21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17
21-25 AUGUST 2017, THE UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER, CANADA

# USING CLUSTERING ALGORITHMS TO IDENTIFY SUBPROBLEMS IN DESIGN PROCESSES

**Morency, Michael (1); Anparasan, Azrah (2); Herrmann, Jeffrey (1); Gralla, Erica (2)**
1: University of Maryland, United States of America; 2: George Washington University, United States of America

## Abstract

Designers work in teams to design complex systems. They separate the design problem into subproblems and solve the smaller, more manageable subproblems. Because this affects the overall quality of their design, it is important to understand how teams decompose system design problems, which will ultimately enable future research on how to design better design processes. We studied teams of experts solving two different facility design problems. We developed a novel approach that combines qualitative and quantitative techniques. It records a team's discussion, identifies the design variables using qualitative coding techniques, and groups these variables into subproblems. A subproblem is a set of variables that are considered together. We evaluated four clustering algorithms that group the coded variables into subproblems. This paper discusses the data collection, the clustering algorithms, and the evaluation techniques. The the algorithms generated similar but not identical clusters, and no algorithm's clusters consistently out-performed the others on quantitative measures of cluster quality. The clusters do provide insights into the subproblems that the design team solved.

**Keywords**: Decision making, Design process, Human behaviour in design, Research methodologies and methods, Teamwork

**Contact**:
Prof. Jeffrey Herrmann
University of Maryland
Mechanical Engineering
United States of America
jwh2@umd.edu

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 2: Design Processes | Design Organisation and Management, Vancouver, Canada, 21.-25.08.2017.

## 1 INTRODUCTION

Engineers work in teams to design complex systems. They separate the system design problem into a set of subproblems and solve the smaller, more manageable subproblems. Solving the subproblem requires determining the values for some of the design variables that define the system. Because this decomposition strategy affects the overall quality of the system design, it is important to understand how teams of engineers solve system design problems and how their strategies affect the quality of the systems that they design. Ultimately, knowing which decompositions lead to better solutions will enable future research on how to design better design processes.

In this research, we are studying design teams in order to identify the subproblems that they consider. We consider a "subproblem" to be a set of design variables that are considered together. To identify the subproblems used by a design team, we developed a novel approach that combines qualitative and quantitative techniques. This approach, described in Section 3, records a design team's discussion, identifies the variables that the team discussed using qualitative coding techniques, and then groups these variables into clusters that suggest the subproblems that team considered. We investigated and evaluated different clustering algorithms to group the coded variables. This is an important first step toward a method that will enable identification of subproblems based on observation of design activities.

This paper presents the results of our investigation: it describes the data we collected, explains the four clustering algorithms that we tested, displays the results for each of four teams, and compares the performance of the algorithms to one another and to the teams' discussions.

The study described in this paper involved design teams solving facility design problems, which are an important class of system design problems. We observed professionals solving two distinct instances of facility design problems during relevant training courses: (1) teams of public health professionals designing a mass antibiotic distribution facility and (2) teams of professional engineers redesigning a manufacturing facility. Results about problem decomposition in realistic facility design settings will enrich the knowledge base of design theory for facility design in particular and system design more generally.

The remainder of this paper proceeds as follows: Section 2 discusses previous work on problem decomposition. Section 3 discusses the research approach and the clustering algorithms that we used. Section 4 presents the results of the clustering algorithms. Section 5 discusses these results. Section 6 summarizes and concludes the paper.

## 2 RELATED LITERATURE

Problem decomposition by designers has been studied in recent years, with an emphasis on understanding how problem decomposition distinguishes expert from novice design approaches (Ho, 2001; Liikkanen and Perttula, 2009; Sun et al., 2016). In the literature, several related phenomena have been investigated: the distinction between 'explicit' and 'implicit' decomposition (Ho, 2001; Liikkanen and Perttula, 2009; Sun et al., 2016). Researchers have also investigated the search type, i.e., whether designers use breadth-first or depth-first search (Ho, 2001; Sun et al., 2016). Finally, researchers have also investigated distinctions in the way goals are utilized in search: working-forward (novice) and working-backward (expert) strategies (Ho, 2001). In this literature, then, decomposition is related to the identification of sub-goals and a designer's search through the problem space.

The important question of which design variables are in which subproblems has not been well studied, however. Other views of decomposition (e.g. from systems engineering and management) emphasize the definition of subproblems made up of strongly linked components, separated from other subproblems to which they are weakly linked (Simon, 1962; Newell, 1972). This definition suggests the question: when humans design, which variables are treated together as subproblems? Answering this question will fill a gap in the literature, by identifying types of decompositions created by human designers.

