A Cognitive Dimensions Questionnaire Optimised for Users

Alan F. Blackwell and Thomas R.G. Green

Computer Laboratory
Cambridge University
Alan.Blackwell@cl.cam.ac.uk

Computer Based Learning Unit
University of Leeds
Thomas.Green@ndirect.co.uk

Keywords: POP-III.C. All Cognitive Dimensions; POP-V.B. Questionnaire Analysis

Abstract

The Cognitive Dimensions of Notations framework (CDs) provides a powerful vocabulary for discussing the usability of programming languages, tools and environments. Although originally proposed as a discussion tool for designers, they have recently been used to design questionnaires intended for system users evaluating the usability of the programming tools they use (Kadoda et al., 1999) We extend their approach: rather than develop questionnaires tailored to specific systems, we propose a generalised questionnaire in which the definitions of the CDs themselves are offered to the users, and respondents can choose for themselves the features of the system that they wish to criticise. This questionnaire has been completed by an extremely diverse range of users, showing that a generalised CDs questionnaire indeed a suitable tool for user evaluation. Not surprisingly, some problems emerged as well.

Introduction

The Cognitive Dimensions of Notations framework (CDs) provides a powerful vocabulary for discussing the usability of information artefacts, especially programming languages, tools and. It was originally proposed as a broad-brush discussion tool, offering a vocabulary with which to discuss the usability tradeoffs that occur when designing programming environments (Green 1989, Green & Petre 1996, Green & Blackwell 1998).

As originally conceived it was expected that the evaluation would be performed by somebody who understood the framework well and also understood the system well. That somebody might be the designer, although it has to be admitted that designers usually have priorities in design rather than evaluation; or that somebody might be an HCI expert, which is how most HCI evaluation schemes work. Green (1989) expressed pious hopes of enriching the conceptual vocabulary of choosers and users by getting them to use the concepts of viscosity etc., but the problem is how to achieve that goal. Just telling designers and HCI folk about CDs is hardly likely to achieve it.

Last year at PPIG Kadoda et al. (1999) took a step toward that goal by introducing a questionnaire approach in which the evaluation was performed by system users rather than by designers or HCI specialists. That has to be a good idea. Not only does it give users a new set of ideas in such a way that they will be motivated to grasp them and use them, but it also means the users do all the work. The questionnaire approach seems to have been in the air last year – Collins and Fung (2000) describe a plan for a similar study, though details have not yet emerged.

The present paper aims to take another step along that road.

Kadoda et al.'s questionnaire only presented those CDs that they thought were relevant to the system under consideration, and to make it easier for the users to pick up the ideas, they paraphrased those CDs in terms of the system under consideration. There may be a problem here: filtering the CDs constrains the respondents to commenting on the CDs that the researcher or designer has already identified, and thereby mixes the HCI expert's evaluation (in choosing the subset) with the users' evaluations, so the data will not be a pure account of what the users thought. This is particularly dangerous when the questionnaire designer is also the designer of the system, and will quite possibly completely overlook aspects that are very important to the users.

And of course, if a different questionnaire has to be designed for each system, then the evaluators still have to do some of the work, rather than leaving it all to the users.

So we have set out to develop a questionnaire which presents all the CDs, and lets the users decide which ones are relevant. We have also attempted to present the CDs in general terms, applicable to all information artefacts, rather than presenting descriptions specialised to the system under consideration. On the plus side, this means (i) the users do all the work, (ii) the data only reflects their opinions, and (iii) the same questionnaire can be used for all information artefacts. On the down side, it means that the resulting questionnaire is longer (since it has to present all the CDs, not just a subset) and possibly harder to understand, since the CDs are presented in general terms.

The CDs framework is not, unfortunately, a stationary target. Green and Blackwell (1998) go more deeply into the notion of a 'sub-device' than previous publications have done, and we have taken account of that in the questionnaire.

In this paper, then, we describe our questionnaire. We have some preliminary results, which will help to establish whether we are making progress towards two important requirements of the approach we have taken.

