An Approach to the Analysis of Design Protocols[1]

John S. Gero and Thomas Mc Neill

Key Centre of Design Computing
Department of Architectural and Design Science
University of Sydney NSW 2006 Australia

{john, tom}@arch.su.edu.au

Abstract

Little research has been done on how designers actually design. Much of design research is concerned with computer based models or is based on anecdotal evidence of the design process. This paper describes the development and application of a methodology that uses protocol studies of designers engaged in design to investigate the process of designing. A coding scheme is developed and applied to design protocols. The scheme brings structure to the unstructured data of the protocols without detracting from the richness of the data. Results are shown that illustrate the utility of this approach in gaining some insight into how designers design.

Keywords

Design Protocols, Design Process, Function-Behaviour-Structure Model, Design Research.

Designing is considered to be one of the significant intellectual activities because of its complexity and the effects its results have on society. Given the large body of design research it is surprising how little we know about designing: the activity carried out by designers. Design research over the last three decades has largely concentrated on computer-based models of design. This certainly made sense since the computer holds the promise of becoming a tool to aid human designers. Whilst progress has been slow, in some areas considerable advances have been made. However, much research still needs to be carried out to provide computer-based support for the more interesting aspects of designing[1]. There still remains a paucity of literature on how designers design which is based not on anecdotes or on personal introspection but on reproducible results, results which are capable of characterising designing. This paper takes the *think aloud* method first described by Ericsson and Simon[2] then further developed by Van Someren et al.[3] and extends it so that it becomes a useful tool in design research to aide in developing an understanding of how designers design.

Better source information on designing as a time-based activity will allow design researchers to develop richer models of designing which in turn will provide the basis for a better understanding of designing. Such an understanding can feed into the development of computer-based support tools.

The remainder of this paper describes the development and use of a methodology for the analysis of design protocols. It commences with a brief introduction to related work before

[1] Gero, J. S. and McNeill, T. (1998). An approach to the analysis of design protocols, *Design Studies* **19**(1): 21-61

developing the overall framework for the analysis methodology. This is followed by a detailed development of the coding scheme and coding method. All the codes and methods are demonstrated through examples. The results of the method applied to four protocols are then presented.

1. RELATED WORK

Empirical studies of designers in a variety of design domains have been increasing in recent years. Lee and Radcliffe[4] presented a design problem to inexperienced designers and gave them a week to produce a solution which was then analysed. Guindin[5] analysed software engineers in an experiment in which they were presented with a problem of medium complexity and given two hours to complete a design. Their actions were recorded and subsequently analysed.

Visser[6] observed a mechanical engineer in his normal place of work engaged in a typical design problem over a period of three weeks. Later Visser[7] made more detailed studies of designers involved in software design and a multi-disciplinary design task. Davies[8] summarised empirical studies conducted in software design. These studies have given a coarse grain view of the cognitive processes involved in design. Lloyd and Scott[9] imposed an external structure based on models of design on the protocols they had collected.

Cross[10] and Cross, Dorst and Roozenburg[11] each give collections of work by researchers of the times. Most recently a workshop was conducted at Delft University of Technology[12] in which participants were each given a video recording and transcript of a design session and asked to analyse the protocol. This revealed a range of approaches which gives insight into many aspects of conceptual design.

Stauffer[13] and Stauffer and Ullman[14] observed five mechanical engineers engaged in open ended design problems lasting form six to ten hours each. Two problems were used, a one-off design and device that would be made in production quantities. This study divided the designers' actions into short duration "operators". These operators would appear in sequences which the author identified as "methods". The frequencies of the four principle methods were recorded.

2. DESIGN REASONING

Protocol data is very rich but unstructured. In order to obtain a detailed understanding of design processes it is necessary to project a framework on to the data. This framework derives both from direct observation of the designer's interaction with the problem domain and from models of design reasoning.

The conceptual design process can be considered as one in which the designer navigates through an abstract problem domain and employs various strategies to elaborate the problem description.[15,16,17,18,19] In order to give a richer representation of the design process a distinction is made between the designer's place in the problem domain and the strategies used by the designer during the design process.

2.1. Problem domain

The designer's navigation through the problem domain can be represented in two orthogonal dimensions. The first of these, involving function, behaviour and structure, is derived from a model of design reasoning.[18,20,19] *Function* relates to the purpose of an artefact. *Behaviour* relates to the actions or processes of an object or artefact. Reasoning in *structure* involves the manipulation of objects or their relations to bring about a physical solution. Reasoning with function, behaviour and structure can be differentiated for any design episode independently of the design problem although the actual categorisation is dependant on the specific design problem.

The second dimension divides the problem domain into a number of levels of abstraction. The designer's attention shifts from high level overall views of the problem down to consideration of low level details of the problem. This dimension is derived from the way in which the designer approaches the problem. Some designers may subdivide a problem into a number of different sub-categories. Other designers may proceed without consciously identifying different sub-problems. In the former case the identification of levels of abstraction is easier.

2.2. Design Strategies

A framework can be brought to the design process by considering the designer's activity as consisting of a sequence of actions or micro strategies each typically lasting for a few seconds or tens of seconds. The design process can be viewed as one in which the designer engages the design problem by calling upon a repertoire of micro strategies.

The micro strategies are self contained and relate mainly to the current state of the process. A rich representation of the designer's actions can be formed by identifying similar actions and creating a list of the repertoire used during the design episode. The representation can be further enriched by classifying the micro strategies into a small number of groups. The result is a view that is both data driven, in that the protocols are the source of the repertoire, and model driven since models of design are used to add further structure to the repertoire. The number of different micro strategies that can be identified in a design process is dependent on both the designer's experience and on the complexity of the problem. To solve a simple problem experienced designers may call upon a subset of their full repertoire. Conversely inexperienced designers faced with a difficult problem may not have a large enough repertoire to call upon.

In addition to identifying micro strategies, the designer's approach can be viewed in the longer term with the designer executing a long term plan or macro strategy typically lasting several minutes. Macro strategies can be identified by looking beyond the current state and assessing the designer's behaviour in the context of the whole design solution. The macro strategy dimension adds richness to the representation by adding context to the micro strategies.

3. PROTOCOL ANALYSIS

The approach to protocol analysis described in this paper involves the development of the coding scheme during the analysis. Whilst this is the basis of all protocol studies the approach presented here differs from the standard approaches in both the addition of model-based codings and in the introduction of a very rich design-dependant set of codes. The protocol is segmented, a coding scheme developed and the segments categorised. Before presenting the coding method used, the design episodes are discussed, the coding scheme is described and then the processing of the results is discussed.

A collection of conceptual electronic design protocols was completed at Loughborough University and is detailed in Mc Neill and Edmonds.[21] The design tasks were selected by the experimenter and each designer from the designers' normal work. The sessions were recorded in the designer's normal place of work. The designers verbalised their thoughts during the design episodes. The designers were video taped. The video equipment was configured to look over the shoulder of the designers and to impact as little as possible on the designers. Each designer’s speech was transcribed and time coded. A description of the designers’ actions was added to the record.

After the protocol analysis of the three electronic design episodes was completed the analysis method was applied to another design episode that was prepared for the Delft Workshop on Analysing Design Activity.[12] The experimental conditions were similar to the three electronic design protocols. One important difference was that the designer was given a brief at the start of the session rather than being asked to work on something that was already familiar to him.

3.1. Design Episodes

Three electronic design episodes were recorded. The first and the third design episodes were undertaken by the same designer, a PhD student with a graduate qualification in Electronic Engineering. The design episodes form parts of the overall design of a system which is to be completed as a part of his PhD requirement. The second design episode was undertaken by a graduate in computer studies with 20 years experience in designing computer hardware of the type undertaken for this study.

The objective of the first design episode is to design part of an interface system between an image processing host computer and a neural network based image processor. The overall system will be used to identify the location of the eyes and mouth in a video image of a human face. The host computer pre-processes the image to identify the regions in which the eyes and mouth are most likely to be and the neural network identifies the exact location of the features. The design episode is concerned with the subsystem (RAM controller) that, given the locations of the features, selects segments of memory corresponding to the regions of interest and feeds this data in the appropriate sequence to the neural network processor board.

The second design episode is concerned with the design of an industrial controller to control a industrial plant using electrically operated pneumatic and hydraulic valves. There is a particular plant intended for the controller in the first instance but the intention is the controller should be general purpose and adaptable to other industrial plants. The design requirement calls for a general controller card that can be programmed to meet a range of functional requirements. This is to be achieved by the use of a Programmable Array Logic (PAL) device and the design session centres on the design of the interface between a PAL device and a typical industrial plant.

The third design episode, occurring much later in the design process, involves the design of another part of the overall system being designed in the first design episode. In the first instance the neural network is pre-programmed to recognise the features. The system is to be modified at a future date to incorporate a learning feature. In learning mode the neural network processor is fed with the input data and the correct results. This mode involves information routing that is fundamentally different to the other modes previously designed. The purpose of the third design episode is to assess whether the existing architecture, designed in previous sessions, will support the learning mode in the future. The objective is to establish some algorithms in a general form that can be implemented on the existing system. The designer will define the algorithms only to a level of detail that will satisfy him that the future implementation of the learning mode will be achievable.

The Delft design episode involved the design of a bicycle rack to carry a back pack on a mountain bike. The designer was a mechanical engineer with twenty years experience currently working as a robotic systems designer. The protocol recording conditions were similar to those for the three design episodes. There were two significant differences with the Delft experiment. Firstly the designer was not exposed to the problem before the design exercise and secondly the designer was given a set time in which to complete the design exercise.

3.2. Segmentation

Van Someren et al.[3] describe a process of aggregation of segments into "episodes". Our method focuses on designer actions or intentions. The protocol is divided along lines of designer intentions. The designer's intention is interpreted for each segment. (A segment in our terminology corresponds closely to an episode in the terminology of van Someren et al.[3]) A change in intention flags the start of a new segment.

Table 1 shows an excerpt of the protocol sequence segmented by designer intentions. The first column shows the time, the Dialogue column provides a transcription of the designer's words and the Actions column records the designers actions and any sketches made by the designer. The three coding columns between the Time column and the Dialogue column are discussed later. The long pauses are still represented by large gaps in the text while short pauses are represented by a series of dots, i.e. "..." A new segment is indicated by the text beginning on a new line and a time code is added.

The note in the actions column informs that the designer was busy sketching during the pause after 21:42. The approach of van Someren et al.[3] segments the protocol first using syntactic and verbal queues. These segments are then aggregated in the coding process. The approach taken here is to code the protocol directly, concentrating on the designer's intentions rather than verbal or syntactic events.

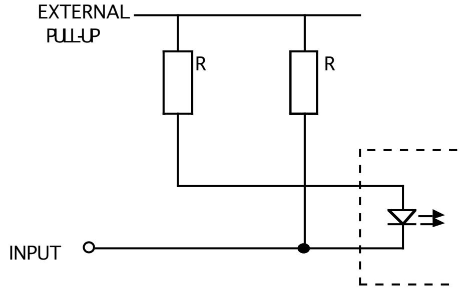
Time	PD	Mi	Ma	Dialogue	Actions
21:30	R1F	Ju	Bu	We need a low impedance OK	
21:32	1S	Ps	Bu	the easiest way to configure it is to do something like this I guess	
21:42				We provide an external pull-up	*Adds the following to the left of the previous drawing.* 
21:51				And some pull-up voltage	
21:56	1F	Ju	Bu	which gives us, the ability for the user to determine that voltage	
22:04	1S	Ps	Bu	A resistor of some sort which is fixed by the external pull-up and this becomes our input from the outside world	
22:18	1B	An	Bu	and what happens now is that when ... its an active low input isn't it because under normal circumstances the LED would be off if there's nothing connected to there the LED would be off ... when we pull that down to ground that will be the way that the input is made the LED will glow, seems reasonable ...	

Table 1. Suggested segmentation in the second design episode.

3.3. Coding Scheme

The coding scheme was allowed to evolve during the analysis. As segments were identified that did not fall neatly into the existing scheme, a new category was introduced or an existing category was redefined. Three coding columns were added to the blank transcripts. The first was used to encode the problem domain over the two dimensions of *Level of Abstraction* and *Function, Behaviour and structure*. The other two coding columns were used for *Micro Strategy* and *Macro Strategy*.

The list of categories was derived from the three design episodes that were analysed and is not intended to be a complete list of categories that might be used to describe design episodes in general.

3.3.1. Problem Domain

The first coding column describes where the designer is within the problem domain. The code consists of a numeral and one or two letters. As this is dependent on the individual problem and how the designer has partitioned the problem, different categories have been used for the first and third episodes and for the second episode.

Problem Domain for First and Third Design Episodes

Level of Abstraction	
0 - System	The designer is considering the system as a whole.
1 - Interactions	The designer is considering the interactions between the sub-systems.
2 - Sub-systems	The designer is considering details of the sub-systems.
3 - Details	The designer is considering the detailed workings of a sub-system.
R - Requirements	The designer is modifying or reconsidering aspects of the initial requirements.
Function Behaviour Structure	
F - Function	The designer is working with the function aspects of the problem domain.
B - Behaviour	The designer is working with the behaviour aspects of the problem domain.
S - Structure	The designer is working with the structure aspects of the problem domain.

Table 2. The problem domain coding used in the first and third design episodes.

Table 2 lists the problem domain categories used for the first and third design episodes. The designer does not make an explicit partitioning of the design problem but four different levels of abstraction can be discerned. *Levels of Abstraction* are denoted by the numerals *0* to *3* to represent the range from system level down to the detailed level. Letters were used to denote *Function (F), Behaviour (B) or Structure (S)* corresponding to the definitions given above. An additional letter *(R)* was used to indicate that the designer is addressing the design *Requirements*.

