

Modelling Unwarranted Commitment in Information Artefacts

Chris Roast

School of Computing and Management Sciences, Sheffield Hallam University, Sheffield, UK.

Abstract: This paper extends an analytic framework based upon that of Cognitive Dimensions which provides formal definitions for assessing the suitability of interactive systems for particular tasks. Elsewhere we have demonstrated that interface design can benefit from interpreting cognitive dimensions as formal tools for assessing interface characteristics relevant to effective use. Our interpretation of these dimensions has the benefit of introducing a level of precision to the otherwise informal notion of cognitive dimension. In general developing a more precise interpretation of the dimensions is a necessary prerequisite for their employment within software engineering.

In this paper an interpretation of the cognitive dimension termed ‘premature commitment’ is examined and its relation the dimensions of ‘viscosity’ considered. We demonstrate the appropriateness of the measures developed as a means of assessing implicit bias in interface behaviour and the general results that their formalisation enables. The effectiveness of the proposed formal characterisations is illustrated with a small case study.

Keywords: Cognitive Dimensions, Evaluation, Formal Modelling, Premature Commitment

1. INTRODUCTION

Numerous conceptual tools have been proposed to alleviate the complexities of effective interface design. This paper develops one such conceptual tool, termed cognitive dimensions. Cognitive dimensions have been proposed by Green (Green, 1989) as providing broadbrush widely applicable usability metrics. These ideas have received considerable interest and are widely recognised as a promising aids to design and evaluation activities within the broad domain of information artefacts (Gilmore, 1997; Lavery et al., 1996; Yang et al., 1995; Green and Petre, 1996; Blandford and Green, 1997). The research described here is concerned with refining concepts drawn

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35349-4_22](https://doi.org/10.1007/978-0-387-35349-4_22)

S. Chatty et al. (eds.), *Engineering for Human-Computer Interaction*

© IFIP International Federation for Information Processing 1999

from cognitive dimensions to provide an analytic basis for understanding and applying them.

Employing formal models has a number of potential benefits: it provides a precise representation that can focus discussion and negotiation; tools and techniques can be employed to analyse a formal design prior to implementation and empirical evaluation; and design tools and techniques may directly feed off such representations. In this paper we are concerned with the first two of these benefits, with the aim of bridging the gap between usability, as a poorly expressed requirement, and precise design representations, as necessitated by the eventual system. In some cases this has involved developing enhanced system models that embody usability requirements (Dix, 1991; Roast and Siddiqi, 1997b; Roast, 1998), while in others explicit models of the user (or task) have been formally combined with system representations (Barnard and Harrison, 1992; Duke et al., 1995; Harrison et al., 1993; Palanque and Bastide, 1997).

An analytical framework incorporating formal interpretations of some of the cognitive dimensions has been developed and applied to systems ranging from individual dialogue boxes to programming environments (Roast and Siddiqi, 1996; Siddiqi and Roast, 1997).

The paper first introduces the particular dimensions examined and illustrates the approach to be taken by briefly reviewing viscosity. Formal interpretations of the dimension of 'premature commitment' are introduced and examined. Finally, the validity of the formalisation is considered using a small case study.

2. INTERPRETING COGNITIVE DIMENSIONS

Cognitive dimensions (Green, 1989) have been developed as psychologically motivated metrics for a wide range of devices that focus upon factors central to successful use. The dimensions suggest a valuable and informal evaluative space in which artefacts can be located and compared (Green and Petre, 1996). In particular the dimensions are intended to reflect general concepts which are familiar and yet not well understood, hence they support the articulation of a range of familiar interactive behaviours and interface characteristics in terms that are sufficiently general to enable diverse design problems, and their solutions, to be easily related. The dimensions ideally provide a framework available to both designers and users which supports both interface evaluation and design. The informal nature of cognitive dimensions is in some cases assumed to benefit to their broad application and adoption as an evaluative aid. However their largely anecdotal nature is highly limiting when considering their uniform and consistent application, as demanded by an engineering context (Gilmore, 1997; Lavery et al., 1996; Yang et al., 1995). In addition, very few studies of the dimensions have focused upon understanding the characteristics of the underlying framework which they presuppose.

In this paper we concern ourselves primarily with the dimension termed premature commitment and identify its potential associations with dimensions of viscosity. For

both premature commitment and viscosity the primary concern is how hard it is for users to reach or transform easily particular goal states.