Our study takes a step toward filling the gap by examining multiple clustering algorithms and evaluating their ability to identify groups of variables that make up subproblems in a design process.

## 3 APPROACH AND METHODS

To identify the subproblems considered by design teams, we studied the activities of several design teams engaged in half-day design sessions. We recorded each team's activities, used inductive qualitative coding to identify the design variables teams considered, and then used clustering algorithms to group the design variables into clusters. This paper focuses on the clustering algorithms but also summarizes the data collection and coding aspects for completeness.

### 3.1 Data collection

We studied the activities of four design teams: two teams redesigned a factory, and two teams designed a point of dispensing (POD) for antibiotics. These teams were selected from a larger set of teams that participated in these design activities. They were selected because they were different from one another and spanned the range of variation of the other teams, so they were useful and representative test cases for the clustering algorithms.

In the factory redesign study, we observed teams consisting of professional industrial engineers and manufacturing managers redesigning a factory as part of several two-day lean facility design courses. The exercise took approximately four hours. Each team was given a scenario that specified information about the existing factory and products and criteria for evaluating the redesigned factory layout, including productivity and material handling effort. For more details, see Azhar et al. (2016).

In the POD design study, we observed teams of public health and emergency response professionals designing a POD, a temporary facility for rapidly dispensing prophylactic antibiotics in response to an anthrax attack (Abbey et al., 2013). In this exercise, the participants created the layout and staffing plan for a POD in a new high school. They were provided with the blueprints of the school and basic data about the population the POD must serve. The exercise took approximately two hours.

Field observation with videorecording was the primary data collection method (Emerson et al., 2011; Spradley, 1980). For each team, a videocamera was pointed down at the layouts from overhead so that the video captured what participants drew on the layouts, when they moved areas around, etc. We also photographed the final layouts of each team and collected all relevant documents.

Because design researchers commonly use time-limited design exercises (Dinar et al., 2015), we believe that our approach 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 (Hendrick and Kleiner, 2002).

### 3.2 Developing and coding variables

We inductively developed a set of codes to represent the variables that the teams considered so that the codes represent what the teams discussed, not the researchers' ideas. We observed the video of each team's discussion and coded it by noting the variables that the team discussed in each two-minute time segment. 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. Throughout the process, we maintained a *codebook* that defined each variable and recorded examples. Our method followed standard qualitative data analysis approaches (Langley, 1999; Strauss and Corbin, 1990).

After determining the variables, we re-coded the videos according to the final codebook. For each two-minute segment, we determined which variables the team discussed. Let $x_{it} = 1$ if variable *i* is coded (discussed) in time segment *t*, for variables $i = 1, \ldots, N$ and time segments $t = 1, \ldots, T$. The data analyzed did not include variables that were not coded at all. Across these four teams, the number of variables ranged from 29 to 50; and the number of time segments analyzed ranged from 30 to 128.

### 3.3 Clustering variables

We sought to identify the subproblems that each team discussed. In this work, a "subproblem" is a set of variables that are determined concurrently by a team. The variables in a subproblem, therefore, are discussed at approximately the same time. Because the codes identified when each team discussed different variables, we used this data to determine (approximately) the subproblems. Identifying the subproblems exactly is difficult because the teams did not explicitly describe the subproblems that they considered or when they began or stopped working on a subproblem, and they have worked on and discussed multiple subproblems simultaneously.

Ideally, it would be possible to partition the time segments and variables into independent clusters so that the variables discussed in one set of time segments were not discussed at all in other time segments. Each set of variables would form a distinct cluster that corresponds to a specific subproblem. Although such independent clusters did occur in some situations, it was also common that a variable was discussed in one time segment with one set of variables and in another time segment with a different set of variables. Therefore, we needed a technique to identify clusters of variables that were often discussed together. Because we did not presume that the subproblems would be the same for different teams, this analysis was done for each and every team. (Our analysis does presume, however, that the subproblems are consistent across time for each team.)

We investigated four computational approaches that we call Ward's clustering, spectral clustering, Markov clustering, and association rule clustering. All of the approaches begin with the coded time segments, and each one yields a set of clusters (each cluster has one or more variables). The Markov clustering algorithm groups every variable into a cluster; the other algorithms may, in some cases, leave some variables in no cluster.

**Ward's clustering** groups variables based on a "distance measure" that is specified in a dissimilarity matrix. The algorithm starts by pairing variables that have a minimum distance between them (least dissimilarity). In subsequent iterations, the clustered variables are paired with other clusters or variables to form larger clusters. To identify one specific set of clusters, a distance threshold is chosen (this was set at 6 in our analysis, to obtain a reasonable number of clusters). Although there are multiple options for specifying the dissimilarity between data points, we used the Euclidean distance. The Euclidean distance between variables *i* and *j* can be calculated as follows:

$$d(i,j) = \sqrt{\sum_{t=1}^{T} \left( x_{it} - x_{jt} \right)^2} \quad (1)$$

We used Ward's method (implemented in the 'hclust ward.D2' function in R) to cluster the variables.

