ABSTRACT

Title of thesis:	EVALUATING CLUSTERING ALGORITHMS TO IDENTIFY SUBPROBLEMS IN DESIGN PROCESSES
	Michael Morency, Master of Science in Systems Engineering, 2017
Thesis directed by:	Professor Jeffrey W. Herrmann Department of Mechanical Engineering and Institute for Systems Research

Design problems are inherently intricate and require multiple dependent decisions. Because of these characteristics, design teams generally choose to decompose the main problem into manageable subproblems. This thesis describes the results of a study designed to (a) explore clustering algorithms as a new and repeatable way to identify subproblems in recorded design team discussions, (b) assess the quality of the identified subproblems, and (c) examine any relationships between the subproblems and final design or team experience level. We observed five teams of public health professionals and four teams of undergraduate students and applied four clustering algorithms to identify the team's subproblems and achieve the aforementioned research goals. The use of clustering algorithms to identify subproblems has not been documented before, and clustering presents a repeatable and objective method for determining a team's subproblems. The results from these algorithms

as well as metrics noting the each result's quality were captured for all teams. We learned that each clustering algorithm has strengths and weaknesses depending on how the team discussed the problem, but the algorithms always accurately identify at least some of the discussed subproblems. Studying these identified subproblems reveals a team's design process and provides insight into their final design choices.

EVALUATING CLUSTERING ALGORITHMS TO IDENTIFY SUBPROBLEMS IN DESIGN PROCESSES

by

Michael Morency

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2017

Advisory Committee:
Professor Jeffrey W. Herrmann/Advisor, University of Maryland College Park
Assistant Professor Erica Gralla, George Washington University
Professor Linda Schmidt, University of Maryland College Park

Acknowledgments

I would like to thank my adviser Dr. Jeffrey Herrmann for his continuous support of my Masters study and research.

I would like to thank the rest of my thesis committee: Dr. Erica Gralla and Dr. Linda Schmidt for their insightful comments and constructive questions.

Finally, I would like to thank my fellow researchers: Azrah Azhar and Connor Tobias for helping me collect, analyze, and organize the data.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

POD	Point of Dispensing
P1	Professional Team 1
P2	Professional Team 2
P3	Professional Team 3
P4	Professional Team 4
P5	Professional Team 5
S1	Student Team 1
S2	Student Team 2
S3	Student Team 3
S4	Student Team 4
MCL	Markov Clustering Algorithm
Med Dsn	Medication Distribution
Medical Mgt	Medical Management

Chapter 1: Introduction and Literature Review

Engineers and designers often work in groups to plan out intricate systems. The engineers usually decompose the larger problem into smaller, more manageable subproblems. It stands to reason that the strategy a team uses to break down the larger overall problem, or the contents and characteristics of the subproblems, influence the overall quality of the final design. Any engineering team would benefit from knowing the optimal decomposition strategy to create the best possible design. Therefore, it is worthwhile to study how teams decompose problems and how to improve upon the design process.

In this research we studied design teams and attempted to identify their subproblems. In order to do this, we utilized both manual and algorithmic methods to analyze nine design teams' discussions. This approach, described in Section 2.2, records a design team's discussion, identifies the variables that the team discussed, and groups these variables into subproblems. In order to group the variables, we employed four different clustering algorithms. We then used a variety of quality measures and compared the algorithm results to our perception of the discussed subproblems in order to gauge the effectiveness of our methods. This thesis presents the results of our investigation

The study described in this thesis involved design teams either studying design problems or public health emergency preparedness professionals. POD design is a necessary emergency preparedness system design problem for most counties in the United States. We observed professionals and graduate students solving a POD design problem with real world properties and constraints. Knowledge about POD design, related subproblems, and problem decomposition in a realistic POD design settings will advance the techniques used in POD design in particular and system design more generally.

1.1 Human Designers

Humans and design teams have been the center of many studies that have examined the design process and quality of solutions to the team's experience level. Researchers have found that professional or expert designers use heuristics from previous experiences to create solutions [1, 2]. Studies have also shown that the decomposition strategy of the team has significant impact on the design process and the final solution [2–4]. Although some research has been done regarding how teams solve problems and value diversity [5, 6], there has been more focus on the differences between the strategies that experts and novices use [4, 7, 8].

1.2 Design Processes

Previous studies have shown that design teams create through a series of distinct decisions [9]. The decision strategy or pattern can become complicated depending on the team, as two separate decisions may be made individually or simultaneously [10]. Design processes require decisions frequently, decisions which are largely dependent on the way the team decomposes problems. When human designers work within organizations, they usually carry out the organization's design process, but they may use informal decompositions when solving design problems within the phases of that process. Studies of teamwork, the process by which members seek, exchange, and synchronize information, show its importance for team decision-making [11, 12]. However these studies failed to thoroughly research the teams' decomposition strategy and results.

1.3 Studies of Designers

Dinar [13] compiled previous attempts to understand design studies, while documenting the development of methods from initial organized studies to current, more outlined and developed methods. Dinar identified that many of the methods were creative and captured a wide variety of data from the design process. However, Dinar observed a major weakness in the lack of a formalized and repeatable method for data capture and analysis. Design studies have employed a wide variety of methods for recording the design team's discussion and have been able to apply that data to numerous research questions. The presentation of results have been just as varied, with some groups including timelines, research sequences, linkographs, or other graphical representations [13].

Dinar stated that engineers are best suited for data analysis regarding the quality of the final design, as opposed to the process that leads to said design. Dinar [13] also reviewed papers that discussed the pitfalls of fixation on a specific design, the consequences of innovation via analogy, and on the problem statements relationship with the team members' perception on design restrictions. Dinar's research on team composition and team dynamics indicated that a positive relationship existed between proper team composition and successful designs and that generating diverse ideas improved design quality.

Dinar finishes by suggesting a standardization of methods for experimental design and data collection. Dinar believed this will benefit the field of design studies by improving the usefulness of results, leading to the development and pursuit of more complicated and penetrating research questions. This field of study would also be more accessible and repeatable if the data analysis methods were less resource intensive. Larger sets of richer data from a variety of research backgrounds would provide more meaningful results.

1.4 Decomposition

Liikkanen [4] defined decomposition as the "processes producing subgoals," and Newell and Simon [14] described subgoals as "desired problem states". In the scope of a design problem, understanding a team's decomposition strategy involves both the identification of subgoals [4] and the chronological order of those identified subgoals. Studying decomposition provides an opportunity to view how design teams make decisions for simpler problems as well as the larger design. However, research in this area indicates the that theoretical perceptions of decomposition do not match how actually teams work [15].

Ho [7] identified two types of decomposition: explicit and implicit. Explicit decomposition is a top-down approach where the team forms an ideological framework for the end design before and while progressing through the problem. Implicit decomposition is characterized by a bottom-up approach where the team finds the end design one solution at a time. Despite explicit decomposition being more effective [8], designers usually rely on implicit decomposition. Novice designers seem to only use implicit decomposition [4].

One study defined implicit decomposition as any strategy that continuously strung together subgoals without previously discussing the end design [4]. In Liikkanen's research, three out of a total sixteen subjects used an explicit decomposition, and based on the results the researchers believed that explicit decomposition could be used to circumvent challenges faced by implicit decomposition by introducing new and varied view points.

A study by Tobias [16] supports this benefit by documenting this perceived importance in decision making and identifying six major types of decision-making communication: problem definition, orientation, solution development, nontask, simple agreement, and simple disagreement [17]. Teams used these communication types while making decisions regarding their problem and their ability to do so had a direct effect on the solution quality.

1.5 Subproblems

The identification, evaluation, and detailed analysis of the decomposed subgoals or subproblems has not been thoroughly researched. Views of decomposition emphasize the definition of subproblems as a combination of significantly linked topics or ideas which are distinct from other subproblems due to a lack of any significant relationships [18]. There is a gap in research regarding the topic composition of the subproblems and a dependable and repeatable method for identifying subproblems out of data.

1.6 Research Questions

There is room for new ideas and organized methods in the field of design teams, specifically when looking at subproblems and their characteristics. Subproblems directly effect the output, and the subproblems are influenced by many factors including their contents, complexity, and creators' experience level. Understanding the subproblems will lead researchers to better understand the way design teams work with problems and with team members. This research hopes to advance the understanding of subproblems and their place in the solution process by answering five key questions:

- Are there easily repeatable ways to identify subproblems from a design team's conversation?
- How can we measure the quality of our methods?

- Do teams take a different or similar approach to designs, and do subproblems indicate this?

- How do subproblems and their characteristics (size, contents, chronological order) affect the end designs of the teams? Do teams with significantly different final designs, such as Professional Team 3, have significantly different subproblems?

- Does experience level affect how design teams create different subproblems?

1.7 Overview

The remainder of this thesis proceeds as follows: Chapter 2 discusses the methods used to capture each team's discussion and format that data for use in one of four clustering algorithms. The chapter also contains the results from each of the clustering algorithms, as well as some discussion involving the algorithms' strengths and weaknesses in relation to the proposed research questions. Chapter 3 discusses the methods and results involving the teams' decomposition strategies. This chapter focuses on additional methods used to supplement the clustering algorithms while identifying the decomposition strategy. The discussion in this chapter includes how variables were clustered, how novice and professional strategies compare, and how the subproblems identified by our methods relate to the final design choices. Chapter 4 summarizes and concludes the thesis.

Chapter 2: Using Clustering Algorithms

2.1 Overview

This chapter includes the methods, results, and discussion for the clustering algorithms. Section 2.2 describes how the data was captured, formatted for each clustering algorithm, how each algorithm was executed, and then how the outputs were formatted. Section 2.3 will present the raw data in a timeline format and show how that data was transformed for the clustering algorithms' inputs. Section 2.4 will note how each algorithm clustered each team's discussion and will stay at a observational level rather than an analytical one. 2.5 will take those observations and draw more robust conclusions about the strengths, weaknesses, and other characteristics of the clustering algorithms.

2.2 Data Collection and Clustering Methods

To achieve my research objectives, two observational studies were performed on design teams of 4 to 5 people. The first study was done with graduate students, while the second study was done with professional or expert participants. The teams were tasked with designing a POD (Point of Dispensing) for a new high school and given certain constraints and requirements in the problem statement (see Appendix A for full problem statement). Researchers stood by in both studies to answer any clarifying questions the participants had throughout the experiment. The participants' discussions and designs were captured on video camera and then later analyzed by the researchers. This method of study was chosen due to its practicality and natural environment. Researchers were able to review the discussions at their leisure, and participants were able focus on the design problem rather than the experiment. The design teams' thought processes and structures were captured using a number of analysis techniques. The following sections give details regarding the design problem and the methods used to capture and analyze the discussions.

Figure 2.1 shows how each method discussed in this chapter relates to one another. The analysis flow starts with the teams, who created the POD designs and discussions captured on video. The video recordings were subjected to direct analysis (described in Sections 3.1.2) as well as a coding process (described in Section 2.2.2.1). With the coded data we created a variety of timelines, concurrency matrices for the clustering algorithms, and then executed the clustering algorithms (described in Sections 2.2.2.2, 2.2.2.3, and 2.2.3). The clustering algorithms gave us potential subproblems, which we used to re-organize the timelines and perform the algorithm quality measurement calculations (described in Section 2.2.4).

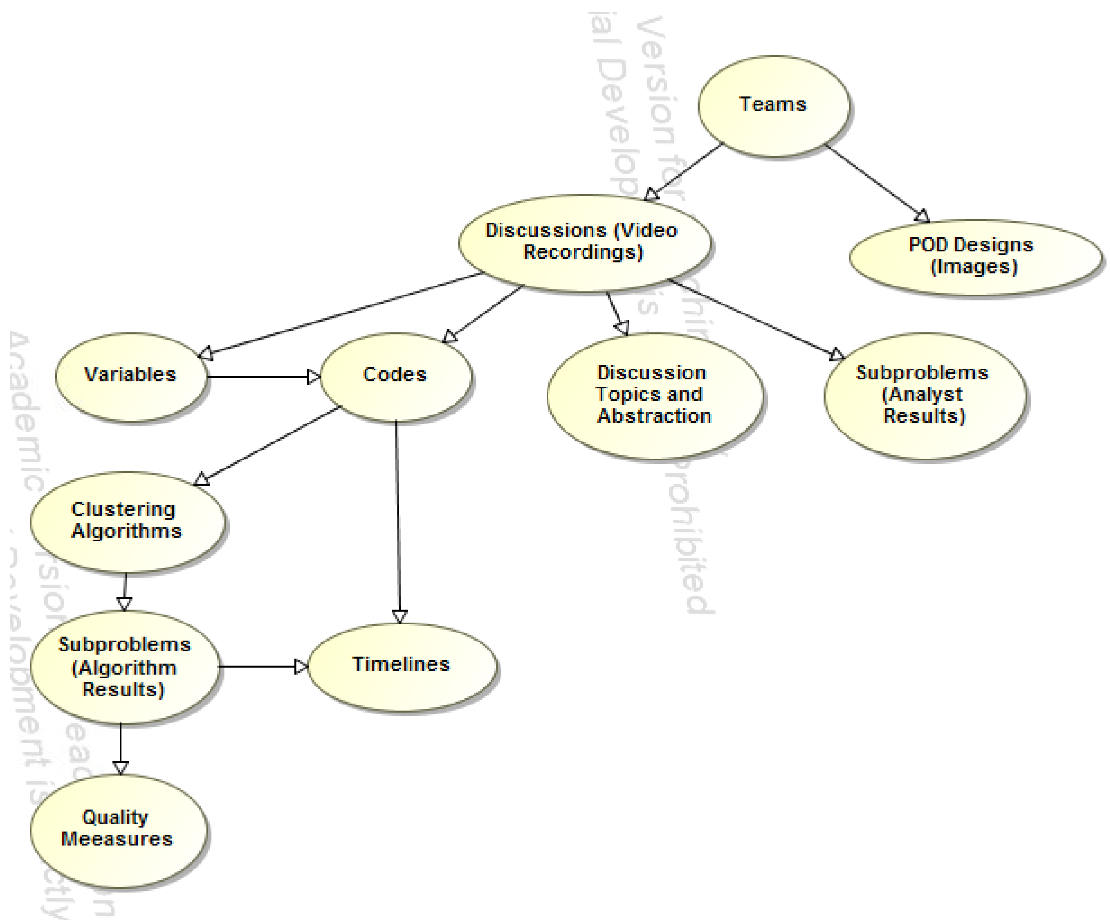


Figure 2.1: Sketch of our methods' data analysis flowchart

2.2.1 Data Collection

2.2.1.1 Participants

Two separate observational sessions were carried out: one with professionals and one with graduate students. Both studies randomly assigned participants to teams of four or five people. The professionals were all emergency preparedness planners for county health departments in Maryland, and most had a background in POD design. The 20 professionals had an average 10.78 years of experience, with a minimum of 6 months and a maximum of 42 years. The participants were

Table 2.1: Frequently Used Terms and Definitions

Term	Definition
Variable	A phrase used by researchers to capture a single aspect of the discussion
Subproblem	A selection of variables that the team chose to discuss together
Cluster	A selection of variables that one or more algorithms grouped together

separated by experience level and then randomly assigned to one of five teams. This distributed the experience levels more evenly across the five teams. The graduate students were volunteers from George Washington University, with no experience in POD design. Since experience was not a factor, these teams were simply created by randomly selecting students. This study had four teams, each with five members.

2.2.1.2 Design Problem

Both groups of participants, professionals and students, were given the same design problem. The participants were asked to create a non-medical model for a POD at a new high school. The POD design study was based on the need for rapidly dispensing prophylactic antibiotics in response to an anthrax attack [19]. The groups were given roughly an hour and a half, although the professionals were allowed to take some extra time if they believed it was needed. All teams were provided with a map of the school's floor plan. This map offered three detailed views: the school and surrounding area including parking lots and roads, the inside of the school at ground level, and an inset of the gymnasium and surrounding fitness rooms. Most teams placed stations in the gymnasium due to the open space and suitability for holding a large number of people. The teams were provided with a number of details regarding the design problem, specifically: the flow per 24 hours of residents in need of medicine, the 40 staff members provided as well as the ability to go over this limit, the four required POD stations as well as a number of other possible stations, and the time each required station took per resident. The expected outputs of the exercise were a POD layout with resident flow drawn on the high school map and a staffing plan for each station.

The four required stations included Greeting, Forms Distribution, Screening, and Medication Distribution. There were 10 other optional stations mentioned such as Flow Control and Parking Plan, but the teams were not limited in what stations they could add if another seemed necessary. Teams were instructed not to consider certain aspects of the situation, including POD staff health, details of forms, and transportation in and out of the area. The teams were also given resources in the problem such as tables, chairs, and blockades, as well as accompanying paper cutouts for the high school map. The map provided to the teams showed the entire school as well as an inset of the gymnasium area. Both can be seen in Figures 2.2 and 2.3.

2.2.1.3 Capturing Discussions

The primary data collection method was observation. Each team's discussion was captured on video camera and later reviewed by a researcher [20, 21]. We also photographed the final layouts of each team and collected all relevant documents.

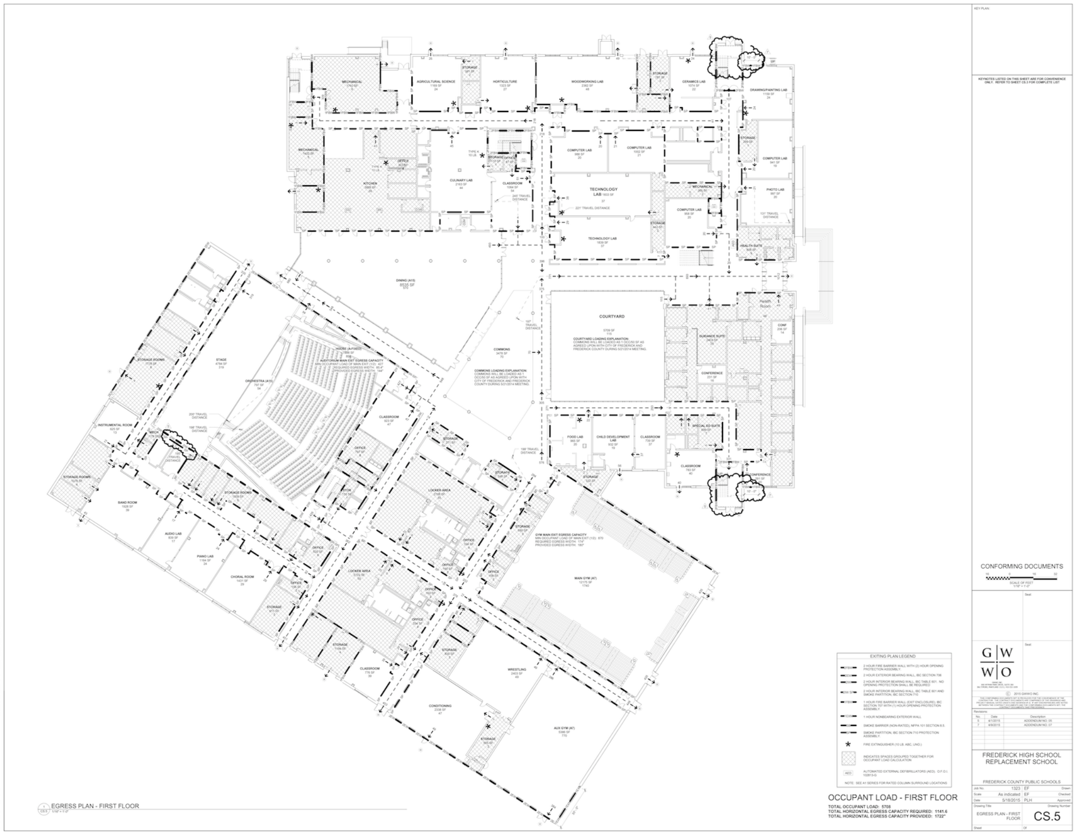


Figure 2.2: Map of the High School Supplied to POD Design Teams

The video camera was mounted on a stand next to the table where the team worked. The camera was pointed down at the layouts from overhead so that the video captured what participants drew on the layouts, when they moved areas around, and any other activities carried out on the layouts. The researchers mainly used the audio as an indication of the team's design strategy but also took physical indications into consideration. Each team's map was photographed at the end of the exercise and each team had an opportunity to explain their solution to the rest of study's participants. The photographed layouts were primarily used to re-calculate staff usage and compare final designs.

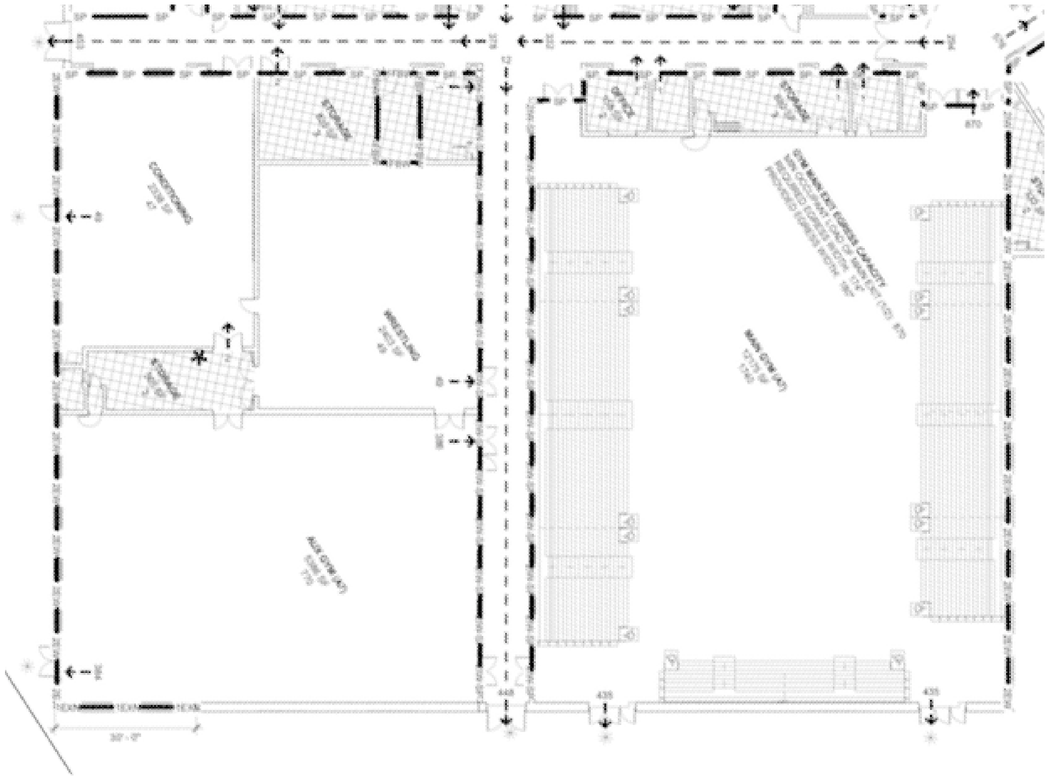


Figure 2.3: Map Inset of Gymnasium Supplied to POD Design Teams

2.2.2 Data Analysis

The data analysis aimed to capture the team's discussions and thought process while creating their solutions. The team's final designs were not compared quantitatively due to their similar nature. Rather, each team was qualitatively analyzed by comparing the order in which topics were discussed and the apparent relationship between topics. This was achieved by developing a data analysis method with a number of steps and outputs. There are a variety of methods described in the following section, all of which were used to capture the variety of data held in the video recordings and in more complicated data analysis methods such as the clustering algorithms. Each method is not necessarily dependent on one another at this level, but serves a purpose in understanding the design process and evaluating the final designs.

2.2.2.1 Development of Variables and Coding

The first step included capturing, or coding, the team's discussion into a Microsoft Excel spreadsheet. Our method was built upon grounded theory [22, 23] and process mapping [24]. We sought to develop a strong method which would capture the team's whole process rather than focus on one aspect. We believe analyzing the data after the team had completed the exercise would be similar to methods used to analyze verbal protocols [25, 26], which have been used to study human designers [27]. One key part in executing these methods is creating distinct variables that capture different topics of discussion and reflect components of the problem. These variables, paired with the frequency and order of discussion during the exercise, allowed us to quantify the team's discussion and analyze that data.

The variables were originally determined from the problem statement. As teams were reviewed, these variables were expanded or removed based on what the teams would most often discuss. After a number of iterations the variables were organized by category such as "Location Of" and then broken down into sub-codes such as "Medical Distribution". Each sub-code may have been found in multiple categories; however each category and sub-code combination captured a unique topic of discussion. A final list of the variables used in coding the teams may be seen in Figure 2.4. Any comments or actions that did not fit under a variable, but provided insight into the team's process, were captured in a notes section for that time segment. For the first two teams, multiple researchers iteratively reviewed the video footage in order to establish coder reliability and a shared understanding of what each variable represented. When the variables were concretely defined, we reviewed the video footage of each team, breaking the footage down into two minute segments. We chose two minute segments because teams would have the ability to delve into an idea and we wouldn't capture too much data per segment and lose the progression of discussion. For each segment, any variables that were discussed were marked with an 'x'.

Location Point of Entry	Staffing at Screening	Internal Layout Registration
Location Greeting	Staffing at Med Dsn	Internal Layout Medical Mgt
Location Forms Dsn	Staffing at Point of Exit	Internal Layout Patient Education
Location Screening	Staffing at Command Post	Internal Layout Behavioral Health
Location Med Dsn	Staffing at Security	Internal Layout Flow Control
Location Point of Exit	Staffing at Data Entry	Internal Layout Parking Control/Mgt
Location Command Post	Staffing at Inventory and Supplies	Flow within Point of Entry
Location Staff Break Room	Staffing at Triage	Flow within Greeting
Location Security	Staffing at Registration	Flow within Forms Dsn
Location Data Entry	Staffing at Medical Mgt	Flow within Screening
Location Inventory and Supplies	Staffing at Patient Education	Flow within Med Dsn
Location Triage	Staffing at Behavioral Health	Flow within Point of Exit
Location Registration	Staffing at Flow Control	Flow within Command Post
Location Medical Mgt	Staffing at Parking Control/Mgt	Flow within Security
Location Patient Education	Additional staffing	Flow within Data Entry
Location Behavioral Health	Calculating Staff Needs	Flow within Inventory and Supplies
Location Flow Control	POD Layout	Flow within Triage
Location Parking Control/Mgt	Flow Entry to Greeting	Flow within Registration
Include Command Post	Flow Greeting to Forms Dsn	Flow within Medical Mgt
Include Staff Break Room	Flow Forms Dsn to Screening	Flow within Patient Education
Include Security	Flow Screening to Med Dsn	Flow within Behavioral Health
Include Data Entry	Flow Med Dsn to Exit	Flow within Flow Control
Include Inventory and Supplies	Internal Layout Point of Entry	Flow within Parking Control/Mgt
Include Triage	Internal Layout Greeting	Visual Aids Hallway (flow control)
Include Registration	Internal Layout Forms Dsn	Visual Aids Station
Include Medical Mgt	Internal Layout Screening	Visual Aids Residents
Include Patient Education	Internal Layout Med Dsn	Include Drive Through
Include Behavioral Health	Internal Layout Point of Exit	Design Drive Through
Include Flow Control	Internal Layout Command Post	Drive Through Flow Entry to Greeting
Include Parking Control/Mgt	Internal Layout Security	Drive Through Flow Greeting to Forms Dsn
Staffing at Point of Entry	Internal Layout Data Entry	Drive Through Flow Forms Dsn to Screening
Staffing at Greeting	Internal Layout Inventory and Supplies	Drive Through Flow Screening to Med Dsn
Staffing at Forms Dsn	Internal Layout Triage	Drive Through Flow Med Dsn to Exit
		Parking Plan and Vehicle Traffic Flow

Figure 2.4: All Possible Variables for Coding POD Teams

Below is excerpt from a two minute segment from Professional Team 1's discussion. Note how the discussion flows from topic to topic, but there are distinct moments where a variable is discussed. For this segment Location Point of Entry, Location Greeting, Location Point of Exit, Location Flow Control and POD Layout were all coded. This excerpt shows the varying levels of ambiguity the teams used when discussing problems. While issues such as location of entry and location of exit are clearly mentioned, the location of flow control and greeting are mentioned almost in passing. One could argue the flow control and greeting comments are staff oriented, but due to the nature of the conversation and the lack of specific staff details these were coded under the location category. Certain variables, such as POD Layout, were always difficult to code due to the high level nature of the discussion. This example shows when coding that variable would be appropriate. The team talks about the general layout of the POD without explicitly labeling station locations or staffing plans.

Professional Team 1, Segment 12

Man: "The problem is, if we have them come in through this entrance were going to have to have a lot of security, because they're going to have to walk a long distance. Because here's the gym, so they have to walk all the way down here to get to the gym. So you're gonna need someone at the front to greet them and tell them where to go, someone over here to direct them..."

Woman: "Where would they go out?"

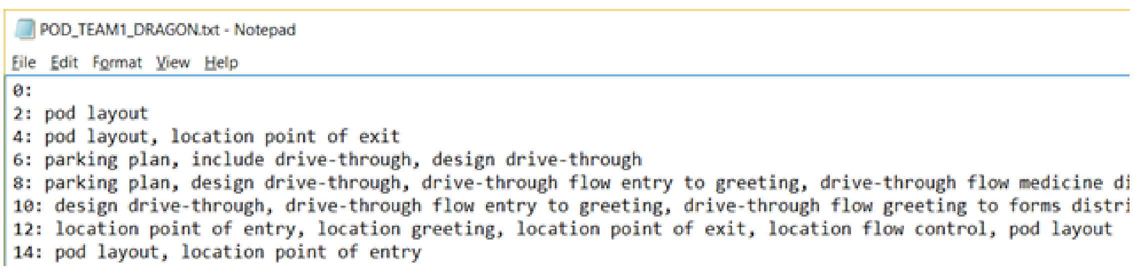
Man: "Well they can just come out the gym doors here."

Woman: "Well technically could we just forget the main entrance and come in one gym door and out the other."

Man: "Well there's a potential for a problem since you have people coming in one door and people with medicine coming out the other."

Currently, there are two methods for coding video recordings of a team's discussions. The first involves a researcher listening to the video and manually entering marks into Excel when a variable is discussed. This method was the most apparent and easily implemented at the time that the research project began, but has the drawback of taking at least an hour and a half (the length of the design session) after the exercise to code the discussion. The second and more recent method involves using voice recognition software to capture variable names. With this method, the coding can happen during the exercise with minimal interference with the team's discussion (the researcher must still say the codes aloud, which could potentially distract the team). One could only use this method after a codebook was established, since the first iterations of the codebook are based on topics frequently discussed by the teams. We were unable to use this method during an exercise, but have used it successfully to code video recordings. While watching a team's video recording, the researcher was able to repeat category and variable names as the team mentioned them. Nuance's Dragon Naturally Speaking voice recognition software was able to capture the researcher's comments and write them to a text document. Later, the text document was automatically read and imported into an Excel spreadsheet using a macro.

The Dragon Naturally Speaking program required very little preliminary voice training. Instead, the program picks up on speech patterns after listening to approximately one hour of the researcher's comments. This allowed the researcher to speak naturally and quickly even while listening to the team discuss the problem. Certain punctuation and time segment labels had to be explicitly stated during the coding, but this had little impact on the researcher's ability to keep up with the conversation. An example of the voice recognition software's output can be seen in Figure 2.5.


POD_TEAM1_DRAGON.txt - Notepad

File Edit Format View Help

```
0:
2: pod layout
4: pod layout, location point of exit
6: parking plan, include drive-through, design drive-through
8: parking plan, design drive-through, drive-through flow entry to greeting, drive-through flow medicine di
10: design drive-through, drive-through flow entry to greeting, drive-through flow greeting to forms distri
12: location point of entry, location greeting, location point of exit, location flow control, pod layout
14: pod layout, location point of entry
```

Figure 2.5: Dragon Speech Recognition Software, Coding Output

2.2.2.2 Timeline Development

Initially, two timelines were created for each team based solely on the coded discussion. The timelines were organized in different ways to provide new views or patterns within the same data. Since the teams' exercises were approximately an hour and a half long, the timelines were able to stay at the 2 minute segment level. Each timeline only shows the variables that were discussed, or the variables with at least one segment coded. Timelines show the segment number on the x-axis and the variable name on the y-axis.

The first timeline is organized by category. For example, every coded variable with the Location category was grouped and color coded together. The second timeline was organized by sub-code, so any variable with the Medical Distribution sub-code was grouped and color coded together. Both of these timelines were ordered alphabetically and illustrated the temporal relationship between category and sub-code respectively. The two types of initial timelines for Professional Team 1 can be seen in Figure 2.6 and Figure 2.7.

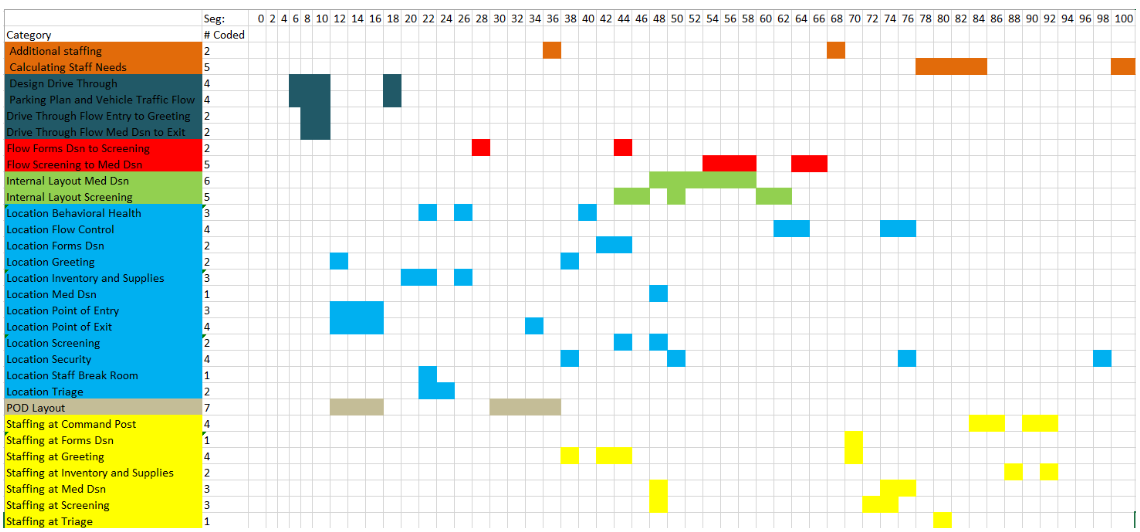


Figure 2.6: Professional Team 1 Timeline Grouped by Category

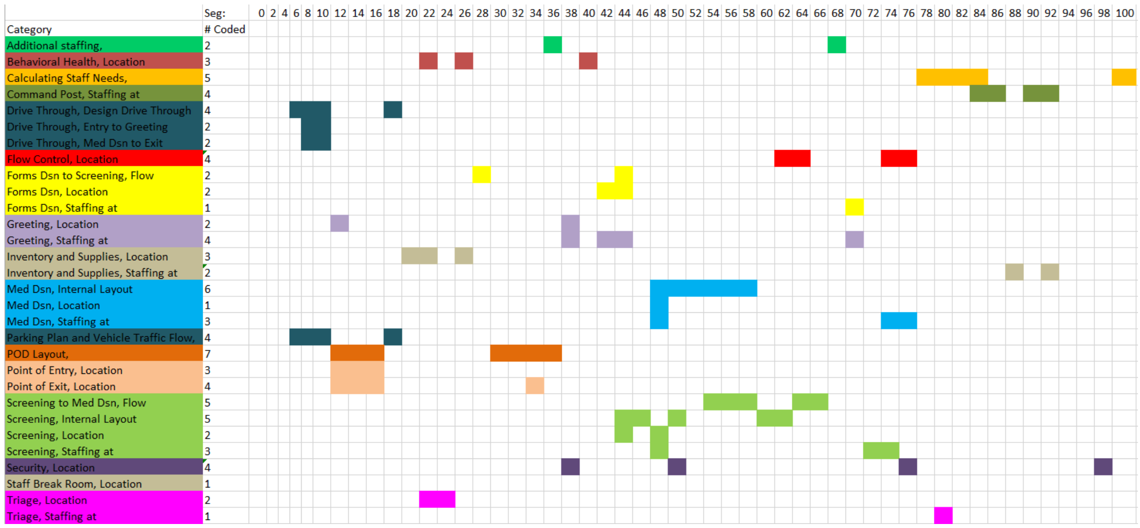


Figure 2.7: Professional Team 1 Timeline Grouped by Subcode

2.2.2.3 Creation of the Concurrency Matrix

The concurrency matrix was calculated by taking number of times variable A was discussed with variable B and dividing by the total number of times variable A was discussed. For each team, we created a $n \times n$ concurrency matrix C that has one row and column for each variable. In Equation 2.1, $n(i,j)$ is the total number of segments where variable i was coded with variable j while $n(i)$ represents the total number of segments where variable i was coded. The concurrency matrix is not symmetric and has an empty diagonal ($c_{ii} = 0$). For $i \neq j$, entry c_{ij} is determined using Equation 2.1.

$$c_{ij} = \frac{n(i,j)}{n(i)} \tag{2.1}$$

This matrix shows the frequency of two variables appearing in the same discussion segments. If a variable was not discussed at all, it was excluded from this

Staffing At	Additional staffing	Calculating Staff Needs
TRUE	FALSE	FALSE
TRUE	FALSE	TRUE
TRUE	FALSE	TRUE
FALSE	FALSE	TRUE
TRUE	FALSE	TRUE
TRUE	FALSE	FALSE
TRUE	FALSE	FALSE
TRUE	FALSE	FALSE
TRUE	FALSE	FALSE
FALSE	FALSE	FALSE
FALSE	FALSE	FALSE
FALSE	FALSE	FALSE
FALSE	FALSE	TRUE
FALSE	FALSE	FALSE

Figure 2.8: Selected time segments from Professional Team 1

matrix. Since some teams have a large number of active variables, a second concurrency matrix was created using the category names. The order of the columns is arbitrary, and the order of the rows matches that of the columns. This creates the diagonal of blank values, since the concurrency between a variable and itself is trivial. An example of the calculation follows. The calculation was done using the data pictured in Figure 2.8.

