

Guardian Project

Project Report

Niko Kotilainen, Ville Pentti, Jussi Rastas and Joni Töyrylä

Version 1.1, January 14, 2003

University of Jyväskylä
Department of Mathematical Information Technology

Abstract

Guardian project implemented a testing and monitoring tool for a Cheddar network developed by Agora Center's InBCT/Cheese Factory Project. Project Report document describes how the project was carried out and compares the realized project to Project Plan.

Contents

1	Introduction	1
2	Terms	2
3	Background of the Project	3
4	Introduction to the Software	4
5	Resources	5
5.1	Human Resources	5
5.2	Material Resources	5
6	Division of the Tasks	6
7	How the Goals Were Reached	8
7.1	Software Goals	8
7.2	Educational Goals	8
8	The Realized Schedule	9
8.1	Project Phases	9
8.2	Planning Phase	10
8.3	Implementation Phase	11
8.4	Test Phase	12
8.5	Final Documentation Phase	12
9	The Realization of the Risks	13
10	Project Experiences	15
10.1	Common Experiences	15
10.2	Niko Kotilainen	16
10.3	Ville Pentti	16
10.4	Jussi Rastas	16

10.5 Joni Töyrylä	17
11 Conclusion	18
References	19

1 Introduction

Guardian was a student software project carried out in the University of Jyväskylä during autumn 2002. The project implemented a software called Peer-to-Peer Studio for testing and monitoring a Cheddar network. The tool enables to remotely control and monitor each Cheddar client in a centralized way. It can be used to show the topology of the network and to draw graphs on monitoring data.

In addition to Project Report the project has written User Interface Specifications, User Interface Design, Server Specifications, Server Design and Software Report. The specifications documents describe how the requirements are transformed to features. The design documents describe the actual implementation with more details including the operations from general overview to the level of subroutines. Software Report has been written in the final phase and the testing phase has provided Test Plan and Test Report.

Project Report describes how the goals of the project were reached and how the project was carried out compared to Project Plan. The terms and abbreviations related to the project are described in Chapter 2. Chapter 3 presents background of the project while Chapter 4 introduces briefly the developed software. Chapter 5 describes the resources used in the project. The realized division of the tasks is presented in Chapter 6 while Chapter 7 discusses how the set goals were reached. Chapter 8 describes the realized schedule of the project. The realization of the risks are discussed in Chapter 9 while Chapter 10 introduces experiences of the project members.

2 Terms

The following terms are used in this document and in the project.

Peer-to-peer	(P2P) is a computing concept which enables sharing of computer resources and services by direct exchange. This allows equal peers on the edges of a peer-to-peer network to connect and share resources without a centralized server.
Chedar	(CHEap Distributed ARchitecture) is a peer-to-peer computing platform developed by Cheese Factory Project. It can be used to build a network of workstations each node providing and consuming computing resources.
XML	(eXtensible Markup Language) is a markup language for documents containing structured information.
XSL	(eXtensible Stylesheet Language) is a language that allows to describe how to transform and display an XML document of a given type.
XSLT	(eXtensible Stylesheet Language Transformations) is a language for transforming XML documents into other XML documents.
Xerces	is an XML parser made by The Apache Software Foundation.
Xalan	is an XSLT processor for transforming XML documents into HTML, text or other XML document types. It implements the W3C recommendations for XSL transformation (XSLT) and the XML Path Language (XPath). It is made by The Apache Software Foundation.
HTML	(HyperText Markup Language) is a language for documents published in the World Wide Web.
Java	is a platform independent programming language developed by Sun Microsystems, Inc.
C++	is an object oriented programming language.

3 Background of the Project

Recently peer-to-peer networks have gained much publicity. They present a different aspect from the vulnerable centralized networks. The structure of a peer-to-peer network is totally decentralized and clients in this kind of a network are equal to each other. Because there is no central servers there won't be any single point of failures or performance bottlenecks and the network is highly fault-tolerant and robust.