Clarity: The questionnaire must explain CDs very clearly. (It has to be admitted that the academic presentations existing are not very clear, partly because the second author is too lazy, dim-witted and muddle-headed to sort out what he's trying to say). Our aim in the questionnaire has therefore been to describe CDs in very simple terms, such that they can be understood by an intelligent user who has a thoughtful attitude toward his or her work and tools.

Generality. The questionnaire will not make any assumptions about the system under study. It has not been designed with any specific system in mind, and aims solely to communicate CDs clearly. We have tested this objective in a pilot study involving an extremely wide range of users of information artefacts, using many different systems for different applications. Furthermore, the questionnaire will not make any assumptions about the type of activity that the user is engaged in. It will identify the nature of the user's work in a way that allows sensible evaluation of the responses by system evaluators.

The User Oriented Questionnaire

Our experimental questionnaire is divided into five main sections, which will be described separately below. The questionnaire form is entitled "Thinking about Notational Systems", and there is a brief preface describing the philosophy of cognitive dimensions analysis from the point of view of users:

> "This questionnaire collects your views about how easy it is to use some kind of notational system. Our definition of "notational systems" includes many different ways of storing and using information – books, different ways of using pencil and paper, libraries or filing systems, software programs, computers, and smaller electronic devices. The questionnaire includes a series of questions that encourage you to think about the ways you need to use one particular notational system, and whether it helps you to do the things you need".

The next four sections of the questionnaire ask about the user's level of experience with the system, define technical terms that are used in the rest of the questionnaire, ask the user to consider the activity for which they use the system and the different components of the system, then present each of the cognitive dimensions, encouraging the user to make comments about usability aspects of the system that are related to that dimension. Finally, there is an optional section that the user can fill in

to comment on other notations within the system. Each of these sections is described in more detail below. The titles in this paper are the same titles used in each section of the questionnaire.

Section 1 - Background information (half page)

The aim of this section is to establish whether the respondent is an expert user of the system under evaluation, and whether he or she has experience of other similar systems. The comments in the rest of the questionnaire can be given weight in accordance with the amount of experience, and the critical expertise that results from using a number of similar systems. This section asks the following questions:

- "What is the name of the system?"
- "How long have you been using it?"
- "Do you consider yourself proficient in its use?"
- "Have you used other similar systems? (If so, please name them)"

Section 2 – Definitions (1 page)

This section aims to describe the theoretical assumptions of the CDs framework in general and simple terms. It is based on the tutorial presentation of CDs by Green and Blackwell (1998).

"You might need to think carefully to answer the questions in the next sections, so we have provided some definitions and an example to get you started:"

"**Product**: The product is the ultimate reason why you are using the notational system – what things happen as an end result, or what things will be produced as a result of using the notational system. This event or object is called the product. Any product that needs a notation to describe it usually has some complex structure."

"**Notation**: The notation is how you communicate with the system – you provide information in some special format to describe the end result that you want, and the notation provides information that you can read. Notations have a structure that corresponds in some way to the structure of the product they describe. They also have parts (components, aspects etc.) that correspond in some way to parts of the product."

"Notations can include text, pictures, diagrams, tables, special symbols or various combinations of these. Some systems include multiple notations. These might be quite similar to each other – for example when using a typewriter, the text that it produces is just letters and characters, while the notation on the keys that you press tells you exactly how to get the result you want. In other cases, a system might include some notations that are hard for humans to produce or to read. For example when you use a telephone the notation on the buttons is a simple arrangement of digits, but the noises you hear aren't so easy to interpret (different dialling tones for each number, clicks, and ringing tones). A telephone with a display therefore provides a further notation that is easier for the human user to understand."

"Sub-devices: Complex systems can include several specialised notations to help with a specific part of the job. Some of these might not normally be considered to be part of the system, for example when you stick a Post-It note on your computer screen to remind you what to write in a word processor document."

"There are two kinds of these sub-devices.

- The Post-It note is an example of a **helper device**. Another example is when you make notes of telephone numbers on the back of an envelope: the complete system is the telephone plus the paper notes – if you didn't have some kind of helper device like the envelope, the telephone would be much less useful.
- A **redefinition device** changes the main notation in some way – such as defining a keyboard shortcut, a quick-dial code on a telephone, or a macro function. The redefinition device allows you to define these shortcuts, redefine them, delete them and so on."

