Proceedings of the ASME 2016 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2016
August 21-24, 2016, Charlotte, USA

IDETC2016-60397

IDENTIFICATION OF SUBPROBLEMS IN COMPLEX DESIGN PROBLEMS: A STUDY OF FACILITY DESIGN

Azrah Azhar*
Erica L. Gralla
Engineering Management and Systems Engineering
The George Washington University
Washington, District of Columbia 20037
Email: azrah@gwu.edu, egralla@gwu.edu

Connor Tobias
Jeffrey W. Herrmann
Department of Mechanical Engineering
and Institute for Systems Research
University of Maryland
College Park, Maryland, 20742
Email:ctobias@umd.edu, jwh2@umd.edu

ABSTRACT

Many design problems are too difficult to solve all at once; therefore, design teams often decompose these problems into more manageable subproblems. While there has been much interest in engineering design teams, no standard method has been developed to understand how teams solve design problems. This paper describes a method for analyzing a team's design activities and identifying the subproblems that they considered. This method uses both qualitative and quantitative techniques; in particular, it uses association rule learning to group variables into subproblems. We used the method on data from ten teams who redesigned a manufacturing facility. This approach provides researchers with a clear structure for using observational data to identify the problem decomposition patterns of human designers.

INTRODUCTION

Decomposition is an important component of design processes. When solving a complex design problem, designers often make decisions in stages because the problem may be too complex to solve all at once. To enable this approach, the problem is decomposed into manageable subproblems. The specific set of subproblems used by designers to solve complex problems may affect the quality of their solutions. These decomposition strategies may be a predefined formal process set up by the organization or a set of informal activities and responsibilities determined by a team of designers. In either case, various problem decomposition strategies could be used, resulting in final solutions of varying quality. Therefore, identifying decomposition strategies used by design teams when solving complex design problems will give us the ability to compare how various strategies affect the quality of the outcome.

Identifying the decomposition strategies of design teams requires observing and analyzing their activities. However, there is no standard method to "capture" the decompositions used by teams of human designers. Designers typically do not explicitly describe the subproblems that they are solving [1]; the decomposition must be inferred based on their discussions about the variables that need to be determined during a design process. Although verbal protocol analysis is often used to describe the activities of a design team [2], there is no standard method to use a design team's discussions to identify different decomposition patterns of multiple teams over time. Therefore, we developed such a method, which results in a set of subproblems that group together variables that the teams likely considered together, and a timeline that shows which subproblems that team was working on at each point in time.

Although it is general and can be applied to other design domains, the method presented here was developed as part

*Address all correspondence to this author.

of a study examining how teams of professional engineers redesigned a manufacturing facility to make it more efficient. Each team's design activities were video-recorded. The first step in the method requires developing a set of codes using qualitative data analysis methods to describe and code team activities. Next, association rule learning is used to identify the subproblems that the team considered. These subproblems are mapped on timelines of each team to visualize and compare their decomposition patterns. The visualization that we develop using timelines helps to "see" the process as a design team solves a complex problem. Our method enables the development of timelines of activities at various levels of aggregation, which could be useful for many kinds of analysis of design. This will eventually enable researchers to study the impact of decomposition on solution quality, on the time to solve a problem, and on other important metrics relevant to design and design processes.

The following sections describe related work, discuss the experimental setting in which we developed the proposed method, present the subproblem identification method, and describe the results of using the method on the data we collected.

RELATED WORK

The following subsections describe related work in methods for observing and analyzing design processes, then in understanding how designers decompose problems.

Methods for Observing Design Processes

The attempt to understand cognitive decision-making processes has a long history in research from fields such as artificial intelligence [3]. In order to understand the cognitive decision-making pattern of designers or design teams, researchers have paid attention to the verbal and nonverbal gestures during the design process. The most common method to capture the content of the design team discussions ins audio and/or video recordings [4–6]. Collecting relevant notes and sketches used by the design team also helps understand the design thought process [7].

