

Problem Decomposition and Recomposition in Engineering Design: A Comparison of Design Behavior Between Professional Engineers, Engineering Seniors, and Engineering Freshmen

Ting Song, Kurt Becker, John Gero, Scott DeBerard, Oenardi Lawanto, & Edward Reeve

Abstract

The authors investigated the differences in using problem decomposition and problem recomposition between dyads of engineering experts, engineering seniors, and engineering freshmen. Participants worked in dyads to complete an engineering design challenge within 1 hour. The entire design process was video and audio recorded. After the design session, members participated in a group interview. Video and audio data were transcribed, segmented, and coded to make comparisons. Results show differences between engineering experts, seniors, and freshman in design thinking. Students tend to use depth-first decomposition, and experts tend to use breadth-first decomposition in engineering design. The results also show that students spend less cognitive effort on the problem-definition stage than engineering experts.

Keywords: engineering design; problem decomposition and recomposition; design thinking; expertise.

Design is recognized as the critical element of engineering thinking which differentiates engineering from other problem-solving approaches (Dym, Agogino, Eris, Frey, & Leifer, 2005). One of the primary goals of engineering design education is to equip students with the capability to become expert design engineers. To develop this capability in students, educators require a detailed knowledge of the cognitive behavior of both undergraduate students and expert design engineers. However, there is insufficient information about the cognitive behavior of expert design engineers because most studies are focused on individual student engineers or early professional engineers.

Engineering design is fundamental for engineering graduates because engineering design is a major skill required of practicing engineers. The use of design strategies plays a significant role in engineering design, and a commonly used strategy is problem decomposition or recomposition. It is frequently used by experienced engineers, especially for solving complex engineering problems (Vincenti, 1990). The process of *problem decomposition* involves breaking the design problem into smaller independent subproblems (Arvanitis, Todd, Gibb, &

Orihashi, 2001). Each subproblem can be further broken into even smaller problems (Arvanitis et al., 2001), and the decomposition process stops when designers can directly approach each subproblem. *Problem recomposition* is a bottom-up process that usually comes with problem decomposition. It is a process of recomposing all subsolutions (Chandrasekaran, 1990) in the premise of satisfying requirements of the combining design (Hall, Jackson, Lanney, Nuseibeh, & Rapanotti, 2002). Instead of focusing on a complex design problem as a whole, engineers can work on several smaller, more approachable subproblems using this process, which makes the process of engineering design more efficient. Studies have identified a gap between engineering novices and engineering experts when it comes to problem decomposition and recomposition skills in engineering design (Ball, Evans, & Dennis, 1994; Ho, 2001; McCracken, 1997).

To the extent that past works are available (e.g., Ball et al., 1994; Ho, 2001; McCracken, 1997), most studies about problem decomposition or recomposition have focused on individuals instead of groups. However, in the real world, engineers usually work in groups to solve engineering problems. By investigating this topic in the context of collaborative engineering design, researchers can have a better understanding of the development of expertise and the use of problem decomposition or recomposition in practical settings. Design is a creative, open-ended, and experiential process that aims at problem solving. Engineering design is a central part of engineering and has been emphasized as a focus for engineering education for several decades (Dym et al., 2005). Engineering design challenges are widely and effectively used in teaching engineering and in engineering education research. Engineering design challenges can be used in both formal academic circumstances and informal settings. Dym, Agogino, Eris, Frey, and Leifer (2005) believed that engineering design challenges could benefit student learning in many ways. In theory, these challenges should include the entire engineering design process, but practical engineering design challenges are extremely complex and ill structured. Classroom educators sometimes only incorporate parts of the design process based on the needs of the curriculum (Atman et al., 2007; Katehi, Pearson, & Feder, 2009).

Research Design

To investigate the gap that exists between skills developed in universities and skills needed in the industry to become an expert engineer, a pilot study was conducted. The research question guiding this research is: In the process of engineering design, how do experts approach the design problem differently from engineering students?