"Note that both helper devices and redefinition devices need their own notations that are separate from the main notation of the system. We therefore ask you to consider them separately in the rest of this questionnaire."

"To review how we intend to use these terms, consider the example of a word processor. The product of using the word processor is the printed letter on paper. The notation is the way that the letter looks on the screen – on modern word processors it looks pretty similar to what gets printed out, but this wasn't always the case. If you want to find and replace a particular word throughout a document, you can call up a helper device, the search and replace function, usually with its own window. This window has its own special notation – the way that you have to write the text to be found and replaced, as well as buttons that you can click on to find whole words, or to find the word in upper and lower case etc."

Section 3 – Parts of your system (1 page)

This section first reviews the previous material, asking the respondent to instantiate the technical definitions in the context of the system they are evaluating:

- "What task or activity do you use the system for?"
- "What is the product of using the system?"
- "What is the main notation of the system?"

The respondent is then asked to characterise the type of activity that they engage in when using this notation. In most programming-like tasks, there will not be a single activity type, but rather a spread of different activities that varies for different users (analysts, testers, coders etc.). We construct this profile in terms of estimated percentage of time allocated to the different activities presented earlier in (Blackwell & Green 1999).

"When using the system, what proportion of your time (as a rough percentage) do you spend:
- Searching for information within the notation […]%
- Translating substantial amounts of information from some other source into the system […]%
- Adding small bits of information to a description that you have previously created […]%
- Reorganising and restructuring descriptions that you have previously created […]%
- Playing around with new ideas in the notation, without being sure what will result […]%"

Finally in this section we ask the respondent to identify other notations within the system, if they have already noticed them. There are further opportunities to identify other notational sub-devices later in the questionnaire.

"Are there any helper devices? Please list them here, and fill out a separate copy of section 5 for each one."

"Are there any redefinition devices? Please list them here, and fill out a separate copy of section 5 for each one."

Section 4 – Questions about the main notation (3 pages)

This section of the questionnaire simply summarises the cognitive dimensions, presenting them in terms of the notational definitions given earlier, and encouraging the respondent to identify features of the system that are relevant to each dimension in terms of their work. Each description in this paper is prefaced with the name of the CD that it describes, but that is for the benefit of academic readers familiar with CDs – the questionnaire itself simply groups each set of questions into a single box so that the respondent sees they are related. The questionnaire does not include the familiar name of the CD, but only an identification code for convenience in later analysis.

We believe that these descriptions, although simple, are true to the intention of previous publications on CDs. In fact, both of us have used these descriptions, rather than quoting earlier publications, when teaching undergraduate and postgraduate university courses on CDs within the last year. We hope that these descriptions may also provide a useful quick reference for other researchers, although of course they do not adequately describe the full implications or application of any of the CDs – a better reference for that purpose is (Green & Blackwell 1998).

Visibility and Juxtaposability
- How easy is it to see or find the various parts of the notation while it is being created or changed? Why?
- What kind of things are more difficult to see or find?
- If you need to compare or combine different parts, can you see them at the same time? If not, why not?

Viscosity
- When you need to make changes to previous work, how easy is it to make the change? Why?
- Are there particular changes that are more difficult or especially difficult to make? Which ones?

Diffuseness
- Does the notation a) let you say what you want reasonably briefly, or b) is it long-winded? Why?
- What sorts of things take more space to describe?

Hard Mental Operations
- What kind of things require the most mental effort with this notation?
- Do some things seem especially complex or difficult to work out in your head (e.g. when combining several things)? What are they?

Error Proneness
- Do some kinds of mistake seem particularly common or easy to make? Which ones?
- Do you often find yourself making small slips that irritate you or make you feel stupid? What are some examples?

Closeness of Mapping
- How closely related is the notation to the result that you are describing? Why? (Note that in a sub-device, the result may be part of another notation, rather than the end product).
- Which parts seem to be a particularly strange way of doing or describing something?

Role Expressiveness
- When reading the notation, is it easy to tell what each part is for in the overall scheme? Why?

- Are there some parts that are particularly difficult to interpret? Which ones?
- Are there parts that you really don't know what they mean, but you put them in just because it's always been that way? What are they?