Premature Commitment concerns the complexities associated with of attaining a specific goal state and the possible inappropriateness of the states encountered and the actions necessary to satisfy the primary goal. In particular, it involves the system engaging the user in making a decision which is not immediately relevant to achieving their primary goal. A simple example is that of a desktop publishing package which when starting a new document asks the user how many pages their document is to consist of. Clearly, for some uses the number of pages in a document may be a primary concern in which case the interface behaviour may not be problematic. However for many document preparation tasks, such as report writing, the number of pages is more commonly a secondary outcome of completing the document (the primary goal). If we take this example to be illustrative of premature commitment, we can observe that in attempting to achieve one primary goal: the user has to engage in achieving a secondary sub-goal; and, the secondary sub-goal is not essential for achieving the primary goal.

By contrast *viscosity* concerns the complexity of achieving changes in the medium offered by an interface. Two types of resistance to change have been examined: *repetitive viscosity* refers to modifications involving intensive, and often repetitive, user effort; *knock-on viscosity* refers to the need for corrective work following a conceptually simple modification.

2.1 Formalisation

The formal interpretation of cognitive dimensions serves a number of purposes. First, it provides a common framework in which the informal notions can be represented and related. In general this benefits interface engineering, supporting the general understanding of how factors influencing effective and appropriate interaction are inter-related. Second, the formal interpretation provides a framework in which the concepts underlying illustrative examples can be teased-out and refined, and their overall validity assessed.

The approach taken to modelling the dimensions in this research is that of employing a design representation as a means of expressing system characteristics and requirements in system objective terms (Jackson, 1997). This approach provides a focused perspective upon the artefact, treating factors such as the context of use, user tasks and psychological make-up as reflected within the artefact. System orientated modelling reflects the complexity of interface design in which system behaviours can only be defined in terms of the interaction history. The same approach has been employed elsewhere to investigate the implicit assumptions underlying interface design and the impact of user centred requirements upon system specification and architecture (Dix, 1991; Dearden and Harrison, 1997; Markopoulos et al., 1997; Roast, 1997).

To characterise cognitive dimensions it has been convenient to provide interpretations based upon: *the goals* (and sub-goals) which users may achieve, and *the user inputs* which enable users to satisfy different goals. This system framework is clearly

not cognitive in nature and hence our interpretations are best viewed as comprises between the cognitive view and the practical demands of the system model.

2.2 Examining Viscosity

A previous investigation of the cognitive dimensions of viscosity has been conducted and used to illustrate inter-dependencies between *repetitive viscosity* and *knock-on viscosity* (Siddiqi and Roast, 1997).

Repetitive viscosity concerns the complexity of actions required to make modifications. We model these in terms of the user inputs which can implement a modification which is characterised by a pre-condition (*pre*) and a post-condition (*post*). The expression *pre* REP *post* is taken to denote the language of all possible input sequences which can satisfy *post* starting from a state in which *pre* is satisfied. The repetitive viscosity of changes are assessed in terms of the complexity of *pre* REP *post*. At present we have taken the length of the minimal elements within *pre* REP *post* as a simple and immediate measure of complexity.

This model can be illustrated using a simple example of editing an enumerated list (such as a reference list). We will assume use goals include: *enum* meaning the list is enumerated, and; *list(...)* indicating the list contents. The repetitive viscosity of adding an item (*N*) to the list is based upon the language:

$$list(I_1, \dots, I_n) \text{ REP } list(I_1, \dots, I_i, N, I_{i+1}, \dots, I_n)$$

This language would include inputs which locate the point of insertion and enter the new item. The added complexity of modifying an enumerated list (without special tool support) can be illustrated when we consider adding an item *and* keeping the list enumerated:

$$(enum \wedge list(I_1, \dots, I_n)) \text{ REP } (enum \wedge list(I_1, \dots, I_i, N, I_{i+1}, \dots, I_n))$$

This language will not only include the insertion of the new item, but also the necessary commands to keep the list enumerated.

By contrast, *knock-on viscosity* is characterised in terms of how a system makes the satisfaction of some goals unnecessarily interfere with others. One characterisation of this interference is where an input satisfies one user goal (*p*), and yet a second goal (*q*) is not satisfied — this relation is written *p* DIS *q*. In our examination of knock-on viscosity, the number of instances of DIS that can influence a modification is taken to be indicative of the degree of knock-on viscosity. For example, considering the modification of an enumerated list we have that: *list(I₁, ...I_n)* DIS *enum* when inserting an item. That is, achieving a specific list content by inserting an item results in the list no longer being enumerated, thus the user may subsequently have to re-satisfy *enum*.