Methods for Analyzing Design Processes

Data from observing design processes are typically unstructured. This is a common issue with process data, which "are messy and making sense of them can be a challenge" [8]. Verbal protocol analysis has been a commonly used method among design researchers to investigate observational data of design teams. Design researchers have transcribed team and individual discussions and developed coding schemes to analyze these transcribed data [4–7, 9, 10]. Coding schemes have focused on issues such as "problem domains" and "search strategies" [6], which "sub goals" are pursued by teams [7] or "communicative acts" and "solution idea" pursued by the designers [4]. While these issues are related to decomposition, they do not identify the specific subproblems, or groupings of variables considered together, used by the team as they come up with the design solution. The identification of specific subgoals, in particular, is related to sub-problems. There is a concern that in defining subproblems researchers develop various subcategories or sub goals and then analyze the data based on these predefined categories [2]. We attempt to build on these existing methods and to address this concern, by investigating a method that would help us generate sets of subproblems unique to individual design teams from the data of team discussions.

Understanding Decomposition in Design

The design literature has focused on many aspects of how teams or individual designers solve a problem. Differences between solution strategies of novices and experienced designers [6, 7, 11–13]; differences in thinking approach between design teams [14, 15]; design decision making process in software design [16]; effect of time spent on analyzing a problem vs. time spent on problem solution [4, 17]; learning progression in the design thinking process [18] are some of the main areas covered in the literature pertaining to teams of designers.

More specifically, the decomposition of complex problems has been widely studied in management science and design literature [19–21], resulting in recommendations about the best types and features of decompositions. Much of this work focuses on formal or "designed" decompositions. We focus on teams that are given no formal decomposition, nor asked to create one, in order to understand how teams intuitively decompose problems. Complex design problems are ill-structured, and therefore teams will not typically have a structured approach to solving them [22, 23]. But to the best of our knowledge no previous studies considered whether teams of designers informally develop different types of subproblems over time, resulting in different decomposition patterns for a given design problem.

EXPERIMENTAL SETTING AND DATA COLLECTION

Our method was developed to identify the design problem decomposition pattern of a design team. We describe here the experimental setting because our method is illustrated using these data. The results of the larger study are published elsewhere [24].

We observed 10 teams consisting of a total of 45 industrial engineers and manufacturing managers redesigning a factory as part of several two-day lean facility design courses. Each team redesigned a fictional factory that makes multiple prod-

ucts and has a traditional functional layout. The exercise took approximately four hours. Each team was given a scenario that specified (a) information about the existing factory and products; (b) constraints, such as areas that cannot be changed; (c) goals for the redesigned factory, such as making space available for other activities; and (d) criteria for evaluating the redesigned factory, including productivity and material handling effort. Teams presented their final designs to each other for discussion and feedback. The activities and discussions of the teams were video recorded as they completed the exercise.

The manufacturing process required building frames, painting them, building various components (five modules of unspecified nature, a drive train, and a control box), and mounting these various components onto the main frame to assemble the final product (a machine). The facility makes small, medium, and large machines that follow the same sequence of operations but use modules of different sizes. Additional functional areas included a machine shop, incoming and outgoing quality control, storage, crating and packaging, and shipping and receiving. In addition to areas for each of these functions, the redesigned facility required an R&D area for developing new products, an area for refurbishing used machines, offices, and a fitness center. Many of these areas were treated as "black boxes," but the assembly operations (building and assembling modules) required determining the detailed layout of these operations.

Field observation with video-recording was our primary data collection method [25, 26]. For each exercise, the complete design activity of each team and their final presentations were video-recorded for the team of researchers to view later. After the first exercise, the teams carried out activities in separate rooms to increase the quality of the captured audio. The video cameras were pointed down at the layouts from overhead, so the video captured what participants drew on the layouts, when they moved areas around, and any other activities carried out on the layouts. We also photographed the final layouts of each team and collected all relevant documents. In order to carry out our analysis method, we found that it was critical that the audio was captured well enough to distinguish the details of the teams' discussions. We found that pointing the camera at the layout was also useful because it enabled us to understand the meaning of statements like "What if we move this here?".

This type of time-limited design exercise is a common way for researchers to study the process of design [2]. Therefore, we believe our method will be useful to others who have similar data. We chose a field experiment because it represents a useful compromise between a completely controlled but highly artificial laboratory experiment and a natural but time-consuming field study [27].

