The Usability of Formal Specification Representations

Babak Khazaei & Chris Roast
School of Computing and Management Sciences
Sheffield Hallam University, S1 1WB
{B.Khazaei,C.R.Roast}@shu.ac.uk

Keywords: POP-I.B. *training specification* POP-II.B. *Formal specification* POP-III.B. *Specification Language Z* POP-IVA. *simple vs. generic approach*

Abstract

Formal specification representations are powerful abstraction tools. Employing such tools requires an ability to effectively exploit the information that they express. We report on an experiment investigating the influence of employing a formal specification on the developer's perception of its solution approach.

Introduction

Within the Human Computer Interaction community there have been many findings focusing upon the usability of notations, Nielsen, J., and Mack, R. L., (1994). Work has mainly reported on the use of notations by non-technical end-users, and has not explored notations designed for technical end-users. The study reported here addresses the influence of a formal notation upon technical tasks in the context of software development. The focus of our study is the use of formal specification notations, such notations have been widely advocated within early phases of software development. Although human factors issues have been largely ignored in the development of such notations, we believe that such issues surround ease of use are just as relevant in the consideration of technical specification notations as they for computer interfaces. An unusable, or hard to use, specification notation is likely to impact upon effective software development. The use and usability of: technical notations, design notations and specification notations have been the subject of some studies, but few have explored formal specification (see Agarwal et.al. (2000), Brun P. and Beaudouin-Lafon M. (1995), Gray, W. D. and Boehm-Davis, D. A., (1966), Roast, C., Khazaei, B., and Siddiqi, J. (2000), Britton, C., and Jones, S., (1999) for exceptions).

The use of specification notations by software developers is considered as beneficial because they provide abstraction facilities that enable developers to attend to "what" needs to be achieved rather than "how" the requirements will be achieved. A good specification notation should provide strong expressive power that suits the modeling of problem domains and requirements. The Z notation is a suitable example of a powerful specification notation and is the object of our study. We are not concerned with the expressive power and adequacy of the abstraction facilities offered by Z, but with whether users employ them effectively. One factor, limiting the uptake of notations such as Z, appears to be the additional effort of working with mathematically precise notations early in development, Monk et al.(1994). There are many facets of notation use that could be focused upon in assessing specification notations such as Z. Britton, C., and Jones, S., (1999), argue on the importance of understandability of a specification representation for untrained users. Their focus is on identifying and listing properties that are good indicators of understandability of a representation and investigating how a specification notation such as Z scores on the properties. Here we have focused up a different facet of a notation: on the influence that the Z formalism has had on developers' perceptions of solutions for a given problem.

The Study

In two stages our subjects were asked to express preferences in employing one or another solution approach to tackle a given problem. In the first stage the subjects were provided with an informal description of the problem and two alternative solution approaches (see figures 1 & 2). In the second stage the same group of subjects were provided with the same problem described formally using Z and the same two solution approaches also expressed in Z.

Seventy-two final year software engineering degree students took part in stage 1. They were asked in a series of four questions to express their preference on choosing either solution approach A or B with regards to requirements: (a) alone, (b) alone, and (a) and (b) together. The fourth question was an opportunity for the subjects to justify their decisions. The solution approaches were expressed as two sets of tables for this stage.

The system is a browser for looking through a library of videos, where the currently selected video is displayed. Each video has a unique title and is given a unique horror rating. The higher the rating the more horrific the video.
Two operations are required:
(a) return the horror rating of a given video title,
(b) set currently shown video to the next more horrific video

Figure 1. Summary of the problem.

Solution Approach A: use a table lookup linking each video title to its horror rating
Solution Approach B: use a table lookup linking each horror rating to a video

Figure 2. Summary of the two solution approaches.

Stage 2 was carried out four weeks after stage1. During this interval the subjects were taught the use of Z notation and had practiced writing and reading Z specifications.

Fifty-one of the same subjects as in stage1 took part, they were asked the same four questions as in stage 1. Both the browser system and the solution approaches were expressed as Z schemas. The subject also had the informal description of the problem as in stage 1 (figure 1). The subjects were therefore aware that they were dealing with the same problem as stage 1 and the only difference was that they had been supplied with a formal description of the problem and its formal solution approaches. Figure 3 summarizes the results of subjects' preferences for for the first three questions. At stages 1 and 2 the preference for approaches for (a) and (b) show a good awareness of the merits of the two approaches. The results for the consideration of (a) *and* (b) show a marked shift in preferences between the two stages.

