

Cognitive Dimensions of Notations: Design Tools for Cognitive Technology

A.F. Blackwell¹, C. Britton², A. Cox², T.R.G. Green³, C. Gurr⁴, G. Kadoda⁵,
M.S. Kutar², M. Loomes², C.L. Nehaniv², M. Petre⁶, C. Roast⁷, C. Roes⁸, A. Wong⁸
and R.M. Young²

¹ University of Cambridge, ² University of Hertfordshire, ³ University of Leeds, ⁴ University of Edinburgh, ⁵ University of Bournemouth, ⁶ Open University ⁷ Sheffield Hallam University
Alan.Blackwell@cl.cam.ac.uk

Abstract. The Cognitive Dimensions of Notations framework has been created to assist the designers of notational systems and information artifacts to evaluate their designs with respect to the impact that they will have on the users of those designs. The framework emphasizes the design choices available to such designers, including characterization of the users activity, and the inevitable tradeoffs that will occur between potential design options. The resulting framework has been under development for over 10 years, and now has an active community of researchers devoted to it. This paper summarizes the current activity, especially the results of a one-day workshop devoted to Cognitive Dimensions in December 2000, and reviews the ways in which it applies to the field of Cognitive Technology.

Introduction

The title of this meeting is *Cognitive Technology: Instruments of Mind*. In this paper, we try to characterize the ways that the instruments of our minds are compromised by the restrictions that our bodies and physical environment place on them. This can be regarded as a proposed approach to the study and practice of cognitive ergonomics. Let us consider a (trivially) simple example to start with. Any cognitive technology transfers information from our heads to our physical environment so that we can “offload” it from short-term memory, and also so we can interact with it. A piece of paper with visible marks on it is one of the simplest such technologies. A very large piece of paper with many small marks can carry a great deal of information, and represent complex structures. But there are limits imposed on this information and its complexity. They are not imposed by the piece of paper (which can be made arbitrarily large) but by our bodies. There is a limit on the ability of our eyes to see far away, and especially on their ability to resolve small marks that are far away. These limitations have predictable effects on the value of this particular cognitive technology: Where we might want to gain a visual overview of the whole information structure, we cannot do so because we can’t see all of it at once. When we need to refer to some specific component of the information, we must search for it by scanning the paper a section at a time.

These observations may seem trivial in the case of large pieces of paper, which are limited (to say the least) as representatives of cognitive technology. But the limitations can be even more severe in more advanced cognitive technologies. Digital technologies can record far more information than single sheets of paper, and they can describe far more complex information structures, not limited by a two dimensional surface. But despite the promise of ubiquitous computing, wall-sized displays and intelligent paper, we generally find that the computer screen only offers a restricted window onto these large and complex information structures. This means that the problem of visibility – initially only a physical restriction imposed by our eyes – also becomes a problem of how to control mechanisms for scrolling and zooming. If we use our arms, hands and fingers to operate them, the simple problem of reading an information display becomes compromised by our bodily limits: manual dexterity, reaction times, positional stability and other factors.

Thus far we have only considered the question of *visibility*, and we have assumed that the user of this information artifact is simply reading information off the display. In fact most of the interesting applications of cognitive technology involve more complex activities – creating information structures, modifying them, adding information to them, or exploring possible design options for completely new information structures. Visibility is an important consideration for almost all of these activities, but many of them place additional constraints on the user beyond simple physical perception and interaction. Examples include *viscosity* – the difficulty of making small changes to the information structure; *provisionality* – ways in which the user can express parts of the structure that are not yet precisely defined; and many others. We call these attributes of information artifacts the *Cognitive Dimensions of Notations* (CDs). In the same way that visibility has a predictable relationship to important aspects of the cognitive activity of reading (above we observed ability to see overall structure,

efficiency of searching for specific components), so the other CDs can be used to predict the consequences of using an information artifact for other types of activity.

Who needs this kind of analysis? It is clear that we are not saying anything profound about human cognition. Neither are we saying anything new about sophisticated information structures, algorithms or tools. The reason that the CDs framework has been developed is that people who are designing new information artifacts – the developers of cognitive technologies – often find themselves encountering the same problems over and over again when designing different systems.

Expert designers of cognitive technologies learn by experience, and eventually (with luck) produce well-designed information artifacts that are appropriate to the user's activity. Unfortunately many developers of new cognitive technologies are not expert at anticipating and providing for the user's needs. They are computer scientists or engineers who understand the technical problems they are addressing far better than they understand the problems of the user. We believe that this problem is best addressed by providing a vocabulary for discussing the design problems that might arise – a vocabulary informed by research in cognitive psychology, but oriented toward the understanding of a system developer. The Cognitive Dimensions of Notations are such a vocabulary. There are other techniques for analyzing the usability of computer systems, but these often focus on the finest details of interaction – key-press times, visual recognition or memory retrieval. Instead the CDs framework attempts to describe the most relevant aspects of our interaction with information artifacts at a broad-brush level, intended to be useful as a discussion tool for designers.

