INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED15
27-30 JULY 2015, POLITECNICO DI MILANO, ITALY

EXPLORING PROBLEM DECOMPOSITION IN DESIGN TEAM DISCUSSIONS

Tobias, Connor (1); Herrmann, Jeffrey W. (1); Gralla, Erica (2)
1: University of Maryland, United States of America; 2: George Washington University, United States of America

Abstract

When faced with the problem of designing a complex system, a design team must make many decisions. Because many design problems are too difficult to solve all at once, the team will decompose the design problem into more manageable subproblems. Although studies of individual designers have shown that explicit decomposition is seldom used, we conducted a pilot study to determine whether, because they need to communicate about their design decision making activities, design teams explicitly decompose their work. This paper describes the results of our study, in which we observed four teams of professional engineers who redesigned a manufacturing facility and analyzed their decision-making processes. The goal of the study was to gain insights into whether and when teams select a decomposition of the problem and whether they utilize explicit or implicit decomposition. The results show that the teams did discuss their decompositions, but not every subproblem was discussed, and they did not decompose the entire problem upfront. Future research will determine if other teams behave similarly and will investigate how teams determine which subproblem to solve next in the design process.

Keywords: Collaborative design, decision making, human behavior in design.

Contact:
Connor William Tobias
University of Maryland
Institute for Systems Research
United States of America
ctobias@umd.edu

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

When faced with the problem of designing a complex system, a design team must make many decisions. Because many design problems are too difficult to solve all at once, the design problem is decomposed into more manageable subproblems. The way in which a problem is decomposed may affect the quality of the solution that can be constructed, especially when time and resources are limited. The decomposition of design problems is governed by the design process, which may be a formal process imposed by an organization or an informal set of activities and responsibilities organically determined by a team of designers. Investigating the impact of decomposition on design solution quality will support the design of better design processes.

Most design is carried out by teams of bounded rational human designers. Although many aspects of engineering design teams have been studied, our understanding of how teams solve design problems and how their decomposition strategies affect solution quality is limited. Some decompositions that appear useful when each subproblem is solved optimally may perform poorly when humans solve them.

The research described in this paper is part of a larger study investigating both *how* and *how well* teams of human designers solve complex system design problems like manufacturing facility layout problems. In other words, the research aims (a) to understand how design teams decompose complex design problems into sets of related subproblems and integrate the solutions; and (b) to assess the impact of these decompositions on the quality of the solutions that the design teams generate. We observed four teams of human designers as they decomposed and solved a facility layout design problem.

This paper focuses specifically on understanding whether and how decomposition strategies are discussed by design teams and examining whether decomposition processes are explicit or implicit (Ho, 2001). Based on our analysis of the results, we have identified when these teams discussed decomposition options and selected decomposition strategies. The results presented here are based on the same data that were analyzed by Gralla and Herrmann (2014), who considered the decompositions that were used and concluded that problem decomposition does indeed influence the character and quality of design solutions and, in particular, that top-down and bottom-up decompositions lead to different types of design solutions. The present study re-analyzed the data to determine whether and when teams discussed decomposition and whether and when an explicit decomposition was used.

2 RELATED WORK

2.1 Humans as Designers

Empirical studies of human designers have shown that they exhibit characteristics of bounded rationality. Designers use various types of heuristics and search processes to find design solutions (Ahmed et al., 2003; Ball et al., 1994). Moreover, the design processes that human designers use are based on how they decompose the design problem (Ball et al., 1994; Ball et al., 1997; Liikkanen and Perttula, 2009).

Problem decomposition is even more important when a problem is solved by more than one designer because each subproblem may then be handed off to different individuals. Studies of teamwork, the process by which members seek, exchange, and synchronize information, show its importance for team decision-making (Lipshitz et al., 2001; Schmidt et al., 2005), but these studies did not address decomposition explicitly. Instead, one recent study of design teams showed how our theoretical concepts of decomposition do not match the ways teams actually work (Austin-Breneman et al., 2012).

Hong and Page (2004) and LiCalzi and Surucu (2012) used simulation models to explore how teams solve problems and measure the value of diversity.

