INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19–21, 2003

USING DESIGN PROCESS TOOLS TO ASSIST IN THE ANALYSIS OF A PROTOCOL RECORDING

S K Sim, P J Clarkson and D C Wynn

Abstract

A protocol recording of a designer at work provides an effective means by which knowledge of the design process may be elicited. Several methods have been proposed for modelling such a process; this paper presents the findings of research concerning the application of the Design Structure Matrix (DSM) and signposting methods to a protocol recording of a real design task. It is shown how the construction and analysis of both models can reveal areas in which knowledge was not captured in sufficient detail, and how comparisons between the two representations can highlight conflicts in understanding. The process of formalisation and analysis of a protocol using these methods can provide a useful tool in consolidating understanding.

Keywords: signposting model, Design Structure Matrix, protocol analysis

1. Introduction

Designing is context dependent in that the designer often reacts to the aspects of the design under the focus of attention before deciding on the next appropriate design activity. The nature of the next design activity is then dependent upon the current context of the design and the designer's immediate goal. Models and theories describing or prescribing the design process often view designing in terms of stages or phases, and hence fail to capture the in-situ nature of the ongoing process. In other words, these models or theories fail to describe the manner in which the designer reacts to the design situation, and the need to provisionally determine the next design action in order for the design to progress.

Exceptions include Ullman's decision support model [1] and Clarkson and Hamilton's [2] signposting model. The latter may be used to describe the contextual nature of the design process and prescribe the next appropriate design activity or task. In order to construct such a model, task knowledge can be elicited through analysing a protocol recording of an actual designer at work.

The work presented here investigates in what manner the design structure matrix (DSM) [3] and signposting methods are effective in modelling a recorded design process. Can these tools provide insights into the task performed by the designer? In addition, does the protocol study reveal any inadequacies in these methods with respect to their effectiveness in modelling the actual design process, or vice versa? These questions are addressed with the aid of a protocol recording of an experienced designer determining the general arrangement of an offshore patrol vessel.

2. Design Models

2.1 Design structure matrix

Steward [3] formulated the Design Structure Matrix (DSM) to represent and analyse dependencies between tasks or activities in the design of complex systems. A nonempty entry $a_{ij} = *$ in the incidence matrix indicates that activity j precedes activity i (alternatively, the output of j is an input of i). For a complex set of activities, no special structure may be visible in the matrix. Using algorithms for partitioning [6] and tearing [3], precedence relationships between activities are clarified and clusters of coupled activities can be more easily recognised. Dependency patterns occurring in models of complex design processes can be classified into various types (e.g. independent, cycle, serial, branch, merge) [5]. Kusiak [5] has shown that the DSM approach can be applied to modelling the decomposition of the activity-parameter matrix, the decomposition of products, systems and processes, and the modularity of products.

2.2 Signposting

Signposting has been developed to dynamically react to the current state of the design under consideration, directing the designer to the next appropriate and available task, or tasks, at each point in the design process. This direction, or *signposting*, is derived from knowledge of possible design tasks and their associated contextual information.

The dynamic nature of signposting adds greater flexibility to the Design Structure Matrices used by Eppinger and others and extends the potential of petri nets to include the notion of parameter *confidence* as a means to differentiate between similar tasks. In addition, signposting eliminates the need to capture the task precedence prior to constructing a model. The resulting process model is truly dynamic, reacting to the successes and failures of the emerging design.

The signposting model is driven not only by the presence of a parameter that enables a task to be executed, but also its associated confidence. Task precedence knowledge is not captured explicitly. However, the tasks, with their associated contextual knowledge, define a dynamic design process which changes as the design progresses. The contextual knowledge implies a task precedence which is dependent upon the current state of the design.

The flexibility of signposting allows the ordering of tasks to be changed in response to measures of success of the emerging process. For example, it is possible to reorder the tasks in response to the predicted cost or duration of the downstream process.

Model assumptions

The signposting model is based upon the assumption that the design process may be thought of as a series of tasks concerned with the identification, estimation and iterative refinement of key design and performance parameters. The design is considered complete once a sufficient level of confidence in the parameters is achieved.

In this context, design parameters are those that define the product's physical structure, such as its geometry and the materials used. Performance parameters, for example stress distributions for given loading (numerical) or aesthetic characteristics (non-numerical), are then derived from the design parameters and used to assess the performance of the design.