**Spectral clustering** (Sarkar et al., 2009, 2014) used a measure called the relative count $a_{ij}$, which equals the number of time segments in which a team discussed both variables *i* and *j* divided by the number of time segments in which they discussed either one or both *i* and *j*. The method involves finding the *k* largest eigenvalues of a matrix with the elements $a_{ij}$. We used hierarchical clustering to create a dendrogram of the variables using the distances between points in the *k*-dimension space (note that this is not the same as the distance used in Ward's clustering). We set a distance threshold to create clusters that included most pairs of variables with a large relative count.

**Markov clustering** (van Dongen, 2008) uses as input a concurrency matrix *C* that has one row and one column for each variable, and element $c_{ij}$ is the number of times both variables *i* and *j* were coded divided by the number of times in which variable *i* was coded (therefore it is not symmetric, unlike the relative count). The Markov clustering algorithm multiplies the matrix *C* by itself to create relationships between elements connected by other elements and then raises each element to a power (the inflation parameter) before renormalizing so that the matrix is a transition probability matrix. We used the MCL application (van Dongen, 2012) to construct the clusters (the inflation parameter was set to create a reasonable number of clusters at 0.6 for the POD teams and 8 for the factory teams).

**Association rule clustering** is based on association rule learning, which identifies sets of items that typically occur together (Agrawal et al., 1993). The technique is typically used on very large datasets. The algorithm finds sets of items with high support (number of times a variable is coded), confidence (proportion of times in which if one variable was coded, a second was also coded), and lift (proportion of observed support of two variables together to that expected if they were independent). For each team, we generated association rules based on the coded data using 'arules' in R (R Core Team, 2015) (using a support of 0.033 and confidence of 0.5 for the POD teams and 0.02 and 0.5 for the factory teams). Each association rule established a relationship between two or more variables, and we developed clusters that include all variables that were related to one another by any association rule.

*Table 1. Clustering results. The number of clusters generated by a clustering algorithm (the number of variables not included in any cluster).*

| Team | Spectral | Markov | Association Rules | Ward's |
|---|---|---|---|---|
| Factory Team 4 | 4 (5) | 7 (0) | 2 (14) | 6 (5) |
| Factory Team X | 7 (1) | 6 (0) | 1 (3) | 8 (1) |
| POD Team 3 | 7 (17) | 7 (0) | 6 (21) | 9 (2) |
| POD Team 4 | 6 (5) | 7 (0) | 7 (28) | 11 (7) |

## 4 RESULTS

We applied the four clustering algorithms to the coded data for four teams, which were labelled POD Teams 3 and 4 and Factory Teams X and 4. Table 1 describes the sets of clusters.

We evaluated the clustering algorithms by determining how well they created clusters that (1) differentiated groups of variables discussed concurrently from those discussed separately and (2) were aligned with the team's discussions (captured in the video-recordings). To assess (1), we used several numerical measures of clustering success and qualitatively compared the clusters that the methods created. To assess (2), we examined several segments of video and evaluated the consistency between the clustering results and the team's discussions.

### 4.1 Cluster evaluation measures

Although we have no external quantitative benchmark against which we can evaluate the clusters that the clustering algorithms generated for each team, we did evaluate the sets of clusters against each other using several measures. We used two versions of the Dunn index $\alpha$ (Dunn, 1974). This measure is the ratio of the minimal "distance" between clusters to the largest cluster "diameter." In Dunn (1974), the diameter equals the maximal distance between two items in the same cluster, and the distance between two clusters is the minimal distance between an item in one cluster and an item in the other cluster. We also used a modified version based on the cluster centroids; in this version the diameter equals the average distance between items and the centroid, and the distance between two clusters is the distance between their centroids.

Rousseeuw (1987) defined the silhouette measure, which describes how well each item lies within its cluster. An item has a positive silhouette value if the average distance from that item to items in the closest cluster is greater than the average distance from that item to the other items in the same cluster.

We also determined the number of high relative count pairs in the same cluster and the number of high concurrency pairs in the same cluster. A high relative count pair is a pair of variables *i* and *j* such that $a_{ij} \geq 0.9$. A high concurrency pair is a pair of variables *i* and *j* such that $c_{ij} \geq 0.9$. A higher value indicates that, for more of these pairs, both variables were grouped into the same cluster.