2.2 Design Processes

Previous work on engineering design has shown that design occurs via a series of decisions (March and Simon, 1993). Some decisions may be done sequentially while others occur concurrently, and different patterns of decision-making can occur (Lewis and Mistree, 1998). A design process includes the tasks needed to make these decisions. The informal design processes employed by humans are largely dictated by the way they decompose 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.

2.3 Explicit and Implicit Problem Decomposition

Ho (2001) described two types of decomposition. An expert designer can design efficiently by using an explicit decomposition in which the structure of the design problem is represented at the beginning of the design process. An implicit decomposition does not begin by establishing the entire structure but creates the structure one step at a time throughout the process.
Liikkanen and Perttula (2009) used this idea in their study of individual designers. They identified as an implicit decomposition any design process in which the designer followed a structured goal-by-goal strategy. Although only three of the sixteen subjects used an explicit decomposition, they suggested that explicitly decomposing the design problem can be used to challenge or overcome the implicit decomposition and find a new way to view the problem.
Poole and Holmes (1995) described the importance of direct observation of group decision making and identified six major types of decision-making communication: problem definition, orientation, solution development, nontask, simple agreement, and simple disagreement. They found that groups (teams) used a variety of decision-making processes and that these processes were related to solution quality.
Studies of human designers have detailed their design activities and described how designers decompose problems but have not devoted much attention to how teams of designers decompose and solve design problems. Moreover, additional work is needed in order to understand how teams utilize explicit or implicit decomposition and how that affects their design processes.

3 METHODS

In this preliminary study, we observed four teams of engineers redesigning a factory as part of a two-day lean facility design course offered by the University of Maryland. Each team redesigned a fictional factory that makes multiple products and has a traditional functional layout. The exercise took approximately four hours. Each team was given a scenario that specified (a) information about the existing factory and products; (b) constraints, such as areas that cannot be changed; (c) goals for the redesigned factory, such as making space available for new activities; and (d) criteria for evaluating the redesigned factory, including productivity and material handling effort. Teams presented their final designs to each other for discussion and feedback. We observed and recorded the activities and discussions of the teams as they completed the exercise.
The 20 participants were professionals at manufacturing firms, with an average experience in industry of 14 years (min = 2 years, max = 38 years). The participants included eight engineers, three production and facility managers, six executives, two sales personnel, and one maintenance technician. The participants were segmented by their experience level, and then those in each segment were divided randomly into four teams, to ensure that every team had some persons with more experience and some with less. Thus, all four teams were nearly the same demographically.
We adopted this type of field experiment because it represents a useful balance between a completely controlled but highly artificial laboratory experiment and a natural but time-consuming field study (Hendrick and Kleiner, 2002). A data analysis of this type proved successful in documenting the activities of teams solving an operational planning problem (Gralla, 2012), so we were confident that it could be applied successfully here.

3.1 Data Collection

The primary data collection method was field observation, or ethnography (Emerson and Shaw, 1995; Spradley, 1980). During each exercise, the teams' design activities and presentations were video-recorded. Each team was observed by one researcher, who took notes during observation and then expanded them into field notes (Emerson and Shaw, 1995). Observation is a useful supplement to recording because researchers may catch behavioral elements that might be missed on camera and interact with participants outside of the design activity. All documents produced during the exercise were either physically collected or photographed, to triangulate the other data and document the design solution created by each team.

3.2 Data Analysis

The goal of the data analysis was to identify whether and when the teams discussed how they would decompose the design problem. Previous analysis (Gralla and Herrmann, 2014) determined how the teams decomposed their design problems and evaluated the solutions (facility layouts) that they created. We used qualitative data analysis methods to describe team activities. The rich but unstructured observational data captured each team's activities, and in the analysis process we built structured descriptions of what happened (Pettigrew, 1990).