In the proposed task-based representation, a generic task is used as the primary building block of the process. The representation couples knowledge describing the specific method to be used to perform the task with contextual knowledge describing both the context in which the task should be performed and the likely consequences of performing the task (Figure 1).

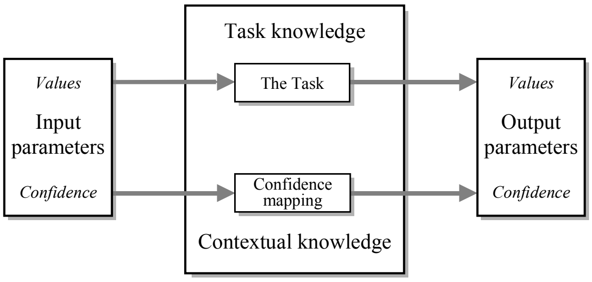


Figure 1. Task representation in the signposting model

Task representation

In the initial stages of design, empirical formulae and rough calculations are used to establish estimates of key design parameters. Later, more exact predictions may be derived using complex computational tools and / or physical tests on prototypes. Due to the higher costs of these later analyses, they are usually only performed when the designer has sufficient *confidence* in the accuracy of key input parameters.

Confidence encompasses a number of meanings. To have high confidence in a parameter means that the parameter is detailed, accurate, robust, well understood, physically realistic and, in the case of a performance parameter, meets predefined performance requirements. The confidence in the output parameters is then a function of both the accuracy of the particular task and the confidence in the input parameters.

In the model, confidence is represented using three discrete levels. These levels are assigned as:

Low -	an initial unproven design or performance estimate;
Medium -	a feasible design or performance estimate;
High -	a feasible design or performance estimate if the resultant product performance satisfies the design requirements.

These three levels of confidence were chosen to demonstrate the concept of confidence mappings. However, in more advanced applications more levels may be desirable. The confidence mapping for a particular design task may be represented by a table relating the minimum required confidence of the input parameters to a particular level of confidence in the output parameters (Figure 2). The confidence mappings are derived from textual descriptions of the tasks and the expert's knowledge of its appropriate use. They describe the maximum benefit to be achieved by executing the task.

Given the parameters available and their associated levels of confidence at any stage in the design process, it is possible to estimate the effect of undertaking specific tasks from their associated confidence mappings. This forms the basis of the dynamic task planning which is at the core of the signposting technique. Given a request to calculate a specific parameter to a given level of confidence, the most appropriate task sequence may be identified [2].

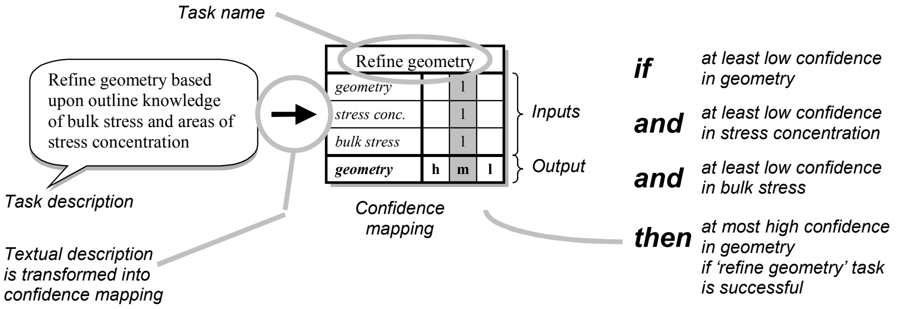


Figure 2. An example confidence mapping

3. Knowledge elicitation using protocol analysis

Knowledge elicitation is a difficult task, with sources of knowledge including human experts, textbooks, work files and previous examples of the process under consideration. While the different sources may each provide insights into the design process, a protocol recording of an actual designer at work remains an effective means by which knowledge of a design process may be elicited. The research presented here is based on a protocol recording of an experienced Senior Ship Designer determining the general arrangement of an offshore patrol vessel. The designer has more than 20 years of design experience in a company that provides consultancy in the design and supervision of construction of naval crafts and warships. The recording was made over a period of 2 hours and 45 minutes, and the AutoCAD system was used to record all the drawings made during the process.