Problem Domain for Second Design Episode

Level of Abstraction	
0 - System	The designer is considering the system as a whole.
1 - Input Block	The designer is considering the input block of the problem.
2 - PAL Block	The designer is considering the main PAL block of the problem.
3 - Output Block	The designer is considering the output block of the system.
R - Requirements	The designer is modifying or reconsidering aspects of the initial requirements.
Function Behaviour Structure	
F - Function	The designer is working with the function aspects of the problem domain.
B - Behaviour	The designer is working with the behaviour aspects of the problem domain.
S - Structure	The designer is working with the structure aspects of the problem domain.

Table 3. The problem domain coding used in the second design episode.

Table 3 shows the problem domain categories used for the second design episode. The designer started the design episode by decomposing the problem into three sub-problems. Numerals are used to represent the *Level of Abstraction* and for the second design episode they reflect the designer's decomposition. *0* is used to denote the top level of abstraction where the designer is considering the problem as a whole. *1* to *3* are used to refer to the sub problems identified by the designer.

Two additional categories are introduced in conjunction with *Function, Behaviour or Structure* for the second design episode. The *Behaviour* category was separated into the *Expense (E)* category, used where the designer is concerned with the cost of the artefact and the general *Behaviour (B)* category used in this case to indicate reasoning about all other behaviour. *Structure* is also divided between *Schematic Structure (S)* and *Physical Structure (P).* Schematic Structure refers to structure in the sense of a device's electrical descriptions and Physical Structure concerns the mechanical descriptions of a device. These arose from the problem domain and were readily distinguished and were therefore used to increase the richness of the representation.

Table 1 demonstrates the encoding of Function, Behaviour and Structure in the second design episode. At 21:30 the designer identifies a functional requirement at a high level so this segment is encoded with the letter *F*. The letter *R* is added to indicate that a requirement has been addressed. At 21:32 the designer proposes some structure (an external pull-up resistor and voltage) and justifies this decision (21:56) with a reference to a functional concern (the ability for the user to determine the voltage). An elaboration of the proposed structure follows and at 22:18 the designer analyses the behaviour of the proposed structure. The changes in designer intentions are recorded by the letters in the problem domain (PD) column.

Reasoning about structure is readily identified when the designer proposes new resistors at 21:42 or elaborates on the connection of the resistors at 22:04. The distinction between reasoning about behaviour and function is more subtle. The justification at 21:56 falls into the category of reasoning about function since the designer is concerned with the *use* of the system. At 22:18 the designer is concerned with the behaviour of the system quite independently of the final purpose of the system.

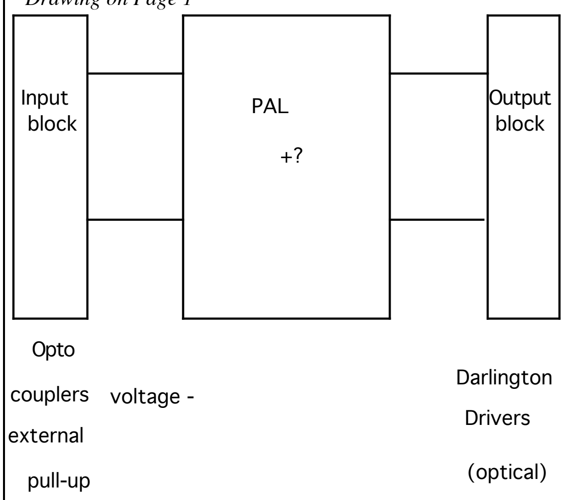
Time	PD	Mi	Ma	Dialogue	Actions
00:38	0S	Ps	De	but I guess what we need is ... some sort of input block there. The PAL, there might be one or two other bits around the PAL, I don't know and the output block. And that's the fundamental picture of what we're going to have to do.	*Drawing on Page 1* 
01:04	3S	La	De	Now its likely that the output block will be fairly trivial because	
01:10	3S	Ps	De	my guess is that I shall use some sort of ... darlington driver ... if at all possible, an optical darlington driver.	
01:27	3S	La	De	I don't think there's going to be problem with doing that.	
01:30	1S	Ps	De	The input block is ... really fairly straight forward ... opto couplers	
01:42	R1S	Ps	De	With of course external pull ups I guess so that we can operate on any voltage.	
01:48	R1F	Ju	De	That's one of the ideas of putting that input block onto ???? not only the safety side but the flexibility side as well.	
				That's the other reason of course for opticals on that side.	
02:05	2S	Ps	De	The PAL, probably just a PAL,	
02:09	2S	La	De	it might be that we need some other logic it depends on exactly which PAL we choose ...	
02:14	2S	Ds	Td	I suppose that is really the first thing to do because exactly which PAL I choose really	

Table 4 shows another protocol sequence from the second design episode. The designer begins at a top level of abstraction by decomposing the overall system (00:38) into three sub-problems, the input block, the 'PAL' block and the output block. He then moves down a level to elaborate the requirements of each of these individual blocks. First the output block (01:04) then the input block (01:30) and then the PAL block (02:05). At any time during this design episode it is possible to easily identify which part of the problem the designer is working on due to the fact that the designer identifies that his attention is shifting with comments such as “Now its likely that the output block will...” (01:04) The numerals in the problem domain column reflect these movements through the problem domain.

For the purposes of comparison between the second design episode and the other two design episodes the *Expense (E)* and *Behaviour (B)* categories can be combined as *Behaviour* as can the *Schematic Structure (S)* and *Physical Structure (P)* into *Structure*. In the *Levels of Abstraction* classification the categories can not be readily compared between the second and other design episodes since they relate to the specific structure of the proposed designs.

3.3.2. Micro Strategies

The categories used in the Micro Strategies classification emerge from the design protocols. The categories can be classified into three groups: analysing a solution; proposing a solution and making explicit references. The first two classifications are found in much of the literature on design.[22,23,24] Table 5 shows each category used in the Micro Strategy classification with a brief description and an example taken from the transcripts. The table is divided into the three groups mentioned above. Many of the definitions are clear but some of the categories are clarified below.

The first eight categories refer to the proposal of a solution or partial solution. The first four of these are very similar to each other. *Proposing a Solution (Ps)* is self explanatory. *Clarifying a Solution (Cl)* indicates that the designer is reiterating a previously proposed structure and perhaps elaborating the details of the structure. *Retracting a Previous Solution (Re)* means that the designer has rejected a whole proposed solution as opposed to modifying a solution by varying parts of it. *Making a Design Decision (Dd)* comes at the end of a period of considering alternatives. It is characterised by a decision without further elaboration of the proposed structure.

The next five categories relate to actions involving analysis of some behaviour. *Analysing a Proposed Solution (An)* indicates that the designer is analysing, qualitatively or quantitatively, a solution idea. This may be in the form of calculations or as a run through of expected behaviour. *Justifying a Proposed Solution (Ju)* does not involve calculations or a run through but the designer makes some comment that indicates that some assessment of the behaviour of a proposed solution has been made. *Evaluating a Proposed Solution (Ev)* differs from the other categories in that it involves some type of value judgement of the proposed solution.

Consulting External Information (Co) is used to denote that the designer is consulting other information to look for options for the solution. It is not used when the designer is analysing some aspect of the external information to gain a greater understanding of a structure’s behaviour. *Looking Ahead (La)* differs from *Postponing a Design Action (Pp)* in that it means the designer is identifying some future structure that will be required whereas the latter indicates that a need for some structure has been identified but its elaboration has been postponed in favour of another, perhaps easier, task. Modifying a solution by varying parts of it is denoted as *Looking Back (Lb).*

Table 5. Micro strategies used in the three design episodes.

The last three categories are used to indicate that the designer is explicitly referring to something. These correspond to times when the designer is not directly engaging the design task. *Application Knowledge (Ka)* refers to knowledge of the application or environment in which the artefact is to be used. *Domain Knowledge (Kd)* refers to knowledge of the domain of the design, here it is electronics in the first three protocols. *Design Strategy (Ds)* identifies when the designer is commenting on the progress of the design episode or is assessing his own design strategies.

The protocol sequence in Table 1 begins with a statement of the requirement for a low input impedance (21:30). At 21:32 the designer proposes an external pull-up resistor to address this requirement. At 21:56 he identifies the fact that this will allow the user to vary the pull-up voltage and then returns to the details of his solution (22:04). At 22:18 he begins a detailed analysis of the proposed circuit to confirm that it meets the requirement.

This protocol sequence of approximately one minute duration and consisting of 119 words can be effectively encapsulated by representing it as a sequence of five micro strategy segments recorded in the micro strategy (Mi) column. Firstly there is justification of the statement to follow (*Ju* at 21:30) then a proposal of a partial solution (*Ps* at 21:32), a further justification (*Ju* at 21:56) then back to the proposal (*Ps* at 22:04) and then to more detailed analysis of the proposed solution with *An* at 22:18.

3.3.3. Macro Strategies

The Macro Strategies emerged from the protocols in a similar fashion to the Micro Strategies but are necessarily much more closely linked to models of design. Five distinct categories were identified. The Macro Strategies generally extend over a number of event segments. *Top Down (Td)* refers to the process where the designer is following the approach of elaborating the desired functions and behaviours and in the process is identifying sub-goals which are then addressed. In *Bottom Up (Bu)* mode the designer is trying a number of different configurations of structure and examining their behaviour to find a match with the design requirements. *Decomposing the Problem (De)* involves the decomposition of either the overall goals or the potential system prior to Top Down design.

Backtracking (Bt) and *Opportunistic (Op)* strategies while occurring over shorter periods are related to the long term processes of the design episode. They occur when the designer has identified that a current approach needs to be modified. *Backtracking (Bt)* occurs when the designer is not achieving what has been expected and is not sure of how to proceed. As a consequence the designer goes back over existing work, possibly changing it. *Opportunistic (Op)* strategies occur when there is an external influence that makes a change of direction advantageous. Table 4 shows a sequence where the designer begins with problem decomposition at 00:38 and then moves into top down strategy at 02:14. These codes are recorded in the macro strategy (Ma) column.

3.4. Method

The process of finalising a protocol analysis can be divided into two stages. The first stage follows an approach similar to the Delphi Method to arrive at a more objective representation of the design session. The second stage involves the processing of the results and the representation of them in graphical form allowing for comparisons of the design processes.

3.4.1. Delphi Method

The Delphi Method is defined by Linstone and Turoff[25] as follows:

> *Delphi may be characterised as a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem.*

The Delphi Method has been applied to a diverse range of applications and there are a number of different approaches that can be considered as following the Delphi Method. The essential feature of the different approaches are that they consist of four phases.[25]

- In the first phase the group explores the issue and individuals contribute additional information they find pertinent to the issue.
- The second phase involves the process of reaching an understanding of how the group views the issue.
- In the third phase if there is any significant disagreement then this is explored to bring out underlying reasons for differences and to evaluate them.
- The fourth phase involves a final evaluation of all previously gathered information and evaluations.

The coding method used encapsulates these four stages although in this case there is only one coder. The sequence of events is summarised in Figure 1. Each protocol is encoded twice. During the coding process the coder refers mainly to the transcripts (these illustrate the designers actions as well as words) but refers to the video recordings when there is ambiguity in the transcript. The four dimensions (two *Problem Domain* and two *Strategies*) are allocated separately. There is at least a ten day break between the first coding exercise and the second coding exercise. The three protocols are encoded sequentially so that in the ten day intervening period two other different protocols are encoded. Throughout the process the coding scheme continues to be developed.

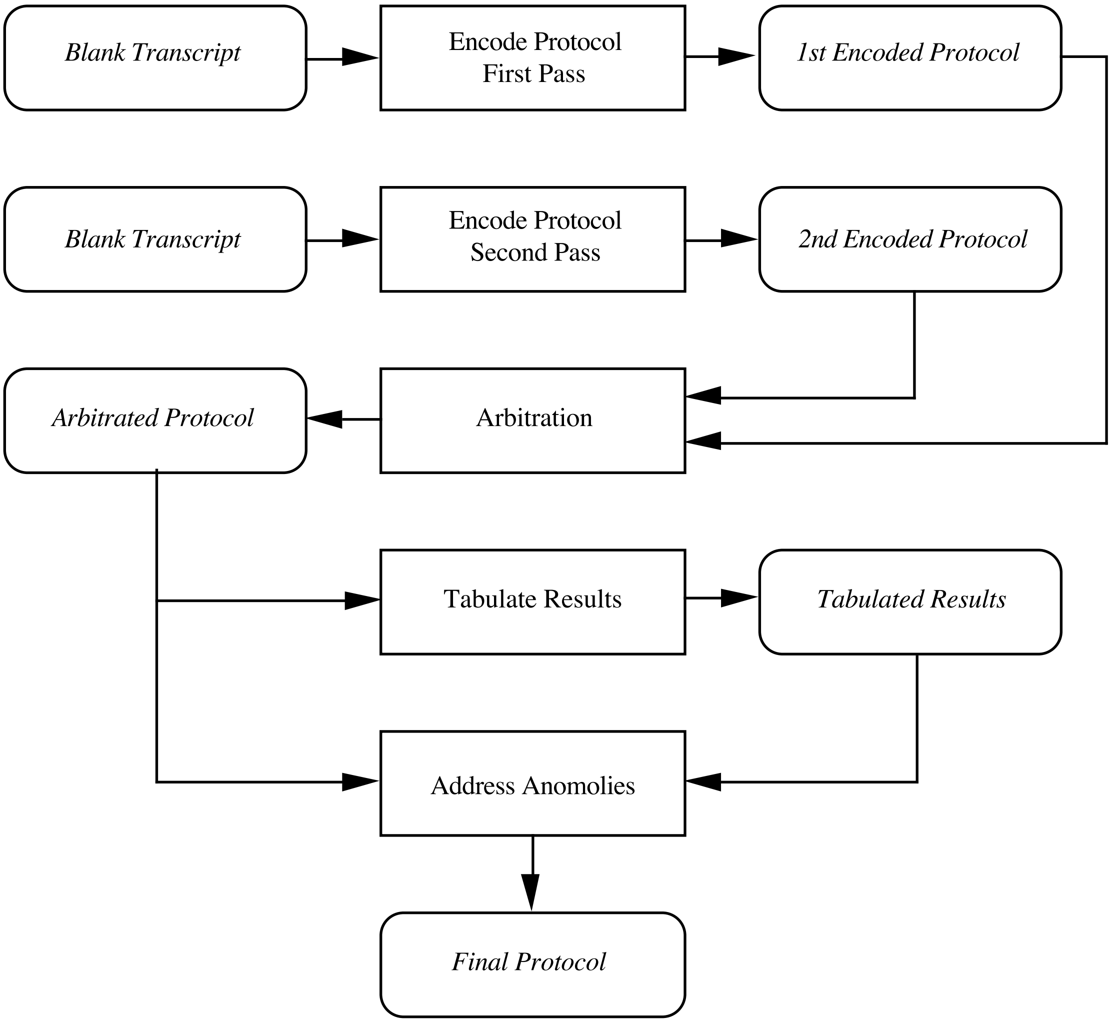


Fig. 1. The coding method.