We used qualitative data analysis methods including grounded theory (Corbin and Strauss, 2008; Glaser and Strauss, 1967) and process mapping (Langley, 1999). These methods complement one another in their emphasis on, respectively, organizing data into conceptual categories and linking actions or categories in a logical manner (e.g. chronologically). These types of methods have been used to analyze verbal protocols (Ericsson and Simon, 1993; Woods, 1993) and study human designers (Cagan et al., 2013).

For this analysis, we aimed to augment the previous analysis by examining whether the previously identified decompositions were determined explicitly by the teams. For this purpose, we developed a total of four codes. The first code, which was called "DO" (decomposition options), identifies segments of the team's conversation in which a team discussed options for decomposition, and the second code, which was called "DS" (decomposition selection), identifies segments in which they agreed to use a specific decomposition. For coding purposes, a decomposition is a set of two or more subproblems used to solve the design problem. We also developed two codes for identifying segments of a team's conversation in which they discussed which subproblem they should solve next and segments in which they agreed to solve a specific subproblem. These codes were called "SO" (subproblem options) and "SS" (subproblem selection). Table 1 lists the codes and archetypes for each code.

These codes were developed in order to identify segments of the teams' activities in which they discussed how they would decompose the problem. Intuitively, a segment of video coded with a "DO" or "DS" code suggests that the team was discussing the decomposition of a problem. A segment of video with "SO" or "SS" codes suggests the team was discussing some portion of a decomposition or deciding which decomposed element to work on next.

The "SO" and "SS" codes were applied only to those instances where the team directly discussed the selection of one or more subproblems to work on. This excluded many instances where it was evident that the team had selected a new subproblem to work on without discussion. Teams frequently transitioned to a new subproblem by proposing solutions to that particular subproblem without discussion of other subproblem options or any verbal confirmation that they would move to a new subproblem.

The researchers coded the data by examining the teams' discussions and actions to determine if they should be labelled with zero, one, or more of these codes. One researcher reviewed the videotapes of the teams, coded the relevant segments, and captured the pertinent phrases or sentences from that segment. (A segment is a two-minute portion of the conversation.) The other researchers reviewed these segments to verify the teams' comments and validate the codes.

Table 1: Archetypes for the codes.

Code	Archetype
"DO" (decomposition options)	"We could create a general flow around the factory and then layout each area; or we could create cells for each product and then determine where to put the cells."
"DS" (decomposition selection)	"Okay, we shall create the cells for each product and then determine where to put the cells."
"SO" (subproblem options)	"Should we determine the layout of the machine assembly cell or determine the layout of the office area?"
"SS" (subproblem selection)	"Let's determine the layout of the office area next."

4 RESULTS AND DISCUSSION

4.1 Decomposition Codes

Across the four teams, 20 segments were coded using these four codes (see Table 2). In Team 1's conversation, six segments were coded; in Team 2's, no segments were coded; in Team 3's, four; and in Team 4's, ten. Seven segments were coded "DO," nine segments were coded "DS," three segments were coded "SO," and one segments was coded "SS."

The approximately 310 minutes of video for each team yielded 155 unique two-minute segments. For four teams, we had a total of 620 segments. Because twenty instances of the codes were noted, just over 3% of all of the segments contained a comment related to problem decomposition. This fraction is consistent with the results of Liikkanen and Perttula (2009), who noted that only three of their sixteen participants had employed explicit decomposition, and only 4% of their coded segments focused on the problem or subproblem being solved.

Table 2: The coded segments for the four teams.

