A Cognitive Dimensions view of the differences between designers and users of theorem proving assistants

Gada Kadoda

Empirical Software Engineering Research Group
School of Design, Engineering and Computing
Bournemouth University
gkadoda@bournemouth.ac.uk

Keywords: POP-I.B. Pre-programming Knowledge, POP-II.B. Design, POP-IV.A. Theorem proving assistants, POP-IV.B. User interfaces.

Abstract

This paper discusses some observations on the differences between what the designers of theorem proving assistants (TPAs) think about the systems they designed and what the users of those TPAs actually find. A questionnaire based on the cognitive dimensions framework was sent to designers and users of a sample of TPAs. The aim of this work is to be able to identify specific areas that designers of TPAs need to devote extra attention to. It was observed that the cognitive dimensions of *closeness of mapping*, *visibility and juxtaposability* and *perceptual cues* are of particular significance.

1.Introduction

A mismatch between the views of the designers and the expectations of the users of computer systems has been cited as one factor that contributes to lower levels of system usability. This is not a recent development in the field of HCI but rather a long-standing and very much alive issue in system's design.

Gingras and McLean (1979) described a study of designers and users of a large information system in the operational area of a major Canadian company in which they investigated the issue of designer orientation. The authors used semantic differential techniques to profile the members of the IS design group and the corresponding user population. When these profiles were compared with each other, it was found that the designers' images of an actual user were significantly different from the users' self image. However, they reported that the users' profile of the designers did *not* differ significantly from the designers' self profile. The authors concluded that when designers claim to be user oriented, they are, but what they have in mind are people that look very much like themselves.

Research into cognitive approaches to users modelling has, without doubt, greatly contributed to an understanding of what occurs at the interface. Frameworks that originated from such research, such as, the GOMS model (Card, 1980), Backus Naur Form in 1981, Task-Action Grammar (Payne and Green, 1986), and others have been in use for some years. However, the mismatch problem between designers and users still persists. This paper does not aim to investigate this issue but rather to report on it from the theorem proving world.

The following section gives some background to the case study. Section 3 exhibits and analyses the results, and the final section puts the case study into perspective and draws some conclusions.

2.Preview of case study

The work reported in this paper is a continuation of a questionnaire-based study that utilised Green's Cognitive Dimensions' framework (Green, 1989), Green & Petre, 1996) to analyse the usability of theorem proving assistants (TPAs). The framework comprises a set of dimensions or generic attributes that are relevant to successful interaction in terms of influencing how the user think about the system and their tasks.

The dimensions were mapped into corresponding TPA's usability issues to devise a questionnaire. The questionnaire was sent to the designers and users of 27 theorem provers. A response rate of 63% was reported together with the results of the case study in (Kadoda, 1997a, b). The application of the cognitive dimensions framework to analyse the usability of TPAs was successful in producing a set of trade-off relationships between the dimensions that could be incorporated in the early stages of

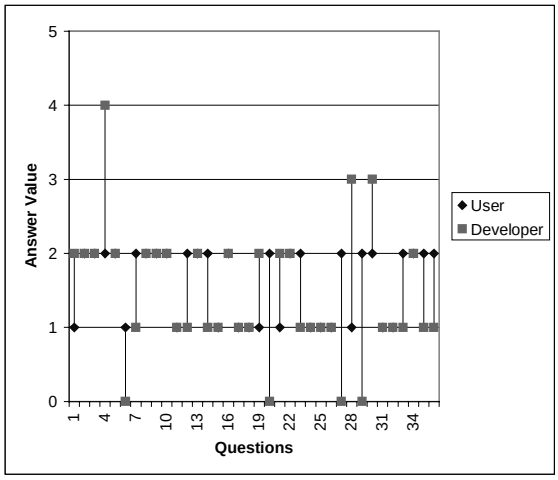
design. Ideally, those trade-off relationships may well be presented to the designers of these systems in the form of a checklist that serves as some sort of guidelines. This exercise of formulating a checklist was carried out using the results of the case study for educational theorem provers' designers (Kadoda, 1999).

It was observed that the majority, by far, of the respondents to the questionnaire were members of the development team of the TPA. A more focused study on the differences between designers and users seemed both possible and could shed light on additional usability issues. With this in mind, the case study reported in this paper was attempted. Using the same TPA sample, the questionnaire was sent to the users or designers (mostly users) depending on who responded to the first one (see Questionnaire in Appendix A).

