers decide which behaviors are significant and needed to assess the designs they produce. Therefore, there are two types of behaviors; it can either be *Expected Behavior* (Be), which is the measurable outcome set by expectations, or derived *Behavior from the Structure* (Bs), which is what the artefact actually does. The *Structure* (S) is the physical components and their relationships or "what the object consists of". The six codes, F, Be, Bs, S, R, and D, are referred to as "FBS Issues" and provide the basis for coding design protocols. See example of FBS Issues of a phone in Fig. 2 and a summary of definitions in Table 1. The phone itself is a physical structure. A purpose or function of a phone is to be mobile and be easy to carry in the pocket. To achieve this, the volume and size, which are physical structures, of the phone is expected to fit the size of a pocket.

Function-Behavior and Part-whole Structure are foundational concepts of systems thinking [2]. Part-whole Structure means that parts of a system can be decomposed into sub-systems and recomposed in a bottom-up fashion until the top-level system is reached. Function and Behavior acknowledge that engineered systems are designed for a purpose.

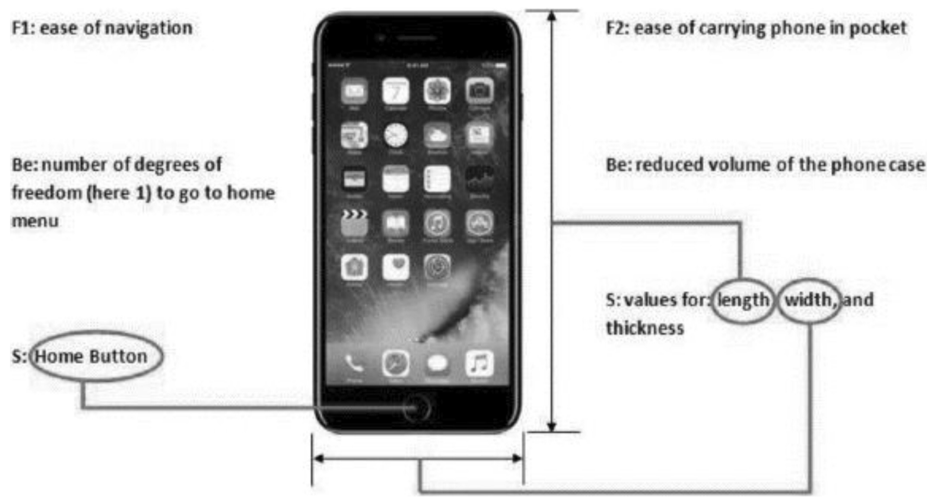

Fig. 2. Example of FBS Using a Phone.

Table 1. FBS Issues Definition and Examples (using Fig. 2)

FBS Issue	Code	Definition	Example
Function	F	The intended teleology or purpose	• Ease of navigation • Ease of carrying phone in pocket
Expected Behavior	Be	A measurable outcome set by expectations	• One degree of freedom to go to home menu • Reduced volume of the phone case
Behavior from Structure	Bs	Behavior of the structure i.e. what the structure does	• Phone *rings* • Phone *vibrates*
Structure	S	The physical components and their relationships	• Phone • Home Button • Length, width, thickness
Requirements	R	Client requirements	• Comply with ADA and safety standards
Documentation	D	Descriptions in the form of documentation	• Designer takes note or document his/her work

Then, structures with their functions are selected and put together to produce some change. The change that results from this is the behavior observed in the system. In this view, systems are treated as hierarchical because the parts of the system are arranged in a manner that can be decomposed and/or recomposed.

In view of systems as hierarchies, the tools developed to measure systems thinking in engineering design accounted for the various hierarchies that exist within a system. One way to analyze design problems is to distinguish the various levels of abstraction in the problem domain [35]. The designer's attention shifts from a high-level view of the problem to a low-level view of the problem. High-level view occurred when the designer considers the problem at the functional or systems level with a holistic view. Low-level view occurred when the designer considers the problem at the details level. The authors defined the various levels by assigning a number to a level of the problem. Level 0 is the System level (high-level or holistic view) – where the designer is considering the system as a whole, level 1 is the Interactions – where the designer is considering the interactions between the sub-systems, level 2 is the Sub-systems – where the designer is considering details of the sub-systems, and level 3 is the Details (low-level view) – where the designer is considering the detailed workings of a particular sub-system. This method was used in previous studies that conducted similar protocol studies in engineering design [27, 29].

This study condensed the four levels of abstraction (0, 1, 2, 3) to *three new levels: 1, 2 and 3* to better distinguish the system hierarchy at the system level – the top level, subsystem level – the middle level, and details level – the bottom level. Specifically, Level 0 was labeled as *Level 1*. Levels 1 and 2 was merged to *Level 2* because subsystems and interactions between subsystems occur at the same level of the hierarchy, which is the subsystems level. Level 3 remained as *Level 3*. The three new levels are referred to as system levels for the rest of the manuscript. Requirements (R) was contextual and could either be system level 1, 2, 3 or O, where O described utterances that did not incorporate the design problem at any system level. An example would be, Documentation (D) from FBS Issues. D occurred when designers took notes on paper or wrote on the whiteboard. It served as external memory and did not contribute anything new to their design nor describe the problem at any system level. Therefore, "O" was a code assigned to D, as well as other utterances that did not pertain to any systems level. System levels 1, 2, 3 and O guided the systems hierarchy coding for this study and are summarized in Table 2.

3.3 Systems Hierarchy Coding

The researcher was coder 1 and two graduate students, who were experienced in FBS coding, volunteered to be coder 2 and 3 for some design sessions. Training on systems level coding was provided to the two graduate students. Training materials included a literature review on systems coding, Table 2, and examples of coding like Fig. 3. The researcher was coder 2 for the remaining design sessions. This meant that the researcher coded each design session twice, with at least ten days in between each coding. The ten-day break addresses the issue of coder fixation on the first round of coding and improves independence of each round of coding [35].