The ten day break addresses the issue of the coder being fixated with the first analysis and repeating the first result. This improves the objectivity of the results and increases the independence of the two results. This represents the first phase described above with the first and second protocols representing differing views of the same data.

After a further ten day break the two encoded transcripts are compared and arbitrated to produce a single protocol. During this stage the differences between the first and second protocols are identified and examined more closely. The method thus highlights areas in the protocols where ambiguities may exist or the coding scheme may need improvement. The arbitration stage represents the second and third phases of the Delphi Method.

The differences between the first and second protocols fall into one of two types. The first is where segmentation varies between protocols. These may be minor segmentation discrepancies where a slightly different part of the designer's verbalisation has been used as the starting point of a segment. There may also be major segmentation discrepancies where a segment is identified in one protocol and not in the other. The second type of discrepancy is between the categories attributed to a segment which we call a coding discrepancy. This will highlight either an ambiguous segment that requires more careful consideration or it may highlight a shortcoming in the coding scheme that needs to be addressed.

Where there is a discrepancy, the original interpretations are reconsidered and the coder either decides on one of the two existing results or decides that a different category or segmentation is appropriate. Table 6 shows an excerpt of the arbitrated protocol for the second design episode. The Actions column has been removed and four sets of coding columns are used. The first two correspond to the first and second protocols with the third being the arbitrated (Final) protocol. The fourth (Re-Coding) set of columns is discussed later.

	1st			2nd			Final			Re-Coding			
Time	**PD**	**Mi**	**Ma**	**PD**	**Mi**	**Ma**	**PD**	**Mi**	**Ma**	**PD**	**Mi**	**Ma**	**Dialogue**
04:43	2S	Co	De	2S	Co	Td	2S	Co	Td				I can see here for example that the 16L8.
04:48	2S	Kd	De										I said 16 inputs can be taken with a pinch of salt its actually 16 input or output pins there's some subtlety with being able to feed some of these things back internally ...
05:00	2S	Co	De										There's a connection from the output back in so these outputs can be inputs but I'm not interested in using that so in fact
05:13				2S	Ps	Td							This is a bit of a turn up for the book the very first thing I choose ... does seem to be quite a useful little device and
05:25	2S	Ds	Td	2S	Ds	Td	2P	Ds	Td				the other thing to determine of course is whether we can be really sneaky with this and not actually make a decision on which PAL to use ...
05:36				2S	Ju	Td	2P	Ju	Td	2B	Ju	Td	by that I mean ...
05:40	2P	Ps	Td										if the pinouts are the same ... or essentially the same on all of them
05:45							2S	Ps	Td				I don't actually need to specify a particular PAL

Table 6. Example of coding history for second design episode.

The segment beginning at 05:25 is an example of full agreement between the first and second protocols. The arbitrated (Final) protocol follows the first two protocols with the exception of the *FBS* dimension of the *Problem Domain*. The change from an *S (Schematic Structure)* category to a *P (Physical Structure)* category has been prompted by the need to more fully define these two categories to resolve a conflict elsewhere in the protocol.

There is an example of segmentation discrepancy at 05:36 and 05:40. In the first protocol the coder has started the new segment after the words "by that I mean" whereas in the second protocol the new segment starts before these words. There is also a coding discrepancy in the *FBS* categories and the *Micro Strategy* codes, a *Ps* category (a proposing type) in the first protocol and a *Ju* category (an analysing type) in the second protocol. Upon reflection the coder realised that there was a discrepancy due to the fact that there were two segments where the coder had identified only one segment in each coding. The arbitrated result is two segments, a *Ju (Justification)* followed by a *Ps (Proposal)*.

The next stage of the process, corresponding to the fourth phase of the Delphi Method, involves tabulating the results to show only the arbitrated categories and a final evaluation of the results. Table 7 shows an excerpt of the arbitrated table. The categories are represented by columns and the rows represent segments in the design episode. The starting time for each segment is shown in the Time column. In this table the *Expense (E)* and *Behaviour (B)* categories have been combined as *Behaviour* and the *Schematic Structure (S)* and *Physical Structure (P)* as *Structure*. The continuous lines show that the segments form some continuous sequence in that dimension whereas an 'x' signifies change of attention or strategy perhaps within the same category.

When represented in this format it is easy to see the relationships between categories in different dimensions. A relationship exists between the *Micro Strategy* dimension and between the *FBS* dimension. Our model of design suggests that proposing *Micro Strategies* relate to *Structure* in the *Problem Domain* and analysing *Micro Strategies* to *Behaviour* in the *Problem Domain.* Table 7 shows a contradiction to this at 05:36 where a *Ju (Justification)* category is found with a *Structure* category. This implies that the designer is making an analysis type of action with structure. This prompts a re-evaluation of the protocol where it is found that the designer's comments on "pinouts" prompted the *P (Physical Structure)* category but on closer examination the designer is considering the behaviour of the "pinouts". ie "if the pinouts are the same..." This results in the *P* category being changed to a *B (Behaviour)* category.

Anomalies of this type were quite rare in the protocols. There were only three such anomalies in the second design episode. In each case the changes are noted in the Re-coding column of the Arbitrated protocol (as shown in Table 6) and a note explaining the change is attached to the end of the protocol. This results in a final protocol that shows the full history of the coding sequence.

Time	Problem Domain								Micro Strategy																Macro				
	R	0	1	2	3	F	B	S	Pa	Ev	Ju	Ca	An	Ps	Cl	Re	Dd	Co	Pp	La	Lb	Ka	Kd	Ds	Td	Bu	Bt	Op	De
3:36																							x						
4:07																			x										
4:25																								x					
4:43																		x											
5:25								x																x					
5:36								x			x																		
5:45								x						x															
5:51								x															x						
6:03								x										x											
6:44								x																x					
6:49																		x											
7:36																								x					
7:37																		x											
8:12														x															
8:25																							x						

Table 7. Activity and event across time for second design episode.

3.5. Delft Protocol

The coding method used for the Delft protocol differed to that described above in that a second coder was used. Each coder coded the protocol twice and then self arbitrated before both coders arbitrated their results to produce a final protocol. In addition some changes were made to the categories used for the coding. Three Level of Abstraction categories were used rather than four. This was due to the fact that three levels were more readily discernible. The Function Behaviour and Structure categories were the same as the three electronic design protocols.

The micro strategies included a fourth group of categories related to analysis of the problem. The fact that the designer was seeing the problem for the first time meant that the designer was spending a more of his time on analysing the problem. The categories follow closely the categories used for analysis of the solution. An additional category *(X)* was used to denote when the experimenter was talking. Table 8 shows a summary of the codes used with examples given for each. A full explanation of this coding scheme appears in Purcell et al.[26]

Micro Strategy Categories for the Delft Design Episode

Analysing Problem

Ap - *Analysing the Problem* "What is the system going to need to do...."

Cp - *Consulting Information about the Problem* "The brief says it has to be light and..."

Ep - *Evaluating the Problem* "That's an important requirement...."

Pp - *Postponing Analysis of the Problem* "I can find that out later"

Table 8. Micro strategies used for the Delft design episode.

4. PRIMARY RESULTS

The consistency of the coding process was checked and the coding results were represented graphically to give an overall summary view of the design episodes.

4.1. Coding consistency

By comparing the results achieved at each stage in the coding process it is possible to assess the robustness of the approach and to identify areas within the approach that may be improved. It can also be used to give an indication of the validity of the results. The consistency of the coding method was assessed by comparing each of the protocols with each other to establish the level of agreement between protocols.

As mentioned discrepancies fall into one of the following three groups: minor segmentation discrepancies; major segmentation discrepancies and coding discrepancies. For minor segmentation discrepancies where a transition is marked as occurring in a slightly different part of an utterance, as at 05:36 and 05:40 in Table 6, then the segmentation is taken to be in agreement and the coding of these two segments can be directly compared.

Major segmentation discrepancies are treated as separate coding events and the decision as to agreement between protocols is more subtle. The section from 04:43 to 05:25 in Table 6 shows a number of major segmentation discrepancies. In the first protocol three segments were identified, in the second two were identified and in the arbitrated protocol one segment was decided upon. For the purposes of comparison the section is divided into four segments, the first from 04:43 to 04:48, the second from 04:48 to 05:00, the third from 05:00 to 05:13 and the fourth from 05:13 to 05:25. Where a code has not been entered for a particular segment in one of the protocols the previous code is retained; eg the second segment in the second protocol would be "*2S-Co-Td*".

For this example we will consider only the Micro Strategy columns. All protocols agree in the first segment (04:43 - 04:48). In the second segment (04:48 - 05:00), the first protocol records a change to *Kd* while second and arbitrated protocols retain the *Co* code from the first segment, so a disagreement is recorded between first and second protocols and first and arbitrated protocols and agreement is recorded between second and arbitrated protocols. In the third segment (05:00 - 05:13) the protocols again all agree since the second and arbitrated protocols again retain the *Co* code from the first segment. In the fourth segment (05:13 - 05:25) the first and third protocols now retain the *Co* code and the second protocol has changed to *Ps*. This is recorded as agreement between first and arbitrated protocols and disagreement between first and second protocols and second and arbitrated protocols.

The protocols were assessed using the above approach and agreement between protocols was recorded as a percentage for each design episode and as an average over the three design episodes and are recorded in Table 9. The table shows raw agreement between protocols and does not take into consideration marginal frequencies of the coding categories. Since the coding scheme was being developed while the protocols were being coded a detailed analysis of consistency is not appropriate. Future applications of the coding method will be more fully assessed using a method such as is suggested by van Someren et al 1994.

The results show a high level of agreement for each of the design episodes and overall. Agreement between the first and second protocols is the lowest. The arbitrated protocol agrees more closely with the second protocol than with the first. The levels of agreement reflect the fact that the coding scheme was evolving during the coding process.

	1st & 2nd	1st & Arbitrated	2nd & Arbitrated
D.E.1	63%	81%	77%
D.E.2	67%	77%	89%
D.E.3	58%	71%	84%
Overall	63%	76%	83%

Table 9. Coding consistency between the three design episodes.

4.2. Activity vs Time

For each of the protocols the results of the coding are recorded on a single graph. Each of the coding dimensions is plotted against time. This includes the time axis that represents the segment lengths in the context of the overall design episode. The explicit categories (*Kd, Ka, Ds*)in the micro strategy dimension have been omitted so that the graphs can be easily understood.

4.2.1. First Design Episode

Figure 2 shows the activity verses time graph for the first design episode. In the macro strategy dimension the designer commences with a bottom up strategy (*Bu*) with occasional periods of decomposing the problem (*De*). At about twelve minutes the designer realises that his current approach is not appropriate and he backtracks (*Bt*). This backtracking is followed by a period of decomposition before returning to a bottom up strategy. At 38 minutes he moves to a top down strategy (*Td*). At this point the designer is revising his design in a methodical manner. At 49 minutes the designer moves back to a bottom up strategy followed by an opportunistic deviation (*Op*) when he realises his existing design is incomplete.

In the micro strategy dimension the designer begins by proposing part of the solution (*Ps*) before moving quite quickly to a long period of analysis (*An*). This is followed by a long period of cycling between proposal of a solution and analysis. During this time the designer occasionally makes use of some of the other micro strategies. At 38 minutes the designer enters quite a long period of mainly clarification (*Cl*) followed by mainly solution proposal (*Ps*). Around 49 minutes the designer makes use of looking ahead (*La*) and looking back (*Lb*) micro strategies. The design episode ends with a short period of analysis.

Correlations can be seen between the macro strategy and the micro strategy. The periods of problem decomposition correspond with the analysis micro strategy. The period of bottom up macro strategy corresponds to cycling between proposal and analysis in the micro strategy. The top down period corresponds to clarification of the proposed design followed by further solution proposal. The opportunistic deviation corresponds with a period of looking back in the micro strategy.