The response rate was extremely poor with only 3 paired questionnaires for three TPAs. Nonetheless, it is possible to draw some observations on the differences in the responses of the user and designer of the same TPA.

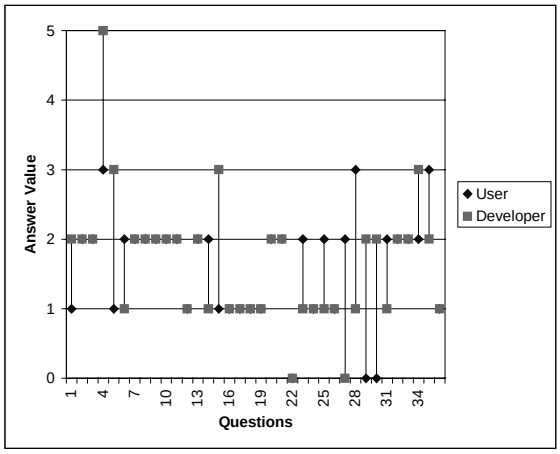
3.Observations and Analysis

The paired questionnaires correspond to CADiZ, the HOL system and ProofPower TPAs. The HOL System is an environment for interactive theorem proving in a higher-order logic. The CADiZ toolset supports the Z notation and produces Z specifications that are syntax, scope and type correct, and also supports reasoning about those specifications. ProofPower is a suite of tools supporting specification and proof in higher order logic and in the Z notation. The graphs below show the extent of disagreeing views between designers and users for the 3 TPAs. In upcoming sections those disagreeing views will be discussed.



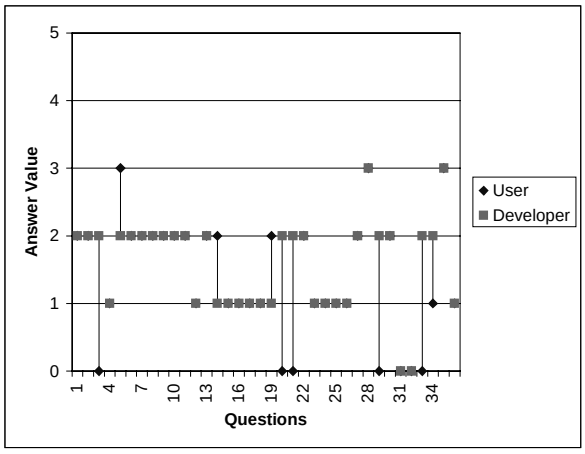
Graph (1) CADiZ

It can be observed that the views of designers of CADiZ and its users varied in 47% of the questions.



Graph (2) (The HOL System)

The designer and user of the HOL system disagreed in approximately 39% of the questions.



Graph (3) (ProofPower)

It can be observed that in 25% of the questions (least percentage among the 3 TPAs), views of the designer of ProofPower differed from the system's user. It is worth mentioning that ProofPower is a commercial system developed by ICL.

As a matter of fact, one expected less conflicting views for reasons relevant to the area of theorem proving. Firstly, the target audience for a theorem proving assistant is very specific, therefore the designers do not have to consider many user profiles. Secondly, it is essential for users of these systems to possess domain knowledge and also be computer scientists. That is, users of TPAs know both about theorem proving mechanisms and software development. This is almost not the case for many other computer systems, e.g., word processors. Nevertheless, the existence of differences in TPAs designers/users views *has* been observed. Percentages of the extent of designer/user disagreements included blank answers, but the following tables give a clearer view by name those differences for CADiZ, the HOL system and ProofPower theorem proving assistants.