Description of the codes are provided; however, the actual coding required the coders to use Table 2 within context as exemplified by Fig. 3, Example A. This was a dialogue between two members of the team at approximately 16 minutes into the design session, where they discussed the possibility of using a lever-type system to help open the sticky window. The explanations or reasons (column F) for the systems level codes (column E) are straightforward and therefore easy to interpret. However, this was not always the case for other design teams, such as Example B, which was a dialogue that evaluated their clamp system and discussed possible user interactions. Some utterances resulted in coder disagreements. In example B, it was clear from row 2 and 3 that their system was a clamp and both coders, coder 1 and coder 2, agreed on system

Table 2. System Levels

Level	Systems Hierarchy	Description
1	System	The designer is considering the system as a whole. This is the top-level view as the designer is obtaining a holistic or big-picture view of the problem.
2	Subsystems and their interactions	The designer is considering the subsystems (as a whole) and their interactions or relationships. This is the middle-level view as the designer is breaking the complex system into smaller and manageable subsystems.
3	Details	The designer is considering the details of the subsystems. This is the low-level view as the designer is working out the inner details of a particular subsystem such as size, dimensions, mathematical analysis, etc.
O	N/A	The designer is not considering the problem at levels 1, 2 or 3.

Example A. System Level Coding in Context. Note. Column C are the utterances between the two students.

	A	B	C	D	E	F
1	Time	Person		FBS Code	Systems Level	Explanation
2		A	Is there like a lever system that we could put in there that could be easy?	S	1	lever is the system that they are discussing
3		B	Umm…you could do like a jack type of system.	S	1	a jack is an example of the system (the lever)
4		B	Like you stick it in there or something, pop it open, and then jack it up which raises it.	BS	3	considering the details of how to use a jack
5		B	And then you remove the jack to put it back down.	BS	3	considering the details of how to use a jack
6	16:11	B	Ya, depending on the type of jack.	S	3	considering the details of how a different jack could be used
7		A	Then how do you put it down.	Be	2	considering the interactions between subsystems, (the user and the down movement of the jack)
8		B	Could you just put weights on that? Just like hang weights and it just…	S	3	considering the details of adding weights

Example B. Coder disagreement.

	A	B	C	D	E	F	G
1	Time	Person	Utterance	FBS Code	Systems Level	Coder 1	Coder 2
2		B	Because our device right here,	S	1	1	1
3		B	this clamp,	S	1	1	1
4		B	should be strong enough to hold	Be	1	2	1
5		B	the window up itself.	S	2	2	2
6		B	And then when you want to close it,	Be	2	2	2
7		B	you just got untwist it.	Be	2	2	2

Fig. 3. Examples of System Level Coding.

level 1 (column E). However, there was a disagreement in row 4 where coder 1 (column F) thought the idea of being "strong enough to hold" should be at system level 2, whereas coder 2 (column G) thought it should be at system level 1. The coders came to an agreement, through an arbitration process, that it should be at system level 1 because an evaluation about the system – the clamp, is at the systems level or level 1.

The arbitration process allowed a final codebook to be developed through an agreement of the codes from the two coders. Although the arbitrated codes were usually the same as one of the coders, it could also be different from both coders. This coding process can be viewed as a continuous improvement method that allowed the coders to learn and change their codes or opinions over time. Consequently, the code agreement between the coder and the final arbitrated codes increased over time. Two measures of intercoder-reliability were used, coder percentage agreement and Cohen's Kappa. The goal was to reach a coder agreement of 80% to be consistent with similar systems level coding [27] and FBS coding [30].

3.4 Research Question

This study is guided by the following research question:

> *What are the differences in systems thinking between professional engineers and engineering students when solving engineering design problems?*

The research question purports to measure systems thinking between engineering students and professional engineers through a comparisons of system levels and system processes. System levels show whether experts and novices are big picture oriented (system level 1), detail oriented (system level 3), or juggle with the sub-systems (system level 2). System processes are the transitions between each system level. It shows their top-down and bottom-up

Table 3. Problem Decomposition and Recomposition

Problem Decomposition/Recomposition	Systems Process
Problem Decomposition	$1 \rightarrow 2, 1 \rightarrow 3, 2 \rightarrow 3$
Problem Recomposition	$2 \rightarrow 1, 3 \rightarrow 1, 3 \rightarrow 2$
Same Level	$1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3$

problem-solving strategy when solving engineering design problems. A top-down problem-solving strategy is problem decomposition, and a bottom-up problem-solving strategy is problem recomposition. Both problem-solving strategies are evident during design [27, 29].

Problem decomposition and recomposition was measured by analyzing the sequential process of system levels 1, 2 and 3. System level O was omitted for this part of the analysis. Sequential means that each systems code is paired with the next code, and these pairs of codes formed the processes of either problem decompositions, problem recompositions, or same level. When the designers went from a higher level to a lower level, for example level 1 to level 3 (in rows 3–4 in Example A of Fig. 3), it is a top-down approach and is considered problem decomposition. Conversely, if the designers went from a lower level to a higher level, for example level 3 to level 2 (rows 6–7 in Example A of Fig. 3), it is a bottom-up approach and is considered problem recomposition. Problem decomposition and recomposition are derived from the systems processes and are summarized in Table 3.

Percentages of system levels (1, 2, 3, and O) and system processes (problem decomposition, problem recomposition, and same level) were computed in Excel with the aid of macros and other Excel built-in functions for each design session. The process was repeated for all the sessions and the results were aggregated based on the three cohorts: professional engineers, senior students, and first-year students. An example of the professional engineers' data is attached in Appendix B. The data was then analyzed via Correspondence Analysis (CA), descriptive and inferential statistics, and Markov models.

CA is a form of multivariate analysis [37]. It provided an overview of the data and pointed to areas of similarities so that the researcher can begin to see emerging relationships. For a more detailed analysis of system levels and system processes for the three cohorts, descriptive and inferential statistics were used. Means and standard deviations were computed and compared across the three cohorts. ANOVA was performed on the percentages of system levels and system processes for the three cohorts. A significance level of 0.05 was selected and post-hoc tests were followed up for ANOVA, $p \leq 0.05$ to identify where and which cohorts differed. ANOVA assumptions for normality was determined by Shapiro Wilk test, $p \geq 0.05$ and homogeneity of variance by Levene's test, $p \geq 0.05$. If normality was not met, a non-parametric test was used instead, and Kruskal-Wallis p-values are reported. Independent samples were met by design of the experiment because each design team's data was collected independently. Statistical significance was coupled with measurements of effect size, specifically, partial eta squared, which measured the strength of the relationship between the variables. A larger effect size implied a stronger relationship and a larger practical significance. Statistical Package for Social Sciences (SPSS) was the software that used for the analysis.