The function-behaviour-structure dimension follows the micro strategy dimension in terms of the proposing micro strategies correspond to reasoning in structure (*S*) while the analysing micro strategies correspond to function (*F*) or behaviour (*B*). In this particular design episode there is no reasoning in function. This due to the fact that the designer is working on a sub problem of a larger system design and this particular sub problem has no relation to the use of the system. This dimension shows cycling up until 38 minutes and then the designer is reasoning principally about structure. The level of abstraction dimension involves reasoning mainly at the sub-system level (2). The first twelve minutes involve the designer working at each of the four levels. After 45 minutes the designer considers some of the detailed level of abstraction (3). Also absent in this design episode is any reference to design requirements.

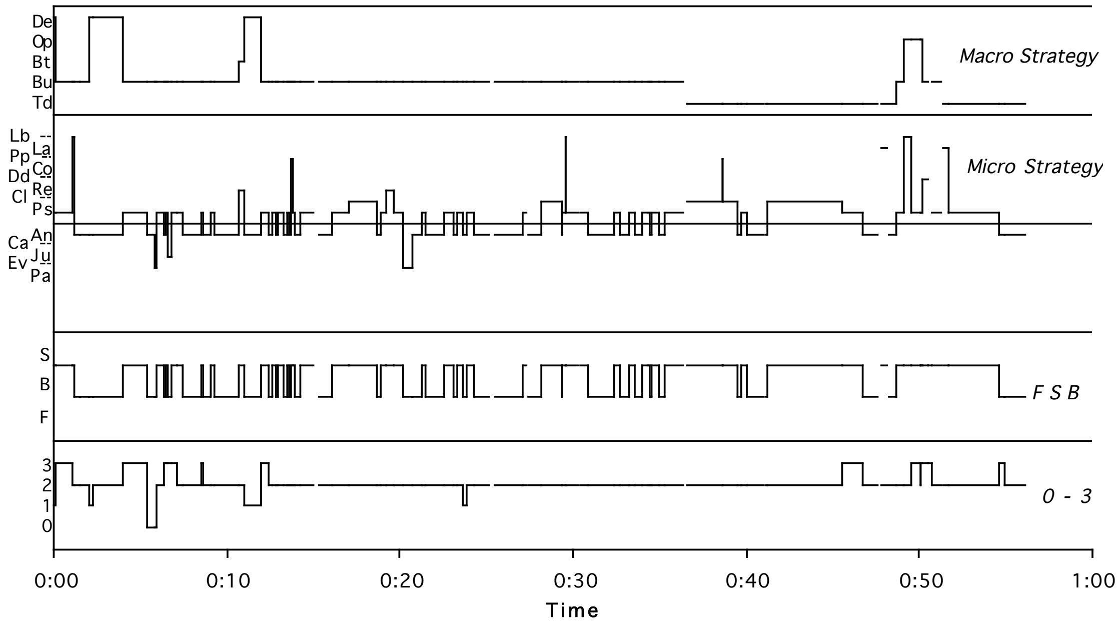


Fig. 2. Activity chart for the first design episode.

4.2.2. Second Design Episode

The second designer began by decomposing the problem (*De*) followed by about 20 minutes of top down design (*Td*). (Figure 3) During the initial decomposition the designer defines the three sub problems that he is going to work on. At 20 minutes the designer moves into a mainly bottom up strategy (*Bu*) which continues until 47 minutes. There is an extended period of opportunistic macro strategy (*Op*) before the designer moves back to a principally top down strategy.

The second designer makes use of a larger number of micro strategies than the first designer. After a short period of cycling between proposing (*Ps*) and looking ahead (*La*) the designer spends a considerable amount of time consulting external data for ideas (*Co*). This is followed by 30 minutes of cycling between proposing micro strategies and analysing micro strategies. During this time the designer makes use of every category of micro strategy except postponing analysis (*Pa*) and calculating (*Ca*). He makes a significant number of value judgments on his design (*Ev*). At 50 minutes the designer spends several minutes clarifying his design (*Cl*) coupled with long periods of analysis (*An*).

The designer begins by reasoning mainly with structure (*S*) before moving into an extended period of cycling between structure and function (*F*) or behaviour (*B*). At 48 minutes the cycling slows to the point where the designer spends several minutes in each category. The second designer spends a considerable amount of time reasoning in function. The levels of abstraction categories for this design episode follow the designer's decomposition of the problem into three sub problems. After the initial period of decomposition in which he spends a brief time considering each of the three sub problems the designer works on each in turn. First sub problem *2* then *3* followed by *1*. At 48 minutes the designer reviews his overall design and sequences through the three sub problems. Throughout the design episode the designer moves back to the system level (*0*) for short periods.

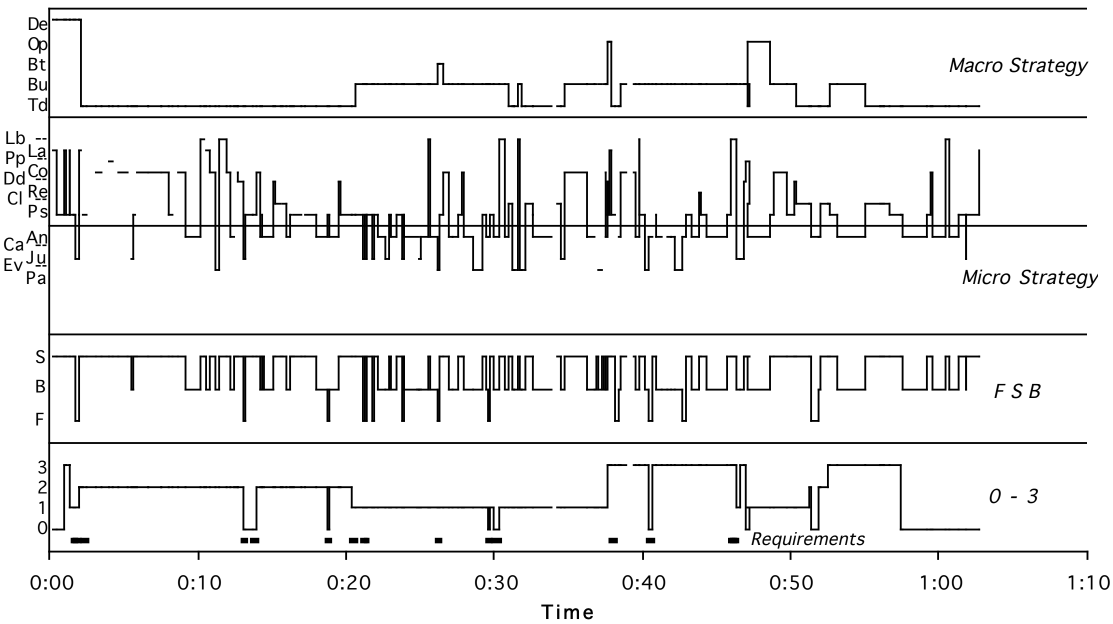


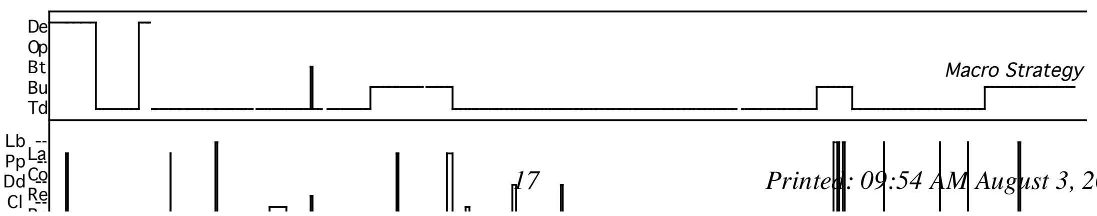
Fig. 3. Activity chart for the second design episode.

There are correlations between the four dimensions in the second design episode. The initial period of decomposition sees the designer move rapidly through the three sub problems (*1-3*) while mainly looking ahead (*La*). This is followed by 18 minutes of addressing sub problem *2* during which time the designer spends much of the initial time consulting data (*Co*) and following a top down strategy (*Td*). For the design of sub problems 2 and 3 there is cycling in the micro strategy dimension and the designer follows a bottom up approach (*Bu*). At 48 minutes the designer recaps the overall design and during this time returns to a top down strategy (*Td*). There is significant correspondence between times when the designer is considering the requirements (*R*), is reasoning at the system level (*0*) and is reasoning in function (*F*).

4.2.3. Third Design Episode

In the third design episode (Figure 4), after beginning by decomposing the problem the designer employs a top down strategy (*Td*) for most of the design episode. At 25 minutes the designer backtracks (*Bt*) briefly and then at 30 minutes he moves to a bottom up strategy (*Bu*). At one hour 14 minutes and for the last eight minutes of the design episode the designer follows a bottom up strategy (*Bu*).

The designer makes use of a limited number of micro strategies with the two main categories being proposal of a solution (*Ps*) and analysis (*An*). Cycling between proposal and analysis is predominant for most of the design episode. From 52 minutes until one hour and eight minutes the designer is engaged in a prolonged analysis (*An*) which involves extensive use of his calculator (*Ca*). From one hour and 15 minutes the designer looks back (*Lb*) seven times. During this time the cycling becomes more rapid.



There is considerable cycling in the function-behaviour-structure dimension reflecting the cycling found in the micro strategies. The long period of reasoning with behaviour (*B*) at 52 minutes corresponds with analysis (*An*) and calculating (*Ca*) in the micro strategy dimension. There are three times when the designer reasons in function (*F*). The designer begins by considering the problem at the system level (*0*) and gradually moves down to the detail level (*3*) at 28 minutes. He then works mainly in the detail level and the sub system level (*2*). Towards the end of the design session he is working in the sub system level. Each time the designer is reasoning with function (*F*) he is also considering the system level of abstraction (*0*). Two of these times also correspond with references to the system requirements (*R*).

4.2.4. Delft Design Episode

As explained in Section 4.4, a slightly different coding scheme was used for the Delft design episode. There was no coding completed for macro strategy. The micro strategy dimension included codes to differentiate between analysis of the problem and analysis of a solution (called evaluation of the problem). Figure 5 shows the results of the Delft design episode. In the first 37 minutes the designer is mainly analysing the problem (*Ap*) or consulting the design brief (*Co*). Another significant activity at this time is consultation with the experimenter indicated as blank periods in the graph.

After 37 minutes the designer is cycling rapidly between analysis categories, proposal categories and evaluation categories in the micro strategy dimension. The designer makes use of most of the micro strategies in addressing the problem.

In the initial stages the designer spends a considerable amount of time reasoning with function (*F*) followed by about 10 minutes of reasoning with behaviour (*B*). For the remainder of the design episode the designer is cycling between structure (*S*) and behaviour (*B*) with occasional periods of reasoning with function. The designer begins by considering the system level of abstraction (*0*) moving to predominantly the sub system level (*1*) and then into the detail level (*2*). Throughout the design episode the designer deviates to other levels of abstraction. The designer considers the requirements (*R*) initially and then for three other brief periods during the design episode.

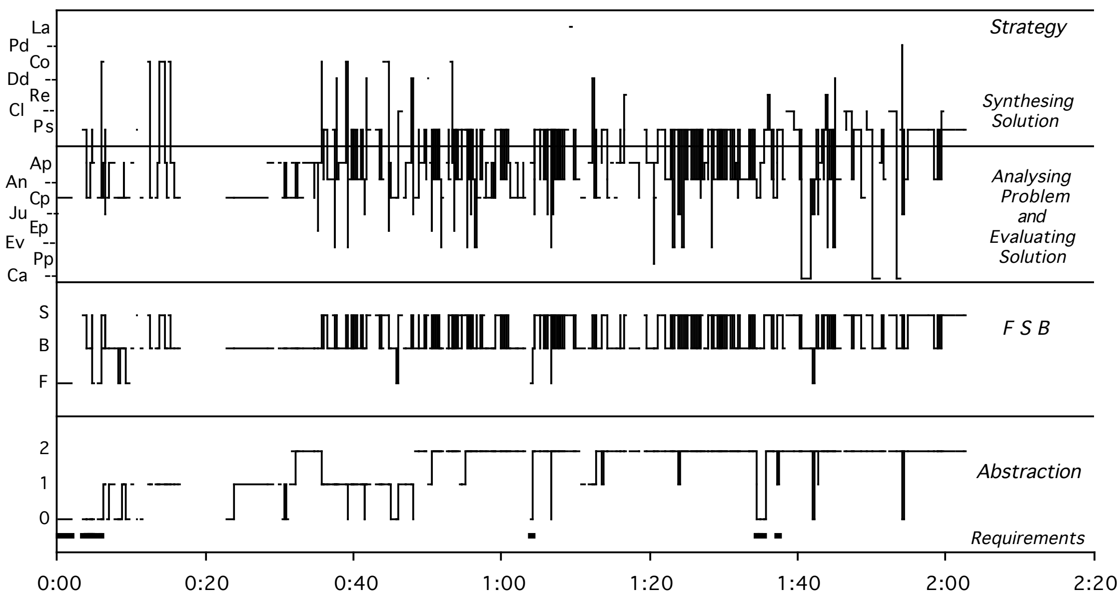


Fig. 5. Activity chart for the Delft design episode.

5. FURTHER ANALYSIS

In the last section, we demonstrated how it is possible to apply a coding scheme in a protocol analysis to obtain highly articulated analyses of the behaviour of designers as they are designing. This articulation provides the opportunity to have a much more detailed analysis of the behaviour of designers by aggregating and analysing these ‘raw’ results. The results are further explored using graphical and filtering techniques to represent the results in more useful ways. This allows comparisons between design sessions to be made more easily. The distribution of micro strategy categories used by the designers is examined. Then ways of representing the relationships between reasoning about Function, Behaviour and Structure are explored. Cycle times are also investigated.