The Role of Cognitive Dimensions

Cognitive Dimensions of Notations (CDs) is a framework for describing the usability of notational systems (e.g. word processors, computer-aided design tools, or music notation) and information artifacts (e.g. watches, radios or central heating controllers). The CDs framework aims to do this by providing a vocabulary that can be used by designers when investigating the cognitive implications of their design decisions. Designers of notational systems do realize that their decisions have an impact on usability, and that the usability problems with notations have cognitive implications for the user. But many designers only know those things in an intuitive way. This makes it difficult for them to discuss usability issues, especially as they seldom have any formal education in cognitive psychology.

This situation becomes more serious in cases where the design process involves making decisions about design tradeoffs. Perhaps the design can be improved in one respect, but only at the expense of making it worse in some other respect. Or perhaps it can be made more appropriate for a particular user group (e.g. the elderly), but only at the expense of becoming less usable for some other user group (e.g. those who have very little time). Or more insidious, perhaps the design can be altered so that it is suitable for users when they are carrying out a certain task, but then becomes unusable for another important task. As an example, consider a notation that expresses some complex procedure on a screen in flow diagram form. Flow diagrams make the possible interactions between different events a lot clearer, but they take up more room on the screen than a simple textual list. And if the user is actually modifying the diagrams, all the connecting lines make it more difficult to change the diagram because they have to be moved around and tidied up after changes. These are generic properties of notational systems, which CDs describe by names like *hidden dependencies* (the visibility of relationships) *diffuseness* (the amount of space that the notation takes up) and *viscosity* (the amount of effort required to make small changes to the notation).

None of these is necessarily a problem; that depends on what the user wants to do – e.g. viscosity is not a problem if the user doesn't need to make any changes. So the framework considers dimensions in the context of user activities.

The CDs framework has been designed for situations where the designer is making choices about notations or representations, and where usability tradeoffs are a factor in the design. It is particularly difficult to design new notational systems and information artifacts. CDs describe some common properties of notations that allow the designer to anticipate the effect of design changes, and make more conscious choices about tradeoffs without having actually to build and evaluate prototypes.

The development of the CDs framework was initiated by Thomas Green in a 1989 publication (Green 1989). Since then over 50 research papers have been published on topics related to the CDs, including a longer description applying CDs to the domain of visual programming languages (Green and Petre 1996) and a tutorial aimed at professional designers (Green and Blackwell 1998). This paper reports the results of a meeting held in December 2000 of researchers who are currently pursuing projects related to CDs. It describes how the state of the art in CDs research can contribute to the overall objectives of cognitive technology.

Summary of the CDs Framework

As mentioned above, we describe CDs as providing, not only a vocabulary for use by designers, but a framework for thinking about the nature of notational systems and the way that people interact with them. This framework provides a structure in which to understand the vocabulary itself, but also includes a number of theoretical activities that extend beyond the demands of many designers applying the vocabulary in more restricted design contexts.

The framework includes definitions of notations and notational systems, characterization of the human activities involving notational systems, a description of the ways that multiple notations can interact within a single system, and a minimal process for applying the resulting insights in a design context for use in evaluating and improving a design. More recently, as larger numbers of researchers have adopted the CDs framework as a research tool, the framework has also developed some reflective components applicable to extending and refining the framework itself. The later parts of this paper, and the workshop from which it has been derived, deal with this latter aspect.

However it is first necessary to review the established parts of the framework. We start with the definitions of notational systems. A notation consists of marks (often visible, though possibly sensed by some other means) made on some medium. Examples include ink on paper, patterns of light on a video screen, and many others. It is possible for several notations to be mixed within a single medium: a computer screen may display multiple windows, each running a different application with its own notation. Even within a window, there may be multiple notations – the main notation of the application, but also generic sub-notations such as menu bars, dialogs etc. A notational system contains both a notation and an environment (such as an editor) for manipulating that notation. CDs describe usability properties of the system, not just the notation. Where the system includes sub-notations, users generally interact with them through sub-devices, which have their own cognitive dimensions. We describe some self-contained notational systems as “information artifacts”. These include things such as telephones, central heating controls, and many ubiquitous automated systems beyond the range of typical computer applications. In all these cases, the notation expresses some structure, more or less complex.

It is important to note that none of the cognitive dimensions are necessarily good or bad by themselves. The usability profile of a system or artifact depends on what kind of activity the user will be engaging in, and on the structure of the information contained in the notation. The activities that are least demanding in terms of usability profile are simply *searching* for a single piece of information (such as looking up a name in a telephone book) and *incrementally understanding* the content of the information structure expressed by a notation (such as reading a textbook). The more interesting activities are those that involve extending the notation: *incrementing* an existing structure by adding new information, *transcribing* information from one notational form to another, *modifying* the structure, or exploring possible new information structures in *exploratory design*.