To further understand problem decomposition and recomposition, Markov models [38] was used to calculate the probabilities of going from one system level to another system level, or the probabilities of the system processes. Markov analysis enable us to see decomposition and recomposition of the cohorts at a granular level by examining each system process. This is explained with the help of a tree diagram in Fig. 4. In general, the probabilities are obtained by weighing the system process of interest to all possibilities of that system process. For example, what is the probability (P) of going from system level 1 ($x = 1$) to system level 2 or P ($1 \rightarrow 2$)? We would count all occurrences of $1 \rightarrow 2$ and divide by the counts of all possibilities: $1 \rightarrow 1$, $1 \rightarrow 2$, $1 \rightarrow 3$, and $1 \rightarrow O$ or mathematically:

$$P(1 \rightarrow 2) = \sum(1 \rightarrow 2) / \sum(1 \rightarrow 1, 1 \rightarrow 2, 1 \rightarrow 3, 1 \rightarrow O) \quad (1)$$

The result is a value between 0 and 1, which is the probability of going from system level 1 to 2. This was repeated for all other values of x for the three cohorts. The results were aggregated based on

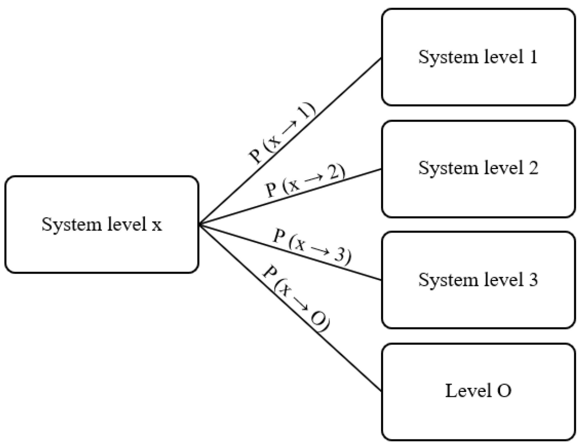

Fig. 4. Probabilities of System Processes. Note. P ($x \rightarrow 1$) is the probability of system level x going to system level 1, where x = 1, 2, 3, or O.

cohorts, and their means, standard deviations, ANOVA, and effect size were computed.

4. Results

Using the methodology described above, systems thinking of experts and novices were measured and compared quantitatively. The results were based on 61 design team's verbal protocol data, which underwent systems hierarchical coding. The average coder agreement was 80% and an intercoder reliability, measured by Cohen's kappa (k), was 0.78. The results from the hierarchical coding produced 61 sessions worth of data for quantitative analysis, which focused on system levels and system processes.

To provide an overview of the data, correspondence analysis (CA) brought together the system levels (level 1, 2, 3, O), system processes (problem decomposition, recomposition, same level), and the three cohorts (professionals, seniors, first-year) into a single 2D plot – see Fig. 5. Systems thinking refer to the system levels and system processes. System levels, system processes, and the cohorts are treated as categorical data, which are shown as circles. A total of 10 categories are presented in Fig. 5. Dimension 1 covers 95.6% of the variance of the data and dimension 2 covers 4.4% of the variance of the data; they add up to cover 100% of the variance of the data. The positions of each category on dimension 1 (x-axis) and dimension 2 (y-axis) on the plot shows how similar or dissimilar the categories are in relation to each other.