### 4.2 Cluster evaluation results

We applied these cluster evaluation measures to the sets of clusters generated for the four teams in this study: Factory Team 4, Factory Team X, POD Team 3, and POD Team 4. The results for the relative count measure and concurrency measure are shown in Table 2. The Markov clusters had generally larger values for these measures for most sets (except for Factory Team X). For the relative count measure, three or four sets of clusters yielded similar values.

The results for the silhouette values are also shown in Table 2. The Markov clusters and Ward's clusters had generally more variables with positive silhouette values, especially for POD Teams 3 and 4. For Factory Team X, Ward's clusters had generally more variables with positive silhouette values.

The results for the modified Dunn index and the original Dunn index are shown in Table 3. Clusters that are more separated from each other will have larger values of the Dunn index. The Ward's clusters often had better values, but it did not dominate the other sets of clusters for all cases.

*Table 2. Cluster evaluation measures: high relative count pairs in the same cluster; high concurrency pairs in the same cluster; positive silhouette values.*

| Team | Spectral | Markov | Association Rules | Ward's |
|---|---|---|---|---|
| Factory Team 4 | 1; 6; 10 | 1; 9; 9 | 1; 6; 16 | 1; 7; 23 |
| Factory Team X | 0; 3; 8 | 0; 7; 6 | 0; 16; 0 | 0; 8; 19 |
| POD Team 3 | 19; 40; 19 | 19; 73; 31 | 6; 21; 15 | 19; 63; 35 |
| POD Team 4 | 11; 49; 14 | 11; 90; 39 | 1; 12; 16 | 11; 57; 35 |

*Table 3. Cluster analysis results: the original and modified Dunn index.*

| Measure | Team | Spectral | Markov | Association Rules | Ward's |
|---|---|---|---|---|---|
| Original Dunn | Factory Team 4 | 0.267 | 0.272 | 0.423 | 0.577 |
| | Factory Team X | 0.167 | 0.174 | 0.000 | 0.333 |
| | POD Team 3 | 0.500 | 0.392 | 0.522 | 0.707 |
| | POD Team 4 | 0.277 | 0.302 | 0.302 | 0.500 |
| Modified Dunn | Factory Team 4 | 0.616 | 0.565 | 1.341 | 0.881 |
| | Factory Team X | 0.498 | 0.443 | 0.000 | 0.547 |
| | POD Team 3 | 0.955 | 0.853 | 0.890 | 1.262 |
| | POD Team 4 | 0.560 | 0.904 | 1.031 | 1.253 |

## 4.3 Timeline comparison

For each set of clusters, we created a timeline that sets the variables in the same cluster next to each other and indicates the time segments in which each variable was coded (each row is a variable, and each column is a time segment). Because many of the first and last time segments contained no coded variables, these were omitted from the timelines.

For convenience, we color-coded the variables so that the variables in the same cluster constructed by the Markov clustering algorithm have the same color. In the timeline for another set of clusters, the variable retains that color, which makes it easier to compare the clusters.

Due to space constraints, this paper includes only the timelines for the clusters for POD Team 3 (Figures 1 to 4). For these clusters, it appears from the timelines that the algorithms were able to effectively group together variables that were discussed with or near each other. This team discussed the variable Calculating Staff Needs repeatedly throughout their design process. Variables that were discussed with this variable were clustered together in the Markov clustering results. For this team, certain small sets of variables were clustered together, sometimes as a distinct cluster, and sometimes in a larger cluster with other variables, so there are some similarities between the clusters.