In this example, we are comparing the concurrency between Calculating Staff Needs and Staffing At. Calculating Staff Needs was coded total of 5 times for this team, while Staffing At was coded 15 times. Obviously, all 15 times are not displayed in Figure 2.8, but the other coded segments are inconsequential to the calculation. Staffing At was coded with Calculating Staff Needs 3 times. These three segments are circled in yellow in Figure 2.8. Following Equation 2.1, let i = Staffing At and j = Calculating Staff Needs. Then $n(i, j) = 3$, $n(i) = 5$, and $c_{ij} = 0.6$. This results in a concurrency of 60% between Staffing At and Calculating Staff Needs.

Note each column and row combination was calculated in the same way so the concurrency between Calculating Staff Needs and Staffing At would not be 60%. If $i =$ Calculating Staff Needs At and $j =$ Staffing At, then $n(i, j) = 3$, $n(i) = 15$, and $c_{ij} = 0.2$. The category concurrency matrix for Professional Team 1 can be seen in Figure 2.9.

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		0%	40%	0%	0%	57%	29%	45%	0%	0%	0%	0%
Include	0%		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Staffing At	32%	0%		0%	60%	0%	14%	27%	0%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	14%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	0%	0%	20%	0%		0%	0%	0%	0%	0%	0%	0%
POD Layout	21%	0%	0%	50%	0%		0%	0%	0%	0%	0%	0%
Flow	11%	0%	7%	0%	0%	0%		36%	100%	0%	0%	0%
Internal Layout	26%	0%	20%	0%	0%	0%	57%		100%	0%	0%	0%
Flow Within	0%	0%	0%	0%	0%	0%	14%	9%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		100%
Parking Plan	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	

Figure 2.9: P1 Category Concurrency Matrix

2.2.3 Identifying Subproblems and Clustering Variables

We sought to identify the subproblems that each team discussed. In this setting, a "subproblem" is a set of variables that are discussed concurrently by a team. The variables in a subproblem should be discussed at the same segment but may span across multiple segments. Because the codes identified when each team discussed different variables, we used this data to identify the subproblems. A subproblem may also have multiple variables that have no segments in common but are linked together by one or more variables. This latter part of the definition is more difficult to correctly judge and often depends on the team's apparent thought process and motivation. It introduces challenges while identifying the subproblems but cannot be completely excluded when dealing with data from an intricate design process.

In an ideal scenario for identifying subproblems, each variable would be coded during the same time segments as 1 to 3 other variables and no other variables would be coded during those time segments. In other words, each variable contained in any one set was discussed with only the other variables in that set. If each set of 2 to 4 variables fit this description we would have N clearly independent clusters. We could then confidently say that each set of variables would form a distinct subproblem and that the team created N subproblems.

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, thus linking the two sets with a usually weak relationship. Thus, 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.

We investigated four computational approaches that we call Ward's method, spectral clustering, Markov clustering, and association rule clustering. Sections 2.2.3.1, 2.2.3.2, 2.2.3.3, and 2.2.3.4 discuss these in detail.

All of the approaches begin with the coded time segments (the values of x_{it} for that team), and they all output clusters (sets of variables). The methods vary in whether the output clusters contain all the variables or some of the variables and in whether variables are allowed to belong to more than one cluster.

2.2.3.1 Ward's Method and Hierarchical clustering

Hierarchical clustering depicts how variables are related through the use of a dendrogram, as seen in Figure 2.10. In a dendrogram, the variable relationships are shaped like a tree graph. This format makes it very easy to cluster the variables and choose an appropriate level of granularity to analyze the data. Variables are grouped based on a "distance measure," specified in a dissimilarity matrix. Variables that have a low distance between one another (very similar) will be connected in the same tree level on the dendrogram, while variables with a large distance (not similar) will be connected at a higher tree level.

The algorithm starts by pairing variables that have a minimum distance between them and in the subsequent iterations the clustered variables are paired with other clusters or variables, again based on minimum distance, to form larger clusters. Although there are multiple options for specifying the dissimilarity (or distance) between data points, we used the Euclidean distance.

The Euclidean distance between variables i and j can be calculated as follows:

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

We used Ward's method [28] to cluster the variables. In particular, we used the *hclust* function in R with the method *ward.D2*. This generated a dendrogram with the clusters of variables.

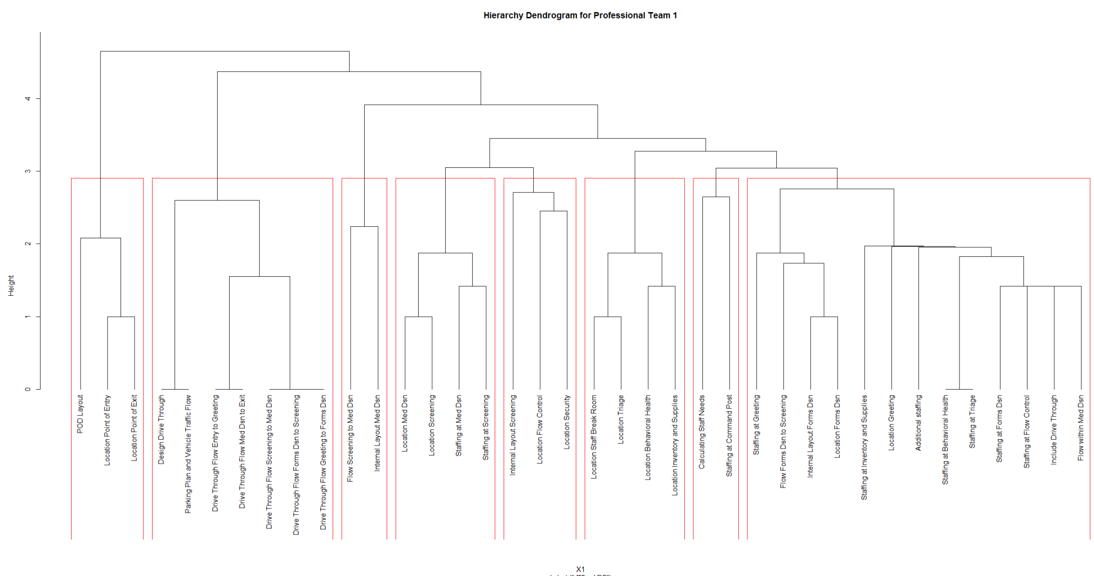


Figure 2.10: P1 Ward's Method Resulting Dendrogram

As seen in Figure 2.10, the y-axis of the dendrogram represents the distance between variables or clusters when agglomerating them using the Ward's method. To create clusters, it was necessary to select a threshold for the distance between clusters; if the distance were greater than this threshold, then the clusters would remain separate; otherwise, the clusters would be combined. Essentially, this is equivalent to drawing a horizontal line across the dendrogram at the threshold value and keeping the clusters below this line together. For the professional teams we used a threshold of 3. As seen in Figure 2.10, the professional teams' dendrograms y-axis ranged from 0 to 5. which provided a reasonable number of clusters for each of the teams. The student teams used a threshold of 0.3. The student teams' dendrograms had a y-axis ranging from 0 to 0.5, and this threshold also provided a reasonable number of clusters.

2.2.3.2 Spectral clustering

We developed a clustering method that uses spectral clustering for identifying subproblems. It is based on the techniques proposed by Sarkar [29, 30].

For variable i, let $n(i)$ be the number of time segments in which the team discussed variable i, so $n(i) = \sum_{t=1}^{T} x_{it}$. Let $n(i,j)$ be the number of time segments in which the team discussed both variables i and j. $n(i,j) = \sum_{t=1}^{T} x_{it}x_{jt}$. Let $s(i,j)$ be the number of time segments in which the team discussed variable i, variable j, or both variables i and j. $s(i,j) = n(i) + n(j) - n(i,j)$. (Note $s(i,j)$ must be at least $n(i,j)$.)

We created A, a $n \times n$ matrix in which a_{ij} is an element of A. The relative count a_{ij} was determined using Equation 2.2:

$$a_{ij} = \frac{n(i,j)}{s(i,j)} \tag{2.2}$$

We reduced the matrix A to matrix A' by removing any rows and columns that have no positive elements. The variables corresponding to these rows and columns did not occur with any other variable in the same segment. Let r be the number of rows and columns in A'. We found the r eigenvalues and eigenvectors of A'. Let D be the $r \times r$ diagonal matrix of the eigenvalues. Let V be the $r \times r$ matrix of eigenvectors.

We identified the k largest eigenvalues in the spectrum of eigenvalues. If there are k clusters of variables, there should a significant gap between the k-th largest eigenvalue and the $k+1$-st largest eigenvalue. We created a $r \times k$ matrix U that contains the eigenvectors for the k largest eigenvalues and create a $k \times k$ matrix S that contains the k largest eigenvalues (the sequence of columns of U and of S are in the same order). Each row of the product US is a point in a k-dimension space, and that point represents the corresponding variable.

We used hierarchical clustering to create a dendrogram of the variables using the distances between the points in the k-dimension space. (Note that this distance does not equal the distance $d(i, j)$ used in the Ward's clustering method.) In particular, we used the MATLAB functions pdist, linkage, and dendrogram.

As in the hierarchical clustering, we set a threshold to generate clusters from the dendrogram. For each team, the threshold was chosen to create clusters that included most pairs of variables with a large relative count (large value of a_{ij}) and clusters with similar distance values.

2.2.3.3 Markov clustering

Our Markov clustering approach uses the algorithm developed by Stijn van Dogen [31]. We downloaded and used the MCL application [32] written by Stijn van Dongen. The application applies van Dongen's Markov Clustering Algorithm to input data in a specific textual format, and outputs clusters in a textual format.

As input data for each team, we created a $n \times n$ concurrency matrix C that has one row and column for each variable. The concurrency matrix is not symmetric and has an empty diagonal ($c_{ii} = 0$). For $i \neq j$, entry c_{ij} is determined using Equation 2.1 For more details about the concurrency matrix, see Section 2.2.2.3.

We then converted the concurrency matrix values to a text format that is accepted by the MCL application. This format involved listing the relationship between each set of variables in the following way:

Variable_A Variable_B 0.5

where this line shows the concurrency between Variable A to Variable B is 50%. Notice the underscore character in the variable names, which is necessary as the space character separates the two variable names and concurrency value. Every combination of variables from the concurrency matrix was entered into a text document in this format. The MCL application was then run with the following command:

mcl input_file.txt –abc -o output_file.txt

This command accepts optional arguments, such as $-I$ to specify the Inflation factor. We experimented with different values of the input *–I*. Larger values of the inflation factor, such as 6 or 8, create more numerous, smaller clusters; smaller values, such as 0.5 or 1, result in larger clusters. The POD design teams were processed using the default inflation value of 0.6.

2.2.3.4 Association rule clustering

In machine learning, association rules are utilized to discern relationships between sets of items that occur together [33]. 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 variables and used those association rules to create clusters of variables.

Three measures are typically used when identifying association rules: the support, confidence and lift. In our approach, the support is the number of time segments in which a variable i is coded for a given team: $Supp(i) = n(i)$. The confidence is the proportion of time segments in which, if variable i was coded, then j was also coded (see Equation 2.3).

$$Conf(i \Rightarrow j) = \frac{Supp(i \cup j)}{Supp(i)} = \frac{n(i, j)}{n(i)} \tag{2.3}$$

The lift is the proportion of the observed support of i and j coded together to that expected if i and j were independent (see Equation 2.4).

$$Lift(i \Rightarrow i) = \frac{Supp(i \cup j)}{Supp(i) \times Supp(j)} = \frac{n(i, j)}{n(i)n(j)} \tag{2.4}$$

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., $i \Rightarrow j$ and $j \Rightarrow i$ will be generated as two separate rules). However, in our context, we are interested only in whether i and i typically occur together and are in the same subproblem. Therefore, we combined such permutations in order to derive a final set of subproblems for each team.

For each team, we generated association rules using the coded data (the values of x_{it}). We used the packages arules and arulesViz in R [34] in order to run the association rule algorithm.

For the POD design teams we used a support level of 0.033 and a confidence of 0.5. A low level of confidence had to be used because the data was sparse compared to typically large sets of transactional data from which association rules are often generated.

Each association rule established a relationship between two or more variables. We clustered the variables by the following policy: if variables i and j are together in an association rule, then variables i and j are in the same cluster.

2.2.4 Cluster quality 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 the following techniques to produce the evaluation measures: the Davies–Bouldin index, the Dunn index, the silhouette coefficient, the number of high-concurrency pairs clustered together, and the number of high relative count pairs clustered together. These methods were also described and discussed by Morency et al. [35].

We used two versions of the Dunn index α [36]. Both versions use the "diameter" of each cluster (Δ_c) and the "distance" (D_{ce}) between clusters S_c and S_e to calculate α.

$$\alpha = \frac{\min\{D_{ce} : 1 \leq c < e \leq C\}}{\max\{\Delta_c : 1 \leq c \leq C\}} \tag{2.5}$$

The two versions define the diameter and distance in different ways. Dunn [36] defined the diameter and distance as follows:

$$\Delta_c = \max_{i,j \in S_c} d(i, j) \tag{2.6}$$

$$D_{ce} = \min_{i \in S_c, j \in S_e} d(i, j) \tag{2.7}$$

We also used a modified version based on the centroids of cluster. The centroid μ_c of a cluster S_c with $|S_c|$ variables is the average value of those variables:

$$\mu_{ct} = \frac{1}{|S_c|} \sum_{i \in S_c} x_{it} \tag{2.8}$$

The modified diameter and distance values are based on the Euclidean distances to these clusters.

$$\Delta_c = \frac{1}{|S_c|} \sum_{i \in S_c} d(i, \mu_c) \tag{2.9}$$

$$D_{ce} = d(\mu_c, \mu_e) \tag{2.10}$$

Davies [37] defined a cluster similarity measure based on the "dispersion" of each cluster and the distances between clusters. These measures are equivalent to the modified diameter and distance that we used for the modified Dunn index. Let R_{ce} be the similarity of clusters S_c and S_e.

$$R_{ce} = \frac{\Delta_c + \Delta_e}{D_{ce}} \tag{2.11}$$

For each cluster, the largest similarity is used, and the Davies-Boulding index $\bar{R}$ is the average of these similarity values.

$$\bar{R} = \frac{1}{C} \sum_{c=1}^{C} \max_{e \neq c} R_{ce} \tag{2.12}$$

Rousseeuw [38] defined the silhouette measure, which describes how well each item lies within its cluster. Let $a(i)$ be the average distance from variable i (in cluster S_c) to the other variables in the same cluster. Let $\bar{d}(i, e)$ be the average distance from variable i (in cluster S_c) to variables in cluster S_e. Let $b(i)$ be the average distance from variable i (in cluster S_c) to variables in the closest cluster.

$$a(i) = \frac{1}{|S_c| - 1} \sum_{j \in S_c} d(i, j) \tag{2.13}$$

$$\bar{d}(i, e) = \frac{1}{|S_e|} \sum_{j \in S_e} d(i, j) \tag{2.14}$$

$$b(i) = \min_{e \neq c} \bar{d}(i, e) \tag{2.15}$$

From these measures, the silhouette value $s(i)$ can be determined. If variable i is the only item in cluster S_c, then $s(i) = 0$ [38]. Let $\bar{S}$ be the average silhouette value.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2.16}$$

$$N = \sum_{c=1}^{C} |S_c| \tag{2.17}$$

$$\bar{S} = \frac{1}{N} \sum_{c=1}^{C} \sum_{i \in S_c} s(i) \tag{2.18}$$

We also defined measures to indicate how many "close" pairs of variables were in the same cluster. The closeness was based on the relative count a_{ij} used for the spectral clustering and on the concurrency c_{ij} used for the Markov clustering.

Let $z_{ij} = 1$ if there exists a cluster S_c such that $i \in S_c$ and $j \in S_c$ (that is, they are in the same cluster) and 0 otherwise. Let $N_p^a(v)$ be the number of variable pairs with a relative count at least v. Let $N_c^a(v)$ be the number of variable pairs in the same cluster with a relative count at least v. Let $I(X)$ be the indicator function that returns 1 if X is true and 0 otherwise.

$$N_p^a(v) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(a_{ij} \geq v) \tag{2.19}$$

$$N_c^a(v) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(a_{ij} \geq v) z_{ij} \tag{2.20}$$

Let $N_p^c(v)$ be the number of variable pairs with a concurrency value at least v. Let $N_c^c(v)$ be the number of variable pairs in the same cluster with a concurrency value at least v. (Recall that the concurrency value is not necessarily symmetric, so c_{ij} may not equal c_{ji}.)

$$N_p^c(v) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (I(c_{ij} \geq v) + I(c_{ji} \geq v)) \tag{2.21}$$

$$N_c^c(v) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} z_{ij}(I(c_{ij} \geq v) + I(c_{ji} \geq v)) \tag{2.22}$$

We then repeated the analysis with a different distance metric and centroid definition. In particular, we defined the modified distance $d'(i,j)$ as follows:

$$d'(i,j) = \frac{1}{s(i,j)} \sum_{t=1}^{T} |x_{it} - x_{jt}| \tag{2.23}$$

The new centroid μ'_c has the median value for each segment. Because there are only two values (0 and 1), $\mu'_{ct} = 1$ if and only if $\sum_{i \in S_c} x_{it} \geq |S_c|/2$.

2.3 Coding Results

2.3.1 Original Timelines

Professional Team 1's timelines can be seen in Figures 2.6 and 2.7. Like many of the professional teams, P1 covered a large number of variable over about 90 minutes. Their two most talked about categories were the Location and Staffing At categories, and there was no one particularly dominant subcode. Their discussion starts outside with the parking lot and drive through being the only topics. They then moved inside the POD and started designing at a high level; their discussion involved the POD Layout variable as well as many location variables. P1 then stops discussing the location of station and instead covers the internal layout, flow, and general relationship between the two main stations: medication distribution and screening. The team ends their discussion by talking about the staffing at many of the required stations, as well as some optional ones.

The timelines for Professional Team 2 can be seen in Figures 2.11 and 2.12. P2 discussed the fewest amount of different variables out of all of the professional teams, but they spent the most time discussing the problem. The covered the most variables in the Location category, but spent a large portion of their discussion talking about the Internal Layout, Flow, and Staffing categories. The team discussed the Med Dsn and Screening subcodes the most, with other subcodes not coming close in code volume. Like P1, P2 started with their discussion outside and then briefly talked about station locations at a very high level. Their discussion quickly found its way to the Med Dsn and Screening subcodes. They started with the Internal Layout category but shortly after discussed the Flow within and between these stations simultaneously. They also discussed Staffing At Med Dsn sporadically throughout the exercise and concluded with staffing considerations for other stations as well.

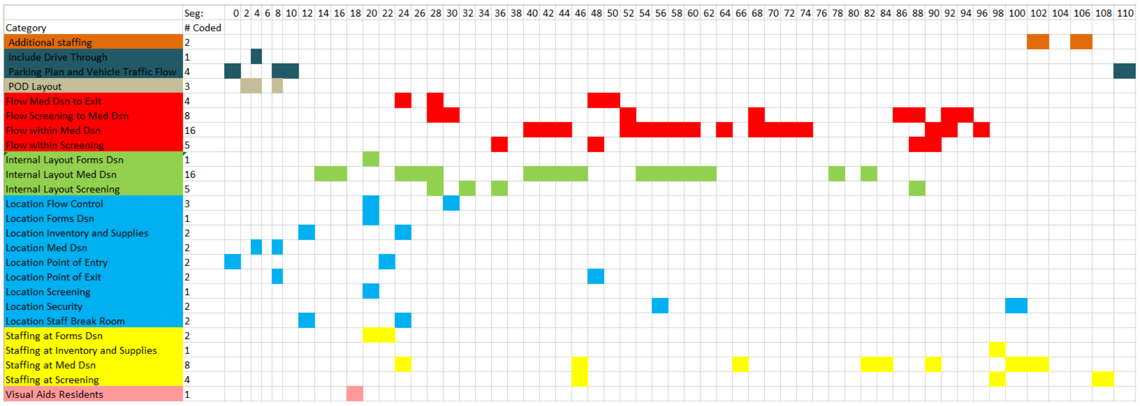


Figure 2.11: Professional Team 2 Timeline Grouped by Category

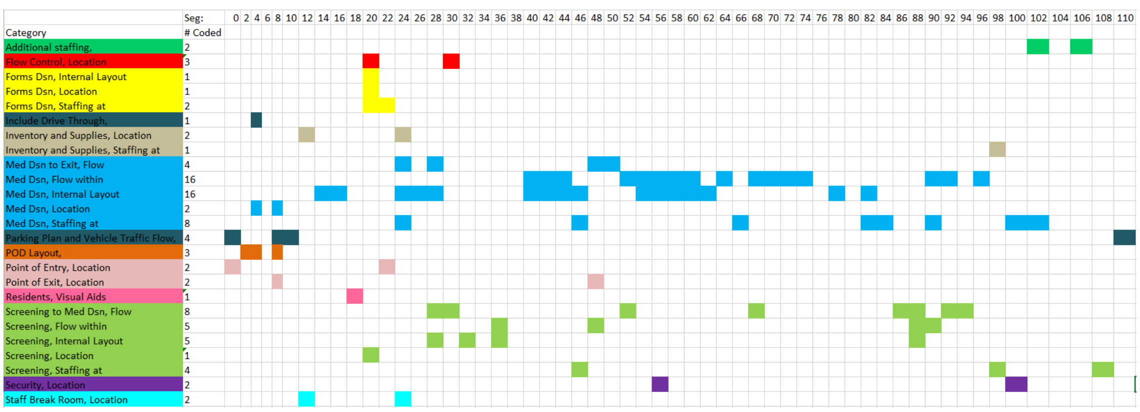


Figure 2.12: Professional Team 2 Timeline Grouped by Subcode

Professional Team 3's timelines can be seen in Figures 2.13 and 2.14. P3 had the shortest discussion time, covering only approximately 60 minutes. They focused mostly on the Staffing at category, but also spent some time on Location and Flow variables. Their subcode timeline shows that Med Dsn and Screening were discussed the most thoroughly, but other subcodes were discussed throughout the exercise. Unlike the previous teams, P3 starts with a high level staffing discussion and doesn't cover any of the parking lot plan or design. Instead, the team talks about staffing and intermittently discusses either station location or Flow within required stations such as medication distribution or screening. They end their session as they began, by talking about staffing at both the station level and at the higher POD level.

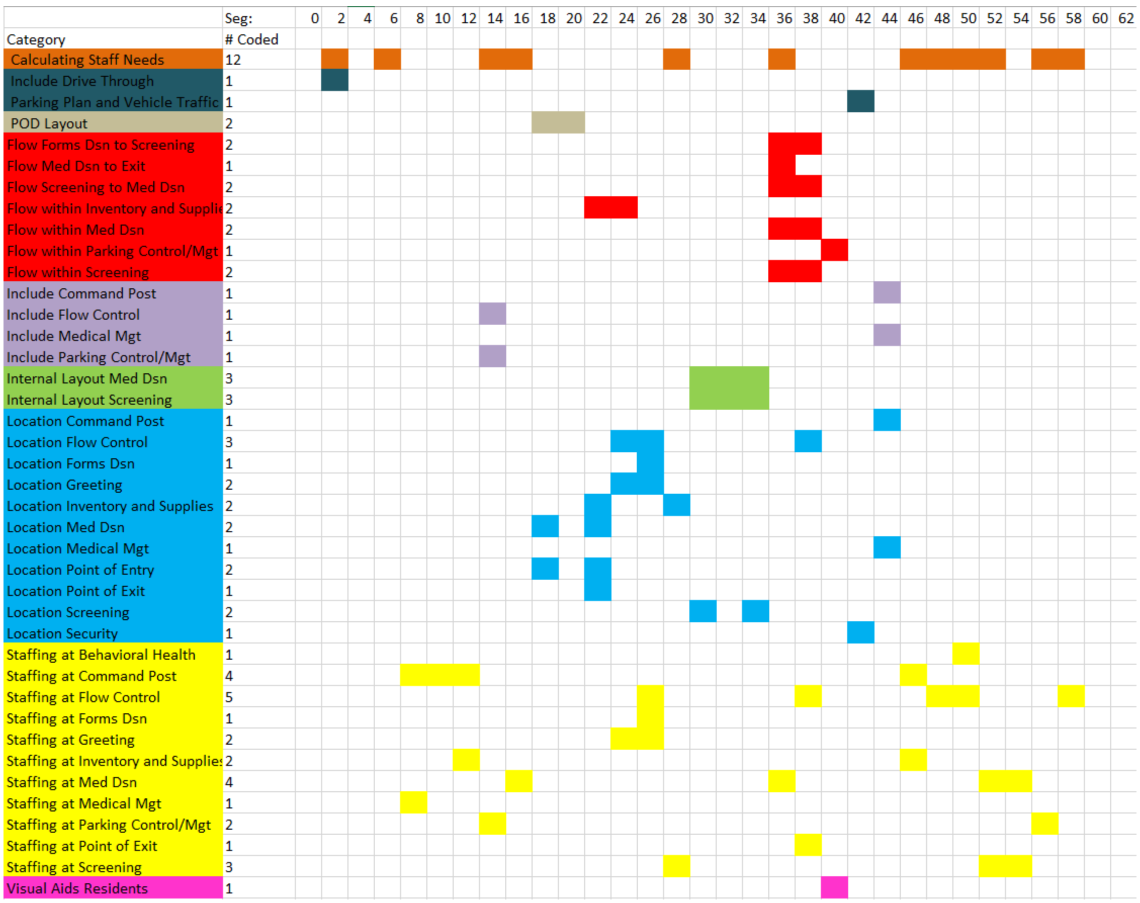


Figure 2.13: Professional Team 3 Timeline Grouped by Category

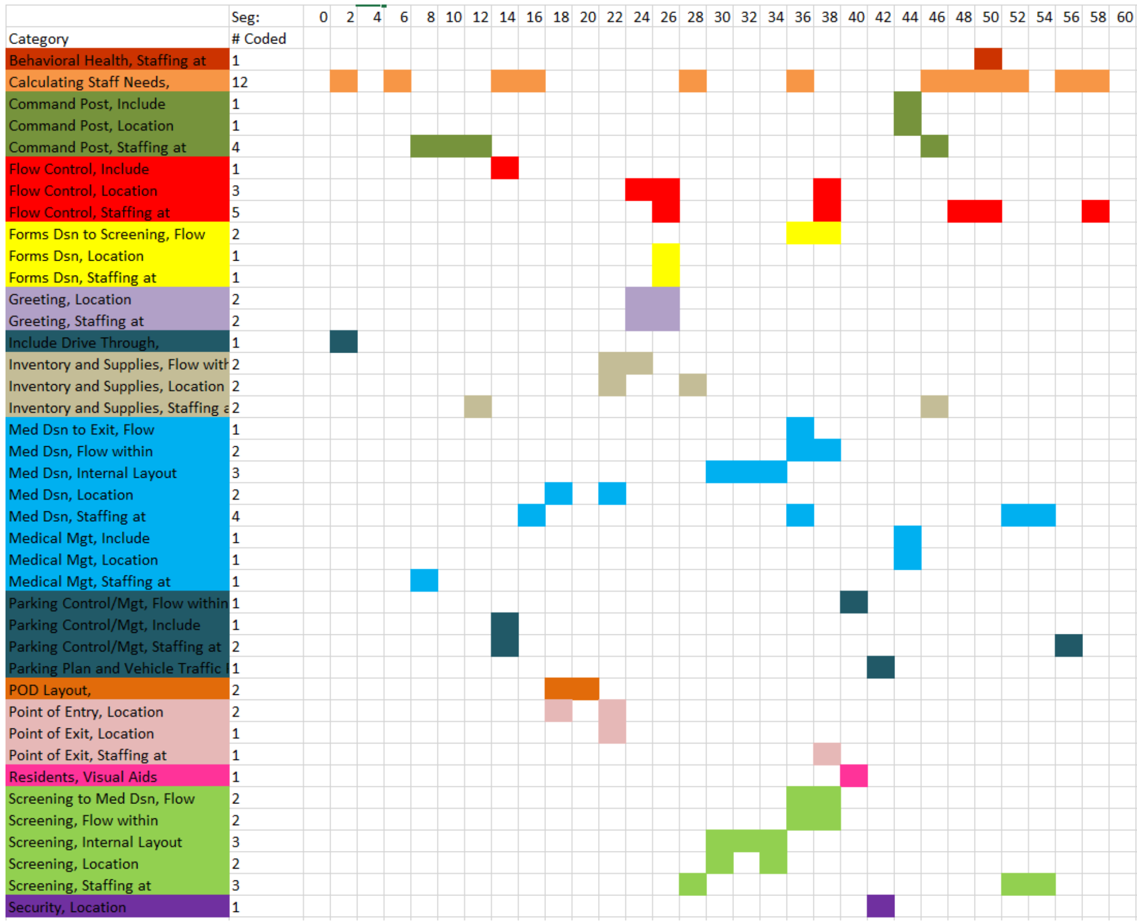


Figure 2.14: Professional Team 3 Timeline Grouped by Subcode

Figures 2.15 and 2.16 show Professional Team 4's timelines. P4 covered a few categories equally, including Staffing At, Location, and Flow. They also discussed a few Internal Layout variables throughout the process. Unlike the previous teams, P4 works on these categories simultaneously and begins discussing them after a brief discussion of the parking plan and outside design. Their subcodes appear to relatively diverse, with more optional stations like Command Post or Triage being coded than seen in previous teams. However, there is some chronological progression seen in the subcode timeline. The team starts outside with the Parking Plan and related variables, and then shortly after discusses the Point of Entry and Point of

Ext variables. After that they briefly discuss greeting and then Forms Dsn. They conclude with a long discussion involving both Med Dsn and Screening variables. Paired with the category timeline, this shows that the team solved the problem by logically through the POD, station by station rather than discussing one category such as Staffing for the entire POD.

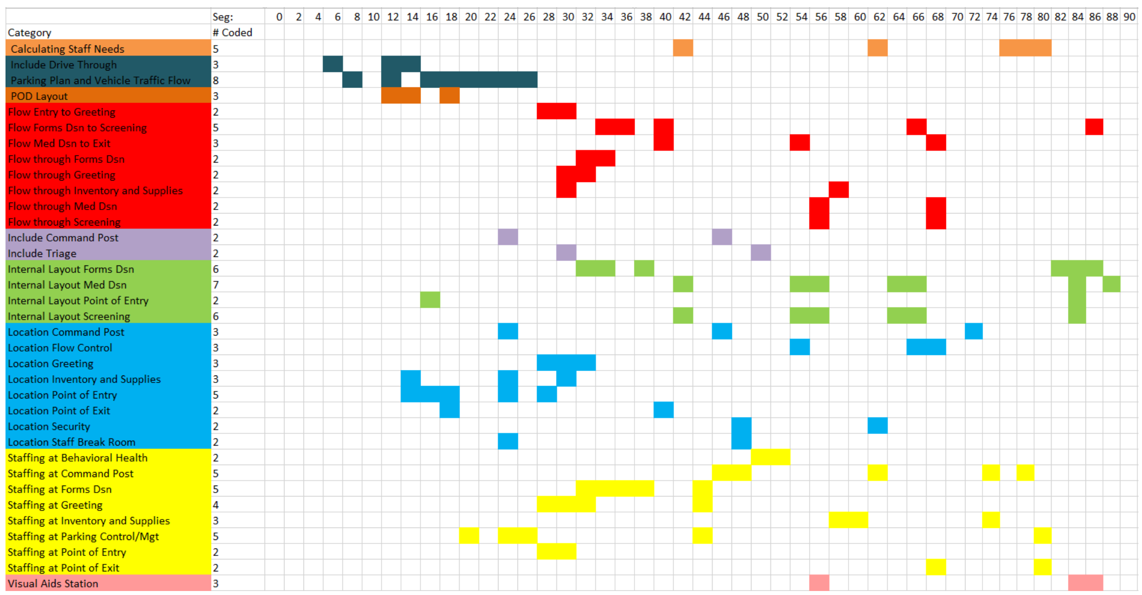


Figure 2.15: Professional Team 4 Timeline Grouped by Category

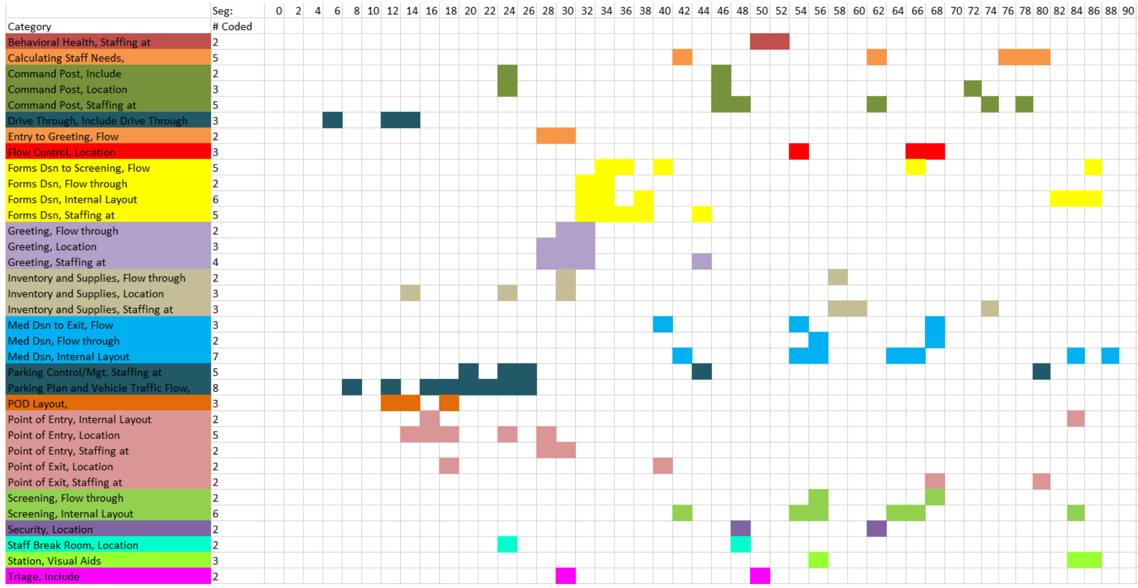


Figure 2.16: Professional Team 4 Timeline Grouped by Subcode

The timelines for Professional Team 5 can be seen in Figures 2.17 and 2.18. This team spent the second most amount of time discussing the problem. During this time, they covered a large amount of Staffing and Location variables. Their discussion involved very little Internal Layout variables, with the exception of the Screening subcode. Screening was discussed thoroughly and frequently, meaning that this area was the most important or difficult to P5. Like previous teams however, P5 started with a discussion of the Parking Plan variable and then moved inside with the Location of Point of Entry and other subcodes. Unlike other teams, P5 finished their design with a large a mount of Staffing variables, and revisit these variables once or twice. This team put a lot of consideration into their staffing plan, and viewed it as an important topic.

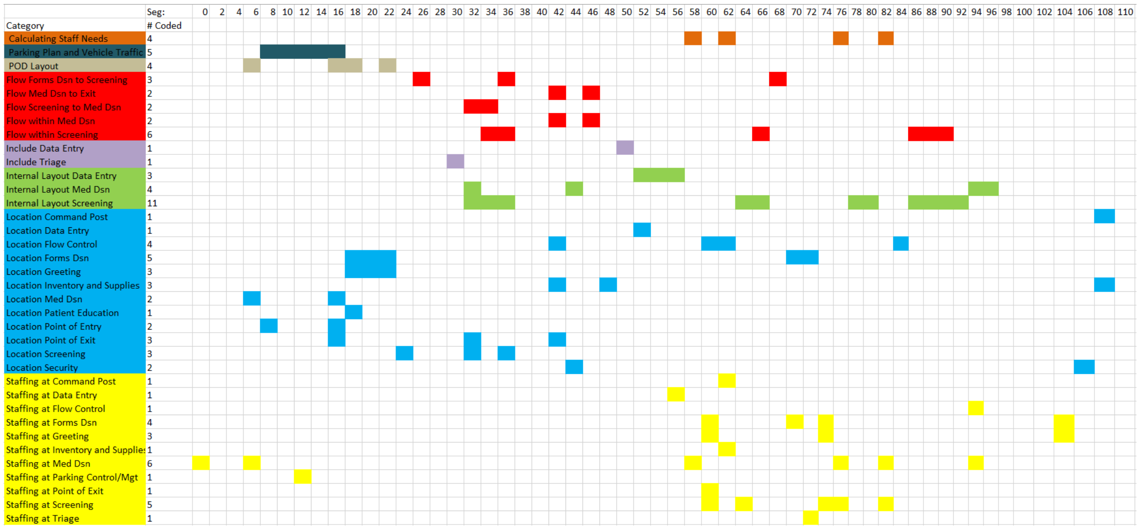


Figure 2.17: Professional Team 5 Timeline Grouped by Category

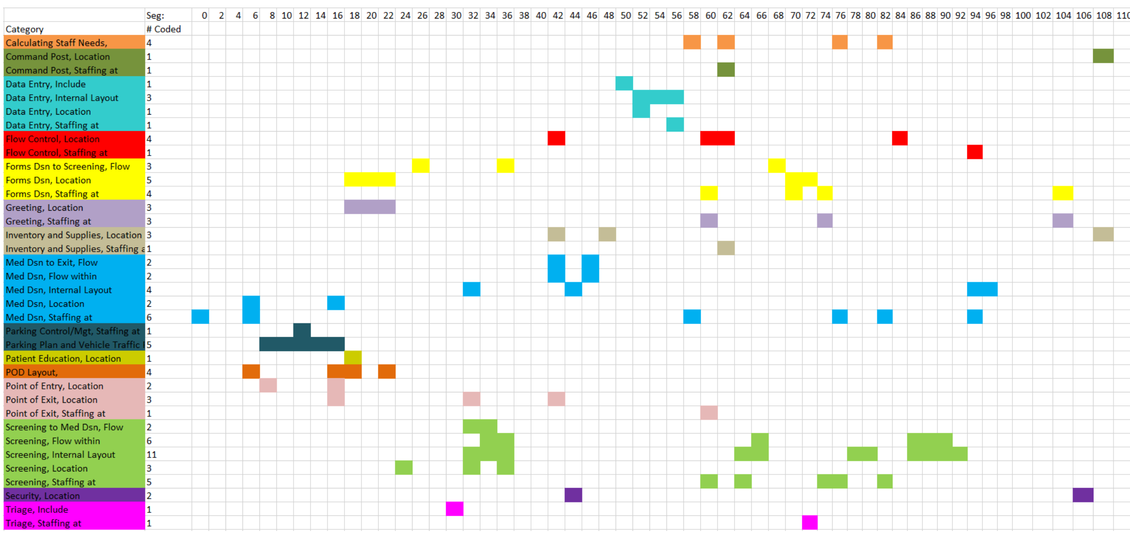


Figure 2.18: Professional Team 5 Timeline Grouped by Subcode

Student Team 1's timelines can be seen in Figures 2.19 and 2.20. S1 discussed the fewest amount of variables in the shortest amount of time out of all of the teams. They failed to discuss forms distribution, but did talk about the other required stations. Their two most talked about categories were Location and Flow. The most talked about variable was POD Layout. This team starts the discussion with the high level POD Layout variable, and then begins discussing the location of essential stations like entry, exit, and medication distribution. The team makes decision regarding flow in and between these stations shortly after deciding the location, and never revisits these topics. Compared to the professional teams, S1 had a short and high level discussion about the POD without considering the finer details of each station.