The actual sequence of activities performed on the various design objects by the designer was extracted from the protocol recording [6]. It was subsequently presented in a DSM, and the resulting matrix analysed to determine interesting features of the process. The individual activities were then defined in terms of their input and output parameters, the cost of performing the task, its probability of failure and its consequence of failure. The resulting lists of tasks and parameters were coded into Lisp statements for analysis using the signposting tool implemented by Melo [8].

4. Analysing the protocol using the DSM model

A high-level component DSM comprising 30 components was constructed following the protocol recording. Analysis of the DSM using the operations of partitioning and tearing revealed connectivity relationships between components of the vessel, as referred to in the protocol.

4.1 Partitioning

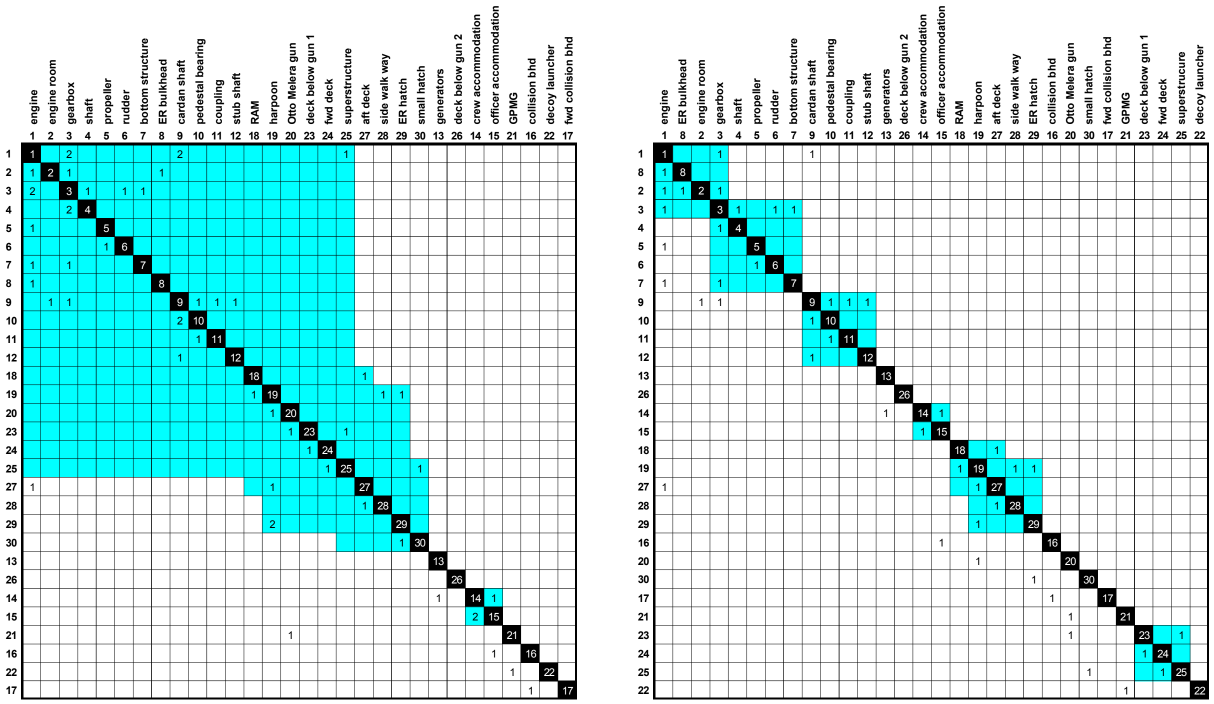
Modelling the results of the protocol analysis in a DSM revealed some prominent clusters of interdependent components, most notably the engine room and related components. The application of a partitioning algorithm further accentuates three related clusters involving the engine room, the weapon suite and the components on the main deck such as superstructure, engine hatch and aft deck. Of particular interest is an interaction revealed between the superstructure and engine components. This link was not directly accounted for in the protocol recording, and was unexpected since the superstructure is usually considered to have little impact on the engines.

4.2 Tearing

The DSM reveals dependency loops which cannot be solved by partitioning; in other words, the model indicates that the process cannot be resolved into a consecutive sequence of activities. However, the designer did complete the process; it can be concluded that these explicit dependency loops were broken by making estimates and utilising existing (tacit) knowledge of the product and design process. After tearing these broken dependencies and then repartitioning, the DSM becomes lower triangular, reflecting the serial process actually executed.

