# DCCD: Dark Corners of Cognitive Dimensions

TRG Green (thos@comp.leeds.ac.uk)

The CDs concept seems so straightforward: a few classes of generic activity, a number of different structural properties, and some idea of how they interact .... but not so fast, Horatio. I shall list some of the areas that are unclear, at least to me, and I hope that someone else will make sense of them.

**Structural/environmental/cognitive classes of dimension?**
Some of the dimensions are properties of the notational structure, possibly alleviated by the environment; others seem to be imposed by the environment in which the notation is used.

*Hidden dependencies* seems to be purely structural; I would argue that the dependencies within spreadsheets, for example, are always expressed as one-way dependencies (cell A draws data from cell B), and that that is a property of the information structure/notation, possibly alleviated by auditing tools that can reveal the inverse dependencies. *Premature commitment*, in contrast, is imposed by environments that constrain the order of actions against the flow of the dependencies in the notation.

Unlike hidden dependencies and premature commitment, which have very little cognitive content, there are other dimensions that have very little else except cognitive content. For example, *discriminability* depends on the properties of the parsing system.

Should we, then, divide the dimensions into 3 groups ... ? Would that simplify things? Unfortunately I suspect that the division is not clean. *Viscosity* can be construed as a mismatch between the user's idea of a single action and the ideas built into the device. Is that structural or cognitive?

Worryingly, it seems to me that this inability to separate classes of dimension suggests that there is something wrong at a deep level.

**Media types**
The properties of different media have not been properly explored -- hardly even at the hand-waving level. Obviously paper is a persistent medium while speech is a transient medium, and there has been an off-hand suggestion that notations with many internal dependencies may be difficult to use in speech-based system because the transience is likely to impose order constraints, leading to premature commitment. (The paradigm example is creating a program: if spoken, dictation is likely to be from start to finish, which means that declarations must be made early; but if written, declarations can be made last, when it is known what declarations are needed.)

The distinction between persistent and transient media is too absolute. Media vary in how much history they reveal. If you cross out a word on paper, it leaves a scar; if you delete a number in a spreadsheet, it leaves a gap, which might be seen as a scar. But if you delete a word from a document in a word-processor the neighbouring words will fill the gap, so word-processors are 'self-healing' whereas paper, and to a lesser extent spreadsheets, are 'cicatrizing' (i.e. they form scar tissue). Whether that's important depends on whether the scar is useful to a person working with the document.

Then, of course, there are systems with history lists and history managers. Unix is no doubt the star example, with a sophisticated tool allowing users to repeat commands with alterations. Experienced users clearly find it very useful.

So the question is: would it be worthwhile to consider what notational structures and what activities benefit from systems with visible scarring (to remind users what they've been doing) and with history management?

**Temporal aspects**
One dark corner leads to another. The only notion of time-constraint in the framework as originally proposed was that some environments constrained order of actions, producing possible premature commitment. But time constraints are much more varied, and can sometimes be cleverly used by interaction designers: for example, cash machines (automated tellers) make you recover your credit card before they give you any money, which prevents the easy 'post-completion error' of collecting the money - which is the primary goal - and forgetting to take the card. (Designers of certain railway ticket machines have yet to rediscover this trick, and it seems that cards are regularly left in certain ticket machines at Leeds station.) In other cases, actions have to be performed in a set order; regardless of any possible premature commitment, we know from

studies of typing that order errors are frequent and in some cases highly consistent - for example, many mistypings result from regularizing the order of hand usage towards left-right-left-right instead of a sequence that is nearly regular but not quite.

Is there room to make the notion of temporality better-defined? if so, would it be cost-effective?

**Layers and focuses**

There has been little thought so far about what I call layers, although I expect there's a better name. A notational layer has an input notation and an output notation. That output might be the user's ultimate goal, but it might instead be the input to another layer. Thus, a text-edtior produces a character stream as an output notation. That stream can be re-interpreted as a program, the input notation to a compiler. The output of the program may be the user's ultimate goal - or instead, the output may be the input to a third layer, and so on. (E.g. characters become script for editor which rewrites a database which is input to a program which generates html which is scraped by a bot ...) With the growth of social software, we see cases where a document is produced by one person with one focus, and is then re-used by someone else for a different purpose: celebrity playlists become collectable podcasts.

The framework has little as yet to say about how the structure of one layer affects the possibilities in subsequent layers. (But see Payne et al, 1990 ). Nor has it much to say about the influence of the focus. In realtime live audio programming concerts (see Toplap) is the focus on the music produced, the virtuoso coding performance, or some of each?

**Meeting up**

What happens when two notations are in use? How do they link up? What are the resuting strengths and weaknesses? Examples are text with diagrams, code with comments, multiparadigm systems. What happens when more than one person is involved?

Toplap, 2005: http://www.toplap.org/

Payne, S. J., Squibb, H. R., and Howes, A. (1990) The nature of device models: the Yoked State Space hypothesis and some experiments with text editors. *Human-Computer Interaction*, 5, 415-444.