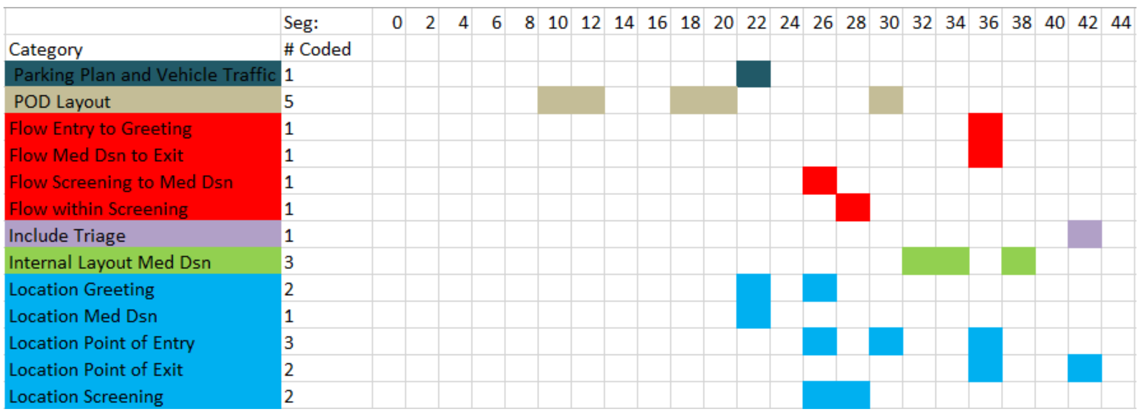


Figure 2.19: Student Team 1 Timeline Grouped by Category

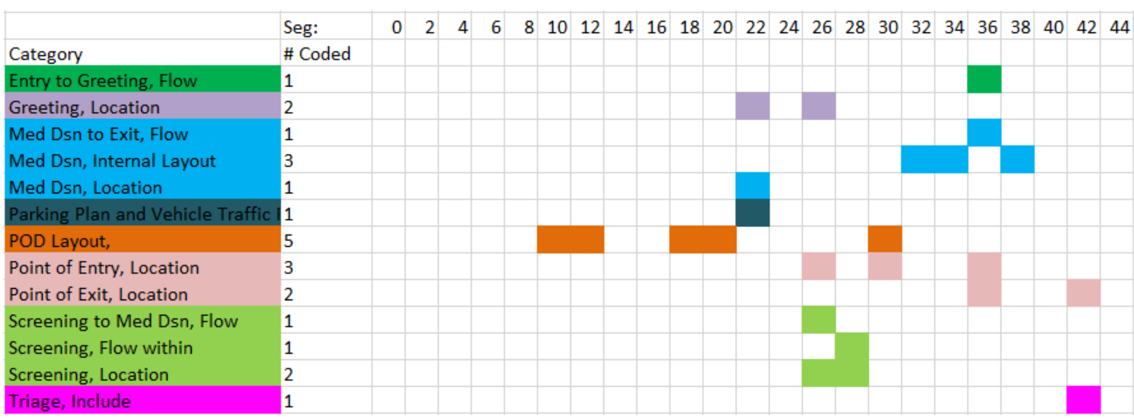


Figure 2.20: Student Team 1 Timeline Grouped by Subcode

Figures 2.21 and 2.22 show Student Team 2's timelines. Although their discussion time is relatively short, S2 managed to cover many variables. Looking at the category timeline, there is a chronological progression from the Location, Flow, and Internal Layout categories. Their subcode timeline shows that the stations Forms Dsn, Med Dsn, and Screening were discussed thoroughly. This team begins by discussing high level layout and staff needs, before moving into the deciding the location of stations. Although they discuss staff sporadically through the exercise, most of the staffing decisions were made at a high level in the first few segments. The location discussion is dominated by the Location Point of Entry variable, something unseen in previous teams. The following segments involve the Flow and Internal Layout categories, and primarily involve the Screening and Med Dsn subcodes. These two categories were discussed simultaneously, meaning the team's decisions were dependent on one another.

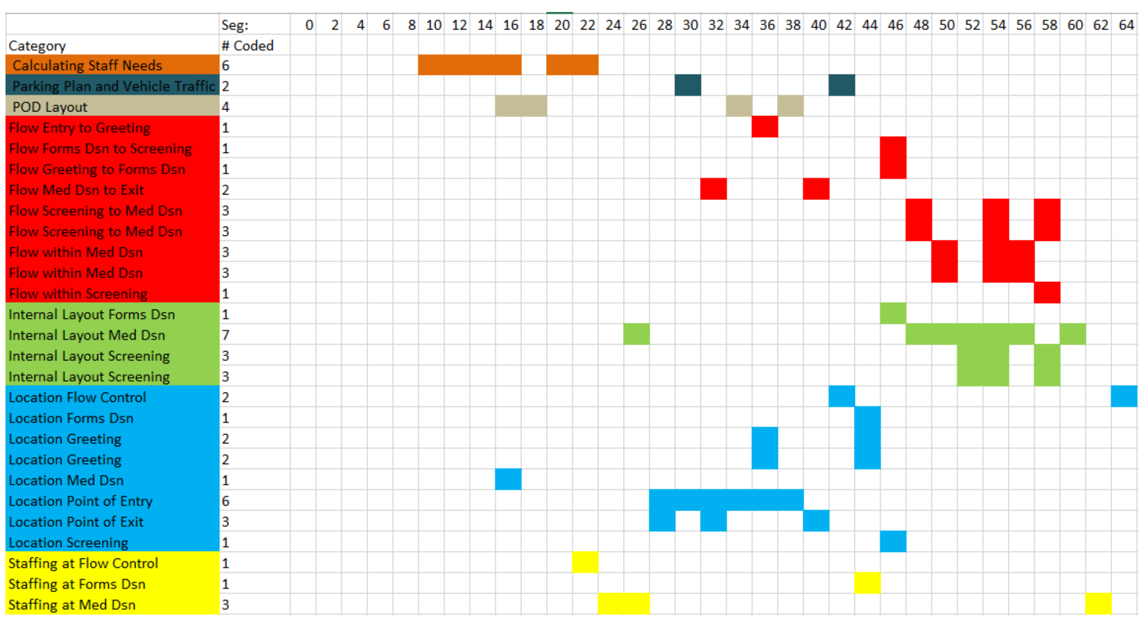


Figure 2.21: Student Team 2 Timeline Grouped by Category

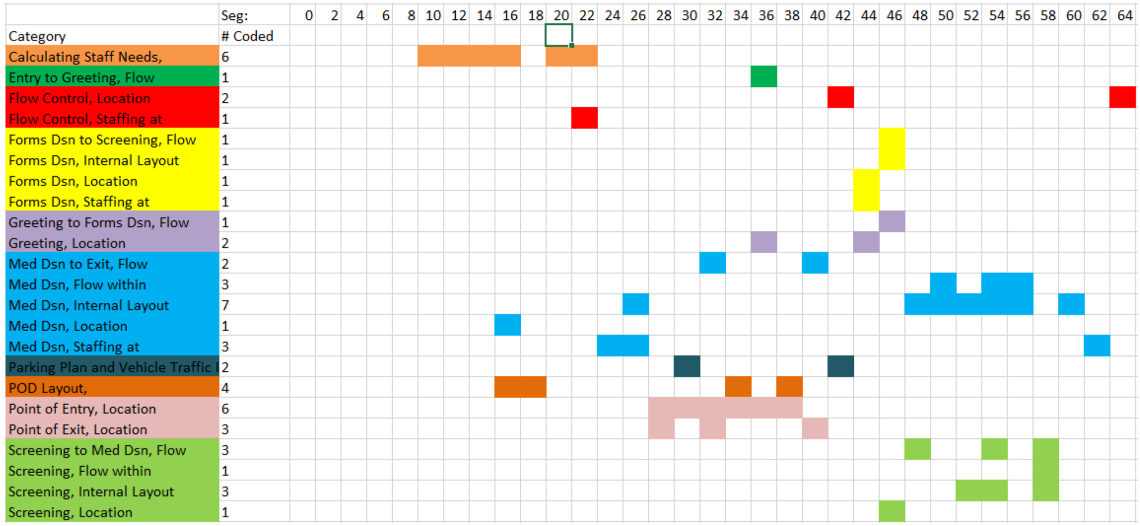


Figure 2.22: Student Team 2 Timeline Grouped by Subcode

The timelines for Student Team 3 can be seen in Figures 2.23 and 2.24. S3's discussion is similar to S2, in that both covered many variables and lasted longer than the other two student teams. S3 started by deciding the location and layout of the POD. This can be seen in the category timeline, where the Location category dominates the discussion for the first half of the exercise, along with the Parking Plan variable and POD Layout. The team then worked on the Internal Layout category, focusing on the Med Dsn subcode. This shows that while the team did make decisions about the whole POD, they believed that the medication distribution area would be he most complicated and need more attention. Towards the end of the exercise, S3 also briefly discussed staffing at the required stations, but did not spend much time on any one station.

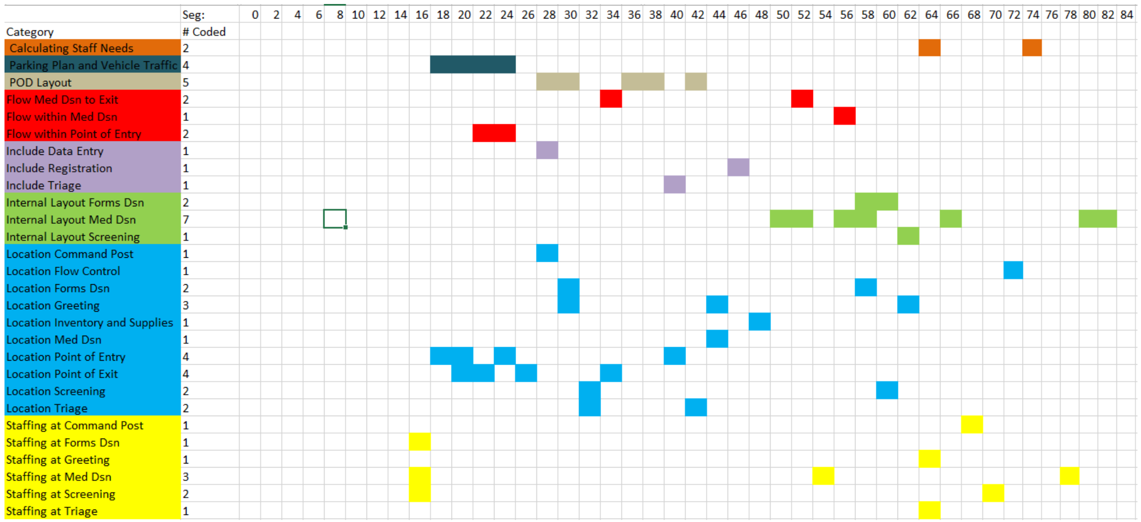


Figure 2.23: Student Team 3 Timeline Grouped by Category

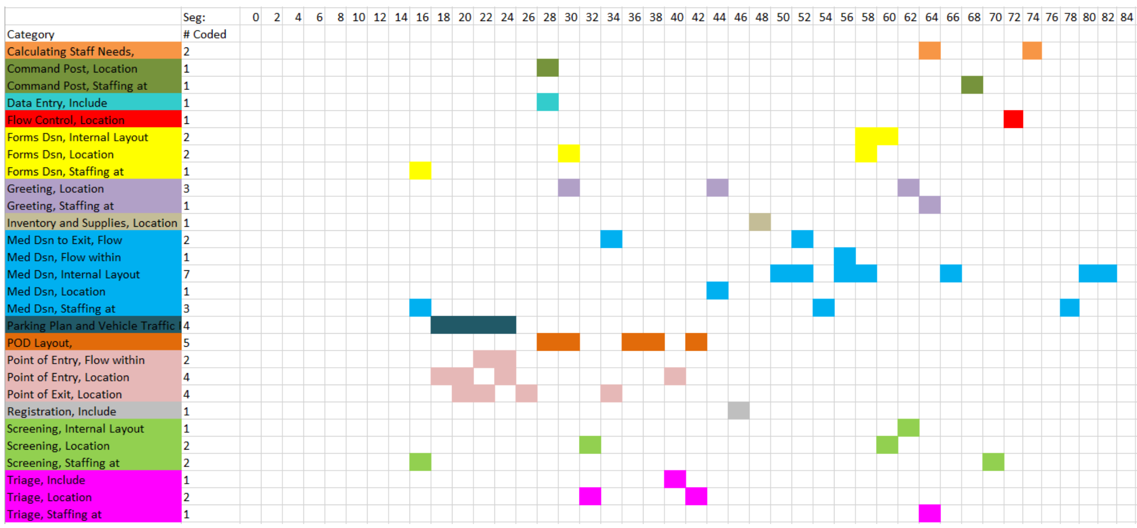


Figure 2.24: Student Team 3 Timeline Grouped by Subcode

Student Team 4's timelines can be seen in Figures 2.25 and 2.26. Like S1, S4 covered a relatively few number of variables and discussed this problem for a short period of time. Their most talked about variable was Calculating Staff Needs, and the Staffing category has a considerable prescience in the category timeline. The team's subcodes are varied, with the Greeting subcode being discussed nearly the entire time. S4 chose to start their discussion with the location of indoor variables rather than anything like Parking Plan. The discussion includes the Internal Layout of Med Dsn and Internal Layout of Screening variables, but then becomes largely about staffing.

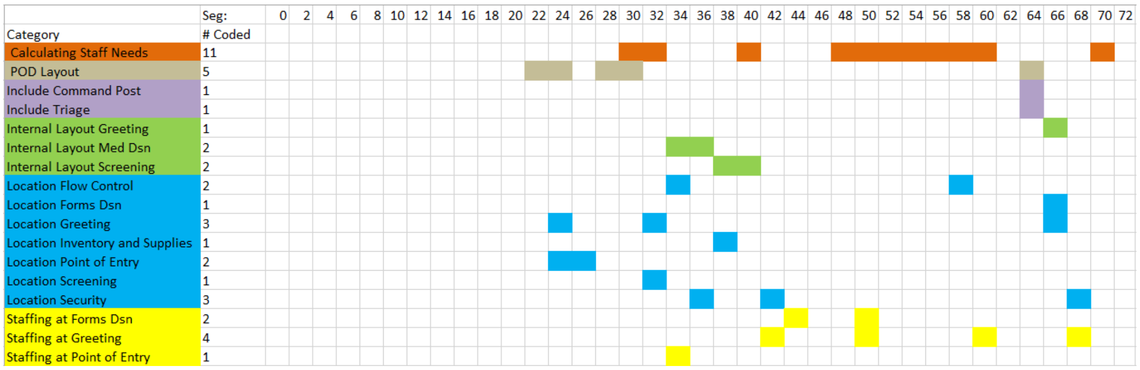


Figure 2.25: Student Team 4 Timeline Grouped by Category

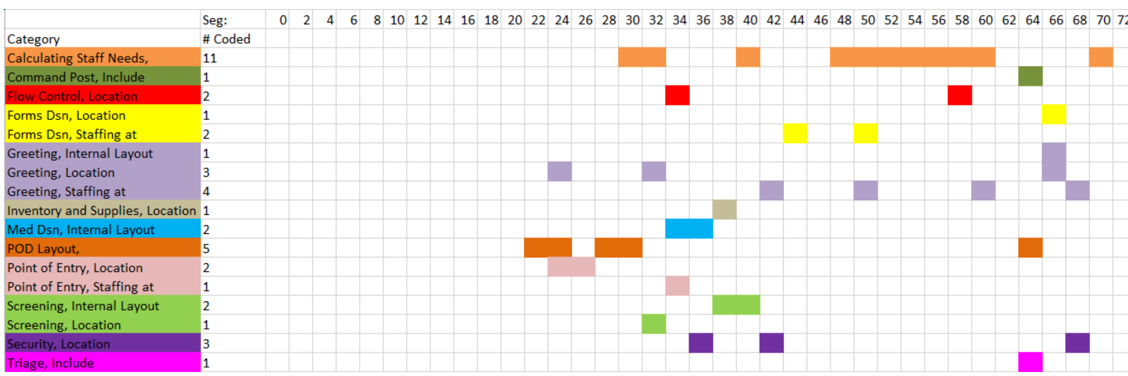


Figure 2.26: Student Team 4 Timeline Grouped by Subcode

2.3.2 Category Concurrency Matrices

The category concurrency matrices for all of the teams can be seen in Figures 2.27, 2.28, 2.29, 2.30, 2.31, 2.32, 2.33, 2.34, and 2.35. These matrices are color coded such that values closer to 100%, or perfect concurrency, are dark green and values closer to 0%, or no concurrency, are white. Note that the diagonal of each matrix is blank, since a category's concurrency with itself is irrelevant. Many teams have a high concurrency between the Parking Plan and Location categories, generally due to the Location Point of Entry and Location Point of Exit variable having a relationship with where the residents would park. Another frequent relationship can be seen between the Flow, Flow Within, and Internal Layout categories. Teams usually discussed the internal layout in terms of how residents would move throughout the station, making these two topics usually related.

Most teams also have concurrency between the Location category and the others, or vice versa. The POD Layout category has concurrency is every team, meaning that it was always talked about in respect to another topic. Both of these characteristics are similar to what was seen in the category and subcode timelines. Teams tended to start at a high level, making decisions about locations and general layout before delving into the more detail oriented categories like Intern Layout or Flow.

Note the amount of highly concurrent categories in the professional teams versus the student teams. This is similar to the trend seen in the timelines (Section 2.3.1) where students discussed fewer variables than the professionals. In Figure 2.32, S1 has hardly any concurrency between categories, and everything is related to Location. In the concurrency matrix for P4, however, there are many more pairs of categories with large concurrency values, as seen in Figure 2.30.

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		0%	40%	0%	0%	57%	29%	45%	0%	0%	0%	0%
Include	0%		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Staffing At	32%	0%		0%	60%	0%	14%	27%	0%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	14%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	0%	0%	20%	0%		0%	0%	0%	0%	0%	0%	0%
POD Layout	21%	0%	0%	50%	0%		0%	0%	0%	0%	0%	0%
Flow	11%	0%	7%	0%	0%	0%		36%	100%	0%	0%	0%
Internal Layout	26%	0%	20%	0%	0%	0%	57%		100%	0%	0%	0%
Flow Within	0%	0%	0%	0%	0%	0%	14%	9%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		100%
Parking Plan	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	

Figure 2.27: Professional Team 1 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		0%	38%	0%	0%	67%	27%	19%	15%	0%	100%	50%
Include	0%		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Staffing At	42%	0%		50%	0%	0%	9%	24%	10%	0%	0%	0%
Additional Staffing	0%	0%	8%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	0%	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%
POD Layout	17%	0%	0%	0%	0%		0%	0%	0%	0%	100%	25%
Flow	25%	0%	8%	0%	0%	0%		14%	25%	0%	0%	0%
Internal Layout	33%	0%	38%	0%	0%	0%	27%		50%	0%	0%	0%
Flow Within	25%	0%	15%	0%	0%	0%	45%	48%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	8%	0%	0%	0%	0%	33%	0%	0%	0%	0%		0%
Parking Plan	17%	0%	0%	0%	0%	33%	0%	0%	0%	0%	0%	

Figure 2.28: Professional Team 2 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		50%	24%	0%	8%	50%	50%	67%	60%	0%	0%	100%
Include	10%		6%	0%	8%	0%	0%	0%	0%	0%	0%	0%
Staffing At	40%	50%		0%	83%	0%	100%	0%	60%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	10%	50%	59%	0%		0%	50%	0%	20%	0%	100%	0%
POD Layout	10%	0%	0%	0%	0%		0%	0%	0%	0%	0%	0%
Flow	10%	0%	12%	0%	8%	0%		0%	40%	0%	0%	0%
Internal Layout	20%	0%	0%	0%	0%	0%	0%		0%	0%	0%	0%
Flow Within	30%	0%	18%	0%	8%	0%	100%	0%		100%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	20%		0%	0%
Drive Through	0%	0%	0%	0%	8%	0%	0%	0%	0%	0%		0%
Parking Plan	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

Figure 2.29: Professional Team 3 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		80%	43%	0%	20%	67%	70%	31%	50%	0%	33%	38%
Include	25%		24%	0%	0%	0%	10%	0%	17%	0%	0%	25%
Staffing At	56%	100%		0%	60%	0%	60%	23%	83%	0%	0%	38%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	6%	0%	14%	0%		0%	0%	8%	0%	0%	0%	0%
POD Layout	13%	0%	0%	0%	0%		0%	0%	0%	0%	67%	25%
Flow	44%	20%	29%	0%	0%	0%		38%	67%	33%	0%	0%
Internal Layout	25%	0%	14%	0%	20%	0%	50%		50%	100%	0%	13%
Flow Within	19%	20%	24%	0%	0%	0%	40%	23%		33%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	10%	23%	17%		0%	0%
Drive Through	6%	0%	0%	0%	0%	67%	0%	0%	0%	0%		13%
Parking Plan	19%	40%	14%	0%	0%	67%	0%	8%	0%	0%	33%	

Figure 2.30: Professional Team 4 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		0%	33%	0%	25%	100%	43%	24%	25%	0%	0%	40%
Include	0%		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Staffing At	25%	0%		0%	100%	25%	0%	18%	0%	0%	0%	20%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	5%	0%	27%	0%		0%	0%	0%	0%	0%	0%	0%
POD Layout	20%	0%	7%	0%	0%		0%	0%	0%	0%	0%	20%
Flow	15%	0%	0%	0%	0%	0%		18%	50%	0%	0%	0%
Internal Layout	20%	0%	20%	0%	0%	0%	43%		75%	0%	0%	0%
Flow Within	10%	0%	0%	0%	0%	0%	57%	35%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%
Parking Plan	10%	0%	7%	0%	0%	25%	0%	0%	0%	0%	0%	

Figure 2.31: Professional Team 5 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		100%	0%	0%	0%	20%	100%	0%	100%	0%	0%	100%
Include	17%		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Staffing At	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	0%	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%
POD Layout	17%	0%	0%	0%	0%		0%	0%	0%	0%	0%	0%
Flow	33%	0%	0%	0%	0%	0%		0%	0%	0%	0%	0%
Internal Layout	0%	0%	0%	0%	0%	0%	0%		0%	0%	0%	0%
Flow Within	17%	0%	0%	0%	0%	0%	0%	0%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%
Parking Plan	17%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

Figure 2.32: Student Team 1 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		0%	20%	0%	17%	75%	57%	11%	0%	0%	0%	100%
Include	0%		0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Staffing At	8%	0%		0%	17%	0%	0%	11%	0%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	8%	0%	20%	0%		25%	0%	0%	0%	0%	0%	0%
POD Layout	25%	0%	0%	0%	17%		0%	0%	0%	0%	0%	0%
Flow	33%	0%	0%	0%	0%	0%		44%	50%	0%	0%	0%
Internal Layout	8%	0%	20%	0%	0%	0%	57%		100%	0%	0%	0%
Flow Within	0%	0%	0%	0%	0%	0%	29%	44%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%
Parking Plan	17%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

Figure 2.33: Student Team 2 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		67%	0%	0%	0%	60%	50%	33%	67%	0%	0%	100%
Include	12%		0%	0%	0%	20%	0%	0%	0%	0%	0%	0%
Staffing At	0%	0%		0%	50%	0%	0%	0%	0%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	0%	0%	17%	0%		0%	0%	0%	0%	0%	0%	0%
POD Layout	18%	33%	0%	0%	0%		0%	0%	0%	0%	0%	0%
Flow	6%	0%	0%	0%	0%	0%		11%	0%	0%	0%	0%
Internal Layout	18%	0%	0%	0%	0%	0%	50%		33%	0%	0%	0%
Flow Within	12%	0%	0%	0%	0%	0%	0%	11%		0%	0%	50%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%
Parking Plan	24%	0%	0%	0%	0%	0%	0%	0%	67%	0%	0%	

Figure 2.34: Student Team 3 Category Concurrency Matrix

	Location	Include	Staffing At	Additional staffing	Calculating Staff Needs	POD Layout	Flow	Internal Layout	Flow Within	Visual Aids	Drive Through	Parking Plan
Location		0%	50%	0%	18%	20%	0%	80%	0%	0%	0%	0%
Include	0%		0%	0%	0%	20%	0%	0%	0%	0%	0%	0%
Staffing At	30%	0%		0%	18%	0%	0%	20%	0%	0%	0%	0%
Additional Staffing	0%	0%	0%		0%	0%	0%	0%	0%	0%	0%	0%
Calculating Staff Needs	20%	0%	33%	0%		20%	0%	20%	0%	0%	0%	0%
POD Layout	10%	100%	0%	0%	9%		0%	0%	0%	0%	0%	0%
Flow	0%	0%	0%	0%	0%	0%		0%	0%	0%	0%	0%
Internal Layout	40%	0%	17%	0%	9%	0%	0%		0%	0%	0%	0%
Flow Within	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%	0%
Visual Aids	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%	0%
Drive Through	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		0%
Parking Plan	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

Figure 2.35: Student Team 4 Category Concurrency Matrix

2.4 Clustering and Subproblem Analysis Results

2.4.1 Overview

This section contains the results from the clustering algorithms and subproblem analysis. This includes the results from the four algorithms used (Ward's, spectral, Markov, association rules), the numerical results from our comparison methods, and a brief section on what the variable abstraction exercise indicated. As with the previous results section, the professional teams will be displayed first, followed by the student teams. Team names will also be abbreviated in the same manner.

We evaluated the clustering results in several ways. Recall that our goal is to develop a method for identifying subproblems based on data describing which variables were discussed together, so we were interested in determining which clustering algorithm makes clusters that (1) best differentiate groups of variables discussed concurrently from those discussed separately and (2) are aligned with the apparent topics of conversation in the video-recordings. To assess (1), we used two strategies. First, we used several numerical measures of clustering success. Second, we qualitatively compared the subproblems resulting from each method to one another. To assess (2), we examined several segments of video and investigated the differences between the clustering results and the apparent topic of conversation in the video. These results are described next.

These results focus on the individual teams and draws attention to how the algorithms performed with each unique coded discussion. A detailed comparison and analysis of the teams, algorithms, and other observations are discussed later in more detail in the Discussion section, Section 2.5.

2.4.2 Baseline Clustering Results

In order to create a baseline understanding of how the clustering algorithms would group variables in certain situations, we created a data set with clearly related variables and ran the clustering algorithms on this data set. We also introduced 3 varying levels of noise into the baseline data set by giving each segment for each variable a 1%, 5%, and 10% chance of becoming coded if it was not coded in the baseline, or becoming not coded if it was coded in the baseline. The variables in this test were named SP#.#, where the first number denotes the subproblem and the second number denotes the variable in that subproblem. All of the clustering algorithms were run with the same input parameters that the used with the professional teams.

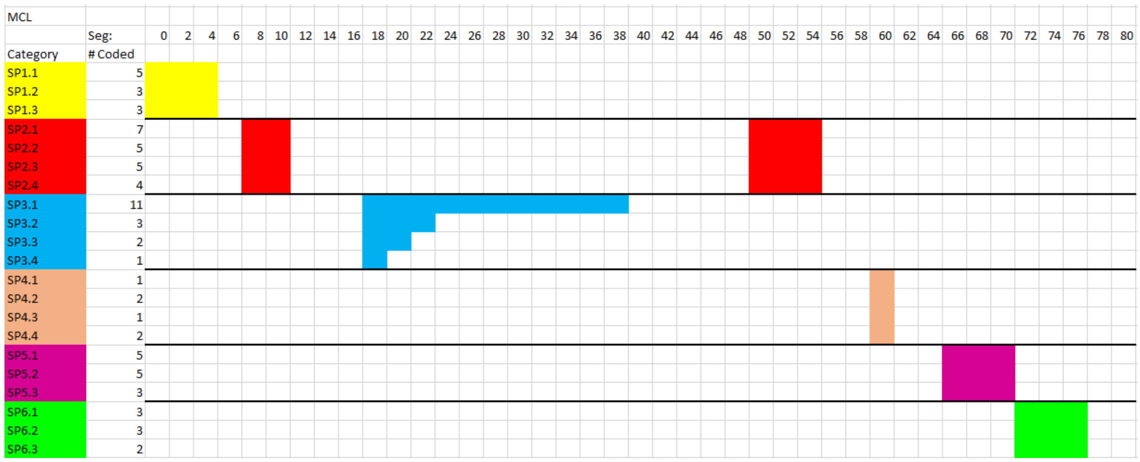


Figure 2.36: Baseline Markov Clustering Timeline

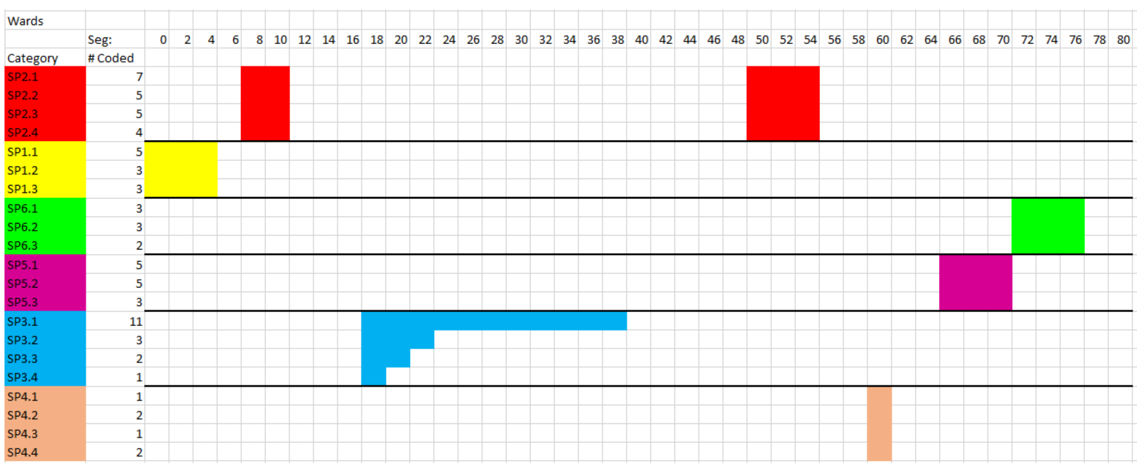


Figure 2.37: Baseline Wards Clustering Timeline

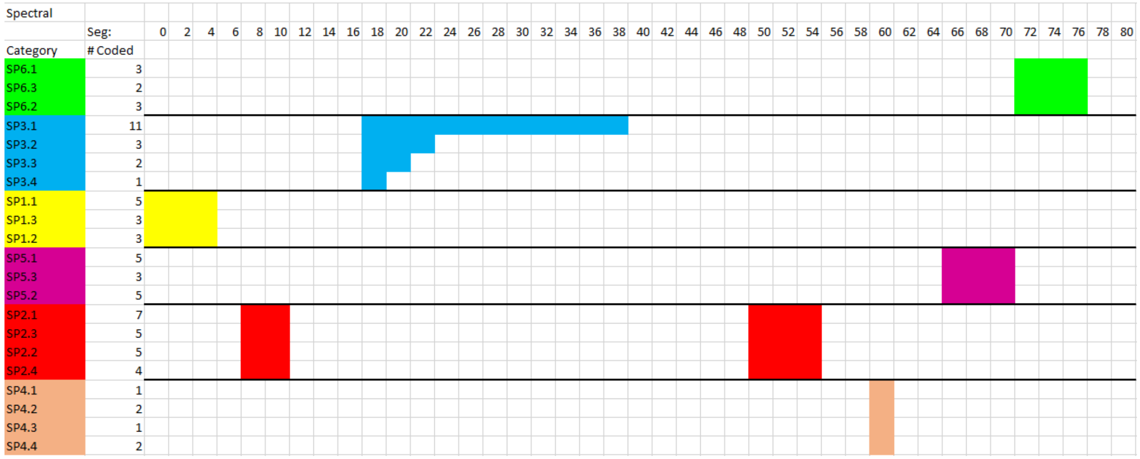


Figure 2.38: Baseline Spectral Clustering Timeline

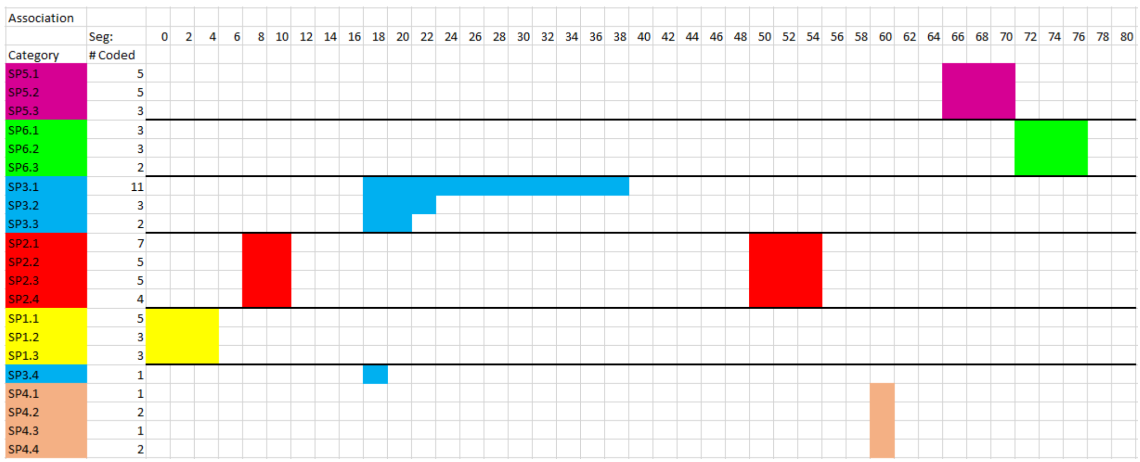


Figure 2.39: Baseline Association Rules Clustering Timeline

The results for the baseline data can be seen in Figures 2.36, 2.37, 2.38, 2.39. The baseline results show that the clustering algorithms work well when dealing with clear cut groups of variables. All of the methods except the association rules correctly clustered the original six subproblems. This suggests that the algorithms will perform well when variables are coded in tight, non-overlapping groups. It also indicates that the algorithms can be used to accurately identify subproblems. The only exception is the association rules method results. This algorithm did not cluster SP3.4 as well as the SP4 variables, all of which were coded only during one segment but had 100% concurrency. This suggests that the association rules will usually not cluster single coded variables.