Sample

In this study, participants were selected using a convenience sampling method (Gall, Gall, & Borg, 2007). Fifty participants took part in this study, including 20 college engineering freshmen, 20 engineering seniors, and 10 engineering experts. All of the participants worked in dyads. It should be noted that this research was a pilot study; therefore, data were collected from a small sample. Results of the quantitative data show preliminary findings only and cannot be generalized because the N size is small.

Design Challenge

All dyads completed the same open-ended engineering design challenge. The design challenge used was a double-hung window opener that would assist the elderly in raising and lowering windows. This design challenge has been used by other researchers to study engineering design (Gero 2010; Lammi & Becker, 2013). There were various engineering and social constraints in this challenge, which made it a typical engineering design challenge. During the design session, participants had access to only five websites related to the design challenge. Participants had limited access to prevent them from searching for solutions to the design problem. They were recommended 1 hour to complete the design challenge. Participants only submitted design proposals as their final outcome. Participants received no instruction about the form or the content of the proposals. They did not build, test, and analyze their design because of the time constraint.

Data Collection

The primary form of data collection was protocol analysis. In the process of engineering design, conversation happened naturally within the dyads. The researcher used audio and video recording to capture participants' conversations and their nonverbal interactions. The researcher did not answer participants' questions. The audio and video data complemented each other to provide rich information about the conversations and actions in engineering design process. The protocol analysis of design sessions was coded based on the function-behavior-structure (FBS) coding scheme. This ontology provides a set of irreducible foundational concepts of design and designing, which covers the acts of designing and the representation of the design. The definition and conceptualization of the ontology is illustrated in Figure 1. *Function* (F) represented designers' expectations of the products, *behavior* (B) represented the ways that designers accomplish their goals, and *structure* (S) represented the solutions to the problem.

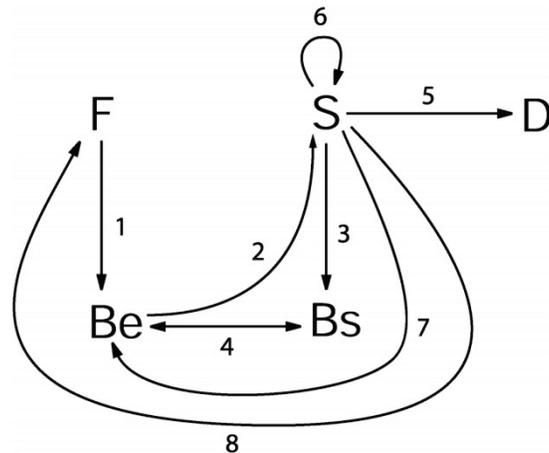


Figure 1. FBS framework (Resource: Gero, Kan, & Pourmohamadi, 2011).

The design actually is a consequence of a series of processes including the above FBS variables:

1. *Formulation* (process 1) transforms the design requirements, expressed in function (F), into behavior (Be) that is expected to enable this function.
2. *Synthesis* (process 2) transforms the expected behavior (Be) into a solution structure (S) that is intended to exhibit this desired behavior.
3. *Analysis* (process 3) derives the “actual” behavior (Bs) from the synthesized structure (S).
4. *Evaluation* (process 4) compares the behavior derived from structure (Bs) with the expected behavior to prepare the decision if the design solution is to be accepted.
5. *Documentation* (process 5) produces the design description (D) for constructing or manufacturing the product.
6. *Reformulation type 1* (process 6) addresses changes in the design state space in terms of structure variables or ranges of values for them if the actual behavior is evaluated to be unsatisfactory.
7. *Reformulation type 2* (process 7) addresses changes in the design state space in terms of behavior variables or ranges of values for them if the actual behavior is evaluated to be unsatisfactory.
8. *Reformulation type 3* (process 8) addresses changes in the design state space in terms of function variables or ranges of values for them if the actual behavior is evaluated to be unsatisfactory. (Gero & Kannengiesser, 2004, p. 3)

Under the FBS ontology, there was another coding system to represent the level of the problem. Typically, engineers decompose the design problem into multiple subproblems and work on each subproblem in order to find a solution. The level of the problem ranged from 1 to 3. The meaning of each number is shown in Table 1. Gero and Mc Neill (1998) adopted this coding system in analyzing design protocols. Ho (2001) used a similar coding system to investigate engineering design strategies used by individual electrical engineers. Few studies have used levels of the problem to code; as such, there is little data available for this study, but Ho (2001) collected very similar data.