These are the main theoretical foundations of the framework – at this point we will give a brief review of the set of dimensions, with thumbnail definitions of each. These descriptions are very brief – note that they are more fully described, with illustrative examples and explanation, in many other publications, including a tutorial that is available online (Green and Blackwell 1998).

Review of Dimensions

Viscosity: resistance to change.

A viscous system needs many user actions to accomplish one goal. Changing all headings to upper-case may need one action per heading. (Environments containing suitable abstractions can reduce viscosity.) We distinguish repetition viscosity, many actions of the same type, from knock-on viscosity, where further actions are required to restore consistency.

Visibility: ability to view components easily.

Systems that bury information in encapsulations reduce visibility. Since examples are important for problem-solving, such systems are to be deprecated for exploratory activities; likewise, if consistency of transcription is to be maintained, high visibility may be needed.

Premature commitment: constraints on the order of doing things.

Self-explanatory. Examples: being forced to declare identifiers too soon; choosing a search path down a decision tree; having to select your cutlery before you choose your food.

Hidden dependencies: important links between entities are not visible.

If one entity cites another entity, which in turn cites a third, changing the value of the third entity may have unexpected repercussions. Examples: cells of spreadsheets; style definitions in Word; complex class hierarchies; HTML links. There are sometimes actions that cause dependencies to get frozen – e.g. soft figure numbering can be frozen when changing platforms; these interactions with changes over time are still problematic in the framework.

Role-expressiveness: the purpose of an entity is readily inferred.

Role-expressive notations make it easy to discover why the programmer or composer has built the structure in a particular way; in other notations each entity looks much the same and discovering their relationships is difficult. Assessing role-expressiveness requires a reasonable conjecture about cognitive representations (see the Prolog analysis below) but does not require the analyst to develop his/her own cognitive model or analysis.

Error-proneness: the notation invites mistakes and the system gives little protection.

Enough is known about the cognitive psychology of slips and errors to predict that certain notations will invite them. Prevention (e.g. check digits, declarations of identifiers, etc) can redeem the problem.

Abstraction: types and availability of abstraction mechanisms.

Abstractions (redefinitions) change the underlying notation. Macros, data structures, global find-and-replace commands, quick-dial telephone codes, and word-processor styles are all abstractions. Some are persistent, some are transient.

Abstractions, if the user is allowed to modify them, always require an abstraction manager -- a redefinition sub-device. It will sometimes have its own notation and environment (e.g. the Word style sheet manager) but not always (for example, a class hierarchy can be built in a conventional text editor).

Systems that allow many abstractions are potentially difficult to learn.

Secondary notation: extra information in means other than formal syntax.

Users often need to record things that have not been anticipated by the notation designer. Rather than anticipating every possible user requirement, many systems support secondary notations that can be used however the user likes. One example is comments in a programming language, another is the use of colours or format choices to indicate information additional to the content of text.

Closeness of mapping: closeness of representation to domain.

How closely related is the notation to the result it is describing?

Consistency: similar semantics are expressed in similar syntactic forms.

Users often infer the structure of information artefacts from patterns in notation. If similar information is obscured by presenting it in different ways, usability is compromised.

Diffuseness: verbosity of language.

Some notations can be annoyingly long-winded, or occupy too much valuable “real-estate” within a display area. Big icons and long words reduce the available working area.

Hard mental operations: high demand on cognitive resources.

A notation can make things complex or difficult to work out in your head, by making inordinate demands on working memory, or requiring deeply nested goal structures.

Provisionality: degree of commitment to actions or marks.

Premature commitment refers to hard constraints on the order of doing things, but whether or not hard constraints exist, it can be useful to make provisional actions – recording potential design options, sketching, or playing “what-if” games.

Progressive evaluation: work-to-date can be checked at any time.

Evaluation is an important part of the design process, and notational systems can facilitate evaluation by allowing users to stop in the middle to check work so far, find out how much progress has been made, or check what stage in the work they are up to. A major advantage of interpreted programming environments such as BASIC is that users can try out partially-completed versions of the product program, perhaps leaving type information or declarations incomplete.

Application

In a design context, the dimensions would be applied after identifying a “main” notation to be analysed. In the course of the analysis, sub-devices might be identified, offering separate notations for purposes such as extending the main notation (an abstraction manager sub-device). The designer would assess usability with respect to some activity profile describing the activities that the user is likely to carry out. The dimensional characteristics of the notational system can then have their implications assessed with respect to that profile. Where problems are identified, the framework offers design maneuvers by which those problems might be addressed although they potentially involve tradeoffs, in which changing the design of the notational system on one dimension may result in additional changes of the system properties on another dimension.