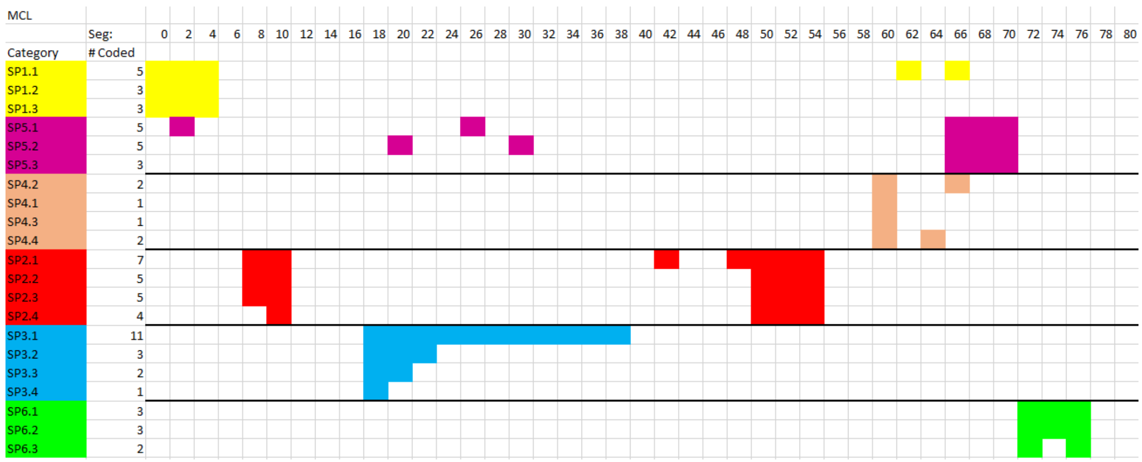


Figure 2.40: 1% Noise Markov Clustering Timeline (6 clusters)

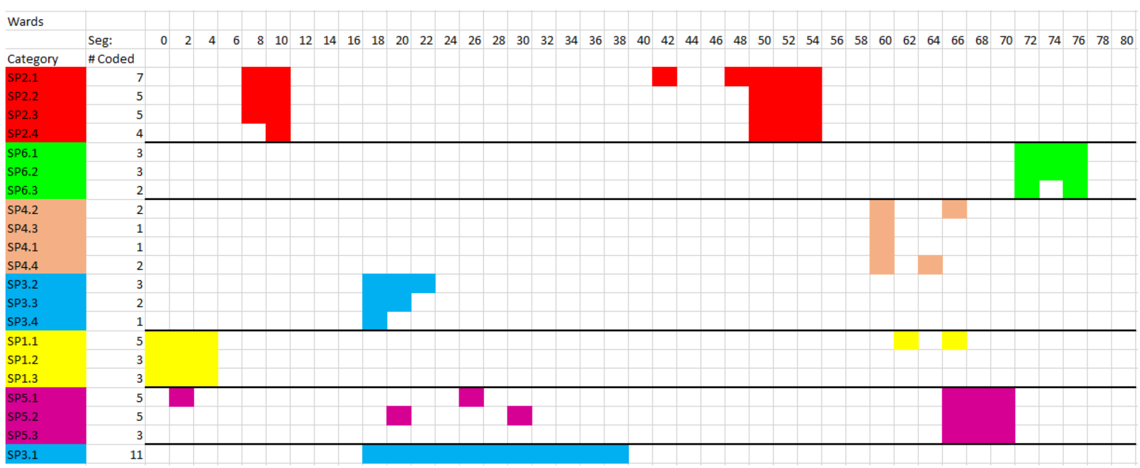


Figure 2.41: 1% Noise Wards Clustering Timeline (5 clusters, 1 unclustered variable)

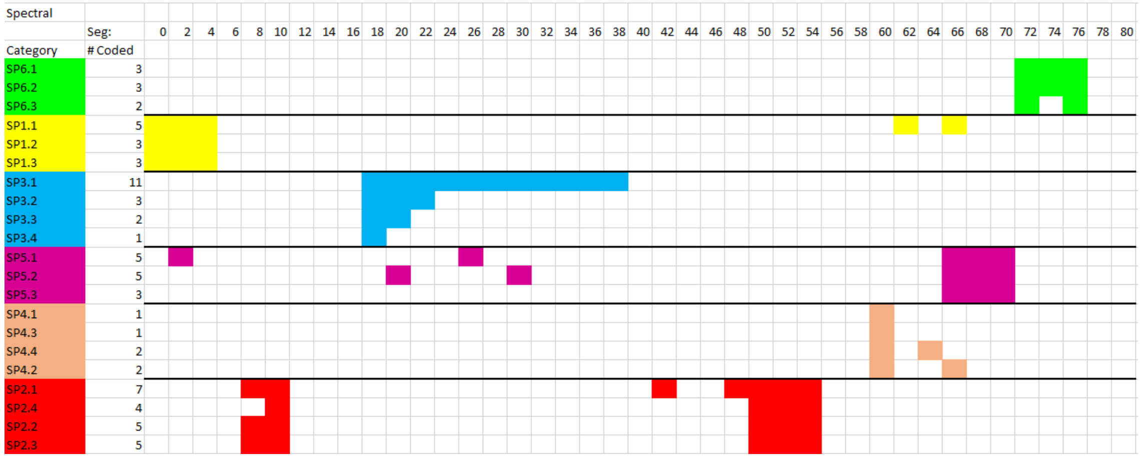


Figure 2.42: 1% Noise Spectral Clustering Timeline (6 clusters)

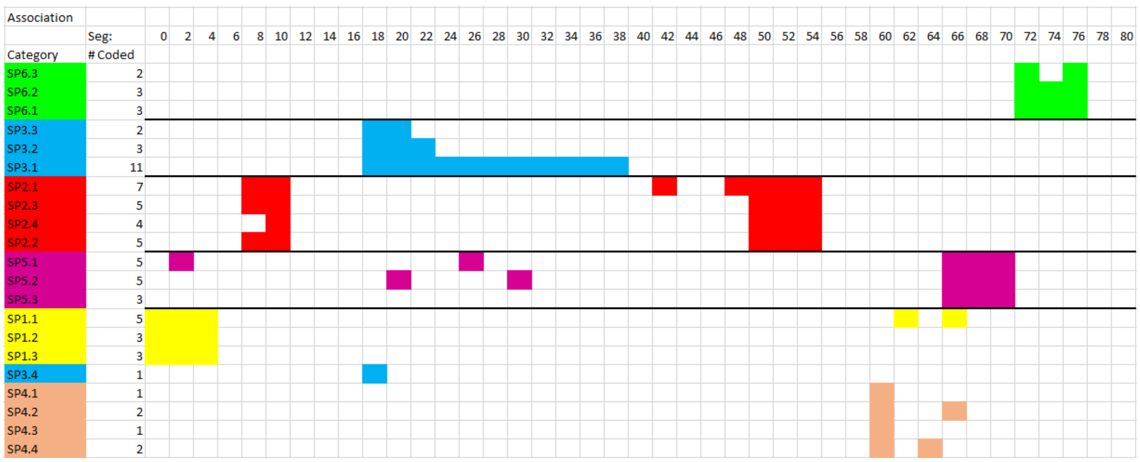


Figure 2.43: 1% Noise Association Rules Clustering Timeline (4 clusters, 8 unclustered variables)

The results for the 1% Noise test can be seen in Figures 2.40, 2.41, 2.42, 2.43. Here we can see how the algorithms cluster variables that slightly overlap. The Markov method clustered the variables from SP1 and SP5 together when they overlapped in two segments. The other algorithms did not cluster these two groups together. The Wards method results show that the SP3 and SP4 were clustered together despite not having any overlapping segments. This is an interesting result from Ward's method and shows that the clusters clearly won't always indicate sensible subproblems.

Figure 2.44: 5% Noise Markov Clustering Timeline (3 clusters)

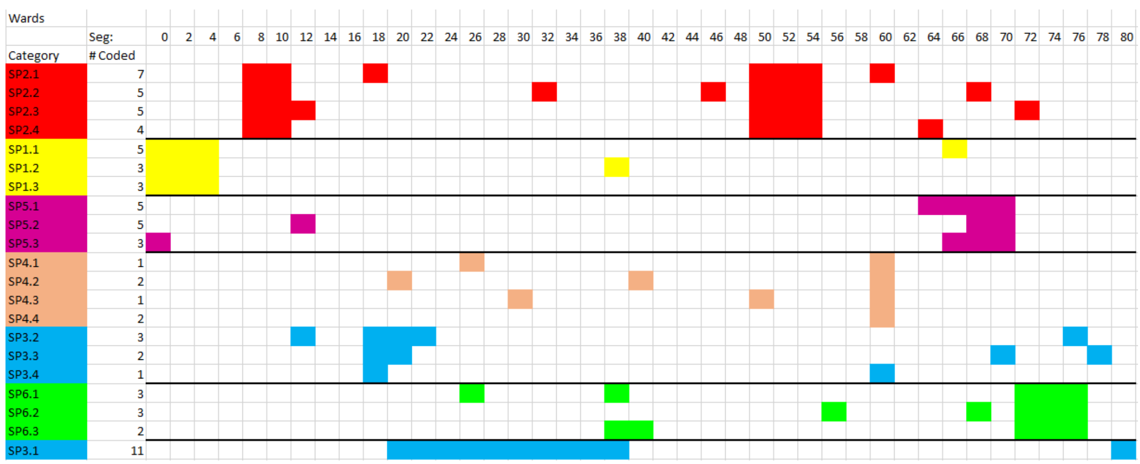


Figure 2.45: 5% Noise Wards Clustering Timeline (5 clusters, 1 unclustered variable)

Figure 2.46: 5% Noise Spectral Clustering Timeline (6 clusters)

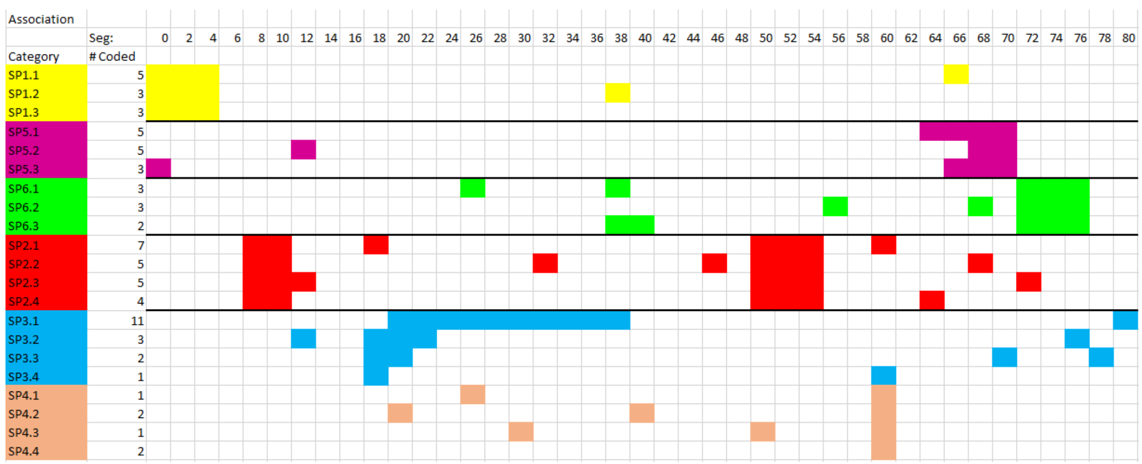


Figure 2.47: 5% Noise Association Rules Clustering Timeline (4 clusters, 8 unclustered variables)

The results for the 5% Noise test can be seen in Figures 2.44, 2.45, 2.46, 2.47. This data set is the most similar to that of the professional teams. Here we can see that despite the number of additional segments, the algorithms still correctly clustered one or two subproblems each. This, along with the previous tests, shows that while one algorithm may fail to cluster the variables correctly, the algorithms as a whole can provide a thorough indication as to what subproblems may exist. The most difficult part of interpreting these results would be determining which clusters are actually subproblems and which clusters are red herrings.

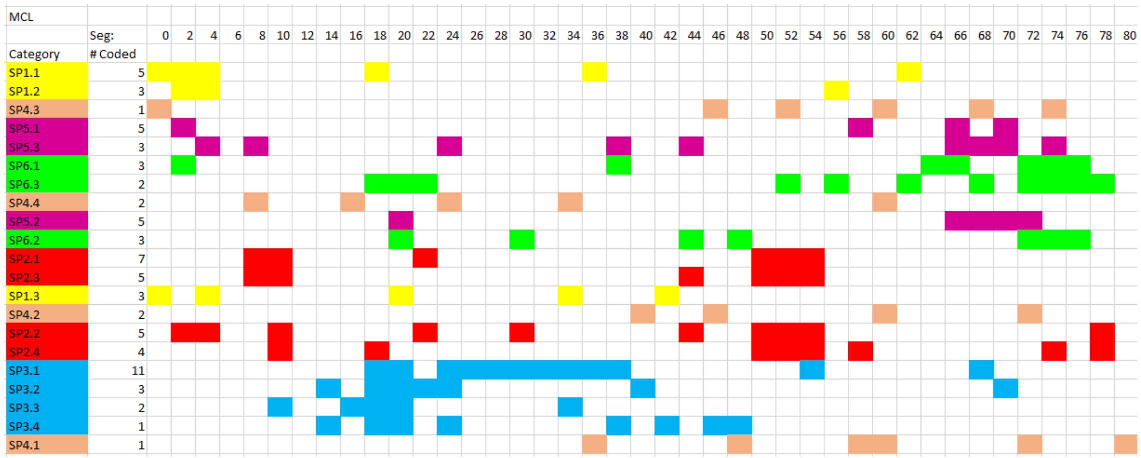


Figure 2.48: 10% Noise Markov Clustering Timeline (1 cluster)

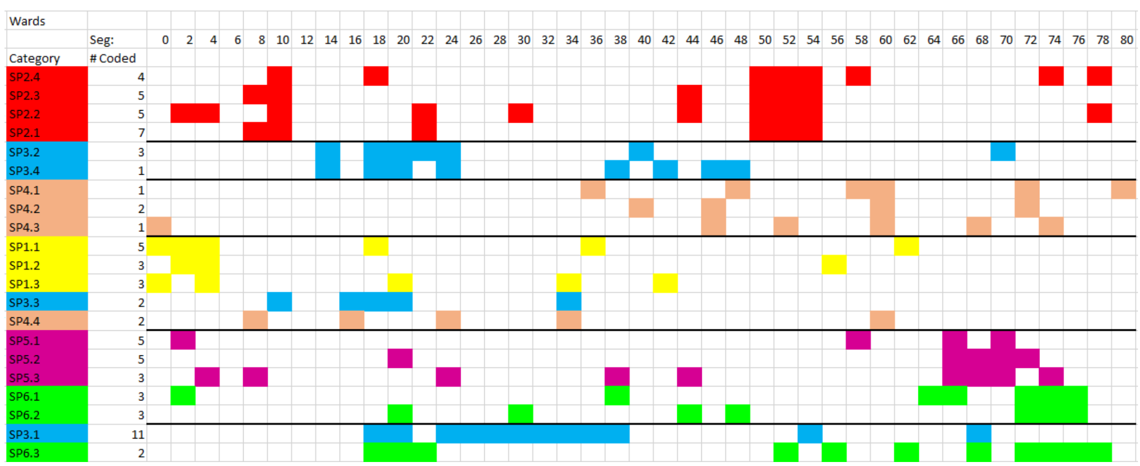


Figure 2.49: 10% Noise Wards Clustering Timeline (5 clusters, 2 unclustered variables)

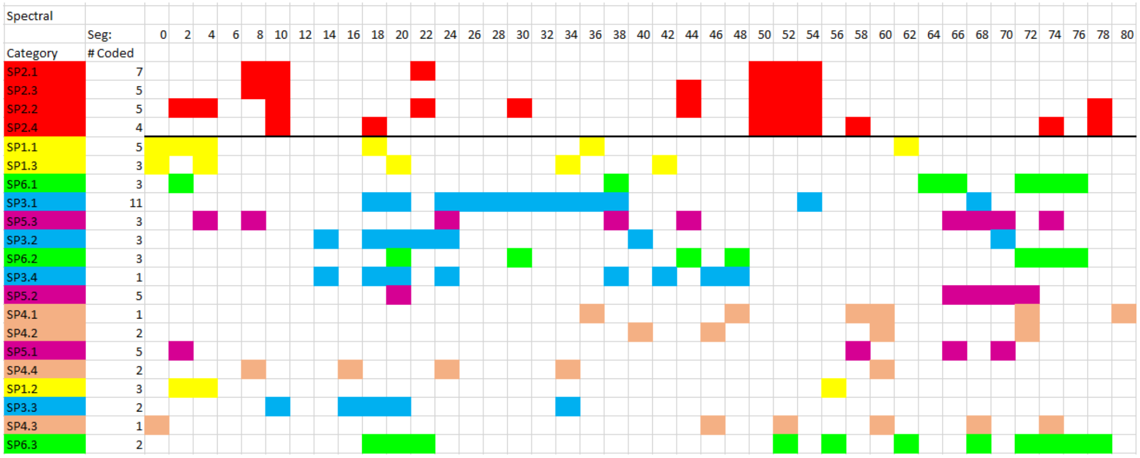


Figure 2.50: 10% Noise Spectral Clustering Timeline (1 cluster, 17 unclustered variables)

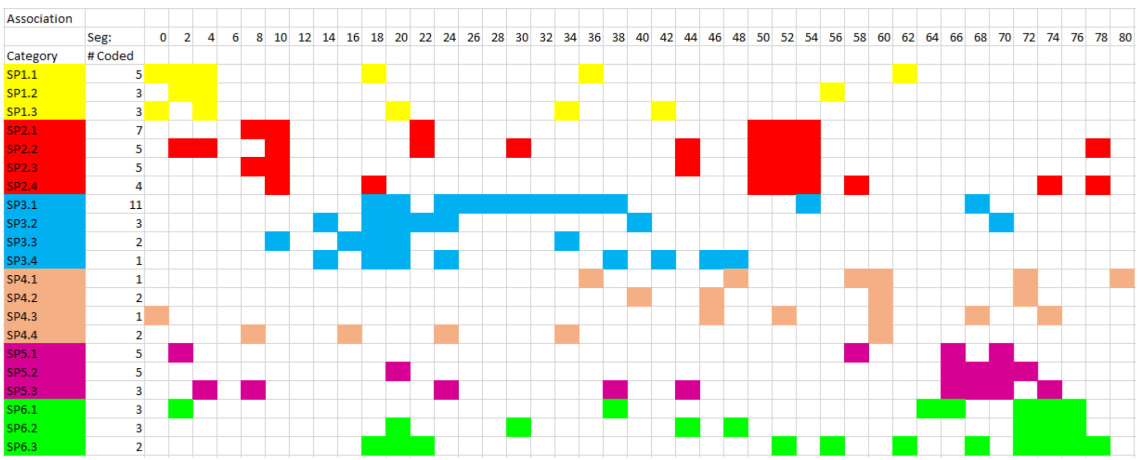


Figure 2.51: 10% Noise Association Rules Clustering Timeline (1 cluster)

The results for the 10% Noise test can be seen in Figures 2.48, 2.49, 2.50, 2.51. As seen in the 10% Noise results, the clustering algorithms perform poorly when there is a lot of large sporadic variables. All of the clustering algorithms created only one or two clusters out of this data set. Although it may be possible that a team with this type of discussion did not create any subproblems, these results indicate that running the clustering algorithms does not reveal much if any new information about the decomposition strategy used. We would expect the algorithms to perform this way when dealing with any large or sporadically coded variables.

2.4.3 Clustering Results as Timelines

The timelines presented here are associated with the clusters that the four different clustering algorithms created for each team. Each timeline shows the time segments as the column header and the variable names as the row headers. Any omitted starting time segments (columns) were hidden due to a lack of discussion. No segments were omitted in the middle of the timelines.

The timelines are color formatted so each of the Markov Clustering Algorithm's subproblems is a solid color. Colors do not indicate any subproblem attributes such as confidence or theme. The other timelines maintain this simple color formatting on the variable level and instead show the subproblem grouping with bold black lines. This allows us to easily compare the movement of variables between algorithms and suggested subproblems. Generally, the subproblem closest to the bottom has the loosest association between variables.

For all of the teams below, the Markov results are shown first, followed by the Ward's results, Spectral results, and finally Association results.

2.4.3.1 Professional Team 1

The clustering results for Professional Team 1 can be seen in Figures 2.52, 2.53, 2.54, and 2.55. P1 had a mix of single coded variables and multiple coded variables. Most of the algorithms managed to cluster the same multiple coded variables together, such as the green Location Point of Entry, Location Point of Exit, and POD Layout variables. However, the algorithms treated the single coded variables differently, ranging from clustering unrelated, single coded variables together (Figure 2.53) to not clustering them at all (Figure 2.55).

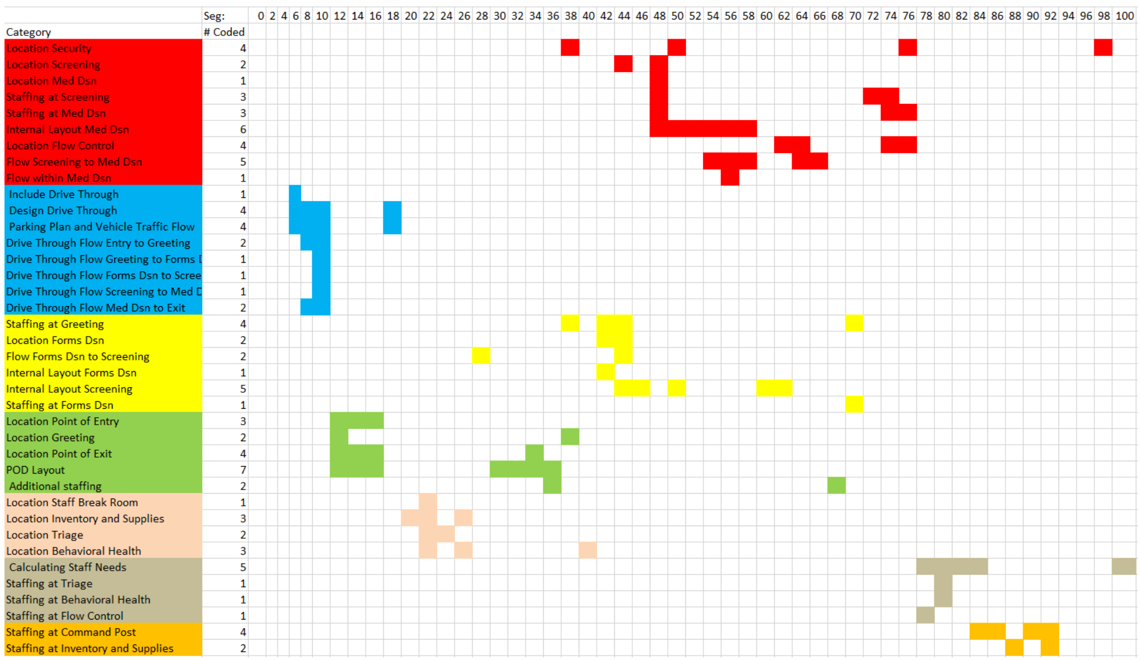


Figure 2.52: Professional Team 1 Markov Clustering Timeline

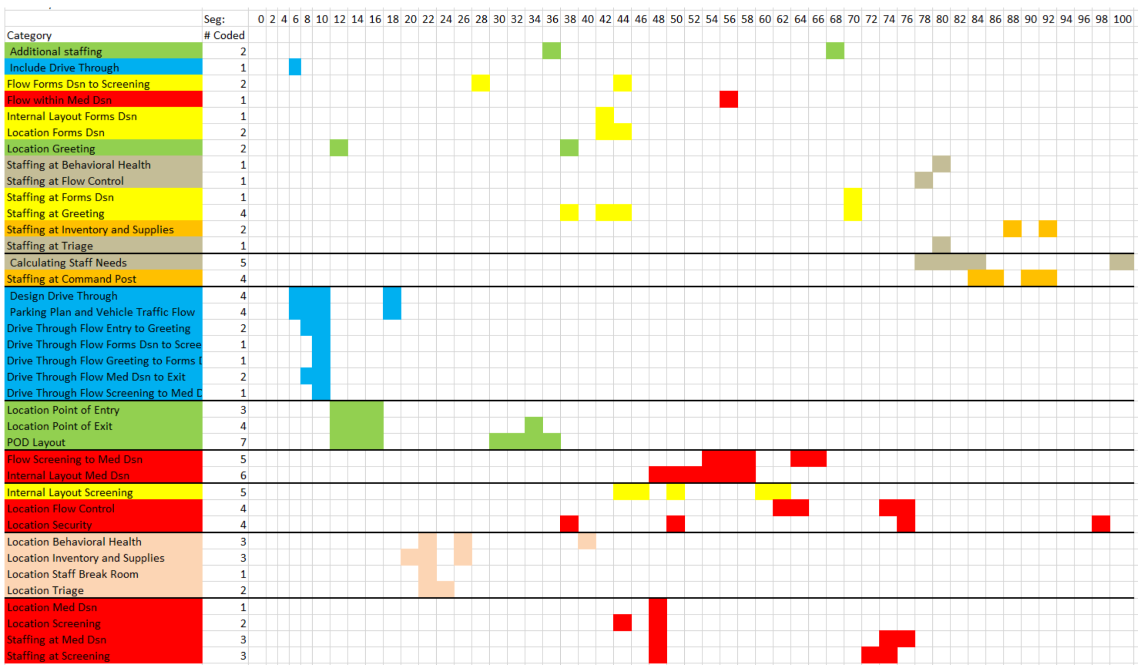


Figure 2.53: Professional Team 1 Wards Clustering Timeline

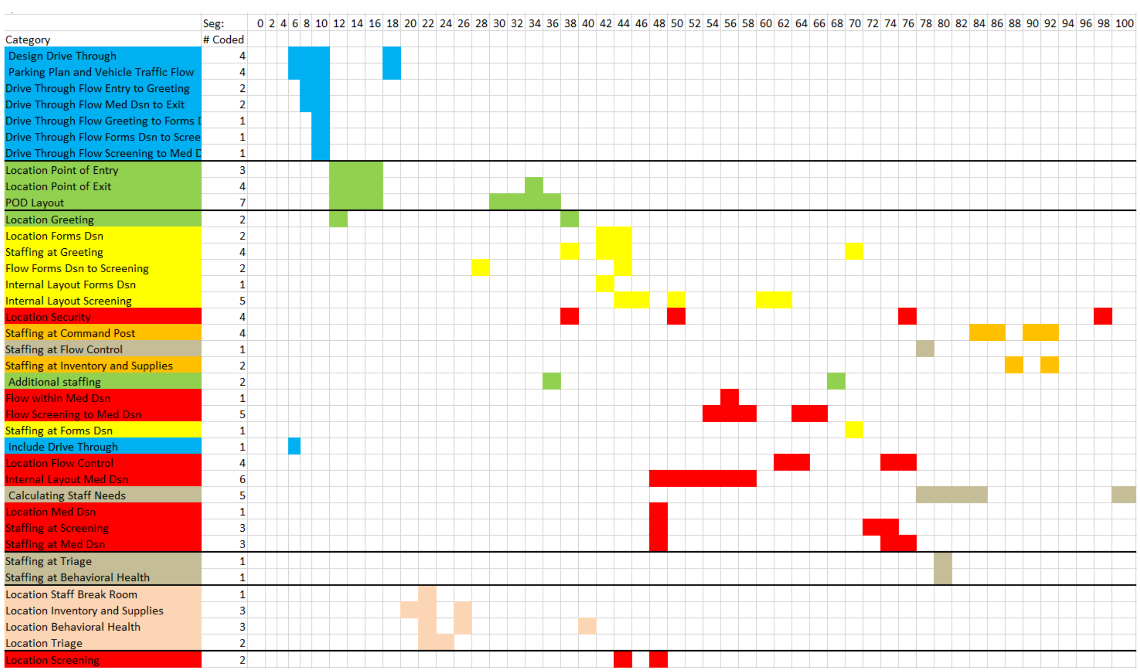


Figure 2.54: Professional Team 1 Spectral Clustering Timeline

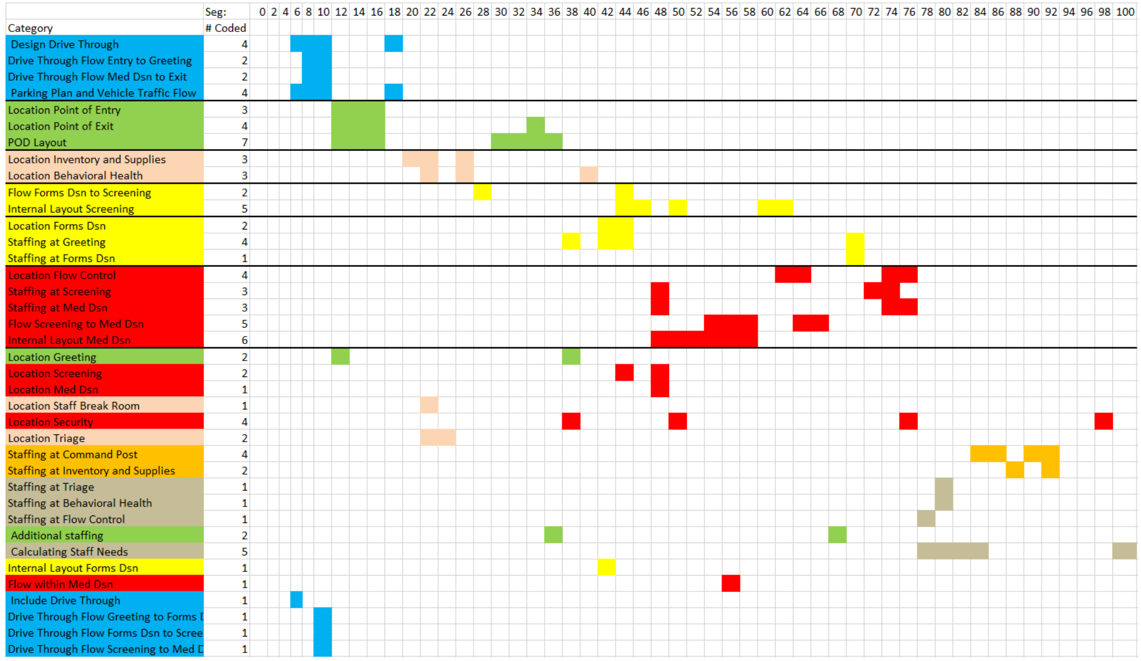


Figure 2.55: Professional Team 1 Association Rules Clustering Timeline

2.4.3.2 Professional Team 2

Professional Team 2's clustering results can be seen in Figures 2.56, 2.57, 2.58, and 2.59. P2 has multiple variables with a large number of coded segments. Each algorithm clustered these variables together, but the Markov and Spectral methods also clustered in many smaller variables. The Ward's and Association methods took the opposite approach and didn't cluster any smaller variables in with the large ones which lead to a cluster with many unrelated single coded variables. The blue and yellow subproblems stayed mostly together across the algorithms, except with the Association method where the cutoff threshold chose not to cluster many of the single coded variables from those subproblems.

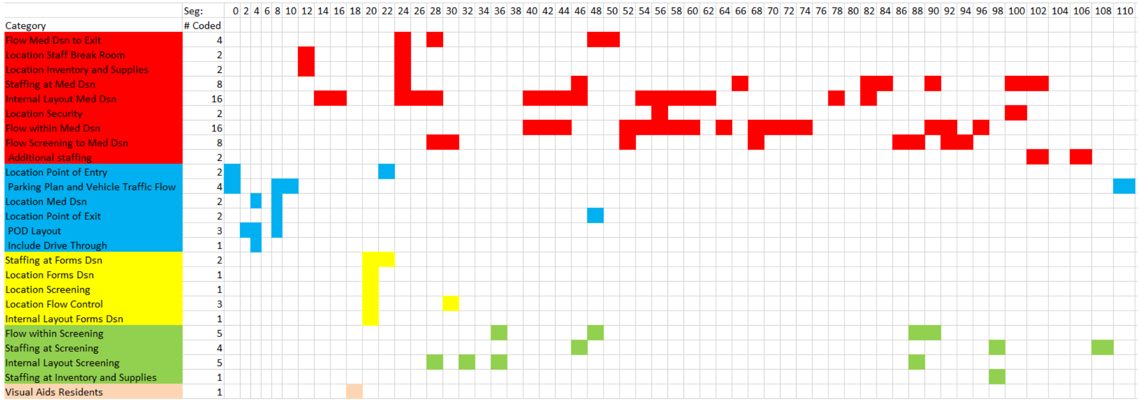


Figure 2.56: Professional Team 2 Markov Clustering Timeline

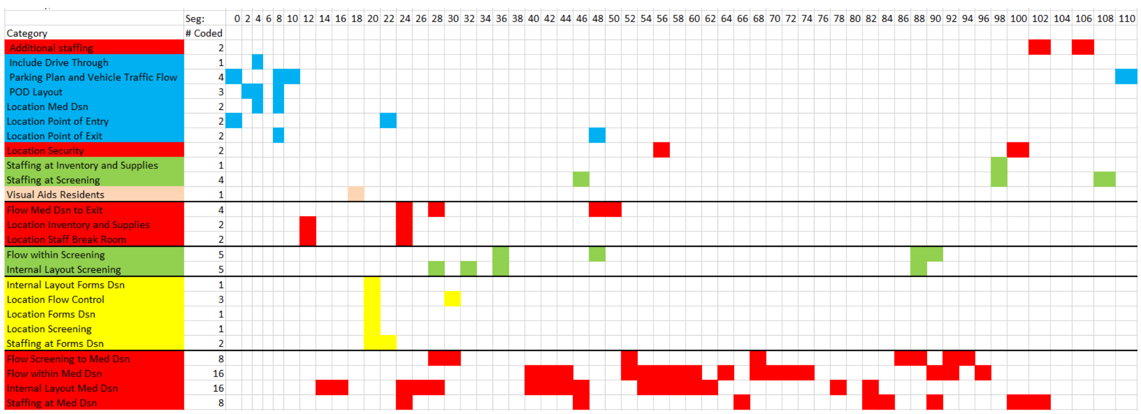


Figure 2.57: Professional Team 2 Wards Clustering Timeline

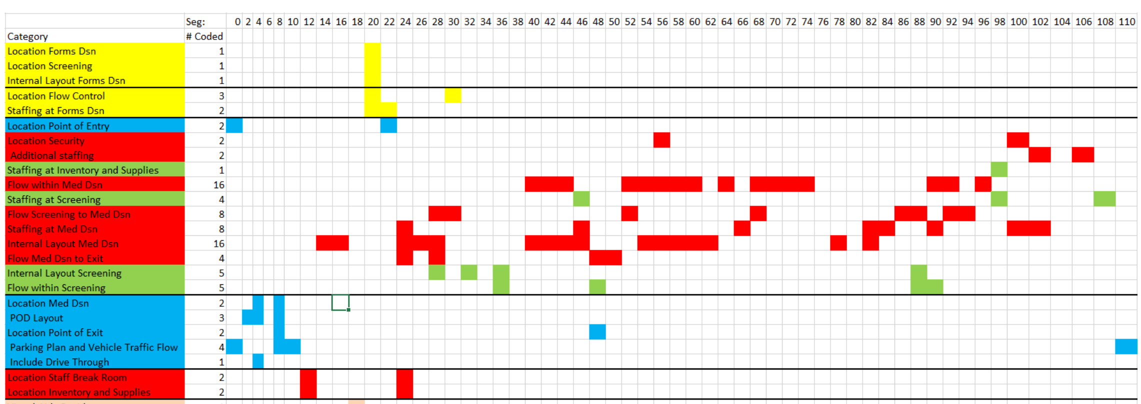


Figure 2.58: Professional Team 2 Spectral Clustering Timeline

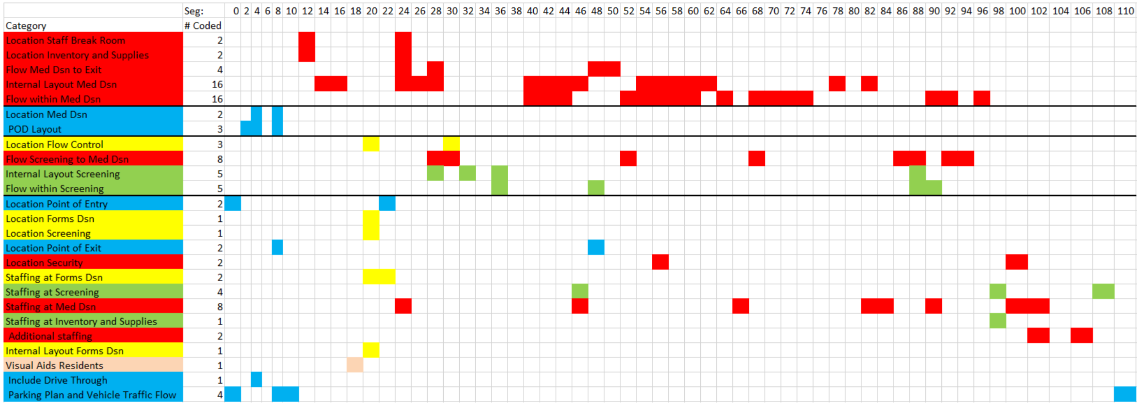


Figure 2.59: Professional Team 2 Association Rules Clustering Timeline

2.4.3.3 Professional Team 3

The results of the clustering algorithms for P3 can be seen in Figures 2.60, 2.61, 2.62, and 2.63. Generally, 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 sporadically throughout the design session, which caused some variables to be grouped together despite not being talked about together, as seen in the association rules timeline's third subproblem (Figure 2.63). The algorithms tended to come up with the same core subproblem groupings, and differentiated in granularity or size.