Table 1
Level of the Problem

Level of the problem	Definition
1: System	Designers focused on the problem as an integral whole.
2: System and subsystems	Designers focused on interactions between subsystems.
3: Subsystems	Designers focused on details of the subsystems.

Immediately after completing the design challenge, participants took part in a focus group interview in which they answered questions. During this semistructured interview, the researcher asked questions about how participants framed the problem, generated alternative solutions, reached agreements, and used strategies. Table 2 (continued on next page) shows the guiding interview questions. Participants' sketches were also collected as a data resource.

Table 2
Interview Guiding Questions

Number	Interview Question
1	How did you define the problem?
2	How did you decide what information to get?
3	How did you develop or come across different ideas (solutions)?
4	How did you know which ideas would work and which would not work?

-
- | | |
|---|--|
| 5 | Why and how did you choose your final idea or plan? |
| 6 | Is there anything else you needed or wanted that would have helped you? |
| 7 | Did you tackle the problem as a whole or decompose it into several subproblems? If you decomposed it, why did you choose it over the other one |
| 8 | What difficulties did you meet in solving the problem? |
-

Data Analysis

Coder training. Prior to analyzing data, two coders were trained to use the coding systems in order to reach an ideal intercoder reliability. They learned the coding systems and started coding sample data from previous studies separately. After coding separately, they compared their codes and calculated the percentage of the codes that they coded the same, which was the intercoder reliability. They also discussed the segments that they coded differently to reach a consistent understanding of the coding scheme. They repeated this process for several rounds until the intercoder reliability remained above 80%. In the social sciences, 70% intercoder reliability is acceptable (Schloss & Smith, 1999).

Data transcribing and segmenting. After participants completed the design challenge, the researcher manually transcribed participants' conversations and movements into spreadsheets. The spreadsheet data containing participants' conversations and movements were further broken into segments based on design issues. Each segment is a coding unit and can only contain one code.

Coding. Coders started the coding process by using the FBS ontology. Table 3 shows a piece of coded data and how coders arbitrated data. After all data were coded using the FBS ontology, Codes D, R, and O were excluded from being coded before coding the level of the problem because they do not pertain to levels of the problem. Code O is about other issues that are not related to design cognition. Code R is the requirement that is given to designers. Code D is the documentation process; the spread sheet didn't record the content of designers' sketches or drawing, so it was excluded as well. Table 4 shows a piece of data coded by two coders. The coding of "levels of the problem" was based on codes of the FBS ontology. The last column shows arbitrated codes, which are also the final codes of "levels of the problem."

Table 3
Example of FBS Codes and Arbitrated Codes

Subject	Utterance	FBS: Coder 1	FBS : Coder 2	FBS: Final code
A	Not what I'm asking, but like how in-depth?	F	R	F
A	Because that's like how I'm in senior drawing ...	O	O	O
A	Like a pulley is just something you go to the store and buy. Like you... You know...Based on, like	R	S	R
A	I don't think we are given all the numbers that we need to be able to figure what type of pulley system or what gear ratio.	S	S	S
B	Yes, and the cost of materials	S	S	S
A	Yes, I mean, I wondering this is more just given what we are given. I don't think we really can say we need a specific number	R	R	R
B	I see, like 15 or whatever the number type going here	S	S	S

Table 4
Sample of Codes for Levels of the Problem

Subject	Utterance	FBS: Final code	Levels of the problem: Coder 1	Levels of the problem: Coder 2	Levels of the problem: Final code
A	Not what I'm asking, but like how in-depth?	F	1	1	1
A	Because that's like how I'm in senior drawing ...	O	–	–	–
A	Like a pulley is just something you go to the store and buy. Like you. You know...Based on, like	R	–	–	–
A	I don't think we are given all the numbers that we need to be able to figure what type of pulley system or what gear ratio.	S	3	3	3
B	Yes, and the cost of materials	S	3	1	1
A	Yes, I mean, I wondering this is more just given what we are given. I don't think we really can say we need a specific number	R	–	–	–
B	I see, like 15 or whatever the number type going here	R	–	–	–