Hidden Dependencies

- If the structure of the product means some parts are closely related to other parts, and changes to one may affect the other, are those dependencies visible? What kind of dependencies are hidden?
- In what ways can it get worse when you are creating a particularly large description?
- Do these dependencies stay the same, or are there some actions that cause them to get frozen? If so, what are they?

Progressive Evaluation

- How easy is it to stop in the middle of creating some notation, and check your work so far? Can you do this any time you like? If not, why not?
- Can you find out how much progress you have made, or check what stage in your work you are up to? If not, why not?
- Can you try out partially-completed versions of the product? If not, why not?

Provisionality

- Is it possible to sketch things out when you are playing around with ideas, or when you aren't sure which way to proceed? What features of the notation help you to do this?
- What sort of things can you do when you don't want to be too precise about the exact result you are trying to get?

Premature Commitment

- When you are working with the notation, can you go about the job in any order you like, or does the system force you to think ahead and make certain decisions first?
- If so, what decisions do you need to make in advance? What sort of problems can this cause in your work?

Consistency

- Where there are different parts of the notation that mean similar things, is the similarity clear from the way they appear? Please give examples.
- Are there places where some things ought to be similar, but the notation makes them different? What are they?

Secondary Notation

- Is it possible to make notes to yourself, or express information that is not really recognised as part of the notation?
- If it was printed on a piece of paper that you could annotate or scribble on, what would you write or draw?
- Do you ever add extra marks (or colours or format choices) to clarify, emphasise or repeat what is there already? [If yes: does this constitute a helper device? If so, please fill in one of the section 5 sheets describing it]

Abstraction Management

- Does the system give you any way of defining new facilities or terms within the notation, so that you can extend it to describe new things or to express your ideas more clearly or succinctly? What are they?
- Does the system insist that you start by defining new terms before you can do anything else? What sort of things?
- If you wrote here, you have a redefinition device: please fill in one of the section 5 sheets describing it.

Finally we ask two further questions which may evoke usability problems that are not addressed by any of the dimensions, or that may allow the respondent to express some problem that other dimensions have reminded them of:

- Do you find yourself using this notation in ways that are unusual, or ways that the designer might not have intended? If so, what are some examples?
- After completing this questionnaire, can you think of obvious ways that the design of the system could be improved? What are they? Could it be improved specifically for your own requirements?

Section 5 – Questions about sub-devices (1 page)

This final section of the questionnaire is printed on a single page, and respondents are given several blank copies of this page, so that they can complete as many as they wish. The contents of the page repeat the material presented earlier in the questionnaire, but in further abbrieviated form. This further abbreviation is not essential – the first pilot version of the questionnaire simply duplicated the whole of section 4 several times. However, this meant that respondents who identified a number of sub-devices quickly got bored with the repetitive nature of the task. In the current version of the questionnaire we therefore ask respondents only to comment on particularly salient features of this sub-device, relying on the brief descriptions to remind them of the original CD definitions.

"Please fill out a copy of this page for each sub-device in the system."

"This page is describing (tick one box): a helper device […], or a redefinition device […]. What is its name? What kind of notation is used in this sub-device?"

"When using this sub-device, what proportion of the time using it (as a rough percentage) do you spend: Searching for information […]% Translating substantial amounts of information from some other source into the system […]% Adding small bits of information to a description that you have previously created […]% Reorganising and restructuring descriptions that you have previously created […]% Playing around with new ideas in the notation, without being sure what will result […]%"

"In what ways is the notation in this sub-device different from the main notation? Please tick boxes where there are differences, and write a few words explaining the difference."

- Is it easy to see different parts?
- Is it easy to make changes?
- Is the notation succinct or long-winded?
- Do some things require hard mental effort?
- Is it easy to make errors or slips?
- Is the notation closely related to the result?
- Is it easy to tell what each part is for?
- Are dependencies visible?
- Is it easy to stop and check your work so far?
- Is it possible to sketch things out?
- Can you work in any order you like?
- Are any similarities between different parts clear?
- Can you make informal notes to yourself?
- Can you define new terms or features?
- Do you use this notation in unusual ways?
- How could the design of the system be improved?