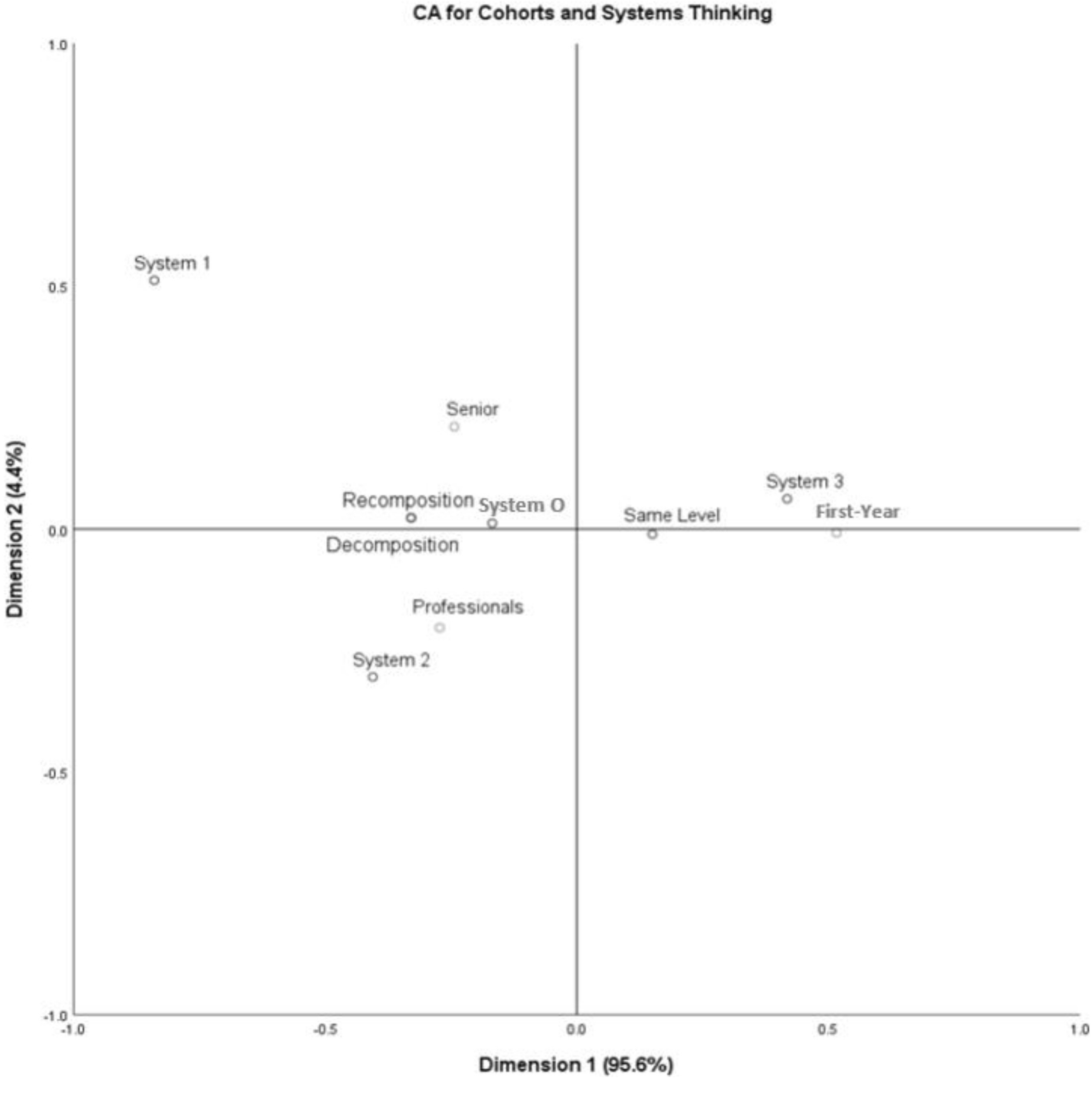

Fig. 5. Correspondence Analysis for Cohorts and Systems Thinking.

Table 4. Results for System Levels

System Levels	Level 1		Level 2		Level 3		Level O	
Cohort (M, SD)	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Professionals (N = 18)	0.12	0.05	0.34	0.08	0.46	0.12	0.08	0.03
Seniors (N = 19)	0.14	0.06	0.30	0.08	0.48	0.10	0.08	0.03
First-year (N = 24)	0.06	0.03	0.23	0.06	0.64	0.08	0.06	0.02
ANOVA (p-Values)	0.000**		0.000**		0.000**		0.052	
Post-Hoc Test (p-Values)								
Professionals vs Seniors	0.679		0.264		0.845		0.965	
Professionals vs First-year	0.001**		0.000**		0.000**		0.135	
Seniors vs First-year	0.001**		0.006**		0.000**		0.071	
Effect Size (Partial Eta Squared)	0.336		0.300		0.432		0.097	

*$p \leq 0.05$. **$p \leq 0.01$. M = mean, SD = standard deviation.

CA plots the categories in a 2D space to indicate categories that are similar. There are three ways to identify categorical similarities and differences: (1) left and right of dimension 1, which covers a majority of the variance of the data (95.6%), (2) top and bottom of dimension 2, which covers 4.4% of the variance of the data, and (3) a combination of dimension 1 and 2, which are the four quadrants. Since dimension 2 only covered 4.4% of the variance of the data, much of the variance of the data is covered in dimension 1. For example, first-year, same level, and system level 3 sit on the positive side of dimension 1, therefore they are categorically similar to each other in dimension 1. This interpretation can be applied for the categories on the negative side of dimension 1 and top and bottom of dimension 2 to show other categorical similarities.

A more meaning way to interpret CA results is by looking at the quadrants, a combination of both dimensions. From Fig. 5, system level 1, 2, and 3 sit in different quadrants, which suggest that they are categorically different from each other. Similarly, the three cohorts: professionals, seniors and first-year, sit in different quadrants, which suggest that they are categorically different. On the contrary, decomposition and recomposition sit in the same quadrant, in fact, the circles overlap, which suggest that they are categorically very similar. Senior students sit in the same quadrant as system level 1, system level O, decomposition, and recomposition, which suggest categorical similarities. The same can be said for professionals and system level 2, and First-year and same level. To better understand the categorical similarities, we conducted descriptive and inferential statistics and Markov models.

Each session was normalized by taking a ratio of the system level (1, 2, 3, or O) over the total system levels. This can be multiplied by 100 to obtain the percentages. System levels and system processes for each session were grouped then aggregated by professionals, seniors, and first-year. The results for the means (M), standard deviations (SD), and ANOVA for system levels and system processes are summarized in Tables 4 and 5 respectively. Refer to Tables 2 and 3 for definitions of system levels and system processes respectively.

The means and standard deviations in Table 4 show that on average, senior students are the highest for System Level 1 at 14% with a 6% standard deviation. Professional engineers are the highest for System Level 2 at 34% with an 8% standard devia-

Table 5. Results for System Processes

System Processes	Problem Decomposition		Problem Recomposition		Same Level	
Cohort (M, SD)	*M*	*SD*	*M*	*SD*	*M*	*SD*
Professionals	0.17	0.03	0.17	0.03	0.66	0.07
Seniors	0.17	0.03	0.17	0.03	0.66	0.07
First-year	0.13	0.03	0.13	0.03	0.73	0.06
ANOVA (p-Values)	0.000**		0.001**		0.000**	
Post-Hoc Test (p-Values)						
Professionals vs Seniors	0.995		0.987		1.000	
Professionals vs First-year	0.001**		0.006**		0.003**	
Seniors vs First-year	0.002**		0.005**		0.002**	
Effect Size (Partial Eta Squared)	0.246		0.217		0.232	

*$p \leq 0.05$, **$p \leq 0.01$, M = mean, SD = standard deviation.

Table 6. Results for Markov Models

System Processes	**Problem Decomposition**						**Problem Recomposition**						**Same Level**					
	1→2		1→3		2→3		2→1		3→1		3→2		1→1		2→2		3→3	
Cohort (M, SD)	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Professional	0.22	0.06	0.19	0.07	0.25	0.05	0.10	0.04	0.04	0.02	0.21	0.10	0.43	0.09	0.58	0.04	0.69	0.11
Seniors	0.22	0.07	0.23	0.11	0.26	0.09	0.11	0.04	0.06	0.03	0.17	0.06	0.40	0.11	0.56	0.08	0.71	0.08
First-year	0.22	0.07	0.28	0.07	0.36	0.06	0.07	0.03	0.03	0.02	0.13	0.04	0.38	0.08	0.52	0.06	0.79	0.06
ANOVA (p-Value)	1.000						0.001**						0.222		0.005**		0.001**	
Kruskal-Wallis (p-Values)			0.001**		0.000**				0.000**		0.005**							
Post-Hoc Test (p-Values)																		
Professionals vs Seniors			0.283		0.927		0.476		0.063		0.276				0.547		0.848	
Professionals vs Freshmen			0.000**		0.000**		0.029*		0.050*		0.002**				0.005**		0.001**	
Seniors vs Freshmen			0.009**		0.000**		0.001**		0.000**		0.040*				0.075		0.007**	
Effect Size (Partial Eta Squared)			0.176		0.364		0.221		0.269		0.202				0.168		0.224	

*$p \leq 0.05$, **$p \leq 0.01$, M = mean, SD = standard deviation.

tion, and first-year students are the highest for System Level 3 at 64% with an 8% standard deviation. ANOVA results were statistically significant across systems levels 1, 2, and 3. Post-hoc tests showed that professional engineers and senior students have no statistically significant differences across system levels 1, 2, and 3. However, professional engineers and first-year students are significantly different for the three system levels. Similarly, senior and first-year students are significantly different for system levels 1, 2, and 3. The effect size for system levels 1, 2, and 3 ranged from 0.3–0.4, which is considered large. No statistically significant differences were found for system level O.