Team	Line	Segment start time (minutes)	Pertinent statement	Code
1	1	14	"As far as like a workflow for this, does it make sense to get a sort of cellular design down for the process? ...The whole idea is that is consolidated, compacted, it can move around as a unit and it will make things a lot easier when you're moving the R&D unit....so maybe we should just focus on the process first and see where it goes after that."	DO
	2	16	"Yeah, yeah that makes sense." [In reply to the above comment made immediately prior.]	DS
	3	98	"Did you guys make, you know the refurb & upgrade area, they want to expand that, as well as try to add [a gym]… And [move R&D near engineering]."	SO
	4	108	"Oh, we should also look at the aisle width thing from yesterday."	SO
	5	128	"Let's work from the paint shop out. So let's go from the paint shop out, and we'll worry about how we're going to pre-load the paint shop."	DS
	6	244	"So we just have a testing area and a shipping area to go somewhere, right?"	SO
2			(No segments for Team 2 were coded.)	
3	1	70	"So we look at it as a machine… As a cell. Yeah, so we build a cell around the sides."	DO
	2	98	"I think what we do is we lay out one cell, we figure out what the one cell would look like, and then what we do is we box those, and then we can lay them all out."	DS
	3	118	"We figure out how many cells we need to build, then we shrink them as much as we can."	DS
	4	160	"We need a place to store some frames."	SS
4	1	12	"I think as we set up a cell we'd get into that." [Referring to takt time.]	DO
	2	20	"You could begin to separate these into separate cells which we can organize in a sort of directional problem."	DO
	3	26	"We have a choice at this point: We can try and talk jointly, a single model, or we can take twenty minutes each and just kind of make an individual schematic and then we choose among those individual drafts."	DO
	4	26	"Okay….That sounds good to me."	DS

5	54	"I think we can start broad and then work down on the next system, so you can satisfy your larger constraints and as you get your boundaries sort of work on the next level down."	DS
6	98	"I was just thinking of getting…the actual manufacturing straight, and then after that we could fit…the stuff like fitness and prototyping and all that there…"	DO
7	114	"We want to look at the cell…in a flow way...and then the next thing is well within this cell, how many work stations do we need, and how many operators can stack all of those work stations."	DS
8	116	"It would be good to know the boundaries of the cell before doing individual work stations. Because we can do the major layout…and then we can look at Us or Ls…."	DS
9	122	"So should we look at the overall cells, spacing those out, then the storage for them, and then figure out the remainder of what we have for the assembly cell?"	DO
10	126	"I think what we do first is we figure out the large blocks and then figure out what space we have available for each of the assembly areas. So do it that way so when we look at the paint shop and the frame and the machine shop, making sure that that is good and co-located, that's going to be a good feeder to our assembly areas . So then we know what we're dealing with with space and go from there to figure out the shapes and u-shapes and flow, the person flow, the people flow."	DS

4.2 Decomposition Patterns

Gralla and Herrmann (2014) presented timelines that indicate when each team was working on each subproblem. On these timelines we marked the times at which coded segments occurred (when the teams discussed the decomposition), as shown in Figure 1. In addition, based on these timelines, we developed the following narrative descriptions of each team's decomposition and solving strategy.

- Team 1's initial discussions about decomposition (their comments at minutes 14 and 16) occurred during their process of understanding the whole problem and before they started attacking any one problem. Team 1 began by selecting locations for the blocks, placing them around the edges and leaving a big blank space in the layout for the assembly cells. In parallel, some team members worked on sequencing operations and allocating staff. They next spent time laying out the assembly operations in cells, independent of the main layout. They added the cell layouts to the main layout, and finally returned to locating blocks (this time including the grouped assembly operations).
- Team 2 began by considering the high-level flow logic, including the number of cells and which products would be assigned to them, independent of the spatial characteristics of the facility layout. Next, they started selecting locations for the major blocks and the assembly operations without specifically and separately discussing assembly cells. At the end, they considered the sequencing of operations and allocation of staff.
- Team 3 began by thinking about the assembly operations, specifically the sequencing and staffing of operations. After considering the overall flow, Team 3 focused on cell design (cf. comment at minute 70). Later, they discussed cell design again (minute 98) and then began designing a generic machine assembly cell, considering operation timing and staffing at the same time. They copied the cell and set it in the layout but realized that they had to adjust the cell layout due to an earlier mistake. They redesigned the cell and then determined the number of cells, set them in the layout, and located the remaining functional blocks around it.
- Team 4 began their work at a very high level, thinking about the direction of flow through the entire facility, and how each element should connect to the other elements. They then selected locations for the major functional areas of the factory and left a big space for the assembly operations (module assembly and machine assembly). Finally, they determined the layout of the

assembly areas, again thinking in terms of overall flow and connections between elements. Team 4 made the most statements about decomposition. Four of their comments (between minutes 12 and 26) occurred during their process of understanding the whole problem. Their comment at minute 98 about manufacturing occurred during a transition from discussing the overall layout to discussing the details of material flow between different manufacturing areas. Their comment at minute 114 about cell details occurred after the team discussed the material flow. Their comments at minutes 116 and 122 were part of the discussion of cell details. Their comment at minute 126 about large blocks marked the end of this phase.