The customer of the project, Cheese Factory, studies peer-to-peer communication and behavior of peer-to-peer networks. Cheese Factory has also implemented a Java-based peer-to-peer computing platform called Cheddar for the research. In order to test and monitor this Cheddar network there was a need for a tool that enables to remotely control and monitor each Cheddar client and the whole network. The project group got the assignment to implement the tool software.

The main research themes that will be investigated with the software will be formation of peer-to-peer networks and different search algorithms in peer-to-peer networks. Also peer-to-peer network resilience under attacks and random failures will be researched with the application.

4 Introduction to the Software

The software was named **Peer-to-Peer Studio**. It was decided that the monitoring software will be divided into two separate programs: The server and the user interface. The user interface communicates with the server which communicates in the middle also with Chedar clients (see Figure 1).

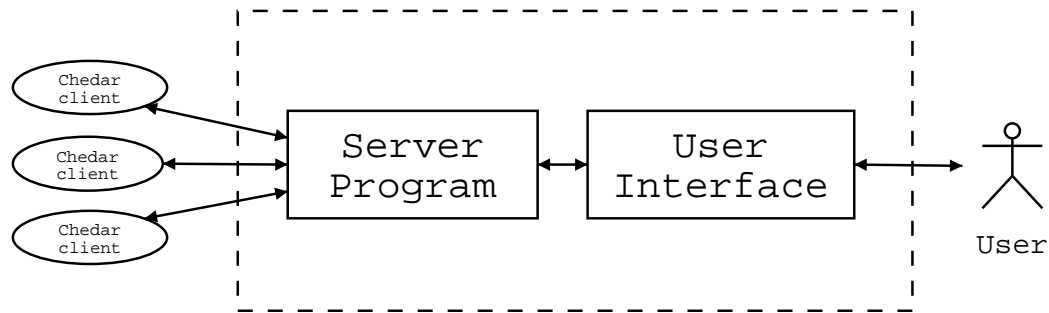


Figure 1: The general structure of the system.

The server part gathers information about monitored Chedar clients, assembles the data for the user interface and carries out the commands received from the user interface. The user interface shows the monitoring data to the user in forms of topology views, graphs and text. To achieve efficiency to the server it was implemented with C++ and the user interface was implemented with Java in order to be runnable on different platforms.

5 Resources

Human and material resources were used in the project and they are described in this chapter.

5.1 Human Resources

The project consisted of the four following students studying at the University of Jyväskylä:

- Niko Kotilainen (npkotila@cc.jyu.fi),
- Ville Pentti (vipentti@cc.jyu.fi),
- Jussi Rastas (jprastas@cc.jyu.fi) and
- Joni Töyrylä (jtoyr@cc.jyu.fi).

The supervisor in charge was Jukka-Pekka Santanen from the Department of Mathematical Information Technology. Matthieu Weber worked as a technical supervisor of the project. The representative of the customer was Mikko Vapa from the Cheese Factory project. Annemari Auvinen took also part in the project developing Chedar.

5.2 Material Resources

Department of Mathematical Information Technology provided a project room with four workstations, a phone and office supplies. There were also ten workstations and Chedar system of Cheese Factory in use of the project.

One of the workstations in the project room was running Windows 2000 and three of them were equipped with Linux Red Hat Release 7.3. Cheese Factory workstations were also running on Linux. In the implementation of the tool the project used JBuilder 6.0 Professional and gcc (GNU C Compiler) version 2.96. The documentation was produced with \LaTeX .

The department provided an E-mail list guardian@korppi.jyu.fi and a mail archive and homepages located at <http://kotka.it.jyu.fi/guardian/>.

6 Division of the Tasks

The planned responsibilities of the tasks are introduced in Project Plan. Some of the items were removed or renamed, because the software and tasks with it have evolved. Table 1 presents roughly how the tasks were divided between the project members.