Question focus	Approach	Stage 1	Stage 2
(a)	A	70 (97%)	45 (88%)
(a)	B	2 (3%)	6 (12%)
(b)	A	9 (13%)	5 (10%)
(b)	B	63 (87%)	46 (90%)
(a) and (b)	A	44 (61%)	10 (20%)
(a) and (b)	B	28 (39%)	41 (80%)

Figure 3: Summary of results for the two stages.

Discussion

The results from the experiment clearly indicate that the introduction of Z as a method for specifying the problem has influenced the choice of solution. In particular, it should be noted that this influence is upon the perception of the problem - subjects were not required to formulate their preferred solution.

Formally the two approaches are adequate as solutions to the problem, however approach A involves more technical details but is more generic, where as approach B is overall simpler. The reasons given for the particular preferences were analysed in order to identify problem characteristics that may be responsible for the results.

Preferences for approach A in stage 1 were based upon it providing a more logical and generic solution. In contrast those who preferred approach B justified this on the ground of simplicity. The majority of the subjects in stage 1 therefore have opted for a more generic solution. In stage 2 where the Z notation was used, the majority has switched towards solution approach B. This indicated a more careful consideration of the two requirements at hand and less attention to providing a generic solution. Subjects in stage 2 commented that the solution approach B accommodates the two requirements better and makes the task simpler.

There is an apparent interplay between two considerations for the subjects: a simplistic approach, and a generic approach - we label these “make it simple” and “make it generic” respectively. At the informal stage 1, "make it generic" was the preferred approach. However, introducing the formal notation into the problem seems to have encouraged subjects to focus upon solution approaches which "make it simple".

Conclusion

A side effect from the influence of enforcing formality and use of Z notation is that our subjects slipped from “make it generic” to “make it simple” considerations. Employing Z with its powerful abstraction tools in theory is to encourage taking an architectural view and bring about the “make it generic” consideration to the forefront. In the face of precision demanded by Z, it appears that the overall problem has been approached in a more naïve manner.

As our subjects were new to formal methods and using Z, we conclude that the development of generic solutions within formal specification is dependent upon skills that are rarely found in textbooks. In summary: the provision of high level abstractions does not necessarily mean that they will be used; and the precision offered by the notation can also mitigate against the effective use of abstractions.

Future work is to be aimed at examining the enhanced use of formal notations through the examining how abstraction is introduced to subjects, and how alternative notations support effective access to, and use of, abstraction. We are also looking into the role of that "Cognitive Dimensions" and "Investment of Attention Approach" Blackwell, A.F., and Green, T.R.G. (1999), Green T.R.G & Blackwell,A.F.,(1998), can play in explaining the phenomenon obsereved in this study. In particular, we would like to look at the relationship between "make it generic" versus "make it simple" and the cognitive dimensions.

References

Agarwal R. De, R. Sinha, A. P and Tanniru, M.,. (2000) On the usability of OO representations. *Communications of the ACM*, 43(10), pp 83-89.

Brun P.,and Beaudouin-Lafon M., (1995) A taxonomy and evaluation of formalisms for the specification of interactive systems formalism in HCI. In *People and Computers X*, Editor M. A. R. Kirby and A. J. Dix and J. E. Finlay, Cambridge Univ. Press, pp 197-212.

Gray, W. D. and Boehm-Davis, D. A., (Eds.), (1996) *Empirical Studies of Programmers 6th Workshop*, pp 147-161, Ablex Publishing Corp..

Monk A. F., Curry M. B., and Wright P.C. (1994) Why industry doesn't use the wonderful notations we researchers have given them to reason about their design In *User-Centred Requirements For Software Engineering*, Editor D. J. Gilmore, R. L. Winder and F. Ditienne, pages185-188, Springer-Verlag.

Nielsen, J., and Mack, R. L., (1994) Usability Inspection Methods, Wiley, Roast, C., Khazaei, B., and Siddiqi, J. (2000) Formal comparison of program modification. *IEEE Symposium on Visual Languages*, pp 165-171. IEEE Computer Society.

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.

Blackwell, A.F., and Green, T.R.G. (1999) Investment of attention as an analytic approach to cognitive dimensions. *11th Annual Workshop of the Psychology of Programming Interest Group*, pp 24-35.

Green T.R.G and Blackwell,A.F.,(1998). Design for usability using Cognitive Dimensions. Tutorial session at British Computer Society conference on *Human Computer Interaction HCI'98*.