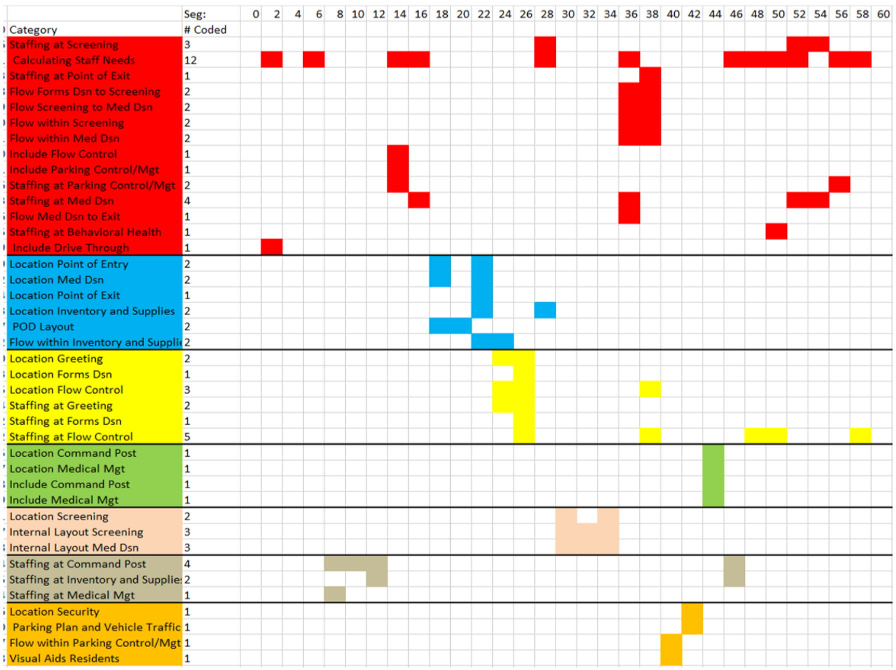


*Figure 1. Timeline for POD Team 3 Markov clusters*

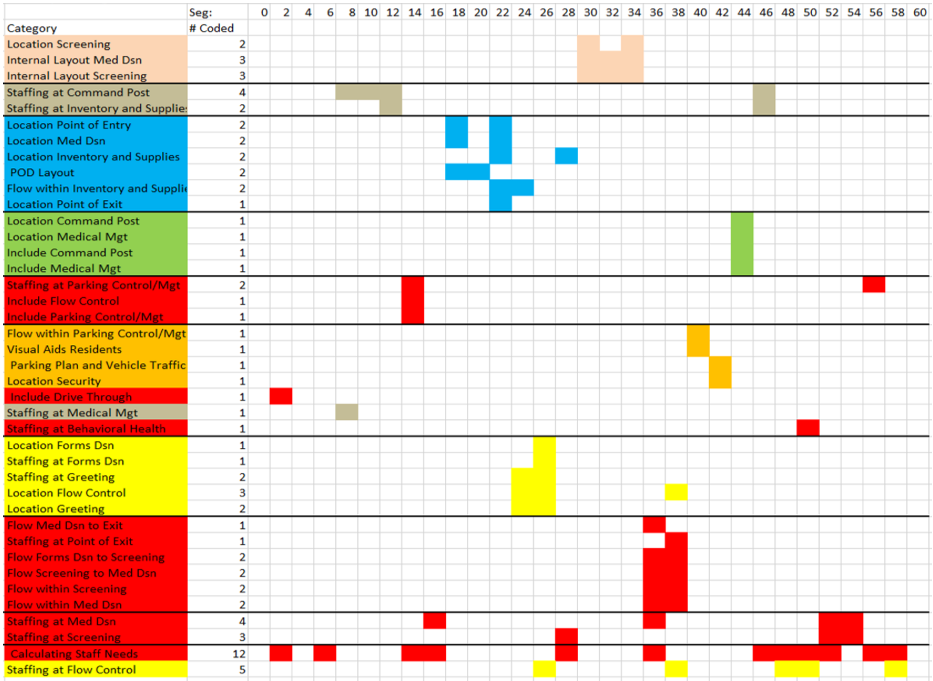


*Figure 2. Timeline for POD Team 3 Ward's clusters*

### 4.4 Comparison with videos

We selected four sections of the videos from POD Team 3 and POD Team 4 in order to compare the algorithms' results to the specific elements that the teams discussed. The sections were chosen because they were periods when the timelines and clusters indicated that the team discussed multiple subproblems, but the clustering results did not completely agree with each other. Four researchers reviewed these sections and noted the elements that the team discussed; these results were then compared and synthesized to generate a consensus description of that section. For space reasons, only the results from POD Team 3 are reported in this paper.

#### *4.4.1 POD Team 3 Segments 22 to 28*

In this section, the team discussed the general flow throughout the POD, focusing on the entry, exit, and key medicine distribution points. The team used the flow of residents as a transition into a discussion about the location and flow through the greeting and forms distribution stations. The team then shifted to discussing the design of the screening and medical distribution stations but also touched on previously discussed topics like the Location of Inventory and Supplies.

The clustering algorithms' results are consistent with this description. See, in particular, the second and third clusters (with the blue and yellow variables) in the Markov clustering timeline in Figure 1. This clustering algorithm did the best job at clustering the variables that the team discussed during this section.

#### *4.4.2 POD Team 3 Segments 40 - 48*

The discussion started with the parking lot flow and flow of people from the exit into the parking lot. At this point, the team, who had determined most of their POD design, discussed other stations such as the Command Post and Medical Management. The team then considered the staffing at these stations and other stations. Finally, they shifted towards the overall staffing needs.