Q	*User*	*Designer*
1.a	CADiZ is a single tool	CADiZ is an environment
1.d	It was built for educational pruposes	Built for commercial, educational and commercial purposes
3	CADiZ Keeps development record	No answer
4	Medium learnability	High learnability
7	Doesn't support backward proof	Supports backward proof
9	Doesn't allow user to sketch proof	Allows user to sketch proof
12.a	There is an indication about meaningfullness of error messages	No indication about meaningfullness of error messages
12.b	Medium meanfullness of error messages	No answer
13.b	There are no means of guiding user through proof script	There are means of guiding user through proof script
17	The proof is fairly obvious	No answer
18	The substructure of lemmas is clear	Not Applicable to CADiZ
20	Some effort is needed to perform a single change	Little effort is needed to perform a single change
22	There is a small difference between tactics in terms of effort in using them	Similar effort in using different tactics
24	It is fairly easy to understand the proof script, and to understand where a goal or subgoal starts and finishes	It is very easy to understand the proof script, and to understand where a goal or subgoal starts and finishes
25	It is not possible to know what each strategy or tactic performs before using it	It is possible to know what each strategy or tactic performs before using it

Table 1 (CADiZ)

The shaded questions are not particularly relevant to the cognitive dimensions. The differences in views can be summarised into two types. The first is features-related and the second is usability-attributes (or cognitive dimensions) related. The difference between the two categories being that the former could be attributed to the possibility that the user is inexperienced with the tool or not aware about the existence of some features, and the latter is about how the tool is presented to the user. For instance, the user of CADiZ thinking that the tool was built for educational purposes while the developer/designer has more scope for the tool hardly fits into designers/users conflicting views scenario. On the other hand, the user of CADiZ thinking that understanding the proof script the tool generates is not as easy as the designer thought it was is of consequence on the usability of the tool.

Q	*User*	*Designer*
1.a	HOL is a single tool	HOL is an environment
1.d	Experimental system	Educational and experimental
2	Has a batch input interface	Supports both batch input and GUI
3	Doesn't support development record keeping	Supports development record keeping
9	Doesn't allow user to sketch proof	Allows user to sketch proof
10.a	Handles all classes of rules in the same way	Doesn't handle all classes of rules in the same way
13.b	There are no means of guiding user through the proof script	There are means of guiding user Through the proof script
15	Doesn't offer feedback on progress of proof	Offers feedback on progress of proof
17	The proof is fairly obvious	No answer
18	No answer	The substructure of lemmas is clear
19	No answer	Sometimes the proof relies on tautologies
20	No answer	Some effort in needed to perform any change
21.a	Cannot visualise all the proof	Can visualise all the proof
23	Doesn't allow comments, indenting and colour to convey extra information	Not applicable to HOL
24	It is not too easy to understand the proof script	It is fairly easy to understand the proof script

Table 2 (The HOL System)

The HOL system is considered one of the veterans and most widely used among theorem proving assistants. There is more than one interface to HOL. The XHOL and the CHOL are X-Windows based interfaces to HOL, the Tk-HOL using Tk/Tcl toolkit, and it is also possible to use the Emacs editors as a simple interface to HOL. This may have affected the results in the sense that the respondent could be answering the questionnaire about an interface while thinking of another. It will be assumed that the questionnaire was answered for the Standard HOL interface.

Q	*User*	*Designer*
1.c	No answer	ProofPower is domain-independent
2	Interface is both Batch input and GUI	Interface is GUI
9	Doesn't allow user to sketch proof	Allows user to sketch proof
12.a	There is no indication about meaningfullness of error messages	There is an indication about meaningfullness of error messages
12.b	No answer	Medium meanfullness of error messages
12.c	Sometimes provides hints about errors	Doesn't provide hints
19	No answer	Sometimes proof depends on tautologies
22	No answer	Small difference in effort in using tactics
23	Does allow comments, indenting and colour to convey extra information	Does not allow comments, indenting and colour to convey extra information

Table 3 (ProofPower)

The discussion will focus on the shared (at least between 2 systems excluding blank answers) disagreements between the designers and users of the three TPAs. It was found that those shared disagreements include:

- TPA allows sketching of proofs *(Closeness of mapping)*

The extent of *'closeness of mapping'* to theorem proving offered by systems has been interpreted in the context of *'proof sketching'* (using the analogy of 'pencil and paper' proofs found in mathematics). To the users of the three systems the theorem-proving assistant in question does not allow sketching of proofs while the system's designers confirmed it did. There seems to be conflicting views on how the proof-sketching or the ability to construct a general description of the proof should be implemented.

- There are means within TPA to guide the user through the proof script *(visibility and juxtaposability)*

According to the designer of CADiZ and the HOL system, the system guides its users through the proof script, however, the user think otherwise. This usability issue was considered as part of the visibility and juxtaposability cognitive dimension. Here, how good and helpful the visual aids offered to users is a key question.