5.1. Distribution of Micro Strategy Categories

The time spent on each category is summed for the whole design episode and graphed with each of the other categories. The time spent is represented as a percentage of the total episode time. The distributions for the first three design episodes are presented in Figure 6. The Delft analysis made use of a different set of micro strategy categories and is not presented.

In all three design episodes the time spent on Analysis of a solution (*An*) was the highest followed by time on Proposing a solution (*Ps*). In the first design episode the designer spends a considerable time clarifying his proposals (*Cl*) but in the third design episode he spends a higher proportion on each of calculating (*Ca*) and evaluating his solutions (*Ev*). This a accounted for by the fact that the first design episode occurs earlier in the design process of the larger system and much of the Structure of the system is yet to be finalised so much of the time is spent on proposing and clarifying. In the third design episode, later in the larger design process, the designer is analysing the limitation of his design so more time is spent analysing and evaluating the existing structure.

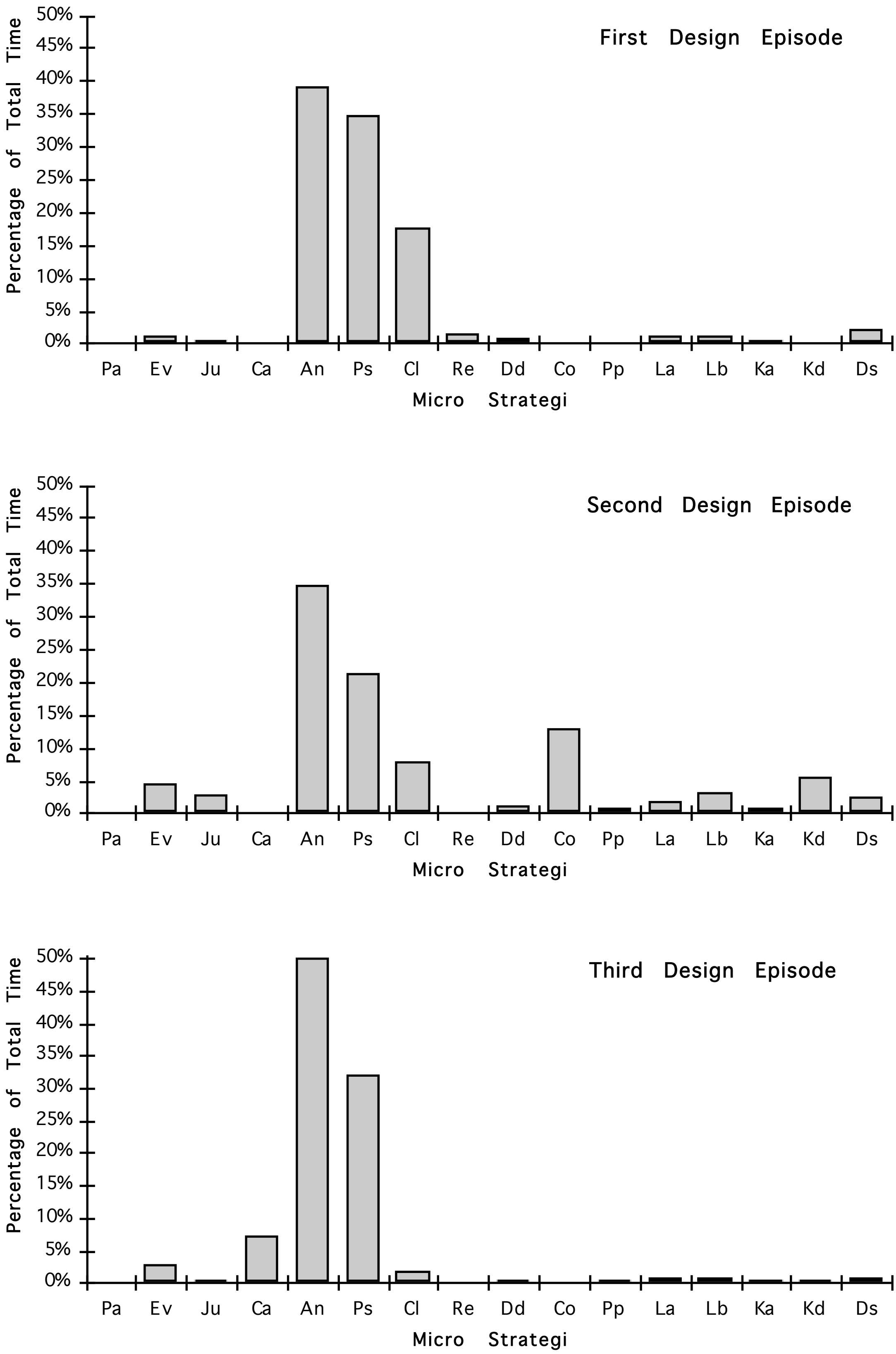


Fig. 6. Distribution of micro strategy categories for first three episodes.

In both episodes five categories account for most of the designer's time. In contrast, the second designer makes use of most of the micro strategies with proposing type categories being spread amongst proposing a solution (*Ps*), clarifying a solution (*Cl*) and consulting external data (*Co*). The designer also spends more time making value judgements on his design (*Ev*) and calling on domain knowledge (*Kd*). This use of a larger repertoire may be explained by the fact that the second designer is more experienced than the first. This provides support for the hypothesis that experienced designers use a larger repertoire of design strategies than inexperienced designers.

5.2. Reasoning with Function and Behaviour to Structure

The designers' reasoning with function, behaviour and structure is analysed to investigate similarities and differences between the designers and design episodes. Several methods of processing the data are explored. In each of the following results the time axis is expressed as a percentage of the total episode time. This is to facilitate comparisons of the designers' behaviours.

5.2.1. Moving Weighted Average

A moving weighted average is taken for each of the design episodes. Firstly the episode is divided into 400 segments of equal duration and the percentage of time in each segment for which the designer is reasoning with Function or Behaviour is calculated. Time when the designer is not dealing with the problem domain (and therefore not reasoning in any of the three areas) is not included in the calculation.

So the calculation for each segment is:

$$\%\ \text{Function \& Behaviour} = \frac{\text{Time spent reasoning with F or B}}{\text{Time spent reasoning with F, B or S}}$$

The segment length is 0.25% of the total episode length for each design episode. Since most event segments in the data last for more than 0.25% of total episode length the resultant graph would consist of points that are 0% or 100% so filtering was applied. The filter is a trapezoidal shape spanning 15% (60 segments) of the design episode as shown in Figure 7. Several filter widths were investigated and it was found that a filter width of 15% gives a balance between the general trends and the details in the design reasoning.

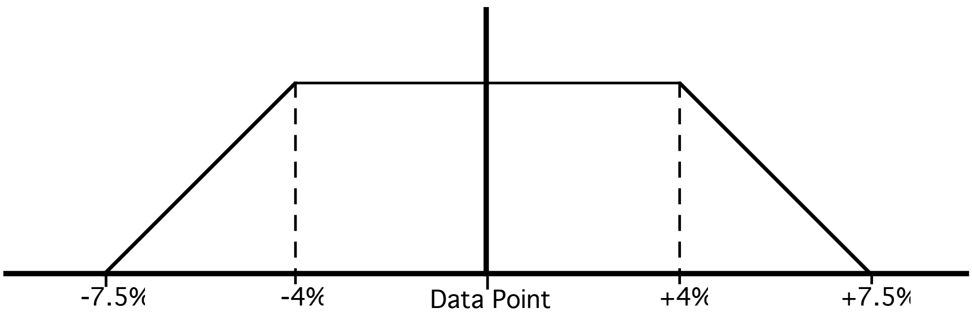
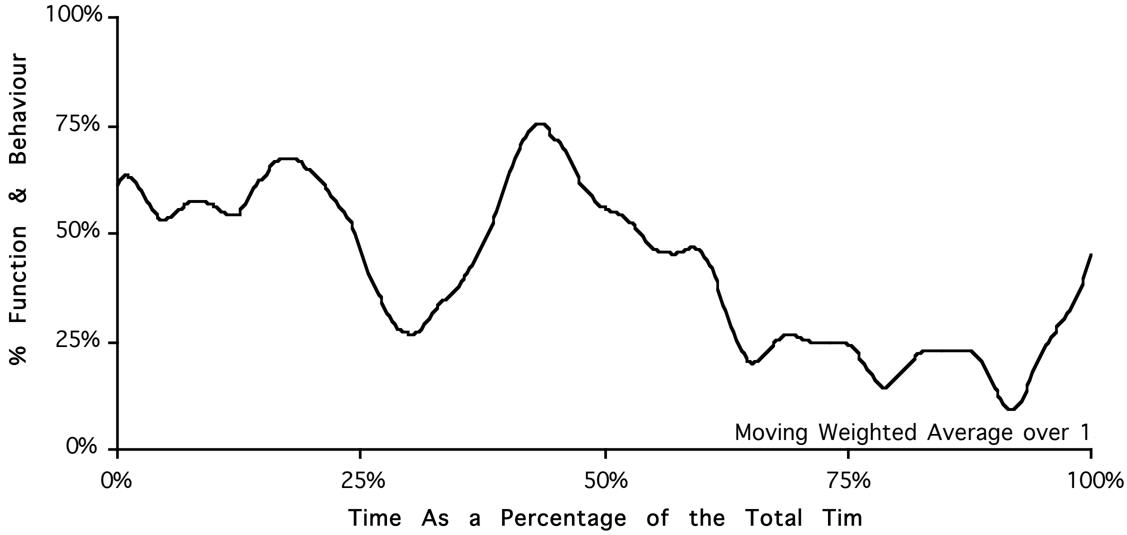


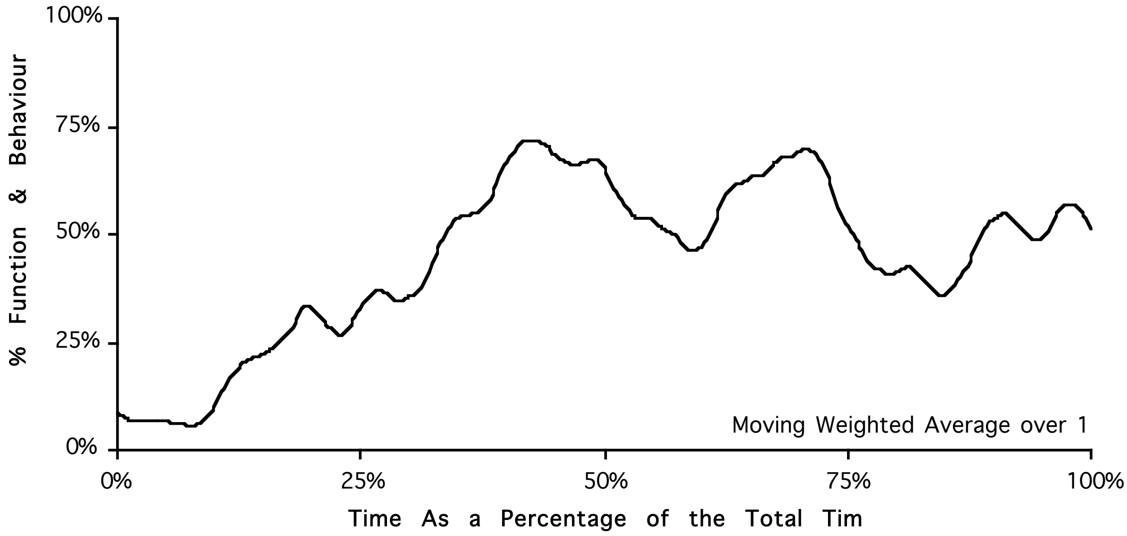
Fig. 7. Trapezoidal filter used for the moving weighted average.

The result of this filter is a smooth curve showing the general trends in the designers' reasoning that contrasts with the Function Behaviour and Structure representation shown in Figures 2 to 5. The filtered graphs (Figure 8(a) – (d)) allow for direct comparisons between the design episodes.

The *first design episode* commences with the designer spending slightly more than 50 percent of his time reasoning with Function or Behaviour corresponding to a period of familiarisation with the problem. At around 20 percent of the elapsed time he begins to spend more time reasoning with Structure as he attempts to establish how many registers will be required. His initial strategy is to consider only what information will be stored with little regard to the process that the RAM controller will need to follow. Here he is reasoning mainly with Structure.



(a) First design episode



(b) Second design episode

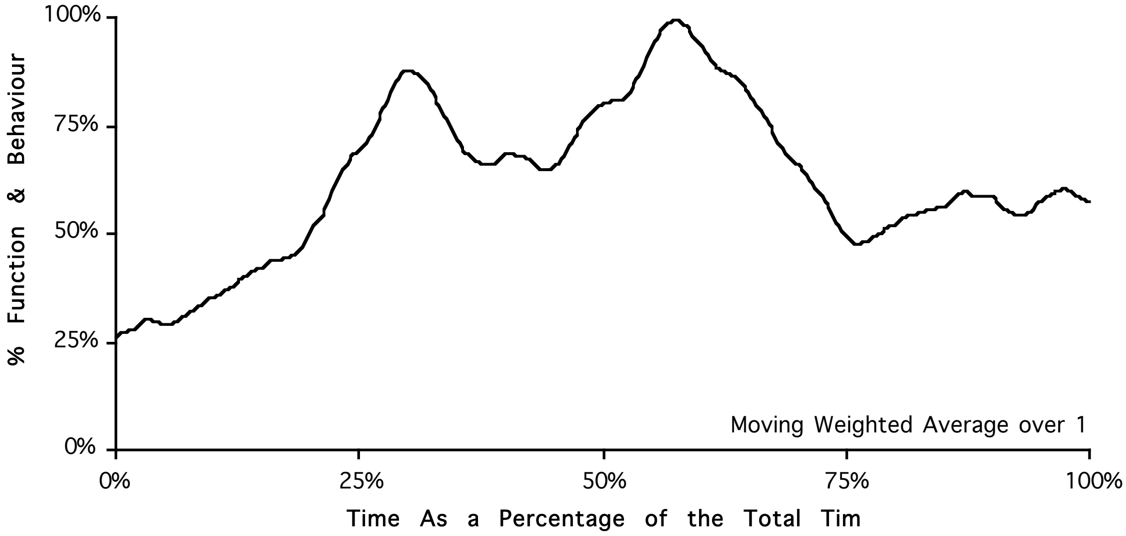
Fig. 8. Ratio of Function and Behaviour to Structure as a moving weighted average (a) first design episode, (b) second design episode.

