Translating between XML and Relational Databases

Sridhar Sarnobat

June 20th, 2004

Contents

Abstract

Abstract The abstract is a very brief summary of the report's contents. It should be about half a page long. Somebody unfamiliar with your project should have a good idea of what it's about having read the abstract alone and will know whether it will be of interest to them.

0.1 Title Page

This should include the project title and the name of the author of the report. You can also list the name of your supervisor if you wish.

0.2 Acknowledgements

It is usual to thank those individuals who have provided particularly useful assistance, technical or otherwise, during your project. Your supervisor will obviously be pleased to be acknowledged as he or she will have invested quite a lot of time overseeing your progress.

0.3 Contents page

This should list the main chapters and (sub)sections of your report. Choose self-explanatory chapter and section titles and use double spacing for clarity. If possible you should include page numbers indicating where each chapter/section begins. Try to avoid too many levels of subheading. Try if possible to stick to sections and subsections; sub subsections are usually avoidable.

0.4 Introduction

This is one of the most important components of the report. It should begin with a clear statement of what the project is about so that the nature and scope of the project can be understood by a lay reader. It should summarise everything you set out to achieve, provide a clear summary of the project's background and relevance to other work and give pointers to the remaining sections of the report which contain the bulk of the technical material.

0.5 Background

The background section of the report should set the project into context by relating it to existing published work which you read at the start of the project when your approach and methods were being considered. There are usually many ways of solving a given problem, and you shouldn't just pick one at random. Describe and evaluate as many alternative approaches as possible. The background section is often included as part of the introduction but can be a separate chapter if the project involved an extensive amount of research. The published work may be in the form of research papers, articles, text books, technical manuals, or even existing software or hardware of which you have had hands-on experience. Don't be afraid to acknowledge the sources of your inspiration; you are expected to have seen and thought about other people's ideas; your contribution will be putting them into practice in some other context.

However, avoid plagiarism: if you take another person's work as your own and do not cite your sources of information/inspiration you are being dishonest; in other words you are cheating. When referring to other pieces of work, cite the sources where they are referred to or used, rather than just listing them at the end.

0.6 Body of report

The central part of the report usually consists of three of four chapters detailing the technical work undertaken during the project. The structure of these chapters is highly project dependent. They can reflect the chronological development of the project, e.g. design, implementation, experimentation, optimisation, evaluation etc. although this is not always the best approach. However you choose to structure this part of the report, you should make it clear how you arrived at your chosen approach in preference to the other alternatives documented in the background. If you have built a new piece of software you should describe and justify the design of your program at some high level, e.g. using UML, OMT, Z, VDL, etc., and should document any interesting problems with, or features of, your implementation. Integration and testing are also important to discuss in some cases.

0.7 Evaluation

Be warned that many projects fall down through poor evaluation. Simply building a system and documenting its design and functionality is not enough to gain top marks. It is extremely important that you evaluate what you have done both in absolute terms and in comparison with existing techniques, software, hardware etc. This might involve quantitative evaluation, for example based on numerical results, performance etc. or something more qualitative such as functionality, ease-of-use etc. It may also involve a discussion of the strengths and weaknesses of what you have done. Avoid statements like "The project has been a complete success and we have solved all the problems asssociated with ¡blah...¿" - you will be shot down immediately! It is important to understand that there is no such thing as a perfect project. Even the very best pieces of work have their limitations and you are expected to provide a proper critical appraisal of what you have done. Your assessors are bound to spot the limitations of your work and you are expected to be able to do the same.

0.8 Conclusions

and Future Work The project's conclusions should list the things which have been learnt as a result of the work you have done. For example, "The use of overloading in C++ provides a very elegant mechanism for transparent parallelisation of sequential programs", or "The overheads of linear-time n-body algorithms makes them computationally less efficient than O(n log n) algorithms for systems with less than 100000 particles". Avoid tedious personal reflections like "I learned a lot about C++ programming...", or "Simulating colliding galaxies can be real fun...". It is common to finish the report by listing ways in which the project can be taken further. This might, for example, be a plan for turning a piece of software or hardware into a marketable product, or a set of ideas for turning your project into an MPhil or PhD.

0.9 Bibliography

This consists of a list of all the books, articles, manuals etc. used in the project and referred to in the report. You should provide enough information to allow the reader to find the source. You should give the full title and author and should state where it is published, including full issue number and date, and page numbers where necessary. In the case of a text book you should quote the name of the publisher as well as the author(s).

0.10 Appendix

The appendices contain information which is peripheral to the main body of the report. Information typically included are things like parts of the code, tables, proofs, graphs, test cases or any other material which would break up the theme of the text if it appeared in situ. You should try to bind all your material in a single volume if possible.

0.11 User Guide

For projects which result in a new piece of software you should provide a proper user guide providing easily understood instructions on how to use it. A particularly useful approach is to treat the user guide as a walk-through of a typical session, or set of sessions, which collectively display all the features of your package. Technical details of how the package works are rarely required. Keep it concise and simple. The extensive use of diagrams illustrating the package in action prove particularly helpful. The user guide is sometimes included as a chapter in the main body of the report, but is often better as an appendix to the main report.

0.12 Program Listings

Complete program listings should NOT be part of the report. Instead you should create a "project" directory containing your code (and report files) and should include the path name somewhere transparent in your report (e.g. at the front). Make sure this directory is readable! You may additionally submit the contents of the directory on a CD.