Sixth International Conference on Construction in the 21st Century (CITC-VI)
"Construction Challenges in the New Decade"
July 5-7, 2011, Kuala Lumpur, Malaysia

THE APPLICABILITY OF THE DESIGN STRUCTURE MATRIX (DSM) METHOD IN REPRESENTING AND PRESCRIBING THE ARCHITECTURAL DESIGN PROCESS

Fahimeh Zaeri, Khairul Anwar Mohamed Khaidzir
Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
f.zaeri.arch@gmail.com, b-anwar@utm.my

Danfulani Babangida Idi
Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
maan2maan79@yahoo.com

Abstract
Iteration is a key feature in the design process. Utilizing the iteration loops prevalent in the design process can help designers explore and understand problems better. Furthermore, designers are be able clarify problems, generate ideas and gain the best solutions through these iterative loops. Understanding the characteristics of design iteration is fundamental to improving solution generating activities. Depending on different design situations, problem types, constraints and criteria, we may encounter different kinds of iteration, namely in terms of their positive or negative effects on the design process. So, identifying iteration characteristics will be crucial in the effective management of the design process. For this purpose, in this paper, we consider an explicit method to represent the clear and compact model of iterative processes that occur in architectural design. This method is derived from the Design Structure Matrix (DSM), which is a heuristic approach that enables us to capture activity iteration and probabilistic looping within the architectural design process. While this tool allows the modeling of sequential or parallel processes, it can address interdependencies (feedback and iteration) and many other complex activities.

Keywords: Iteration, Design Process, Management of the Design Process, Architectural Design, Design Structure Matrix

1.INTRODUCTION

There are many ways to describe a design. In fact, we cannot propose one single description of design, since design disciplines are continuously evolving and expanding into new dimensions (V.Margolin & R.Buchanan, 1998).

As Goel proposed "design as an activity has a distinct conceptual and cognitive realization from non-design is largely independent of the character of the thing designed" (P.Pirolli & V.Goel, 1989). Also, Thomas (1987) analyses communication as a design process.

But design as an architectural term has two levels of meaning:

‘*in a restricted sense it means a designation, the delineation in general terms of a scheme in mind, usually by means of visual symbols; more comprehensively it refers to the adaptation of means to a desired end or purpose. These applications in an architectural context are entirely consistent with general usage: where the term ‘design’ implies only aesthetic content. However, it is an abuse of the word, thoroughly improper’(G.Herbert, 1965);* and we don’t use this meaning in this paper.

Based on dual nature of the word ‘design’, it is discussed as a bridge between means and goal, in its *emphasis on the adaptation of means to ends, and in its emphasis on purpose; design has overtones of problem solving(G.Herbert, 1965)*. This point of view of the design process as a problem solving process is what that is pursued in this paper, and Architectural Design Problem (ADP) refers to Design Problem Solving aspect of the design process. In fact, ADP is discussed as a problem defining and problem solving process. The prominent characteristic of design problems is that they are often not apparent (B.Lawson, 2005), but recognizing and defining them is necessary in order to begin to solve them.

ADPs that usually originate with a client and then architects try to translate the problem from the domain concepts of client to the domain concepts of the designer toward making final design. Due to complex nature of this process, the initial situation of the problem is not usually completely specified. As every problem has a structure of its own, and then a designer tries to define the problem and find the solution as unique.

Also the kind of problems which architects encounter them is ill-defined, means that there are no clear start-state and specific criteria to supply designers with all information required to solve the problem. On the other hand, ill-defined problems do not have a single optimal solution. Unlike non-design problems and science problems in design problems, there are no right or wrong answers; rather there are many satisfactory solutions. In fact, solutions that are proposed by designers are not necessarily true or false.

The other case important to mention about ill-defined problems is that they do not have the explicit statement to provide requisite information for designers to solve them. Indeed, means and ends for finding the solution are unknown in design problems. Then it’s better said that ADPs are problems without a definitive formulation. Based on P.G.Rowe(1987)’*differing formulations of the problems of ill-defined problems imply differing solutions, and vice versa. In other words, the problem’s formulation depends on a preconception that in turn implies a definite direction toward the problem’s solution’*. Anyhow, although the solutions are not just the logical outcome for problems, but problems and design solutions are inexorably interdependent (B.Lawson, 2005). Addition, solutions are never predictable and there is no clear best solution for every problem.