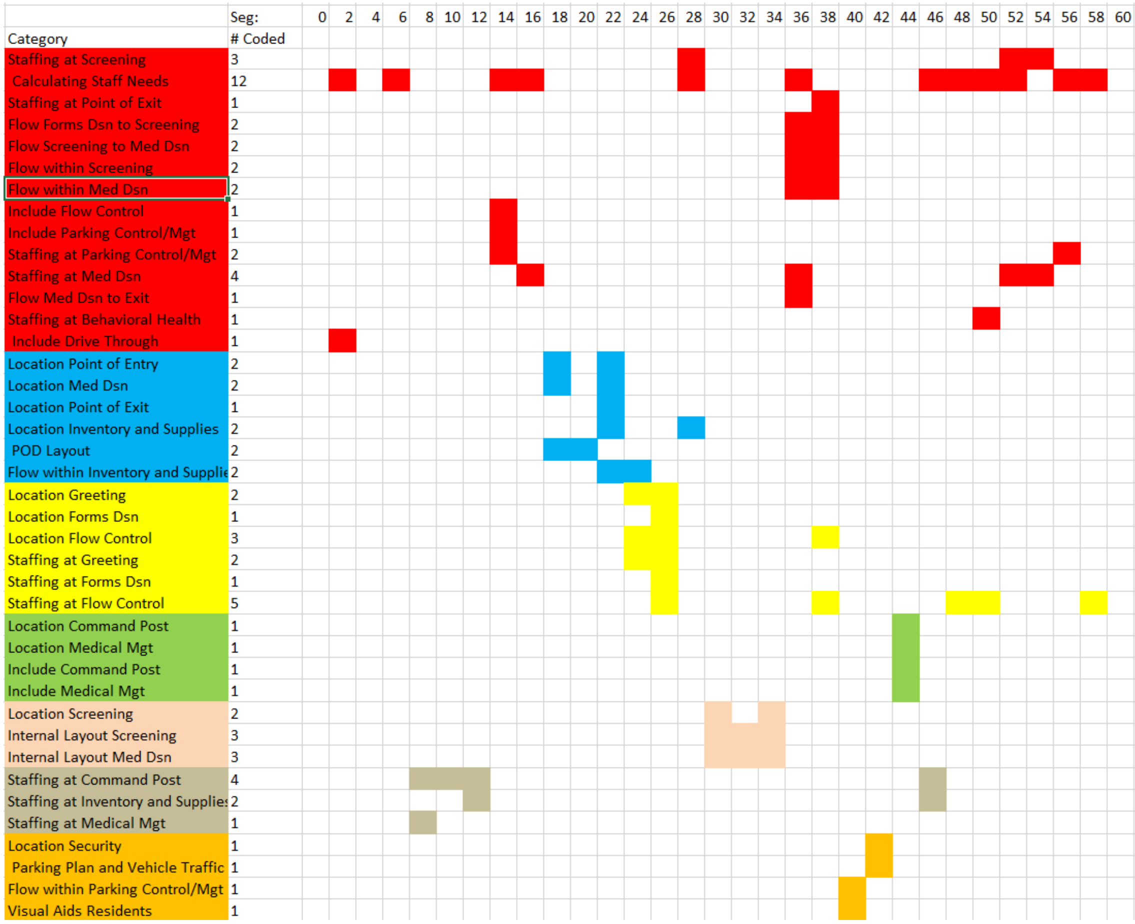


Figure 2.60: Professional Team 3 Markov Clustering Timeline

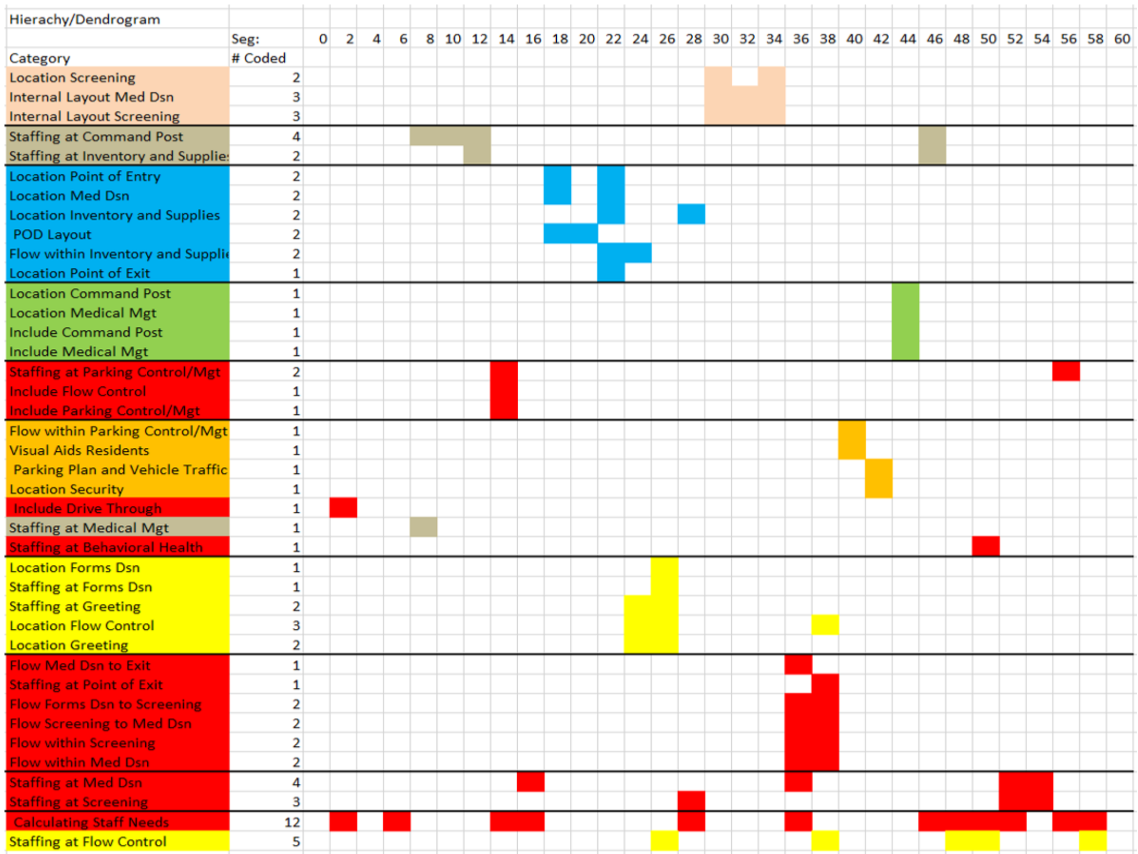


Figure 2.61: Professional Team 3 Wards Clustering Timeline

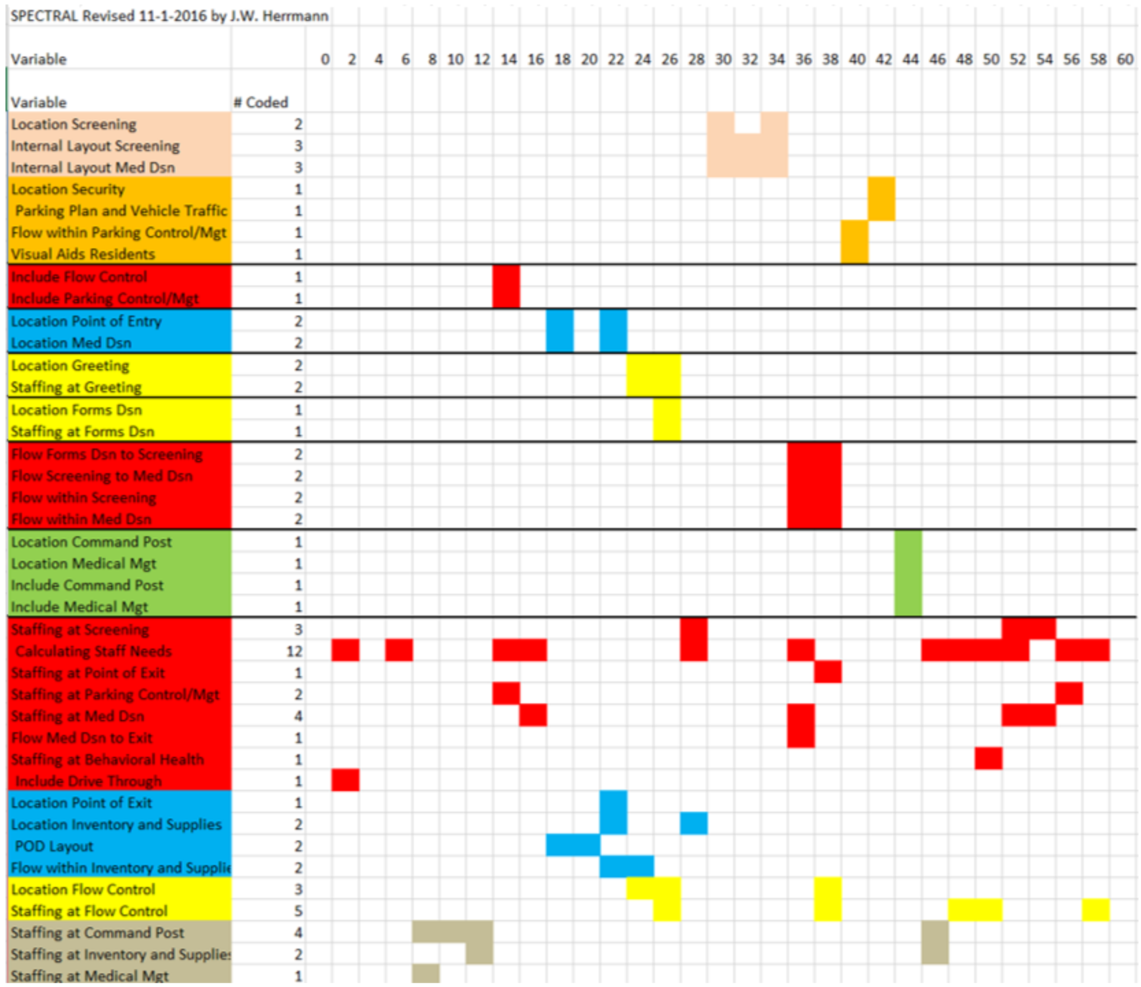


Figure 2.62: Professional Team 3 Spectral Clustering Timeline

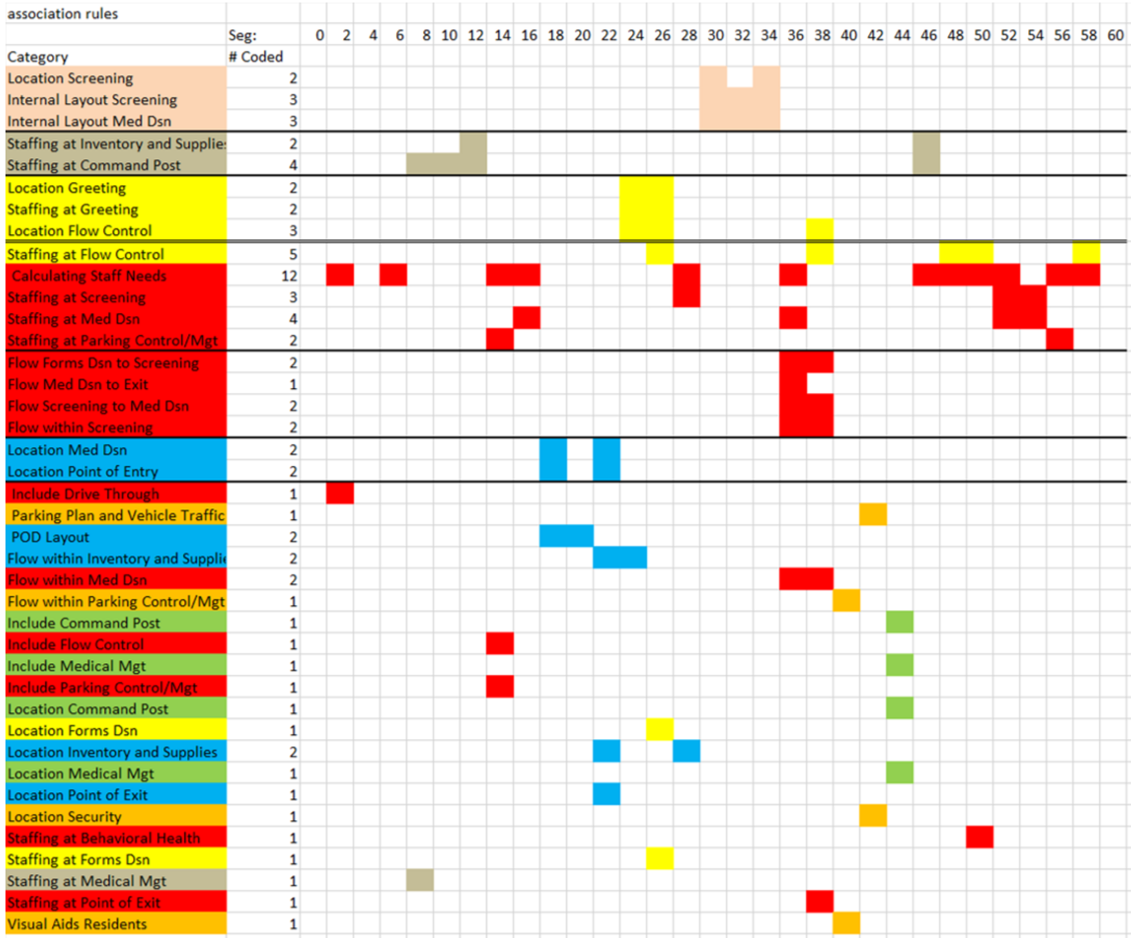


Figure 2.63: Professional Team 3 Association Rules Clustering Timeline

2.4.3.4 Professional Team 4

The results of the clustering algorithms for P4 can be seen in Figures 2.64, 2.65, 2.66, and 2.67. Generally, 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 sporadically throughout the design session, which caused some variables to be grouped together despite not being talked about together, as seen in the association rules timeline's third subproblem (Figure 2.67). The algorithms tended to come up with the same core subproblem groupings, and differentiated in granularity or size.

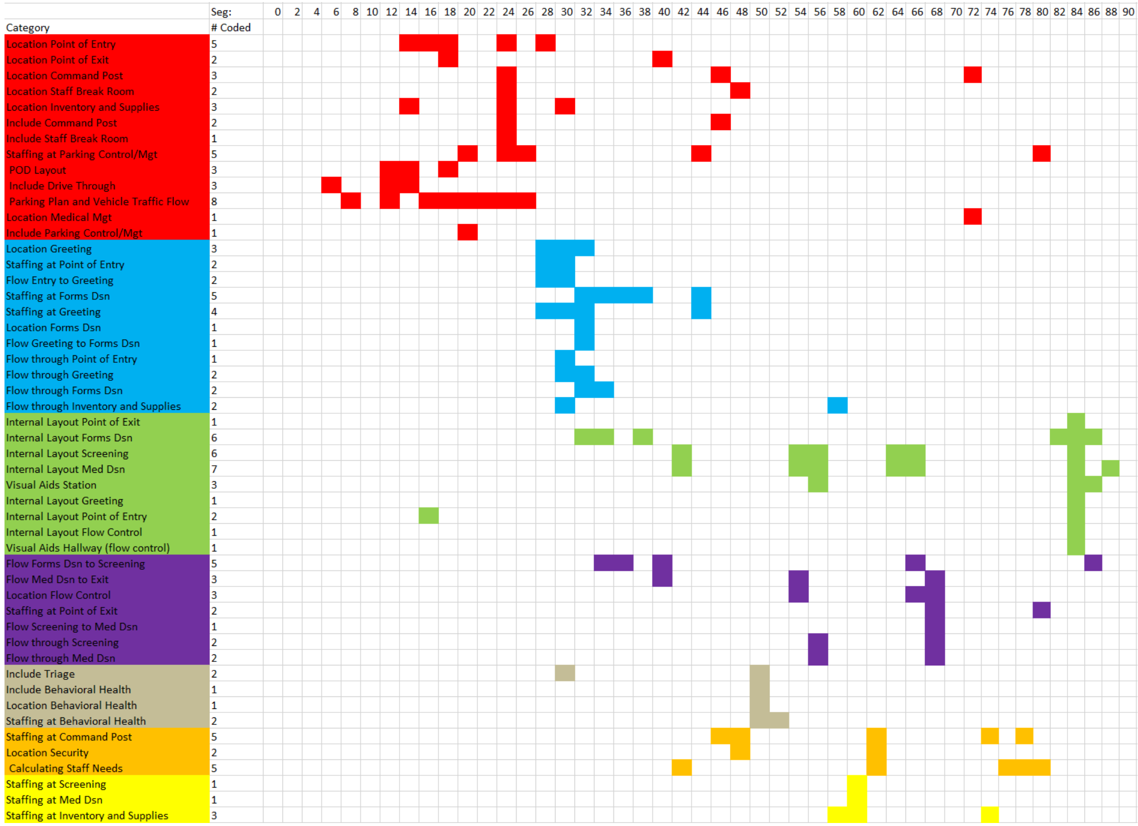


Figure 2.64: Professional Team 4 Markov Clustering Timeline

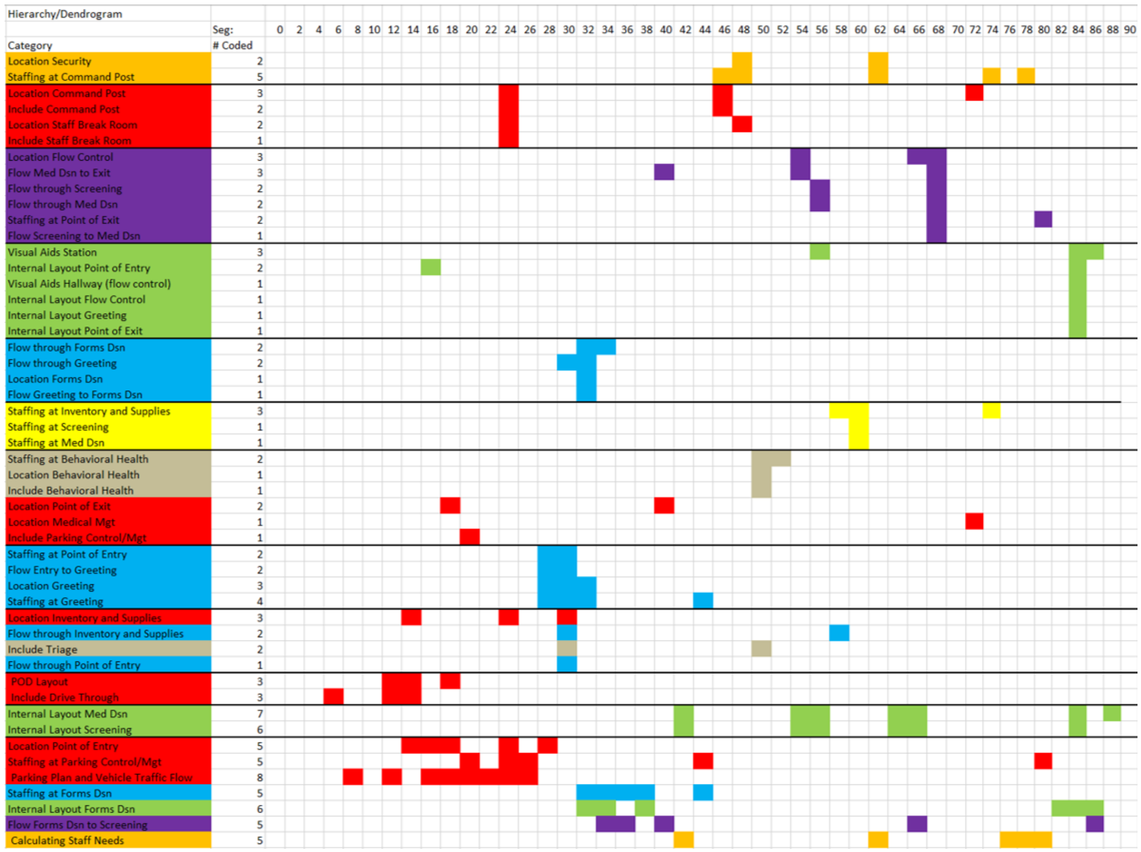


Figure 2.65: Professional Team 4 Wards Clustering Timeline

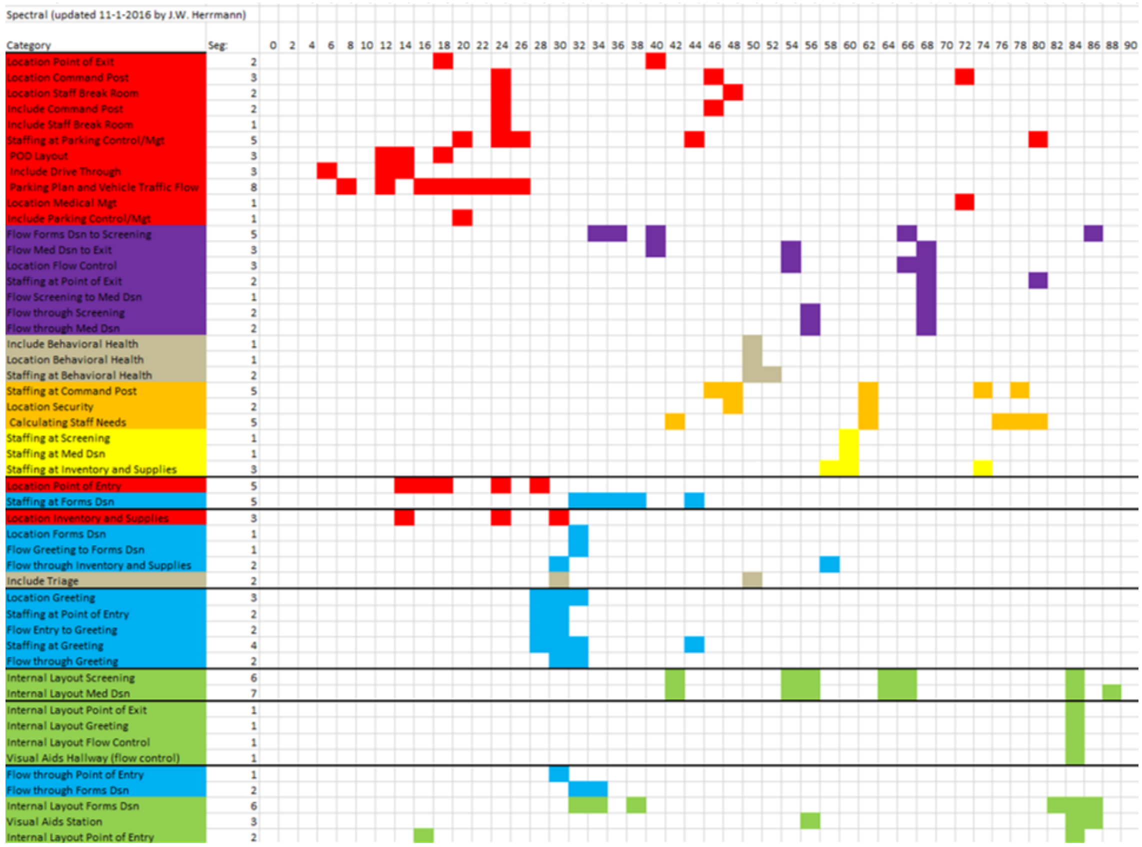


Figure 2.66: Professional Team 4 Spectral Clustering Timeline

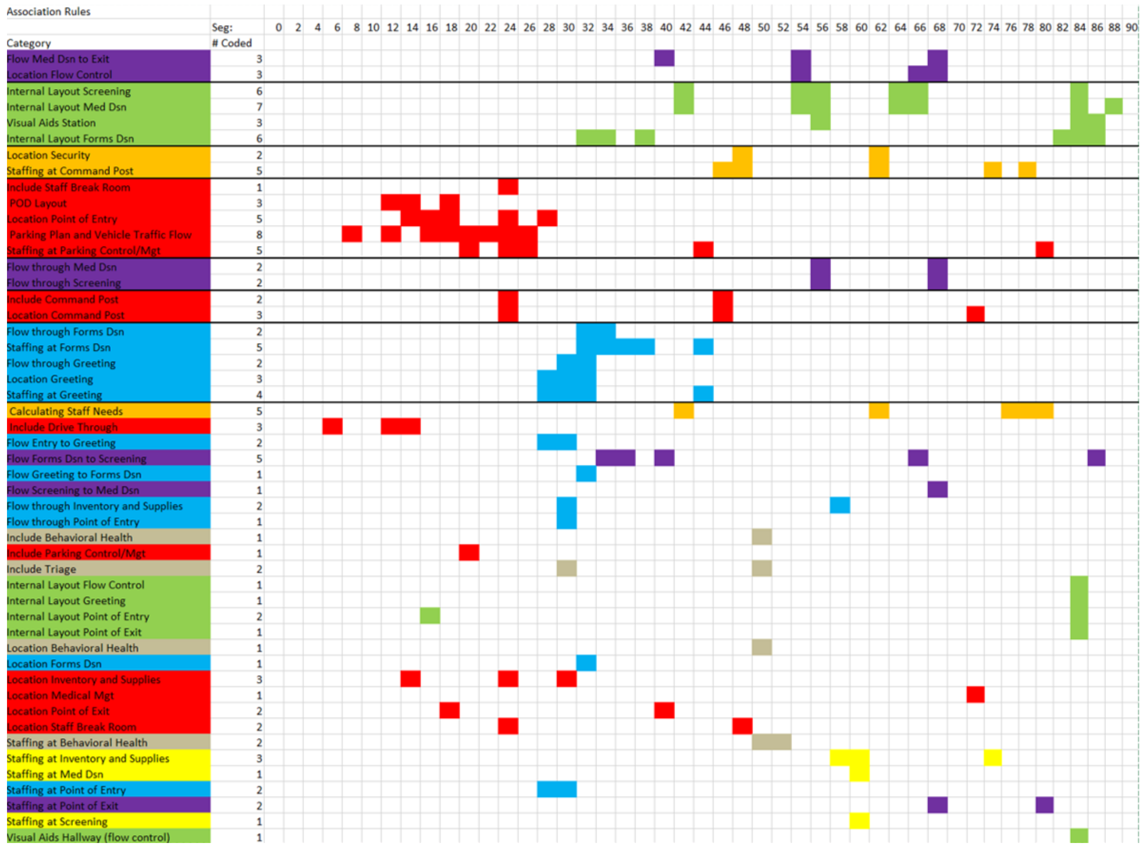


Figure 2.67: Professional Team 4 Association Rules Clustering Timeline

2.4.3.5 Professional Team 5

Professional Team 5's clustering results can be seen in Figures 2.68, 2.69, 2.70, and 2.71. This team has a large amount of single coded variables, which gave the clustering algorithms a difficult time. The red, blue, and yellow clusters in the Markov results appear to be clear subproblems, but the other algorithms did not cluster those variables together. Instead, the other algorithms chose to either cluster many of the variables together into one large cluster or cluster weakly related variables that share one or two coded segments. The green subproblem does stay clustered for all of the algorithms, except the Association method which clusters in more variables.

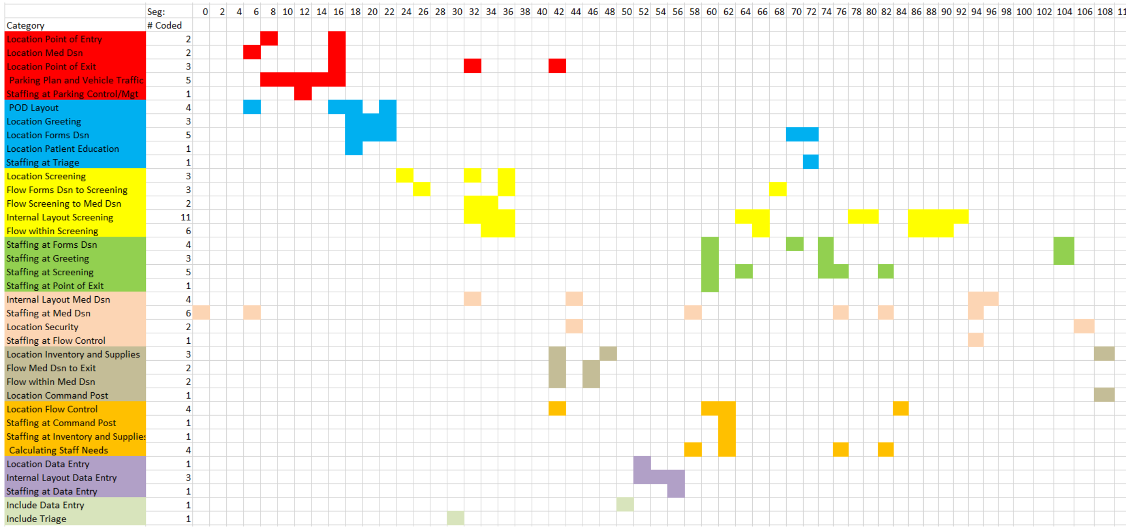


Figure 2.68: Professional Team 5 Markov Clustering Timeline

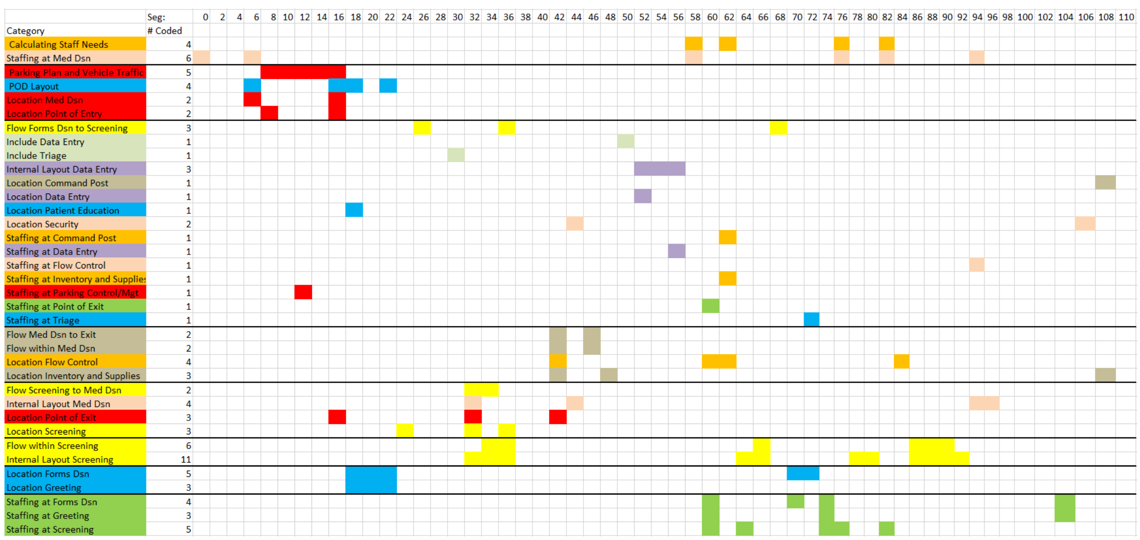


Figure 2.69: Professional Team 5 Wards Clustering Timeline

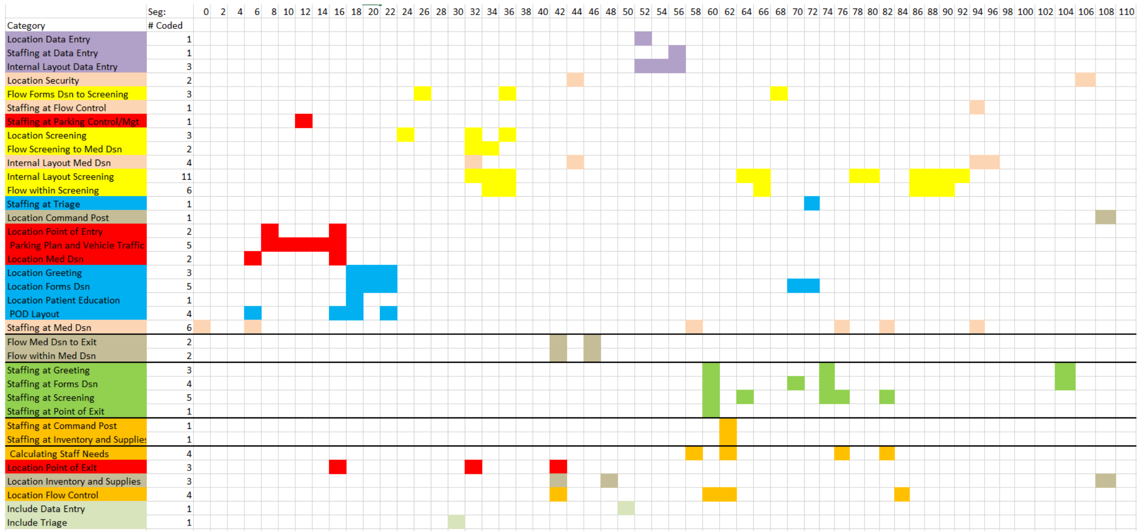


Figure 2.70: Professional Team 5 Spectral Clustering Timeline

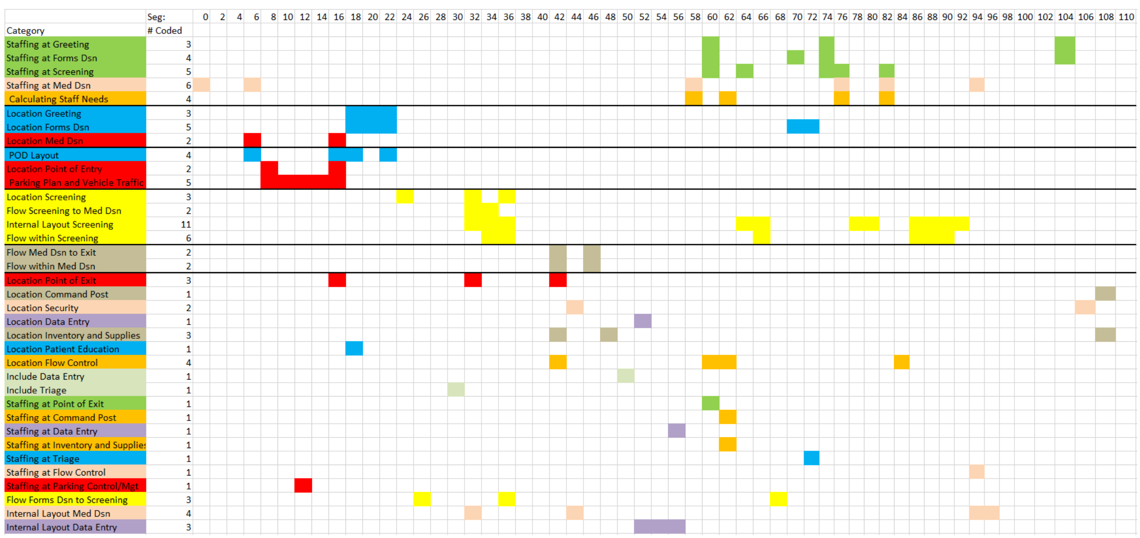


Figure 2.71: Professional Team 5 Association Rules Clustering Timeline

2.4.3.6 Student Team 1

The clustering results fro Student Team 1 can be seen in Figures 2.72, 2.73, 2.74, and 2.75. S1 has very few segments where multiple variables were coded, but the clustering results still differ greatly. The Association method only creates one cluster and chooses to not use two variables. The other methods have multiple clusters and tend to keep the blue and yellow clusters the same. Although some of the red variables are generally clustered together, the POD Layout variable is weakly related to other red variables and gets clustered differently by each algorithm.

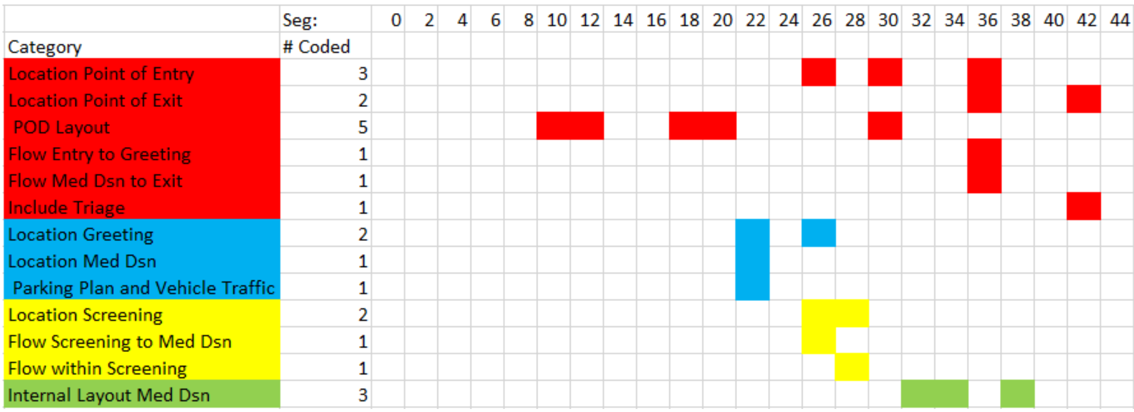


Figure 2.72: Student Team 1 Markov Clustering Timeline

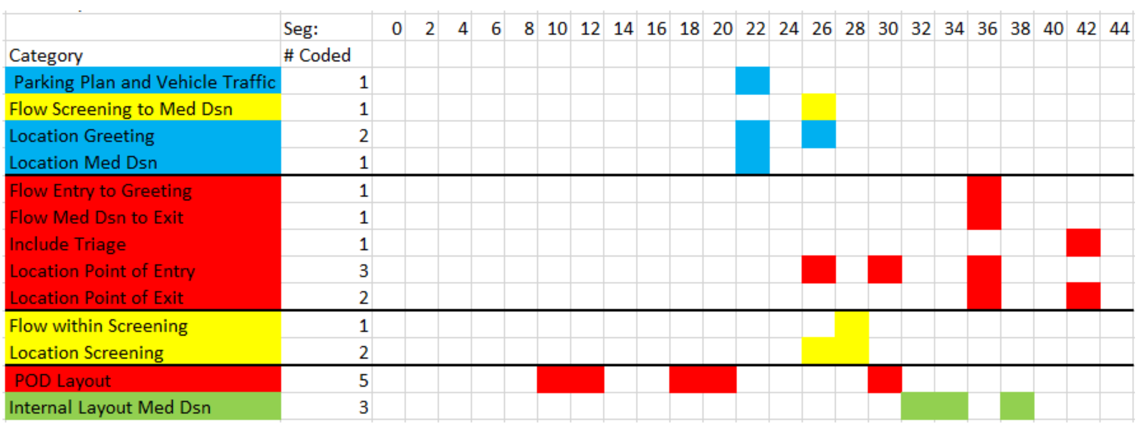


Figure 2.73: Student Team 1 Wards Clustering Timeline

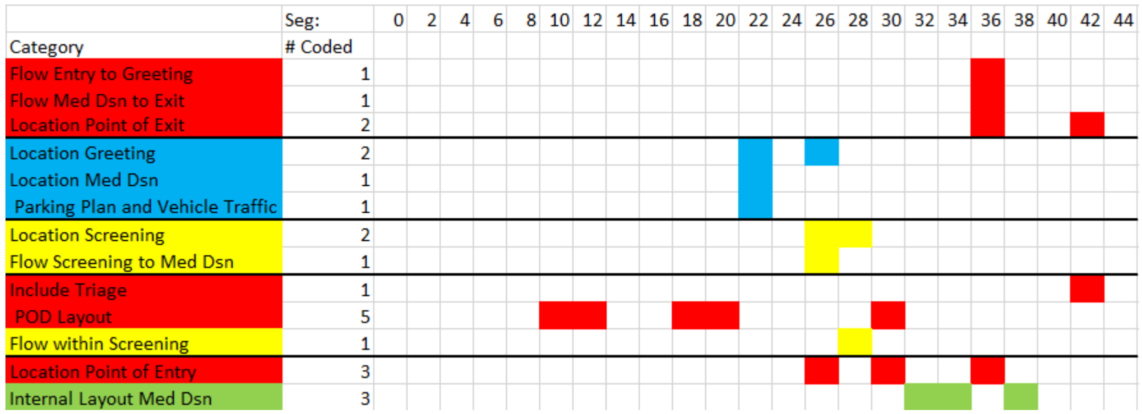


Figure 2.74: Student Team 1 Spectral Clustering Timeline

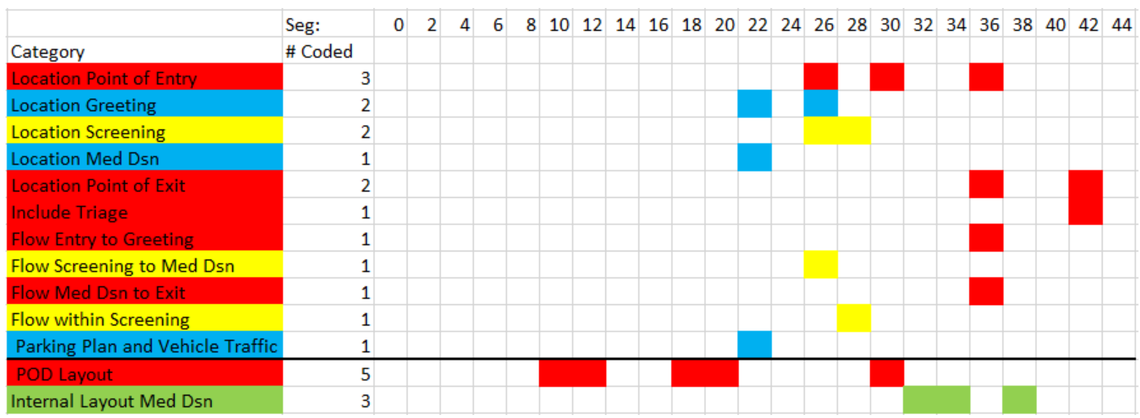


Figure 2.75: Student Team 1 Association Rules Clustering Timeline

2.4.3.7 Student Team 2

Student Team 2's clustering results can be seen in Figures 2.76, 2.77, 2.78, and 2.79. The green and red clusters can be seen in all of the results since they have no overlap with any of the other subproblems. The other clusters can also be seen in all of the results, but the Spectral and Association methods combined many of the variables into one big cluster. Ward's method did the opposite and split some of the clusters into two and did not use some of the larger variables.