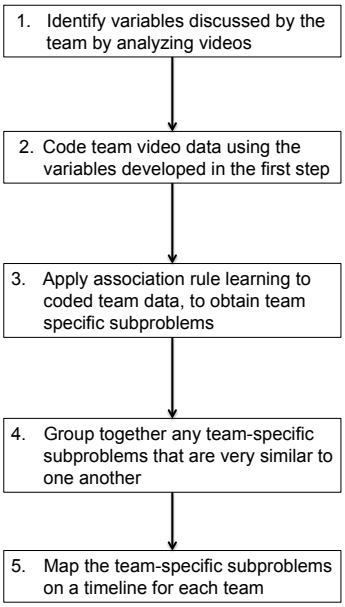

FIGURE 1. FLOWCHART OF THE SUBPROBLEM IDENTIFICATION METHOD.

A METHOD FOR IDENTIFYING SUBPROBLEMS

Our method can be used to generate a timeline that shows which subproblems a team considered and when they considered each subproblem. We define a subproblem as a set of variables considered collectively. The method that we developed has several steps. First, we identified the "variables" that the team discussed and when they discussed them. Second, we determined how those variables were grouped into subproblems, i.e., which collections of variables were usually considered together. Then, with the subproblems identified, we plotted the subproblems on a timeline to show the team-specific decomposition pattern. Figure 1 presents a flowchart of the method. The numbers in the boxes of this figure are the steps described in the following paragraphs, which also mention examples from the the setting that we studied.

Development of Variables and Coding

Developing Variables. We inductively developed a set of codes to represent the variables that the teams considered, to ensure that the codes represent what the teams did rather than a framework imposed by the researchers. While we started with a standard set of codes, they were modified extensively through the analysis process, as described in more detail below. Our method followed standard qualitative data analysis approaches [8, 28].

In Step 1, we developed an initial set of codes to label the variables that the teams considered. For example, facilities are

commonly designed using block layouts that abstract away the complexity of detailed operations, and the problem that was given to the teams included some predefined blocks like the machine shop, so one variable code was "location of machine shop."

Next, we observed videos of a team's discussions and identified the specific variables that the designers discussed. Based on such observations, variables were inductively developed to code the team discussions. The initial codes were refined by re-examining similarly coded data segments, editing, removing, or adding code definitions to better represent the observed behavior of the teams, and re-coding the data in an iterative and inductive process.

We created and refined a code book during these iterations; this document clearly defines each variable to indicate what types of discussion are considered evidence that the team is discussing that variable. For example, the variable "high-level flow logic" should be coded when the video shows teams includes discussing whether to use assembly cells or assembly lines and the number of lines.

Coding of Videos. In Step 2, each team's videos were coded using the identified variables. Four researchers coded a subset of the data to ensure good agreement across researchers, then one researcher coded the remainder of the dataset. The data were coded by examining the teams' discussions and actions to determine which sets of variables were being considered by each team during every two-minute segment of video. For each two-minute time segment, every variable that the team explicitly discussed or determined during that time segment was coded (In the coding spreadsheet for that team, the value "x" was entered in the cells corresponding to that time segment and the coded variables).

Identifying Subproblems

In Step 3, we identified each team's subproblems, which are collections of variables considered together. We used association rules to identify groups of variables that were usually considered at the same time, which suggests that they were considered together by the design team.

Association Rule Learning. In machine learning, association rules are utilized to discern relationships between sets of items that occur together [29]. Association rules identify relationships such as "if a customer buys bread and milk, they are also likely to buy eggs." The technique is typically used on very large datasets, but it can be used on smaller datasets as well. We utilized association rule learning on the coded timelines, in order to identify variables that are frequently coded together (in our case, the directionality of the rules was not relevant). For example, in a team, if the three variables "staffing in area," "operations sequencing and balancing," and "location of areas" were coded together (i.e., in the same two-minute time segment) frequently within the team's coding matrix, then it is likely that these three variables constitute one subproblem. The algorithm in package "arules" in R [30] was used for obtaining frequently coded item sets.