Planning	
Project Plan	Jussi
Server Program Specifications	Ville
Server Program Design	Joni, Ville
User Interface Specifications	Jussi
User Interface Design	Niko
Server Program	
Communication	Joni
Xerces	Joni
Xalan	Joni
XSL	Joni, Jussi, Ville
User Interface	
Communication	Niko
Interface	Niko
Topology visualization	Niko
Testing	
Testing Plan	Ville, Ville
Testing	All
Testing Report	Jussi
Final documentation	
Project Report	Jussi
Software Report	Ville, Jussi
Presentations	All
Project Management	Jussi

Table 1: Task division.

To explicate Table 1 Jussi and Ville took care of the documentation except the design documents which were mainly created from the comments in the code. The entire server program was implemented by Joni. The only assistance was provided to Joni in writing the XSL rules by Jussi and Ville. Niko implemented the user interface completely on the basis of common ideas.

The design work including program, interface and protocol designs were taken care of by all of the group members. Jussi took care of the project management. The work amount was equally divided between the group members in spite of the described division.

The planned division of the tasks differs a bit from the realized division. The original idea was that the two parts of the software would have been programmed in teams of two group members: Joni leading the programming of the server and Niko leading the programming of the user interface.

However it turned out that the documenting in English with L^AT_EX required more time than expected. It was also realized that the timetable was very strict. The project had to be carried out in four months and the software had to be ready in three months. Because of the timetable and the fact that we had to add more resources to the documenting the tasks were divided among the group members mainly on the basis of prior skills. This was probably the only way to achieve the goals of the project on the timetable.

The new division of the tasks did not cause any troubles. The only negative thing which derived from the rearrangements was that the programming lost some amount of resources.

7 How the Goals Were Reached

The chapter describes how the goals set in Project Plan were reached.

7.1 Software Goals

The main goal of the project to implement the software with the defined features for testing and monitoring a Cheddar network was reached. The high priorities described in the Requirements document provided by Cheese Factory were implemented except the firewall controlling and *TrafficMeter* showing features were implemented partly. Some of the medium priorities and a couple of low priorities were realized. All the realized requirements are introduced with more details in Software Report.

7.2 Educational Goals

The group members learned the elements of teamwork, communication inside a project group and to the customer and the supervisors as well as meeting behaviour. Respect for the different parts of the project was also learned and interaction with the customer was found important.

The project members got to know new tools and learned programming and documenting, but all of the group members did not get experience in all fields. This was because the short time for the project did not allow the project members to switch the roles. In addition the English information technological vocabulary of each group member was expanded.

8 The Realized Schedule

The chapter shows with timetables how the project was carried out comparing to the estimations in the project plan. The white lines in the figures represent the estimated time while the black lines describe how much time was really needed.

8.1 Project Phases

Figure 2 shows how the project phases were carried out in time. In general each phase was completed almost on estimated time except the testing phase. The whole project was finished by the middle of January as expected. More detailed descriptions for each phase can be found from the following sections.

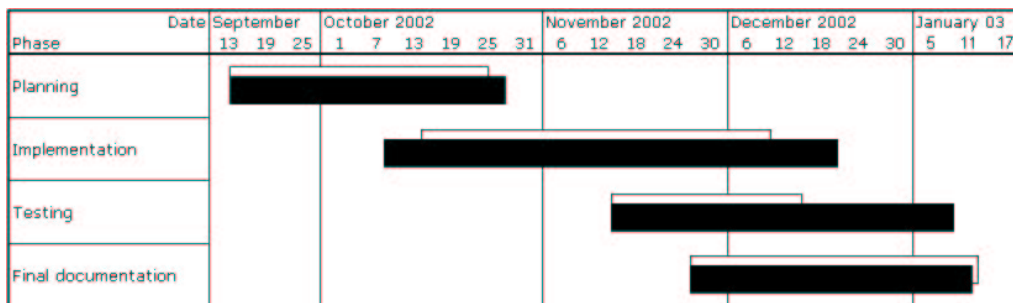


Figure 2: The project phases.

8.2 Planning Phase

Figure 3 describes how the planning phase was realized. The planning phase was delayed, because the composing of the documents took more time than expected and the ways of implementing the software were changed relatively late.

