

# COMPARING DESIGNING ACROSS DIFFERENT DOMAINS: AN EXPLORATORY CASE STUDY

Jeff WT Kan<sup>1,2</sup>, John S Gero<sup>2</sup>

(1) Taylor's University, Malaysia (2) Krasnow Institute for Advanced Study, Fairfax, USA

## ABSTRACT

This paper presents the results of comparing designing across different domains. It is claimed that designing involves ontological issues and processes that can be observed and appear across all domains in contrast to the belief that designing is unique to its domain. A case study of designing in three different domains, architectural design, software design, and mechanical design, was conducted. A concise qualitative analysis of each session is presented before reporting results from quantitative measurement of the design issues and design processes in the three design sessions. Differences were found in terms of design issues and design processes in the three design sessions. The results from this case study support the claim that an ontologically based approach allows for comparisons of designing across multiple domains.

*Keywords: Design ontology, protocol analysis, linkography*

## 1 INTRODUCTION

Is designing unique to its domain of practice or is designing a general activity practiced in multiple domains? There have been numerous design cognition studies of designers within their own domain. These studies have examined a wide range of variables but have kept the domain fixed. There have been very few studies comparing the design cognition of designers from different domains.

One of the more commonly used methods to study design cognition is protocol analysis [1] [2]. Whilst protocol analysis has been used extensively to study designing there has been a lack of comparability of results from different studies due to the use of different coding schemes applied by different researchers [3]. The Design Thinking Research Symposium 2 [4] and 7 [5] are examples of researchers using the same data set but with different kinds of models and analysis methods, producing a rich and wide range of results. However, the differences in methods make it difficult if not impossible to compare and validate results from various researchers. The extant studies of designers from various domains have produced results that cannot be compared.

Kan [6] used a framework of methods to study design protocols quantitatively and uniformly. Methods based on a design ontology [7], sequential analysis [8], linkography [9], and information theory [10] were developed. The combinations of these methods quantify designing in terms of issues, processes and idea development opportunities. These methods of analyzing design protocols are laborious. A tool [11] has been developed to automate all the calculational methods of the analysis. This paper uses this generic tool to compare and contrast designing in three different design domains. The analysis methods are independent of design domain, number of designers, expertise of designers, whether tools are used and what kinds of tools are used [3] [11].

The rest of the paper presents the results of analyzing protocols from cases in the three different domains, software design, architectural design and mechanical design, to determine whether comparisons across domains are feasible.

## 2 THE THREE DESIGN SESSIONS

The three different design domains of architectural design, software design and mechanical design have been studied using generic methods that allow for comparisons. This experiment is a case study of three design sessions, one for each domain. All the three sessions were in-vitro design sessions that involved two collocated participants collaborating to design for 30 to 60 minutes. The architectural design session was about designing an art gallery. The mechanical design session was about designing a device for elderly and disabled to open and close a window. The software design session was about

designing a traffic flow simulation program. Section 2.1 presents a brief qualitative overview of each session.

## **2.1 Qualitative Analysis of the three sessions**

The participants in the software and architectural design sessions were experts while the participants of the mechanical session were novices. In the experts' sessions, participants spent less time in studying the brief/requirements while in the novice session participants spent more time in focusing on the brief. Participants in both the software and mechanical design sessions used a whiteboard to draw and document their design. The participants in the architectural design session used tracing paper to draw, sketch and document their design.

### **2.1.1 Architectural Design Session**

In the architectural design session, the participants were one senior architect and one architect. The senior architect took the leadership role, made most of the decisions, and produced most of the sketches. Observing the session, it can be divided into four parts. In the first part they dealt with the brief and site (about 10 per cent of the time of the session). In the second part they analyzed, planned and developed concepts in the plan (about 30 per cent of the time); issues like location of main entrance and service entrance, an icon to capture attention were discussed. In the third part they developed the 3D form in elevation (about 30 per cent of the time); ideas like "ribbon", "hole in the middle" were suggested. In the final part they worked on the layout and calculated the required areas until the end of the session (about 30 per cent of the time).

### **2.1.2 Software Design Session**

In the software design session, the two participants were both experienced designers. This session can be divided into three parts. In the first part they studied the brief individually without verbalizing (about 8 per cent of the time of the session). In the second part they tried to understand the required simulation by drawing an analogy from real world traffic (about 42 per cent of the time). Different objects and variables were discussed such as cars, signals, and intersections. In the last part, they focused on the program structure until the end of the session (about 50 per cent of the time). Participant 1 seemed to take the leadership role and did most of, if not all, the drawing and writing on the board.

### **2.1.3 Mechanical Design Session**

In the mechanical design session, the participants were undergraduate engineering students. They were given access to a specific website to provide them the necessary information regarding the requirements. The division of activities of this session is not as clear as the two other sessions, however it can be divided into three parts; in the first part the major activities were to learn and understand the problem which was interwoven with the proposing of solutions (about 20 per cent of the time of the session). In the second part, they developed their ideas of crank system, pulley and gears which were intertwined with the learning aspect (about 62 per cent of the time). In the last part they tried to document their ideas but prompted design activities as well (about 18 per cent of the time).

