A Study of Usability of Z formalism Based on Cognitive Dimensions

Emma Triffitt and Babak Khazaei
School of Computing & Management Sciences
Sheffield-Hallam University
Harmer Building
Sheffield S1 1WB
B.Khazaei@shu.ac.uk
Tel:+44(0)114-225 3175
Fax:+44(0)114-2253161

Keywords: POP-I.B. teaching specification, POP-II.B. formal specification, POP-III.C. cognitive dimensions

Abstract

This paper discusses research undertaken to look at the usability of Z –a formal specification notation and TranZit –a computer editor for writing Z specifications.

The cognitive dimensions framework was the usability evaluation tool chosen for this research. This study focused on how particular cognitive dimensions were relevant to novice use and understanding of the Z notation when reading and writing specifications. This was investigated firstly by completing the 'Cognitive Dimensions questionnaire optimised for users' (Blackwell & Green, 2000) and subsequently by an empirical study developed for some chosen dimensions. The study backed up the choices made and brought up other issues of interest. This has led to ideas for future studies. Further studies may focus on other cognitive dimensions which were found to be important for novice use of Z; or involve trying this experimental approach on other exploratory design activities.

1. Introduction

In many software applications such as in the critical safety areas, it is important to have correct and bug free software. Formal specification is one approach to produce good quality, correct and error free software. The purpose of using a notation like Z is to produce an accurate specification from initial client requirements. The notation has a restricted syntax so it is precise but still abstract enough so as not to constrain how a developer will go on to design the application. However, formal specification has not received widespread support in the software development industry. One of the main reasons for this is that generally specification notations are complex and therefore hard to understand by non-experts. This makes them difficult to teach and learn. Also, it is argued, the resultant specification may be so complex that clients cannot interpret it to see if it agrees with their initial requirements (Wordsworth 1999).

Figure 1 illustrates an example of Z, which was produced in TranZit. TranZit is a computer editor which is part of a system for producing executable specifications from initial client requirements. The editor provided by TranZit is similar to any other Window's interface. It was developed at Sheffield Hallam University and is not yet in widespread use commercially. It displays the Z specification in schema boxes, allows free naming of all variables and has a syntax-checking tool. It is part of a larger system that can be used to animate an executable version of a specification. For more information on TranZit see Morrey et al (1998) and Department of Computing & Management Sciences, Sheffield Hallam University (2001).

The example represents a model (the state model) of 'Cookery' and 'PE' classes in a 'School'. This is the initial stage of producing a Z specification for this particular 'School'. The first two lines in the editor window declare two user-defined types; one is 'PUPILNO' (included in [ ] to define it as a type), and 'MARK' another type (defined to be the same as '$\mathbb{N}$' the set of natural numbers). The state model schema (represented as a box) is named 'School'. Inside, the box contains a declaration of the variables (using the mathematical notation of sets, functions, relations, sequences etc.); this is in the top section (signature or syntax box). In the example shown the variables are: 'Class' representing a set of 'PUPLINO' (Pupil numbers); 'Cookery' a partial function representing some 'Marks' for some of the pupils; and finally 'PE' a(nother) partial function representing some (other) 'Marks' for some of the pupils. The relationships between the variables -which must not change –(the invariant) –are in the bottom of the box (semantic box). Which in this example states that in this 'School' no pupil can take both 'Cookery' and 'PE' (line 1) and only pupils from the set 'Class' can have 'Marks' for these two subjects.