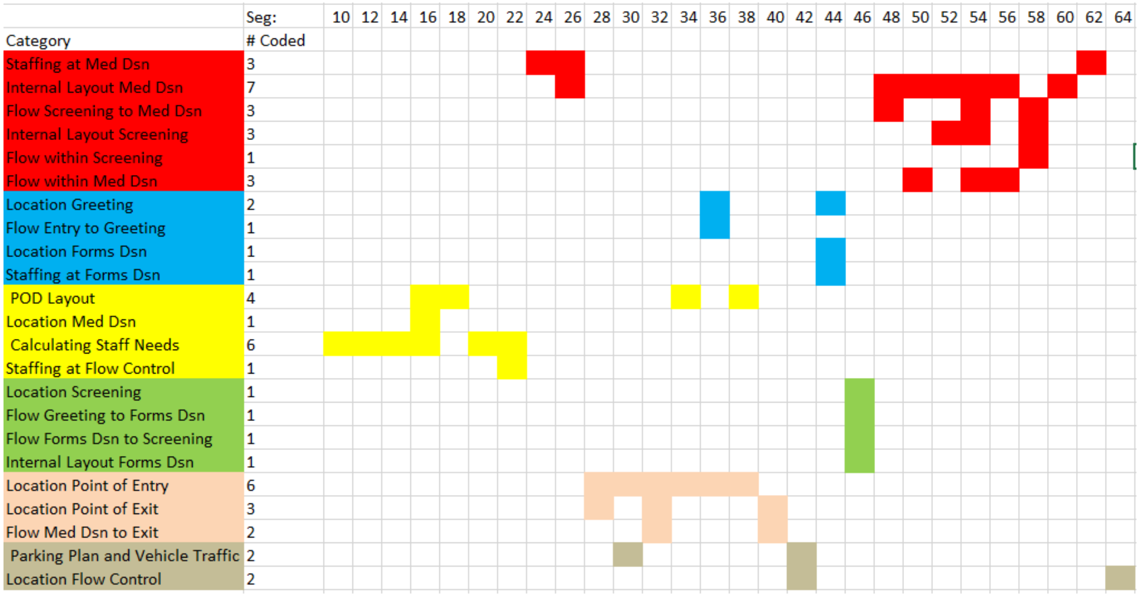


Figure 2.76: Student Team 2 Markov Clustering Timeline

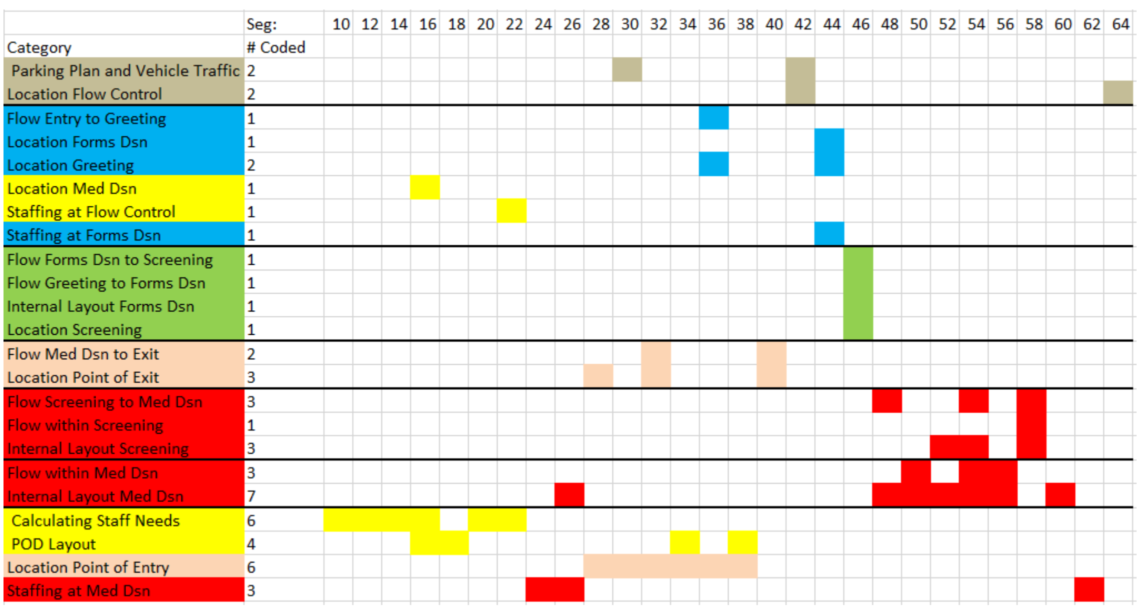


Figure 2.77: Student Team 2 Wards Clustering Timeline

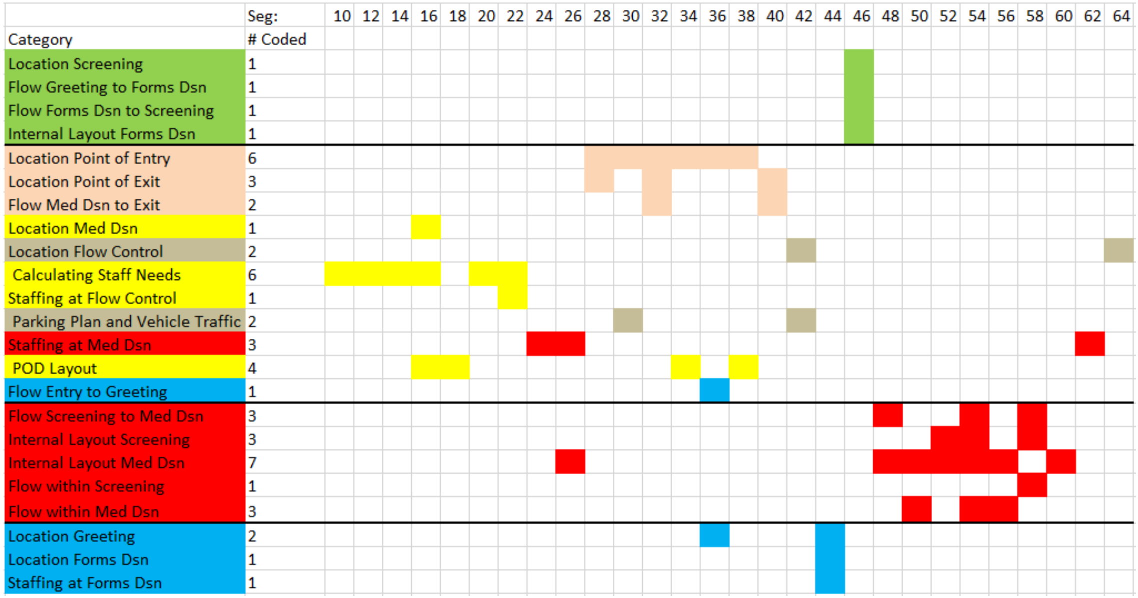


Figure 2.78: Student Team 2 Spectral Clustering Timeline

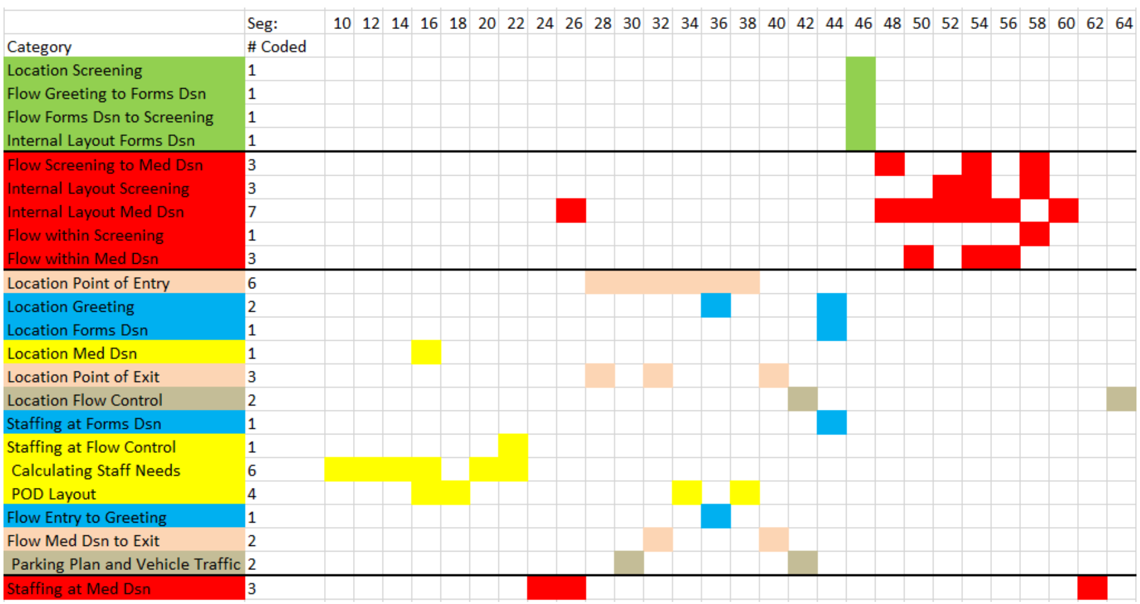


Figure 2.79: Student Team 2 Association Rules Clustering Timeline

2.4.3.8 Student Team 3

The clustering results fro Student Team 3 can be seen in Figures 2.80, 2.81, 2.82, and 2.83. Despite having many spread out or sporadically coded variables, S3 has some clusters that the algorithms recognized. The blue, tan, and gray subproblems are distinct clusters in all of the algorithms, although the blue subproblem was did lose or gain variables depending on the algorithm. The other subproblems were either all grouped together due to their sporadic variables, or split several times (as seen in the Ward's method results). The orange variables were left unclustered by all methods except Ward's due to not being coded with anything else but still being coded relatively close to other small variables.

Figure 2.80: Student Team 3 Markov Clustering Timeline

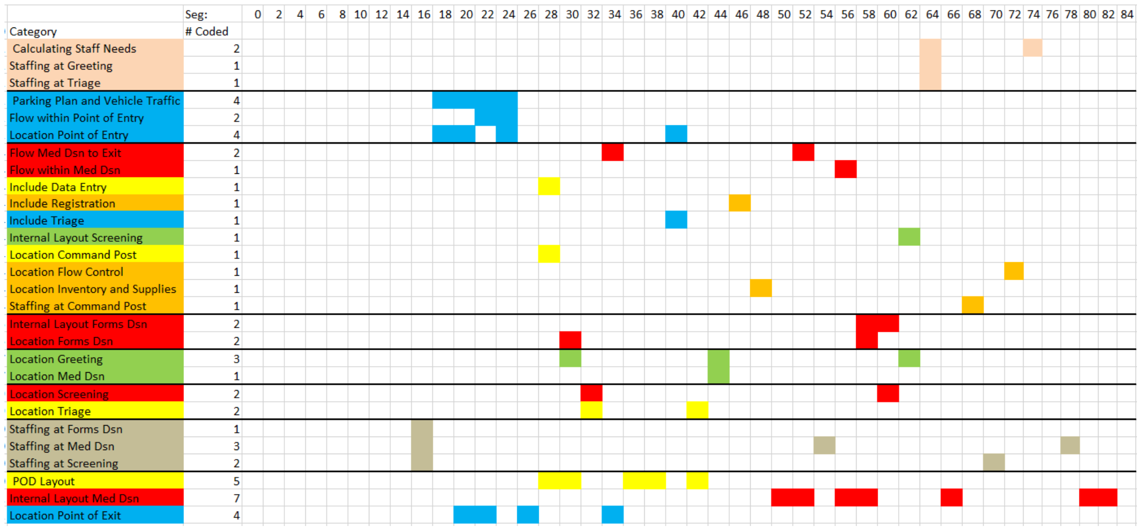


Figure 2.81: Student Team 3 Wards Clustering Timeline

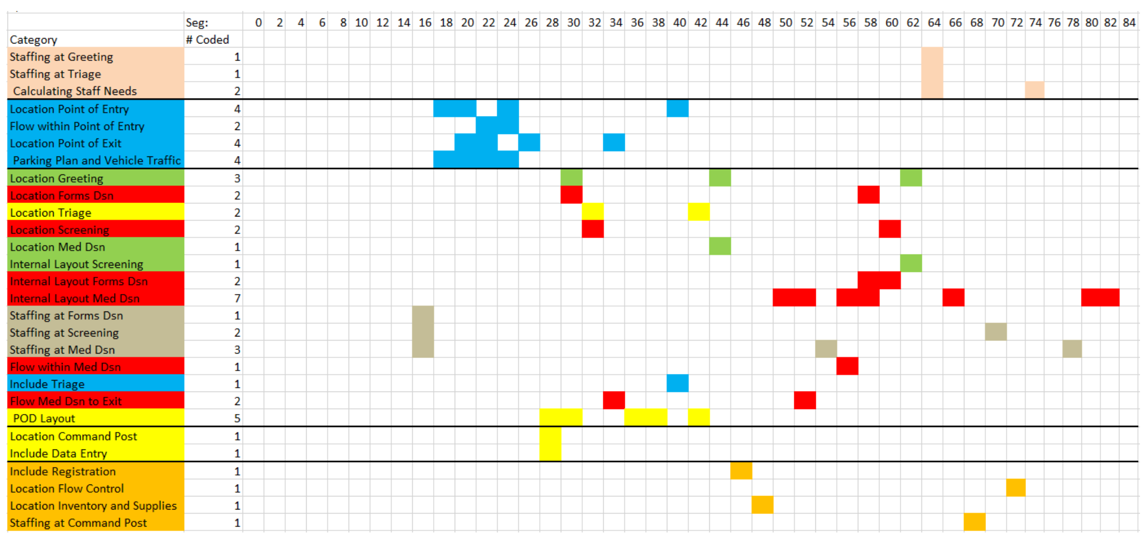


Figure 2.82: Student Team 3 Spectral Clustering Timeline

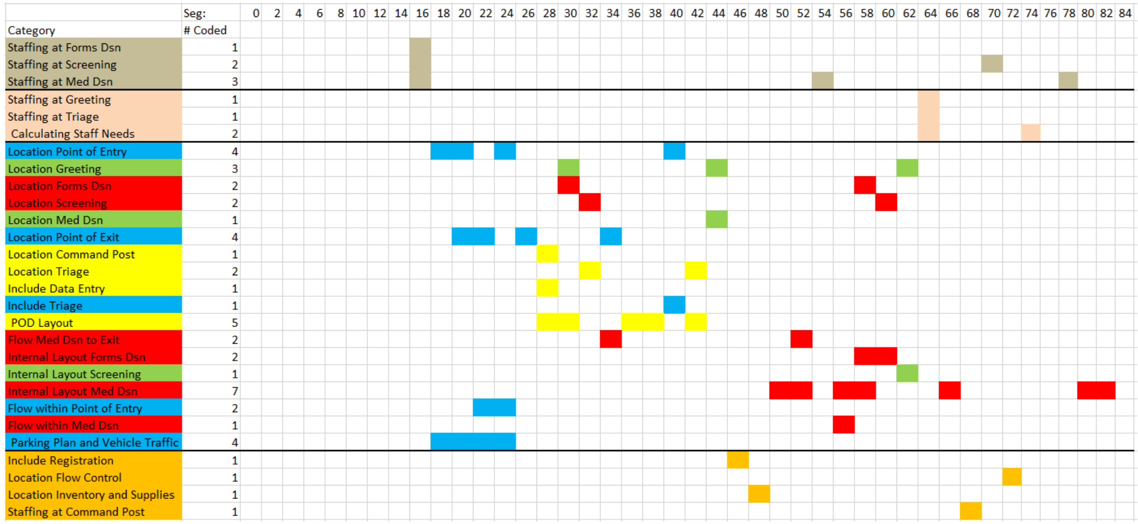


Figure 2.83: Student Team 3 Association Rules Clustering Timeline

2.4.3.9 Student Team 4

Student Team 4's clustering results can be seen in Figures 2.84, 2.85, 2.86, and 2.87. S4's Calculating Staff Needs variable was coded a relatively large number of times, and overlaps with over sporadic variables such as Staffing at Greeting. This caused all but the Ward's method algorithm to create one very large cluster. While the Markov and Spectral methods were able to at least create some smaller clusters alongside the large one, the Association method did not cluster any of the variables. Ward's method removed the large two variables from the final clusters, and was able to create 5 smaller clusters that have a relatively strong relationship.

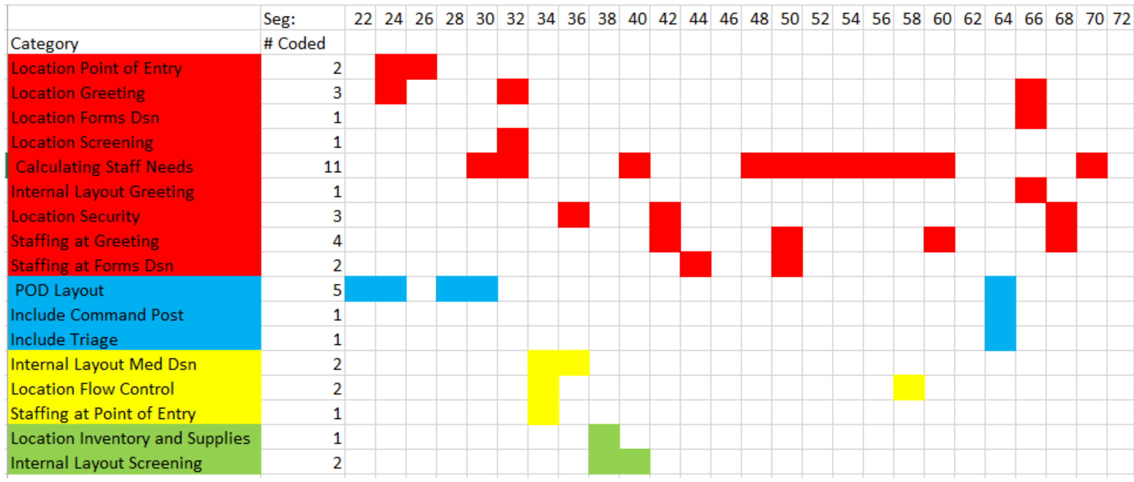


Figure 2.84: Student Team 4 Markov Clustering Timeline

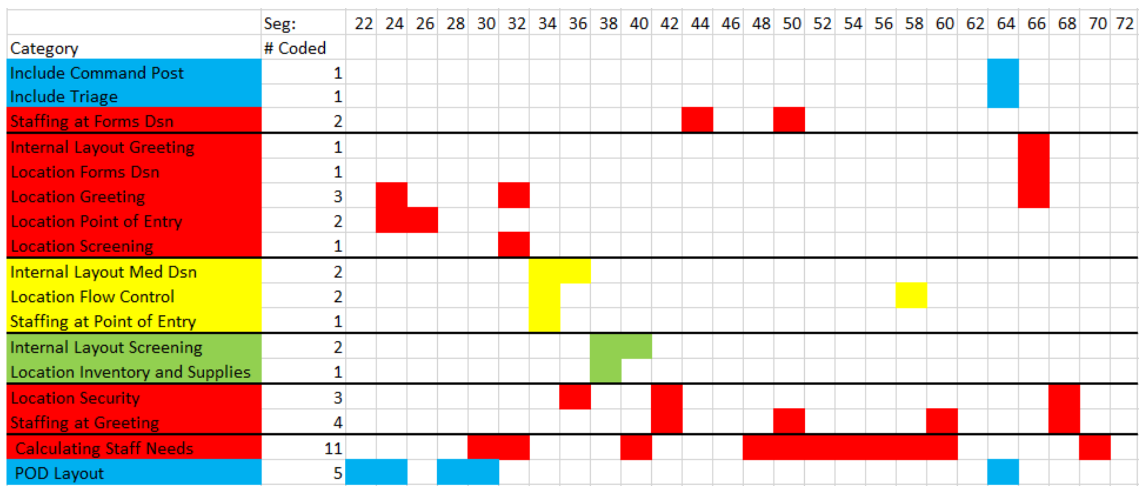


Figure 2.85: Student Team 4 Wards Clustering Timeline

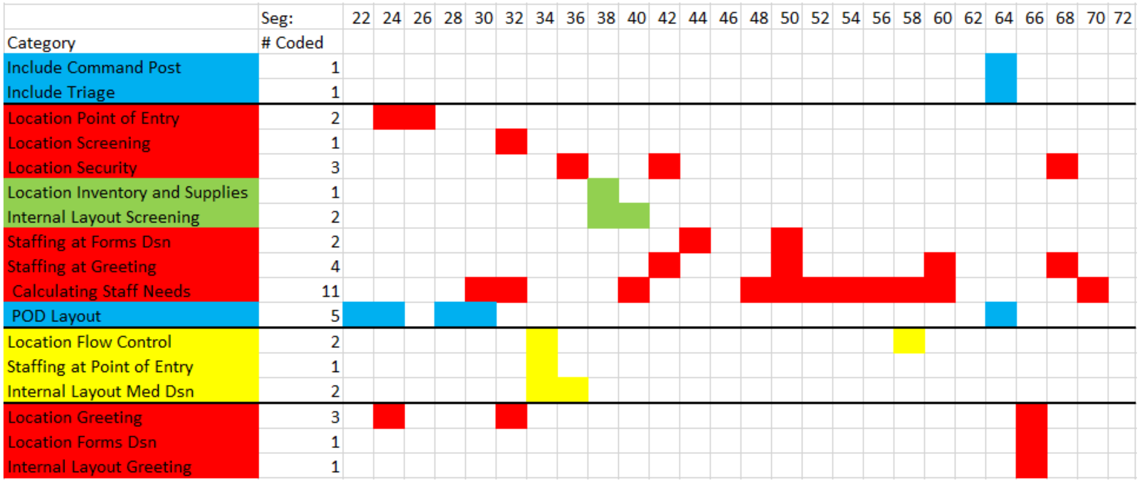


Figure 2.86: Student Team 4 Spectral Clustering Timeline

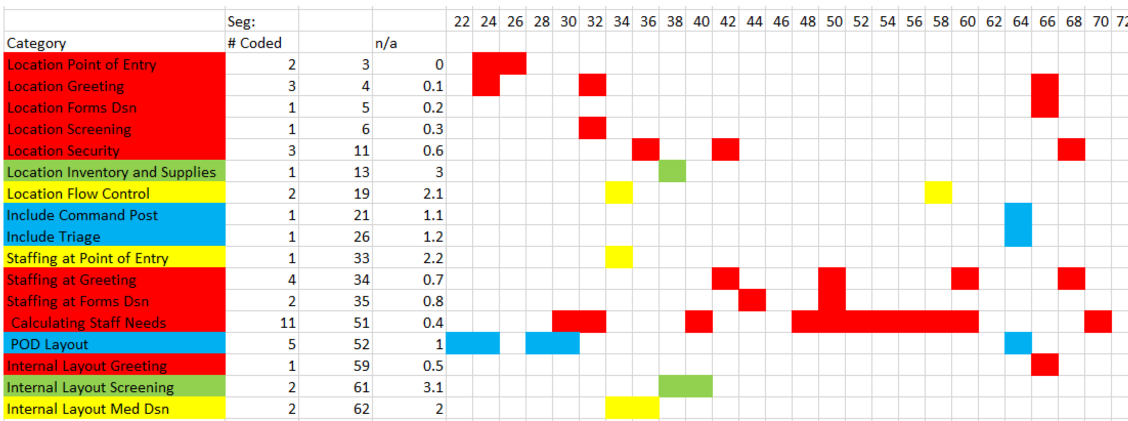


Figure 2.87: Student Team 4 Association Rules Clustering Timeline

2.4.4 Cluster Quality Results

We applied the cluster evaluation measures described in Section 2.2.3 to the clusters generated for the 9 teams in this study. For each team, we evaluated four sets of clusters: those generated by spectral clustering, by Markov clustering, by the association rules, and by Ward's method.

The results for the relative count measure are shown in Tables 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, and 2.10. The results for the concurrency measure are shown

Table 2.2: High relative count pairs in the same cluster: Professional Team 1

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	80	64	69	20	49
0.2	59	47	54	15	45
0.3	34	31	32	14	31
0.4	26	25	26	13	25
0.5	24	23	24	11	24
0.6	7	7	7	3	7
0.7	7	7	7	3	7
0.8	6	6	6	2	6
0.9	6	6	6	2	6
1	6	6	6	2	6

in Tables 2.11, 2.12, 2.13, 2.14, 2.15, 2.16, 2.17, 2.18, and 2.19. The Markov clusters had generally larger values for these measures for most sets. For the relative count measure, three or four sets of clusters yielded similar values.

The results for the silhouette values are shown in Table 2.20. The Markov clusters and Ward's clusters had generally more variables with positive silhouette values.

The results for the modified Dunn index, the original Dunn index, and the Davies-Bouldin index are shown in Table 2.21. Recall that clusters that are more separated from each other will have larger values of the Dunn index and smaller values of the Davies-Bouldin index. The Ward's clusters often had better values, but it did not dominate the other sets of clusters for all cases.

Table 2.3: High relative count pairs in the same cluster: Professional Team 2

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	36	25	30	8	20
0.2	23	15	21	6	20
0.3	13	9	12	3	12
0.4	10	7	10	3	10
0.5	9	6	9	2	9
0.6	5	5	5	2	5
0.7	4	4	4	1	4
0.8	4	4	4	1	4
0.9	4	4	4	1	4
1	4	4	4	1	4

Table 2.4: High relative count pairs in the same cluster: Professional Team 3

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	80	64	69	20	49
0.2	59	47	54	15	45
0.3	34	31	32	14	31
0.4	26	25	26	13	25
0.5	24	23	24	11	24
0.6	7	7	7	3	7
0.7	7	7	7	3	7
0.8	6	6	6	2	6
0.9	6	6	6	2	6
1	6	6	6	2	6

Table 2.5: High relative count pairs in the same cluster: Professional Team 4

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	80	64	69	20	49
0.2	59	47	54	15	45
0.3	34	31	32	14	31
0.4	26	25	26	13	25
0.5	24	23	24	11	24
0.6	7	7	7	3	7
0.7	7	7	7	3	7
0.8	6	6	6	2	6
0.9	6	6	6	2	6
1	6	6	6	2	6

Table 2.6: High relative count pairs in the same cluster: Professional Team 5

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	70	40	44	18	27
0.2	48	28	34	14	22
0.3	16	14	14	9	12
0.4	9	8	7	8	8
0.5	6	6	5	5	6
0.6	4	4	4	3	4
0.7	3	3	3	2	3
0.8	2	2	2	1	2
0.9	2	2	2	1	2
1	2	2	2	1	2

Table 2.7: High relative count pairs in the same cluster: Student Team 1

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	18	7	13	17	12
0.2	17	7	12	17	12
0.3	14	7	11	14	11
0.4	10	7	9	10	9
0.5	10	7	9	10	9
0.6	2	2	2	2	2
0.7	2	2	2	2	2
0.8	2	2	2	2	2
0.9	2	2	2	2	2
1	2	2	2	2	2

Table 2.8: High relative count pairs in the same cluster: Student Team 2

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	31	28	27	30	16
0.2	23	22	22	23	16
0.3	16	15	16	16	16
0.4	13	12	13	13	13
0.5	12	11	12	12	12
0.6	8	8	8	8	8
0.7	7	7	7	7	7
0.8	7	7	7	7	7
0.9	7	7	7	7	7
1	7	7	7	7	7

Table 2.9: High relative count pairs in the same cluster: Student Team 3

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	30	26	25	30	13
0.2	22	18	19	22	13
0.3	14	14	13	14	11
0.4	7	7	7	7	7
0.5	7	7	7	7	7
0.6	3	3	3	3	3
0.7	2	2	2	2	2
0.8	2	2	2	2	2
0.9	2	2	2	2	2
1	2	2	2	2	2

Table 2.10: High relative count pairs in the same cluster: Student Team 4

Value v	Relative count $N_p^a(v)$	Spectral $N_c^a(v)$	Markov $N_c^a(v)$	Association $N_c^a(v)$	Wards $N_c^a(v)$
0.1	18	12	15	18	11
0.2	15	10	14	15	11
0.3	10	9	10	10	10
0.4	6	6	6	6	6
0.5	5	5	5	5	5
0.6	2	2	2	2	2
0.7	2	2	2	2	2
0.8	2	2	2	2	2
0.9	2	2	2	2	2
1	2	2	2	2	2

Table 2.11: High concurrency pairs in the same cluster: Professional Team 1

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	160	128	138	40	98
0.2	151	122	130	38	98
0.3	110	88	102	33	80
0.4	97	78	89	30	72
0.5	96	77	88	29	71
0.6	56	51	56	20	47
0.7	50	45	50	14	42
0.8	49	44	49	13	41
0.9	49	44	49	13	41
1	49	44	49	13	41

Table 2.12: High concurrency pairs in the same cluster: Professional Team 2

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	91	67	70	20	40
0.2	77	55	59	18	40
0.3	55	37	50	14	37
0.4	46	29	41	14	30
0.5	43	26	39	11	30
0.6	18	15	18	6	18
0.7	15	12	15	3	15
0.8	15	12	15	3	15
0.9	15	12	15	3	15
1	15	12	15	3	15

Table 2.13: High concurrency pairs in the same cluster: Professional Team 3

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	313	77	147	172	48
0.2	238	62	113	134	44
0.3	165	46	91	96	35
0.4	97	31	51	57	25
0.5	79	27	45	47	22
0.6	44	16	26	25	12
0.7	27	10	17	15	8
0.8	21	6	12	9	7
0.9	18	6	9	6	7
1	18	6	9	6	7

Table 2.14: High concurrency pairs in the same cluster: Professional Team 4

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	313	77	147	172	48
0.2	238	62	113	134	44
0.3	165	46	91	96	35
0.4	97	31	51	57	25
0.5	79	27	45	47	22
0.6	44	16	26	25	12
0.7	27	10	17	15	8
0.8	21	6	12	9	7
0.9	18	6	9	6	7
1	18	6	9	6	7

Table 2.15: High concurrency pairs in the same cluster: Professional Team 5

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	144	82	89	36	54
0.2	135	73	82	32	54
0.3	96	56	65	32	44
0.4	65	45	49	29	33
0.5	59	41	45	23	30
0.6	34	27	31	14	15
0.7	30	23	27	10	13
0.8	28	22	26	8	11
0.9	28	22	26	8	11
1	28	22	26	8	11

Table 2.16: High concurrency pairs in the same cluster: Student Team 1

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	36	14	26	34	24
0.2	36	14	26	34	24
0.3	35	14	25	34	24
0.4	28	14	21	28	21
0.5	28	14	21	28	21
0.6	15	9	13	15	13
0.7	15	9	13	15	13
0.8	15	9	13	15	13
0.9	15	9	13	15	13
1	15	9	13	15	13

Table 2.17: High concurrency pairs in the same cluster: Student Team 2

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	62	56	54	60	32
0.2	54	50	49	53	32
0.3	50	46	45	49	32
0.4	41	38	37	41	30
0.5	40	37	36	40	29
0.6	31	30	30	31	24
0.7	25	24	24	25	21
0.8	25	24	24	25	21
0.9	25	24	24	25	21
1	25	24	24	25	21

Table 2.18: High concurrency pairs in the same cluster: Student Team 3

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	60	52	50	60	26
0.2	56	48	46	56	26
0.3	45	41	38	45	25
0.4	39	35	34	39	22
0.5	39	35	34	39	22
0.6	17	14	17	17	12
0.7	17	14	17	17	12
0.8	15	12	15	15	10
0.9	15	12	15	15	10
1	15	12	15	15	10

Table 2.19: High concurrency pairs in the same cluster: Student Team 4

Value v	Relative count $N_p^c(v)$	Spectral $N_c^c(v)$	Markov $N_c^c(v)$	Association $N_c^c(v)$	Wards $N_c^c(v)$
0.1	42	28	33	42	22
0.2	41	27	32	41	22
0.3	35	24	29	35	22
0.4	28	22	24	28	18
0.5	28	22	24	28	18
0.6	14	11	14	14	11
0.7	13	10	13	13	10
0.8	13	10	13	13	10
0.9	13	10	13	13	10
1	13	10	13	13	10

2.5 Clustering Discussion

2.5.1 Overview

This section will discuss what the clustering results mean and how they relate to the research questions proposed in Section 1.6. Section 2.5.2 will address how well these algorithms captured what we believed to be the subproblems after reviewing the videos from two professional teams. Section 2.5.3 will cover the strengths, similarities, weaknesses, and differences between the algorithms by comparing the timelines and cluster quality results. It will also cover any attempts that were made to improve the clustering results by manipulating or refining the inputs of the Markov clustering algorithm.

Table 2.20: Number of variables with positive silhouette values with original distance metrics

Team	Clusters	$d(i,j)$
P1	Spectral	16
	Markov	30
	Association	18
	Wards	32
P2	Spectral	9
	Markov	10
	Association	7
	Wards	12
P3	Spectral	19
	Markov	31
	Association	15
	Wards	35
P4	Spectral	14
	Markov	39
	Association	16
	Wards	43
P5	Spectral	7
	Markov	22
	Association	14
	Wards	33
S1	Spectral	9
	Markov	10
	Association	0
	Wards	10
S2	Spectral	10
	Markov	17
	Association	8
	Wards	15
S3	Spectral	8
	Markov	17
	Association	6
	Wards	24
S4	Spectral	9
	Markov	9
	Association	0
	Wards	15

Table 2.21: Cluster analysis results with original distance metric and centroids. [a]: only one cluster, so this metric is undefined.

Team	Clusters	Modified Dunn	Original Dunn	Davies-Bouldin
P1	Spectral	0.72	0.426	1.604
	Markov	0.859	0.447	1.746
	Association	0.986	0.447	1.25
	Wards	0.836	0.535	1.512
P2	Spectral	0.468	0.213	1.68
	Markov	0.639	0.302	2.026
	Association	1.017	0.447	1.536
	Wards	0.917	0.535	1.387
P3	Spectral	0.955	0.500	0.749
	Markov	0.853	0.392	1.488
	Association	0.890	0.522	1.173
	Wards	1.262	0.707	1.116
P4	Spectral	0.560	0.277	1.921
	Markov	0.904	0.302	1.699
	Association	1.031	0.302	1.194
	Wards	1.253	0.500	1.204
P5	Spectral	0.732	0.343	1.469
	Markov	0.866	0.408	1.551
	Association	1.225	0.447	1.225
	Wards	1.012	0.535	1.405
S1	Spectral	1.055	0.408	1.289
	Markov	1.119	0.378	1.334
	Association	0	0	0
	Ward	1.524	0.5	1.101
S2	Spectral	0.823	0.289	1.63
	Markov	0.972	0.5	1.365
	Association	0.868	0.408	1.589
	Ward	1.19	0.707	0.988
S3	Spectral	0.787	0.408	1.52
	Markov	0.968	0.471	1.639
	Association	0.838	0.408	1.676
	Ward	1.14	0.707	1.345
S4	Spectral	0.752	0.378	1.642
	Markov	0.899	0.378	1.625
	Association	0	0	0
	Ward	1.282	0.816	1.193

2.5.2 Comparing Clustering Results and Direct Analysis

2.5.2.1 Video Comparison to the Clustering Algorithms

Section 3.2.3 we summarizes each segment of video. We identified the subproblems each team discussed, the order in which they discussed them, and how each subproblem related to one another. Knowing all of this, we proceeded to compare our findings to the results from the clustering algorithms. The following paragraphs describe which method produced the results which most thoroughly reflected our observations.