## **3 QUANTITATIVE MEASUREMENTS**

### **3.1 Design Protocol**

Protocol analysis uses verbal protocols and concurrent reporting as qualitative data from which quantitative data is derived for studying thought processes. In the standard procedure of studying design using protocol analysis there are five steps: conducting experiments that produce the protocols, transcribing protocols, parsing the transcription into segments, encoding segments according to a coding scheme, and interpreting the encoded protocols [12]. The experiment, in the first step, is derived from the research aim and method, which also delineates the coding scheme. However, there are theories about designing that assume the fundamental issues and processes involved in designing are the same, unrelated to what is being designed and the environment within which designing occurs [13] [7]. Within a theoretical framework, in a typical protocol analysis study, design issues and design processes are usually encoded separately. In this method, instead of encoding processes we only

encode design issues and use additional steps to derive design processes from the design issues. The additional steps are: construction of a linkograph and consequentially deriving design processes from the coded segments and links.

### 3.2 FBS Design Ontology Based Coding Scheme

We study designers' thinking in terms of the FBS ontology, Figure 1 [7]. In this ontology there are fundamentally only five design issues a designer focuses on: function or purpose (F); expected behavior of the resulting design (Be); behavior derived from the proposed design (Bs); a mental representation of the design itself called structure (S) and an external representation or description of structure (D). In addition there is the issue that comes from the client in the form of requirements (R). Designing commences with requirements (R) and ends with design descriptions (D). The transformations between design issues are the design processes, which are not separate from and are a consequence of the design issues. There are 8 design processes from the design issues. Formulation is the design process of transforming requirements (R) into functions (F) and transforming functions into expected behaviors (Be). Synthesis is the process of transforming expected behaviors into structure or form (S). Analysis is the process of deriving the actual behavior (Bs) from the structure, and evaluation is the process of comparing the expected behavior and analyzed behavior. Documentation is the process of externalizing the design issues into drawings or other kinds of descriptions. There are three reformulation processes of introducing new structures, new behaviors and new functions, Figure 1.

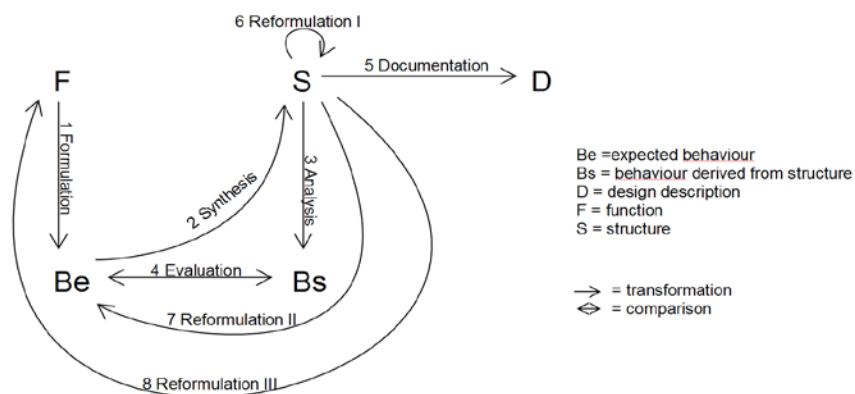


Figure 1. The FBS ontology of fundamental design issues and processes

Protocols are segmented and coded as design issues according to this framework. Unlike other protocol analyses there is an isomorphism between segments and design issues and hence their codes. Examples of some of the segments and design issues are presented in Section 4.2, which is an excerpt from the architectural design session

### 3.3 Design Issues Software Design

In designing physical objects, structure design issues (S) usually resemble some physical aspect. However, in the design of software the structure (S) will not have any physicality. In a typical object-oriented program, we refer to the objects as structure design issues. The software designers formulate the expected behaviors from the functions of these objects. With these expected behaviors they can synthesize the codes or the relationships of codes of those objects. With these objects they can derive their behaviors by either running that part of the program or mentally simulating their behaviors. For example in this set of data, the input, roads and intersections are objects that were considered as part of the (program) structure (S). During designing there were discussions concerning the expected behavior such as: drag and drop environment of the input; signal timing of the intersections. An example of behavior derived from structure is the time for cars to travel between intersections.

## 4 CASE STUDY RESULTS AND ANALYSIS

The architectural design session lasted for 30 minutes and had 357 design issues. In the 58 minutes of the software design session, 602 design issues were identified and segmented. The mechanical design session last about an hour and 15 minutes and contained 236 design issues. The segments and codes of the software and mechanical sessions were arbitrated by two researchers with inter-coder agreement

above 80% while the architectural session was self-arbitrated with agreement of over 90%. The segments were of different length of time and varied in the number of words. Table 1 shows the average number of words, and their standard deviations, per segment of the three sessions.