Three measures are typically used when identifying association rules: the support, confidence and lift. The support is the proportion of time segments in which a variable (such as a) is coded for a given team ($Supp(a)$), out of all the time segments in the dataset. The confidence is the proportion of time segments in which if variable a was coded, then b was also coded (see equation 1).

$$Conf(a \Rightarrow b) = \frac{Supp(a \cup b)}{Supp(a)} \tag{1}$$

The lift is the proportion of the observed support of a and b coded together to that expected if a and b were independent (see equation 2).

$$Lift(a \Rightarrow b) = \frac{Supp(a \cup b)}{Supp(a).Supp(b)} \tag{2}$$

These measures indicate the "reliability" of the rule, in that higher measures typically mean it is more likely that the variables are associated. The algorithms used to identify rules within datasets typically require setting cutoffs for these measures. We selected low cutoffs because our dataset was small.

Association rules may produce permutations of the same set of variables as different rules (e.g., $a \Rightarrow b$ and $b \Rightarrow a$ will be generated as two separate rules). However, in our context, we are interested only in whether a and b typically occur together, i.e., that $\{a, b\}$ form a single subproblem. Therefore, we combined such permutations in order to derive a final set of subproblems for each team.

Subproblems from Association Rules. In Step 4, we examined these subproblems and identified those that overlapped (shared one or more variables). For instance, the association rules would sometimes not only group two variables with one rule but also group the same two variables and a third one with another rule. In such cases (and other similar ones), we removed the subproblems with fewer variables and kept the ones with more variables or combined two smaller subproblems into one larger one. For example, we combined the subproblems staffing in area, location of areas, high level flow logic and staffing in area, facility staffing, high level flow logic

because they shared many of the same variables. We also labeled each of the subproblems based on the concept or concepts that distinguished each subproblem from the others. In this example, the subproblem was labeled “staffing.”

Generating Timelines for Each Team.

In Step 5, we created a timeline divided into two-minute segments and rows for each subproblem. In each row, the timeline indicates each and every two-minute segment in which at least two variables in that subproblem were discussed. This method of creating timelines assumes that design teams discuss subproblems without discussing every variable included in the subproblem within a two-minute time period. If we had only included segments in which teams discussed every variable, we would have shown very little design activity on the timelines. We selected this method in order to better represent the teams’ activities.

RESULTS

The method explained in the previous section was applied to the data from the ten teams that participated in the factory redesign study. This yielded ten sets of subproblems and ten timelines. We have included the results for two teams (named “X” and “J”) in this paper, in order to illustrate the types of results that this method can produce.

Each team’s videos were coded using these 18 variables by applying the codes to two-minute time segments. Figure 2 shows an example of the resulting coding matrix for Team X, where the leftmost column indicates the two-minute time segments. Although not shown in this figure, in the original data we also included a column titled “excerpts” that captured the key discussions between the team members that affect the variable(s) being coded.

If the team split into sub-teams, then the second team’s discussions were captured using a "y". If teams were not explicitly deciding on a variable but mentioned it - for example, planning to work on the variable at a future time - these discussions were captured using the code “p”. This yielded 120 time segments for each team (four hours of video and 30 segments per hour). Thus, across all 10 teams, we collected 1200 time segments to analyze.

We considered an association rule significant if it had a minimum support of 1% and a confidence of 50%. A support of 1% means that each of three variables would need to be coded at least 0.01 times out of the total number of time segments of a given team. A confidence of 50% means that to consider the association rule $a \Rightarrow b$, both the variables have to be coded at least 50% of the total number of instances of which a was coded. A support level as low as 1% was chosen because the number of coded time segments in our data set was relatively