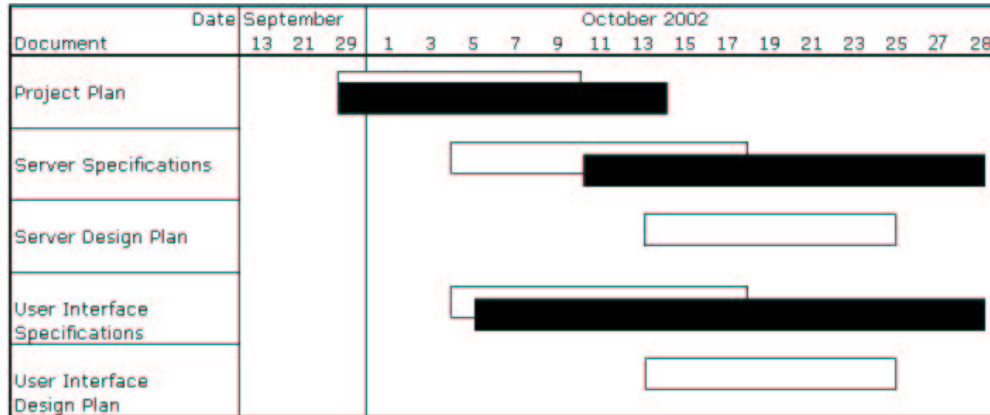


Figure 3: The planning phase.

The planning for the project and the software was started right after the first meeting on September 13th. The project plan was composed in the beginning of the phase and approved almost on estimated time.

The writing of the server specifications document was delayed a bit, because the ways of implementing the software were unclear until the middle of October. The user interface specifications document took also more time than estimated, because the user interface was designed more carefully. Also composing text in English and achieving the required layout with \LaTeX took a remarkable amount of time. The first versions of the specification documents were provided on 28th of October.

It was decided that the design documents will be provided afterwards as a part of the final documentation. They were mainly created out of the comments.

8.3 Implementation Phase

Figure 4 shows how the implementation phase was carried out. The names of the tasks are different from Project Plan, because the implementation of the server was changed after the document was provided. The implementation phase took a little longer than estimated.

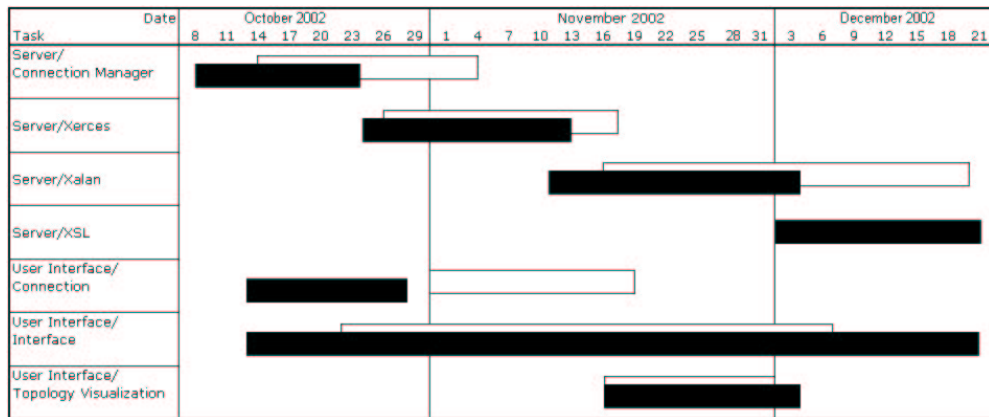


Figure 4: The implementation phase.

The implementation was started ahead of the estimated time with the communication components of the software. Both of them were completed much faster than expected. Xerces was added to the server on time as well as Xalan regardless of the problems caused by bad documentation. The XSL rules and the handling of the messages was the last, but time-consuming task for the server. The time needed for the task was underestimated and this caused a little delay for the software. The implementation of the user interface was started earlier than estimated and completed almost on time including the graphs component and topology drawing component.

