Preliminary results of a study of the completeness and clarity of structured abstracts

David Budgen¹, Barbara Kitchenham², Stuart Charters¹, Mark Turner²,
Pearl Brereton², Stephen Linkman²

¹Department of Computer Science, Durham University, Durham, Science Laboratories, South Road, Durham City, DH1 3LE, UK
david.budgen@durham.ac.uk, charters@lincoln.ac.nz

²School of Computing and Mathematics, Keele University, Keele, Staffordshire ST5 5BG, UK
{barbara,m.turner,o.p.brereton,steve}@cs.keele.ac.uk

ABSTRACT

CONTEXT: Systematic literature reviews largely rely upon using the titles and abstracts of primary studies as the basis for determining their relevance. However, our experience indicates that the abstracts for software engineering papers are frequently of such poor quality they cannot be used to determine the relevance of papers. Both medicine and psychology recommend the use of structured abstracts to improve the quality of abstracts.

AIM: This study investigates whether structured abstracts are more complete and easier to understand than non-structured abstracts for software engineering papers that describe experiments.

METHOD: We constructed structured abstracts for a random selection of 25 papers describing software engineering experiments. The original abstract was assessed for clarity (assessed subjectively on a scale of 1 to 10) and completeness (measured with a questionnaire of 18 items) by the researcher who constructed the structured version. The structured abstract was reviewed for clarity and completeness by another member of the research team. We used a paired 't' test to compare the word length, clarity and completeness of the original and structured abstracts.

RESULTS: The structured abstracts were significantly longer than the original abstracts (size difference = 106.4 words with 95% confidence interval 78.1 to 134.7). However, the structured abstracts had a higher clarity score (clarity difference = 1.47 with 95% confidence interval 0.47 to 2.41) and were more complete (completeness difference = 3.39 with 95% confidence intervals 4.76 to 7.56).

CONCLUSIONS: The results of this study are consistent with previous research on structured abstracts. However, in this study, the subjective estimates of completeness and clarity were made by the research team. Future work will solicit assessments of the structured and original abstracts from independent sources (students and researchers).

Keywords: *Evidence-based software engineering; systematic literature review; structured abstracts.*

1. INTRODUCTION

A key requirement for Evidence-Based Software Engineering is the ability to be able to find, evaluate and aggregate all of the appropriate sources of evidence. In particular, the evidence-based paradigm is one that relies heavily upon the use of systematic literature reviews to assemble the (empirical) evidence that is needed to address a research question (Kitchenham, 2004, Webster and Watson, 2002). A secondary study such as a systematic literature review requires exhaustive searches of the literature in order to identify potentially relevant primary studies. Such searches involve two stages: firstly researchers need to perform a wide search to identify as many candidate primary studies as possible; secondly they must undertake a more detailed review of these candidates against specific inclusion and

exclusion criteria. Indeed, the first step of the search process is very likely to identify a large number of studies, of which many will actually be irrelevant.

Current procedures, based on experience from clinical medicine, suggest that a review of the title and abstract of a primary study should be sufficient to enable the researcher to determine whether or not it is relevant to the study being undertaken (Kitchenham, 2004). However, recent attempts to conduct systematic literature reviews in the domain of software engineering have reported difficulties with identifying whether or not primary studies are relevant to a topic of interest. This is because the information provided in abstracts is often incomplete, with the effect that the researchers find it necessary to read other parts of the paper to determine whether or not it is of interest (Brereton et al., 2007).

One approach to improving the standard of abstracts that has been adopted in medicine and in other domains such as psychology is to use structured abstracts (Hartley 2004). The result of empirical studies conducted in Educational Psychology suggests that structured abstracts are a potentially valuable approach to improving the readability and value of abstracts (Hartley, 2003). In addition, Bayley and Eldredge (2003) identify other benefits of adopting this form of abstract to help improve the design of empirical studies.