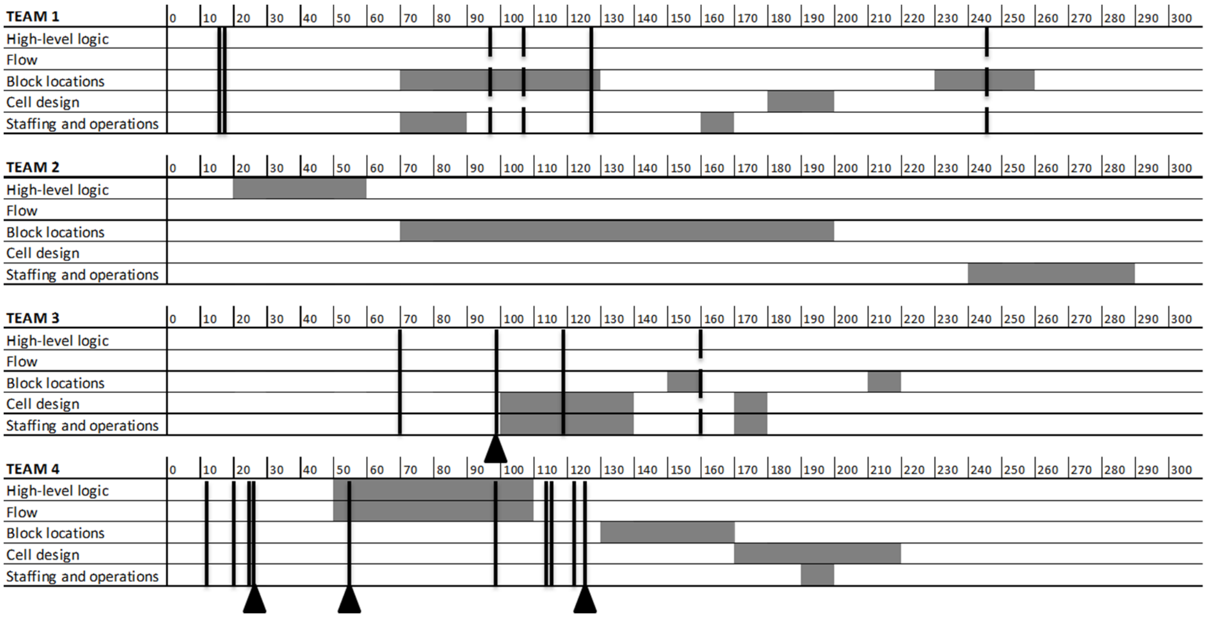

Figure 1: Timeline of subproblems with decomposition and subproblem codes overlaid. The solid lines denote segments coded as DO and DS, and the dashed lines denote segments coded as SO and SS. The triangles at the base of several codes denote decompositions that were discussed and subsequently implemented.

4.3 Analysis and Discussion

An examination of the data displayed in Figure 1 suggests that there are three cases that must be distinguished:

1. *Discussed and implemented:* teams discussed a decomposition strategy, *and* their subsequent actions were consistent with their strategy discussion.
2. *Discussed but not implemented:* teams discussed a decomposition strategy, but their subsequent actions were not consistent with their strategy discussion.
3. *Undiscussed yet implemented:* teams did not discuss decomposition, but their actions indicate that they nevertheless utilized decomposition within their design process.

Instances of the first case are most important for this analysis, because they represent segments of discussion that likely influenced the team's design process. Instances of the second case suggest that teams were thinking about decomposition, but it is not clear that those discussions had much impact on their design process. They may, however, have helped the team get "on the same page" about the problem they were solving and the goals they were trying to reach or helped the team think about decomposition in general. Instances of the third case are interesting because it is unclear how the teams decided how to decompose the problem if they did not talk about it.