8.4 Test Phase

The test phase was delayed with the whole software. Test Plan was written on time, but the actual tests could not be held until the entire software was completed. The Christmas holidays also delayed the phase. Eventually the tests were held on January and they included the testing of the user interface, the server and the Cheddar network.

The tests revealed a bunch of lacks in the server and the Cheddar system. Test Report was filled in the test sessions.

8.5 Final Documentation Phase

The final documentation phase was carried out on time. The composing of Project Report was started first and Software Report right after it. The design documents were provided with the report documents by the middle of January. Also the final version of User Interface Specifications was provided.

9 The Realization of the Risks

The chapter discusses about the realization of the risks and also about avoiding and handling them. Table 2 describes how each risk was realized in scale from 1 to 5. The item caused some minor problems when the realization is low and a high degree implies about greater problems.

Risk	Realization
Unfamiliarity with the tools	4
Chedar under development	3
Used language	2
Inexperience in software projects	2
Other risks	1
Unfamiliarity with the programming languages	-

Table 2: Realization of the risks in scale from 1 to 5.

The biggest problems were faced with the unfamiliar tools. In the implementation of the server, Xalan C++ and Xerces C++ caused problems, because the documentation and the examples did not match with the versions in hand. It is also worth mentioning that Xalan C++ does not support new XML Trees of Xerces C++. None of these facts were informed in the specifications.

The documentation of the project was composed with L^AT_EX which is not so easy in modifying the layout. Because the software projects need to hold on to quite strict layout, the project group should have known T_EX programming language.

The project group was also partly unfamiliar with XML, but managed to cope with it. This is due to the fact that all of the project members knew HTML which has similarities with XML. The project group had to study also XSL which was not so familiar.

Some of the project group members did not have experience with Linux, but the operating system was found good and quite easy to use. Though WTS Veivi was documented poorly, it was seized without bigger problems. The unfamiliarity with the tools could only be taken care of by studying the tools and by giving more time for the tasks which require them.

The implementation of the Chedar system which the software was developed for was delayed. It also had some bugs and clients had a habit to

crash after a while when started. This complicated the implementation and the testing of the software and partially delayed it. Some of these problems might have been avoided by testing and studying the system.

The language used in the project was English while the native language of the project members is Finnish. This slowed down the composing of the documentation and caused some minor misunderstandings. It was also usual that all of the group members did not catch everything in the meetings.

Inexperience in software projects caused some misjudgements in task scheduling. In some cases times for the tasks were hard to set when the ways of implementation were erratic and not so clear. The only phase which was remarkably delayed was the test phase which was due to inexperience. We were too optimistic in the implementation phase. Another thing which derived from inexperience was that it was quite hard to start planning and implementing a software of this magnitude.

There were also some other risks which also affected a bit to the project. All of the group members were ill at least once. There were also some amount of other studies to be carried out. In addition Joni worked half time in a company. These risks were taken into account in the planning.

Unfamiliarity with the programming languages did not cause any problems if we do not take \TeX into account. This was due to the fact that the project members used the programming languages they were familiar with.

10 Project Experiences

The chapter discusses about the experiences of the group members collectively and individually.

10.1 Common Experiences

Generally the project was a good experience and the project group turned out to be a very good and active. The group members were very satisfied with the project and the outcome of it. It is worth mentioning that it was very easy and comfortable to work with the understanding customer of the project.

The project was on the other hand found a bit hard due to the huge amount of working hours. The time for the project turned out to be a very short time period to carry out a software project of this magnitude. The time should have been at least doubled to fulfil the project properly and to reach all of the educational goals. With this kind of a tight schedule the group members cannot focus on the matters which would particularly require learning.

Also because of the short time period there should not have been so many documents for the project group to compose. The software plan which has been usually written in the software projects was in this project divided into four separate documents. The composing of four documents instead of one took a multiple time and they served hardly the original purposes in the short project.