Current Frontiers in CDs Research

This section summarizes the presentation, some discussion, and the results from the December workshop described above.

Activities and Profiles

Profiles are where users’ activities mesh with the cognitive dimensions of the notation: a profile specifies what is needed to support an activity. No dimension is evaluative on its own - one can't know whether it is relevant until one knows what activity is to be supported. There have been several attempts to define a broadly useful set of generic activities. Hendry and Green (1994) defined three different types of activity using notational structures: incremental growth, transcription and presentation. This list has been refined in various ways. The original CDs tutorial defined four activities in constructing notations: *transcription*, *incrementation*, *modification* and *exploratory design*. Soon afterward the CDs questionnaire for user evaluation added a fifth: *search*.

The December workshop also considered the newly proposed *exploratory understanding*, which is relevant both to notational tools such as software visualization systems, and to distributed notations such as the world-wide web. We expect that this will offer new insights from related analysis techniques such as information foraging theories. There may be further activities related to other areas of human activity that have not yet been addressed by CDs to date. The workshop offered some possible new activities including *play*, *competition*, and *community building*. But these are dangerous – the addition of new activities introduces credibility obstacles for the framework to a greater extent than the addition of new dimensions does.

We also feel that the activities are currently formulated in too abstract a way, despite the fact that they are critical to the evaluative use of CDs. We have taken great pain that every dimension should be described with illustrative examples, case studies, and associated advice for designers. Activities, on the other hand, are described at a rather abstract level in terms of the structure of information and constraints on the notational environment. This makes it unlikely that usability profiles will be exploited effectively. The workshop concluded that the activities must be paraphrased in everyday language to make them as accessible to designers as the dimensions themselves. These descriptions will be supplemented by examples of relevant tasks, some of which may be juxtaposed within the context of a specific class of information artifact: this is currently being pursued through a series of simulated central heating controllers, implemented in JavaScript, and available on the web through the CDs archive site.

Britton and Jones have studied the creation of a CDs profile for a specific task: the validation of a requirements specification. They reported this work at the December workshop. They wished to evaluate the comprehensibility of different specification languages for non-specialist readers. This profile therefore measured the *intelligibility* of specifications, characterized by the user activities of a) extracting information from the representation and b) checking the correspondence of the represented information with existing knowledge. These activities are not externally observable, but form the basis for user activities that can be observed.

Selecting a limited set of dimensions resulted in a more streamlined profile and allowed them to concentrate on those dimensions that were of particular interest. These were then used to compare two specifications of the temporal aspects of an interactive system. One was written in the logic language TRIO_≠, the other in a version of TRIO_≠ extended to make temporal properties easier to understand. The conclusion was that prior selection of a subset of CDs may be unhelpful. Using the full set of dimensions can produce some unexpected, but useful results and should be done in order to discover as much information as possible. This suggests that profiles should describe the weighting of dimensions for different activities, rather than attempting to eliminate dimensions.

The evaluation of some notational system should always be conducted according to a defined profile of use – we suggest that this might be called a profile *instance*, as opposed to more generic sets of dimensions with associated consequences and tradeoffs that would be called a profile *class*. The result of assessment for a specific

profile instance is a CD's *assessment*. CD's assessment can be achieved by relatively untrained users of CDs, while the creation of new profile classes is more difficult, potentially requiring the assistance of CD's researchers acting as consultants. This effort might be reduced by creating profile *clusters* that describe a group of related profiles. The process of assessment itself will be facilitated by having a better-constructed set of standard questions, such as: what is the notation of the main device; how do the dimensions apply to it; what abstractions are available; are there abstraction managers; and are the abstractions transient or persistent?

Tradeoffs

Trade-offs are frequently-observed patterns in CD's analyses – they are situations in which one source of difficulty is fixed at the expense of creating another type of difficulty. At present too little is known about what trade-offs occur in real life, but some observations will be reported. Questions arising are: are these tradeoffs correctly identified and specified? Are they always correct, or only in certain situations? Can we find more examples? Is there a methodology we can use to account for and correct them? How do we (CD's researchers) communicate the ideas for use by designers?

One way to communicate is by looking for everyday language; see for instance the questionnaire developed by Blackwell and Green. This questionnaire, along with other resources, is available from the CD's archive site – a URL is included in the bibliography. Another attempt at communication is to present working examples for consideration. All the examples need to present alternative solutions, in a minimalist form, in order to emphasize the tradeoffs. Some examples can be seen at the following URL:

<http://www.ndirect.co.uk/~thomas.green/workStuff/devices/controllers/HeatingA2.html>

Formalisation

Several current research projects are investigating approaches to formalization of CD's. At its most basic level such a theory would be expected to be descriptively adequate - replicating examples of cognitive dimensions. A more mature theory would be expected to predict instances of dimensions and provide general theorems regarding cognitive dimensions. Clearly, the eventual goal is a theory which is valid within recognized boundaries and which is capable of directly contributing to our understanding.