Table 1. The average number of words in segment of the three sessions

	<b>Architectural Design</b>	<b>Software Design</b>	<b>Mechanical Design</b>
Average no. of words in segment	15.85	10.31	13.34
Standard deviations	14.50	7.27	10.02

#### 4.1 Design Issues

Table 2 and Figure 2 show the distributions of design issues in the three sessions. Two tailed pair-t tests were used to determine whether the distributions differ from each other in a significant way. The results show the distributions of the independent design issues of the three sessions are very different; the pair-t test returns  $p > 0.99$  for any combinations of the two sessions.

In this case, the architects were more concerned with the function and requirement issues of the design than the other two design domains. The software designers were more concerned with behavior than structure while the mechanical and architectural designers had approximately equal concern for structure issues. The software designers had little consideration of the functional issues; this could due to in this particular software design exercise, they emphasized on the data modeling part of the software. They were more concerned with the behavior of the objects in their data model rather than the functions of those objects.

Table 2. The percentage distribution of design issues of the three sessions

<b>Issue Distribution (%)</b>	<b>Architectural Design</b>	<b>Software Design</b>	<b>Mechanical Design</b>
Requirement (R)	5.1	2.8	3.1
Function (F)	6.8	0.5	1.8
Expected Behavior (Be)	19.2	31.1	10.7
Behavior from Structure (Bs)	13.5	18.8	31.3
Structure (S)	42.5	30.8	42.9
Description (D)	13	16	10.1

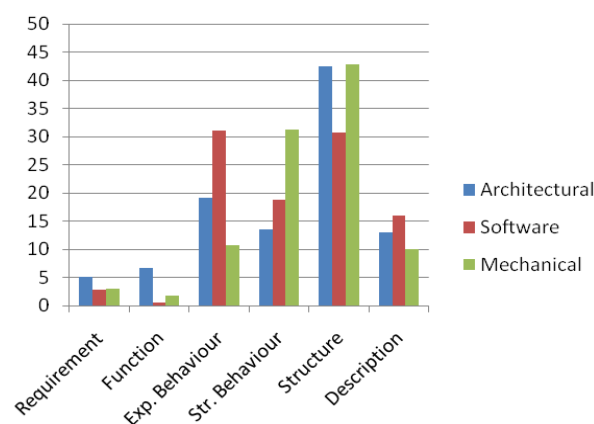


Figure 2. The distributions of design issues for the three sessions

#### 4.2 Variation of Design Issues during Session

From the qualitative observations some issues cluster more at the beginning of a session, for example function and requirement issues. Figure 3 counts the design issues with a dynamic 30 segment window across the whole session [11]. It indicates the issues they were focusing on during different times in the design session. With a design ontology that starts with design requirements and ends with design descriptions, one would expect a pattern with more function and requirement issues at the beginning of a session and more structure and description at the end. This is the case for the first 50 segments of the

architectural design session as can be observed in Figure 3(a). Software designers appear to be only concerned with function issues at the beginning of the session, Figure 3(b). The mechanical designers, Figure 3(c) had requirements and function issues over the first two thirds of the session.