The process of problem decomposition and problem recomposition was identified by the change of the level of the problem. Table 5 (continued on next page) shows a piece of sample data of an individual dyad in which problem decomposition and problem recomposition were coded. As previously illustrated, the problem decomposition is a top-down process, whereas the problem recomposition is a bottom-up process. When the level of the problem transitions from a higher level to a lower level, it is defined as the problem decomposition, and when it transitions from a lower level to a higher level, it is defined as the problem recomposition.

Table 5
Example of Problem Decomposition and Problem Recomposition

Subject	Utterance	FBS final code	Levels of the problem final code	Decomposition/recomposition
A	Not what I'm asking, but like how in-depth?	F	1	–
A	Because that's like how I'm in senior drawing ...	O	–	–
A	Like a pulley is just something you go to the store and buy. Like you.. You know...Based on, like	R	–	–
A	I don't think we are given all the numbers that we need to be able to figure what type of pulley system or what gear ratio.	S	3	D
B	Yes, and the cost of materials	S	1	R
A	Yes, I mean, I wondering this is more just given what we are given. I	R	–	–

	don't think we really can say we need a specific number			
B	I see, like 15 or whatever the number type going here	R	–	–

The numbers of utterances generated by each dyad were different, so simply comparing the frequencies of each type of code would affect the validity of the study. The percentages of codes were used in order to compare the differences between dyads. The percentage of each code from each dyad was calculated by dividing the frequency of the code into the total number of effective codes of the dyad.

Interview and sketch data. The analysis of qualitative data was connected with quantitative data. The constant comparative method (Glaser, 1965) was adapted in analyzing data. The first step included going through each interview and sketch and themes were recorded. The second step was sorting themes to different categories. The third step was looking for differences between expert dyads, senior dyads, and freshman dyads within each category. The analysis of qualitative data allowed any themes or new phenomena to emerge that could not be discovered by analyzing quantitative data. Table 6 provides a small piece of interview data in order to show the first three steps of constant comparative method used in analyzing qualitative data. There were many themes and categories that emerged from the entire qualitative data, and some themes emerged repeatedly. The conclusions drawn in Step 3 were based on the analysis of the entire data instead of the small piece shown in Table 6 (*please note that Table 6 has been divided into several cells for readability purposes*). The fourth step of analyzing qualitative data was writing the theory which is not shown in Table 6. The analysis of sketches followed the same four steps.

Table 6
Example of Qualitative Data Analysis

Interview Question	Step 1: Themes Emerged from Answers F- Freshmen S – Seniors E - Engineers	Step 2: Categorizing	Step 3: Comparing students and engineers
How did you define the problem?	Increase the force (F)	Problem definition	Problem definition: Engineers understood the problem better than students.
	Modify existing window (F)	Problem definition	
	ADA Guidelines (S)	Problem definition	
	Old people in wheel chairs (S)	Problem definition	
	ADA requirements (E)	Problem definition	
	No major construction required (E)	Problem definition	
How did you know which ideas would work and which would not work?	Safety issues (E)	Problem definition	Design Experiences: Engineers had more design experiences and intent to use their experience in solving new problems.
	Daily experiences (F)	Design experiences	
	Don't know if the ideas would work and how much it would cost (F)	Cost	
	Analyze pros and cons (S)	Alternative solutions	
	Comparing the design with devices used before (E)	Design experiences	
Analyze clients (E)	Problem definition		