The distinctive nature of the two dimensions modelled can be illustrated by the considering the effect of modifying the list edit example to include an ‘auto-enumerate’ command. In this case the repetitive viscosity can be reduced since the work to enumerate a list is now far simpler using the ‘auto-enumerate’ command. However, de-

spite the availability of the new command, the instance of DIS still holds, thus indicating the persistence of a more general level of resistance to change.

2.2.1 Composition

Assuming that these interpretations are reasonable characterisations, their formal definitions can be used to investigate general properties of the dimensions. Such general properties can be used to provide a predictive assessment of general classes of interfaces and identify legitimacy of particular design proposals (Roast, 1997). This can be illustrated by considering interface designs expressed in terms of the composition of interface elements.

Briefly, given two interfaces (A and B) which both enable access to the same functionality. If a design is proposed which combines the two interfaces ($A \oplus B$) enabling users to freely switch between A and B , then the viscosity of the combined interface can be inferred using the definitions. The repetitive viscosity of $A \oplus B$ will be the minimum of that of A and B ; and, the knock-on viscosity of $A \oplus B$ will be summative, that is the knock-on viscosity of A and B combined. Hence, under this form of composition *repetitive viscosity* will be no worse than that of the individual elements composed, where as *knock-on viscosity* will be no better.

These observations illustrate one way in which a cognitive dimensional space can be investigated by virtue of relying on accurate definitions. In particular, such observations can be related back to the informal concepts as a means of clarifying their effective application.

3. PREMATURE COMMITMENT

This section develops a proposed formal characterisation of the system behaviours we would consider to be indicative of premature commitment. The notation used to represent the dimension formally is an action logic ACTL (Nicola et al., 1991). Temporal action logics have been utilised elsewhere in formally modelling interaction (Paterno, 1997). In this work atemporal propositions correspond to system state properties and the influence of user inputs is represented by temporal constructs.

Briefly, for any property p :

- $[op]p$ is true iff following the input op , property p is true;
- $AG p$ is true iff following any sequence of inputs p is true;
- $EF p$ is true iff there is some sequence of inputs that may lead to p being true;
- $A p U q$ is true iff in all futures p is true until q is true.

When using ACTL we assume that all inputs, or operations, are user driven and that all user goals correspond to state properties.

In the introduction we employed the notion of a primary goal and a secondary (unforeseen) goal, and described premature commitment as the user having to satisfy the secondary goal prior to achieving the primary goal. In terms of the initial example, the primary goal would be to complete the document and the secondary goal would be to

declare the size of the document in pages. To develop a formal estimation of premature commitment we focus upon three facets of the concept: (i) the characterisation of how the secondary goal is unavoidable; (ii) the relationship between the secondary and primary goals which indicates the inappropriateness of having to commit to the first, and; (iii) the initial conditions implicit in cases of premature commitment.

Imposed ordering For a secondary goal to be premature, it is not possible for the primary goal to be satisfied without the user having to explicitly encounter the secondary goal.

A special notation will be used to represent user actions which explicitly concern possible goal properties. The user action assert_g will be used to represent an input by the user which confirms or inputs information asserting the goal property g .

In order to represent unavoidably encountering the secondary goal, we have all possible futures satisfying the primary goal (p) only after asserting the secondary goal (q):

$$A (\neg p) \text{ U } [\text{assert}_q]q$$

Always not p until q has been asserted

It can be noted that this is a weak interpretation of the imposed ordering that has been attributed to some examples of premature commitment. In particular the above property does not demand that the secondary goal is satisfied strictly before the primary, since it is possible that the input assert_q can satisfy it. Thus, a similar investigation of premature commitment could focus upon the stronger characterisation, such as:

$$A (\neg p) \text{ U } [\text{assert}_q](q \wedge \neg p)$$

Contingency In addition to the imposed ordering the description of premature commitment suggests that the goals involved are related within the domain. Here, we consider the possible dependencies between the primary and secondary goals that systems can impose.