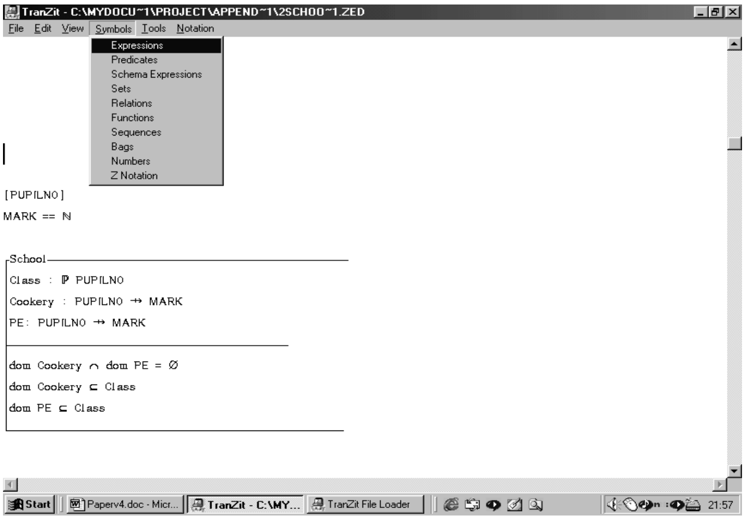

Figure 1. Z specification state model on TranZit

The next stage of producing a Z specification is to specify the operations –to model events which need to take place within the finished system. An example of an operation is to return the PE mark for a particular Pupil. Operations interact only with the variables declared in the state model and are also represented in schemas. An operation must preserve the invariant i.e. the relationships between the variables. The operation usually accepts inputs and returns output values. There is also a precondition (a predicate stating the relationship between the input and the state at the beginning of the operation) –the precondition must be true for the operation to occur. The post condition is a statement of the relationship between the input, output and the state before and after the operation. In Z specifications the model of the software is always described in terms of the state of variables- how they are influenced by operations.

The Cognitive dimensions framework (Green, 1989,1991) is a useful tool with which the usability of information artefacts may be evaluated. Cognitive dimensions have been successfully used for evaluating the usability of different programming tools and systems (two examples are Kadoda (2000), and Cox (2000). Cognitive dimensions were used in this research because unlike many other usability evaluation tools they are designed for non-specialists (of usability evaluation); they take into account the environment the system is being used in and the user type; and finally have been developed as a discussion tool which encourages looking at a system from the viewpoint of whether a preferred cognitive strategy for design activities can be adopted (Green 1989). This study intends to assess the usability of Z and its editor TranZit using the cognitive dimensions and to suggest possible improvements for TranZit.

2. Selecting Relevant Dimensions

We used the 'Cognitive dimensions questionnaire optimised for users' (Blackwell & Green 2000) as our starting point for selecting cognitive dimensions relevant to this system. This questionnaire introduces the various concepts –information structures, sub devices etc. and then asks a series of questions grouped into sections which map to each cognitive dimension. The questions are presented in very general terms so they can be used for any information structure. The results from completion of the questionnaire highlighted 8 of the 14 cognitive dimensions as being more relevant for novice use of the Z notation and TranZit. Britton and Kutar (2001) suggest that if only a subset of dimensions is used in the creation of a profile some important aspects may be overlooked. We are aware that we will need to address the full set of dimensions if we are to form a full investigative profile. However, we consider our study to be an initial study of cognitive dimension analysis and not a profile. We aim to move towards a profile in our future studies.

The 8 dimensions were separated into two groups and are described below:

Group 1. -Due to features of the actual notation (symbols)

Role Expressiveness -The purpose of a component (or action or symbol) is readily inferred.

Generally Z notation symbols do not easily relate to the concept they represent, especially not for novices. Although the fact that a lot of them are mathematical and Z is based on extended typed set theory will mean that mathematicians will recognise many of the symbols and what they represent.

Consistency -Are similar semantics expressed in similar syntactic forms?

Similar notation which means similar things:

‹ -is a member of	□ -is not a member of
§ -Domain Restrict	Œ –Domain Subtract
˜ -Range Restrict	¨ -Range Subtract
† -Partial Function	© -Total Function

Similarity between symbols which have similar conceptual meanings can help users of Z, but can also lead to confusion and mix up about which symbol is required for a certain purpose.

Error proneness - Does the notation invite mistakes?

One of the common errors when using Z is using the wrong symbol, when two or more are very similar. This is a consequence of consistency compounded by low motivation i.e. symbols which are not explicit in their meaning. Examples include: # representing cardinality (size) of a set and − representing inequality; also domain subtract/restrict and range subtract/restrict (see above in the Consistency part).