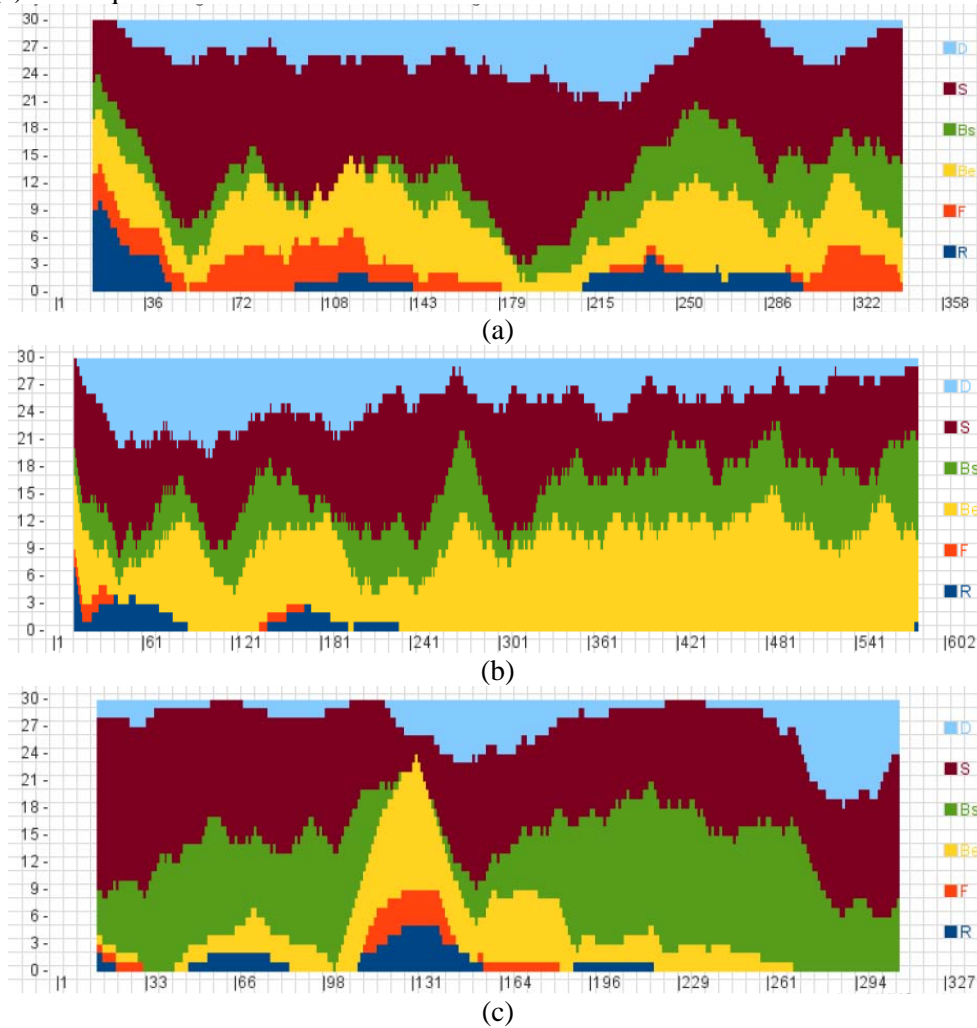


Figure 3. Dynamic distributions of design issues of (a) architectural, (b) software and (c) mechanical design session

#### 4.2 Design Processes Measurement: Derived from Sequential Analysis

Design protocols have been analyzed by statistical methods that often contain the assumption that each segment is an independent event. Markov chains describe the probability of one event leading to another. Kan [6] used Markov analysis to examine the sequence of occurrence of the FBS design issues. If we consider the sequence of design issues in Figure 4, there is a high chance that the move from one design issue to its consecutive design issue represents a design process. For example moving from segment 50 to 51 is the formulation process. Moving from segment 51 to 52 is the synthesis process. Gero et al [11] called this kind of sequence a “syntactic link” that produces a weak model of designing. Figure 5 shows the distributions of these syntactic processes. Details of the method can be found in [14]. It is claimed that this weak design process model provides additional information about the characteristics of a design session. Equations 1, 2, and 3 are the Markov probability matrices of the three design sessions.

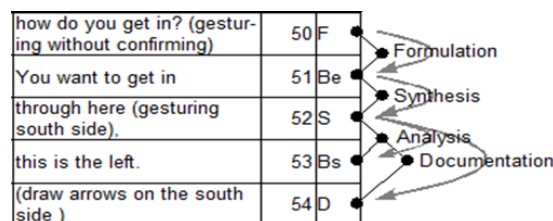


Figure 4. Example of sequence of design issues, construction of linkograph, and deriving

design processes from links and design issues

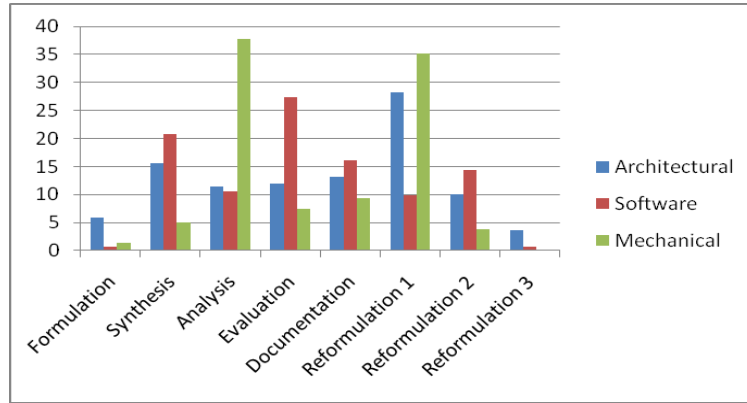


Figure 5. Percentage distributions of syntactic design processes of the three sessions

In the architectural design session, the highest probability of moving from one issue to another is found in equation 1 as 0.54 and is from description to structure. This matches the understanding of how architects design [15] [16] [17], they interpret their sketches as they design.

In the software design session, equation (2), it can be observed that the probabilities of moving to function related design issues are very small. The probabilities of design issues after a design description are split between expected behavior (0.41) and structure (0.43). The highest probability is from function to expected behavior design issued (0.67), which indicates a formulation process. The probability from expected behavior to structure is also high (0.52), which implies a synthesis event.

For the mechanical design session, equation (3), the probability of moving from description to structure is the highest, which is similar to the architectural design session. The next highest move probability is from structure to structure behavior and reverse (0.43 and 0.52).