Using the analysis of the original DSM following partitioning (Figure 3, left), the superstructure-engine interaction was chosen as an initial candidate for such a broken dependency. Tearing this link and then repartitioning resulted in several distinct but related clusters (Figure 3, right).

Following the leading diagonal down, the first cluster depicts the dependencies among the engines, gearbox, engine room and engine room bulkhead. The second shows the relationships between the components in the shafting system (i.e. gearbox, drive shaft, propeller and rudder) to the bottom structure. The third relates the coupling and pedestal bearing to the shafts. The fourth distinct cluster highlights the dependencies between a bank of harpoon missiles, the side walk way and the engine room hatch. The fifth cluster shows the influence of the superstructure on the forward deck and the under deck carousel for the gun above.



	engine 1	engine room 2	gearbox 3	shaft 4	propeller 5	rudder 6	bottom structure 7	ER bulkhead 8	cardan shaft 9	pedestal bearing 10	coupling 11	stub shaft 12	RAM 18	harpoon 19	Otto Melera gun 20	deck below gun 1 23	fwd deck 24	superstructure 25	aft deck 27	side walk way 28	ER hatch 29	small hatch 30	generators 13	deck below gun 2 26	crew accommodation 14	officer accommodation 15	GPMG 21	collision bhd 16	decoy launcher 22	fwd collision bhd 17
1	1		2						2									1												
2	1	2	1					1																						
3	2		3	1		1	1																							
4			2	4																										
5	1				5																									
6					1	6																								
7	1		1				7																							
8	1							8																						
9		1	1						9	1	1	1																		
10									2	10																				
11										1	11																			
12									1			12																		
18													18						1											
19													1	19						1	1									
20														1	20															
23															1	23		1												
24																1	24													
25																	1	25				1								
27	1													1					27											
28																			1	28										
29														2							29									
30																					1	30								
13																							13							
26																								26						
14																							1		14	1				
15																									2	15				
21															1												21			
16																										1		16		
22																											1		22	
17																												1		17

	engine 1	ER bulkhead 8	engine room 2	gearbox 3	shaft 4	propeller 5	rudder 6	bottom structure 7	cardan shaft 9	pedestal bearing 10	coupling 11	stub shaft 12	generators 13	deck below gun 2 26	crew accommodation 14	officer accommodation 15	RAM 18	harpoon 19	aft deck 27	side walk way 28	ER hatch 29	collision bhd 16	Otto Melera gun 20	small hatch 30	fwd collision bhd 17	GPMG 21	deck below gun 1 23	fwd deck 24	superstrucure 25	decoy launcher 22
1	1			1					1																					
8	1	8																												
2	1	1	2	1																										
3	1			3	1		1	1																						
4				1	4																									
5	1					5																								
6						1	6																							
7	1			1				7																						
9			1	1					9	1	1	1																		
10									1	10																				
11										1	11																			
12									1			12																		
13													13																	
26														26																
14													1		14	1														
15															1	15														
18																	18		1											
19																	1	19		1	1									
27	1																	1	27											
28																			1	28										
29																		1			29									
16																1						16								
20																		1					20							
30																					1			30						
17																						1			17					
21																							1			21				
23																							1				23		1	
24																											1	24		
25																								1				1	25	
22																										1				22

Figure 3. The partitioned DSM before (left) and after (right) the tearing operation

All the resulting clusters can be directly related to the properties of the design; the first three result from direct and complex connectivity between components, and the last two from geometric arrangements. In summary, the tearing operation produced supporting evidence that the superstructure-engine dependency was broken by the designer. The activity of estimation was not captured in the protocol, suggesting non-verbalised use of tacit knowledge.

5. Analysing the protocol using the signposting model

The protocol has been analysed to give a sequence of activities or tasks as defined by Sim [6]. Based on the task representation of the signposting model, each task was defined in terms of its input and output, confidence mapping, cost of performing the task, probability of failure and consequence of failure [7]. These tasks were coded into Lisp statements and analysed using the signposting model as implemented by Melo [8], but meaningful simulation was not possible due to the decoupling or non-connectivity of these tasks. However, the building and analysis of the model has proven useful in improving understanding of the process.