We are currently undertaking a research program to assess whether structured abstracts would be of benefit for empirical software engineering articles. This research program has two purposes. The primary purpose is that of identifying whether having access to information in the form of structured abstracts can improve the task of performing a systematic literature review. Should the outcomes of our program be positive, then the secondary purpose will be to provide evidence that can be used to help persuade journal editors and conference proceedings editors to adopt the practice of requiring authors to provide structured abstracts. We have completed the first part of our program which was an observational study (Kitchenham et al., 2006a). This confirmed that structured abstracts are longer than unstructured abstracts but score better than unstructured abstracts with respect to readability indexes. The results reported in this paper arise from the initial part of a formal experiment intending to assess the clarity and completeness of structured and unstructured abstracts.

In Section 2 we describe the experimental design of the formal experiment and indicate how the results reported here were obtained. In Section 3 we present our analysis based on clarity and completeness assessments of structured and unstructured abstracts made by the project team. In Section 4 we discuss our results and future work.

2. EXPERIMENTAL MATERIALS AND METHODS

Prior to starting our experiment we prepared an experimental protocol outlining the rationale, design and procedures for the experiment (Budgen et al., 2006). The basic design of the formal experiment is a within-subject design with two treatments (a structured abstract and an unstructured abstract). Participants will be asked to evaluate two abstracts (one structured and one unstructured). The abstracts will be obtained from different software engineering articles describing experiments. The order in which participants see the abstracts will be randomized. In addition, we will have sufficient participants to ensure each abstract in each format is reviewed first by one participant and second by another. In order to achieve this design we require four participants for each software engineering article.

In order to prepare for this experiment, the research team needed to:

1. Obtain a selection of appropriate software engineering articles.
2. Identify a means of scoring the abstracts for completeness and clarity.
3. Prepare structured versions of the abstracts in the selected articles.
4. Select an appropriate number of participants.
5. Prepare a tool to present an HTML style version of the abstracts to the participants.

This study presents results obtained from the first three steps in the experimental process.

2.1 Selection of software engineering articles

A key question is the number of abstracts to use. For convenience, and based upon the size of our research team, we decided to use 25 papers, since this spread the re-writing task evenly among members of the team.

We used 25 papers taken from the set of 103 empirical papers previously identified and analysed by Sjøberg et al. (2005). This set of papers are taken from nine journals and three conference proceedings, published over the period 1993–2002. Dag Sjøberg provided a stratified random sample of the 103 papers that maintained the proportion of

journal papers and conference articles as present in the complete set. This resulted in a set of 6 conference articles and 19 journal papers.

2.2 Scoring Abstracts for Clarity and Completeness

Following the approach used by Hartley and Benjamin (1998), we measured completeness by means of a set of 18 questions asked about the content of the abstract. We adapted the questions used by Hartley and Benjamin to fit the context of Software Engineering experiments. The questions are shown in Appendix A. The questions map to different elements of a structured abstract as shown in Table 1. There are some disagreements between our allocation of questions to topic area and the allocation used by Hartley and Benjamin. Our allocation fits with our definition of the main elements of a structured abstract (see Section 2.3). Using the same approach as Hartley and Benjamin, clarity was scored as a subjective assessment on a scale of 1 to 10.

TABLE 1 Topic area addressed by Completeness Questions

Topic	Question Number
Background	1
Aim	2, 3
Method (experimental context)	4, 5, 6, 7, 8
Method (experimental conduct)	9, 10, 11, 12
Results	13, 14, 15
Conclusions	16, 17, 18

2.3 Rewriting abstracts into a structured form

Hartley's studies used two approaches to re-writing:

- all abstracts being re-written by the principal investigator (Hartley 2003);
- all abstracts being re-written by the original authors. This was required by one journal as a condition for publication (Hartley & Benjamin 1998).