Group 2. -Due to the conceptual background of Z

Visibility & Juxtaposability -Can components be viewed easily and/or can any two components be placed side by side?

This requires a consideration of the environment in which the system is being used; in this case TranZit. When using TranZit the state model and operations schemas are produced one after another on the screen. Usually two schemas cannot be seen together on the same screen, if for example the last operation specified needed to be checked against the state model –this probably wouldn't be possible, the user would have to keep scrolling up and down or print one of them.

Viscosity -The level of work required to add, remove or to replace a plan level component of an information structure.

It can be quite difficult to make changes to previous work in Z. If it were the initial state model that required changing this would almost certainly impact on the operation schemas. It is quite probable that whilst working through the problem domain a novice would find that the original model didn't fit with a particular operation and have to go back to the state model and may even have to change other operations schemas as well. Z (formal specification in general) is mainly an exploratory design activity thus viscosity will be a problem in this context (Green & Blackwell 1998).

Premature Commitment - Constraints on the order of doing things which forces the user to make a decision before all information is available.

When constructing a Z specification firstly decisions have to be made about what variables are needed, the properties of the variables and how they are related to each other- this forms the predicate in the state model. Any schema operations developed after this must use only the variables in that model and operate within the relationships and constraints.

Hard Mental Operations Is there a high demand on cognitive resources?

There are many Hard Mental Operations needed when writing a Z specification. Examples include: having to break a complex problem down into relevant variables and operations, putting ideas down in a mathematical notation, then subsequently sticking to just those variables and deciding how various operations are going to affect those variables due to the already stated relationships. In general a novice may know what they want to happen but is not sure of how to express it in the notation.

Abstraction Management -Making the notation more like the user's conceptual structure, creating entities and grouping entities.

Abstraction is an inherent feature of Z and is important when using Z to produce a specification. However, abstraction may be a difficult concept for novices to grasp. Z allows free naming of schemas, variables and data types which helps relate the specification better to the problem domain. This abstraction means inventing and naming variables and data types is an essential first stage of producing a Z specification. There are usually several ways of expressing a problem domain and the state model and operations schemas are all abstract.

The cognitive dimensions described under the first group above (features of the notation) were the subject of an empirical study described in good detail in Britton and Jones (1999). That study focused on how those dimensions related to individual symbols of the notation. Taking the symbols out of the context of a specification some of the participants had to recognise what they stood for by studying a group of symbol names and picking the correct one. Another group were given a description and asked to pick from a group of several similar symbols.

We decided to focus on the cognitive dimensions from the second group (concepts of Z) for the design of the study. However, Abstraction Management was omitted from this study, as although it does cause problems for novices it is an essential aspect of Z (Z has much scope for abstraction). Of course it is important to consider abstract management as well as all the other dimensions in any future studies. Nevertheless, more thought is needed to investigate Abstract Management as an experimental factor and is the subject of another experimental study (Khazaei & Roast (2002) in press).

3. The Study

The Experimental Settings

The structure of this study was a questionnaire of open-ended questions, a method that has been used successfully in other experiments using cognitive dimensions (Blackwell & Green 2000, Kadoda 1999, Clarke 2001). The questionnaire consisted of 4 sections, one for each cognitive dimension: Premature Commitment, Hard Mental Operations, Viscosity, Visibility & Juxtaposability –in this order so that the cognitive dimensions could be logically evaluated.

The Subjects

Seven students from the MSc Computer Studies course participated in this study. The subjects were all familiar with both Z and TranZit through studying a term long formal specification unit. The timing of the experiment was three months after the completion of this unit. Each subject was individually asked to carry out the tasks using TranZit and answer the questions from the questionnaire.

The Task

Each subject was asked to specify a simple database system, to be used by a school, in Z (similar to the school example used in figure 1). The specification was given in English and the subjects were invited to formulate their own state model, using TranZit. They were subsequently given a prepared state model and asked to compare it to their attempt. Next they were asked to specify two operations: 1) to enrol a new pupil, and 2) to update some exam marks.