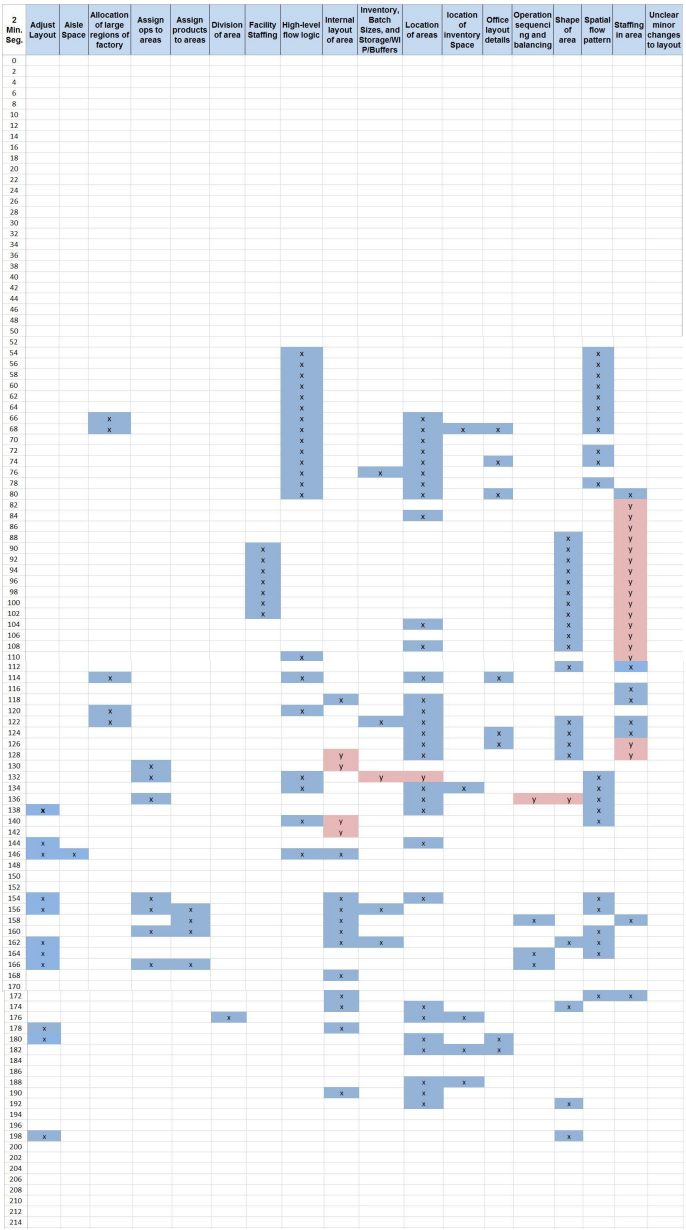

FIGURE 2. SUBSET OF THE CODES GENERATED FOR TEAM X.

small, and a higher support level would have overlooked some important subproblems.

Table 1 and Table 2 show that teams X and J discussed different subproblems. It is clear that the subproblem "High Level Logic" is the same for both teams, but all the other subproblems have different variables being discussed in each of the teams. Further, "inventory" appears as a separate subproblem in Team X, but it does not appear in any form (either as a variable in one of the other subproblems or as an independent

TABLE 1. SETS OF SUBPROBLEMS FOR TEAM X.

Subproblem	Variable Codes in Subproblem
High Level Logic	High level flow logic Spatial flow pattern Allocation of large regions Location of areas
Operations	Assignment of operations to areas Assignment of products to areas Internal layout Spatial flow pattern Location of areas
Staffing	Staffing in area Facility staffing Shape of area Office layout details
Inventory	Inventory/batch sizes/storage/WIP/buffers Assignment of space for inventory Spatial flow pattern Location of areas
Office layout	Office layout details High level flow logic Location of areas

TABLE 2. SETS OF SUBPROBLEMS FOR TEAM J.

Subproblem	Variable Codes in Subproblem
High Level Logic	High level flow logic Spatial flow pattern Allocation of large regions Shape of area Location of areas
Operations	Assignment of operations to areas Assignment of products to areas Spatial flow pattern High level flow logic Location of areas
Operation Sequencing	Operations sequencing and balancing Staffing in area
Staffing	Facility staffing Shape of area Office layout details
Office Layout	Office layout details Location of areas
Internal Layout	Internal layout of area Spatial flow pattern

subproblem) in Team J's discussion. These differences in subproblems for each team show the importance of analyzing decomposition based on team-specific subproblems, since each team considers different groups of variables during the solution process.The original timelines for teams X and J are shown in Figure 3.