Team discussions of decomposition provide some evidence about whether they utilized explicit or implicit decomposition. Recall from Section 2 that explicit decomposition occurs when an entire problem is decomposed or structured up-front at the beginning of the design process, while implicit decomposition creates the structure one step at a time throughout the process (Ho, 2001).

Our analysis is based on the assumption that explicit decomposition cannot occur without instances of *discussed and implemented* decomposition. It is unrealistic that a team could fully decompose a problem at the beginning of the design process without discussing that decomposition. On the other hand, implicit decomposition could occur with or without evidence of "discussed and implemented" decomposition: Team 2, for example, clearly implemented a decomposition but did not discuss it, suggesting they used implicit decomposition or some other unknown strategy.

A few of the 20 coded segments stand out as examples of the *discussed and implemented* case, where discussion of a decomposition strategy was quite clearly followed by actions consistent with their discussion (marked with triangles in Figure 1).

- Team 3, line 2: At minute 98, a team member proposed that the team build a single manufacturing cell so that they can define the space that will be required for a single cell. The team member then said that this will allow them to make copies of the cell, placing as many into the factory layout as will be necessary. After this point, the team proceeded to design a single cell for replication and then created multiple copies of the cell as planned. The Team 3 timeline in Figure 1 shows a decomposition code followed by the time devoted to the "cell design" subproblem (minute 98).
- Team 4, line 2: At minute 26, a team member suggested that the team choose whether to continue planning as a group or to pursue individual designs for later comparison. The team decided to work on multiple designs for later reconciliation and then proceeded to do so. Roughly twenty minutes later, the team resumed working together and discussed the merits of the different designs. The timeline in Figure 1 shows that the "high-level logic" and "flow" subproblems began at that point (minute 50).
- Team 4, line 4: At minute 54, the team outlined a plan to create a top-level design and work their way down to the details. The timeline in Figure 1 shows that the comment occurred at the beginning of the "high-level logic" and "flow" subproblems.
- Team 4, line 10: At minute 126, a team member proposed that the team first work on developing an overall plan for the locations of several of the larger stand-alone elements and then focus on the assembly areas. The timeline in Figure 1 shows that, after this comment, the team worked on the "block locations" subproblem and then worked on the "cell design" subproblem.

This evidence shows that Teams 3 and 4 sometimes *discussed and implemented* their decomposition, which directed at least part of their work. In the case of Team 4, discussions of decomposition preceded and explained nearly all of their work on subproblems, suggesting that their discussions of decomposition directed nearly all of their work. In the case of Team 3, one discussion preceded the critical choice to design cells before other elements, but their later choices to work on different subproblems do not appear driven by discussions of decompositions (*undiscussed yet implemented* decomposition, the third case above). Thus, discussions of decomposition influenced one, but not all, of their choices about decomposition.

The evidence also suggests that Team 1 did not *discuss and implement* decomposition to the same extent and that Team 2 did not discuss decomposition strategies at all. While Team 1 had some discussions about decomposing the problem, their actual choices about the order of subproblems to solve were not consistent with these discussions (*discussed yet not implemented* decomposition, the second case above). Team 2 did not have discussions about decomposing the problem, yet nevertheless their actions showed evidence of decomposition (*undiscussed yet implemented* decomposition).

This evidence thus suggests that discussion of decomposition is not necessary for teams to decompose a problem. In fact, only one of the four teams appears to have discussed all aspects of the decomposition they ultimately employed. Future research must investigate how they decompose when they do not discuss doing so.

In terms of the timing of discussions, the evidence indicates that some discussions of decomposition immediately preceded the beginning of work using that decomposition, but other discussions of decomposition occurred at other times and were not clearly linked to decomposition choices. For instance, Teams 1 and 4 discussed decomposition options while exploring the problem.