At 27 percent of the time (15 minutes and 33 seconds) he realises that this approach will not be sufficient and he then begins to analyse the required behaviour of the system. This increased time spent reasoning with behaviour is reflected by the peak of approximately 75 percent of his time being spent on this. His reasoning then tends towards structure again as he is defining variables based on the process and assigning physical registers to the variables.

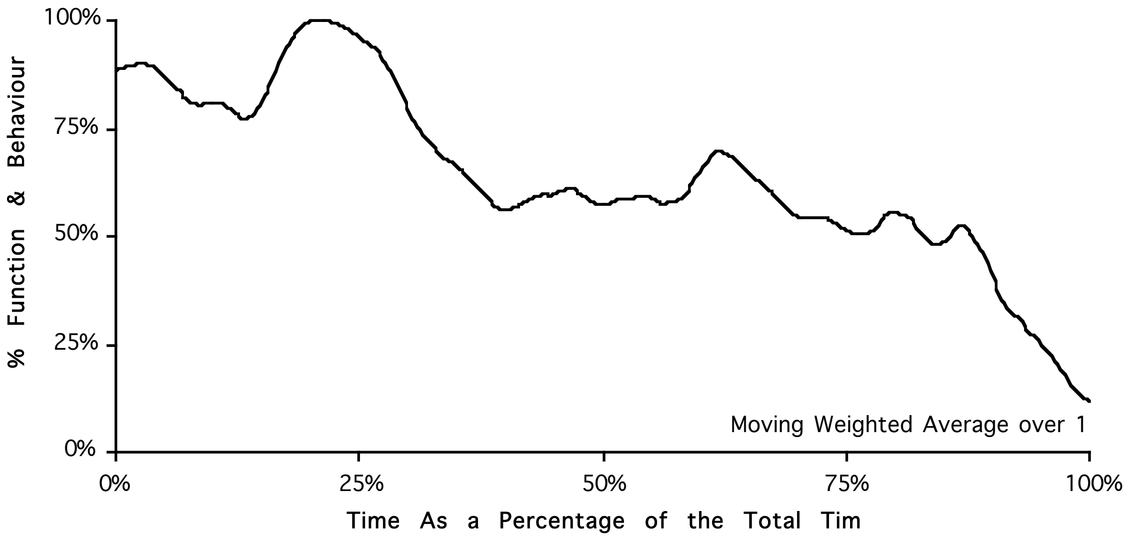
At 90 percent of the episode time the designer begins a final check of his design, an activity which involves mainly the analysis of Behaviour. This is reflected in the sharp rise towards Function and Behaviour at the end of the design episode.

In the *second design episode* the designer commences by defining the high level structure of the system and identifying three almost independent parts to be designed. The first part that he concentrates on is the PAL controller. This part of the design involves selecting the appropriate device from a catalogue of available devices and lasts until just over 30 percent of the episode time. During this time the graph, Figure 8(b), shows that there is a gradual rise from less than 10 percent to around 35 percent of the designer's activity being focussed on Function or Behaviour. This is consistent with the process of selecting a device from the catalogue and then analysing the device Behaviour to ensure that it is appropriate.

From 30 to 60 percent of the elapsed time of the design episode the designer is working on the input circuit and from 60 to 74 percent he is working on the output circuit. Both these segments on the graph, Figure 8(b), begin at a lower level of Function or Behaviour, rise to a maximum and then tail off again to wards the end of the segment. This is consistent with the designer's approach of proposing some Structure and then analysing its Behaviour then subsequently adjusting the proposed structure accordingly.



(c) Third design episode



(d) Delft design episode

After the 74 percent point the designer is recapping his design. He begins by re-drawing the whole design on one sheet and then checks the overall performance. This is consistent with the dip towards more Structure followed by a trend towards more Function or Behaviour.

The *third design episode* begins with the designer clarifying the existing structure with the intention of establishing whether the addition of the requirement of a learning mode is going to require changes to the existing structure. This corresponds to the first twenty percent of the design episode where the designer is working more with structure. From 20 to 30 percent of the time the designer is analysing the behaviour of the existing structure, reflected by the sharp rise in the graph, Figure 8(c), and at 30 percent the designer realises that he can improve the existing structure.

From around 50 to 65 percent the designer is analysing the behaviour of the system in terms of its learning mode. This involves some lengthy analysis and calculations and is reflected by the graph rising to almost 100 percent Function or Behaviour during this period. For the last 35 percent of the design episode the designer is defining registers and so on that will be used in the learning mode and is running through the learning mode to check that requirements will be met. This is reflected by an initial trend towards more Structure and then a levelling off of the graph.

The *Delft design episode* differs from the electronic design episodes in two significant ways. Firstly the Delft designer has no idea of the design task until the protocol begins whereas the electronic designers have been contemplating their designs before the design sessions. This is reflected in the much higher percentage of time spent reasoning with Function and Behaviour in the first 25 percent of the Delft protocol as the designer is coming to terms with the problem.

From 25 percent to forty percent the tends towards reasoning with Structure as he is establishing the most appropriate position for the back pack on the bike. Until the 85 percent point the designer is cycling between proposing a solution (Structure) and analysis (Behaviour).

The second difference between this episode and the electronic design episodes is that the Delft designer has a fixed time in which to produce a result. The electronic designers work at their own pace and decide on their own endpoints. At 85 percent of the episode time the Delft designer realises that he has a short time to go and spends the remaining minutes fleshing out his current design which involves mainly proposing structure. This results in the graph moving almost entirely to reasoning about Structure.

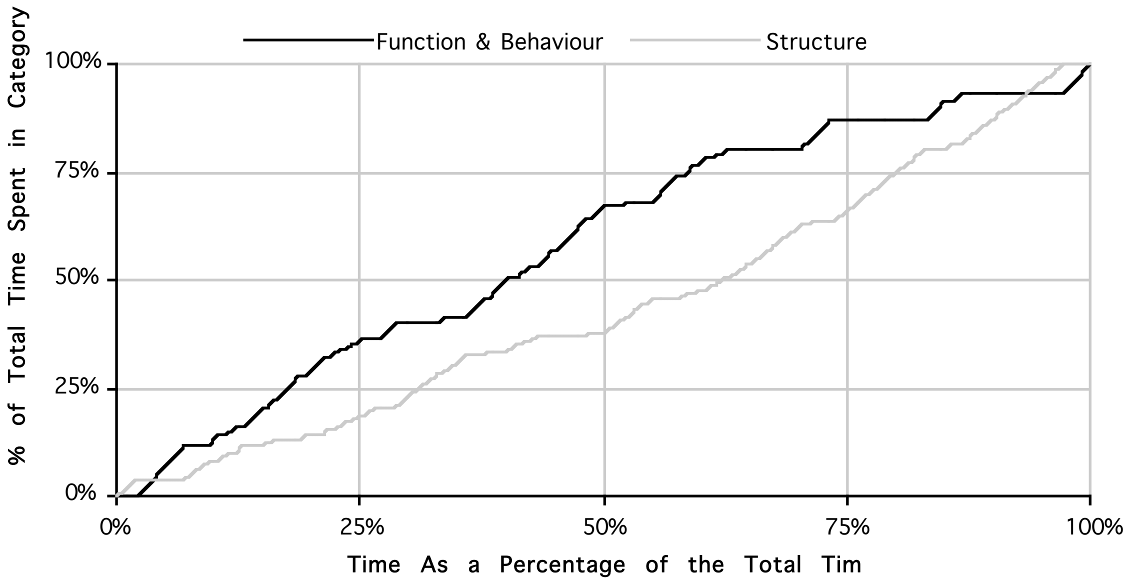
At a large granularity Function and Behaviour map onto analysis and evaluation and Structure onto synthesis in classical models of design.[27,28] The results in Figure 8 provide strong evidentiary support for the applicability of such models with the addition of a constant cycling in focus between Function and Behaviour and Structure which matches the notion of iterating between analysis, synthesis and evaluation.

These graphs can be seen as design process ‘signatures’ which can later be used to categorise designing styles.

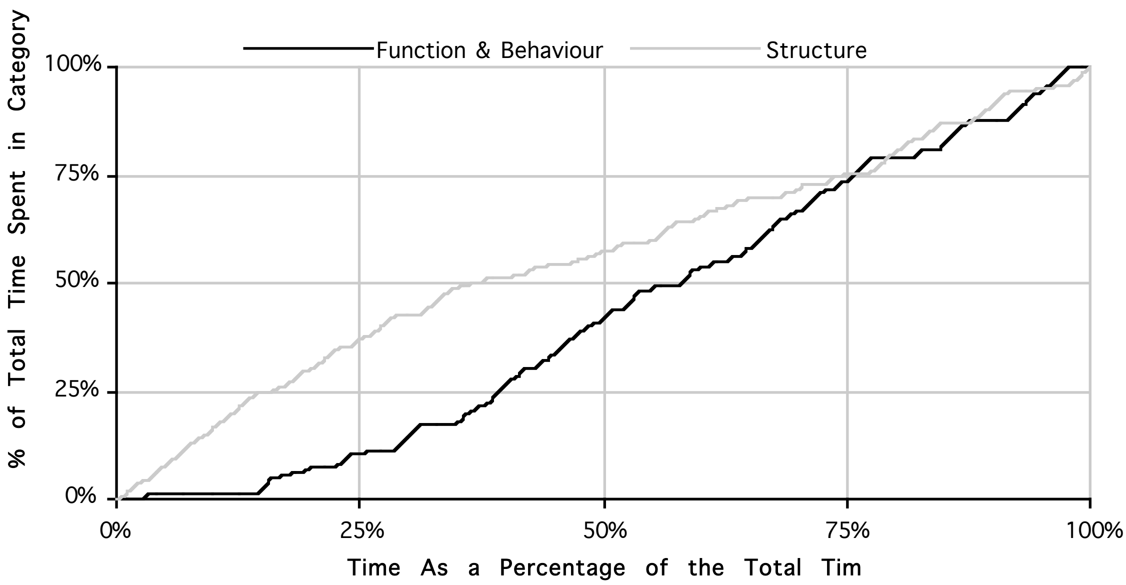
5.2.2. Time Spent Reasoning with Each Category

Another way to view the Function-Behaviour-Structure dimension is to plot the time spent reasoning with each category as a percentage of the total time spent in the category against elapsed time for a design episode. Two values are plotted, one representing Function and Behaviour and the other representing Structure. (Figure 9). Each graph's ordinate begins at 0 and ends at 100 percent. Such graphs provide information on the style of a designer in terms of where they focus in the process independent of the specific design requirements and the domain of the structures proposed. Some designers will focus primarily on Function and Behaviour whilst others will focus primarily on Structure. Others still will focus on both. The precise nature of their style will be encapsulated in the graph.

The first and Delft design episodes are similar in the way the Function and Behaviour curves precede the Structure curves. The designers have reached the 50 percent threshold for Function and Behaviour at around 40 percent of the episode time whereas the 50 percent threshold for Structure is reached at 62 percent for the first episode and 70 percent for the Delft episode. This indicates a greater focus on Function and Behaviour than Structure.