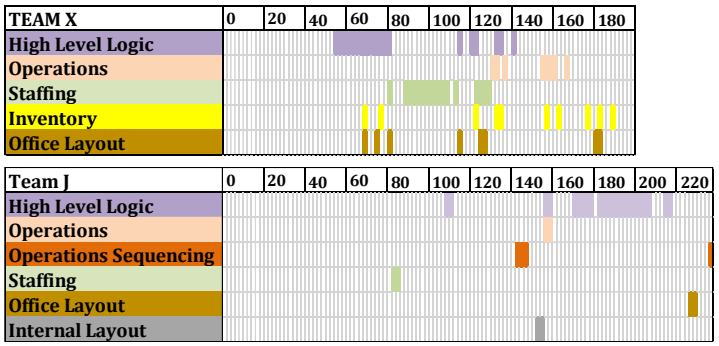

FIGURE 3. ORIGINAL TIMELINES OF TEAMS X AND J.

One of the issues with the original timelines was that specific variables such as "staffing in area" or "location for inventory" were captured only when they were discussed along with other variables in the subproblem. However, there were instances when these variables were discussed independently when solving the design problem. In order to capture this, we incorporated specific variables relevant to "staffing", "inventory", "internal layout", "operations" and "high level logic" into the relevant subproblems if they were discussed independently for more than 2 minutes. Furthermore, because the variable Location of Areas was discussed frequently by both teams we capture "location of areas" as a single variable in the timelines. The timelines were updated to incorporate these considerations as seen in Figure 4.

The updated timelines show that the two teams have very different decomposition patterns. Team J spent a large amount of time discussing the staffing and high level logic subproblems but discussed the other subproblems much less. Although Team X also spent a significant amount of time on the staffing and high level logic subproblems, they also spent time discussing the assignment of operations (which also covers the variable internal layout) and inventory-related subproblems. In general, these decomposition patterns suggest that Team J tended to work more sequentially on the different subproblems while Team X tended to work on subproblems in parallel.

Because this paper focuses on the method of developing team-specific subproblems based on variables that the team discussed and identifying the decomposition patterns across different design teams, we have only shown two teams' results in this paper to illustrate the type of findings that result from our method. Additional results from this research will be published in a later version of this paper.

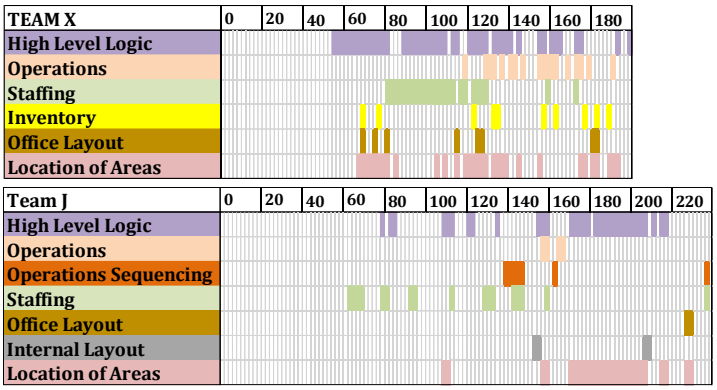

FIGURE 4. COMPARISON OF TEAMS X AND J WITH DIFFERENT TIMELINE CHARACTERISTICS.

DISCUSSION AND CONCLUSIONS

This paper presented a method to code team discussions based on the variables being discussed by the design team. We identified the subproblems, collections of variables that are frequently discussed together, in order to observe the distinct decomposition patterns of the design teams.

Wallace and Hales [31] stressed that "a hybrid of quantitative and qualitative approaches need to be developed for the analysis of empirical field data" - but such a hybrid is not commonly seen in research on decomposition of complex problems by design teams. The method that we developed is an early step in this direction and is consistent with methods used thus far by researchers in the design community, such as observing videos of design teams and development of a coding scheme to code the observed data. However, the proposed method also introduces quantitative methods, such as association rule learning, to develop the subproblems from the coded data. This method is an early step in this direction, and would benefit from further refinement, as discussed below.

Our method identifies team-specific subproblems which are important in understanding the different decomposition patterns, since as we have shown, subproblems may differ across teams even when solving the same design problem. This is not surprising, since design problems are ill-structured and can have multiple solutions [23]. We then constructed timelines that allow one to visualize and compare the decomposition patterns across multiple teams. This comparison is a useful strategy for making sense of the messy observational data because it showcases "precedence, parallel processes, and the passage of time" [8].