- The ease by which the proof script produced by TPA can be understood *(Perceptual cues)*

Perceptual cues dimension has been partly interpreted in the context of theorem proving assistants as how easily users can understand the proof script that has been developed. Users find that understanding the proof script generated by CADiZ and HOL is more difficult than the designers thought it was. It appears that the cues designers incorporated into their system's design to improve the clarity of the proof script are not good enough.

- TPA allowing use of colour, comments and indenting to convey extra information about the proof *(Secondary Notation)*

It is not possible to reflect on this last designer/user conflict because although they disagree, they are not on the same side for the two TPAs.

Thus far, the cognitive dimensions of *closeness of mapping, visibility and juxtaposability* and *perceptual cues* seem to have an effect on the differences between the views of designers and users of the same TPA. The inter-relationships between those dimensions were identified in the findings of the original case study (Kadoda 1997a, b). Further statistical tests established that these dimensions are positively correlated.

4.Conclusion

The difference between the vision of the designer and the reality of the user's performance has been identified as an influential factor in determining the usability of TPAs. Research aiming at bridging this gap suggests that a way forward is by understanding the nature of the proving activity and incorporating this knowledge when building the user interface (Merriam, 1996 & Aitken, 1995). The relevance of cognitive studies of programming to theorem proving activities has been recognised in Merriam (1995) and Aitken (1996). This has been of particular importance in instigating the use of the cognitive dimensions framework to analyse the usability of theorem proving assistants. Work reported in this paper aimed at further exploiting the framework to reveal where the mismatch between designers and users occurs in terms of the dimensions. The cognitive dimensions of *closeness of mapping, visibility and juxtaposability* and *perceptual cues* were observed to be of particular significance.

The objective of this work was to be able to identify specific areas that designers of TPAs need to devote extra attention to. Unfortunately, this was not achieved due to the poor response to the questionnaire that in turn reduced the results into observations and discussion headlines. Further work should endeavour to improve the sample size. However, this may be as good as it gets, that there are possibly more TPAs on shelves than on active service.

Acknowledgements

Thank you to the respondents to my questionnaires. Special thanks go to *CADiZ*, *HOL* and *ProofPower* teams.

References

Aitken, J.S. *et al.* (1995) Interactive Theorem Proving: An Empirical Study of User Activity. *Journal of Symbolic Computing*.

Aitken J.S. (1996) Problem Solving in Interactive Proof: A Knowledge-Modelling Approach, *Proceedings of the European Conference on Artificial Intelligence 1996 (ECAI96)* :335-339, Edited by Wahlster W., Budapest: 11-16, August.

Gingras, L., McLean, E.R. (1979) A study of users and designers of information systems. *IS working paper 2-79, Graduate School of Management, UCLA*.

Card, S.K. *et al.* (1980) Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology*, 12, 32-74.

Card, S.K. *et al*., (1983) The psychology of human-computer interaction. Hillsdale, New Jersey: Lawerance Erlbaum Associates Inc.

Green T.R.G. (1989) Cognitive Dimensions of Notations, *People and Computers V* : 443-460, Edited by Sutcliffe A. and Macaulay L.

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.

Kadoda, G.F. (1997a) Formal Software Development Tools (An investigation into usability). *PhD Thesis*. Loughborough University.

Kadoda, G.F. and Stone R.G (1997b) Cognitive Dimensions Analysis of Theorem Provers. *Proceedings of the 3rd International Workshop on User Interface Design for Theorem Proving Systems*: France.

Kadoda, G.F. *et al.* (1999) Desirable features of educational theorem provers - a cognitive dimensions viewpoint. *Proceedings of the 11th Psychology of Programming Interest Group Annual Workshop*. Leeds - UK.

Merriam N.A. (1996) Two Modelling Approaches Applied to User Interfaces to theorem Proving assistants. *Proceedings of the 2nd International Workshop on User Interface Design for Theorem Proving Systems*: University of York, UK.

Merriam N. A. *et al.* (1995) Assessing Theorem Proving Assistants: Concepts and Criteria. *Proceedings of the 1st International Workshop on User Interface Design for Theorem Proving Systems*: University of Glasgow, UK.

Payne, S.J. and Green T.R.G. (1986) Task-action grammars: A model of the mental representation of task languages. *Human-Computer Interaction*, (2), 93-133.