The means and standard deviations in Table 5 show that on average, professional engineers and senior students had 17% with a 3% standard deviation for problem decomposition and recomposition, and 66% with a 7% standard deviation for same level. First-year students had 13% with a 3% standard deviation for problem decomposition and recomposition, and 73% with a 6% standard deviation for same level. Problem decomposition and recomposition constitute 13–17% of system processes or 26–34% combined, whereas same level constitutes 66–73% of system processes.

ANOVA results showed that statistically significant differences were found between the cohorts for all three system processes; problem decomposition, problem recomposition, and same level. In fact, the results were very significant with $p < 0.001$. Post-hoc tests indicate that professional engineers and senior students have no statistically significant differences across all system processes. Professional engineers and first-year students are significantly different for all system processes. Senior vs first-year students are significantly different for all system processes. The effect size for system processes were between 0.22 and 0.25, which is considered a large effect.

To identify specifically where the cohorts differed in system processes, a Markov model was employed. The results are summarized in Table 6. The means and standard deviations are the probabilities of the systems process. For example, in the first-row professionals have a 19% chance, with a 7% standard deviation, to go from system level 1 to 3 (1→3). Seniors have a 23% chance to go from systems level 1 to 3 and freshmen have the highest chance at 28%. The rest of the table can be interpreted in a similar manner. Problem decompositions 1→3, 2→3, recompositions 2→1, 3→1, 3→2, and same level 2→2, 3→3 were found to be very statistically significant. No statistically significant differences were found for decomposition 1→2 and same level 1→1. The effect size for the Markov models ranged from 0.17 to 0.36, which is considered a large effect.

5. Discussion

This study had several limitations. First, a majority of the participants were white male. This does very little to shed knowledge on minorities and less represented groups in engineering. Second, engineering students were recruited from a single large public university in western U.S., which may not be representative of all engineering programs. Future studies should diversify their participants and investigate systems thinking that include effects of gender, race, and other engineering disciplines which were not considered in the design of this study.

5.1 Systems Thinking at the System Levels – Big Picture vs Details

A closer look at the system levels in Table 4 show that professionals (with a system level 1 of 0.12) and seniors (with a system level 1 of 0.14) spend their time thinking twice as much in system level 1 than first-year students (with a system level 1 of 0.06). System level 1 is thinking about the system as a whole, where designers adopt a holistic or big picture view of the design problem. It is concerned with design functions, design goals, and desired outputs of the system that are at the top level of the system hierarchy. On the contrary, first-year students spend nearly 1.5 times more time than professionals and seniors in system level 3. System level 3 is thinking about the details of the subsystems, where designers analyze the parts and components of a particular subsystem, which are at the bottom level of the system hierarchy. Professionals and seniors were nearly 1.5 times higher than first-year students in system level 2. System level 2 is the relay between system level 1 and system level 3. It is the middle level in the system hierarchy and is concerned with subsystems or subproblems of the complex system. This is where designers consider partial behaviors of the system, major structures of the system, and user interactions with the system. The differences between professionals and seniors, to first-year students were statistically significant. It can be concluded that there are no differences in systems thinking of professionals and seniors at the system levels. Specifically, professionals and seniors adopted a holistic or big picture view of the design problem. This was lacking among the first-year students. Instead, the first-year students adopted a microscopic or detailed view of the design problem; they were more concerned with analysis and details of the design problem than their counterparts. Moreover, professionals and seniors identified more subsystems and subproblems of the complex design problem than first-year students. The ability to identifying subsystems and subproblems is aided by their ability to decompose complex systems into solution elements. This is discussed next.

5.2 Systems Thinking at the System Processes – Problem Decomposition/Recomposition

In view of a complex system as a hierarchy, problem decomposition and recomposition are processes of going from one system level to another. Based on the results in Table 5, a significant difference was found in problem decomposition and recomposition for professionals and first-year students, seniors and first-year students, however, no significant differences were found for professionals and seniors. We conclude that there was no difference in the problem decomposition and recomposition of professionals and seniors. This conflicts with results from a previous pilot study that found seniors and first-year students to be alike in problem decomposition and recomposition [27]. The results of this research show that seniors and professionals decomposed and recomposed significantly more than first-year students, and the effect size for this difference was large. One explanation is that seniors have received some education in solving complex design problems through senior capstone, engineering ethics, multidisciplinary engineering, or similar courses. First-year students preferred the alternative to problem decomposition and recomposition, which was to stay at the same level. Same level is a horizontal traverse in the system hierarchy. Cognitively, this means that when the designers are at a system level, they stay at that level and do not jump vertically to system levels above or below. First-year students tend to do this more than professionals and seniors, about 10% more. The difference was significant, and the effect was large. In order to explain why this is the case, it requires a deeper look at the probabilities of system processes for the three cohorts. This is explained using Markov analysis and the results shown in Table 6.

There is almost an 80% chance for first-year students to stay in the details level, which is level 3 to level 3 ($3 \rightarrow 3$). In fact, problem decomposition and recomposition show that first-year students are more likely to move towards the details from any other system level. The effect size ranged from 0.17 to 0.36, which is considered to have a medium to large effect. This shows a strong desire for first-year students to analyze details of the system. Based on the literature, there are multiple possible explanations for this: (1) This is a reflection of their early engineering education experience – they are detail oriented and focus on analysis such as application of equations to well-structured problems [29, 39], (2) They have low tolerance for ambiguity – once they talk about the details, then all the details must be flushed out before they move on. This confirms earlier studies that found first-year students to be frustrated when details of the problem are unknown [27] and not accepting ambiguity in the design of ill-structured problems [40], and (3) A lack of confidence to work with bigger, more complex, and ill-structured problems that are higher up in the system hierarchy. Instead, they are fixated in the details level. Fixation is a designer's tendency to adhere to existing features from the examples they encounter in their immediate surroundings or day-to-day activities [39]. This is common in engineering design [41], but first-year students in particular, tend to fixate on features of examples that they have encountered [39, 42]. The example that first-year students encounter in engineering classrooms are mostly well-structured and oriented towards detailed analysis [43], which help explain their fixation on details.