As shown in the timeline in Figure 1, the results of the Markov clustering approach includes clusters that are consistent with the first parts of this discussion, but none of the algorithms created clusters that were consistent with the final staffing discussion.

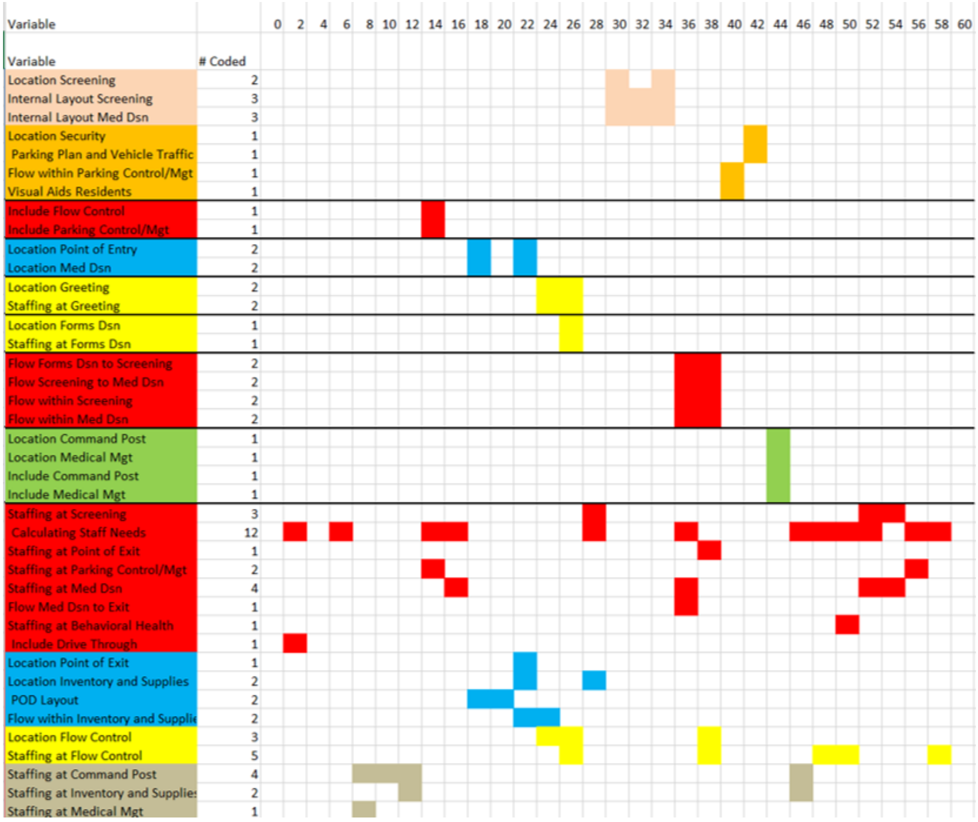


*Figure 3. Timeline for POD Team 3 spectral clusters*

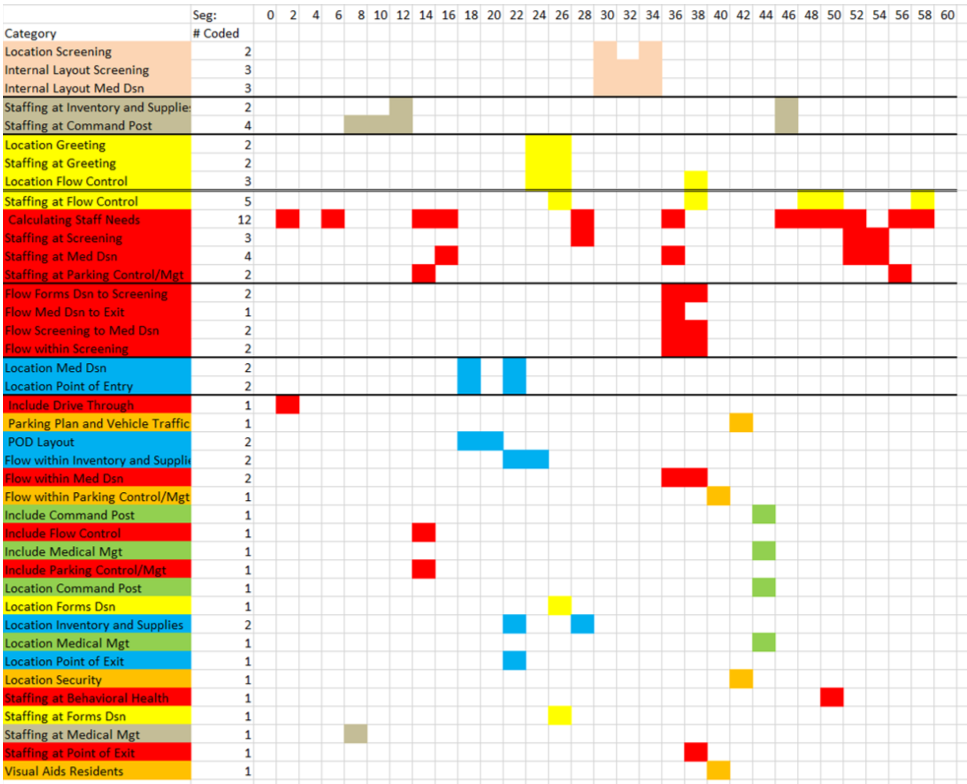


*Figure 4. Timeline for POD Team 3 association rules clusters*

Overall, considering these results and those (unreported) from the POD Team 4 videos, the algorithms performed well at creating clusters that were consistent with the teams' discussions, although they sometimes missed certain transitions, included unrelated variables, or failed to include relevant variables in the clusters.

## 5 DISCUSSION

In some ways, the clustering algorithms generated similar results. They consistently included in the same cluster variables that were coded in the same time segments but no others. For example, in POD Team 3's discussions, the variables Location of Screening, Internal Layout of Screening, and Internal Layout of Medicine Distribution (the fifth cluster in Figure 1) were coded at the same time and in no other segments. All four clustering algorithms grouped these three variables in the same cluster.

The clustering algorithms generated different results for a variable that was coded in many time segments and with different sets of variables. For example, in POD Team 3's discussions, the variable Calculating Staff Needs was coded in 12 segments (Figure 1). The Markov clustering algorithm and the association rules clustering algorithm included both this variable and the Staffing at Screening variable in the same cluster, but Ward's clustering algorithm and the spectral clustering algorithm did not include these two in the same cluster. For any variable that shared only some of those coded segments, the concurrency value with Calculating Staff Needs was large. For example, the Staffing at Screening variable was coded in three time segments, two of which included Calculating Staff Needs as well. Because the concurrency value is 2/3, the Markov clustering algorithm yielded a cluster that included both variables. Likewise, an association rule included both variables, so they were included in the same cluster by the association rules clustering algorithm. Because there were eleven time segments in which only one of the variables was coded, which increased the distance between them, Ward's clustering algorithm did not include these two in the same cluster. Indeed, the Staffing at Screening variable was not included in any cluster by this algorithm. The relative count between these two variables is 2/13, so the spectral clustering algorithm did not include these two in the same cluster. In this way, the large first cluster that the Markov clustering algorithm generated for POD Team 3's discussion (shown with red variables in Figure 1) was broken apart, and those variables were grouped in different ways by the other clustering algorithms.