To aid the process of validation Roast et. al. have developed a tool for modeling formal interpretations of the dimensions (called CiDa). The tool is designed to support theory validation through enabling the consequences of posited CD definitions to be examined. CiDa analysis requires that the target system is modeled in terms of simple non-deterministic state based machine and that states of this machine are associated with potential user goals. The objective of this work is to develop CD's theory through an example-driven approach, where it is the artifact that is modeled rather than the cognitive processes. The ideal is that it should be possible to observe the artifact, model it, and then validate the model. CiDa creates formal models of a variety of tasks, rather than being restricted to tasks that have been selected to illustrate specific CD's.

The Empirical Modelling (EM) research group based at the University of Warwick aim to analyze artifacts by focusing on identifying patterns of agency and dependency through observation and experiment, and embodying these patterns in computer models called Interactive Situation Models (ISMs). An artifact comprises many different aspects of state. The explicit state is the visible state of the artifact. The internal state is all the physical states of the information artifact. The mental state is the state that users project upon the artifact when considering expectations about possible next state/interpretation of current state. The situational state is knowledge of the real world context to which the artifact refers. The EM group suggest that the CD's can be applied to understand how each of the above aspects of state interact in trying to make appropriate use of an information artifact. Their current research indicates that the construction of an ISM of an artifact may give a modeller a better understanding of the CD's of that artifact. This work is reported in detail elsewhere in this meeting. For more information see: <http://www.dcs.warwick.ac.uk/modelling>.

Operationalization

An alternative approach to normalization is operationalization: identifying practical questions and activities that help designers of information artifacts to reason about cognitive consequences of making a particular collection of design choices. This work starts from the perspective that cognitive dimensions lay out a design space, and that they provide a 'broad brush' framework supporting reasoning about how those choices place the design in the space. Again, cognitive dimensions are not binary, but descriptive, establishing where in a space of inter-related factors and choices a design lies.

As demonstrated in the Green and Petre (1996) paper, this approach identifies pragmatic 'yardsticks' and 'straw tests'. These are not canonical or definitive tests, but simply a set of practical questions used to fuel a

cognitive dimensions analysis. They are cast in operational terms: they enquire how the effects of the dimension translate into work required. They are meant both to make the evaluation concrete and to provide a basis for comparison between designs or design choices. For example, regarding 'Hidden dependencies': Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic? Or regarding 'Imposed Look-Ahead': Are there order constraints? Are there internal dependencies?

For some dimensions, we also apply some 'straw' tests: simple tests based on typical activities (modification, examination, comparison) and chosen to measure 'work done' in terms of the dimension. For example, timing typical modifications in order to evaluate 'viscosity'.

The value in this approach is its immediacy; the usage is pragmatic and accessible, making a cognitive dimensions analysis a low-cost tool to add to a design repertoire. Putting CDs readily into use is the best way to demonstrate their relevance to practice. But the process of operationalization itself is informative and feeds back into cognitive dimensions theory, giving perspective on definitions and concepts, exposing interrelationships among design choices, reflecting on the impact of tasks and environments, and so on.

Extending the Framework with New Dimensions

The core of the Cognitive Dimensions of Notations framework is the list of dimensions itself. This list has been gradually expanding – Thomas Green's early publications (Green 1989,1990,1991) described only a few selected dimensions, as did other researchers in early publications (Gilmore 1991). By the time the Green and Petre (1996) paper was published, 13 dimensions were listed. Green and Petre did not claim that the set of dimensions was then complete. On the contrary, they have continued to encourage discussion of new additions. As it turns out, the process of defining new dimensions has slowed down. This may partly be because the existence of a definitive publication made the initial step of defining one more dimension a daunting one. More importantly, few researchers have seen the addition of new dimensions as an important end in itself. The 1996 paper, under the heading of "Future progress in cognitive dimensions", observed that the framework was incomplete – but not in the sense that more dimensions were urgently needed. Rather it emphasized the need for formalization and applicability.

Nevertheless, new dimensions do get proposed from time to time. Some of these proposals have been published, but more of them exist only in the form of informal conversations with Green and other central researchers. But it is neither necessary nor desirable for the development of the framework to depend on any individual acting as a gatekeeper / coordinator for new additions. The December workshop therefore considered possible future approaches to the process of identifying and defining new Cognitive Dimensions.

Some examples

Some examples of a few candidate dimensions, taken from informal sources, are included here. Some of these have been published before, but most are appropriated from other research fields (in the sense that they are inspired by authors who did not consider themselves to be working on cognitive dimensions). None of them should be considered at this stage to have canonical status – in fact the question of how to assemble the canon is the main topic of discussion.