	Practical and feasible (F)	Alternative solutions	
	Easy to work (F)	Alternative solutions	
Why and how did you choose your final idea or plan?	Consider how solutions fit requirements (S)	Alternative solutions	Cost: Students did not pay enough attention to the cost of the design. Some of them did not know the cost of materials.
	Only came up with one idea (S)	Alternative solutions	
	Cost effective (S)	Cost/ Problem definition	
	Easy to use and maintenance (E)	Problem definition	
	Fits the goal of the design (E)	Alternative solutions	
	Not block views (E)	Problem definition	
	Have difficulties deciding which solution to choose (F)	Alternative solutions	
	They are good at solving homework problems but not the real problems. (F)	Homework problem and real life problems	
What difficulties did you meet in solving the problem?	The design challenge is very different from problems in class. (F)	Homework problem and real life problems	Alternative solutions: Engineers evaluated alternative solution more effective than students.
	Need more constraints and criteria (S)	Problem definition	
	They don't know the cost of materials (S)	Cost	
	They need more information about clients' design consideration (E)	Problem definition	

They wanted to talk to a window producer to make a product of their design (E) Design experiences

Results

The researchers calculated frequencies of using problem decomposition and problem recomposition in students and engineer dyads. The means and standard deviations of percentages of using problem decomposition and problem recomposition are shown in Table 7. From these numbers, we can determine that engineer dyads used more problem decomposition more than engineering freshmen and seniors. In order to see if there are any statistically significant differences existing, *p*-tests were conducted, effect sizes (ES) were calculated. The results of statistical tests are shown in Table 8. In the use of problem decomposition, freshmen used problem decomposition as much as seniors did in engineering design. Engineers used more problem decomposition than both freshmen and seniors did in engineering design. In the use of problem recomposition, the results are similar; freshmen used problem recomposition as much as seniors did in engineering design, and engineers used more problem recomposition than both freshmen and seniors.

Table 7
Means and Standard Deviations of Problem Decomposition and Problem Recomposition

Type of Dyad	Problem decomposition (%)		Problem recomposition (%)	
	Mean	Standard Deviation	Mean	Standard Deviation
Freshmen (N=10)	15.13	2.66	15.28	2.64
Seniors (N=10)	15.22	3.27	15.31	3.14
Engineers (N=5)	22.38	3.01	22.57	3.54

Table 8
Comparisons of Problem Decomposition and Problem Recomposition

Types of dyads compared	Problem decomposition		Problem recomposition	
	<i>p</i> value	<i>Effect Size</i>	<i>p</i> value	<i>Effect Size</i>
Freshmen vs. seniors	.496	N/A	.980	N/A
Freshmen vs. engineers	.000**	2.55	.001**	2.33
Seniors vs. engineers	.010**	2.28	.001**	2.17

** $p \leq 0.01$.

The analysis of qualitative data had similar findings. In the interview, a question was “Did you tackle the problem as a whole or decompose it into several subproblems? If you decomposed it, why did you choose this over the problem as a whole?” Students’ answers varied from dyad to dyad. Some dyads broke the problem into multiple subsystems, some dyads solved the problem as a whole because they thought the problem was too simple to break down, and some student dyads did both. For engineer dyads, the answers were more consistent. They started with considering the whole problem to get the big picture, then broke it into small pieces to work on, and finally combined small pieces into the final solution.

The study also compared the effort spent on different levels of the problem. These results are shown in Table 9. Means and standard deviations from the three types of dyads show that freshmen, seniors, and engineers spent most of their cognitive effort on Level 3 (details of subsystems). All three types of dyads spent the least amount of cognitive effort on Level 1 (system). In order to see if there are any statistically significant differences existing, a series of tests were conducted, and the results of statistical tests of are shown in Table 10.

Table 9
Means and Standard Deviations of Levels of the Problem

Type of dyad	Level 1 (%)		Level 2 (%)		Level 3 (%)	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
Freshmen (N = 10)	11.08	2.46	17.42	3.99	71.50	4.87
Seniors (N = 10)	12.59	3.57	16.75	4.61	70.66	4.37
Engineers (N = 5)	20.89	13.68	23.79	6.63	55.32	15.66

Table 10
Comparisons of Cognitive Effort on Different Levels of the Problem

Types of dyads compared	Level 1		Level 2		Level 3	
	p value	Effect Size	p value	Effect Size	p value	Effect Size
Freshmen vs. seniors	.286	N/A	.732	N/A	.842	N/A
Freshmen vs. engineers	.020*	1.00	.035*	1.16	.009**	1.40
Seniors vs. engineers	.043*	0.83	.031*	1.23	.009**	1.33

* $p \leq 0.05$.