In this experiment we used a modified form of the first approach. Team members re-wrote the abstracts but then sent the re-written abstract to the original authors to ask their permission to use it and to invite their comments and suggestions about the revised form. This was the approach taken in our previous observational study (Kitchenham et al., 2006a).

Each abstract was re-written by one member of the team and then checked and reviewed by another member of the team. Five team members were involved in re-writing abstracts and six were involved in checking. Team members were allocated to re-writing and reviewing at random (while ensuring no-one reviewed the same abstract they re-wrote). We used the following headings and contents guidelines to construct the structured abstracts:

Background: Previous research or rationale for a study.

Aim: Hypotheses to be tested or goal of the study.

Method: Description of the type of study, treatments (including control), number and nature of experimental units (which may be people, teams, algorithms, programs etc.) the experimental design, outcome being measured.

Results: Treatment outcome values, standard deviation and/or level of significance.

Conclusions: Future work, limitations of study.

We agreed that:

- Each heading should be set in boldface type and should end with a colon.
- The sentence beginning after the heading should start with an initial capital letter.
- References should be removed and acronyms should be expanded.
- Word count should be below 300 if possible.

As part of this process, we used the marking process for clarity and completeness ourselves to ensure that the questionnaire was usable and that we had some baseline score for future analysis (in particular we wanted a means of arbitrating results if participants marked the same abstract very differently). The researcher responsible for re-writing the abstract completed the questionnaire and the subjective assessment of clarity for the *original* abstract.

After the structured abstracts were reviewed by another researcher, they were sent to one of the authors of the original papers who were asked to confirm that the structured abstracts were a faithful representation of the article and/or to make any suggestions to improve/correct the abstract.

Sixteen of the authors responded to our e-mail and either agreed with the abstract or made suggestions for correcting the abstract. For the nine cases where we could not contact an author of the article, another member of the research team did a final review of the structured abstract. One other author replied agreeing with the abstract after the third review process. After all required changes were incorporated, the original reviewer assessed the structured abstract for completeness and clarity.

2.4 Data Preparation and Collection

One team member took responsibility for collating the data from each version of the abstract. The following data were collected for each abstract:

- Title and publication details of article
- Person responsible for constructing the structured version of the abstract (and evaluating the original abstract).
- Person responsible for reviewing the structured abstract (and evaluating the final version of the structured abstract).
- Word count for original abstract.
- Word count for structured abstract.
- Clarity score for original abstract.
- Clarity score for structured abstract.
- Answers for each completeness question for the original abstract.
- Answers for each completeness question for the structured abstract.

3. RESULTS

3.1 Descriptive Statistics

Table 2 shows the summary statistics for length, clarity and completeness for the original and structured abstracts. The completeness score was obtained by adding the number of 'Yes' responses to the 18 completeness questions. It appears that the structured abstracts are longer than the original abstracts but more complete and easier to read (as measured by clarity). These results are investigated in more detail below.

TABLE 2 Summary statistics for clarity and length (words)

Data	Observations	Mean	Standard deviation
Clarity original abstracts	25	5.64	2.02
Clarity structured abstracts	25	7.08	1.29
Length original abstracts (words)	25	161	59.15
Length structured abstracts (words)	25	267.4	47.09
Completeness score for original abstracts	25	5.64	2.62
Completeness score for structured abstracts	25	11.8	2.22

3.2 Length of Structured and Original Abstracts

Figure 1 shows the box plots of the length of the original and the structured abstracts. A paired "t" test indicates that the difference between the length of the structured and length of the original versions of the abstracts (106.4) is statistically significant ($p < 0.001$) with a 95% confidence interval 78.1 to 134.7.

FIGURE 1 Length of original abstracts and structured abstracts

The relationship between length of the original and the increase in length for the structured abstracts is shown in Figure 2. It is clear that the increase in length is greater for the smaller original abstracts than for the larger original abstracts. It should be noted that some of the original abstracts were very short because of publication limits on the size of abstracts, in particular abstracts from *IEEE Software*.