Creative Ambiguity

The extent to which a notation encourages or enables the user to see something different when looking at it a second time (based on work by Hewson (1991), by Goldschmidt (1991), and by Fish and Scrivener (1990))

Specificity

The notation uses elements that have a limited number of potential meanings (irrespective of their defined meaning in this notation), rather than a wide range of conventional uses (based on work by Stenning and Oberlander 1995)

Detail in context

It is possible to see how elements relate to others within the same notational layer (rather than to elements in other layers, which is role expressiveness), and it is possible to move between them with sensible transitions, such as Fisheye views (based on work by Furnas (1986) and by Carpendale, Cowperthwaite and Fracchia (1995))

Indexing

The notation includes elements to help the user find specific parts.

Synopsis

(originally “grokkiness”) The notation provides an understanding of the whole when you “stand back and look”. This was described as “Gestalt view” by some of the respondents in the survey by Whitley and Blackwell (1997).

Free rides

New information is generated as a result of following the notational rules (based on work by Cheng (1998) and by Shimojima (1996))

Useful awkwardness

It’s not always good to be able to do things easily. Awkward interfaces can force the user to reflect on the task, with an overall gain in efficiency (based on discussions with Marian Petre, and work by O’Hara & Payne (1999))

Unevenness

Because things are easy to do, the system pushes your ideas in a certain direction (based on work by Stacey (1995))

Lability

The notation changes shape easily

Permissiveness

The notation allows several different ways of doing things (based on work by Thimbleby, not yet published).

Where do they come from?

As is apparent from the above list, most candidates for new dimensions come from other research, whether or not the author is aware of the CDs framework. This is a good thing. One objective of CDs is that they should be credibly derived from psychological or cognitive science research. This is largely what gives them authority among notation designers (and the implication is intentional, through the use of the word “cognitive”).

This suggests that an immediate point of good practice would be to encourage the participation of the original researchers in the process of defining new dimensions. This would obviously include due credit via citation of the author’s original work, as well as the opportunity for the original author to review the dimension derived from his or her work – both our characterization of the dimension itself, and the way that it is related to the rest of the framework through profiles, tradeoffs, dependencies and design manoeuvres.

Criteria for acceptance

What are the criteria that define a good (or even an acceptable) new cognitive dimension of notations? The process by which the current set were derived has been the subject of reflection, but not thorough documentation.

As the number of dimensions grows, it is also becoming crucial to identify a useful subset for new users (including undergraduate courses). Commercial users are already impatient with the size of the set that exists now. We could perhaps create a CDs-lite for commercial friends – perhaps with 7 plus or minus 2 dimensions. These might be selected as the most important, or possibly the easiest to understand. We might possibly adopt Jack Carroll’s minimal documentation approach to presentation, so that people only have to deal with the dimensions that they need.

Orthogonality

Most important, the term “dimension” was chosen to imply that these are mutually orthogonal – they all describe different directions within the design space. Furthermore, it is hoped that the trade-off relationships between them might be similar to those of the Ideal Gas Law – so that it is probably not possible to design a notation system that achieves specific values on any two dimensions, without having the value of a third imposed by necessary constraints. But these notions of orthogonality are intuitive rather than exact, and they are described in this way mainly so that designers recognize the nature of the constraints on their design. There is ongoing work on formalization of dimensions that should allow more precise statements to be made regarding orthogonality and trade-offs for a few dimensions, but such analysis cannot yet be required when proposing new dimensions.

Instead, mutual orthogonality can only really be tested at present via a qualitative approach – going through all current dimensions, and checking to see whether any of them might describe the same phenomenon as that described by the proposed new dimension. This checking ought to be done by more than one person. It is so

common for individual researchers to misunderstand the nature of one or two of the dimensions, that it is highly likely a proposed new dimension will simply be a rediscovery of an existing dimension (which the researcher had understood to refer to something else). It is also necessary to be aware that the new dimension might simply be the obverse case of an existing dimension.

Granularity

The CDs seem to describe activities at a reasonably consistent level of granularity. They should probably continue to describe phenomena at a similar scale. They do not directly describe large cognitive tasks (design a system, write a play), but the structural constituents of those tasks. They also tend not to describe low-level perceptual processes. Some things that are too low a level of granularity might include Gestalt phenomena, or observations related to individual motions (e.g. selection target size, as analyzed by Fitts' law). If they were to be characterized using GOMS analysis, we might say that CDs do not apply either to leaf nodes in the goal tree, or to the whole tree, but to sub-trees.