On the other hand, seniors demonstrate the ability to systems think like expert professional engineers in viewing the system as a whole. Through a holistic view, they realize the big picture functions, objectives, and goals of the design, which are complex. The complexity was partitioned into manageable subsystems, subproblems, and all the way down to the details via a decomposition strategy. The details are then synthesized into subsystems, sub-solutions, and all the way back up to the overall functions, objectives, and goals via a recomposition strategy. There are several explanations why seniors were expert-like in this regard, more importantly, a transformation from detail oriented as a first-year undergraduate engineering student to more big-picture oriented as seniors. (1) Capstone Design. Senior students are required by the college of engineering to enroll in a Capstone Design course during their senior year. The educational experiences of Capstone Design, and the transfer of this experience into complex engineering design, may be accountable for seniors' expert-like systems thinking. (2) Level of difficulty in the design task. Level of difficulty of the design task can be defined in terms of complexity such as breadth of knowledge required, intricacy of procedures, and structuredness such as interdisciplinarity and heterogeneity of interpretations [44]. A design task that is difficult for first-year students may not be as difficult for seniors and professionals if they encountered similar design problems in their engineering courses or work experiences. Therefore, the level of difficulty in the task may have contributed to the finding. Future studies should explore systems thinking with different design tasks that vary in level of difficulty.

6. Conclusion

Through the lens of engineering education, this study investigated systems thinking of experts and novices in engineering design through comparisons of systems levels and system processes. System levels measured their big picture (or holistic) view vs. details focus, and system processes measured their top-down (decomposition) and bottom-up (recomposition) problem-solving strategy. From the results of this study, it can be concluded that first-year students' systems thinking were different from senior students and experts. While the difference between first-year students to senior students and experts were clear, there were no differences in the systems thinking of experts and senior students. This study provided some evidence that systems thinking competencies can be acquired through engineering education. Engineering educators and researchers are encouraged to adapt the findings of this study to their unique environments and incorporate systems thinking education that more closely align engineering students' thought processes with those of experts.

Acknowledgements – This material is based upon work supported by the National Science Foundation under Grant Numbers: 1463873 and 1463809. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We thank the two graduate students that helped with the coding.

References

1. D. V. Behl and S. Ferreira, Systems thinking: An analysis of key factors and relationships, *Procedia Comput. Sci.*, **36**(C), pp. 104–109, 2014.
2. W. T. Chan, The Role of Systems Thinking in Systems Engineering, Design and Management, *Civ. Eng. Dimens.*, **17**(173), pp. 126–132, 2015.
3. E. Crawley, B. Cameron and D. Selva, *System Architecture: Strategy and Product Development for Complex Systems*, Pearson, 2016.
4. W. B. Rouse, Engineering complex systems: implications for research in systems engineering, *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.)*, **33**(2), pp. 154–156, 2003.
5. E. Ryen, Overview of the System Engineering Process, no. March, pp. 10–13, 15–16, 2008.
6. M. Frank, A. Sadeh and S. Ashkenasi, The Relationship Among Systems Engineer's Capacity for Engineering Systems Thinking, Project Types, and Project Success, *Proj. Manag. J.*, pp. 31–41, 2011.
7. F. Robinson-Bryant, Developing a systems thinking integration approach for robust learning in undergraduate engineering courses, *ASEE Annu. Conf. Expo. Conf. Proc.*, vol. 2018 – June, 2018.
8. T. W. Simpson and J. R. R. A. Martins, Multidisciplinary Design Optimization for Complex Engineered Systems: Report From a National Science Foundation Workshop, *J. Mech. Des.*, **133**(10), 2011.
9. D. H. Kim, Introduction to Systems Thinking, *Pegasus Commun.*, 1999.
10. E. C. Honour, Understanding the Value of Systems Engineering, in *INCOSE International Symposium*, 2004, **14**(1), pp. 1207–1222.
11. C. S. Wasson, System engineering competency: The missing course in engineering education, in *ASEE Annual Conference and Exposition, Conference Proceedings*, 2012.
12. H. L. Davidz, Enabling Systems Thinking To Accelerate The Development of Senior Systems Engineers, *Dr. Diss.*, 2006.
13. J. R. Armstrong and J. Wade, Development of systems engineering expertise, *Procedia Comput. Sci.*, **44**(C), pp. 689–698, 2015.
14. C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey and L. J. Leifer, Engineering design thinking, teaching, and learning, *IEEE Eng. Manag. Rev.*, **34**(1), pp. 65–90, 2006.
15. F. Camelia and T. L. J. Ferris, Systems Thinking in Systems Engineering, *INCOSE Int. Symp.*, **26**(1), 2016.
16. D. H. Meadows, *Thinking in Systems: A Primer*, Earthscan, 2009.
17. J. P. Monat and T. F. Gannon, What is Systems Thinking? A Review of Selected Literature Plus Recommendations, *Am. J. Syst. Sci.*, **4**(1), pp. 11–26, 2015.
18. T. J. Higgins, INCOSE: Systems engineering handbook, *INCOSE Tech. Prod.*, **281**(6), 2004.
19. INCOSE and Wiley, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Fourth Edition, John Wiley & Sons, Inc., 2015.
20. ABET, Criteria for Accrediting Engineering Programs, 2018. [Online]. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-programs-2019-2020/#GC2.
21. C. Ng, Findings From a Web-based Survey of Degree Programs in Engineering Systems By Christine Ng, 2003.
22. P. Godfrey, R. D. Crick and S. Huang, Systems thinking, systems design and learning power in engineering education, *Int. J. Eng. Educ.*, **30**(1), pp. 112–127, 2014.
23. N. J. Hayden, D. M. Rizzo, M. M. Dewoolkar, M. D. Neumann, S. Lathem and A. Sadek, Incorporating a systems approach into civil and environmental engineering curricula: Effect on course redesign, and student and faculty attitudes, *Adv. Eng. Educ.*, **2**(4), pp. 1–28, 2011.
24. M. Lammi and T. Thornton, The Cognitive Processes and Strategies of an Expert and Novice in the Design of a Wireless Radio Frequency Network, *J. Technol. Stud.*, **39**(2), pp. 92–101, 2013.
25. G. Haupt, Learning from experts: fostering extended thinking in the early phases of the design process, *Int. J. Technol. Des. Educ.*, **25**(4), pp. 483–520, 2015.
26. R. A. Dixon and S. D. Johnson, Experts vs. novices: Differences in how mental representations are used in engineering design, *J. Technol. Educ.*, **23**(1), pp. 47–65, 2011.
27. T. Song, Expert vs. Novice: Problem Decomposition/Recomposition in Engineering Design, Utah State University, 2014.