** $p \leq 0.01$.

On Level 1, which indicates the designer considering the problem as an integral whole, results show that freshman dyads and senior dyads spent the same amount of cognitive effort when they considered the problem as an integral whole. Engineer dyads spent more cognitive effort than freshman dyads and senior dyads when they considered the problem as an integral whole. On Level 2, which indicates designer considering interactions between subsystems, engineers spent more cognitive effort than freshmen and seniors did when they considered interactions between subsystems in engineering design. On Level 3, which indicates designer considering details of subsystems, engineers spent less cognitive effort than freshmen and seniors did when they considered details of subsystems in engineering design.

In the interview, participants were asked how they defined the problem. Engineer dyads considered many more factors in this stage. They defined the problem by thinking about both the problem and their client's needs. All of them made sure that the design met requirements from Americans with Disabilities Act (ADA). They also considered safety issues, aesthetic issues, maintenances of the device, implementation, and cost. Some of them even considered the noises generated by the device because the device would be used in a nursing home in which a quiet environment is preferred. When freshman dyads defined the problem, the focus was to assist in opening the window. Two dyads mentioned clients of the design. A few dyads talked about the ADA, but most of them ignored ADA standard. Most senior dyads focused on the device itself, although they did better than freshman dyads. Most of them were aware of ADA standards, and a few dyads mentioned cost effectiveness as one of their criteria.

Discussion and Implications

The results showed that engineer dyads used problem decomposition and problem recomposition more than senior dyads and freshmen dyads. Qualitative data from interviews also support this result. In spite of differences in research settings, the results of this study are consistent with Ho's (2001) study. Both studies suggested that there is a gap in using problem decomposition and recomposition between experts and novices. In fact, in interviews with engineer participants, they emphasized the importance repeatedly.

Although problem decomposition and recomposition are crucial strategies in engineering design, the results of this study showed that there was no difference between freshman dyads and senior dyads using this strategy in engineering design. This would suggest that, throughout the engineering program, students do not learn adequate knowledge about problem decomposition and problem recomposition; hence, students in the first year of the engineering program perform similar to students about to finish the engineering program.

The results of the study showed that engineer dyads, senior dyads, and freshmen dyads all spent the most cognitive effort on Level 3 and the least cognitive effort on Level 1. The quantitative data showed that on Level 1, engineer dyads spent more cognitive effort than senior dyads and freshman dyads. On Level 2, engineer dyads spent more cognitive effort than senior dyads and freshman dyads. On Level 3, engineer dyads spent less cognitive effort than senior dyads and freshman dyads.

Past studies identified two types of problem decomposition: the breadth-first approach and the depth-first approach. The breadth-first decomposition approach focuses on exploring various solutions of each subproblem and avoids deep exploration to any specific solution in the early stage, whereas depth-first decomposition tends to explore a specific subproblem in detail before other subproblems are investigated (Ormerod & Ridgway, 1999). In this research,

Level 3 represented designers considering details of subproblems. Student dyads spent more cognitive effort on this level because most of them used depth-first decomposition and spent a majority of cognitive effort exploring details of a certain subproblem. Engineer dyads used a breadth-first approach. Unlike student dyads, the distribution of their cognitive effort was more balanced across three levels of the problem.

A series of interesting findings emerged from the interviews and the analysis of participants' sketches as well. In the process of generating alternative solutions, student dyads tended to generate too many or too few solutions compared with engineering dyads. Some dyads only generated one solution and finished their design at a premature stage. They did not make use of the time that they could have used to optimize their design. When examining engineering curriculum, we find that, in most courses, students are taught to generate only one solution instead of multiple ones. It also explains why some dyads only generated one solution through the entire design. For those dyads who generated too many alternative solutions, they spent a lot of time analyzing solutions, which lead them to either go way beyond the time limitation or to haphazardly select a final solution at the end of the design period.