Professional Team 3

For the first section of video reviewed (Segments 22 - 28) it appears that the Markov Clustering Algorithm seen in Figure 2.60 did the best job at capturing and grouping the subproblems discussed during this section, although all of the algorithms correctly identified and grouped at least part of these subproblems.

After reviewing the clustering algorithms for the second video section (Segments 40 - 48), the Markov clustering algorithm seen in Figure 2.60 captured the first two subproblems the best, and all of the algorithms failed to capture the final staffing discussion as a distinct subproblem.

Professional Team 4

For the first video section (Segments 26 - 32) the Markov Clustering Algorithm seen in Figure 2.64 did the best job at capturing the transition into and the discussion of the first subproblem. Other algorithms either grouped improper variables together or only grouped together a fraction of the variables. None of the algorithms were able to properly sort the final two variables discussed. This was due to the two variables only being coded together once and having a concurrency below 33%.

The second section of video from P4's discussion (Segments 56 - 64) had two subproblems. The first subproblem was discussed at the beginning and end of this section, while the second subproblem was discussed in the middle. As stated in Section 3.2.3, these subproblems were closely related in topic and the clustering algorithms failed to accurately capture this relationship. The results from the clustering algorithms were similar in that each algorithm split each subproblem across two clusters. However, Ward's method provided these clusters with the least amount of seemingly unrelated variables and thus performed the best in this section.

Comparing the clustering algorithm subproblems to the subproblems identified by an analyst is helpful, but is relatively costly in man-hours. This method verifies the clustering algorithm results and provides some insight into the more complicated discussion segments. However, it requires at least one analyst to review the video recordings twice: once to code the video for the clustering algorithm inputs, and a second time to focus on identifying subproblems. For these reasons, it would beneficial to limit the use of this method. The most productive situations to use this method in would either be to iteratively verify and refine the clustering algorithms or to identify subproblems in time segments where the algorithms show significantly different results.

This exercise also reveals that a small subsection of a larger discussion may appear to have certain subproblems which may not be a prevalent subproblem throughout the discussion. An example of this can be seen in P4's segments 56 - 64. The Internal Layout of Forms Dsn and Internal Layout of Screening variables were discussed multiple times before, during, and after this section of video. Based solely on this section one might try to cluster these variables with the Flow Through Screening and Flow Through Med Dsn variables. However, the rest of the discussion does not indicate a relationship between these four variables. Without reviewing the full video and focusing solely on subproblems, it would be very difficult to determine which variables fall into which subproblems at certain times.

2.5.3 Comparing Clustering Algorithms

2.5.3.1 Algorithm Strengths

The clustering algorithms performed very well when a few variables (two to four) were coded in the same one to three segments. An example of this type of cluster can be seen in Figure 2.52. P1's tan colored subproblem consisting of Location of Staff Break Room, Location of Inventory and Supplies, Location of Triage, and Location of Behavioral Health appears to be a cluster. The variables are tightly grouped around the same segments, do not overlap with other coded variables, are all optional stations , and fit under the Location category. Figures 2.53, 2.54, and 2.55 show that every algorithm grouped these variables together, with the exception of the association rules which grouped some of these variables together and did not group the others at all. This is due to the threshold used in the association rules, which is discussed further in Section 2.5.3.2.

The clustering algorithms all group together these types of clusters but tend to vary on how selective they are based on their methods. This characteristic can be seen clearly in P3's results, Figures 2.60, 2.61, 2.62, and 2.63. All of the subproblems undergo some pruning depending on the algorithm. The Markov algorithm is generally the most inclusive, and the results show that no variables were unclustered. The Ward's and spectral results are less inclusive but still manage to capture the variables that were coded in the same segments. In this example, we can see that Ward's method removes the variable Staffing at Flow Control from the yellow cluster due to its multiple, sporadic codes. The spectral method goes a step farther by removing Location of Flow Control variable from the yellow cluster, most of the variables such as Location Point of Exit, Location Inventory and Supplies, and POD Layout from the blue cluster, and does not cluster the Staffing at Command Post, Staffing at Inventory and Supplies, and Staffing at Medical Mgt variables from the gray cluster. These characteristics may be helpful in some situations, for if the sporadic variable Staffing at Flow Control was unrelated to the other variables then Ward's method would be better than the Markov method.

Finally we can view the most selective algorithm, the association rules. This algorithm does not show variable relationships below a certain threshold (see Section 2.2.3.4 for more details), and many variables that are coded once fall below this threshold despite being coded with other one time variables. It does capture clearly related variables such as the tan subproblem with Location of Screening, Internal Layout of Screening, and Internal Layout of Medicine Distribution just like the other algorithms. A clear example of its extreme selectivity, however, can be seen in the single coded variables such as Include Command Post and Location Security in Figure 2.63. Although the other algorithms clustered these variables into two different subproblems, the association rules method did not cluster them at all. Again, this strength is situational since these one time variables may be independent decisions quickly solved by the team or all one connected topic that was briefly discussed.

The quality measures show that the algorithms perform similarly to one another, with no clearly dominant option. However, looking at the relative count tables (Tables 2.2 - 2.10) reveals that all algorithms except the association rules recognize nearly all highly concurrent (greater than 50%) variable pairs. This speaks to the algorithms' ability to cluster clearly related variables and identify subproblems that do not overlap.

Overall these algorithms give us a way to quickly and objectively identify most subproblems so long as the video discussion are coded objectively with a proper set of variables. By using multiple algorithms and comparing the results, we can also identify segments of the video that need to be reviewed to discern which variables are involved in a certain subproblem. The varying levels of exclusiveness created by the algorithms individual methods can be useful for analyzing design problems, where teams may change topics within the same segment and give the illusion of two related variables.

2.5.3.2 Algorithm Weaknesses

The algorithms struggled with two different kinds of coding scenarios: a variable with a relatively large number of coded segments or sporadically coded variable like the Calculating Staff Needs variable in Figure 2.60, or multiple one time coded variables like the ones seen in the third subproblem in Figure 2.69. Each algorithm handles these situations differently, with varying levels of usefulness. However, it can be said that they generally struggle to appropriately cluster these variables.

For the large or sporadic variables, the Markov and spectral algorithms usually performed most poorly. They clustered what may have been two separate clusters together because they were both linked by the larger variable. This can be seen in P5's results, Figures 2.68 and 2.70, where the variable Staffing at Med Dsn connects three seemingly loosely related variables into one cluster for the Markov and connects many different variables together for the Spectral. Another example would be P3's Calculating Staff Needs variable in Figure 2.60, where there are clearly separate clusters being linked together by the one variable. In Figure 2.62 the spectral method used the same Calculating Staff Needs variable to connect many different clusters. The Markov cluster and spectral cluster can be seen side by side in Figure 2.88. Although the other algorithms struggled with these large or sporadic of variables, they seemed to be less consistent. For example, the Calculating Staff Needs variable was not clustered by Ward's method in Figures 2.61 but was clustered by the association rules in Figure 2.63. Conversely, in Figures 2.69 and 2.71 the Staffing at Med Dsn variable was clustered for Ward's method, but not clustered at all for the association rules. The differences in the algorithms' results could be used to determine which cluster a large variable most likely belongs in and allow the researcher to calculate a confidence level for this result. For example, if more algorithms that place the large variable in a certain cluster then the confidence level that the variable belongs in said cluster would be higher than if only one algorithm had clustered the variable in that group.

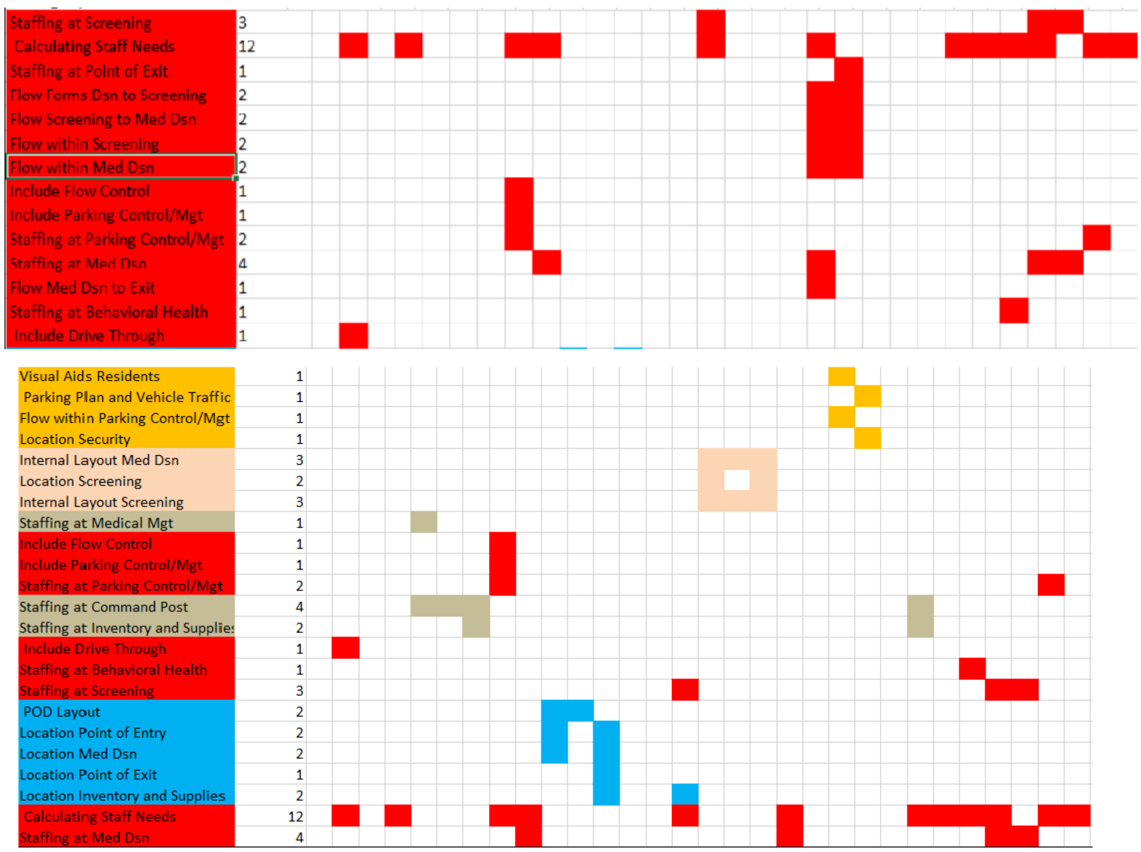


Figure 2.88: Top: How the Markov method clustered P3's Calculating Staff Needs, Bottom: How the spectral method clustered P3's Calculating Staff Needs

Variables coded in one or two separate segments also presented a problem for the algorithms, particularly the Ward's and association rules methods. Ward's method tended to group these single coded variables all together, despite never being coded during the same or even adjacent time segment. Examples of this can be seen in P2's first subproblem in Figure 2.57 and P5's third subproblem in Figure 2.89. In both of these subproblems, there are variables that were not coded during the same time segments. The association rules method handled these variables by not clustering them at all. This can also be seen in the unclustered variables in P2's and P5's association rules results (Figures 2.59 and 2.71).

The association rules method also fail to cluster clearly related single coded variables, such as the Include Command Post and Include Medical Mgt variables in Figure 2.63. These methods differ because the Ward's method clusters based on a distance calculation while the association rules cluster based on a concurrency and frequency relative to the the entire video time. If a group of single coded variables are in the same segment with no other variables, Ward's method will view them as relatively close and group them together. The association rules will also see them as close, but will view their relationship as less significant than a cluster with frequently coded variables. This basically defeat the purpose of these algorithms, which is to cluster the variables into closely related groups.

As discussed in the previous two paragraphs single coded variables, whether coded with another variable or by themselves, are clustered in a way that defeats the purpose of these algorithms. Instead of creating meaningful clusters that show the discussion based relationship between a set of variables and illustrate a team's decomposition method, single coded variables are clustered based on the lack of relationship or not clustered at all.

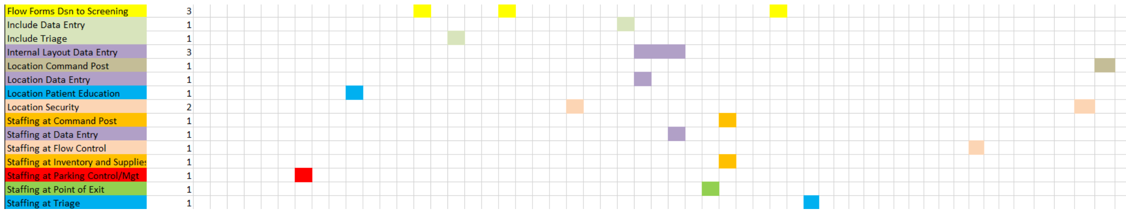


Figure 2.89: How Ward's method clustered a number of single coded variables from P5's discussion

Any algorithm that clusters single coded variables with other variables has a chance of incorrectly clustering that variable. A single coded variable may be something the team mentioned in passing, and did not consider to be a part of the subproblem being discussed during that time segment. If the variables were coded together multiple times, the relationship between the two is likely less coincidental. However, if all of the variables are frequently coded, it may also be difficult to see distinct clusters and determine which variables are related. These weaknesses indicate that there is a balance that should be achieved in order to make these algorithms useful. Variables that are too abstract may result in large coded segments, while variables that are too specific would result in single codes. Additionally, the problem and team may suggest breaking the teammates into two groups and solving problems simultaneously. This would obviously be difficult if not impossible for the algorithms to cluster and would require the researchers to separate the conversations as we did with four sections of video for two professional teams with our method described in Section 3.1.2.

2.5.3.3 Quality Measure Results

The quality results showed that no single algorithm dominated the others. This can be seen most clearly in the Modified Dunn, Original Dunn, and Davies-Bouldin results in Table 2.21. Here you can see that no method had the largest value for all three measure per team and the methods actually scored quite similarly in most cases. For example, the Markov method performed the best in most of the Davies-Bouldin measures, but for teams like S3 and S4 other algorithms performed as well or even slightly better. Similarly, the association rules scored well for most teams and was frequently the best according to the Modified Dunn score. However there were multiple teams where the association rules scored 0. This shows that these clustering algorithms excel in different situations and we cannot say there is one algorithm for all teams. However we can still study the quality measures in an attempt to determine where certain algorithms perform more poorly or better than the others.

Both the relative counts and concurrency counts for the professional teams show that the association rules method falls short of all others. For example, in Table 2.11 the association rules has a concurrency count of only 13 while all the methods range between 40 and 50 at the most stringent value of 1 (meaning the number of variables with a concurrency of 1). This trend persists through the professional teams but does not hold true for the student teams, which will be discussed further in Section 3.3.3. In all other cases, the association rules were similar to the other methods, and even led the others in the modified Dunn and original Dunn measures. This indicates that the association rules method provides good clusters but fails to cluster variables that may still be relevant. These results support the statements made earlier in this section and are the main weakness of the association rules method.

The remaining quality calculations do not show any seemingly significant data for determining which algorithm performs the best. This may be due to the relatively small data set collected from all of the teams, but may also simply indicate that no algorithm is better than the others. There is no reason to believe that these quality measures, which have been used to determine the quality of clustering algorithms in previous scenarios, are not well suited to analyze the results here [36–38]. However, without studying more teams or introducing new quality measures, little more can be said about the appropriateness of these measures or the quality of the clustering algorithms.

Chapter 3: Studying Decomposition Strategies

3.1 Methods for Identifying Decomposition Strategies

3.1.1 Overview

This section contains additional methods used to identify the decomposition strategies used by the teams. Section 3.1.2 describes how analysts reviewed the video recordings again to identify what they perceived to be subproblems, based only on the team's conversation.

3.1.2 Reviewing Videos and Manually Identifying Subproblems

In addition to regular video coding, described in the previous chapter, we also reviewed the video recordings to identify subproblems. This viewing was done at a separate time than the initial coding, and it was done by three researchers to diminish the method's subjectivity. We reviewed four sections of the videos from P3 and P4's videos in order to compare the clustering algorithms' subproblems to what we perceived to be the subproblems discussed. The sections were chosen based on their potential to be a transitional period where the team ends discussing one subproblem and begins discussing another, as well as the amount of disparity between algorithms when assigning variables in the section to subproblems. The two sections from P3 included time segments 22-28 and 40-48. The two sections from P4 included time segments 26-32 and 56-64.

This involved simply viewing the videos and noting what the team discussed during each two minute segment. Rather than identifying a specific variable or set of variables, each analyst was allowed to take notes however they pleased so long as they captured and understood how each discussion topic was or wasn't related. The three researchers later discussed their understanding of the video and the subproblems that they believed each team created during those time segments. Finally, the subproblems identified by the three analysts were compared to those identified by the clustering algorithms.

The results from this analysis can be seen in Section 3.2.3, and the discussion can be seen in Section 2.5.2.1.

3.2 Video Discussion and Decomposition Identification Results

3.2.1 Overview

This section contains the results before subproblem and clustering analysis was done to the discussions. This includes the pictures of the final designs, the original timelines, and the concurrency matrices. For each section, the professional teams will be presented first, followed by the student teams. Team names will be abbreviated, such that professional team 1 is P1 and student team 1 is S1.

3.2.2 Final Design Images

Figures 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 show the final designs done by the professional teams. Any designs or noteworthy labels were captured in the photos. All of the designs used the school's main entrance as the point of entry, and all but P3 used the gym as the medication distribution area. The layouts within the medication distribution area differ slightly from one another and use a variety of resources provided to the team such as barricades and tables.

Teams denoted staff differently as well. Some teams chose to not include staff numbers explicitly on their final design, while others denoted the number of staff near each stations, while still others used cutouts to place each staff member individually. Another variance between teams was the amount of detail in the hallways. For example P3 in Figure 3.4 labelled their hallway plan extensively while P2 provided very little details about the use of the hallway, shown in Figure 3.3.

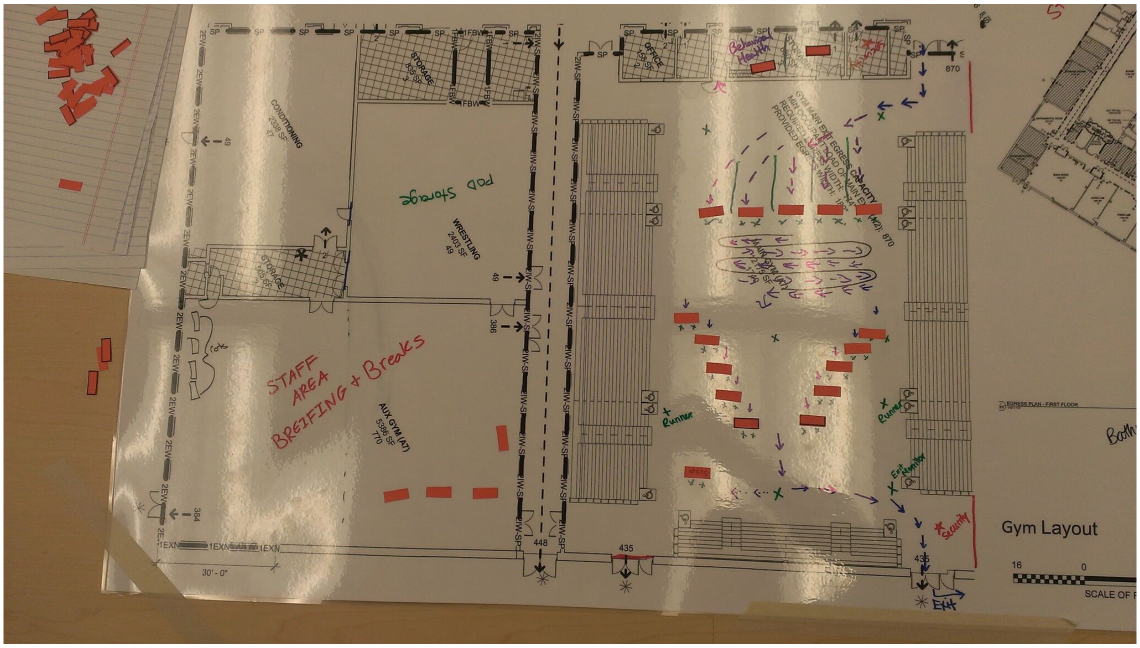


Figure 3.1: Professional Team 1 Final Design

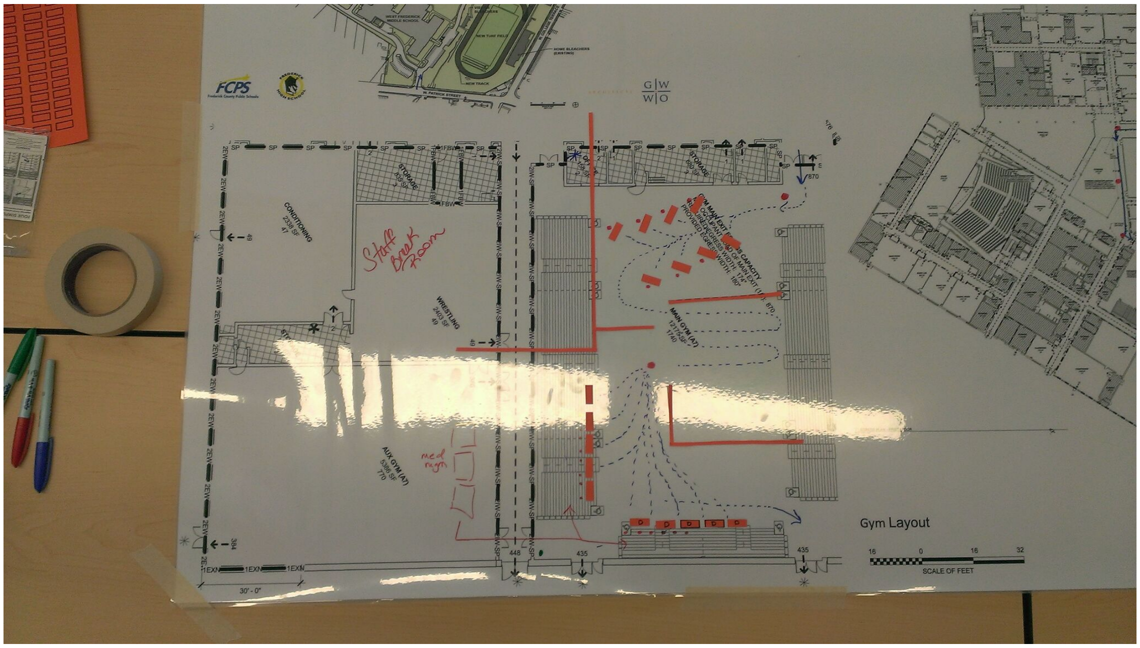


Figure 3.2: Professional Team 2 Final Design - Gym Only

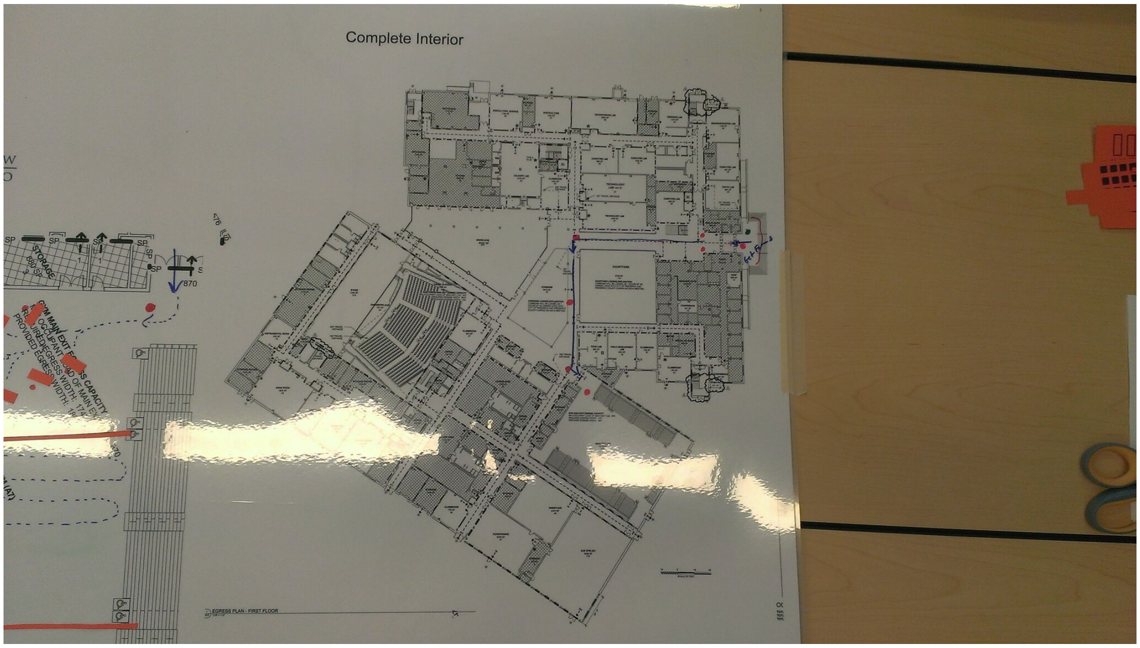


Figure 3.3: Professional Team 2 Final Design - Hall Only

Figure 3.4: Professional Team 3 Final Design

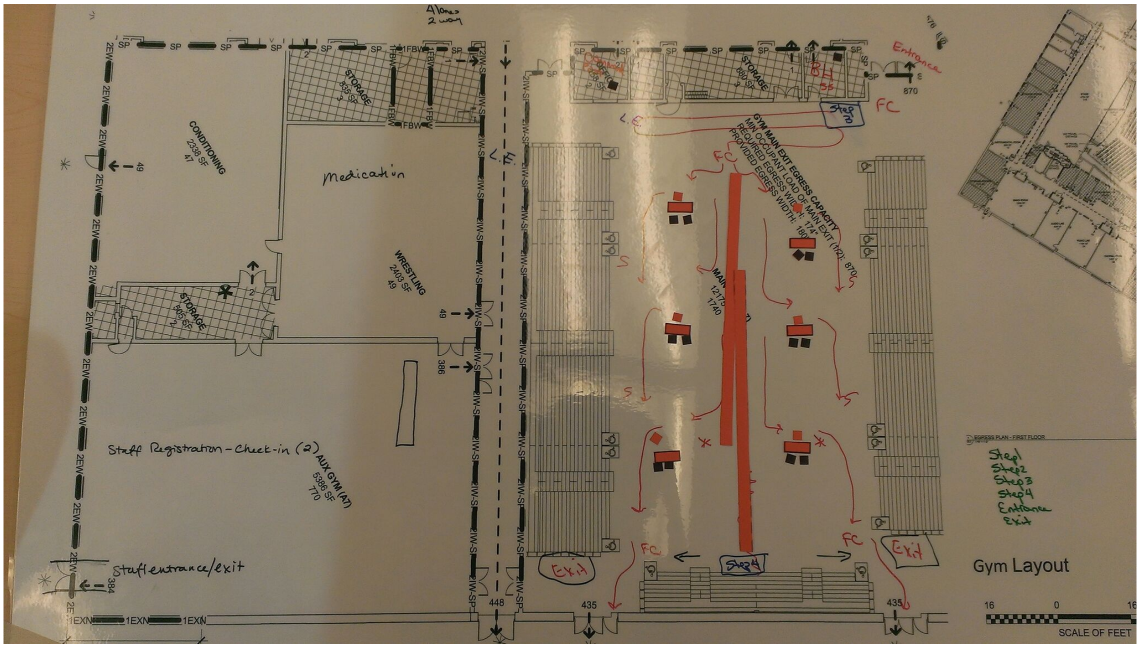


Figure 3.5: Professional Team 4 Final Design - Gym Only

Figure 3.6: Professional Team 4 Final Design - Hall Only

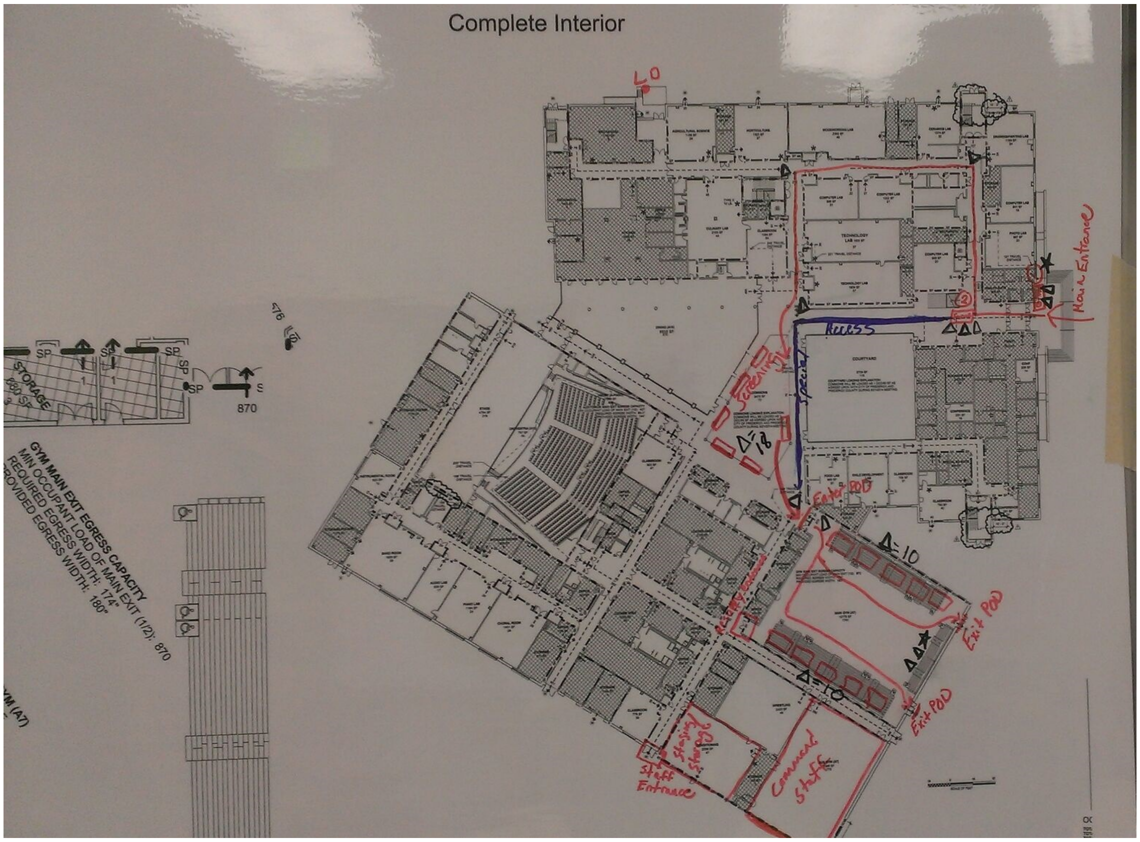


Figure 3.7: Professional Team 5 Final Design

3.2.3 Identifying Subproblems Directly from the Videos

We reviewed four sections of the videos of P3 and P4's discussions in order to compare the algorithms' subproblems to what we perceived to be the subproblems discussed. The sections were chosen based on their potential to be a transitional period where the team ends discussing one subproblem and begins discussing another, as well as the amount of disparity between algorithms when assigning variables in the section to subproblems. The two sections from P3 included time segments 22-28 and 40-48. The two sections from P4 included time segments 26-32 and 56-64.

Professional Team 3 Segments 22 - 28

Here the team discussed the general resident flow throughout the POD, focusing on the entry, exit, and key medication distribution points. We agreed that this was the first subproblem of this section. The second subproblem started as the discussion moved towards the location of and flow through stations after entry: specifically the greeting and forms distribution stations. The team used the flow of residents to transition from the first set of topics to the second set. In the first subproblem, the discussion encompassed resident flow through the entire POD, while in the second subproblem the discussion was station specific and centered around the resident queues forming after entry due to the time required at the forms distribution station. The team finished the segment by agreed upon the location and staffing of these initial stations.

During the final time segment the team began to move into the design of the screening and medical distribution stations but also touched on previously discussed topics like the Location of Inventory and Supplies. We felt that the transition to these variables made sense, because the screening and medication distribution follow the greeting and forms distribution in the POD process but did not neatly fall into the first or second identified subproblems.

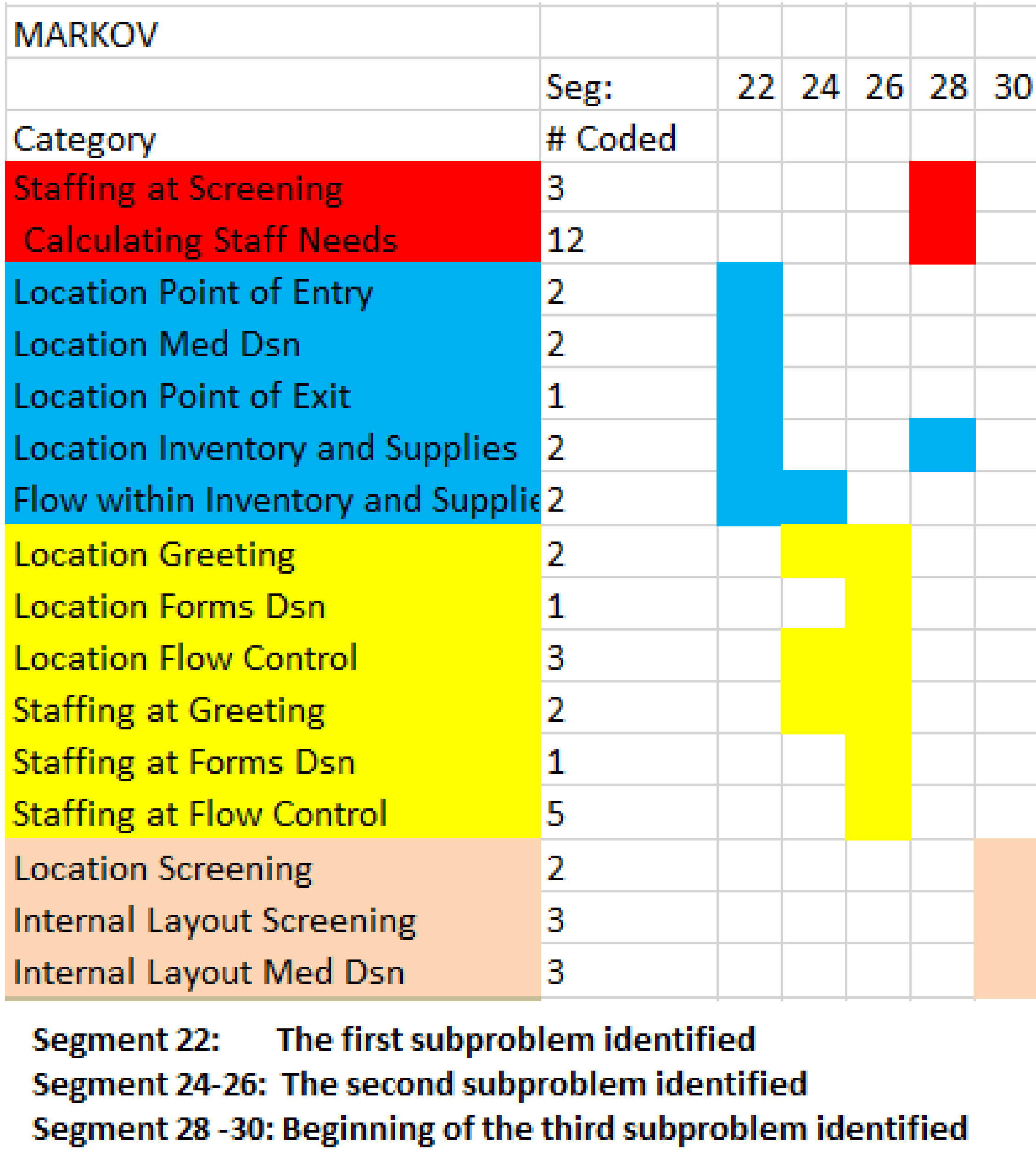


Figure 3.8: Excerpt of Markov clustering results from P3, Variables with no locally coded segments have been omitted

Professional 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 in the video, the team had a majority of their designed finalized and spent the next few segments discussing miscellaneous or noncritical stations such as the Command Post and Medical Management. We believed these two topics, exit/parking flow and miscellaneous stations, were two separate subproblems. The team then began talking about staffing at the miscellaneous stations but also considered other stations and the POD as a whole. This was a distinct shift from the previous subproblem and had a strong theme of staffing, which we identified as this section's third and final subproblem.