Comparing equations (1), (2), and (3) we can observe the distribution of high and low probabilities cluster in different part of the matrices.

$$P_{archit} = \begin{pmatrix} & R & F & Be & Bs & S & D \\ R & 0.28 & 0.17 & 0.00 & 0.11 & 0.33 & 0.11 \\ F & 0.12 & 0.25 & 0.54 & 0.00 & 0.04 & 0.04 \\ Be & 0.06 & 0.03 & 0.13 & 0.12 & 0.52 & 0.14 \\ Bs & 0.02 & 0.02 & 0.41 & 0.10 & 0.41 & 0.04 \\ S & 0.03 & 0.05 & 0.15 & 0.17 & 0.41 & 0.19 \\ D & 0.02 & 0.09 & 0.13 & 0.17 & 0.54 & 0.04 \end{pmatrix} \quad (1)$$

$$P_{soft} = \begin{pmatrix} & R & F & Be & Bs & S & D \\ R & 0.29 & 0.00 & 0.12 & 0.00 & 0.41 & 0.18 \\ F & 0.00 & 0.00 & 0.67 & 0.00 & 0.00 & 0.33 \\ Be & 0.01 & 0.01 & 0.22 & 0.23 & 0.39 & 0.15 \\ Bs & 0.03 & 0.00 & 0.46 & 0.18 & 0.27 & 0.06 \\ S & 0.03 & 0.01 & 0.27 & 0.20 & 0.18 & 0.30 \\ D & 0.02 & 0.00 & 0.41 & 0.14 & 0.43 & 0.01 \end{pmatrix} \quad (2)$$

$$P_{mech} = \begin{pmatrix} & R & F & Be & Bs & S & D \\ R & 0.10 & 0.10 & 0.10 & 0.10 & 0.40 & 0.20 \\ F & 0.17 & 0.33 & 0.33 & 0.00 & 0.00 & 0.17 \\ Be & 0.06 & 0.03 & 0.43 & 0.14 & 0.23 & 0.11 \\ Bs & 0.02 & 0.01 & 0.07 & 0.33 & 0.52 & 0.05 \\ S & 0.02 & 0.00 & 0.04 & 0.43 & 0.40 & 0.11 \\ D & 0.03 & 0.03 & 0.12 & 0.09 & 0.55 & 0.18 \end{pmatrix} \quad (3)$$

Figure 6 shows the dynamic distributions of the syntactic design processes of the three sessions with a window of 30 segments. The distributions of syntactic processes of the three sessions are different. In the architectural design session formulation appears throughout the session whereas it only appears at the beginning of the software and mechanical session.