5.1 Decoupling reveals uncaptured knowledge

The signposting model consists of a group of tasks and a set of associated parameters elicited through the protocol recording. In analysing the protocol, the designer's actions are divided into discrete activities, each of which may be viewed as a knowledge transformer linking input and output knowledge. Each activity is then modelled as a signposting task, with the input and output knowledge represented as parameters involved in the execution of that task. The characteristics of the elicited model are then dependent upon the level of granularity chosen; in other words, upon the conceptual size or length of each task.

As activities become shorter the model becomes less data-driven and more knowledge-driven; more tasks represent tacit thought processes which are difficult to elicit. This distinction is illustrated in Figure 4, where external data are categorised separately from explicit knowledge and tacit knowledge. As the model becomes more finely grained, more parameters and then tasks fall into the internal knowledge category. Explicit internal knowledge may be represented as parameters in the model, but these parameters are likely to be difficult to conceptualise, separate and quantify; tacit knowledge is unlikely to be expressed in any form. Modelling linked activities in cases where the connecting parameter is not made explicit results in seemingly disconnected tasks. (eg. Tasks A, B in figure 4)

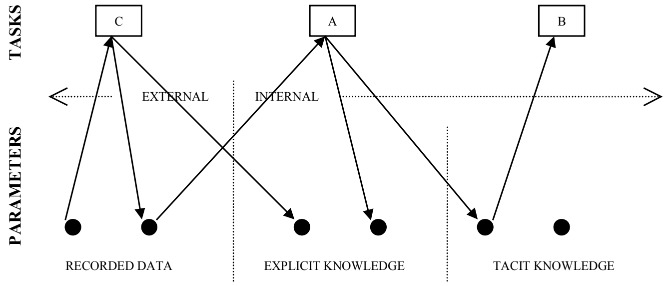


Figure 4. Relationship between explicit links between tasks in the model revealed by the protocol and the internal knowledge of the designer

Given the inherently serial nature of the protocol recording, the degree of decoupling present may thus be seen as indicative of the completeness of a process model. Assuming that the designer's actions were guided by knowledge-based reasoning, a lack of connectivity suggests that much of this knowledge is not represented explicitly in the model. Without all the knowledge used encoded in the form of parameters, the model is incomplete and cannot fully explain the designer's choice of actions. Building such a model following a protocol recording highlights areas of missing knowledge, and prompts the analyst to explore these areas in greater depth. The process of encoding the protocol into this model prompts consideration of the rationale behind the designer's decisions, and has proven a useful tool in understanding how the changing state of the design drives the process.

In the example studied, a signposting model of 160 tasks was derived using the protocol recording together with the analyst's existing knowledge of the process. Figure 5 shows a partial visualisation of the model.

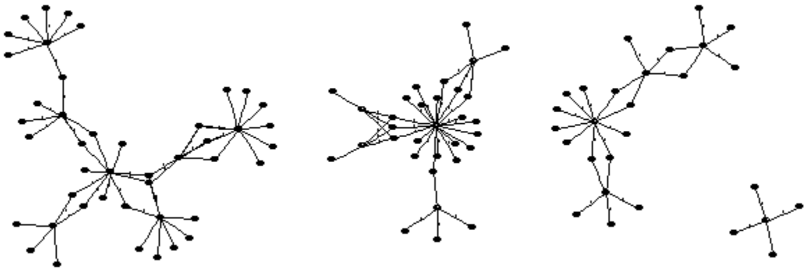


Figure 5. Partial process visualisation revealing disjoint nature of model. Parameters representing captured knowledge are shown as nodes and tasks representing activities as arcs. Decoupled clusters highlight independent processes which are not linked by interactions between components of the design.

The generation of this diagram revealed many distinct parameter clusters linked by tasks; in other words, the model consisted of many distinct clusters of knowledge and design activities which did not interact. This characteristic indicated that much knowledge had not been captured and conceptualised in enough detail. Following this analysis the protocol recording was revisited, leading to a deeper understanding of the process which was reflected in greater connectivity between tasks. The final model still consisted of several groups of non-interacting clusters, but this can be seen to be consistent with the conceptual nature of the design task. In other words, the designer relied heavily upon previous knowledge and tacit reasoning to guide progression between activities; these connections are difficult to capture and explicitly represent in the model, but their existence has been made clear by the formalisation and analysis. The quality of the elicited model has been assessed without first-hand knowledge of the process under consideration.