According to Goldschmidt(1997) ‘*the aspects turn designing into an in-deterministic process is very difficult to model’*. But in this paper to model design process (design problem solving process), we initially made the framework consisting of main phases, which encompass design activities. We know that interfaces between these activities will conduct the process to achieve the final design and in the other hand, we know that in such a complex and uncertain process, there is no sequence for performing activities. Actually, due to both problem and designer the process would be pursued differently, as every designer proceeds with the process different from another designer. Therefore, there is no standard sequencing to arrange activities by 1-2-3; as a fixed sequence for all design problems solving process.

By regard to these features of a design process what thing that is prompted us to prescribe the above mentioned framework for modeling the process is associated with the salient feature of a design process; that is called as iterative.

On one hand design process is a complex and vague process, and on the other hand, architectural designer as a human designer whose knowledge and information are limited, then s/he cannot find or define problem and criteria accurately once the process starts. Indeed selecting the start point of this complex

process is impossible for designer. Therefore, the process of problem solving would be strictly influenced by designer's knowledge and experiences. As in this way designer utilizes her/his information to get the initial understanding of the problem to start the process. But to achieve the final solution designer needs to do thinking back and forward to improve information as without this feedback and feed forward design process cannot approach the final design. This is an important feature of a design process that is called iteration. In fact, it is an inherent and inevitable feature for complex process that conducts the process to make a final product. Iteration by influencing the important factors of the design process in terms of cost, time, and quality would be considered as an essential characteristic of process management. Identifying and controlling the iterative behavior of the process is the most important successful key for design process management. Therefore, by regards to the importance of this feature, it seems that we need to identify the iteration and determine its reasons.
To date, in this context, project managers have developed variety techniques to model the process in such a way that can address iteration. These methods have captured the way of planning, organizing, monitoring, and optimizing complex projects, but the most commonly used to address workflow within the process and dividing the projects' activities into sub-activities (individual activities), as leads to optimize project duration. Then, these are not effective in handling the iterative process characterizing design processes. Since this prompted us to utilize a flexible method, which is a matrix based and called DSM; to model the process, and finally representing its iterative feature. We provided detailed description of this method as the main objective of this study in next section.

Table.1 Architectural Design Process Activities

Phase	DSM Labels	Activity
Clarification	A	Identify needs: identify criteria and constraints making requirements list identify problems
	B	Decompose complex problems
	C	Search for solutions(Find ideas)
Conceptual Design	D	Grouping and Firm up ideas into concept variants
	E	Implementation solutions(concepts)
	F	evaluate solutions(ideas) against criteria and constraints
	G	develop concept to prepare preliminary layouts
Embodiment Design	H	evaluate preliminary layouts against constraints and criteria
	I	refine and improve selected preliminary layouts
	J	evaluate selected layouts against constraints and criteria
Detail Design	K	prepare the preliminary parts list
		elaborated detail drawings and parts list
		improve detail design

2. ITERATION IN ARCHITECTURAL DESIGN PROCESS

The dictionary defines Iterative as "involving repetition"(Merriam-Webster, 1987). Most concepts of iteration imply a repetitive character. Nukala et al. and Eppinger et al. define Iteration as "the repetition of activities to improve an evolving design"(M.V.Nukala & Eppinger, 1995),(M.V.Nukala & Eppinger, 1997). Tully emphasizes "repeated operations"(C.J.Tully, 1986). There are also other concepts and definitions associated iteration with implicitly repetitive activity such as revision, rework, redesign, and trying again.

Through this repetitive action designer breaks the problems up into sub problems and does refining and redesigning toward reaching the best solution (design). Exact recognition of problems is impossible without these repetitions of design activities.

Based on iterative nature of a design process, it is a sequence of cognitive activities is performed by a designer to make final design. Here, final design refers to a final solution (is achieved by designer), that is fitted to the situation of problem, criteria and constraints.

On the other word, this process as a heuristic process is looking for finding suitable solution through its activities that depend on information from each other. By regards to this salient feature of a design process, then it is important for designer to be able to determine where/when iteration happens, and also s/he can identify the effect of each kind of iteration on the output of the process, as well.