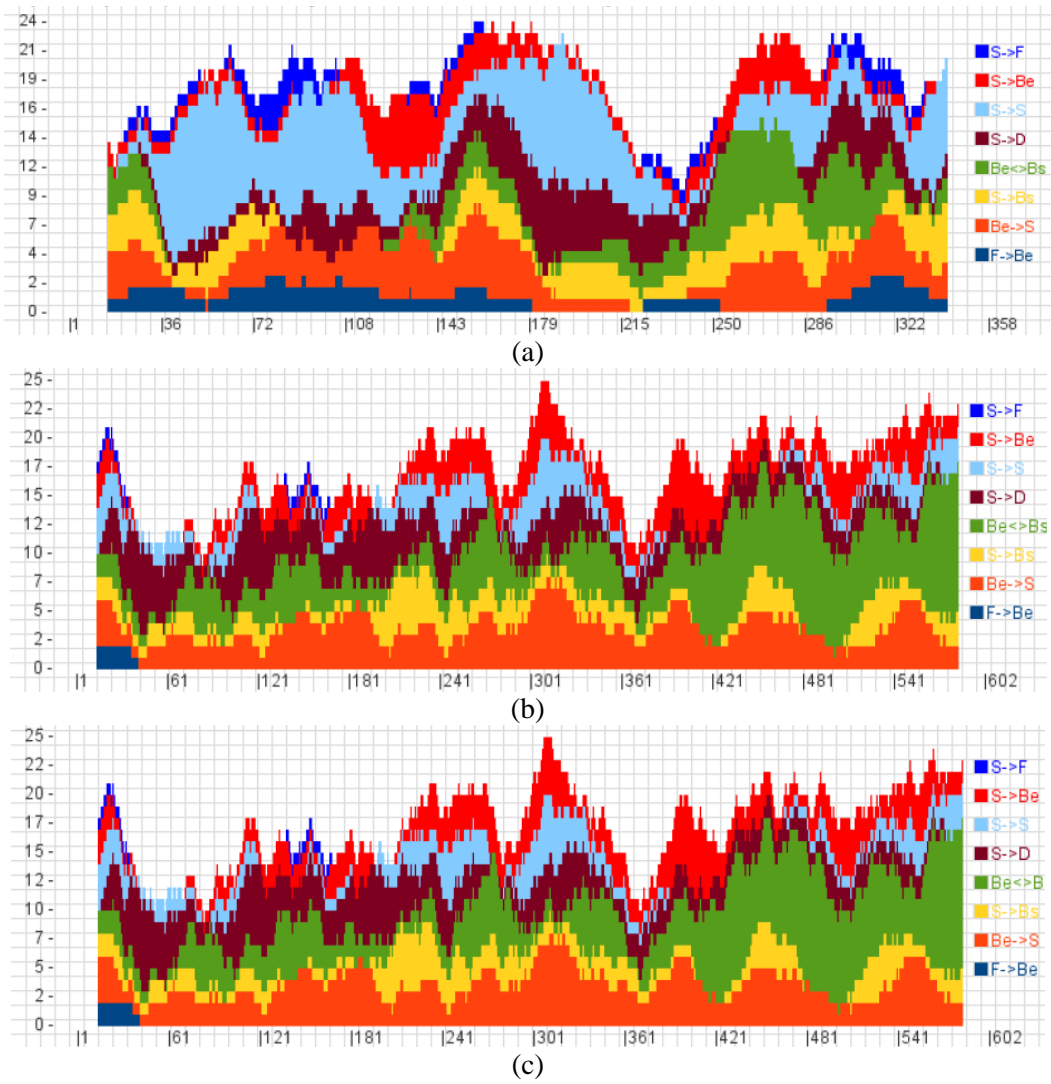


Figure 6. Dynamic distributions of syntactic design processes of (a) architectural, (b) software and (c) mechanical design sessions

The mean first passage time is the average number of steps traversed before reaching a design issue from other issues. Equations (4), (5) and (6) are the normalized mean first passage time matrices of the three design sessions. In these equations, the number of steps equals the total number of segments times the individual elements in the matrix. For example the shortest passage in the architectural design session is from Be to S and the mean passage time is 2.00 ( $357 * 0.56/100$ ). The shortest passage in the software design session is from F to Be and the mean passage time is 1.98 ( $601 * .33/100$ ); and the shortest passage in from Bs to S and the mean passage time is 1.89 ( $326 * .58/100$ ). Table 3 shows the design processes associated with the five shortest mean first passage times of the three sessions in descending order. The possible design processes associated with these passages are shown in the brackets. Table 3 suggests the architectural design processes, in this session, are based heavily on synthesizing and reformulating together with formulating expected behaviors. Software design, in this session, involves, in order of importance, formulation, evaluation, type one reformulation and synthesis. Mechanical design, in this session, involves mainly type 1 reformulation and analysis.

$$M_{\text{nor}(357)\text{arch}} = \begin{pmatrix} R & 4.65 & 4.06 & 1.18 & 1.99 & 0.87 & 5.73 \\ F & 8.61 & 3.91 & 0.68 & 3.69 & 0.92 & 2.51 \\ Be & 8.25 & 6.29 & 1.44 & 2.56 & 0.56 & 3.09 \\ Bs & 7.93 & 6.06 & 1.01 & 2.04 & 0.73 & 3.76 \\ S & 9.54 & 5.91 & 1.69 & 2.67 & 0.67 & 2.35 \\ D & 10.08 & 6.27 & 1.77 & 2.25 & 0.63 & 1.97 \end{pmatrix} \quad (4)$$

$$M_{\text{hor}(601)\text{soft}} = \begin{pmatrix} & R & F & Be & Bs & S & D \\ R & 6.20 & 10.00 & 0.73 & 1.52 & 0.38 & 2.07 \\ F & 0.89 & 11.06 & 0.33 & 0.61 & 0.72 & 0.89 \\ Be & 22.37 & 7.51 & 0.53 & 0.81 & 0.51 & 0.96 \\ Bs & 21.90 & 8.85 & 0.46 & 0.87 & 0.51 & 1.06 \\ S & 22.70 & 8.87 & 0.56 & 0.88 & 0.54 & 0.88 \\ D & 19.26 & 11.20 & 0.55 & 0.94 & 0.48 & 0.94 \end{pmatrix} \quad (5)$$

$$M_{\text{hor}(326)\text{mech}} = \begin{pmatrix} & R & F & Be & Bs & S & D \\ R & 9.92 & 8.42 & 2.02 & 3.83 & 1.96 & 2.70 \\ F & 4.86 & 9.39 & 2.61 & 5.57 & 3.48 & 1.69 \\ Be & 6.34 & 7.07 & 2.17 & 3.57 & 2.45 & 2.82 \\ Bs & 7.00 & 15.39 & 3.45 & 0.95 & 0.58 & 5.04 \\ S & 9.04 & 14.10 & 3.54 & 1.32 & 0.72 & 3.94 \\ D & 9.95 & 12.03 & 3.31 & 2.10 & 0.78 & 2.89 \end{pmatrix} \quad (6)$$

Table 3. Design processes associated with the five shortest mean first passage times of the three sessions in descending order

Architectural Design	Software Design	Mechanical Design
Be>S (Synthesis)	F>Be (Formulation)	Bs>S
D>S (Reformulation I)	R>S	S>S (Reformulation I)
S>S (Reformulation I)	Bs>Be (Evaluation)	D>S (Reformulation I)
F>Be (Formulation)	D>S (Reformulation I)	Bs>Bs
Bs>S	Be>S (Synthesis)	S>Bs (Analysis)

### 4.3 Design Processes Measurement: Derived from Linkograph

Linkography was first introduced to protocol analysis by Goldschmidt [9] to assess the design productivity of designers. Like other protocol studies, protocols are segmented according to intentions and a linkograph is constructed by linking related segments. The links are established by discerning, using domain knowledge, whether a segment is semantically connected to any previous segments. Kan [6] used the ontological coding scheme and linkograph to derive ontological design processes, called semantic design processes or just design processes. A constructed linkograph and its relation to the design issues are shown in Figure 4. The dots outside the segments represent links between segments. The arrows represent the derived design processes that maps onto the FBS ontology. Only four (formulation, synthesis, analysis and documentation) of the eight possible design processes are in Figure 4.