Difficulties encountered during modelling

Two key difficulties encountered in modelling the design process as captured in the protocol lie in selecting appropriate granularity for the model, and in the need for consistent nomenclature. Experimentation has shown that the usefulness of the above analysis is highly dependent upon the granularity of the model. If the process is modelled at a high level where activities make use of only explicit parameters, little or no missing knowledge is revealed. Similarly, at too low a level the model becomes extremely disconnected, depending upon existing knowledge and tacit reasoning which is difficult to elicit and formalise reliably.

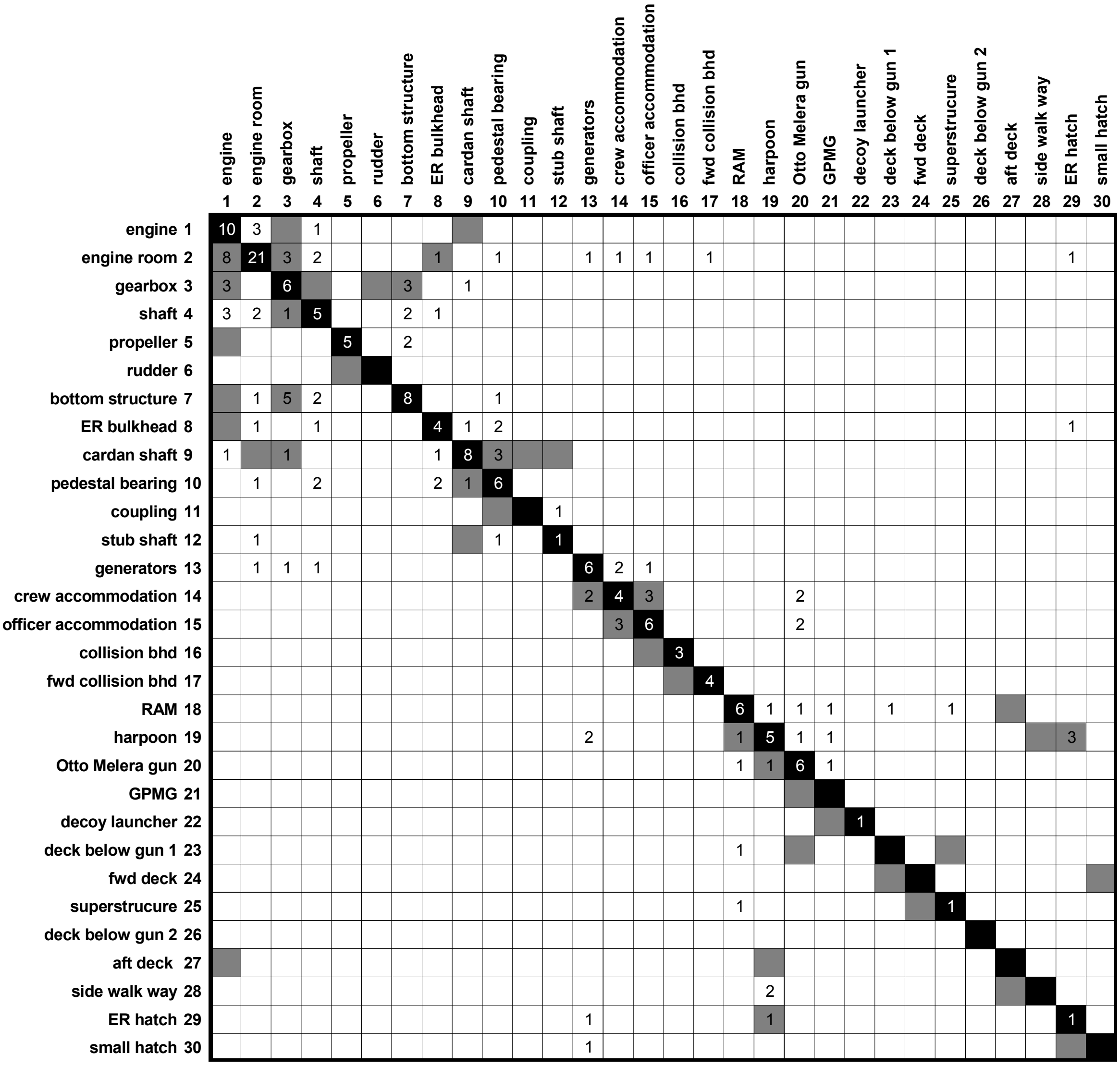
The signposting model is extremely sensitive to the nomenclature used for parameters; if an item of knowledge is referred to using different parameter names, decoupling will result. Studying an otherwise consistent model can reveal these decouplings and indicate areas of knowledge which have not been conceptualised cleanly.