(a) First design episode

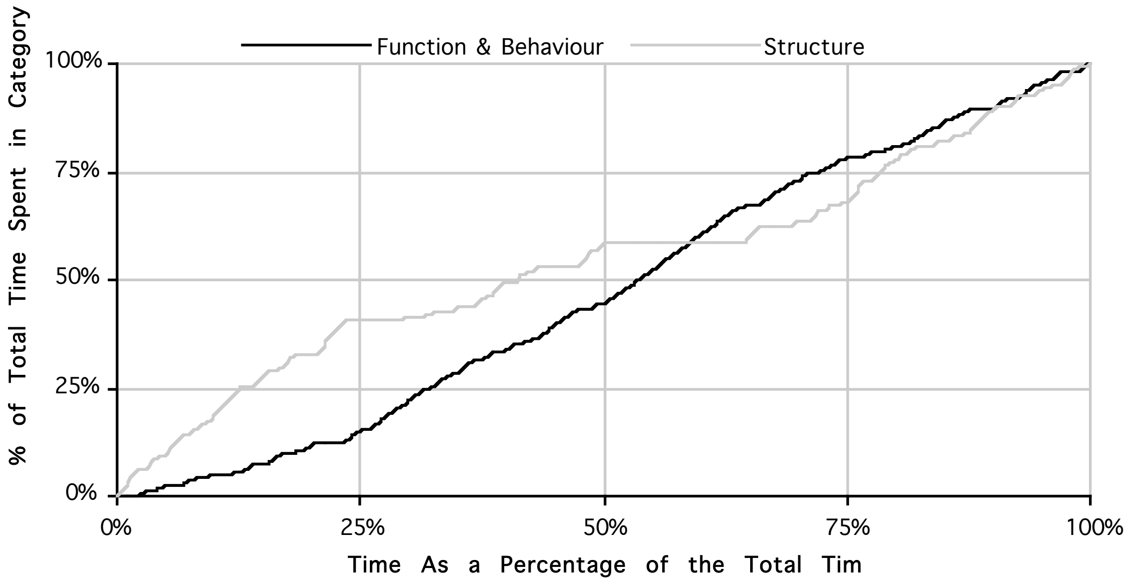


(b) Second design episode

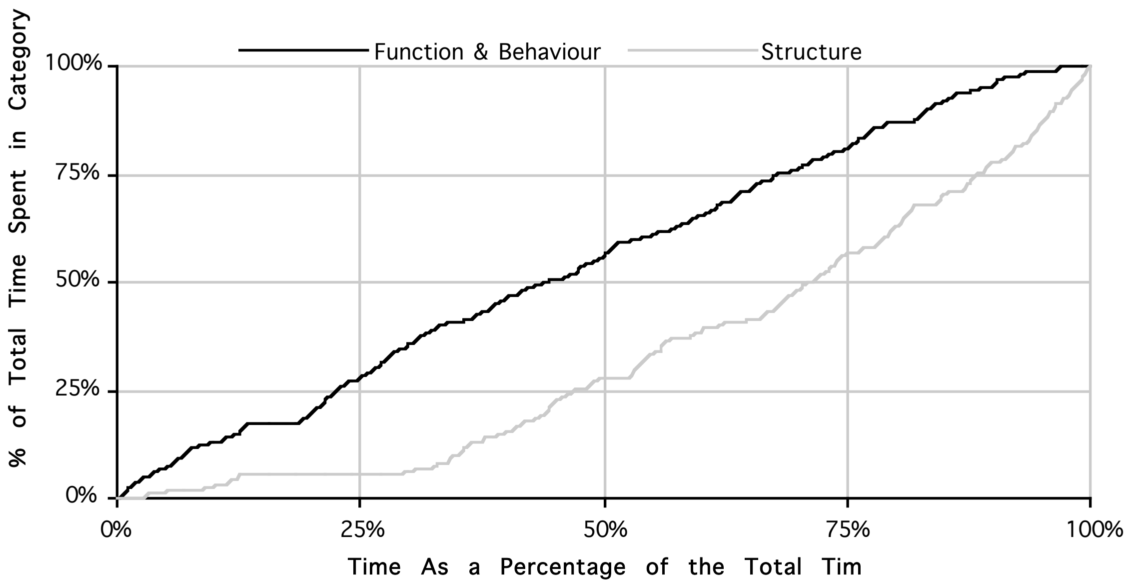
Fig. 9. % Total time spent reasoning with Function and Behaviour
vs % total time spent reasoning with Structure
(a) first design episode, (b) second design episode.

In the second and third design episodes the designers begin in the opposite fashion with the Structure curves preceding the Function and Behaviour curves. In the third design episode the curves cross at 60 percent of the elapsed time. In both of these episodes the curves converge in the last 25 percent of the time of the design episodes.

It would appear that this method of viewing the protocol results removes much more of the detail of the design episode than the moving weighted average method. It is not possible to see in the graphs any of the details described in the section above. It is possible however to characterise the design sessions in general using this process.



(c) Third design episode



(d) Delft design episode

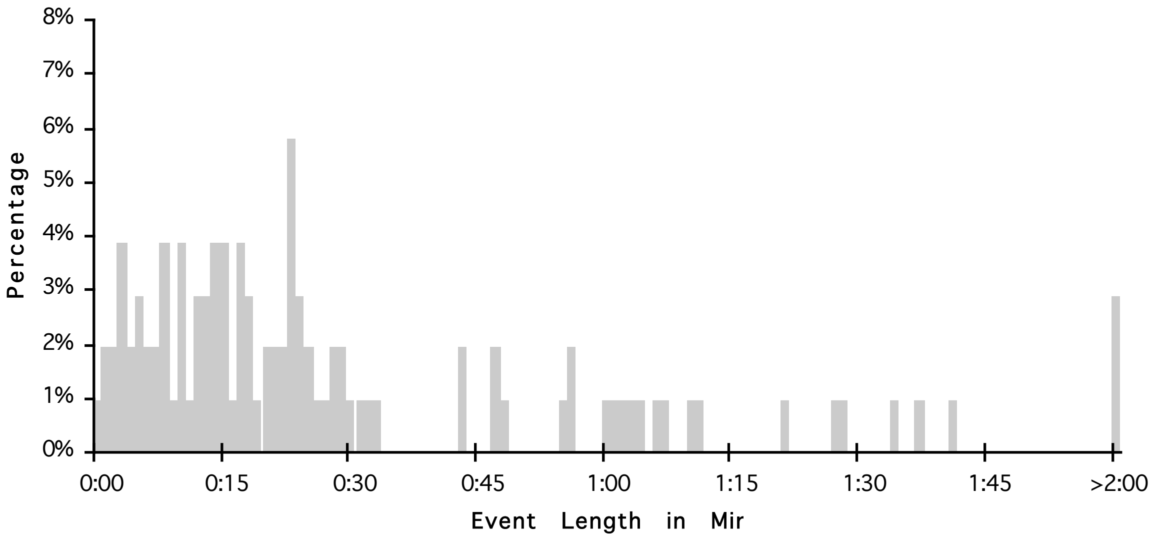
Fig. 9. % Total time spent reasoning with Function and Behaviour vs % total time spent reasoning with Structure (c) third design episode, (b) Delft design episode.

5.3. Category Event Lengths

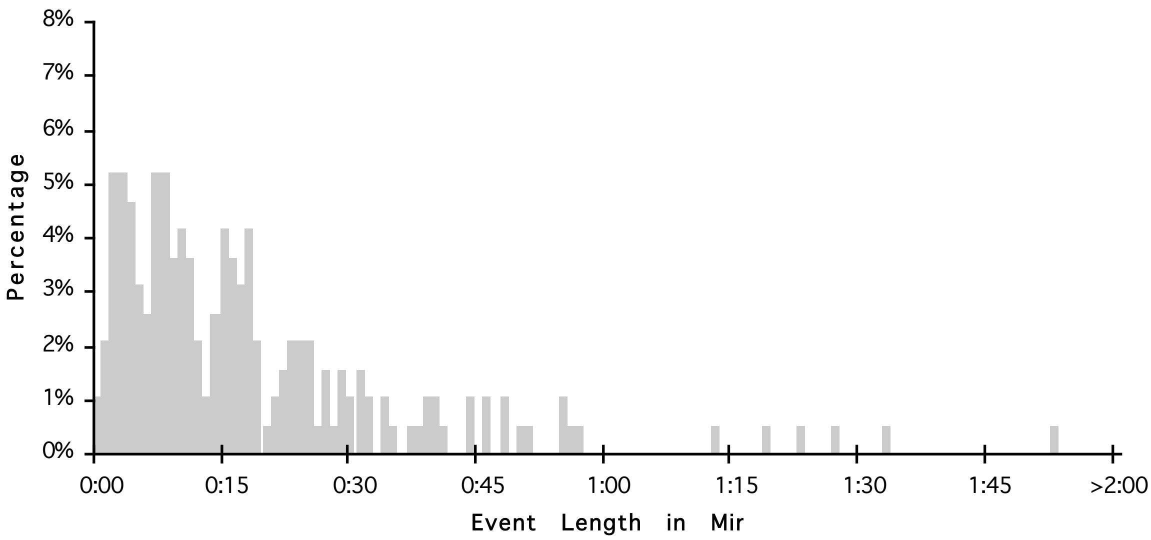
The time each designer spends on different categories is also of interest. This gives a measure of how quickly the designer engages the design tasks and provides information on the rate at which designers change their micro strategic focus. This form of knowledge about the behaviour of designers has not been readily discernible in previous protocol studies. Two ways of looking at event lengths are used.

5.3.1. Spectrum of Category Event Lengths

A spectrum of event lengths is plotted in the range of 0 to 2 minutes. The quantum of time measurement in the protocol coding is one second. For each second the number of events of that length of time is recorded as a percentage of the total number of events in the design episode. By plotting percentages the result is independent of the number of events in the episode.



(a) First design episode

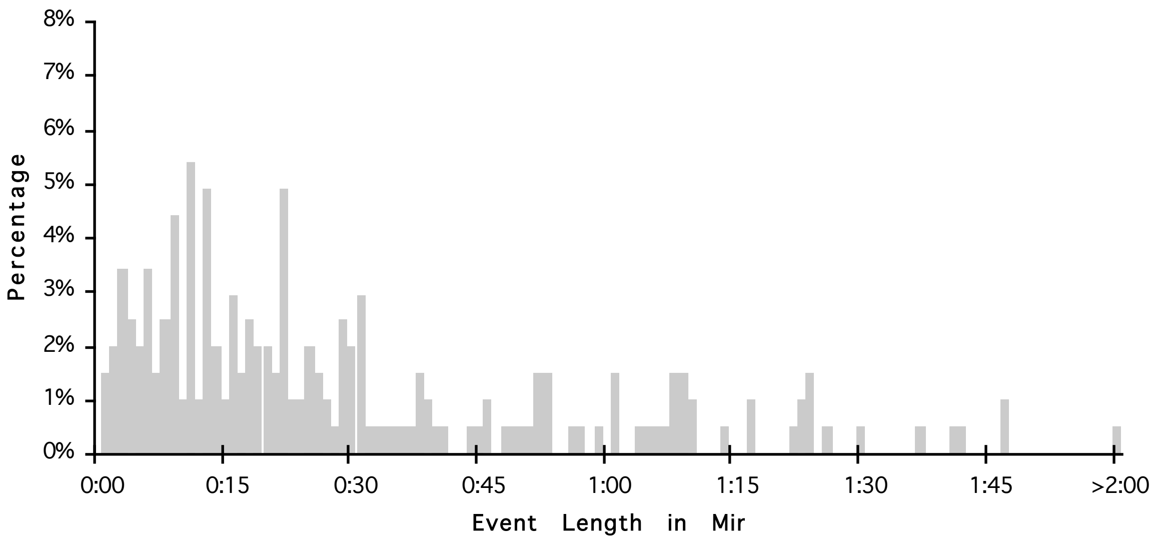


(b) Second design episode

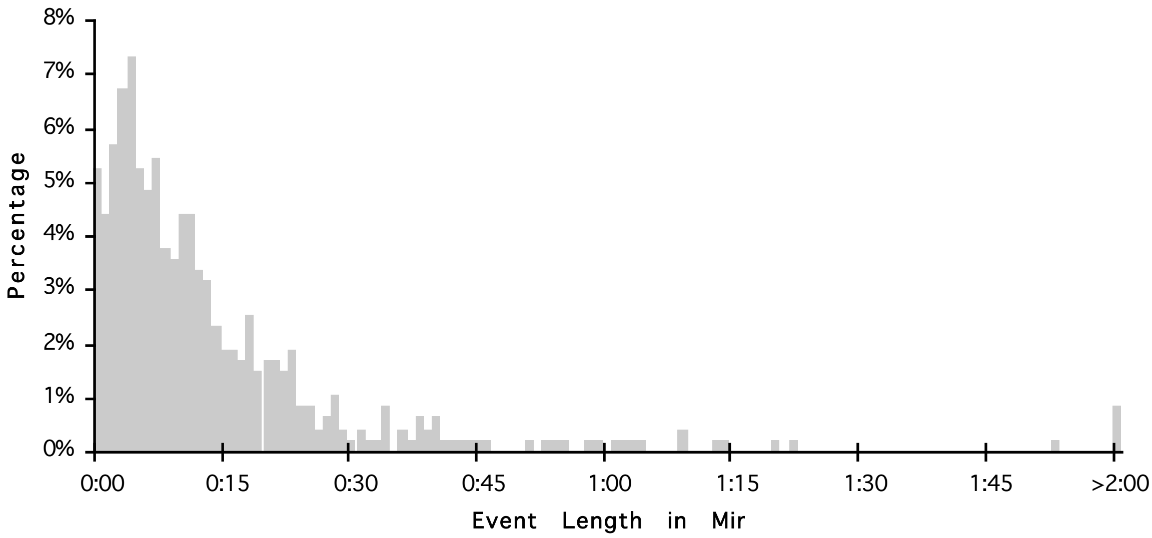
Fig. 10. Spectrum of coding event lengths (a) first design episode, (b) second design episode.

The spectra of the experienced designers (second and Delft), Figures 10(b) and (d), differ from the inexperienced designer, Figures 10(a) and (c). For the first and third design episodes most event lengths are less than 30 seconds and generally more distributed with rarely more than 4 percent of events in any one category. In the second and Delft design episodes the majority of event lengths are less than 20 seconds with the vast majority of events falling in the categories less than 15 seconds.

These differences may reflect differences in expertise with the experts moving more quickly through the design task or they may reflect differences in verbalisation between the designers. A greater number of designers would need to be examined before the reasons for the differences could be determined. What is important is that significant differences can be seen between different groups of designers.