Questionnaire Pilot Study

We have conducted a pilot study using this questionnaire with a very wide range of respondents, intended to capture the wide variety of notational systems employed by computer users for tasks that are more or less like conventional programming. The respondents in our pilot study have also included a wide range of different programming experience, ranging from expert programmers conducting postdoctoral research in computer science to computer users with no programming experience at all. The 18 pilot respondents are described in Table 1.

Resp.	*Programming Experience*	*System Evaluated*
1	Expert – CompSci PostDoc	*LaTeX*
2, 3	Expert – CompSci PhD Students	*OpenInventor*
4	Expert – PhD Student	*C++/Emacs*
5	Expert – PhD Student	*Cadence Verilog*
6	Expert – Professional Programmer	*Oracle* procedural SQL
7	Expert – University Technician	*PMS* music typesetting language
8	Expert – CompSci PostDoc	*Xisabelle* theorem prover
9	Expert – CompSci PhD Student	*Mathematica*
10	Expert – CompSci PostDoc	*PVS* theorem prover
11	Expert – PhD Student	*Finale* music typesetting package
12, 13, 16, 18	None – Musician	*Calliope* music typesetting package
14, 15	Some music research programming	*Calliope* music typesetting package
17	Some music research programming	*Finale* music typesetting package

Table 1 – Pilot Questionnaire Respondents.

Method

None of the 18 subjects in the study had any degree of prior familiarity with cognitive dimensions. They were recruited through individual approaches to members of the first author's department, and to other academic contacts with an interest in usability. Several respondents were recruited from the Music Department of Glasgow University, whose interest in notation usability results from their use of the music typesetting package *Calliope*, developed by William Clocksin of the Cambridge Computer Laboratory.

Subjects 1 -- 10 completed the questionnaire without supervision. Subjects 11 -- 18 completed the questionnaire in an interview, with the first author as interviewer. The interview was prefaced with the observation that the term "notation" did not necessarily refer to music notation, but to the content of the computer screen, however that might differ from conventional music notation. In the interview, no further attempt was made to explain the cognitive dimensions beyond the content of the questionnaire itself. When respondents in the interview setting did not understand a question, the interviewer simply proceeded to the next question, in order to make the interview as close as possible to unsupervised completion of the questionnaire. Where the next question related to the same cognitive dimension, this often had the effect of clarifying the question and eliciting a response.

Results

The two main groups in the pilot study were those using music notations (who were interviewed) and those using various programming-related tools (who completed the questionnaire without supervision). In the case of the interview group, interviews were conducted to a half-hour schedule.

This was too short – most interviews lasted 35 minutes, and a one-hour interview would have been more comfortable. In the case of the unsupervised group, most respondent took several days (ranging to several weeks) to return the completed questionnaire.

The version of the questionnaire used with the two groups had some slight differences also. The version used by the programmers group did not include the "search" activity, and also did not include the final two questions in section 4, which asked respondents about further aspects of the system beyond CDs.

There was a clear difference in the activity profiles of these two groups, as shown in table 2. The musicians spent the majority of their time transcribing music from other sources, while the programmers spend more time exploring possible solutions.

Activity	*Programmers*	*Musicians*
Search	(--)	8.5 (7.7) %
Transcribe	34.9 (24.8) %	51.5 (15.7) %
Add	24.7 (12.3) %	19.1 (9.1) %
Restructure	22.3 (15.6) %	19.4 (9.6) %
Explore	18.0 (19.6) %	2.4 (4.0) %

Table 2: Mean (and Standard Deviation) percentage of time spent on different activities. Note that the version of the questionnaire used by programmers did not incorporate the search activity.

Neither group appeared to have much trouble in interpreting the CDs, nor in identifying interesting usability problems in the systems they used. Table 3 summarises the number of explicit usability issues that were raised when considering each of the CDs. For the purpose of this analysis, an explicit usability issue is one where the respondent identified some specific aspect of the system that can be improved, rather than simply saying that the system is good or bad with respect to that CD (which also happened).