At different stages the subjects were asked questions highlighting the relevant cognitive dimensions. They had to fill in their answers in the questionnaire provided. Some of these questions and the subjects' responses are discussed below as the preliminary experimental results.

The Preliminary Results

Premature Commitment

The first task in the study was to produce a state model to specify some given requirements. Many of the subjects had initial difficulties recalling details of Z in order to formulate the state model. We feel the timing of the follow up experiment needs to be improved, however it is an interesting observation that in a relatively short period of time from the course the subjects had already forgotten many of the main principles of developing a Z specification.

An interesting answer for one of the questions in this stage, namely: "What was the first thing you did when writing the Z specification?" was 'I wrote it out on paper first'. This supports the result reported by Roast (1998) who conducted a survey of 23 TranZit users (TranZit tool was called wiZe in that paper- the tool was virtually the same). The users were asked to reflect on the best way in which to use the tool by asking them to select instructional statements for new TranZit users. The outcome of the survey showed that more experienced TranZit users favoured instructions which encouraged the choice of schema name and schema details prior to its creation in TranZit.

Writing a Z specification is a process of exploratory design, that is, the finished product is unknown and has to be 'discovered' during the design process (Green 2000). It is clear that in this kind of situation, when a design is first being conceived it is much easier to use a medium which is more flexible. Where the train of thought is not broken for example, by having to look in the menus to find the correct symbol. With pen and paper one can put ideas down anywhere on the page, annotate and circle important points, work through first thoughts, invent 'dummy data'. After primary ideas are developed then is maybe a better time to start using TranZit to make use of the structure of schemas and the Z notation that it supports. This is not a problem restricted to TranZit. In many applications, especially for design tasks, going straight to the computer does not facilitate the flow of ideas. TranZit has been designed to not inhibit creativity (there is no agreed method for constructing a Z specification), the creators carried out user studies whilst designing the application and realised that specification was an iterative process and the editor is therefore free-form (Morrey et al 1998).

Hard Mental Operations

The answers to the questions in this section showed the problems the subjects had encountered when trying to write the state model in this particular study. Most of them believed their models were confused and were not at all confident in what they had produced. Again, this is due to their poor recollection of Z details, as one stated 'I wanted to say the same things [as the example specification], but my notation was hideously wrong'. In Z when creating the state model the onus is on getting the Z correct when there are factors involved which will impact on future tasks.

The difficulties with symbols was also nicely demonstrated, one subject used ‡ -maplet instead of † -Partial Function this relates to error-proneness and consistency (2 of the cognitive dimensions). Many novices get symbols confused because of their similarity, both in structure and concept.

The occurrence of the error proneness cognitive dimension (sometimes due to the consistency of the notation) has increased the impact of hard mental operations. Cox observed a similar phenomenon in his study, which (amongst other things) looked at usability issues with the specifics of symbols in the

UML notation, and classed this as an interdependency (Cox 2000). Our study supports Cox's assertion on these interdependencies.

When asked about TranZit's aid in writing specifications: "How do you think using TranZit helps or hinders producing a model (rather than using pen and paper)?" there were answers for both sides. The negative aspects of TranZit included unhelpful error messages. For a novice who will undoubtedly make errors the messages produced by TranZit are difficult to understand. It is one thing knowing an error is there but not understanding why can be very frustrating. Admittedly one should use the syntax checker after typing each schema to stop errors being compounded. The risk of errors going unchecked is traded off against losing the flow when writing a specification. The creators of TranZit decided not to make syntax checking automatic '…this allows the specifier to concentrate on capturing the essence of the model of a problem, without initial concern for the complexities of the Z notation syntax' (Morrey et al 1998). This feature of TranZit is a plus point for the exploratory process of design, in that attention is not broken while primarily typing the specification by an automatic error checker.

Positive aspects of TranZit stated by the subjects included the syntax checking. (Although the error messages may not be clear to novices at least they pinpoint the exact position (column & line co-ordinates) of the error, which is of great benefit.) Other positives were: the ability to edit and the menu of symbols. The symbols in the menu all have their names next to them, which is a big help to novices who easily confuse similar symbols.