First, the secondary goal is often deemed unforeseen or inappropriate when focusing upon the primary goal. Hence, one view of the word processor example is that the size of the document in pages is independent of the goal of completing the document. Thus, in general the secondary goal is superfluous. One way of characterising this lack of dependency between the goals is to require that no logical alternative is intrinsically prevented:

$$AG (EF (p \wedge q) \wedge EF (\neg p \wedge q) \wedge EF (p \wedge \neg q) \wedge EF (\neg p \wedge \neg q))$$

No domain dependencies exist between p and q

Second, the account of premature commitment can be interpreted as focusing the users attention upon secondary details which would have to be answered at some point in order to achieve the primary goal. Hence, in the context of the example, this interpretation suggests that the number of pages is a facet of the users' primary goals, which they are required to declare earlier than they wish. In this case the relation between the primary and secondary goal, is that the secondary goal is entailed by the primary.

$$\text{AG } (p \Rightarrow q)$$

p entails q

Initial Conditions Examples of premature commitment describe behaviour starting from some situation initiating a period of interaction. Hence its formal characterisation should make explicit the initial condition for such a period. Formally, there is no simple way to characterise what is perceived as a period of use, but for it being the case that neither primary or secondary goals are met when the period starts. Hence, we propose a relatively weak initial condition: $\neg p \wedge \neg q$.

We combine these elements to obtain formal interpretation of *strong prematurity* and *weak prematurity*. Strong prematurity is distinguished by the primary goal not logically entailing the secondary goal.

Definition The goal q is said to be **strongly premature** to p (written $q \text{ PREM } p$), iff:

$$\begin{aligned} & \neg p \wedge \neg q \wedge \\ & \text{A } (\neg p) \text{ U } [\text{assert}_q]q \wedge \\ & \text{AG } (\text{EF } (p \wedge q) \wedge \text{EF } (\neg p \wedge q) \wedge \text{EF } (p \wedge \neg q) \wedge \text{EF } (\neg p \wedge \neg q)) \end{aligned}$$

The two goals are domain independent, however it is not possible to satisfy p without first asserting q .

Definition The goal q is said to be **weakly premature** to p (written $q \text{ prem } p$), iff:

$$\begin{aligned} & \neg p \wedge \neg q \wedge \\ & \text{A } (\neg p) \text{ U } [\text{assert}_q]q \wedge \\ & \text{AG } p \Rightarrow q \end{aligned}$$

In general, p entails q within the domain, however it is not possible to satisfy p without first asserting q .

3.1 The Interpretation Applied

In this section we characterise three reported examples of premature commitment.

3.1.1 Programming By Example

Modugno et al. (Modugno et al., 1994) describe a GUI front end to an operating system which provides a 'programming by example' facility — the user demonstrates an operation and the system derives a script for performing the operation. Two examples,

of high and low premature commitment are reported with this facility. First, if the user wishes to ensure a derived script is robust, they must engage in an activity of checking its behaviour for all possible classes of data object that it may encounter. Second, the derivation in itself makes assumptions about the generalisable characteristics of a demonstration, and the user does not have to speculate about specific details.

Both examples, depend upon distinguishing an operation which is properly defined to accommodate all contingencies (termed robust), from a derived operation which is adequate for many purposes but not necessarily all. The first is considered prematurely committing since the user has to check details which they may not have envisaged, and the second has low commitment since the user does not have to examine the generalisations made. For both cases we characterise an operation as a set of condition action pairs $(c_1, a_1), \dots, (c_n, a_n)$ which identify the function performed. In the case of a derived operation (*drvd*) there will be conditions under which the operation does not function correctly, since the operation is mechanically ‘guessed’. In the case of a robust operation (*robust*) which a user may wish to develop, all conditions will be accommodated.

The first case of premature commitment has the primary goal of *robust* and recognises that some conditions action pairs have to be provided by the user in order to achieve the goal. The condition action pairs asserted are premature in the sense that the conditions will include unforeseen system configurations, hence: $(c_i, a_i) \text{ prem } robust$. This is weak prematurity since *robust* will entail the asserted condition action pair (c_i, a_i) .

The second case of *low* premature commitment is a negative example — it is not the case that the user has to assert, say, (c_j, a_j) in order to achieve the goal of having the operation defined. Since the derived operation *drvd* is the primary goal, its lack of robustness is not a concern. Thus, assuming the derived operation *drvd* includes (c_j, a_j) , then: $\neg(c_i, a_i) \text{ prem } drvd$.

3.1.2 Visual Programming

In many visual programming environments programs are constructed out of icons representing data and operations and arcs (or wires) linking them. As a result the programmer ‘draws’ a program and often has to guess as to appropriateness of physical locations for the icons used without a full awareness of their best location when the program is complete. Hence, the user’s primary goal may be to declare a data source, without having to commit themselves to a physical location for the data source icon (Green and Petre, 1996).

Assuming that for a required data source, the source can be identified as well as its location, we have: *locationIDed* prem *dataSourceIDed*. The premature commitment is weak since within the domain of visual programming the identification of the data source cannot be achieved without providing a location.