MARKOV							
	Seg:	40	42	44	46	48	50
Category	# Coded						
Staffing at Screening	3						
Calculating Staff Needs	12						
Staffing at Behavioral Health	1						
Staffing at Flow Control	5						
Staffing at Command Post	4						
Staffing at Inventory and Supplie:	2						
Location Security	1						
Parking Plan and Vehicle Traffic	1						
Flow within Parking Control/Mgt	1						
Visual Aids Residents	1						

Segments 40-42: The first subproblem identified
Segment 46: The second subproblem identified
Segments 48-50: The third and final subproblem identified

Figure 3.9: Excerpt of Markov clustering results from P3, Variables with no locally coded segments have been omitted

Professional Team 4 Segments 26 - 32

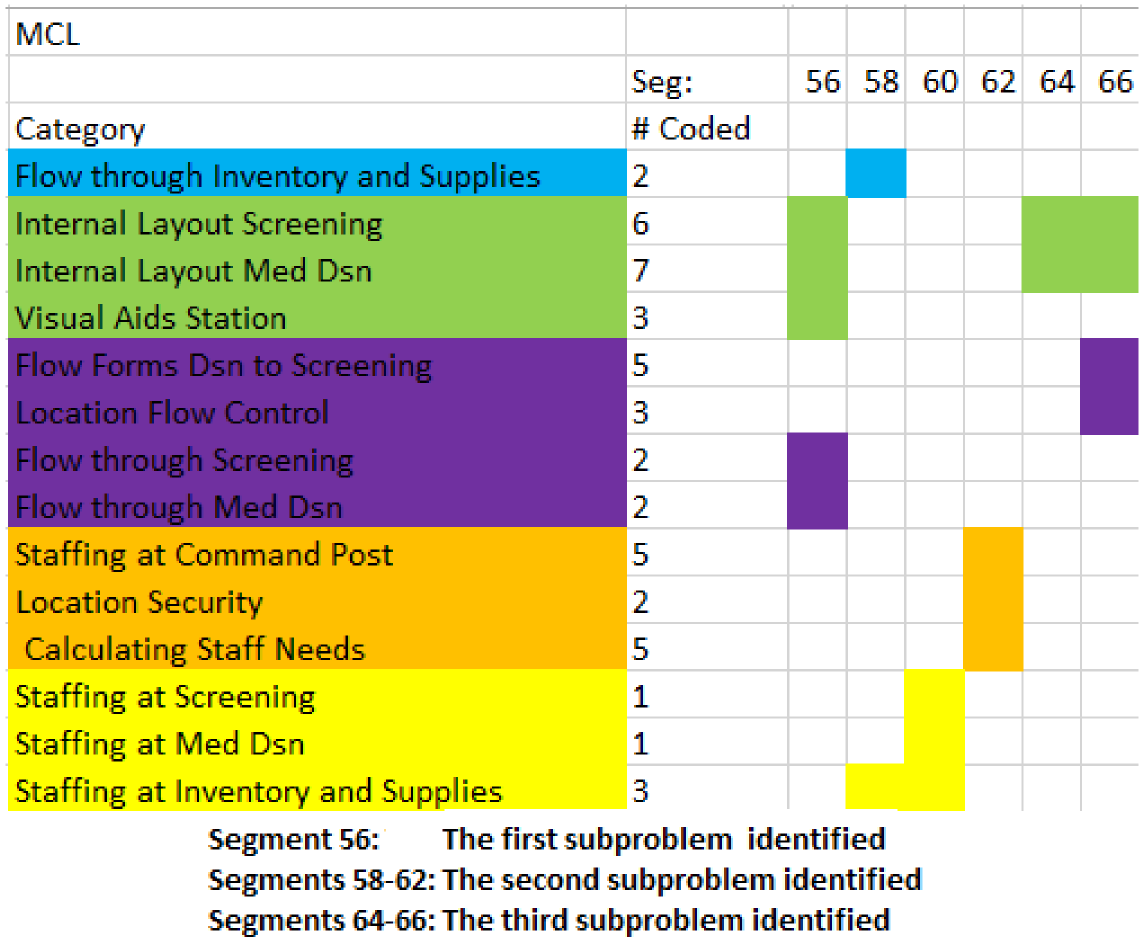
P4 took a slightly different approach to developing their design. In this section, the team started by discussing the flow of people from the parking lot to the entrance. They then moved their focus to the POD's indoor stations, such as greeting and forms distribution. The transition from outdoor planning to indoor marked the start of the first subproblem. Team 4 talked about the flow of people between the entry, greeting, and forms distribution station much like Team 3 did. However, instead of emphasizing the location of these stations this team was more concerned with the actual flow of residents through the building. This section ended with the design of forms distribution station. This primarily involved the Internal Layout of Forms Distribution and Flow Through Forms Distribution to Screening variables. We believed that the internal layout conversation was a completely new subproblem, while the flow between forms and medication distribution belonging in the first subproblem.

Figure 3.10: Excerpt of Markov clustering results from P4, Variables with no locally coded segments have been omitted

Professional Team 4 Segments 56 - 64

This section of P4's discussion started with the team talking about the internal layout of the medication distribution and surrounding stations. This quickly turned into a consideration of the staffing in the medication distribution area. We identified this as a transition point between two subproblems. The team focused on the Staffing at Screening, Staffing at Inventory and Supplies, Location of Security, and Staffing at Medication Distribution variables.

These four variables were clearly related to one another and the previous decisions the team had made, so we recognized this as the second subproblem of the section. The team concluded their discussion by returning to the first subproblem and confirming the internal layout of the screening and medication distribution stations. We decided that this was a return to the first subproblem and not a unique issue and, despite the clear relationship between the first two subproblems, the team tackled these topics separately.



MCL							
	Seg:	56	58	60	62	64	66
Category	# Coded						
Flow through Inventory and Supplies	2						
Internal Layout Screening	6						
Internal Layout Med Dsn	7						
Visual Aids Station	3						
Flow Forms Dsn to Screening	5						
Location Flow Control	3						
Flow through Screening	2						
Flow through Med Dsn	2						
Staffing at Command Post	5						
Location Security	2						
Calculating Staff Needs	5						
Staffing at Screening	1						
Staffing at Med Dsn	1						
Staffing at Inventory and Supplies	3						

Segment 56: The first subproblem identified
Segments 58-62: The second subproblem identified
Segments 64-66: The third subproblem identified

Figure 3.11: Excerpt of Markov clustering results from P4, Variables with no locally coded segments have been omitted

3.3 Decomposition Discussion

3.3.1 Overview

This section will discuss how the teams decomposed the problem and how these results help to answer the research questions proposed in Section 1.6. Section 3.3.2 will cover different similarities and dissimilarities between the variables coded and subproblems identified. This will also include comparing how the students and professionals decomposed the problem. Section 3.3.4 will compare the professionals' final design images to the subproblems that were identified, and discuss their possible decomposition strategies related to their design choices.

3.3.2 How Variables Were Clustered

The professional teams started discussing the basic flow through the POD and deciding on the main entrance and exit points relatively early in the exercise. The variables Location Point of Entry, Location Point of Exit, and POD Layout are clustered together by at least the Markov algorithm in all teams except P3, where the three variables were coded together in a later cluster. Table 3.1 compares which variables were clustered in the first chronological subproblem by the Markov method for each professional team. Notice how consistently the Location Point of Entry, Location Point of Exit, and POD Layout variables are discussed together compared to the other most frequently discussed variables in the first subproblem. This relationship makes sense to an observer but also suggests that teams start the design problem by understanding school's layout, setting boundaries, and making very high level decisions. The Include Drive Through and Parking Plan variables also appear relatively early, are clustered together, or are clustered with the Location of Entry and POD Layout variables. This can be seen in P1's second subproblem (cf. Figure 2.52), where Parking Plan and Include Drive Through variables were coded first and clustered together, as well as in P2's second subproblem where the Parking Plan, Include Drive Through, Location Point of Entry, and POD Layout

Table 3.1: Most Frequently Discussed Variables in the Chronologically First Subproblem (Markov)

Variable Name	P1	P2	P3	P4	P5
Location Point of Entry	X	X		X	X
Location Point of Exit	X	X		X	X
POD Layout	X	X		X	
Location Greeting	X				
Parking Plan and Vehicle Traffic Flow		X			X
Location Medication Distribution		X			X

variables were also coded first and grouped together (cf. Figure 2.56). This suggests teams worked from the outside in, and made their entry and layout decisions based on the parking plan. The one exception to this can be seen P3's second and seventh subproblems in Figure 2.60. This team did not discuss the Parking Plan variable until much later and did not discuss it near the Location Point of Entry nor POD Layout variables. This may have influenced their unique final design, which is discussed in more detail in Section 3.3.4.

Teams also usually discussed certain stations together. Two common combinations include the medication distribution and screening stations as well as the forms distribution and greeting stations. Many times the teams discussed the medication distribution and screening stations together due to proximity and the need for a well managed queue between the stations. The forms distribution and greeting stations were generally combined together due to staff members at those stations potentially being able to do both jobs at once. Interestingly, S2 is the only student team that combined these variables together. Their first subproblem in Figure 2.76 shows a close relationship between the medication distribution and screening stations. The ability to recognize the relationship between these stations may have been something that comes with experience, which explains why a novice group would be unable to think of discussing these topics together.

Many subproblems seem to be connected by a theme, whether that be a specific station like greeting or certain category like internal layout. These can be seen throughout the professional teams, but two examples are P4's green internal layout subproblem and gray behavioral health subproblem in Figure 2.64. Other teams have combinations of themes, like P3's yellow subproblem in Figure 2.60 which combines the greeting, forms distribution, and flow control variables. These three examples are shown together in Figure 3.12. These different themes may reflect how the team solved those individual subproblems. For example, P3's final design (seen in Figure 3.4 and discussed in Section 3.3.4) combined greeting, forms distribution, and some flow control while investing more time into the screening and medicine distribution areas.

P4, Subset of MCL Results, Internal Layout Themed Subproblem

Internal Layout Point of Exit
Internal Layout Forms Dsn
Internal Layout Screening
Internal Layout Med Dsn
Visual Aids Station
Internal Layout Greeting
Internal Layout Point of Entry
Internal Layout Flow Control
Visual Aids Hallway (flow control)

P4, Subset of MCL Results, Behavioral Health Themed Subproblem

Include Triage
Include Behavioral Health
Location Behavioral Health
Staffing at Behavioral Health

P3, Subset of MCL Results, Mixed Themed Subproblem

Location Greeting
Location Forms Dsn
Location Flow Control
Staffing at Greeting
Staffing at Forms Dsn
Staffing at Flow Control

Figure 3.12: Subproblems with a Category, Subcode, or Mixed Theme, P3 & P4

Because only one theme appears frequently between teams, it appears that each team took a different approach to decomposing the overall problem. The one theme that does appear frequently involves optional stations such as behavioral health, triage, and command post. Subproblems with these variables can be seen in gray subproblem for P1, P3, and P4 as well as the purple subproblem for P5 (Figures 2.52, 2.60, 2.64, and 2.68). These subproblems usually occur in the middle or second half of the exercise and span only a few time segments. It appears that teams talk about the optional stations after making high level decisions and usually arrive at a solution relatively quickly. While we may learn a few things by analyzing which variables are clustered together simply by looking at the subproblems, there is more to gain by comparing the variable groupings to the design choices. This allows us to better understand how the team's design evolved and what connections the team made when moving from one variable to another.

In more general terms we can say that there are clearly relationships between aspects of a design problem and teams will focus on those. These themes seem to be broader than a set of specific variables and involve high level ideas like general layout or what to do with additional resources (stations in this instance). However it is still unclear why teams discuss certain themes and not other. In this study, each professional team covered at least one theme in a subproblem, varying from staffing to internal layout to all aspects of a single station. But there was no theme that every team covered. Considering the similarities between each of the teams designs (and the uniqueness of P3's design) it is difficult to say why each team picked their individual themes.

All of the themes are widely applicable when dealing with design, which may be why they are found in all of the professional teams in this study. Perhaps because these themes are so high level and generic they are often taught or experienced throughout a person's career. This is supported by the lack of complex themes in the student discussions seen in Figure 3.13 and would help to explain why no single theme appeared in each professional team. The professionals have a variety of experiences and design choices that may have left an impression on them if the choices worked well or poorly.

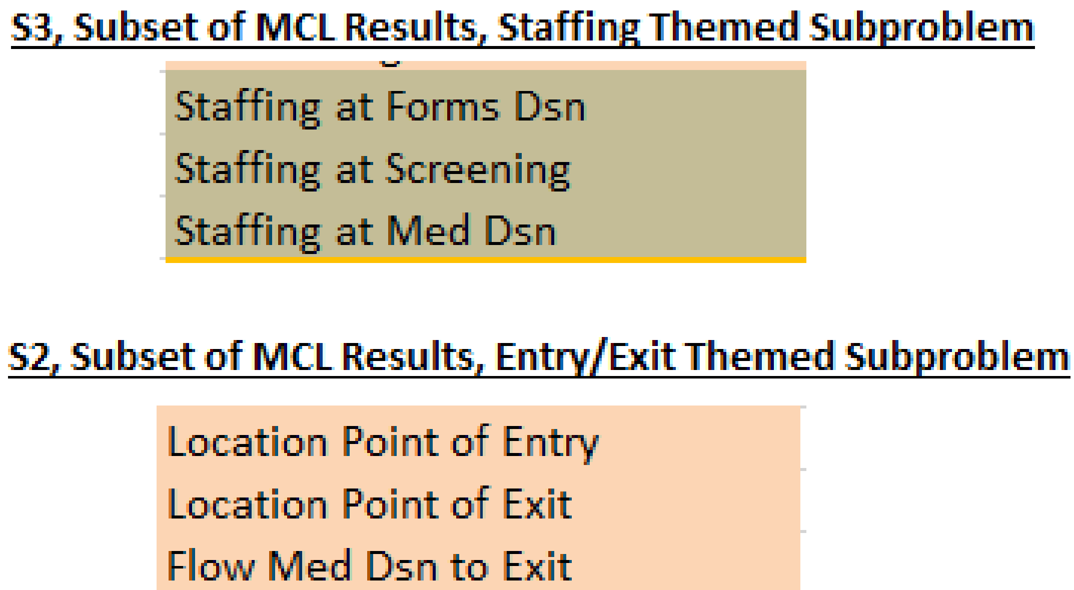


Figure 3.13: Themed Subproblems in Student Teams, S2 & S3

3.3.3 How Student and Professional Subproblems Compare

The students and professionals were presented with the same problem and had the option to use a similar amount of time to discuss the problem. However, the students discussed fewer variables and generally discussed the problem for a shorter amount of time. Student teams discussed variables in sporadic bursts rather than long stretches, with all but one variable coming in at under 6 total segments coded. In contrast, each of the professional teams had at least one variable with over 16 segments coded and some variables reaching over the students' maximum of 11 segments. The significant difference between the professional and student teams can be seen in Table 3.2. The student teams were unable to talk about a single topic

Table 3.2: Most Discussed Variable for Each Team

Team Name	Most Discussed Variable	Number of Times Coded	Avg
P1	POD Layout	7	
P2	Flow Within Med Dsn, Internal Layout Med Dsn	16	10.8
P3	Calculating Staff Needs	12	
P4	Parking Plan and Vehicle Traffic Flow	8	
P5	Internal Layout Screening	11	
S1	POD Layout	5	
S2	Internal Layout Med Dsn	7	
S3	Internal Layout Med Dsn	7	7.5
S4	Calculating Staff Needs	11	

as much as the professional teams. This may be due to the students oversimplifying the topic, or being unable to choose one topic as more important than the others.

S3 and S4 started by considering the Location of Entry and POD Layout variables, similar to the professional teams as discussed in Section 3.3.2. However, these variables were not clustered together, and only S4 discussed the Location of Exit variable towards the beginning of the problem. This suggests the students did not view the problem as a series of related decisions. The students chose the location of the entrance and exit based on where the school's doors were rather than how the POD would flow. This also relates to the subproblems' themes, as discussed in Section 3.3.2. The students' subproblems appear to either have no theme or a simple theme less than three variables (as seen in Figure 3.13), which makes sense since they are novices and should be less experienced with how to effectively decompose design problems [4]. The few exceptions to this observation, such as S3's gray staffing subproblem in Figure 2.80, are relatively simple or smaller in both variable quantity and segment length when compared with the professional teams.

The quality measures for the student teams were smaller and less varied from clustering algorithm to clustering algorithm than the ones for the professional teams. The association rules quality measures had the most pronounced difference between novice and expert teams. For all of the students' relative and concurrency counts, the association rules had similar results to the other methods, whereas for the professional teams the association rules' values were significantly lower. This may be attributed to the fewer number of variables discussed by the students, which made it difficult for any algorithm to perform as well as they did for the professionals. Another explanation may be that the students have fewer single coded variables, so all of the relationships were considered above the association rules threshold. The lack of single coded variables suggests that the students designed at a very high level.

Overall, the lack of complexity and fewer discussed variables for the student teams reflects what previous researchers found [1, 2]. It appears that the students thought about the problem without considering how parts work together or thinking about the problem's parts as a whole. They may also have oversimplified the problem due to their inexperience; without knowing what parts of POD execution are particularly difficult they may have ignored intricate issues. This would also explain why many of the student teams finished relatively early. However, more data would allow us to confirm these observations.

3.3.4 Subproblems Related to the Final Design

P1 and P2 share similar designs, but subtleties emerge after further reviewing their solutions and subproblems. Comparing P1's design in Figure 3.1 to P2's design shown in Figure 3.2 (both designs can be seen side by side in Figure 3.14), we see that both teams use the gym for medication dispensing, but their designs start with a screening area that leads into a queuing area, and the resident flow flows out of the gym to the parking lot. Although there are staff members in charge of resupplying the medication distribution tables, we do not see any inventory and supplies routes for P1 (Figure 3.1). This, combined with the lack of discussion involving the Flow Within Med Dsn variable, indicates that the team did not spend much time discussing how the stations would be resupplied. There appears to be no way to resupply the medicine distribution tables without crossing a line of residents or coming extremely close to them, which may pose a security risk. P2, however, discussed the Flow Within Med Dsn and Internal Layout of Flow Dsn variables more than any other variable. These two variables were also clustered with the Location of Inventory and Supplies, suggesting that the team made their decision based on a relationship between the three. As shown in Figure 3.2, their design has a clear route from the inventory staging area to the medication distribution tables and does not interfere with the flow of residents whatsoever.

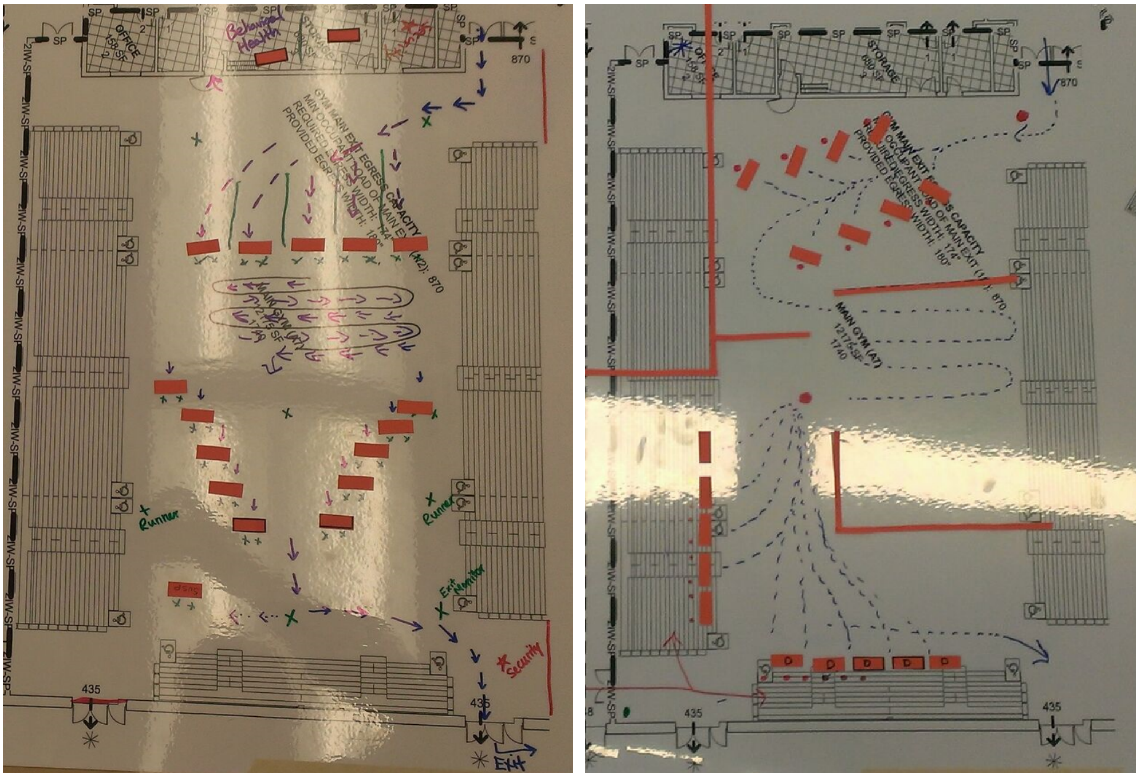


Figure 3.14: Left: P1's Gym Design, Right: P2's Gym Design

The design created by P3 stands out from the designs created by the other professional teams in that their medication distribution station was located in the cafeteria instead of the gymnasium. Most teams seemed to choose the gym due to its larger usable area and distance from the main entrance (so other stations could have adequate room leading up the gym). However, it is worth noting that all of the teams except P3 discussed traffic flow and parking plan relatively early on in the exercise. The gym has an exit that leads directly to the main entrance's parking lot, meaning that residents exiting the POD would have a clearer and shorter path to walk back to their cars. As shown in Figure 2.60 P3 starts by discussing staffing and then shifts to the general layout and flow of the POD but skips over the parking plan. Their next major subproblem, the blue one, involves the Location Point of Entry, Location Med Dsn, Location Point of Exit, and Location Inventory and Supplies variables. As shown in Figure 3.4, the team may have chosen this direct path from the entrance to the exit and used the cafeteria as their medication distribution point due to the uncomplicated and adaptable nature of the path as well as the distribution center's proximity to the loading dock (marked with a star on their final design). By not considering the residents' ability to return to the original parking lot from the exit, they were able to create a seemingly compact POD design.

P3 indicates that the order in which variables or subproblems are discussed affects the design. This makes sense, especially after considering P3's design process. If a team begins the design process without considering the system as a whole then they may make design choices that result in undesired consequences. P3 made a pivotal design choice without first considering the flow of residents from beginning to end, and thus later realized that the residents would have to walk around the school to get back to their cars. However it is still unclear if the order in which subproblems are discussed has as much impact as we saw in P3's design. Since the other teams discussed other subproblems at different times there is reason to doubt the importance of the chronological order of the subproblems. It may also be true that the first subproblem has an amplified effect on the final design.

Compared to other teams, P4 used the hallway and added more detail to their design before reaching the gym. Looking at P4's subproblems in Figure 2.64, we can see these choices reflected in the red, blue, and other subproblems. The team began with high level decisions involving the general location of stations, whether or not to include stations, and the parking plan captured by the red subproblem. They then moved inside with the blue subproblem and talked about the first required stations: greeting and forms distribution. Here they covered the location, internal layout, and flow within the stations which explains why their design has a significant amount of detail before reaching the gym. Finally, the team finished with the gym and surrounding area's design. We see that this process may have been iterative or done in parallel, as the green, purple, gray, orange, and yellow subproblems are all occurring at the same time.

P5 created a design that is similar to those created by teams P1, P2, and P4 in that the flow of residents in the POD starts at the main entrance, navigates through the hallway, goes through screening either in or right before the gym, and then goes through medication distribution in the gym. P5's red and blue subproblems seen in Figure 2.68 indicate they picked the location for most of their mandatory stations first. They then moved on to the flow, staffing, and internal layout of the POD as a whole, which may explain why their design is more comprehensive but not detailed in any one station. It appears that they worked on staffing quite frequently, as seen in their green and tan subproblems. This suggests a focus on resident flow rate, or making sure no one station acts as a bottleneck for the process. Thus this design process is unlike the processes used by the other teams, such as P2, who focused more on the medication distribution area, the inventory and logistics flow, or flow of resident through the medication distribution area.

Overall the subproblems help to supplement the team's designs and indicate a thought process that isn't easily seen in other analysis techniques like the concurrency matrices. Although distinctly clustered variables like P3's tan subproblem in

Table 3.3: Number of Segments Coded Per Each Third of Discussion

Team Name	1st Third of Discussion	2nd Third of Discussion	3rd Third of Discussion
P1	39	38	26
P2	38	36	24
P3	18	42	23
P4	44	53	36
P5	39	43	22

Figure 2.60 show a set of decisions made during a period of time, longer and more complicated subproblems like P5's tan subproblem in Figure 2.68 or P4's green and purple subproblems in Figure 2.64 provide insight into a recurring discussion or a parallel and dependent decision process.

The lack of similarity between each team's subproblems makes the similarity between the final designs surprising. However this does indicate that design teams can reach the same basic design without having an identical decomposition strategy. Based on the similarities, it appears that while subprobblem content does influence the end design, the extent of this influence may be somewhat limited. Instead, the beginning discussion (in this case the initial 30 minutes) may be more influential on the final design. The one team that differed from the others, P3, had a unique opening discussion and started their design based on some very different observations. Rather than studying how teams decompose the problem throughout the exercise, it may be more useful to focus how teams initially interpret and break down the problem in the first half hour at a much finer level.

Table 3.3 shows how few segments were coded during the first third of P3's discussion. Obviously P3 did not discuss the problem as thoroughly as the other teams, and perhaps did not understand all aspect of the problem as well as the others teams did. This supports the conclusion that the groundwork for a design is made in beginning of the discussion and the middle of the exercise is used to expand and refine these ideas rather than change them.

Chapter 4: Summary and Conclusions

4.1 Summary

This work set out to determine if clustering algorithms could identify subproblems, and if those subproblems would provide any meaningful insight into the design process. By using four different clustering algorithms, that worked in four different ways, we managed to gain a well rounded idea about these algorithms' strengths and weaknesses. The results from these clustering algorithms allowed us to compare the design process to final designs, and students to experts. Although the results were not definitively conclusive, they did present a new tool researchers can use to further understand the design process.

The work done here also helped to refine and document the methods used to capture and analyze team discussions, as well as add data for other researchers to review. Although complex and open ended, the data and results from this work provides answers to the research questions set at the beginning of this thesis, and presents new questions to be answered in the future.

4.2 Conclusions

Although many conclusions were made in Sections 2.5 and 3.3, a few are worth mentioning again. The clustering algorithms are useful as tools to identify subproblems, and their usefulness may be increased as data increase and tweaks are made to the methods. The task of measuring and comparing each algorithms' quality is a difficult one, but mathematical methods may prove to be useful in this endeavor. Determining what subpropblems were used and calculating the quality of their methods will allow researchers to analyze designs without having to be a part of the design process, or without having to record the design team. It also allows designers to better understand the intricacies of breaking down a large problem and identifying what issues are critical to a good design.

Other methods for analyzing the design process, such as the concurrency matrices and original category timelines, are useful as supplementary material but are not refined enough to show the finer details of a design process. Although they show relationships between variables, generally the data set is too large to effectively notice any patterns or too complex to see the subproblems. Together though, these methods and the clustering algorithms provide a well rounded story of the design process, and reveal relationships between the team's thought process and the final design not easily seen in the raw data.

4.2.1 Research Answers

We found that there are indeed easily repeatable ways to identify subproblems from a design team's discussion. However the results and quality measures we used indicated that the clustering algorithms are not useful in every situation and multiple algorithms should be used to get a more complete understanding of the team's decomposition method.

We also found that the order in which subproblems are discussed may affect the end design. When a team addresses a problem, they may focus on certain objectives or constraints that other team's dismiss, and their future decisions and subproblems will revolve around that initial thought process.

Finally, we found that experience level does affect how teams are able to decompose their problems. Novice teams will struggle to produce complex subproblems and thus create a less detailed adn thorough final design.

4.3 Future Work

Design teams and the design processes they use are extremely complicated. The number of teams studied in this paper, as well as the amount of data captured in these video recordings is not enough to make any definite statements. Any data that can be added to this collection and analyzed will help to identify characteristics of both the design process and the subproblem analysis methods. I would suggest using professional level individuals, and keeping each individuals number of years of experience above 1. As seen in this study, the professional participants discuss more topics in greater detail than the students. This provides the researcher with richer and plentiful data. Finding the correct design problem is also an important consideration, since previous work done with factory design problems resulted in more complex timelines and data [16]. While the POD design problem does yield a manageable amount of data, varying the degrees of freedom in these problems may help to make the results better. One possible way to do this would be to restrict the design choices each team can make either by reducing possible stations or forcing the teams to use certain room for medication distribution. This could in turn force a team to discuss a particular subproblem or focus on finer design details which would only be discussed for a few time segments.

Other work may be done in trying to improve, fine tune, or replace the clustering algorithms studied in this thesis to yield better subproblems. Although some research was done to improve upon a known weakness of the Markov method, there is still a lot of opportunity for fine tuning and input manipulation with all four of the methods. This may also improve upon the quality measurements and allow us to remove some subjectivity from the subproblem analysis.

Finally, the way teams are coded or recorded may be improved. Removing the human researcher from coding the design team's discussion may remove variances in the coded data as well as improve accuracy of the codes. Any reduction in subjectivity in this area would help to provide consistent, repeatable results. Methods such as voice to text software and machine learning may make computers the perfect tools for capturing and coding these discussions in real time. This would also mean that the researcher could discuss their results with the team shortly after they completed the exercise. This would not only validate the methods, but also give new insight into the design process.

Appendix A: Problem Statement

Below is the full POD design problem statement provided to the teams. Each participant was presented a copy of this form, as well as some markers and paper cutouts representing tables and barricades.

POD Design Exercise

SCENARIO

You are a member of a team that is creating a plan for setting up and operating a point-of-dispensing (POD) at Frederick High School in Frederick, Maryland. A brand new high school will be built there, so a new plan is needed. The POD will dispense antibiotics to the public if needed in response to a suspected aerosolized anthrax attack in the region.

The POD should be able to dispense the appropriate antibiotics to 14,000 residents in 24 hours. A "resident" represents one head of household getting medication for the persons in that household. The average household has 2.7 persons; thus, the POD will need to dispense 37,800 regimens. You can assume that sufficient medication will be available at the POD.

The medications to be dispensed include the following:

- Doxycycline (100 mg tablets, 20 tablets per bottle)
- Ciprofloxacin (500 mg tablets, 20 tablets per bottle)
- Amoxicillin (500 mg capsules, 30 capsules per bottle)
- Oral suspensions available for all three antibiotics as well

The POD will operate under a non-medical model. That is, the head of household is allowed to receive medication for others by providing some basic information about them (medical history, health status, and medication allergies), which will determine which medication each member of the household should receive.

The POD will operate for 24 hours, with two shifts of 12 hours.

RESOURCES

A total of 40 staff will be available for each 12 hour shift. You may request additional staff if necessary.
In addition, the Frederick Police Department will provide 2 officers at all times.

See the site map for information about parking and approaches. See the layouts for information about the size and location of various spaces on the first floor of the new high school.

Furniture: Numerous round cafeteria tables (72" diameter) and standard chairs will be in the dining area. Classrooms will have standard school desks. Folding tables (72" by 30") and folding chairs will be available for use in other spaces, such as the gym.

PROCESS AND STAFF

Under the non-medical model, the process flow for residents must include the following stations, but you may add others if you wish (see the list of Optional POD Components at the end of this document):

- Greeting
- Forms distribution
- Screening
- Medication Distribution

Average time for one staff person to serve one resident (head of household) at each station:

- Greeting: 15 seconds
- Forms Distribution: 15 seconds
- Screening: 1.75 minutes
- Medication Distribution: 2 minutes

The following POD staff positions are often included in POD plans. Your team may include some or all of these and may include additional positions if you wish (see the list of Optional POD Components at the end of this document).

- Command post: Site director and other leaders.
- Greeting: Staff greet arriving residents, answer questions, and direct residents to the appropriate place in the POD.
- Forms Distribution: Staff provide forms to residents to complete about medical history, health status, and medication allergies.
- Screening: Staff ask each resident for information about medical history, health status, and medication allergies and determine which medication(s) should be dispensed.
- Medication Distribution: Staff dispense medication to residents.
- Security: Staff who monitor crowds, call for emergency personnel, and respond to incidents.
- Data Entry: Staff who enter data about residents and medications dispensed.
- Inventory and Supplies: Staff who manage and distribute materials and medications.

Your team does not need to worry about the following factors:

- Number of PODs in the jurisdiction, traffic flow and/or transportation to and from the facility site, training, prophylaxis to POD staff and their families, media and public relations, medication distribution and resupply outside the POD, details of the forms, design of information systems, the organization chart, and local political pressures.

Optional POD Components (you may include these components in your POD if you wish to, but they are not required):

- Triage: Residents who are ill are examined by staff to determine if they need medical treatment (not prophylaxis) or should go to a hospital.
- Registration: Staff record the names and addresses of residents.
- Medical Management: Staff who can treat ill residents.
- Patient Education: Staff provide information about the diseases and the medications that are being dispensed.
- Behavioral Health: Staff who can provide mental health counseling and treatment to residents.
- Flow control: Staff who direct residents to the correct location in the POD.
- Parking: Support staff who direct traffic in the parking lots.

DESIGN PROBLEM

Your design team must determine the following features to complete the plan for this POD:

- The process followed by residents (that is, which stations should residents visit).
- The layout of stations within the facility (that is, where should each station be placed).
- The layout within each station, including the arrangement of tables, staff, queues, inventory, and supplies.
- The staffing at each station (how many people are needed there).

Expected Outputs:

- Laminated poster with facility layout drawn and flow through the facility clearly indicated.
- Staffing plan for each station.
- Presentation describing your POD design and explaining your design choices.

MAP SCALES

Gym Layout:	1” = 7.4 feet.
Complete Interior:	1” = 27.0 feet
Site Plan:	1” = 120.5 feet

Bibliography

[1] Saeema Ahmed, Ken M. Wallace, and Luciënne T. Blessing. Understanding the differences between how novice and experienced designers approach design tasks. *Research in engineering design*, 14(1):1–11, 2003.

[2] L.J. Ball, J.S.B.T. Evans, and I. Dennis. Cognitive processes in engineering design: A longitudinal study. *Ergonomics*, 37(11):1753–1786, 1994.

[3] L.J. Ball, J.S.B.T. Evans, I. Dennis, and T.C Ormerod. Problem-solving strategies and expertise in engineering design. *Thinking and Reasoning*, 3(4):247–270, 1997.

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

[5] L. Hong and S.E. Page. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46):16385–16389, 2004.

[6] M. LiCalzi and O. Surucu. The power of diversity over large solution spaces. *Management Science*, 58(7):1408–1421, 2004.

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

[8] Ganyun Sun, Shengji Yao, and Juan A. Carretero. An experimental approach to understanding design problem structuring strategies. *J. of Design Research*, 14(1):94, 2016.

[9] James G. March, Herbert A. Simon, and Harold Steere Guetzkow. *Organizations*. Blackwell, Cambridge, Mass., USA, 2nd edition, 1993.

[10] K. Lewis and F. Mistree. Collaborative, sequential, and isolated decisions in design. *Journal of Mechanical Design*, 120(4):643–652, 1998.

[11] Raanan Lipshitz, Gary Klein, Judith Orasanu, and Eduardo Salas. Taking stock of naturalistic decision making. *Journal of Behavioral Decision Making*, 14(5):331–352, December 2001.

[12] L.C. Schmidt, J.A. Schmidt, P.E. Smith, D.I. Bigio, and J.B. Contardo. *BESTEAMS: Building Engineering Student Team Effectiveness and Management Systems*. College House Enterprises, 2005.

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

[14] Allen Newell and Herbert A Simon. *Human Problem Solving*. Prentice-Hall, 1972.

[15] J. Austin-Breneman, T. Honda, and M.C. Yang. A study of student design team behaviors in complex system design. *Journal of Mechanical Design*, 134(12):124504 (7 paages), 2012.

[16] C. Tobias, J.W. Herrmann, and E. Gralla. Exploring problem decomposition in design team discussions. *International Conference On Engineering Design, ICED15*, 2015.

[17] M.S. Poole and M.E. Holmes. Decision development in computer assisted group decision making. *Human Communication Research*, 22(1):90–127, 1995.

[18] Herbert A. Simon. The Architecture of Complexity. *Proceedings of the American Philosophical Society*, 106(6):468–482, 1962.

[19] Rachel L. Abbey, Katherine A. Aaby, and Jeffrey W. Herrmann. Planning and managing mass prophylaxis clinic operations. In Brian T. Denton, editor, *Handbook of Healthcare Operations Management*. Springer, New York, 2013.

[20] R.M. Emerson, R.I. Fretz, and L.L. Shaw. *Writing Ethnographic Fieldnotes*. University of Chicago Press, Chicago, 2011.

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

[22] B. Glaser and A. Strauss. *The Discovery of Grounded Theory*. Aldine de Gruyter, 1967.

[23] J. Corbin and A. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, third edition*. Sage, 2008.

[24] A. Langley. Strategies for theorizing from process data. *Academy of Management Review*, 24(4):691–710, 1999.

[25] D.D. Woods. Process-tracing methods for the study of cognition outside of the experimental laboratory. *Decision Making in Action: Models and Methods*, 13:228–251, 1993.

[26] K.A. Ericsson and H.A. Simon. *Protocol Analysis; Verbal Reports as Data – Revised Edition*. MIT Press, 1993.

[27] J. Cagan, L. Leifer, S.M. Smith, M. Dinar, J.J. Shah, J. Linsey, and N.V. Hernandez. Process-tracing methods for the study of cognition outside of the experimental laboratory. *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE*, 2013.

[28] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[29] Somwrita Sarkar, Andy Dong, and John S. Gero. Design optimization problem reformulation using singular value decomposition. *Journal of Mechanical Design*, 131(8):081006, 2009.

[30] Somwrita Sarkar, Andy Dong, J.A. Henderson, and P.A. Robinson. Spectral characterization of hierarchical modularity in product architectures. *Journal of Mechanical Design*, 136(1):011006, 2014.

[31] Stijn Van Dongen. Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, 30(1):121–141, 2008.

[32] Stijn Van Dongen. MCL - a cluster algorithm for graphs. http://micans.org/mcl/, 2012. Accessed: 2016-11-10.

[33] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. pages 207–216. SIGMOD '93 Proceedings of the 1993 ACM SIGMOD International Conference on Management of data, 1993.

[34] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[35] M. Morency, A. Azrah, J.W. Herrmann, and E. Gralla. Using clustering algorithms to identify subproblems in design processes. *International Conference on Engineering Design;, ICED*, 2017.

[36] Joseph C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[37] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[38] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20:53–65, 1987.