There are several kinds of classification of iterations, which are introduced here as follows:

1- Based on what is repeated in the process of design; then we have two kinds of iterations; iteration of design tasks, and iteration of mental activities(P.Chuslip & Y.Jin, 2006).
2- Based on where iteration makes changing in a design process toward changing the design level or problem space; there are two following types of iterations(R.Costa & DK.SobekII, 2003):
   - Design iteration
   - Behavioral iteration.
3- Based on different kinds of design problems; it might be happened variety types of iterations(RS.Adams & CJ.Atman, 1999).
4- Based on the design process' strategies; progressive or incremental strategies; there are three following types of iterations(M.J.Safoutin, 2003):
   - Repetition iteration
   - Progression iteration
   - Feedback iteration

However, despite these classifications of iterations; the manner, which iteration impacts a process is the important one in this context. These may be considered as positive iteration in contrast to negative iteration, but it should not be ignored that negative iterations through speeding up can influence the process positively. Accordingly, the other taxonomies of iterations, which have considered in design process management domain, are in terms of necessary vs. unnecessary iterations.

Therefore, the main reason that leads iterations occur in such complex process is due to the interaction of design activities that arises from exchanging information between activities.

3. DSM (DESIGN STRUCTURE MATRIX)

Many of the traditional project management tools such as Gantt, PERT, CPM, and IDEF methods do not address problems stemming from project complexity. They allow project and engineering managers to model sequential and parallel tasks is dependent on one another. These interdependencies of design activities just are characterized by cyclic flows of information, as if activities (A) and (B) are interdependent activities where activity (A) needs information from activity (B), and activity (B) needs information from activity (A). But these tools have failed to interdependencies because they are work-flow methods(P.Tjandra & RP.Smith, 1998). While the DSM method has been able provide this representation in a simple and elegant manner(S.D.Eppinger & RP.Smith, 1997). Nevertheless, this method, which is considered in terms of Interaction Matrix by J.Christopher Jones(Jones, 1972), is as a method of exploring problem structure. Where provides a systematic search for connections between the process' activities that cooperate to solve the design problem.
However, this explicit method through some X marks depicts whether or not there is a connection between each pair of activities. On the other word, these X marks are as the matrix's components to show us which activity supply information for another one, and which activity receives its input from another one. In fact, it enables us to identify the kind of activities' interdependencies by regard to information flows between them. The variety kinds of activities' interdependencies as following:

- Dependent (sequential or series)

- Independent (parallel or concurrent)
- Interdependent (coupled)

The DSM matrix which is constituted of a square matrix is the powerful method, where fulfils all following prospects.

- Representing the iterative feature of Architectural Design Process
- Representing the information-based relations between activities
- Capability to analyze the process toward controlling iterations
- Help designer/manager to improve overall the process(optimizing time duration, quality and cost)

DSM captured in different statements such as ‘the Dependency Structure Matrix’, ‘the Problem Solving Matrix’, and ‘the Design Precedence Matrix’; and what description of DSM, which is considered in this study; is ‘Design Structure Matrix’.

3.1. Implications of DSM

The implications of DSM in this paper are discussed through utilizing this method in a real paradigm. As a practical instance, we asked the final year undergraduate student to participate in the process of designing the Internet Café. Some information is required to make design such as problem, constraints, and criteria would be identified with the design brief and site plan that were given him (sheets A, B, C, and D).

To gather data, we recorded the process and after that by interview, we try to validate data obtained from observing the video. In fact, to obtain more information from the designer, this interview was conducted, as based on all data resulted from both video observing and interview, which made the matrix to present the information-based model of the process. These data are pertinent to activities are shown in DSM label in table.1.

So now we have a matrix 11×11, means that the square matrix with 11 rows and 11 columns that all diagonal elements are filled with solid black cycle to represent activity cannot depend upon its completion. Other components, are filled with X marks if there is an information-based dependency between two activities, where activity (A) needs information from activity (B), we put X mark along the row (A) in the intersection of column (B), and when activity (C) supply information for activity (D) then we put X mark along the column (C) in the intersection with row (D). On the other along each row, the marks indicate the activity correspond to this row receives required information as an input from which other activities. Also, through reading down each column, the marks indicate the activity correspond to this column supplies information for activities at the left hand of each mark.