3.1.3 Prototype Objects

In some event based programming environments, a prototype object sort can only be built from an object instance. Hence, ideally the user has to create an instance

object with the characteristics that will suit the prototype. However, the mechanism can prematurely commit the inexperienced programmer to prototype objects with the instance specific characteristics of the object from which it was built (Gilmore, 1997).

A prototype object with generic attributes cannot be created without creating an instance object with specific attributes. Since the prototype object does not necessitate specific attributes, we have:

specificAttributes PREM prototypeObject

4. EFFECTS OF PREMATURE COMMITMENT

Here we examine the interpretations developed and consider how their formal properties inform our understanding of the informal dimensions, and how an analysis of the interpretations can motivate their refinement and their scope of applicability.

4.1 Prematurity under composition

It is valuable to be able to assess the influence of module composition upon premature commitment, since it can help inform system development involving the extension and the combination of sub-systems. As with viscosity, the formal interpretation of premature commitment enables the effect of composition upon the dimension to be easily assessed. We consider combining two alternative interfaces (A and B) which access the same underlying functionality. The combined interface ($A \oplus B$) allows users to freely move between and use A and B independently. The interpretation of premature commitment under this type of composition shows that both types of prematurity can be reduced:

If p prem q for A , and not p prem q for B , then not p prem q for $A \oplus B$,

Hence, the degree of prematurity for $A \oplus B$ is less than (or equal to) that of A or B .

This concurs with our intuitions, consider a system A involving a case of premature commitment, making another system B equally available to the user has the potential to allow the user to still satisfy their goal while avoiding the prematurity embodied in A . Hence cases of premature commitment can be avoided by the provision of additional tool support. Thus for the visual programming example, a tool offering an alternative view, such as text, would enable a data source to be identified and used (*dataSourceIDed*) without having to commit to its specific visual location (*locationIDed*).

4.2 Relations with Viscosity

4.2.1 Knock-on Viscosity

Knock-on viscosity characterises a change which affects the overall integrity of the object being altered. As a result knock-on viscosity is not so much concerned with

resistance to change but with the impact that alterations can have upon a system, or object, as a whole. The notion has been interpreted formally with the concept of ‘disruption’ which relates two goals and an input, such that: if the input results in the first goal being satisfied, then the second goal is negated.

In principle, formal association between the characterisation of knock-on viscosity and that of premature commitment can be sought. It is possible to employ formal methods to determine in what cases disruption may entail premature commitment and vice-versa. Several potential dependencies between the two have been examined yielding highly specific conditions qualifying the dependency relation. On reflection, these qualifying conditions are sufficiently specific that we conclude there to be minimal inherent dependencies linking the two dimensions.

4.2.2 Repetitive Viscosity

Repetitive viscosity has been characterised by the language of user inputs capable of achieving a specified change, and the subsequent assessment of the complexity of that language. Relating this to premature commitment, it can be shown that instances of prematurity imply an inequality between two comparable languages. This suggests that premature commitment contributes to the viscosity of satisfying the primary goal.

Assuming an instance of premature commitment q prem p , the repetitive viscosity of achieving p from the initial conditions is $(\neg p \wedge \neg q)REPp$, and the repetitive viscosity of achieving q is $(\neg p \wedge \neg q)REPq$. The definition of q prem p entails the satisfaction of q prior to p , hence for the above two languages: *there is an input sequence achieving the secondary goal q which is a prefix of an input sequence achieving the primary goal p* . This links the languages, showing that the two are not independent, the complexity of one can influence the other. Taking the minimal language as a measure of repetitive viscosity, the above observation shows that achieving the secondary goal may be less viscous than achieving the primary.

The linkage of the concepts of premature commitment and repetitive viscosity is weakened by the fact that premature commitment is specific to the initial conditions of an interactive use, where as repetitive viscosity is not specific to any temporal context.

The weak dependency between premature commitment and both types of viscosity can be of benefit — it informs us that the dimensions may be addressed independently. Such an observation supports the feasibility of ‘design guidelines’ which propose that the impact one dimension can be compensated for by another. For example, Gilmore (Gilmore, 1997) proposes that impact of high premature commitment can be combated with low viscosity in order to allow premature decisions to be easily revised.

5. CASE STUDY

Having introduced a formal interpretation for the notion of premature commitment we illustrate its application in more detail by examining an interactive tool. The particular tool to be examined is called “WiZe”, it is a WYSIWYG-style editor for writing specifications in the formal language ‘Z’. We use an example of WiZe’s behaviour to

illustrate the manner in which our account of premature commitment relates to properties of the tool which would be informally judged as imposing unwarranted ordering and load upon users.