6. Comparisons between representations reveal inconsistencies

In addition to aiding elicitation, these models can be useful in verifying understanding of the process. A signposting model of the process can be converted into a more compact DSM form at the cost of some information loss, and compared to a component DSM model of the product. Comparisons between the different representations of the same process can then reveal inconsistencies in understanding.

The list of parameters elicited through the protocol represent the architecture of the design, and these parameters were first categorised into the 30 high-level components which they describe. Each of the 160 tasks may then be viewed as a process dependency between two of these components. The signposting model is ‘rolled up’ into DSM form, with each mark in the matrix representing one or more tasks linking the component in the row to that in the column. Individual tasks which cause the signposting model to diverge from the DSM model may then be extracted, highlighting misunderstandings between product and process.

The signposting model elicited from the protocol recording is data driven, with the high level form of the model dependent upon the interaction between the individual tasks and parameters; this form is difficult to visualise and hence influence while the model is built. In contrast, the DSM elicited directly is subject to interpretation at a high level. Comparing the representations reconciles a high level understanding of the process with the individual activities and knowledge flow.



	engine 1	engine room 2	gearbox 3	shaft 4	propeller 5	rudder 6	bottom structure 7	ER bulkhead 8	cardan shaft 9	pedestal bearing 10	coupling 11	stub shaft 12	generators 13	crew accommodation 14	officer accommodation 15	collision bhd 16	fwd collision bhd 17	RAM 18	harpoon 19	Otto Melera gun 20	GPMG 21	decoy launcher 22	deck below gun 1 23	fwd deck 24	superstrucure 25	deck below gun 2 26	aft deck 27	side walk way 28	ER hatch 29	small hatch 30
engine 1	10	3	▨	1					▨																					
engine room 2	▨ 8	21	▨ 3	2				▨ 1		1			1	1	1		1												1	
gearbox 3	▨ 3		6	▨		▨	▨ 3		1																					
shaft 4	3	2	▨ 1	5			2	1																						
propeller 5	▨				5		2																							
rudder 6					▨	■																								
bottom structure 7	▨	1	▨ 5	2			8			1																				
ER bulkhead 8	▨	1		1				4	1	2																			1	
cardan shaft 9	1	▨	▨ 1					1	8	▨ 3	▨	▨																		
pedestal bearing 10		1		2				2	▨ 1	6																				
coupling 11										▨	■	1																		
stub shaft 12		1							▨	1		1																		
generators 13		1	1	1									6	2	1															
crew accommodation 14													▨ 2	4	▨ 3					2										
officer accommodation 15														▨ 3	6					2										
collision bhd 16															▨	3														
fwd collision bhd 17																▨	4													
RAM 18																		6	1	1	1		1		1		▨			
harpoon 19													2					▨ 1	5	1	1							▨	▨ 3	
Otto Melera gun 20																		1	▨ 1	6	1									
GPMG 21																				▨	■									
decoy launcher 22																					▨	1								
deck below gun 1 23																		1		▨			■		▨					
fwd deck 24																							▨	■						▨
superstrucure 25																		1						▨	1					
deck below gun 2 26																										■				
aft deck 27	▨																		▨								■			
side walk way 28																			2								▨	■		
ER hatch 29													1						▨ 1										1	
small hatch 30													1																▨	■

Figure 6. Comparison between component DSM and signposting models

(DSM dependencies are highlighted, signposting connections are numbered. For example, the number ‘8’ in cell (*engine room,engine)* indicates that 8 tasks in the signposting model link parameters describing component *engine* to those describing component *engine room.* Note that signposting tasks may connect two parameters describing the same component, leading to numbers on the leading diagonal)

Figure 6 depicts the comparison between the 30 component DSM and original 160 task signposting model in the example study. In this case, it is clear that there is significant divergence between the models, indicating incomplete understanding of the process and supporting the evidence from the signposting analysis described above. For example, a dependency of *shaft* upon *engine* is strongly suggested by the signposting model, but was not considered in the high-level product model. In contrast, a dependency of *engine* upon *shaft* is weakly suggested by the signposting model but not present in the DSM model. The single task causing this dependency was isolated, and some further consideration suggests that the connection is incorrect and should be removed.