28. C. J. Atman, R. S. Adams, M. E. Cardella, J. Turns, S. Mosborg and J. Saleem, Engineering Design Processes: A Comparison of Students and Expert Practioners, *J. Eng. Educ.*, no. October, 2007.
29. C. H. Ho, Some phenomena of problem decomposition strategy for design thinking: Differences between novices and experts, *Des. Stud.*, **22**(1), pp. 27–45, 2001.
30. K. Becker, J. Gero, Y. Luo, L. Almeida, S. Abdellahi and M. Pourmohamadi, Empirical Foundations for Improved Engineering Education: Differences Between Engineering Students and Professional Expert Engineers while Designing, in *Proceedings of the ASEE Annual Conference & Exposition*, 2019.
31. E. L. Gralla, J. W. Herrmann and M. Morency, Design problem decomposition: an empirical study of small teams of facility designers, *Res. Eng. Des.*, **30**(2), pp. 161–185, 2017.
32. C. Tobias, J. W. Herrmann and E. Gralla, Exploring Problem Decomposition in Design Team Discussions, *Proc. 20th Int. Conf. Eng. Des. (ICED 15), Vol. 1 Des. Life*, no. July, pp. 1–10, 2015.
33. M. T. H. Chi, Two Approaches to the Study of Experts' Characteristics, *Cambridge Handb. Expert. Expert Perform.*, pp. 21–30, 2006.
34. J. S. Gero, Design Prototypes: A Knowledge-Based Schema for Design, *AI Mag.*, **11**(4), pp. 26–36, 1990.
35. J. S. Gero and T. Mc Neill, An approach to the analysis of design protocols, *Des. Stud.*, **19**(1), pp. 21–61, 1998.
36. J. S. Gero and U. Kannengiesser, A function-behavior-structure ontology of processes, *Artif. Intell. Eng. Des. Anal. Manuf. AIEDAM*, **21**(4), pp. 379–391, 2007.
37. F. Husson, S. Lê and J. Pagès, *Exploratory multivariate analysis by example using R*. 2010.
38. J. W. T. Kan and J. S. Gero, *Quantitative Methods for Studying Design Protocols*. GX Dordrecht, The Netherlands: Springer Nature, 2017.
39. V. K. Viswanathan, N. E. Esposito and J. S. Linsey, Training tomorrow's designers: A study on the design fixation, *ASEE Annu. Conf. Expo. Conf. Proc.*, 2012.
40. E. Dringenberg and Ş. Purzer, Experiences of First-Year Engineering Students Working on Ill-Structured Problems in Teams, *J. Eng. Educ.*, **107**(3), pp. 442–467, 2018.
41. M. Corley and R. J. Hartsuiker, The Effect of Prototyping and Critical Feedback on Fixation in Engineering Design, *Proc. Annu. Meet. Cogn. Sci. Behav.*, pp. 807–812, 2011.
42. I. Villanueva, S. Jones, L. Putney and B. Campbell, Puzzling the pieces: Conceptual blocks of engineering student ideas in a service learning project, *Int. J. Eng. Educ.*, **34**(1), pp. 56–68, 2008.
43. D. Jonassen, J. Strobel and C. B. Lee, Everyday problem solving in engineering: Lessons for engineering educators, *J. Eng. Educ.*, **95**(2), pp. 139–151, 2006.
44. D. H. Jonassen and W. Hung, All Problems are Not Equal: Implications for Problem-Based Learning, *Interdiscip. J. Probl. Learn.*, **2**(2), pp. 10–13, 2008.

Appendix A: Design Task, Window Opening Device

Double-Hung (Sash) Window Opener

Your design team has been approached by Warm Heart Estates, a local nursing home, to design a new product to assist its elderly residents.

The nursing home administrators have noticed that changes in humidity during the summer months cause the windows of the 65-year old building to "stick," thus requiring significant amounts of force to raise and lower the window panes. The force required to adjust the windows is often much too large for the nursing home tenants, making it very difficult for them to regulate their room temperature.

Your team has been tasked with designing a device that will assist the elderly tenants with raising and lowering the building's windows. Since each window is not guaranteed to be located near an electrical socket, this device should not rely on electric power.

The building's windows are double-hung (as seen in the figure above). The double-hung window consists of an upper and lower sash that slide vertically in separate grooves in the side jambs. This type of window provides a maximum face opening for ventilation of one-half the total window area. Each sash is provided with springs, balances, or compression weatherstripping to hold it in place in any location.

Your team has identified the following websites as potential sources of useful information:

- "Double Hung Window Construction"
  - http://www.oldhouseweb.com/how-to-advice/double-hung-window-construction.shtml
- "Double Hung Windows – Everything You Need to Know" (1 min. 34 sec.):
  - http://www.youtube.com/watch?v=xW7OMHYI4kY
- American Disabilities Act (ADA) information:
  - http://www.ada.gov/
  - http://www.ada.gov/pubs/adastatute08.htm (full act, as amended in 1990)
  - http://en.wikipedia.org/wiki/Americans_with_Disabilities_Act_of_1990
- ADA Accessibility Guidelines for Buildings and Facilities (ADAAG):
  - http://www.access-board.gov/adaag/html/adaag.htm