In the beginning it was hard to start planning and programming, because none of the project group members had experience in implementing applications of this magnitude especially in this field. The tasks became clear after a short or a longer while when examined the issue.

If the project was started now from the beginning there would not be any bigger changes. Of course we know now how everything should be done and get rid of the unnecessary complications. Worth mentioning is that Test Plan should have been composed next to implementing the software as the technical supervisor of the project suggested.

10.2 Niko Kotilainen

The project was very time-consuming and demanding. The subject was really interesting but maybe too broad when taking into consideration the fact that the time of the project was so limited. The members of the project group were highly motivated and it was a pleasure to work with them.

Personally I have learnt a lot about software projects and about Java programming.

10.3 Ville Pentti

The subject of the project was very interesting, because it was related to data communications. The system to be implemented wasn't very clear at the beginning, which produced much work. But finally the implementation got started at reasonable time. Project succeeded quite well and requirements were implemented quite extensively. On the other hand project was a little late.

The software project taught me some things about project working. I also learned many new technics and things from data communications. I would have liked to do more coding, but it was impossible, because the documenting took so much time in the project.

10.4 Jussi Rastas

The project group got a big and challenging task to carry out. After hard work the project was completed successfully in a very short time though the test phase was delayed.

The group members grew into a good and effective team in which everyone had a clear and important role. The group members really immersed themselves in the project what can be concluded from the amount of working hours.

Many things were learned about software projects, new working tools and topical network concepts. Also my English is more fluent than ever especially in the field of information technology. In addition I learned some things about managing a project and even leadership.

The only thing that harms me a bit is the fact that the tight timetable didn't allow me to do programming. In my opinion the project group members can be very satisfied with the outcome and the whole project.

10.5 Joni Töyrylä

The subject of the project was interesting but also quite large and complex. Although there were lots of new things to learn we managed to complete the project on time. Project members were hardworking and dedicated. In this autumn I have made lots of progress in my coding skill, but also learned how a project starts, how it is carried on and how it is finally finished.

11 Conclusion

The project was carried out successfully and nearly on estimated schedule. The goals were reached and the software was implemented with almost all features presented. The group members as well as the customer and the supervisors were very satisfied with the project.

References

Aho Pasi, Härkönen Henrik, Lahti Miikka and Rajala Minna, “Project Plan”, Nexus Project, University of Jyväskylä, Department of Mathematical Information Technology, 18.3.2002.

Arnold Ken, Gosling James and Holmes David, “The Java Programming Language, Third Edition”, Addison-Wesley, November 2000.

Cheese Factory Project, “Cheddar-Control Tool Requirements”, Agora Center, InBCT project, University of Jyväskylä, October 2002.

Cheese Factory Project, “Guardian-to-Cheddar Protocol Specification”, Agora Center, InBCT project, University of Jyväskylä, October 2002.

Cheese Factory Project, “Overview”, Agora Center, InBCT project, available in HTML form at <URL: <http://tisu.it.jyu.fi/cheesefactory/>>, read on 15.10.2002.

Diller Antoni, “ \LaTeX Line by Line”, Second Edition, John Wiley & Sons Ltd, 1999.

Free Software Foundation, “The GNU C Library”, available in HTML form at <URL: <http://www.gnu.org/manual/glibc-2.0.6/>>, December 30th, 1998.

Friman Juha, Kinnunen Suna and Syrjälä Ilkka, “Projektisuunnitelma”, Veivi Projekti, University of Jyväskylä, Department of Mathematical Information Technology, 11.3.2002.

Friman Juha, Kinnunen Suna and Syrjälä Ilkka, Ohjelmakoodit, Veivi Projekti, University of Jyväskylä, Department of Mathematical Information Technology, 17.5.2002.

Kaijanaho Antti-Juhani, “ \LaTeX ja $\text{\AMS-}\text{\LaTeX}$ ”, University of Jyväskylä, Computer Center, September 30th 1998.