Viscosity

The answers to the questions in this section clearly showed that the subjects understood problems that occurred due to the 'viscous' nature of Z. Most of them (5 out of 7) stated that the modification that required changes to the state model would also require all the operations schemas to be modified as well and classed this as the more difficult adaptation. The other modification was just to specify a new operation which would not impact on any other schemas. One of the subjects stated 'Adding a new operations schema is infinitely easier than modifying existing schemas'. They realised that specifying a new operation would only require using the variables that were already declared and sticking to the invariant.

General problems the subjects found with modification seemed to be when existing schemas needed to be modified: 'altering everything you've done so far …especially with complicated schemas', 'updating other schemas', 'overlooking something when modifying'. These statements clearly indicate the viscosity of the Z notation –when even one minor change may require changes to other schemas.

Visibility & Juxtaposability

The questions in this section brought up some clear usability problems with TranZit. One of the main issues the subjects reported (6 out of 7) were the problems with finding the symbol they want from the menus –'I sometimes have to search through all the symbols for the one I'm looking for'. The creators of TranZit claim that '…dialog boxes are structured such that characters belonging to the same functional group of operators are located together' (Morrey et al. 1998). The subjects in this study did not feel that the symbols were grouped well one stated 'Sometimes the symbols seem to be grouped quite illogically'. The main problem with the symbols menu is that unless a user has a good grasp of the mathematical basis of Z the names of the groupings do not mean much to them and thus the contents of those groups are a mystery. Figure 1 shows the symbols menu groups.

The juxtaposability issue demonstrated in this system is the fact that two schemas –unless they are one after another, cannot be seen at the same time. For instance a user may wish to compare the state model with one of the operations schemas. All the subjects in this study spotted this problem and stated that a tiling facility or split screen –the ability to have more than one window open would be very beneficial.

4. Summary

The results of this study support our assertion that the cognitive dimensions chosen from the initial questionnaire by Blackwell and Green (2000) are extremely relevant when considering the usability of Z. Lack of consideration of these dimensions hinders the required exploratory design activities when using Z in TranZit and distracts focus of attention of novice users.

Overall this investigation has raised several important points for the usability of TranZit and the Z notation by novices. In summary the main results are: Some features of TranZit appear to make it difficult for novices to use it effectively; these features include the menu system, restriction of one window, the error messages, more advanced features which they have no experience of but cause distraction. Further practice and experience of using TranZit would reduce these problem areas, but for a novice needing some support TranZit can prove daunting. However, TranZit does have many features which support users of all abilities. One of the main challenges with any software for this type of user activity is to reach a balance, where some structure is needed but creativity is not inhibited. Cognitive dimensions enabled us to highlight some of the deficiencies associated with TranZit and described the problem in the framework of cognitive dimensions. Terms such as viscosity, visibility etc. provides a common vocabulary for user evaluation of information structures. Many of these identified deficiencies are easily fixable for future versions of TranZit.

The features of the Z notation that appear to cause most problems are the initial stages of developing a state model which requires careful choice of data types, variables and relationships between variables. This has to be done with a view to how the operations to be further specified will fit with those data types and variables. TranZit does not support the use of examples in running through the model whilst formulating the initial model. This seemed to be needed by subjects in our experiment. The facility to use an example is part of an animation tool that works with TranZit. However, this tool is used after a model has been built and is used to validate the model with the users. It appears that this tool or a similar one is needed during the model building as well. For the 'exploratory design activity' of developing a model, a far less rigid environment than TranZit seems to be needed. The logical decomposition of the problem domain and design of the specification are often not problematic for novices. The constraints of the notation and expressing ideas using the correct symbols and syntax are.

There was a clear demonstration of usability issues cropping up under the guise of more than one cognitive dimension e.g. the symbols menu of TranZit as both a hard mental operations problem and a visibility issue. The specification of the state model demonstrates hard mental operations as well as premature commitment. Again the framework of cognitive dimension was extremely helpful in describing these problems using meaningful and generic terms.