Appendix B: Example Data - Systems Processes for Professional Engineers

Systems Processes for Professional Engineers

	Systems Processes											
Session No.	**1→1**	**1→2**	**1→3**	**2→1**	**2→2**	**2→3**	**3→1**	**3→2**	**3→3**	**Decomposition**	**Recomposition**	**Same Level**
1	0.07	0.06	0.02	0.05	0.32	0.14	0.02	0.13	0.20	0.21	0.20	0.59
2	0.05	0.06	0.02	0.06	0.38	0.14	0.03	0.13	0.13	0.22	0.22	0.56
3	0.10	0.07	0.01	0.06	0.25	0.12	0.02	0.11	0.25	0.20	0.19	0.60
4	0.03	0.03	0.01	0.03	0.29	0.14	0.02	0.14	0.31	0.19	0.19	0.62
5	0.03	0.04	0.01	0.03	0.24	0.16	0.02	0.15	0.31	0.21	0.20	0.58
6	0.05	0.05	0.04	0.05	0.25	0.12	0.04	0.12	0.31	0.20	0.20	0.61
7	0.02	0.04	0.01	0.02	0.28	0.15	0.02	0.13	0.33	0.19	0.18	0.64
8	0.04	0.03	0.02	0.02	0.21	0.13	0.02	0.13	0.41	0.17	0.17	0.66
9	0.10	0.05	0.04	0.05	0.24	0.09	0.05	0.09	0.29	0.19	0.18	0.63
10	0.04	0.03	0.01	0.02	0.24	0.12	0.01	0.11	0.42	0.15	0.15	0.70
11	0.06	0.03	0.02	0.02	0.26	0.11	0.03	0.10	0.35	0.17	0.16	0.68
12	0.09	0.06	0.03	0.04	0.19	0.10	0.05	0.08	0.36	0.19	0.17	0.64
13	0.06	0.03	0.03	0.02	0.12	0.08	0.04	0.08	0.54	0.14	0.14	0.72
14	0.08	0.04	0.04	0.04	0.16	0.07	0.04	0.07	0.45	0.15	0.15	0.70
15	0.04	0.01	0.03	0.01	0.11	0.06	0.03	0.06	0.66	0.10	0.10	0.80
16	0.16	0.05	0.03	0.05	0.20	0.06	0.03	0.07	0.35	0.14	0.15	0.71
17	0.05	0.04	0.02	0.03	0.15	0.08	0.03	0.08	0.54	0.14	0.13	0.74
18	0.09	0.03	0.03	0.02	0.21	0.06	0.04	0.05	0.47	0.12	0.11	0.76
M	0.06	0.04	0.02	0.04	0.23	0.11	0.03	0.10	0.37	0.17	0.17	0.66
sd	0.03	0.02	0.01	0.02	0.07	0.03	0.01	0.03	0.13	0.03	0.03	0.07

M = Mean, *sd* = standard deviation. Decomposition, Recomposition, and Same Level were calculated by taking the sum of the respective systems processes where:

Decomposition = systems processes 1→2, 1 → 3, 2 →3. Recomposition = 2 →1, 3 →1, 3 →2. Same Level = 1 →1, 2 →2, 3 →3.

Yuzhen Luo joined the Engineering Education Department at Utah State University as a PhD student in 2016. His research interests are in areas of systems thinking and engineering design cognition. During his PhD program, Yuzhen taught an introductory course for computer engineering drafting, where he was recognized and awarded the Graduate Student Teacher of the Year award from the College of Engineering at Utah State University. He had recently graduated from his program and is seeking to build his professional career.

Kurt H. Becker is a Professor in the Department of Engineering Education at Utah State University where his research focus is in the area of engineering design thinking. He currently is working on a National Science Foundation (NSF) funded project quantifying differences between professional expert engineers and engineering students designing. His other areas of research include adult learning cognition, engineering education professional development and technical training. He has extensive international experience working on technical training and engineering projects funded by the Asian Development Bank, World Bank, and U.S. Department of Labor, USAID. Countries where he has worked include Armenia, Bangladesh, Bulgaria, China, Macedonia, Poland, Romania, and Thailand. In addition, he taught undergraduate and graduate courses in engineering education for the department.

John S. Gero is a Research Professor in Computer Science and Architecture at the University of North Carolina, Charlotte, and a Research Professor at the Krasnow Institute for Advanced Study and at the Department of Computational Social Science, George Mason University. Formerly he was Professor of Design Science and Co-Director of the Key Centre of Design Computing and Cognition, at the University of Sydney. He is the author or editor of 54 books and over 700 papers and book chapters in the fields of design science, design computing, artificial intelligence, computer-aided design, design cognition and design neurocognition. He has been a Visiting Professor of Architecture, Civil Engineering, Cognitive Science, Computer Science, Design and Computation or Mechanical Engineering at MIT, UC-Berkeley, UCLA, Columbia and CMU in the USA, at Strathclyde and Loughborough in the UK, at INSA-Lyon and Provence in France and at EPFL-Lausanne in Switzerland. His former doctoral students are professors in the USA, UK, Australia, Finland, India, Japan, Korea, New Zealand, Singapore and Taiwan.

Idalis Villanueva, PhD, joined the Engineering Education Department in the University of Florida at Gainesville in summer 2020 as an Associate Professor. Prior to this position, she was an Assistant Professor of Engineering Education at Utah State University and before that, she was a Lecturer in the Fischell Department of Bioengineering at the University of Maryland at College Park. She has a PhD in Chemical and Biological Engineering from the University of Colorado-Boulder and a postdoctoral degree in Analytical Cell Biology from the National Institutes of Health. Her research interests include hidden curriculum in engineering, mentoring of minoritized students and faculty in science and engineering, and study of motivation and learning pathways in science and engineering education using mixed- and multi-modal tools (biological and physiological). In 2019, she received the Presidential Early Career Award for Scientists and Engineers (PECASE) award for her 2017 NSF CAREER project on hidden curriculum in engineering.

Oenardi Lawanto, PhD, is an associate Professor in the Department of Engineering Education at Utah State University. His research interests are in the areas of Cognition, E-Learning, Learning, Problem-Solving, Design, and his teaching interests are in the areas of Electric Circuits for Non-Electrical Engineering Students, Cognition in Engineering Education, and Developing Online Educational Curriculum. He has received numerous awards for excellent research, teaching, and service from Utah State University, National Science Foundation, University of Illinois at Urbana-Champaign, World Bank Institute and Singapore Polytechnic, World Bank Institute, Asian Institute of Technology, and University of Science Malaysia.