FIGURE 2 Scatter plot of the increase in length against the length of the original abstract

3.3 Clarity of original and Structured Abstracts

Figure 3 shows a box plot of the clarity values for the structured and original abstracts. A paired t test of the clarity values reveals that the structured abstracts are on average 1.47 points better than the unstructured abstracts ($p < 0.01$) with a 95% confidence interval of 0.47 to 2.41.

The clarity score has a maximum of 10, so it is interesting to identify whether the clarity score improvement for the structured abstracts is related to the score of the original abstract. Figure 4 makes it clear that the original abstracts that scored badly improved by much more than the original abstracts that scored well. In fact some of the top scoring original abstracts were scored at a lower level after being structured.

FIGURE 3 Box plot of clarity scores for structured and original abstracts

FIGURE 4 Scatter plot of the change in clarity compared with the original clarity score

3.4 Completeness of structured and original abstracts

Figure 5 shows the box plot of the completeness scores for the original and structured abstracts. A paired “t” test indicates that the structured abstracts are significantly more complete than the original abstracts ($p < 0.001$). The average difference between completeness of structured and original abstracts is 3.39 with a 95% confidence limit of 4.76 to 7.56.

Figure 6 confirms that increase in completeness for the structured abstracts is usually greater for the abstracts that originally exhibited a low level of completeness than for the abstracts that exhibited greater completeness.

FIGURE 5 Box plot of the completeness score for the original and structured abstracts

Difference in Completeness score

FIGURE 6 Difference between Completeness of Original and Structured Abstract compared with Original Score

3.5 Analysis of Missing Information

Table 3 indicates which parts of the original and structured abstracts were judged to be most incomplete. Fewer of the structured abstracts provided no information on a topic than the original abstracts. More than half the original abstracts provided no information about the Experimental context or the Results. For the structured abstracts, the worst result was for Background information, where 4 of the abstracts were judged to have provided no information.

TABLE 3 Distribution of Missing Data

Name	Questions	Number of questions	Average response for original abstracts	Original abstracts with no information	Average response for structured abstracts	Structured abstracts with no information
Background	Q1	1	0.72	7	0.84	4
Aim	Q2-Q3	2	1.08	4	1.32	0
Method – Experimental context	Q4-Q8	5	1.04	14	3.8	0
Method – Experimental conduct	Q9-Q12	4	0.96	9	2.04	2
Results	Q13-Q15	3	1.08	3	1.88	1
Conclusions	Q16-Q18	3	0.72	13	1.92	0

4. DISCUSSION AND CONCLUSIONS

Our results indicated that structured abstracts of software engineering experiments are longer than unstructured abstracts but are likely to be more complete and easier to read. These results are consistent with studies in other disciplines (Hartley, 2004, Hartley and Sydes, 1997, Hartley and Benjamin, 1998). However, the results are based on assessments made by the research team who produced the structured abstracts. It is clear that our interest in structured abstracts may have had an influence on our assessment of their completeness and clarity. For this reason, the experiment of which this is a part aims to elicit assessments from other researchers and students using a rigorous experimental design outlined in Section 2.

We found that longer unstructured abstracts generally did not require much additional content and were usually generally clearer than shorter unstructured abstracts. So would simply requiring authors to write longer abstracts improve abstracts? Currently we cannot tell whether additional length is a cause or an effect of good abstracts. This is an area for future work. However, structured abstracts at least provide a clear rationale for what should be reported that can help researchers decide what additional material to supply.

Another limitation of this study is that the completeness criteria are based on the structure we used to construct the structured abstracts. In order to address this issue, we compared our questions to the questions defined in the perspective-based evaluation of guidelines for reporting experiments compiled by Kitchenham et al. (2006b). We found no questions in our questionnaire that were not included in the perspective-based questions. Two perspective-based questions were not addressed by our questions, so we added two related questions to our questionnaire (Budgen et al., 2006).