WiZe has been developed as part of an on going research project which is now nearing completion (Morrey et al., 1996). The tool is used by both staff and students in the preparation of papers and teaching related material.

WiZe is designed to help its users author and prepare system specifications in the *Z* notation. The *Z* notation has been developed to enable the specification of systems in terms of discrete mathematics (Spivey, 1988). One distinctive feature of *Z* is that it provides a structuring mechanism for specifications which enables a collection of related elements, termed schemas, to be introduced and then used in various combinations to define more complex elements and systems.

WiZe provides a basic word processor interface with the addition of menus and dialogues providing access to the mathematical symbols of *Z* and schema constructs. Schemas are created using a dialogue in which the user identifies the type of the new schema and a new schema name. (Note that for purposes of this examination we only consider labelled schemas within the *Z* notation.) The new schema is then created without any content and the user is free to enter details when they wish to. WiZe provides a number of schema specific operations, such as schema-cut and schema-copy, and automatically maintains the uniqueness of each schema name.

In order to characterise premature commitment within WiZe, we consider the following possible user goals orientated to schema creation:

- *sc* = a schema exists with a specific content
- *sn* = a schema exists with a specific name

Different styles of use of WiZe can be characterised within the context of these goals by considering alternative combinations of them as primary user goals. For example:

- The primary goal of just *sc* corresponds to use focused upon the expression of specific schema content without consideration for the type of schema required or its name.
- An alternative type of use which maybe viewed as more strategic and promoting the structure and readability of a specification is for the primary goal to be *sn*. In this case the user is focusing upon naming specification elements, and not detailing their content or formal meaning.
- The final example is where the primary goal is $sc \wedge sn$. Here, both aspects of the specification being prepared are of equal importance. One can think of this case as being close to the use of WiZe purely as a type-setting tool where a previously prepared specification is being transcribed using the tool.

When creating a schema WiZe's dialogue requires the name of the schema before any schema construct appears in the document and before any content can be provided. This behaviour can be characterised as prematurely committing the user to the schema name.

The initial condition is a WiZe document in which a required schema is not given in terms of its content, its type, or its label: $\neg sc \wedge \neg sn$.

Satisfying the goal of having a new schema with the required content is only possible following the dialogue in which the user has to assert the schema name. This behaviour conforms with: $A(\neg sc) \cup [\text{assert}_{sn}]sn$.

Finally, WiZe maintains a degree of specification integrity which prevents the author from creating some specifications which are ambiguous in various respects. Within WiZe although a schema can be created without content, a schema without a name cannot be created. Thus we have: $AG(sc \Rightarrow sn)$.

Combining these formal properties together yields a weak premature commitment to a schema name: $sn \text{ prem } sc$.

The biases that instances of premature commitment reflect have been described in terms of primary and secondary goals. More generally such biases reflect often implicit assumptions about the way in which a tool is intended to be used, and thus they can be seen as claims about what are users' primary goals. For example, the examination of WiZe suggests premature commitment to sn , however this can be viewed as claim about WiZe users — *Claim: WiZe users primary goal within the context of schema creation is the schema name.*

In fact since users frequently adapt to the nuances of the tools they employ, we can look towards user experiences of WiZe for evidence of adopting sn as a primary goal, in preference to sc . A survey examined this hypothesis, novice and experienced WiZe users were encouraged to reflect on the best ways in which to use the tool by asking them to select instructional statements suitable for new WiZe users. The outcome of the survey showed that more experienced WiZe users favoured instructions which encouraged the choice of schema name prior to its creation. Out of 23 users a total of nine expressed an opinion on the choice of schema name prior to creating a schema. Of the nine, four of the more experienced users, and one inexperienced user, recommended the choice of schema names, and the other four inexperienced users rejected the same advice. In summary:

experience	recommended	no opinion	rejected
≥ 6 hours	4 (17%)	19 (83%)	0 (0%)
< 6 hours	1 (4%)	18 (79%)	4 (17%)

Although this survey did not form part of a planned experiment, the outcome does provide some supportive evidence for sn being adopted as a primary goal. In addition, this suggests that the behaviour classed as premature commitment is relevant in assessing how tools such as WiZe get used.

6. SUMMARY AND CONCLUSION

This paper contributes to an investigation into the formal characterisation of psychologically motivated measures of notations and tools. This work is novel in adapting

a primarily cognitive perspective to a system based framework. Positioning the dimensions within a well defined framework has enabled their precise exposition, this not only enables trade-offs and comparisons between dimensions to be investigated but it also encourages the re-examination, and informal exposition, of the dimensions currently used. In addition we have also addressed a major weakness of cognitive dimensions by employing an interpretative framework relating directly to the characteristics of interactive systems. This not only allows the dimensions to be provided with a system-based meaning, but it serves as a basis for employing them constructively in development.