Object of description

There is an outstanding question regarding what it is that the dimensions are supposed to describe. Some possible options for suitable objects of description (no doubt not a complete list) are:

- (i) structural properties of the information within the notation/device
- (ii) the external representation of that structure
- (iii) the semantics of that information
- (iv) the relationship between the notated information and domain-level concepts – some of which are inevitably not notated

Depending on which of these are chosen, the CDs field gets bigger or smaller. Useful awkwardness and permissiveness are both defined partly by domain-level concepts, so they might not be members of the CDs list, if we restrict objects of description to (say) (i) & (ii).

Effect of manipulation

It ought to be possible to consider each dimension and say 'if you change the design in the following way, you will move its value on this dimension'. This is a criterion of understanding how the dimension works, as well as the basis for design manoeuvres. When we define a new dimension, we should be able to say something about how to manipulate it.

Applicability

One of the desirable properties of a CD is that it should make sense to talk about it in a wide range of different situations. This has not always been achieved with the current set of dimensions.

Polarity

As CDs are not supposed to be either good or bad (more on this below), they should have interesting properties in both directions – i.e. both when present and absent. Error-prone-ness is not a very good dimension when considered from this perspective.

Choosing names

It is hard to find good names for new dimensions. "Grokkiness" (which persisted for almost a year) shows just how hard it is! Some of the criteria for good names include:

Length of name

It seems like one or two words should be enough (Closeness of Mapping is really on the limit).

Vernacularity

CDs should sound both technical and approachable at the same time. They must sound sufficiently technical that they don't get confused with everyday meanings, and that they can be accorded some respect by notation designers. In an effort to get something sufficiently technical, we have sometimes had mixed results, either by resorting to neologism (grokkiness) or archaism (synopsis).

There is also a problem of cultural specificity. It turns out that knock-on viscosity is unintelligible to Americans (recently reported by Margaret Burnett, and confirmed by several other delegates at VL2000). Some Americans guess correctly, but others think that it might have something to do with door knockers. They have suggested "domino" or "consequent" viscosity – is either of these too technical, or too approachable?.

Polarity

It gives a false impression of the CDs framework if readers treat the dimensions as representing “usability problems” rather than trade-offs. But this constantly happens, especially if the audience is already familiar with Nielsen’s heuristic analysis of usability. We have partly caused the problem ourselves, because most of the names do imply negative consequences “Hidden dependencies” rather than “Visible dependencies”, for example. There are several options for addressing this problem:

- Choose neutral names (desirable, but hard to achieve).
- Purposely choose names with alternating obverse polarities.
- Choose positive names if at all possible (to avoid the usability problem assumption).
- Provide dual definitions for all dimensions, illustrating positive and negative aspects.

With regard to polarity, it is also important to remember that dimensions only become evaluative when applied to some specific activity. For this reason, it should be possible to describe the characteristics of a dimension without any evaluative emphasis – evaluative observations should ideally be localised within the profile.

Supporting Apparatus

A cognitive dimension is more than just a name and a definition. All of the current dimensions are supported by a range of documentary and tutorial apparatus.

Examples

Each dimension is supported by examples of situations in which it can occur, with the consequences of that occurrence. There should be one “killer example” that immediately reveals to the reader the essence of the dimension. Ideally, examples should be drawn both from programming and other user interface domains.

Pictorial examples

In future, it would be very useful for every “killer example” to be supported by a pictorial illustration that can be incorporated in published papers referring to and citing the dimension. There is no real harm in repeating the same illustration, and a nicely illustrated example would help to promulgate CDs as a whole. We hope to add some examples of such reusable illustrations to the Cognitive Dimensions archive site.

Impact

Different dimensions have different impacts on various activity types and profiles. Some kind of characterisation should be attempted.

Trade-offs

Should be noted. But if there is a specific trade-off that invariably occurs, that might be a sign that this dimension is only the obverse case of an existing dimension, rather than an orthogonal dimension.

Sources

Research sources should be cited, both as supporting evidence, and also to give appropriate credit to previous researchers.

Manœuvres and workarounds

It is valuable to have some observations regarding design manœuvres and also the ways that users might try to work around the effects of the dimension.

Conclusion

Many of the usability evaluation methods that have been applied to cognitive technologies in the past were derived from models of machine ergonomics, stressing manual efficiency rather than appropriateness to the user. A reaction to this has now led to an alternative emphasis on anecdotal transfer of trade skills and aesthetic criteria (as in the highly popular books of Tufte). The current usability criteria for activities such as Information Architecture for Web design combine these cognition-free accounts of design criteria with an idealized view of the contributions offered by technological innovation.

The CDs framework offers an account of information artifacts that respects the value of the user's activity, seeking to recognize the cognitive constraints that the artifact places on that activity. This is very much in accordance with the overall goals of the Cognitive Technology field.