Although developed within the context of a study of facility design problems, the method can be applied to other design settings in which observations of designers are collected. A different design setting would certainly have different variables, but the qualitative and quantitative techniques in our method can be used to identify and visualize the relevant variables and subproblems. Once subproblem timelines are created for each team, one can further analyze the decomposition patterns across teams and link them to other metrics such as the quality of the design solutions. This paper provides researchers with a preliminary quantitative and qualitative method for using observational data to identify problem decomposition patterns of human designers.

This study reaffirms that ill-structured problems do not have a uniform decomposition pattern. However, some patterns can still be identified. As observed in teams X and J, some teams will tend to have a more sequential subproblem solving strategy compared to others that try to solve multiple subproblems in parallel. Using the method described in this paper, we hope that researchers will be able to build upon these initial insights to further explore design team decomposition strategies. In practice, a better understanding of the influence of decomposition strategies can help design organizations better advise design teams and suggest the most effective design strategy.

This research is part of a larger effort that is seeking to understand how humans decompose complex system design problems, discover the relationships between decomposition and solution quality, and develop useful models for evaluating the performance of decompositions [1, 32–34].

Application

The method described in this paper is an early step towards better ways to understand design processes based on observations of designers, and it would benefit from further refinement. We wish to highlight some important points about applying the method presented here to other design domains. The variables and subproblems in each domain will be quite different from those mentioned in the examples of the paper. Judgment is required in deciding whether and how to combine the association rules into subproblems when they overlap significantly. Further experimentation with the method should provide more insight on this issue. The thresholds for the confidence and support of the association rules need to be adjusted to find the most useful rules (thresholds that are too low will yield "too many" subproblems, but thresholds that are too high will yield "too few" subproblems). Finally, although our ongoing work is concerned with the correlation between decomposition strategy and solution quality, there are many other demographic factors and team characteristics that can influence team performance.

ACKNOWLEDGMENT

The authors acknowledge the assistance of David Rizzardo, who organized and led the facility design course and

evaluated the teams' facility layouts. The authors are supported by National Science Foundation Grant CMMI-1435074 and CMMI-1435449.

REFERENCES

[1] Tobias, C., Herrmann, J. W., and Gralla, E. L., 2015. "Exploring problem decomposition in design team discussions". International Conference on Engineering Design.

[2] Dinar, M., Shah, J. J., Cagan, J., Leifer, L., Linsey, J., Smith, S. M., and Hernandez, N. V., 2015. "Empirical studies of designer thinking: Past, present, and future". *Journal of Mechanical Design,* ***137***(2), Feb., pp. 021101–1 – 021101–13.

[3] Laird, J. E., Newell, A., and Rosenbloom, P. S., 1987. "SOAR: An architecture for general intelligence". *Artificial Intelligence,* ***33***(1), Sept., pp. 1–64.

[4] Stempfle, J., and Badke-Schaub, P., 2002. "Thinking in design teams - an analysis of team communication". *Design Studies,* ***23***(5), Sept., pp. 473–496.

[5] Atman, C. J., and Bursic, K. M., 1998. "Verbal Protocol Analysis as a Method to Document Engineering Student Design Processes". *Journal of Engineering Education,* ***87***(2), Apr., pp. 121–132.

[6] Ho, C.-H., 2001. "Some phenomena of problem decomposition strategy for design thinking: differences between novices and experts". *Design Studies,* ***22***(1), pp. 27–45.

[7] Liikkanen, L. A., and Perttula, M., 2009. "Exploring problem decomposition in conceptual design among novice designers". *Design Studies,* ***30***(1), Jan., pp. 38–59.

[8] Langley, A., 1999. "Strategies for Theorizing from Process Data". *The Academy of Management Review,* ***24***(4), Oct., p. 691.

[9] Cross, N., Christiaans, H., and Dorst, K., eds., 1996. *Analysing design activity.* Wiley, Chichester ; New York.

[10] Gero, J. S., and Mc Neill, T., 1998. "An approach to the analysis of design protocols". *Design Studies,* ***19***(1), Jan., pp. 21–61.

[11] Ahmed, S., Wallace, K. M., and Blessing, L. T., 2003. "Understanding the differences between how novice and experienced designers approach design tasks". *Research in engineering design,* ***14***(1), pp. 1–11.