The next question is whether these teams utilized explicit or implicit decomposition. Assuming, as discussed above, that explicit decomposition cannot occur without evidence of *discussed and implemented* decomposition, the evidence suggests that *none* of the teams utilized explicit decomposition. Teams' discussions of decomposition did not necessarily lead to a determination of a "complete" decomposition, meaning that some aspects of the order and composition of subproblems remained to be determined. Team 4 provides a good example. Their discussion at minute 54 determined that they would start with a top-level design but did not specify what would happen after that. They later had a discussion at minute 126 that settled some of these details. Thus, they did not decompose the entire problem up-front. As mentioned above, discussion of decomposition instead immediately preceded the work utilizing that decomposition.

This suggests the need to treat explicit and implicit decomposition not as two discrete modes but as two ends of a spectrum of possible design processes. Fully explicit decomposition requires a problem to be entirely decomposed up-front. Moving down the spectrum towards more implicit strategies involves increasingly smaller sub-problems being determined as the design progresses. On this spectrum, Team 4's process was the closest to explicit decomposition because they had two major discussions that identified multiple subproblems that needed to be solved. Similarly, Team 3 held a single discussion which identified multiple subproblems. The remaining teams do not provide enough evidence to determine their decomposition strategies based on discussion alone (this is left to future research).

Thus, the evidence from this study suggests that teams rarely employ fully explicit decomposition strategies. These findings are similar to those of Liikkanen and Perttula (2009) for individual designers. Our analysis also suggests the need for a spectrum of strategies ranging from explicit to implicit. It is not yet clear which strategies are more effective, nor why each team employed various strategies. An additional question remaining to be answered is how teams decide on a decomposition when they do *not* discuss it. This question will be investigated in future research.

5 SUMMARY AND CONCLUSIONS

This paper described the results of an empirical study of four teams of engineers who were presented with the problem of redesigning a factory layout. The goal of the study was to gain insights into whether and when teams discuss and select a decomposition of the problem and whether they utilize explicit or implicit decomposition in doing so.

The evidence from this study suggests that, despite the need to coordinate activities among the members of the team, some teams began working on some subproblems without discussing what to do next, while some teams discussed their options and selected them. Thus, it is clear that some design processes include undiscussed yet implemented decompositions.

Two of the four teams discussed their decompositions to some extent, but these decompositions were not complete, explicit determinations of the order and composition of the subproblems. In addition, discussions generally directly preceded work on the discussed subproblems, which is closer to implicit decomposition strategies. Our analysis suggests the need to consider a spectrum of strategies ranging from explicit to implicit.

These preliminary results motivate future work to examine the decomposition decision in more detail, determine if other teams behave similarly, investigate how teams informally determine which subproblem to solve next in the design process, and generate more insights based on these observations. When does discussing the decomposition affect the subsequent team decision making, design activities, and solution quality? When are teams consistent with the decompositions that they discuss and select? When (and how) do teams select explicit decompositions, partial decompositions, and implicit decompositions? How does a team decompose a design problem without discussing it? Would requiring teams to discuss their decomposition and problem selections affect their actions and results?

These results increase our understanding of how and when engineering teams decompose design problems and how the decomposition affects the quality of the solutions that are generated. This line of research will eventually lead to a broad taxonomy of decompositions across various types of design problems and a more general understanding of their impact on the quality and characteristics of design solutions, from which we can expand our theories of design, support the design of better design processes, and improve our ability to organize and manage effective engineering design teams.

ACKNOWLEDGEMENTS

The authors acknowledge the assistance of David Rizzardo, who organized and led the facility design course and evaluated the teams' facility layouts. The authors are supported by National Science Foundation Grant CMMI1435074.

REFERENCES

Ahmed, S., Wallace, K.M., and Blessing, L. (2003) Understanding the Differences Between How Novice and Experienced Designers Approach Design Tasks. Research in Engineering Design, Vol. 14, pp. 1-11.

Austin-Breneman, J., Honda, T., and Yang, M.C. (2012) A Study of Student Design Team Behaviors in Complex System Design. Journal of Mechanical Design, 134(12), 124504 (7 pages).