Although similar results were seen for the clusters for POD Team 4's discussion, the results for Factory Team 4 and Factory Team X showed little similarities across the clustering algorithms. First, the variables coded for the factory teams were coded in more segments. The average number of coded segments per variable was 2.1 for POD Team 3 and 2.7 for POD Team 4. The average number of coded segments per variable was 6.2 for Factory Team 4 and 7.0 for Factory Team X. In part, this occurred because some of the factory variables were high-level and some applied to many manufacturing functions (unlike the station-specific variables that were coded for the POD design teams).

It also occurred because the teams would return to certain location variables repeatedly as they discussed the new design. In the POD design problem, the high school had plenty of space for the POD; in the factory redesign problem, however, the factory was just large enough for the functions that were being rearranged, so space conflicts were more common, and moving or changing one manufacturing function often required moving or changing another one to make space. Thus, a team often had to reconsider a variable that they had already determined.

Overall, with some caveats, these results indicate that using clustering algorithms to group the variables can be effective for indicating the subproblems that a design team considered. Different algorithms may generate different sets of clusters, but variables that are often discussed at the same time will be put into the same cluster. The similarities between different sets of clusters may also provide insight into the most tightly-coupled subproblems that the team considered.

When evaluated using quantitative measures, no algorithm's clusters consistently outperformed the others. Moreover, based on our comparison between the clusters and some of the teams' discussions, the clusters generated by the algorithms do not perfectly match the teams' discussions. Thus, we cannot identify the situations in which any clustering algorithm would generate better clusters. Still, the clustering algorithms do identify key subproblems, and it appears that using multiple clustering algorithms will provide the most well-rounded description of a team's design process.

## 6 SUMMARY AND CONCLUSIONS

We have developed a qualitative and quantitative approach for analyzing a design team's discussions.