It can also be noted that in this case several links in the original DSM are not represented at all in the signposting model, and that certain components (eg. *engine*, *engine room*) are involved in many more activities than others. This suggests that either the components are inappropriately selected with varying levels of granularity, or that the protocol is lacking detail in certain areas. Figure 6 shows clearly that most of the recorded data was concentrated on the components in the top left quadrant of the matrix.

7. Conclusions

The DSM provides a succinct depiction of the relationships among the design objects as captured in the protocol. While it may not capture the rationale behind the activities performed by the designer, it does provide a means to show the interdependencies among the captured design objects. However, dependencies between components may not be captured, and some design objects may not be represented at all. Although the captured relationships are represented clearly in the matrix, only the information which is verbalised by the designer can be shown. The DSM describes 'how' the process proceeded but not 'why' the particular sequence of activities was chosen; hence the possible existence of missing or inaccurate information is not made explicit.

In contrast, a signposting model represents a deeper understanding of the rationale behind the designer's actions. When modelling activities as tasks which transform design parameters between states, the analyst must consider the verbalised data as a representation of a chain of activities linked by a flow of information. Missing links in this chain suggest that the process has not been captured fully; for example, this may highlight the non-verbalised use of existing knowledge when configuring elements of the design. Although a signposting model of a process can provide more insight than a DSM representation, it is also more difficult to elicit, visualise and reason about.

Constructing these models is one way to analyse a protocol recording of a designer at work. The methods outlined can reveal areas of inadequate or possibly inaccurate data, and thus help direct further study of a particular process. More generally, further research in the use of descriptive design process models as analysis tools may lead to an improved understanding of the protocol study as a method for process elicitation.

In summary:

- *Formalising knowledge elicited from the protocol in the form of a signposting model and/or DSM can aid understanding of the process;*
- *Analysing these models can reveal areas where knowledge was not captured in the protocol, and highlight the use of tacit knowledge and/or reasoning by the designer;*
- *Application of these techniques could provide a useful tool in interpreting protocol recordings of design tasks.*

References

[1] Ullman D.G., "Toward the Ideal Mechanical Engineering Design Support System", Research in Engineering Design, Vol. 13, No. 2, 2002, pp.55-64.

[2] Clarkson P.J. and Hamilton J.R., "Signposting: a parameter-driven task-based model of the design process", Research in Engineering Design, Vol. 12, No. 1, 2000, pp. 18-38.

[3] Steward D., "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management, EM - Vol. 28, No. 3, 1981, pp.71-74.

[4] Gebala D.A. and Eppinger S.D., "Methods for analysing design procedures", Proceedings of ASME Design Theory and Methodology, Miami, Florida, 1991.

[5] Kusiak A. "Engineering Design - Products, Processes and Systems", Academic Press, USA, 1999.

[6] Sim S.K., "Modelling Learning in Design", PhD Thesis, University of Strathclyde, Glasgow, 2000.

[7] Clarkson P.J., Melo A. and Connor A., "Signposting for Design Improvement -A dynamic approach to design process planning", Proceedings of Artificial Intelligence in Design, Worchester, Massachusetts, USA, 2000.

[8] Melo A., "A State-Action Model for Design Process Planning", PhD Thesis, University of Cambridge, 2002.

Sim Siang Kok
School of Mechanical and Production Engineering,
Nanyang Technological University,
Nanyang Avenue,
Singapore 639798.
Republic of Singapore

David Wynn
Engineering Design Centre,
University of Cambridge,
Trumpington Street,
Cambridge CB2 1PZ,
United Kingdom.
Tel. Int +44 1223 332 673
Fax. Int +44 1223 332 662
E-mail: dcw24@eng.cam.ac.uk
URL: http://www-edc.eng.cam.ac.uk/people/bcw24.html