Ball, L.J., Evans, J.S.B.T., and Dennis, I. (1994) Cognitive Processes in Engineering Design: A Longitudinal Study," Ergonomics, Vol. 37, No. 11, pp. 1753-1786.

Ball, L.J., Evans, J.S.B.T., Dennis, I., and Ormerod, T.C. (1997) Problem-solving Strategies and Expertise in Engineering Design. Thinking and Reasoning, Vol. 3, No. 4, pp. 247-270.

Cagan, J., Leifer, L., Smith, S.M., Dinar, M., Shah, J.J., Linsey, J., and Vargas-Hernandez, N. (2013) Empirical Studies of Design Thinking: Past, Present, Future. DETC2013-13302, Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, IDETC/CIE 2013, Portland, Oregon, August 4-7, 2013.

Corbin, J., and Strauss, A. (2008) Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, third edition. Sage, Thousand Oaks, California.

Emerson, R., Fretz, R., and Shaw, L. (1995) Writing Ethnographic Fieldnotes. University of Chicago Press, Chicago.

Ericsson, K.A., and Simon, H.A. (1993) Protocol Analysis; Verbal Reports as Data – Revised Edition. MIT Press, Cambridge, Massachusetts.

Glaser, B. and Strauss, A. (1967) The Discovery of Grounded Theory. Aldine de Gruyter, New York.

Gralla, E.L. (2012) Human and Modeling Approaches for Humanitarian Transportation Planning. Ph.D. dissertation, Massachusetts Institute of Technology.

Gralla, E.L., and Herrmann, J.W. (2014) Design Team Decision Processes in Facility Design. Proceedings of the 2014 Industrial and Systems Engineering Research Conference, Y. Guan and H. Liao, eds., Montreal, Canada, June 1-3, 2014.

Hendrick, H.W., and Kleiner, B.M. (2002) Macroergonomics: Theory, Methods, and Applications. Lawrence Erlbaum Associates, Mahwah, New Jersey.

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

Hong, L., and Page, S.E. (2004) Groups of diverse problem solvesr can outperform groups of high-ability problem solvers. Proceedings of the National Academy of Sciences of the United States of America, Vol. 101, No. 46, pp. 16385-16389.

Langley, A. (1999) Strategies for theorizing from process data. Academy of Management Review, Vol. 24, No. 4, pp. 691–710.

LiCalzi, M., and Surucu, O. (2012) The power of diversity over large solution spaces. Management Science, Vol. 58, No. 7, pp. 1408-1421.

Liikkanen, L.A., and Perttula, M. (2009) Exploring problem decomposition in conceptual design among novice designers. Design Studies, Vol. 30, pp. 38-59.

Lipshitz, R., Klein, G., Orasanu, J., and Salas, E. (2001) Focus Article: Taking Stock of Naturalistic Decision Making. Journal of Behavioral Decision Making, Vol. 14, pp. 331-352.

March, J., and Simon, H. (1993) Organizations, second edition. Blackwell, Cambridge, Massachusetts.

Lewis, K., and F. Mistree (1998) Collaborative, Sequential, and Isolated Decisions in Design. Journal of Mechanical Design, Vol. 120, No. 4, pp. 643-652.

Pettigrew, A.M. (1990) Longitudinal field research on change: Theory and practice. Organization Science, Vol. 1, No. 3, pp. 267–292.

Poole, M.S., and Holmes, M.E. (1995) Decision Development in Computer-Assisted Group Decision Making. Human Communication Research, Vol. 22, No. 1, pp. 90-127.

Schmidt, L.C., Schmidt, J.A., Smith, P.E., Bigio, D.I., and Contardo, J.B. (2005) BESTEAMS: Building Engineering Student Team Effectiveness and Management Systems. College House Enterprises, Knoxville, Tennessee.

Spradley, J.P. (1980) Participant Observation. Holt, Rinehart and Winston, New York.

Woods, D.D. (1993) Process-tracing methods for the study of cognition outside of the experimental laboratory. in Klein, G., Orasanu, J., Calderwood, R., and Zsambok, C. E., editors, Decision Making in Action: Models and Methods, pp. 228–251, Ablex.