This approach provides a way to describe a team's design process. This paper describes the results of a study to evaluate four different clustering algorithms. The results indicate that the algorithms generate similar but not identical clusters, and no algorithm's clusters consistently outperformed the others on quantitative measures of cluster quality. The clusters that are generated provide insights into the

subproblems that the design team solved. These insights help us understand and compare design processes.
We plan to use these clustering algorithms on the discussions of additional teams that tackled the POD design and factory redesign problems, and we will use the results to compare the teams' design processes. We also plan to look for correlations between the subproblems and the quality of the solutions that the teams constructed.
Based on the results of this study, it may be useful to adapt our approach in two ways: (1) defining variables that are as specific as possible based on the teams' discussions and (2) adapting the clustering algorithms to identify and separate variables that are discussed repeatedly, with different sets of variables, throughout a design process (cf. Wilschut et al., 2016).

## REFERENCES


Abbey, R. L., K. A. Aaby, J. W. Herrmann (2013), "Planning and managing mass prophylaxis clinic operations", In B. T. Denton, ed., *Handbook of Healthcare Operations Management*, Springer, New York.
Agrawal, R., T. Imielinski, A. Swami (1993), "Mining association rules between sets of items in large databases", 207-216. *SIGMOD '93 Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*.
Azhar, A., E. L. Gralla, C. Tobias, J. W. Herrmann (2016), "Identification of subproblems in complex design problems: a study of facility design", *Proceedings of the ASME 2016 International Design Engineering Technical Conferences Computers and Information in Engineering Conference*.
Dinar, M., J. Shah, J. Cagan, L. Leifer, J. Linsey, S. Smith, N. V. Hernandez (2015), "Empirical studies of designer thinking: past, present, and future", *Journal of Mechanical Design*, Vol. 137, No. 2, 021101.
Dunn, J. C. (1974), "Well-separated clusters and optimal fuzzy partitions", *Journal of Cybernetics*, 4(1), 95-104.
Emerson, R., R. Fretz, L. Shaw (2011), *Writing Ethnographic Fieldnotes*, University of Chicago Press, Chicago.
Hendrick, H., B. Kleiner (2002), *Macroergonomics: Theory, Methods, and Applications*, Lawrence Erlbaum Associates, Mahwah, New Jersey.
Ho, C.-H. (2001), "Some phenomena of problem decomposition strategy for design thinking: differences between novices and experts", *Design Studies*, Vol. 22, No. 1, pp. 27-45.
Langley, A. (1999), "Strategies for theorizing from process data", *The Academy of Management Review*, Vol. 24, No. 4, pp. 691-710.
Liikkanen, L. A., M. Perttula (2009), "Exploring problem decomposition in conceptual design among novice designers", *Design Studies*, Vol. 30, No. 1, pp. 38-59.
Newell, A., Simon, H. A. (1972), *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, New Jersey.
R Core Team. (2015), R*: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
Rousseeuw, P.J. (1987), "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", *Computational and Applied Mathematics*, Vol. 20, pp. 53-65.
Sarkar, S., A. Dong, J. S. Gero (2009), "Design optimization problem reformulation using singular value decomposition", *Journal of Mechanical Design*, Vol. 131, No. 8, 081006.
Sarkar, S., A. Dong, J. Henderson, P. Robinson (2014), "Spectral characterization of hierarchical modularity in product architectures", *Journal of Mechanical Design*, Vol. 136, No. 1, 011006.
Simon, H.A. (1962), "The Architecture of Complexity", *Proceedings of the American Philosophical Society*, Vol. 106, No. 6, pp. 468-482.
Spradley, J. (1980), *Participant Observation*, Holt, Rinehart and Winston, New York.
Strauss, A., J. Corbin (1990), *Basics of Qualitative Research*, Sage, Newbury Park, California.
Sun, G., S. Yao, J. A. Carretero (2016), "An experimental approach to understanding design problem structuring strategies", *Journal of Design Research*, Vol. 14, No. 1, pp. 94-117.
Van Dongen, S. (2008), "Graph clustering via a discrete uncoupling process", *SIAM Journal on Matrix Analysis and Applications*, Vol. 30, No. 1, pp. 121-141.
Van Dongen, S. (2012), "MCL - a cluster algorithm for graphs", http://micans.org/mcl/. Accessed: 2016-11-10.
Wilschut, T., P. Etman, J. Rooda, I. Adan (2016), "Multi-level flow-based Markov clustering for design structure matrices", *Proceedings of the ASME 2016 International Design Engineering Technical Conferences Computers and Information in Engineering Conference*.


## ACKNOWLEDGEMENTS


The authors acknowledge the assistance of David Rizzardo, who organized and led the facility design course, and Connor Tobias, who assisted with the data collection and analysis. This research was supported by National Science Foundation grants CMMI-1435074 and CMMI-1435449.