(c) Third design episode



(d) Delft design episode

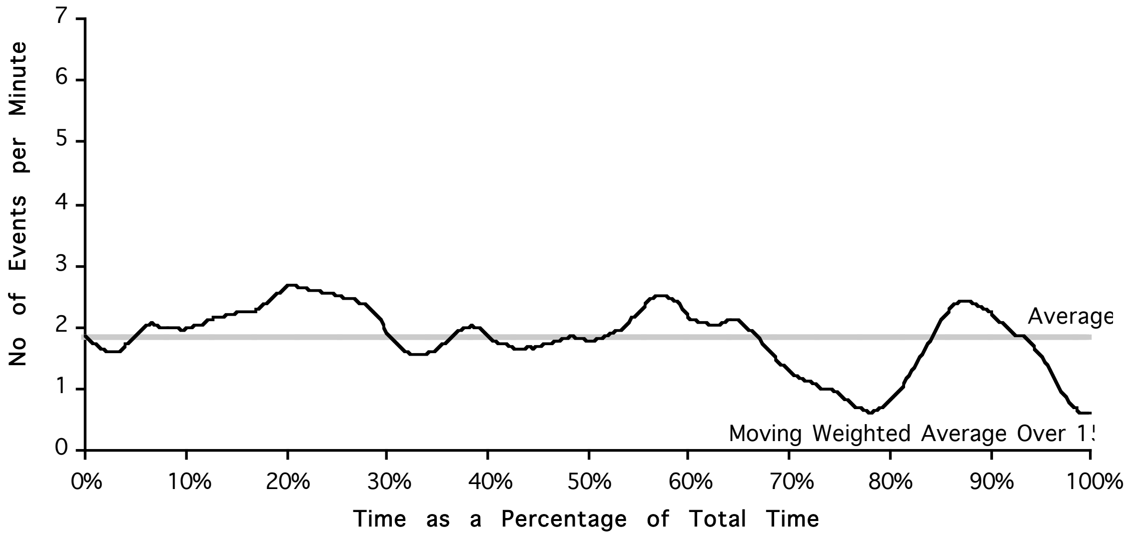
Fig. 10. Spectrum of coding event lengths (c) third design episode, (d) Delft design episode.

5.3.2. Category Event Lengths as a Moving Weighted Average

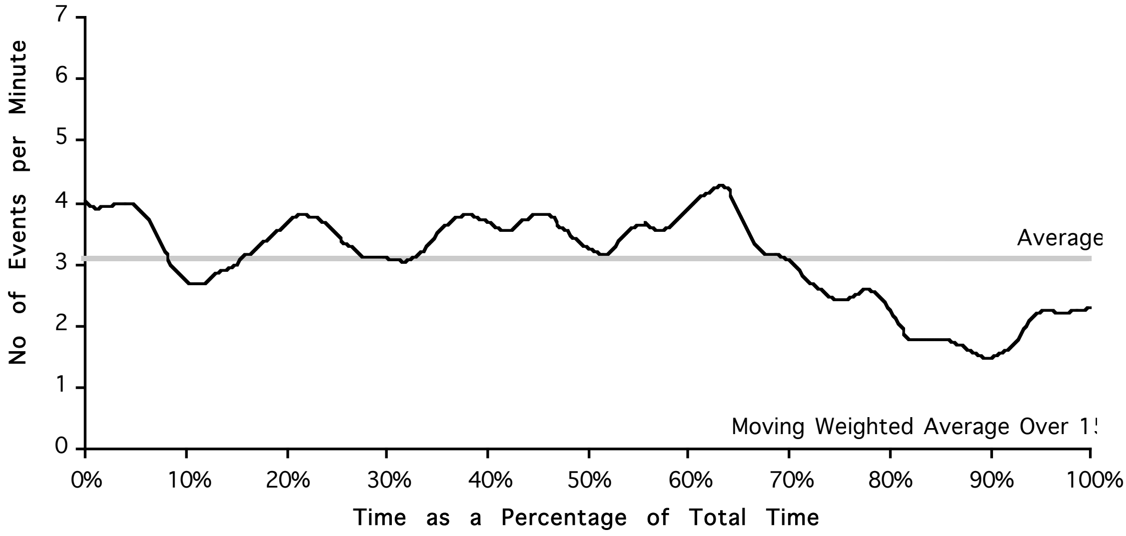
The above spectral distributions show a summary for each design episode. By plotting the number of events per minute it is possible to view designer activity over the duration of the design episode. An moving average is again employed to give a smoothened representation of the graph, Figure 11. The filter used was the same shape as shown in Figure 7.

The first and third design episodes show an average number of categories of just under 2 events per minute. The two experienced designers, designer 2 and the Delft designer record on average 50 percent higher numbers of events per minute.

With the exception of the second designer each graph begins at a level below average rising at the beginning and then falling off towards the end of the design episode. The second designer is also tailing off until the 90% stage where his activity increases. The graphs all have the same general form with the designer activity varying noticeably throughout the design episode. Each designer shows a variation of a factor of around 4 between minimum and maximum rates of activity. The Delft designer shows the greatest variation from 1.25 to 6.1 events per minute.



(a) First design episode

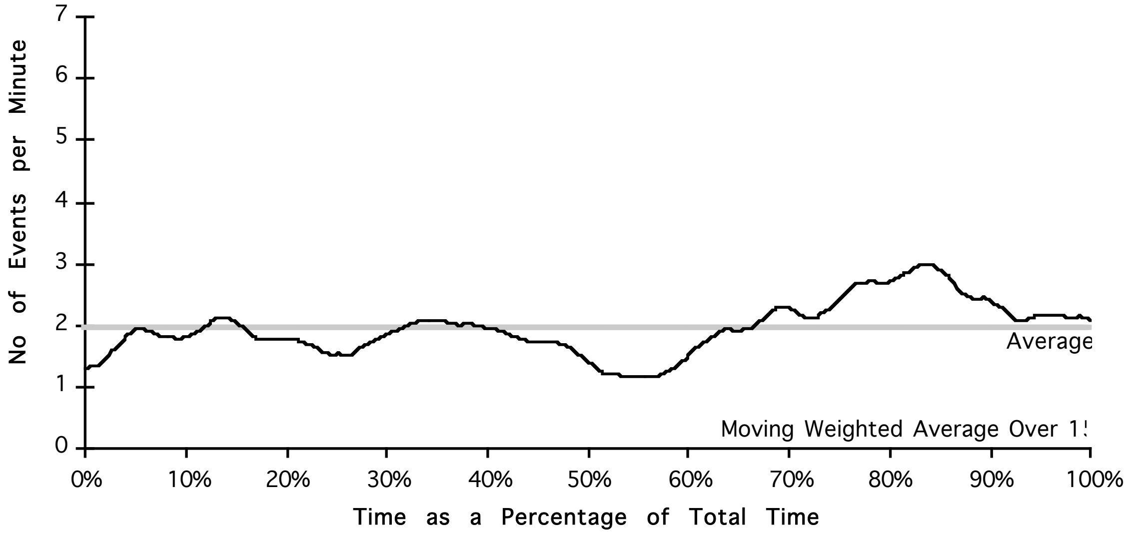


(b) Second design episode

Fig. 11. Coding event lengths as a moving weighted average (a) first design episode, (b) second design episode.

6. DISCUSSION

Design research has, over the last thirty years, largely focussed on the development of computer-based models of design processes. These models have fallen into two categories. Those which attempt to model some human designing process as understood through either introspection or abstract hypothesising. In some cases these models do not claim to model a human designing process, rather just the abstraction of one; 'designing by analogy' is one such example. Then there are those which do not attempt to base their processes on any human activity; 'designing by genetic evolution' is an example of this approach. However, there has been remarkably little research on capturing, presenting and analysing the activity of designing as carried out by human designers as a set of phenomena which are to be modelled and for which an explanatory theory is to be developed.



(c) Third design episode

(d) Delft design episode

Fig. 11. Coding event lengths as a moving weighted average (c) third design episode, (d) Delft design episode.

This paper aims to develop methods which can be used to begin that process. It uses the think aloud or protocol technique as a means of capturing and representing designing as carried out by human designers as a time sequence of activities. The think aloud or protocol technique is extended through the use of a domain-dependent coding scheme based on generic models of designing and a more robust coding methodology. This produces a much richer coding structure. As a consequence more information becomes available.

The analysis methods developed and applied here provide the basis for articulating different aspects of the behaviour of individual designers and for distinguishing the designing behaviours of different designers. As could be seen different designers exhibit some similarities such as those exemplified in Figure 6 and at the same time quite remarkable differences such as those exemplified in Figure 10. The differences in these cases appear to be related to the level of experience of the designers involved.

The development of such a tool as the one described in this paper offers opportunities for 'measuring' designing. It now becomes possible to test different hypotheses about how designers design. Some of the questions that may be able to be answered by the application of this tool include the following.

Are there differences between the designing activities of experienced and inexperienced designers?

What is the difference between student designers before and after they take a design course?

Are there fundamental differences between designers from different disciplines?

Are there differences when designing with and without computer aids?

Are there differences when designing with and without the use of sketches?

Tools such as this one are still in their infancy and it is likely that further tool development will be required. Certainly, further data analysis will need to be included to allow a more detailed study of the similarities and differences exhibited by designers when they are designing.

The application of this approach to the analysis of design protocols should provide a basis for a better understanding of designing as well as the basis for possible future computer-based design aids. Future work includes the collection and protocol analysis of a large number of design sessions using both longitudinal and lateral studies within and across design disciplines in order to begin to answer some of these and other questions.

Acknowledgments

This work is supported by a grant from the Australian Research Council.

REFERENCES

[1] Gero, J. S. and Maher, M. L. (eds) (1993). *Modeling Creativity and Knowledge-Based Creative Design,* Lawrence Erlbaum, Hillsdale, New Jersey.

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

[3] van Someren, M.W., Barnard, Y.F. and Sandberh, J.A.C. (1994). *The Think Aloud Method; A Practical Guide to Modelling Cognitive Processes,* Academic Press, London.

[4] Lee, T.Y. and Radcliffe, D.F. (1990). Innate design abilities of first year engineering and industrial design students, *Design Studies*, **11** (2):96-106.

[5] Guindin, R. (1990). Knowledge exploited by experts during software system design, *International Journal of Man-Machine Studies* **33**: 279-304.

[6] Visser, W. (1989). More or less following a plan during design: Opportunistic deviations in specification, *International Journal of Man-Machine Studies*, **33**: 247-278.

[7] Visser, W. (1992). Designers' activities examined at three levels: Organization, strategies and problem-solving processes, *Knowledge Based Systems* **5** (1): 92-104.

[8] Davies, S.P. (1991). Characterizing the program design activity: Neither strictly top-down nor globally opportunistic, *Behaviour & Information Technology* **10**(3): 173-190.

[9] Lloyd, P. and Scott, P. (1994). Discovering the design problem, *Design Studies*, **15** (2): 125-140.

[10] Cross N. (1984). *Developments in Design Methodology,* John Wiley, London.

[11] Cross, N., Dorst, K. and Roozenburg, N. (1992). *Research in Design Thinking,* Delft University Press, Delft.

[12] Dorst, K., Christiaans, H., and Cross, N. (eds) (1994). *Analysing Design Activity* (Pre-Prints for the Workshop on Analysing Design Activity), Delft University of Technology, Delft.

[13] Stauffer, L.A. (1987). An Empirical Study on the Process of Mechanical Design, *Doctoral Dissertation ,* Oregon State University, Corvallis.

[14] Stauffer, L.A. and Ullman, D.G. (1991). Fundamental Processes of Mechanical Designers Based on Empirical Data, *Journal of Engineering Design,* **2** (2): 113-125.

[15] Simon, H.A. (1981). *The Sciences of the Artificial*, MIT Press, Cambridge, Massachusetts.

[16] Simon H.A. (1983). Search and reasoning in problem solving, Artificial *Intelligence,* **21**: 7-29.

[17] Ullman, D.G., Stauffer, L.A. and Dietterich, T.G. (1987). Toward Expert CAD, *Computers in Mechanical Engineering,* **6** (3): 56-70.

[18] Gero, J.S. (1990). Design prototypes: A knowledge representation schema for design, *AI Magazine* **11**(4): 26-36.

[19] Gero, J. S., Tham, K. W. and Lee, H. S. (1992). Behaviour: a link between function and structure in design, *in* D. C. Brown, H. Yoshikawa and M. Waldron (eds), *Intelligent Computer-Aided Design,* North-Holland, Amsterdam, pp. 193-225.

[20] Gero, J.S. and Rosenman, M.A. (1990). A conceptual framework for knowledge-based design research at Sydney University's Design Computing Unit, *Artificial Intelligence in Engineering* **15** (2): 65-77.

[21] McNeill, T. and Edmonds, E.A. (1994). An empirical study of conceptual electronic design, *Revue Sciences et Techniques de la Conception*, **3** (1):61-86.

[22] Pahl, G. and Beitz, W. (1984). *Engineering Design,* The Design Council, London.

[23] French, M.J. (1985). *Conceptual Design for Engineers,* The Design Council, London.

[24] Coyne, R.D., Rosenman, M.A., Radford, A.D., Balachandran, M. and Gero, J.S. (1990). *Knowledge-Based Design Systems,* Addison-Wesley, Reading, Massachusetts.

[25] Linstone, H.A. and Turoff, M. (1975). *The Delphi Method; Techniques and Applications,* Addison-Wesley, Reading Massachusetts.

[26] Purcell, A.T. , Gero, J.S., Edwards, H.M. and Mc Neill, T. (1994). The Data in Design Protocols: The Issue of Data Coding, Data Analysis in the Development of Models of the Design Process, Preprint for *Workshop on Analysing Design Activity*. Delft, Netherlands, September.

[27] Archer, B.(1969). The structure of the design process, *in* G. Broadbent and A. Ward (eds), *Design Method in Architecture,* Lund Humphries, London, pp. 76-102.

[28] Asimov, W. (1962). *Introduction to Design,* Prentice-Hall, Englewood Cliffs, New Jersey.