Annotated metamodel for MSR course papers
Version as of 30 June 2015

Ralf Lämmel

Software Languages Team, CS Faculty, University of Koblenz-Landau

Abstract. This should be 100-200 words. Use present or past tense, but use it consistently. Prefer "we" over "I" here and throughout the paper. The abstract should summarize the methodology and results; see the corresponding sections below. Your paper is preferably typeset with LaTeX and it should comply with the Springer LNCS style. (That style is also available for word users.)

1 Introduction

Address these questions:

- What is the context of this research?
- What is the overall challenge addressed by this research?
- What is the overall approach adopted by the research?
- Summarize results very briefly.

2 Research question

State 1-5 research questions. One question could be enough. More is not better.

For each question, make clear why this is an interesting question. For example, try to make clear why a scientist or a practician would benefit from knowing the answer to the question.

Lay out some background for your research questions. That is, what body of related work may be worth mentioning and comparing to the research at hand. This may be related in terms of both relevant methodology or similar research questions.

You should make sure to carry out a proper related work study. Use DBLP on the MSR conference for 2+ years.[1] You are welcome to search other conferences and journals. Coming to the conclusion that there is no related work is absolutely unacceptable. Again, you need to show related work awareness so that your research question is substantiated as being relevant and your methodology is deeply informed by existing publications.

In many cases, you want to refine your research questions into actual hypotheses, i.e., something that is potentially falsifiable, subject to an appropriate approach towards measuring.

[1] http://dblp.uni-trier.de/db/conf/msr/

It is also common to refine research questions. That is, one starts with a general (overall) research question and takes it appart into several questions that are concrete enough to be researched on and to be answered. In this manner, you may often suffice with one or two research questions which are however refined into perhaps several questions. Again, even without refinement, just one question may be Ok. This depends on the nature of the project and on the kind of question.

Here are some recommended MSR papers by the Software Languages Team [3,4,6] which you could consult to see instances of the concepts discussed in this metamodel. These papers may or may not be in an optimal form to function as good advice. You are also advised to have a look at some MSR papers (from the MSR conference), but again, they may also slightly vary in terms of consistency of style. if you like to get general advice on scientific writing, then I recommend relevant textbooks [1,7]. The best way to learn writing is by writing and by having someone with experience to provide feedback.

You should most definitely have a look at this excellent tutorial on empirical software engineering (ESE) [2] to which MSR relates. General ESE is quite often concerned with controlled experiments and other research methods that involve subjects, whereas MSR is more concerned with automated analysis, but the text is still very helpful in coming to grips with research methods in software engineering overall. Specifically, it helps with formulating research questions.

3 Methodology

The methodology section must give a very clear account of what sort of procedure was executed. Here are some detailed criteria:

- The section is preferably a step-by-step procedure overall. First we did this. Then we did this.
- Steps should be motivated or defended, whenever it is not obvious why a certain choice was made. However, one should be careful not to preempt the discussion of threats to validity; see the extra section below.
- The text should support reproducibility. In particular, a reasonably knowledgeable person should be able to re-execute the methodology and compare the results.
- The text should not get into low-level (implementation) details. These can be deferred to separate documentation; see Appendix A.

3.1 Data extraction

This metamodel is concerned with MSR papers. Thus, the methodology section would be reasonably subdivided into the three major phases of an MSR research project.

The subsection on data extraction is concerned with issues like the following:

- What repository was chosen and why?

- What sort of extraction techniques is applied?
- What sort of extra processing (e.g., filtering) is applied?
- What constraints were chosen to help with scalability?

3.2 Data synthesis

Even the simplest MSR project carries out synthesis; data analysis (see below) may not be carried out in all the cases. The subsection on data synthesis is concerned with issues like the following:

- What metrics are used?
- What machine learning techniques are used?
- What information retrieval techniques are used?

The typical MSR paper picks either metrics or machine learning or IR.

3.3 Data analysis

Data analysis would be concerned with any sort of statistical quality analysis of the data. More specifically, the subsection on data analysis is concerned with issues like the following:

- Simple statistics like median of metrics.
- Analysis of regression or correlation or distribution.
- Analysis of accuracy such as precision and recall.

It should be noted that the methodology section describes the 'how' (and to some extent the'why'), but it does not yet report the various results; see the following section. For instance, the methodology may explain why it is using a certain metric and define it and announce that the median for the metric is going to be determined, as it would provide a certain insight, but the actual tables or charts for the metric and the interpretation of the findings would be deferred to the results section.

4 Results

Summarize results in terms of tables, charts, and other kinds of figures.

Explain and interpret the results.

Get back to the research question and make sure that it is explicitly answered.

5 Threats to validity

There are external and internal threats to validity. See the softlang papers and MSR papers for examples. See this resource [5] for an explanation. Be concise and systematic about answering questions.

6 Conclusion

This is essentially a redundant section. Try to summarize your research from a more informed point of view. (We assume that the reader has read the paper and is ready for some deeper conclusion.) You could very well add some perspectives for future work, without repeating smaller scale weakness of research, as they were already identified in the threats to validity section.

A Details

Any sort of interesting details such as source code, extra data (extra to the key results), extra procedural details supporting reproducibility can be listed here.

References

1. R. A. Day. *"How to Write & Publish a Scientific Paper"*. B&T, 1998. 5th edition.
2. S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. Selecting Empirical Methods for Software Engineering Research. In *Proc. of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, ASE '07, pages 574–574. ACM, 2007. Technical report. Available online `http://www.cs.utoronto.ca/~sme/papers/2007/SelectingEmpiricalMethods.pdf`.
3. R. Lämmel, R. Linke, E. Pek, and A. Varanovich. A Framework Profile of .NET. In *Proc. of WCRE 2011*, pages 141–150. IEEE Computer Society, 2011. Available online `http://softlang.uni-koblenz.de/dotnet/`.
4. R. Lämmel and E. Pek. Understanding privacy policies - A study in empirical analysis of language usage. *Empirical Software Engineering*, 18(2):310–374, 2013. Available online `http://softlang.uni-koblenz.de/p3p/`.
5. R. Michael. Threats to Internal & External Validity. Part of a course 'Strategies for Educational Inquiry'. Available online `http://www.indiana.edu/~educy520/sec5982/week_9/520in_ex_validity.pdf`, 2004.
6. T. Schmorleiz and R. Lämmel. Similarity management of 'cloned and owned' variants. Submitted for publication. 10 pages. Available online `http://softlang.uni-koblenz.de/simman/`, 2015.
7. J. Zobel. *"Writing for Computer Science"*. Springer, 2009. 2nd edition.