As a result the matrix that is made to represent this practical process is shown as below (fig.1).

	A	B	C	D	E	F	G	H	I	J	K
A	A										
B		B									
C			C								
D	×	×	×	D					×		×
E	×	×	×		E						×
F						F					
G	×	×		×	×		G				×
H	×	×		×				H			
I	×	×		×	×				I	×	×
J	×	×								J	
K	×	×		×	×		×		×	×	K

Fig.1 Original DSM

3.1.1. Activities' interdependencies

Hereunder all kinds of activities are identified in this matrix based on interaction among them; as introduced before are discussed.

- Consider the row (G) of above matrix; there are two marks along column (A) and column (B) in the intersection of the row (G). It means that activities (A) and (B) supply information for activity (G), in the other word activity (G) can receive its required information from activities (A) and (B) after completion these activities.
- If we consider activities (A) and (B), as shown in following matrix; there is no mark along the row (A) and row (B). These blank rows indicate activities (A) and (B) are independent activities, means that, to complete their performance, there is no need to receive information from other activities. Also note that activity (A) and activity (B) require no information from each other, and this indicates that these activities can be performed in **parallel**.

- Now, we select the column (F), where there is no mark; that indicates activity (F) would not produce any information required by another task.

- Reading along the column (C), the X mark at the intersection with row (D) indicates the output of activity (C) is as input for activity (D). In the other word, activity (C) supplies information for activity (D).

- Mark above the diagonal; for example, consider activity (I) and activity (K). As seen there is the X mark in the intersection of the row (I) with column (K), and another (X) mark in the intersection of the row (K) with column (I), these marks indicate that during the completion of activities (I), and (K); one iteration loop is happened. In fact, activities (I) and (K) are two activities, that outcome of (K) impacts on the activity (I), which performed before. In the other word, these activities are **coupled** activities, which information from a subsequent activity (K) may force a reworking of a prior activity (I). On the other word, in this case these marks refer to required inputs that are not available at the right time of executing of the task (I). Therefore, complex iterations will be identifiable within such coupled blocks

- Another kind of interdependency is **sequential** (series), like activities (C) and (D), where task (C) has to be performed first before task (D) can start.

Therefore, based on this original DSM, we can identify 32 interactions, that 4 loops of iterations occur among them. As seen through using DSM by capturing iteration, we enable monitoring all changes during the process. These changes can potentially increase costs and duration, especially in large projects; these effects of iterations would be more obvious. Thus, the need of controlling and managing this feature becomes essential.

The DSM permits designer/manager could be aware about this reworks, and lets them minimize them and improve throughput and design quality by using DSM's analyzing methods.

3.2. Analyzing the DSM's matrix

3.2.1. Partitioning

As before mentioned one prospect of manipulation of DSM is the capability of this method that enables designer/manager to eliminate or reduce unnecessary iterations. Actually, DSM provides this capability through analyzing matrix. Analyzing leads to re-sequence activities, where original matrix transfers into a new matrix with less upper marks. On the other words, the intention of analyzing is to approach not-entirely sub diagonal matrix, but move upper marks as close as possible to the diagonal. Despite being variety methods for analyzing DSM's matrix, in this paper Partitioning Method is pursued to analyze and re-arranging the original matrix (based on DSM macro (http://www.dsmweb.org/)), and then we utilize Triangularization Algorithm (TA) to determine the loops/cycles of an iteration (http://www.icaen.uiowa.edu). The result for partitioning and utilizing of the TA are presented as follows.
As seen within partitioning, the sequence of DSM's rows and columns was changed in such a manner that the new arrangement causes fewer iteration marks than the original matrix. This partitioning resulted in an iterative loop encompassing five design activities; (D), (E), (G), (I), (K), instead of four iterative loops that were revealed from the original matrix, that consisted of eight design activities; (D), (E), (F), (G), (H), (I), (J), (K). Although, partitioning appears to reduce iteration and shorten the design cycle, the result of such re-sequencing is untenable when we consider the logic of design activities. For example, consider activities (D) (Grouping and Firm up ideas into concept variants) and (J) (evaluate selected layouts against constraints and criteria), which through partitioning suggested that activity J should be performed before executing activity (D); while we know this is not possible.