Figure 7 compares the percentage distributions of design processes derived from the semantic links of the three design sessions. Of the 601 segment of the software design session, only the first 130 segments were studied with linkography. The results generally are in agreement with the Markov analysis of the mean first passage time for these sessions: comparatively type 1 reformulation is high in both architectural design and mechanical design sessions; evaluation is high in the software design session. However, Figure 7 also shows additional information – type 2 reformulation is high for the software design session.

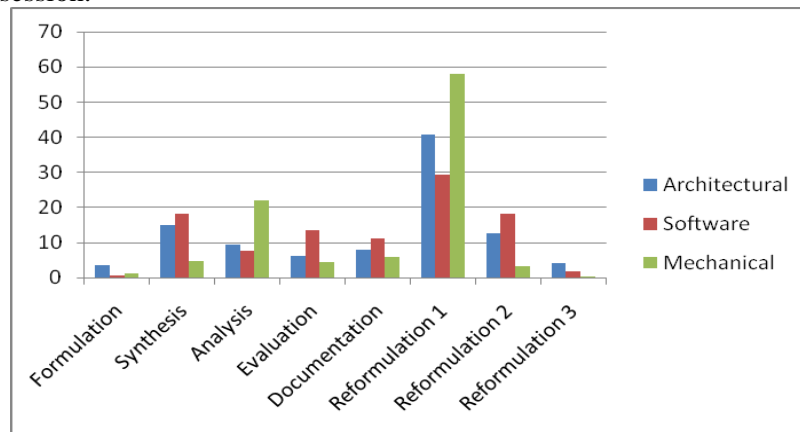


Figure 7. Percentage distributions of semantic design processes of the three sessions



Figure 8 shows the dynamic design processes of the three sessions. It shows the intensity of different design processes across the design session.