In analyzing qualitative data, engineers were found to be more comfortable working in groups than students were. Student dyads had various difficulties when they worked together. A freshman dyad of students expressed their inadequacy in understanding each other's ideas. Another freshman dyad of students had disagreements about which final solution to choose, which cost them a lot of time. A few senior dyads pointed out that they did not make good use of their time by working individually on different tasks at the same time. Typically, engineering students take foundational engineering courses before taking design classes. Most engineering fundamental courses focus on learning mathematical and scientific theories, which does not provide enough opportunities for students to work on team projects. This may be the main reason why some freshman dyads in this study had issues working with each other. As engineering students move forward in their program, they take design classes and participate in group projects, which explained why senior dyads performed better than freshman dyads when it comes to working in groups. However, the performance of senior dyads was still very different from the performance of engineer dyads. This finding is consistent with a series of previous studies (Holcombe, 2003; Meier, Williams, & Humphreys, 2000; Sageev & Romanowski, 2001; Scott & Yates, 2002).

Implications

Engineering design has always been a significant content area in engineering education. Problem decomposition and recomposition strategies are frequently used by professional engineers. The results of this study showed that there is a gap between engineering students and engineering experts in using

problem decomposition and problem recomposition. In addition, no differences were found between engineering freshmen and engineering seniors, which indicates that students did not learn the skills of problem decomposition and problem recomposition in their undergraduate study. In order to better prepare students for future careers, it is extremely important to incorporate this content into engineering education. There is a need to develop supplemental teaching materials featuring problem decomposition and problem recomposition.

Considering the design problem as a whole was a common practice among professional engineers. They would analyze the big picture of the design problem, and this process is part of the problem-definition stage in engineering design. Problem definition is the first stage of engineering design. This study found that students spent significantly less time on this stage compared with engineering experts. This conclusion is consistent with previous studies (Atman et al., 2007; Jain & Sobek, 2006). Both freshman and senior dyads were found to spent significantly less effort in defining the problem, which implies that engineering education should place more importance on teaching problem definition in general.

References

- Arvanitis, T. N., Todd, M. J., Gibb, A. J., & Orihashi, E. (2001). Understanding students' problem-solving performance in the context of programming-in-the-small: An ethnographic field study. In *Proceedings of the Frontiers in Education Conference* (Vol. 2, pp. F1D-20-23vol.2). Washington, DC: IEEE Computer Society.
- Atman, C. J., Adams, R. S., Cardella, M. E., Turns, J., Mosborg, S., & Saleem, J. J. (2007). Engineering design processes: A comparison of students and expert practitioners. *Journal of Engineering Education*, 96(4), 359–379. doi:10.1002/j.2168-9830.2007.tb00945.x
- Ball, L. J., Evans, J. St. B. T., & Dennis, I. (1994). Cognitive processes in engineering design: A longitudinal study. *Ergonomics*, 37(11), 1753–1786. doi:10.1080/00140139408964950
- Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI Magazine*, 11(4), 59–70. doi:10.1609/aimag.v11i4.857
- Dym, C. L., Agogino, A. M., Eris, O., Frey, D. D., & Leifer, L. J. (2005). Engineering design thinking, teaching, and learning. *Journal of Engineering Education*, 94(1), 103–120. doi:10.1002/j.2168-9830.2005.tb00832.x
- Gall, M. D., Gall, J. P., & Borg, W. R. (2007). *Educational research: An introduction* (8th ed.). Boston, MA: Pearson, Allyn & Bacon.
- Gero, J. S. (2010). *Generalizing design cognition research*. In K. Dorst, S. C. Stewart, I.
- Staudinger, B. Paton, & A. Dong (Eds.), *DTRS 8: Interpreting design thinking* (pp. 187–198). Sydney, Australia: University of Technology Sydney.