Finally, our results apply to papers that describe experiments, according to the definition provided by Sjøberg et al. (2005). We cannot claim that structured abstracts of all software engineering articles would exhibit the same properties. However, given the results of our previous observational study which covered more general empirical studies (Kitchenham et al., 2006a), we would expect our results to generalize to most types of empirical study.

REFERENCES

- Bayley, L. and Eldredge, J. (2003) The structured abstract: an essential tool for researchers, *Hypothesis*, **17**(1), 11–13.
- Brereton, O., Kitchenham, B., Budgen, D., Turner, M. & Khalil, M. (2007) Lessons from applying the Systematic Literature Review process within the Software Engineering domain, *J. of Systems & Software*, **80**(4), 571-583.
- Budgen, D., Kitchenham, B., Charters, S., Turner, M., Brereton, P. and Linkman, S. (2006) Protocol for an Experimental Study of the Use of Structured Abstracts, Version 1.3, October 5th.
- Hartley, J. (2003) Improving the clarity of journal abstracts in psychology: The case for structure, *Science Communication*, **24**, 366–379.

- Hartley, J. (2004) Current findings from research on structured abstracts, *J. of Medical Library Association*, **92**, 368–371.
- Hartley, J. and Benjamin, M. (1998) An evaluation of structured abstracts in journals published by the British Psychological Society, *British J. of Educational Psychology*, **68**, 443–456.
- Hartley, J. and Sydes, M. (1997) Are structured abstracts easier to read than traditional ones?, *J. of Research in Reading*, **20**, 122–136.
- Kitchenham, B. (2004) Procedures for undertaking systematic reviews, Technical Report TR/SE-0401, Department of Computer Science, Keele University and National ICT, Australia Ltd. Joint Technical Report.
- Kitchenham, B.A., Brereton, O.P., Owen, S., Butcher, J. and Jefferies, C. (2006a) Length and Readability of Software Structured Abstracts, Keele University Technical Report.
- Kitchenham, B., Al-Khilidar, H., Babar, M. A., Berry, M., Cox, K., Keung, J., Kurniawati, F., Staples, M., Zhang, H. & Zhu, L. (2006b) Evaluating Guidelines for Empirical Software Engineering Studies. *Proceedings of ISESE 2006*, ACM Press.
- Sjøberg, D., Hannay, J., O.Hansen, Kampenes, V., Karahasanovic, A., Liborg, N.-K. and Rekdal, A. (2005) A Survey of Controlled Experiments in Software Engineering, *IEEE Trans. on Software Engineering*, **31**(9), 733–753.
- Webster, J. and Watson, R. (2002) Analysing the Past to Prepare for the Future: Writing a Literature Review, *MIS Quarterly*, **26**, xiii–xxiii.

Acknowledgements

We thank Dag Sjøberg for providing a random selection of papers to use in this study. This work was supported by the EPSRC Evidence-Based Software Engineering (EBSE) project (EP/C51839X/1).

APPENDIX A COMPLETENESS QUESTIONS

1. Is the rationale for the study reported?
2. Is the aim/purpose of this study reported?
3. Is a hypothesis (or hypotheses) provided?
4. Is there any indication of where this study took place? (E.g. in industry or academia, what the application domain was, etc.)
5. Is the number of participants reported?
6. Are the types of participant (e.g. students) reported?
7. Is any information about the experience of the participants provided?
8. Is the skill level of the participants described?
9. Is there any description of how the study was performed?
10. Does it report how the participants were allocated to different tasks or conditions?
11. Is the way that the data was collected reported?
12. Is there any description of the form of analysis performed?
13. Are the main results summarised in the abstract?
14. Are actual numbers from the results presented in the abstract?
15. Is any statistical information provided about the results?
16. Are any conclusions drawn?
17. Are any limitations of the study identified?
18. Is there any discussion of required future research?