[12] Kavakli, M., and Gero, J., 2002. "The structure of concurrent cognitive actions: a case study on novice and expert designers". *Design Studies,* ***23***(1), pp. 25–40.

[13] Atman, C. J., Adams, Robin S.and Cardella, M. E., Turns4, J., Mosborg, S., and Saleem, J., 2007. "Engineering design processes: A comparison of students and expert practitioners". *Journal of Engineering Education,* ***96***(4), Oct., p. 359–379.

[14] Goldschmidt, G., and Rodgers, P., 2013. "The design thinking approaches of three different groups of designers based on self-reports". *Design Studies,* ***34***(4), pp. 454–471.

[15] Dorst, K., and Cross, N., 2001. "Creativity in the design process: co-evolution of problem–solution". *Design Studies,* ***22***(5), pp. 425–437.

[16] Carmen, Z., and Frank, M., 2005. "A qualitative empirical evaluation of design decisions". HSSE '05 Proceedings of the 2005 Workshop on Human and Social Factors of Software Engineering, pp. 1–7.

[17] Mc Neill, T., Gero, J. S., and Warren, J., 1998. "Understanding conceptual electronic design using protocol analysis". *Research in Engineering Design,* ***10***, pp. 129–140.

[18] Goldschmidt, G., and Weils, M., 1998. "Contents and structure in design reasoning". *Design Issues,* ***14***, pp. 85–100.

[19] Baldwin, C. Y., and Clark, K. B., eds., 2000. *Design Rules, Volume 1: The Power of Modularity.* MIT Press, Cambridge; Massachusetts.

[20] Ethiraj, S. K., and Levinthal, D., 2004. "Modularity and innovation in complex systems". *Management Science,* ***50***(2), Feb., pp. 159 – 173.

[21] Eppinger, S. D., and Browning, T. R., eds., 2012. *Design structure matrix methods and applications.* MIT Press, Cambridge; Massachusetts.

[22] Hatchuel, A., 2001. "Towards design theory and expandable rationality: The unfinished program of herbert simon". *Journal of Management and Governance,* ***5***(3), pp. 260–273.

[23] Braha, D., and Reich, Y., 2003. "Topological structures for modeling engineering design processes". *Research in Engineering Design,* ***14***, pp. 185–199.

[24] Tobias, C., Azhar, A., Gralla, E. L., and Herrmann, J. W., 2016. "Exploring problem decomposition in design team discussions". 5th International Engineering Systems Symposium, CESUN 2016.

[25] Emerson, R. M., Fretz, R. I., and Shaw, L. L., 2011. *Writing ethnographic fieldnotes.* University of Chicago Press.

[26] Spradley, J. P., 1980. *Participant observation.* Holt, Rinehart and Winston, New York.

[27] Hendrick, H. W., and Kleiner, B. M., eds., 2002. *Macroergonomics: theory, methods, and applications.* Human factors and ergonomics. Lawrence Erlbaum Associates, Mahwah, N.J.

[28] Strauss, A., Corbin, J., et al., 1990. *Basics of qualitative research*, Vol. 15. Newbury Park, CA: Sage.

[29] Agrawal, R., Imieliński, T., and Swami, A., 1993. "Mining association rules between sets of items in large databases". SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207–216.

[30] R Core Team, 2015. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Com-

puting, Vienna, Austria.
[31] Wallace, K., and Hales, C., 1987. "Detailed analysis of an engineering design project". In Proceedings of the International Conference on Engineering Design (ICED'87), Vol. 13, ASME: American Society of Mechanical Engineers.
[32] Gralla, E. L., and Herrmann, J. W., 2014. "Team design processes and decompositions in facility design". Industrial and Systems Engineering Research Conference, 2014.
[33] Herrmann, J. W., 2010. "Progressive Design Processes and Bounded Rational Designers". *Journal of Mechanical Design,* ***132***(8), p. 081005.
[34] Herrmann, J. W., 2015. "Predicting the Performance of a Design Team Using a Markov Chain Model". *IEEE Transactions on Engineering Management,* ***62***(4), Nov., pp. 507–516.