5. Further studies

The study was very useful in demonstrating the fruitfulness of this type of approach in evaluating an information structure.

Following on from carrying out the study here is a summary of recommendations to improve this study and conduct a full experiment in this area:

The most important factor in a full study would be to use subjects who are currently studying a Z module or have just finished one. The questionnaire could then be made relevant to their coursework and this would reduce any problems with recall.

Scenarios used for the questions could be more complex, to especially highlight the hard mental operations and viscosity issues in Z notation and TranZit.

The other cognitive dimensions identified as of relevance to the Z notation (Consistency, Role expressiveness, Error-proneness) could be incorporated into the questions in a more direct way than the one used in this study (i.e. not just through Hard Mental Operations or Viscosity questions).

The questionnaire could be tried out on other tasks and other exploratory design activities to see if in fact there are similar cognitive dimensions that have a direct effect on usability issues.

6. References

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

Britton, C., and Jones, S. (1999), "The Untrained Eye: How Languages for Software Specification Support Understanding in Untrained Users, Human Computer Interaction, 14, pp. 191-244, ISSN 0737-0024.

Britton, C., and Kutar, M. (2001),'Cognitive Dimensions Profiles: A Cautionary Tale'. In G Kadoda (Ed), Proceedings of the thirteenth Annual Meeting of the Psychology of Programming Interest Group, 265-274.

Clarke, S. (2001), ‘Evaluating a new programming language’. In G. Kadoda (Ed), Proceedings of the thirteenth Annual Meeting of the Psychology of Programming Interest Group, 275-289

Cox, K. (2000), ‘Cognitive Dimensions of Use Cases –feedback from a student questionnaire’. In A. F. Blackwell, E. Bilotta (Eds). Proceedings of the twelfth Annual Meeting of the Psychology of Programming Interest Group, 99-122.

Department of Computing and Management Sciences, Sheffield Hallam University, (2001), Requirements Engineering Research Group, [online], last accessed on 26 March 2002 at URL:

http://www.shu.ac.uk/schools/research/mitri/crc/req.html

Green, T. (1989), "Cognitive Dimensions of Notations", In A. Sutcliffe & L. Macaulay (Eds.), People and Computers V Proceedings of HCI'89, Cambridge University Press.

Green, T. (1991)."Describing information artefacts with Cognitive Dimensions and Structure maps. In D. Diaper, & N. Hammond (eds.), People and Computers VI, Proceedings of HCI'91. Cambridge University Press.

Green, T. R. G. (2000), ‘Instructions and Descriptions: some cognitive aspects of programming and similar activities’. In V. Di Gesu, S. Levialdi, L. Tarantino (Eds) Proceedings of Working Conference on Advanced Visual Interfaces (AVI 2000). New York: ACM Press, 21-28

Green, T. R. G., and Blackwell, A. F. (1998), "A Tutorial on Cognitive Dimensions", Available on-line at:

http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/index.html

Kadoda, G. (2000), ‘A Cognitive Dimensions view of the differences between designers and users of theorem proving assistants’. In A. F. Blackwell & E. Bilotta (Eds), Proceedings of the twelfth Annual Meeting of the Psychology of Programming Interest Group, 33-44

Khazaei, B., Roast, C. (2002), 'The Influence of Formal Representation on Solution Specification', in Press, Submitted to Requirement Engineering Journal.

Morrey, I., Siddiqi, J., Hibberd, R., Buckberry, G. (1998), ‘A toolset to support the construction and animation of formal specifications’. The Journal of Systems and Software, 41 (1998), 147-160

Roast, C.R. (1998), 'Modelling unwarranted commitment in information artefacts.' In S. Chatty and P. Dewan, editors, Engineering for Human-Computer Interaction, pages 77-90. Kulwer Academic Press, 1998.

Wordsworth, J. B. (1999), ‘Getting the best from formal methods’. Information and Software Technology, 41 (1999), pp. 1027-1032