In summary, generalising from the formal examination:

- at least two formally distinct classes of premature commitment can be distinguished
- cases of premature commitment can be reduced by the introduction of complementary sub-systems.
- cases of both knock-on viscosity and repetitive viscosity are not strongly related to instances of premature commitment.

There have been a variety of approaches to capturing interface usability issues within formal frameworks. The formal frameworks developed have been motivated by concerns, ranging from cognitive modelling and requirements engineering through to enabling rigorous software development and appropriate software architectures. In many cases these approaches have been driven by a rich set of examples, or scenarios, and have demonstrated a 'craft' approach to modelling which is largely conducted and employed by experts but not widely exploited.

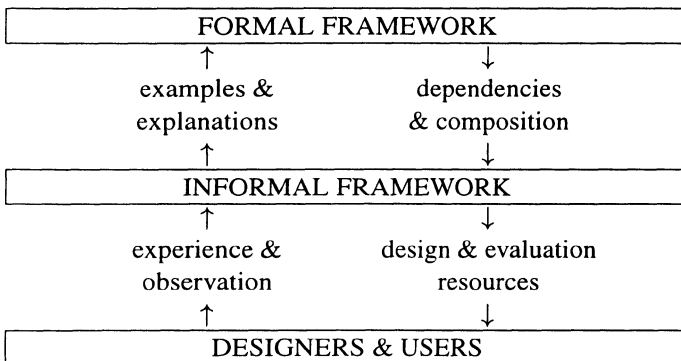


Figure 1: A role for generic formal modelling in HCI

By contrast, the modelling conducted in this paper is orientated towards developing the underlying framework upon which the informal and compelling notion of cognitive dimensions may depend. Hence, the specifics of the formal framework inform and elaborate the foundations for conceptual tools which have been developed for widespread adoption and exploitation (figure 1). This approach places modelling in context where its utility lies in contributing to the understanding, and subsequent effective use of, what would otherwise be informal tools for design and evaluation.

ACKNOWLEDGEMENTS

The author is indebted to staff within the Requirements Engineering interest group at Sheffield Hallam University for comments and reflections upon the WiZe tool. In addition, the author is grateful for the constructive advice provided by anonymous reviewers of an earlier version of this paper.

REFERENCES

- Barnard, P. J. and Harrison, M. D. (1992). Towards a framework for modelling human computer interactions. In Gornostaev, J., editor, *Proceedings International Conference on HCI, EWHCI'92*, pages 189–196. Moscow:ICSTI.
- Blandford, A. and Green, T. (1997). OSM an ontology-based approach to usability engineering. In *Representations in Interactive Software Development*. Workshop at Queen Mary and Westfield College, Department of Computer Science.
- Dearden, A. M. and Harrison, M. D. (1997). Abstract models for hci. *The International Journal of Human-Computer Studies*, (46):151–177.
- Dix, A. J. (1991). *Formal Methods for Interactive Systems*. Academic Press.
- Duke, D. J., Barnard, P. J., May, J., and Duce, D. A. (1995). Systematic development of the human interface. In *Proceedings of APSEC'95: Second Asia-Pacific Software Engineering Conference*. IEEE Computer Society Press.
- Gilmore, D. J. (1997). Cognitive dimensions as a tool for comparative evaluation. Technical report, Psychology Department, University of Nottingham.
- Green, T. and Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *The Journal of Visual Languages and Computing*, 7(2):131–174.
- Green, T. R. G. (1989). Cognitive dimensions of notations. In Sutcliffe, A. and Macaulay, editors, *People and Computers V*, pages 443–460. Cambridge University Press.
- Harrison, M. D., Blandford, A. E., and Barnard, P. J. (1993). The software engineering of user freedom. Technical Report Amodeus 2 Document, University of York.
- Jackson, M. (1997). The meaning of requirements. *Annals of Software Engineering*, 3.
- Lavery, D., Cockton, G., and Atkinson, M. (1996). Cognitive dimensions: Usability evaluation materials. Technical report, Department of Computing Science, University of Glasgow.
- Markopoulos, P., Rowson, J., and Johnson, P. (1997). Composition and synthesis with a formal interactor model. *Interacting with Computers*, 9(2):197–223.
- Modugno, F., Green, T. R. G., and Myers, B. A. (1994). Visual programming in a visual domain: A case study of cognitive dimensions. In Cockton, G., Draper, S. W., and Weir, G. R. S., editors, *People and Computers IX*, pages 91–108.
- Morrey, I., Siddiqi, J., Buckberry, G., and Hibberd, R. (1996). A toolset to support the construction and animation of formal specifications. *Journal of Systems and Software*.
- Nicola, R. D., Fantechi, A., Gnesi, S., and Ristori, G. (1991). An action based framework for verifying logical and behavioural properties of concurrent systems. In *Proceedings of 3rd Workshop on Computer Aided Verification*.