Dimensions	*Programmers*	*Musicians*
Visibility and Juxtaposability	13	14
Viscosity	5	12
Diffuseness	12	7
Hard Mental Operations	11	16
Error Proneness	13	11
Closeness of Mapping	8	5
Role Expressiveness	8	3
Hidden Dependencies	9	14
Progressive Evaluation	5	9
Provisionality	4	8
Premature Commitment	10	16
Consistency	0	4
Secondary	8	17

Notation		
Abstraction Management	3	7

Table 3: Specific usability issues raised by respondents with respect to each cognitive dimension.

Overall, the musicians raised an average of 18.1 usability issues when answering the 16 questions in section 4 of the questionnaire, while the programmers raised 9.9 usability issues when answering the 14 questions in an earlier version of the questionnaire (simply omitting the last two questions). The difference can be partially attributed to the interview format, which would have encouraged more complete responses.

Many respondents simply stated that the system was adequate with regard to a particular dimension, or that it was inadequate. In most of the rows of table 2 where no specific comment was made about usability features, the respondent did not leave the box blank, but answered “yes”, “easy” or “no”, “hard”. The nine programmers made 59 general positive comments and 16 general negative comments. There were also 2 points at which musicians indicated that they did not understand the question. By comparison, the programmers made 66 general positive comments, 18 general negative comments and 9 equivocal comments. There were 4 points at which programmers indicated that they did not understand the question. There does not appear to be any difference between the two groups on these measures – we suggest that the usability of a system is best measured not by these general expressions of satisfaction (or non-satisfaction), but by the number of specific criticisms that are made by questionnaire respondents.

Evaluation

Good questionnaires should be *valid* and *reliable*, meaning that they measure what they are supposed to and they give the same results every time. At the very least we would expect consistency in responses made by people evaluating the same system. Happily, that is what we observed. Of the six respondents describing the Calliope music typesetting package, for instance, five of them mentioned the same problem under “error-proneness” – they found it very easily accidentally to click on the wrong line of the music staff, and thereby enter the wrong pitch. Three of them noted under “viscosity” that it was difficult to change the layout of staves on the page once notes had been entered. There were many other similar patterns, not only from the Calliope users but from uses of Finale and Open Inventor. This consistency in response tends to support the validity of the questionnaire as an evaluation tool.

Nevertheless some problems emerged. This report has not included much detail on the analysis of sub-devices. In general, this is a problematic aspect of CD analysis. Respondents found it difficult to understand the concept of the sub-device, especially in the context of Calliope, which has very few (a couple of respondents mentioned the toolbar as an example of a sub-device). Programmers generally found this an easier concept to consider, listing 20 sub-devices between them. The sub-devices listed included text files, pencil and paper, user manuals and compiler output in addition to the expected browsers and sub-languages. In many cases it was not especially easy to apply cognitive dimensions to some of these.

It also became clear that for our sample of musicians, drawn from the academic world, the staff notation of music is a given. Music *is* staff notation, and vice versa. That opinion would not be shared by exponents of some traditions of Western music, such as choral singers using tonic sol-fa or the many composers in popular idioms who work by recording sequencer notations directly onto the computer, but evidently musicians in our sample had not strayed into those fields. Some respondents used slight variations on the standard notation (e.g. Gregorian chant notation, which has four lines rather than five, and diamond-shaped neumes rather than note-heads), but musicologists clearly found it more difficult than programmers to consider the posibility that the mapping between notation and product might be completely different. Of course they were also far more expert than any

programmer – an academic musicologist such as these respondents will typically have been reading music notation since childhood. Even if the programmer respondents had been programming since childhood, there is no programming environment that has presented the same notation for the last 40 years.

This is an important factor to consider when applying CDs outside the field of programming environments: where a single representation or notation dominates the thinking of users, they may not realise that other representations are conceivable and they might regard the notational system as outside the range of criticism.. Thus, although the responses received might indicate that musicians did not understand the CDs of role-expressiveness or consistency, as they made few comments on these topics, it may be the case that they understood the concept but did not have the perspective to identify those types of problems in their notation.

The larger number of usability issues raised by musicians, although partly due to the presence of an interviewer, may also reflect the fact that there is a clear distinction between the Calliope system, and other forms of music notation (such as sheet music). As far as musicians are concerned, Calliope is not music, any more than a word processor is the English language.