- Gero, J. S., Kan, J. W. T., & Pourmohamadi, M. (2011). Analysing design protocols: Development of methods and tools. In A. Chakrabarti (Ed.), *Research into design: Supporting sustainable product development* (pp. 3–10). Bangalore, India: Research Publishing.
- Gero, J. S., & Kannengiesser, U. (2004). The situated function–behaviour–structure framework. *Design Studies*, 25(4), 373–391. doi:10.1016/j.destud.2003.10.010
- Gero, J. S., & Mc Neill, T. (1998). An approach to the analysis of design protocols. *Design Studies*, 19(1), 21–61. doi:10.1016/S0142-694X(97)00015-X
- Glaser, B. (1965). The constant comparative method of qualitative analysis. *Social Problems*, 12(4), 436–445.
- Hall, J. G., Jackson, M., Lanney, R. C., Nuseibeh, B., & Rapanotti, L. (2002). Relating software requirements and architectures using problem frames. In *Proceedings of the IEEE Joint International Conference on Requirements Engineering* (pp. 137–144). Los Alamitos, CA: Institute of Electrical and Electronics Engineers. doi:10.1109/ICRE.2002.1048516
- Ho, C-H. (2001). Some phenomena of problem decomposition strategy for design thinking: Differences between novices and experts. *Design Studies*, 22(1), 27–45. doi:10.1016/S0142-694X(99)00030-7
- Holcombe, M. L. (2003). ET grads—How’d the transition go? In *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition* (pp. 8.537.1–8.537.6). Washington, DC: American Society for Engineering Education. Retrieved from <https://peer.asee.org/12078>
- Jain, V. K., & Sobek, D. K., II. (2006). Linking design process to customer satisfaction through virtual design of experiments. *Research in Engineering Design*, 17(2), 59–71. doi:10.1007/s00163-006-0018-2
- Katehi, L., Pearson, G., Feder, M. (Eds.). (2009). *Engineering in K–12 education: Understanding the status and improving the prospects*. Washington, DC: National Academies Press. doi:10.17226/12635
- Lammi, M., & Becker, K. (2013). Engineering design thinking. *Journal of Technology Education*, 24(2), 55–77. Retrieved from <https://scholar.lib.vt.edu/ejournals/JTE/v24n2/pdf/lammi.pdf>
- McCracken, W. M. (1997). *Portfolio assessment in design education*. Atlanta, GA: EduTech Institute and College of Computing, Georgia Institute of Technology. Link does not work.
- Meier, R. L., Williams, M. R., & Humphreys, M. A. (2000). Refocusing our efforts: Assessing non-technical competency gaps. *Journal of Engineering Education*, 89(3), 377–385. doi:10.1002/j.2168-9830.2000.tb00539.x
- Ormerod, T. C., & Ridgway, J. (1999). Developing task design guides through cognitive studies of expertise. In S. Bagnara (Ed.), *European Conference on Cognitive Science* (pp. 401–410). Sienna, Italy.

- Sageev, P., & Romanowski, C. J. (2001). A message from recent engineering graduates in the workplace: Results of a survey on technical communication skills. *Journal of Engineering Education, 90*(4), 685–692.
doi:10.1002/j.2168-9830.2001.tb00660.x
- Schloss, P. J., & Smith, M. A. (1999). *Conducting research*. Upper Saddle River, NJ: Prentice Hall.
- Scott, G., & Yates, K. W. (2002). Using successful graduates to improve the quality of undergraduate engineering programmes. *European Journal of Engineering Education, 27*(4), 363–378. doi:10.1080/03043790210166666
- Vincenti, W. G. (1990): *What engineers know and how they know it: Analytical studies from aeronautical history*. Baltimore, MA: Johns Hopkins University Press.

About the Authors

Ting Song (songitaly@gmail.com) is Professor in the Department of Applied Science at South Puget Sound Community College in Olympia, Washington.

Kurt Becker (kurt.becker@usu.edu), is Professor in the Department of Engineering Education at Utah State University.

John Gero (jgero@gmu.edu), is Research Professor in the Krasnow Institute for Advanced Study at George Mason University & University of North Carolina.

Scott DeBerard (scott.deberard@usu.edu), is Associate Professor in the Department of Psychology at Utah State University.

Oenardi Lawanto (olawanto@usu.edu), is Professor in the Department of Engineering Education at Utah State University.

Edward Reeve (ed.reeve@usu.edu), is Professor in the Department of Technology and Engineering Education at Utah State University.