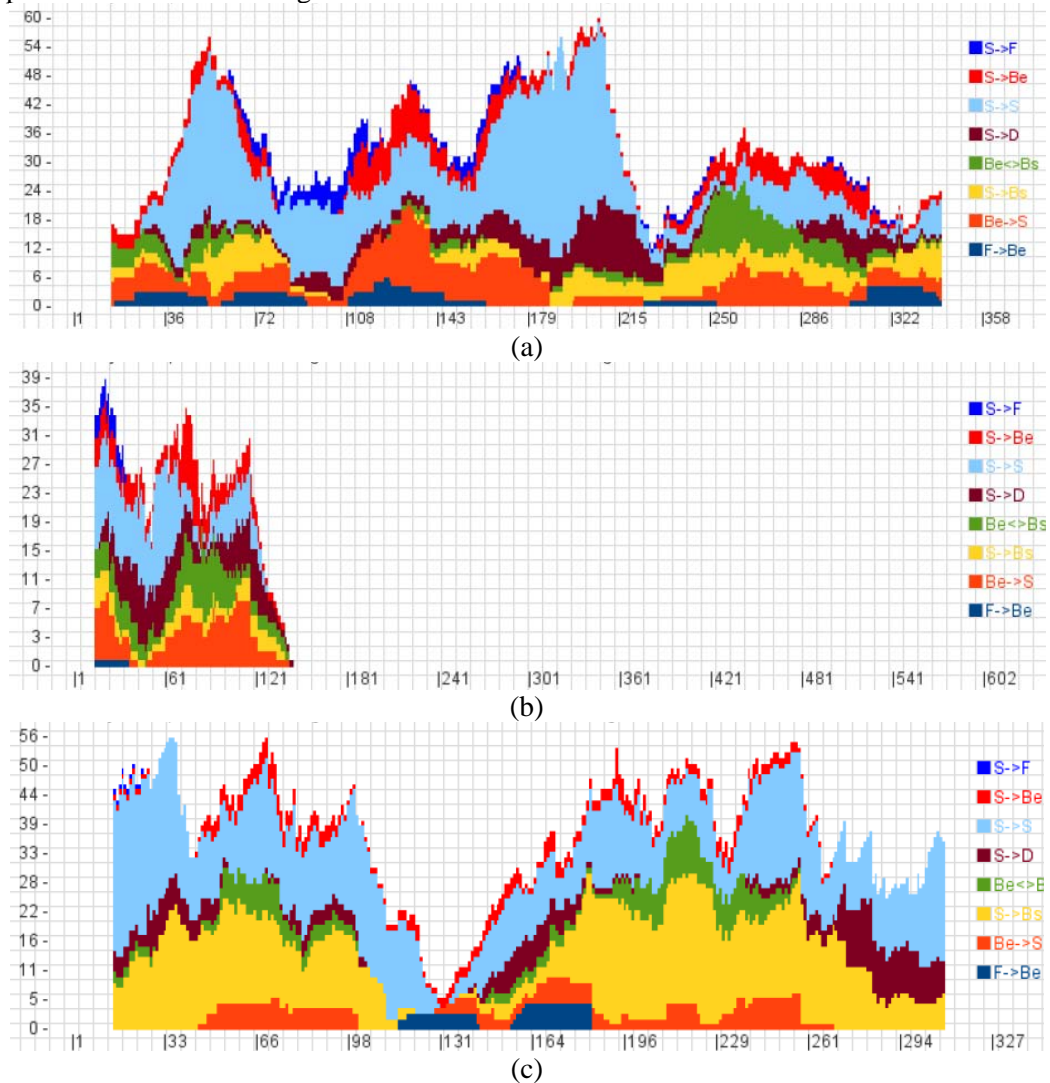


Figure 8. Dynamic distributions of semantic design processes of (a) architectural, (b) software and (c) mechanical design sessions

## 5 CONCLUSIONS

In this case study the software designers spent most of the time on behavior issues of the software being developed, while the architectural and mechanical designers focused more on structure issues. In the sequential analysis, the probability matrices also show the high probability events cluster around behavior issues in the software design session and in the architectural design and mechanical design sessions they cluster around structure issues. The mean first passage time shows the expectancy of different design processes. The expectancy of synthesis and type 1 reformulation is high in the architectural design session while the expectancy of formulation and evaluation is high in the software design session. In the mechanical design session, possibly because they were students, the expectancy of design processes is lower than the other two sessions. However, type 1 reformulation and analysis have high expectancy. The results from these cases will have to be verified using a statistically significant sample from each domain to determine if they are generalizable.

This case study has shown that it is feasible to study designing across domains. Utilizing a generic coding methodology and a standardized set of measurements provide the foundation for statistically robust comparisons, notwithstanding the high initial cost of carrying out the segmentation and coding.

## ACKNOWLEDGEMENTS

This research is supported in part by a grant from the US National Science Foundation grant numbers CMMI-0926908 and IIS-1002079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

- [1] Cross N., Christiaans H., and Dorst K. *Introduction: the Delft protocols workshop*, pp. 1–14, 1996, (John Wiley & Son)
- [2] Ericsson K.A. and Simon H.A. *Protocol Analysis: Verbal Reports as Data*. revised edition 1993, (The MIT Press)
- [3] Gero J.S. Generalizing design cognition research. In *Design Thinking Research Symposium 8*, 2010 (University of Technology, Sydney)
- [4] Cross N., Christiaans H., and Dorst K., eds. *Analysing Design Activity* 1996, (John Wiley & Sons)
- [5] McDonnell J. and Lloyd P., eds. *About: Designing. Analysing Design Meetings* 2009.(CRC Press)
- [6] Kan W.T. *Quantitative Methods for Studying Design Protocols*. Ph.D. thesis, The University of Sydney, Faculty of Architecture, Design & Planning, Key Centre of Design Computing and Cognition, 2008.
- [7] Gero J.S. Design prototypes: a knowledge representation schema for design. *AI Magazine*, 1990. 11(4), 26–36.
- [8] Gottman J.M. and Roy A.K. *Sequential Analysis: A Guide for Behavioral Researchers*, 1983 (Cambridge University Press)
- [9] Goldschmidt G. Linkography: assessing design productivity. In *Cybernetics and System '90*, R. Trappl, ed., 1990 pp. 291–298.
- [10] Shannon C.E. A mathematical theory of communication. *The Bell System Technical Journal*, 1948. 27, 397–423.
- [11] Gero J.S., Kan J.W., and Pourmohamadi M. Analysing design protocols: Development of methods and tools. In *Research into Design, Research Publishing*, A. Chakrabarti, ed., 2011 pp. 3–10.
- [12] van Someren M.W., Barnard Y.F., and Sandberg J.A. *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Knowledge-Based Systems, 1994 (Academic Press)
- [13] Asimov M. *Introduction to design*, 1962 (Prentice Hall)
- [14] Kan J.W.T. and Gero J.S. A generic tool to study human design activity. In *Human Behavior in Design*, M. Norell Bergendahl, M. Grimheden, L. Leifer, P. Skogstad, and U. Lindemann, eds., 2009 pp. 123–134.
- [15] Suwa M. and Tversky B. What do architects and students perceive in their design sketches? a protocol analysis. *Design Studies*, 1997. 18(4), 385–403.
- [16] Suwa M., Purcell T., and Gero J.S. Unexpected discoveries and s-invention of design requirements: important vehicles for a design process. *Design Studies*, 2000. 21(6), 539–567.
- [17] Goldschmidt G. The backtalk of self-generated sketches. *Design Issue*, 2003. 19(1), 72–88.