This is a distinction that is far more difficult for programmers to draw. Many comparisons of different programming language features are made without reference to the editor or development environment. The CDs framework emphasises that information artefacts include both a notation and an environment, and that usability is a function of the two. As a result, a number of programmers encountering this questionnaire (both respondents in this study and more casual readers) make comments that CDs do not describe the programming language but the editor that one uses. In the case of very similar textual languages (C and Pascal, for example), this may be true, and questionnaires of this type must therefore be applied with caution to ensure that the respondent understands the object of consideration.

Enforced Reflection

Answering the questionnaire does make people think – especially, it seems, system designers. The two users of logic theorem provers who responded in this pilot study are members of a large development group in Cambridge that develops theorem proving systems. That group has had a sceptical attitude toward usability, on the grounds that the only users of theorem provers are the developers, so they can improve usability as they wish, or on the alternative ground that theorem proving is meant to be hard, so there is no point trying to camouflage it with nice user interfaces. This pilot questionnaire inspired a special meeting of the theorem proving group to consider the issues that it raised, even though few members of the group ever got around to completing the questionnaire.

Reading the answers also made the authors think. The need to improve the account of sub-devices has been mentioned above. Clarifying our usage of ‘notation’ would also be useful; respondents find it difficult to regard actions such as mouse-clicks as a form of notation. Perhaps we should call actions something else, like an interaction language. And we realised that we have overlooked a generic type pf activity, that of exploring a structure. Future work on the framework will have to take these issues into account.

Conclusions

The pilot study met our objective – a generalised questionnaire that could be used for very different systems. Bboth programmers and non-programmers were able to criticise the systems they use in response to a description of the cognitive dimensions, and the descriptions were mostly acceptably clear.

This is very encouraging. Questionnaire-based evaluation could be quick and effective, and it may help to develop simplified descriptions of CDs for design education. But there are still problems. How, for instance, do we make users realise that the only notation they know is in fact entirely

arbitrary, rather than a fixed part of the natural order of things? We look forward to reading further questionnaire projects developing these issues.

Acknowledgements

We are grateful to the respondents who took part in the pilot study, especially William Clocksin and the Music Department of Glasgow University. Alan Blackwell's research is funded by the Engineering and Physical Sciences Research Council under EPSRC grant GR/M16924 "New paradigms for visual interaction".

References

Blackwell, A.F. & Green, T.R.G. (1999). Investment of attention as an analytic approach to Cognitive Dimensions. In T.R. G. Green, R. Abdullah & P. Brna (Eds.) *Collected Papers of the 11th Annual Workshop of the Psychology of Programming Interest Group* (PPIG-11), pp. 24-35.

Collins, T. & Fung, P. (1999). Evaluating Hank, a cognitive modelling language for psychologists. Paper presented at PPIG-11, available on-line from http://www.ppig.org/workshops/11th-programme.html.

Green, T.R.G. (1989). Cognitive dimensions of notations. In A. Sutcliffe & L. Macaulay (Eds.), *People and Computers V*. Cambridge University Press.

Green, T.R.G. & Blackwell, A. F. (1998). *Cognitive Dimensions of information artefacts: a tutorial*. Version 1.2, October 1998. (revision of "Cognitive Dimensions of notations and other Information Artefacts" at HCI'98) Available at http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/

Green, T.R.G. & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' approach. Journal of Visual Languages and Computing, 7,131-174.

Green, T.R.G., Petre, M. & Bellamy, R.K.E. (1991). Comprehensibility of visual and textual programs: A test of superlativism against the 'match-mismatch' conjecture. In J. Koenemann-Belliveau, T.G. Moher & S.P. Robertson (Eds.): *Empirical Studies of Programmers: Fourth Workshop* Norwood, NJ: Ablex, pp. 121-146.

Kadoda, G., Stone, R. & Diaper, D. (1999). Desirable features of educational theorem provers – a cognitive dimensions viewpoint. In T.R. G. Green, R. Abdullah & P. Brna (Eds.) *Collected Papers of the 11th Annual Workshop of the Psychology of Programming Interest Group* (PPIG-11), pp. 18-23.