Kallela Maija-Leena, Suurpää Anna-Liisa and Bird Timothy, “REACTION”, 4.-15. painos, WSOY-yhtymä Weilin+Göös, 1996.

Kielikone Oy, “MOT Sanakirjasto”, available in HTML form for the University of Jyväskylä at
<URL: http://www.trantor.fi/MOT_Sanakirjasto.htm>, read on 20.12.2002.

Kopka Helmut and Daly Patrick W., “A Guide To \LaTeX ”, Third Edition, Addison-Wesley, 1999.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “Project Plan”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, November 7th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “Server Specifications”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, November 28th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “User Interface Specifications”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, January 8th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “Server Design”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, January 8th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “User Interface Design”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, January 8th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “Test Plan”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, December 16th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “Test Report”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, January 10th 2002.

Kotilainen Niko, Pentti Ville, Rastas Jussi and Töyrylä Joni, “Software Report”, Guardian Project, University of Jyväskylä, Department of Mathematical Information Technology, January 9th 2002.

Lappalainen Vesa, “Ohjelmointi++”, University of Jyväskylä, Department of Mathematical Information Technology, 1998.

Manninen Kalevi, “XSL, XSLT ja XPath”, University of Jyväskylä, Department of Computer Science and Information Systems, available in HTML form at <URL: <http://www.infotech.jyu.fi/~kamannin/eds2002/weppisivut/XSL/XSL.doc>>, read on 11.12.2002.

Oksanen Erkki, “UNIX ja sen hallinta”, Pagina, 1992.

Ruini Henri, “Extensible Stylesheet Language: Transformations (XSLT)”, available in HTML form at <URL: <http://www.cs.helsinki.fi/u/ruini/structure/xsl/xslt-perusteet/index.html>>, March 6th, 2002.

Santanen Jukka-Pekka, “Opinnäytteiden kirjoittaminen, lyhyt oppimäärä”, University of Jyväskylä, Department of Mathematical Information Technology, available in HTML form at <URL: <http://www.mit.jyu.fi/santanen/info/kirjoittamisesta.html>>, 23.8.2000.

Santanen Jukka-Pekka, “Tietotekniikan Sovellusprojektien ohje”, University of Jyväskylä, Department of Mathematical Information Technology, available in HTML form at <URL: <http://www.mit.jyu.fi/opetus/sovellusprojektit/projohje.html>>, 25.9.2002.

Stevens W. Richard, “Unix Network Programming”, Prentice Hall Software Series, 1990.

Stroustrup Bjarne, “C++-ohjelmointi”, Teknolit, July 2000.

The Apache Software Foundation, “The Apache XML Project”, available in HTML form at <URL: <http://xml.apache.org/>>, read on 18.12.2002.

The JFreeChart project, “JFreeChart”, available in HTML form at <URL: <http://www.object-refinery.com/jfreechart/>>, read on 21.10.2002.

TopXML, “XSLT & XPath Tutorial”, available in HTML form at <URL: <http://www.vbxml.com/xsl/tutorials/intro/default.asp>>, read on 16.12.2002.

Walsh Norman, “What is XML?”, O’Reilly & Associates Inc., available in HTML form at <URL: <http://www.xml.com/pub/a/98/10/guide1.html>>, October 3rd, 1998.

World Wide Web Consortium, “The Extensible Stylesheet Language”, available in HTML form at <URL: <http://www.w3.org/Style/XSL/>>, read on 15.10.2002.

World Wide Web Consortium, “XML Path Language (XPath)”, available in HTML form at <URL: <http://www.w3.org/TR/xpath>>, read on 16.12.2002.

World Wide Web Consortium, “XSL Transformations (XSLT) Version 2.0”, available in HTML form at <URL: <http://www.w3.org/TR/xslt20/>>, November 15th, 2002.

Vesterholm Mika and Kyppö Jorma, “Java-ohjelmointi”, Talentum Media Oy, 2001.

XMLFiles.com, “XSL”, available in HTML form at <URL: <http://xmlfiles.com/xsl/>>, read on 16.12.2002.