	A	B	C	F	J	D	E	G	I	K	H
A											
B											
C											
F											
J	×	×									
D	×	×	×						×	×	
E	×	×	×							×	
G	×	×				×	×			×	
I	×	×			×	×	×			×	
K	×	×	×		×	×	×	×	×		
H	×	×				×					

Fig.2 Partitioned Matrix

4. CONCLUSION

This paper is an effort to determine the applicability of DSM in representing complexity and uncertainty of the design process. As, depicted practically in this study, DSM as a tool enables architects to manage the process in a more desirable way. This method is one of the most useful design aids for revealing iterative behavior within the trajectory of specific design activities. Its main value is as a means of representing the critical interdependencies between process' activities. Thus, identifying and determining this, enables designer/manager conduct the process in the optimum manner in terms of cost, and time duration. In spite of, what that mentioned above, there are some difficulties of using this method, as J.Christopher also denoted; making one matrix with more than twenty components needs to spend long time; nevertheless, there is a high probability of errors in entering connections on to even a small matrix. Other difficulties arise when elements are not all of the same hierarchical levels (i.e. if some components are, in fact, parts of other components)(Jones, 1972). The last one, which shown in this study is pertinent to the DSM analyzing; as you can see the re-arranging of activities acquired by DSM Partitioning is not logical. Consequently, we can consider this as an obstacle in optimizing arbitrary and uncertainty process like Architectural Design Process.

5. ACKNOWLEDGEMENT

We hereby acknowledge the contribution made by Universiti Teknologi Malaysia, which have provided financial support for the publication of this paper under an Institutional Research Grant Scheme (VOT 77260).

6. REFRENCES

B.Lawson. (2005). How Designers Think.

C.J.Tully. (1986). Software Process Models and Iteration, Iteration in Software Process. Paper presented at the Third International Software Process workshop, *IEEE*.

G.Goldschmidt. (1997). Capturing indeterminism: Representation in the design problem space. *Design Studies, 18*(4), 441-456.

G.Herbert. (1965). The architectural Design Process.

http://www.dsmweb.org/.

http://www.icaen.uiowa.edu. Triangularization Algorithm.

JC.Thomas. (1987). A Design Interpretation Analysis of Natural English with Applications to Man-Computer Interaction. *International Jurnal of Man-Machine Studies, 10,* 651-668.

Jones, J. C. (1972). Design Methods : Seeds of human futures: *WILEY-INTERSCIENCE*.

M.J.Safoutin. (2003). A Methodology for Empirical Measurement of Iteration in Engineering Design Processes. *Unpublished Program Authorized to Offer Degree: Mechanical Engineering, University of Washington.*

M.V.Nukala, & Eppinger, S. D. (1995). Generalized Models of Design Iteration Using Signal Flow Graphs. *ASME Design Theory and Methodology Conference,Boston MA.*

M.V.Nukala, & Eppinger, S. D. (1997). Generalised Models of Design Iteration Using Signal Flow Graphs. *Research in Engineering Design, 9,* 112-123.

Merriam-Webster (Ed.) (1987).

P.Chuslip, & Y.Jin. (2006). Impact of Mental Iteration on Concept Generation. *[Transition of tha ASME]. Journal of Mechanical Design, 128*(15).

P.G.Rowe. (1987). Design Thinking: *MIT Press.*

P.Pirolli, & V.Goel. (1989). Motivating the Notion of Generic / Design within Information-Processing Theory: The Design Problem Space. *AI Magazine, 10.*

P.Tjandra, & RP.Smith. (1998). Experimental Observation of Iteration in Engineering Design. *Research in Engineering Design*, 107-117.

R.Costa, & DK.SobekII. (2003). Iteration in Engineering Design:Inherent and Unavoidable or Product of Choise Made? *Proceeding of DETC'03 ASME.*

RS.Adams, & CJ.Atman. (1999). Cognitive Processes in Iterative design Behavior. *[29th ASEE/IEEE Frontiers in Education Conference].*

S.D.Eppinger, & RP.Smith. (1997). Identifying Controlling Features of Engineering Design Iteration.

V.Margolin, & R.Buchanan. (1998). The Idea of Design.