- Palanque, P. A. and Bastide, R. (1997). Synergistic modelling of tasks, users, and systems using formal specification techniques. *Interacting with Computers*, 9(2):129–154.
- Paterno, F. (1997). Formal reasoning about dialogue properties with automatic support. *Interacting with Computers*, 9(2):173–196.
- Roast, C. R. (1997). Formally comparing and informing design notations. In Thimbleby, H., O’Conaill, B., and Thomas, P., editors, *People and Computers XII*, pages 315–336.
- Roast, C. R. (1998). Designing for delay in interactive information retrieval. *Interacting with Computers*, 10:87–104.
- Roast, C. R. and Siddiqi, J. I. (1996). Relating knock-on viscosity to software modifiability. *Proceedings of OZCHI 96, Hamilton, New Zealand*, pages 222–227.
- Roast, C. R. and Siddiqi, J. I. (1997b). Usability requirements as specification constraints — an example of WYSIWYG. *IEE Proceedings Software Engineering*, 144(2):101–110.
- Siddiqi, J. I. and Roast, C. R. (1997). Viscosity as a metaphor for measuring modifiability. *IEE Proceedings — Software Engineering*, 144(4):215–223.
- Spivey, J. M. (1988). *The Z Notation: A Reference Manual*. Prentice Hall International.
- Yang, S., Burnett, M., DeKoven, E., and Zloof, M. (1995). Representation of design benchmarks: a design-time aid for VPL navigable static representations. Technical Report TR 95-60-3, Oregon State University.

Discussion

Gilbert Cockton: What would a developer lose by using UAN rather than modal logic to specify and analyse premature commitment?

Chris Roast: IT is not for developers, it is for theory developers, to improve cognitive dimensions.

Gilbert Cockton: Okay, so what would I lose from using UAN for theory development?

Chris Roast: What you lose and gain depends on the inferential mechanisms that come with UAN and logic.

Gilbert Cockton: UAN has no inferential mechanisms, but I can still spot premature commitment.

Chris Roast: That is okay, but you may not be able to prove it with UAN alone.

Ken Fishkin: What is the difference between formal statements and textual statements

Chris Roast: The property is more precisely represented in the formal statements and the inferences associated with them are more reliable.

Nick Graham: Have you attempted to use model checking techniques to examine the properties expressed by your ACTL specifications?

Chris Roast: Just started using model checking techniques.

Fabio Paterno: To apply model checking techniques you need to provide also some specification of the system that you want to analyse, what notation do you use for this purpose?

Chris Roast: We have started first exercises using Augmented State Machines encoded in Prolog.

John Grundy: I have been using cognitive dimensions in the analysis of high level design tools. In the case of premature commitment I've found that some users see premature commitment within such tools as a 'good' reflection of the methodology the tool supports, where as other users see the premature commitment as restrictive and under-desirable.

Chris Roast: The traditional view of cognitive dimensions is that there is no value judgement with a dimension hence one could not say whether a case of premature commitment is 'good' or 'bad' out of context. In the context you describe one account would be say that the same tool is being used in significantly different ways: one use

is for those who prefer the premature behaviour and adherence to a methodology; and, the other use is one which is not so closely allied to a prescribed method of use.

Joelle Coutaz: What about using the length of interaction trajectory to reason about premature commitment?

Chris Roast: I have related only viscosity with respect to interaction trajectory length. Hence trajectory length may give an measure of severity, however in the paper we only look at the concept and not its severity.

Henrik Christensen: How to relate premature commitment with technical description?

Chris Roast: Clearly the technical description looses characteristics of the informal concept. The work has progressed with aim of interpreting cognitive dimensions in terms that are relevant to interface engineers while remaining meaningful for usability engineers. Reflecting this, the interpretations examined so far are not “written in stone” but are best viewed as potentially valuable interpretations of cognitive dimensions, we are keen to entertain other interpretations and their refinement.