The CDs framework has, over the last 10 years, developed into a useful tool. But it is not complete, and further work remains to be done. This paper has presented a "state of the nation view" from active researchers in the field, and also offered a joint agenda for ongoing research. Within the context of Cognitive Technologies, this has served two purposes. First, the ultimate goals of the CDs framework are closely aligned with those of Cognitive Technology, and we wish to see further cross-fertilisation in future. Second, we have offered in this paper an insight into the process of developing and maintaining a theoretical framework as it is transferred into the wider research community and also to industrial practitioners. We believe that this process of "rubbing up against" a broader community of users and collaborators has enriched the CDs framework. This is an experience that we recommend to other researchers developing theoretical models of Cognitive Technology.

References

Note that many of these publications are available online from the Cognitive Dimensions archive site: <http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/>

- Blackwell, A.F. & Green, T.R.G. (2000). A Cognitive Dimensions questionnaire optimised for users. In A.F. Blackwell & E. Bilotta (Eds.) *Proceedings of the Twelfth Annual Meeting of the Psychology of Programming Interest Group*, 137-152.
- Carpendale, M.S.T., Cowperthwaite D.J. and Fracchia, F. D. (1995). 3-Dimensional pliable surfaces for the effective presentation of visual information navigation. *Proceedings of the ACM Symposium on User Interface Software and Technology* p.217-226.
- Cheng, P.C. (1998). AVOW diagrams: A novel representational system for understanding electricity. In *Proceedings Thinking with Diagrams 98: Is there a science of diagrams?* pp. 86-93.
- Fish, J. & Scrivener, S. (1990). Amplifying the mind's eye: Sketching and visual cognition. *Leonardo*, **23**(1), 117-126.
- Furnas, G.W. (1986). Generalized fisheye views visualizing complex information spaces. *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems* p.16-23.
- Gilmore, D. J. (1991) Visibility: a dimensional analysis. In D. Diaper and N. V. Hammond (Eds.) *People and Computers VI*. Cambridge University Press.
- Goldschmidt, G. (1991). The dialectics of sketching. *Creativity Research Journal*, **4**(2), 123-143.
- Green, T. R. G. & Petre, M. (1996) Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages and Computing*, **7**, 131-174.
- Green, T. R. G. (1989). Cognitive dimensions of notations. In *People and Computers V*, A Sutcliffe and L Macaulay (Ed.) Cambridge University Press: Cambridge., pp. 443-460.
- Green, T. R. G. (1990) The cognitive dimension of viscosity: a sticky problem for HCI. In D. Diaper, D. Gilmore, G. Cockton and B. Shackel (Eds.) *Human-Computer Interaction — INTERACT '90*. Elsevier.
- Green, T. R. G. (1991) Describing information artefacts with cognitive dimensions and structure maps. In D. Diaper and N. V. Hammond (Eds.) *Proceedings of "HCI'91: Usability Now"*, Annual Conference of BCS Human-Computer Interaction Group. Cambridge University Press.
- Green, T.R.G. & Blackwell, A.F. (1998). Design for usability using Cognitive Dimensions. Tutorial presented at *British Computer Society conference on Human Computer Interaction HCI'98*. Available online from the Cognitive Dimensions archive site <http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/>
- Hendry, D. G. and Green, T. R. G. (1994) Creating, comprehending, and explaining spreadsheets: a cognitive interpretation of what discretionary users think of the spreadsheet model. *Int. J. Human-Computer Studies*, **40**(6), 1033-1065.
- Hewson, R. (1991). Deciding through doing: The role of sketching in typographic design. *ACM SIGCHI Bulletin*, **23**(4), 39-40.
- O'Hara K.P., and Payne, S.J. (1999). Planning and the user interface: The effects of lockout time and error recovery cost *International Journal of Human-Computer Studies* **50**(1), 41-59.
- Shimojima, A. (1996). Operational constraints in diagrammatic reasoning. In G. Allwein & J. Barwise (Eds) *Logical reasoning with diagrams*. Oxford: Oxford University Press, pp. 27-48.
- Simos, M. & Blackwell, A.F. (1998). Pruning the tree of trees: The evaluation of notations for domain modeling. In J. Domingue & P. Mulholland (Eds.), *Proceedings of the 10th Annual Meeting of the Psychology of Programming Interest Group*, pp. 92-99.
- Stacey, M. K. (1995) Distorting design: unevenness as a cognitive dimension of design tools. In G. Allen, J. Wilkinson & P. Wright (eds.), *Adjunct Proceedings of HCI'95*. Huddersfield: University of Huddersfield School of Computing and Mathematics.
- Stenning, K. & Oberlander, J. (1995). A cognitive theory of graphical and linguistic reasoning: Logic and implementation. *Cognitive Science*, **19**(1), 97-140.
- Whitley, K.N. and Blackwell, A.F. (1997). Visual programming: the outlook from academia and industry. In S. Wiedenbeck & J. Scholtz (Eds.), *Proceedings of the 7th Workshop on Empirical Studies of Programmers*, pp. 180-208.