
Abschlußbericht



Im Rahmen der Veranstaltung:
Dezentrale Systementwicklung am Beispiel GNU/LINUX
an der TU-Berlin – Sommersemester 2002



MICHAEL GLÄSSEL, MATTHIAS ERCHE, JÖRG KÜSTER
Betreuung: STEFFEN EVERS

26. September 2002

Zusammenfassung blah ...

Inhaltsverzeichnis

1	Einleitung	4
1.1	Der Projektkontext	4
1.1.1	Dezentrale Systementwicklung am Beispiel GNU/Linux	4
1.1.2	Das Projekt Mumie	5
1.2	Ziele	5
2	Organisation des Gesamtprojekts	8
2.1	Meilensteine	8
2.2	Workshops	8
2.3	Gruppenaufteilung	8
3	Muminav – Das Navigationstools	10
3.1	Lizenz	10
3.2	Entwicklungsumgebung	11
3.2.1	Entwicklungswerkzeuge	11
3.2.2	Projekthoster	12
3.3	Probleme und ihre Lösungen	12
3.4	Verwendete Komponenten und Ressourcen	12
3.5	Programm Aufbau	12
4	Beurteilung des Projekts	15
4.1	Diskussion des Status quo	15
4.2	Einsatzfähigkeit	15
4.3	Ausblick	15

1 Einleitung

1.1 Der Projektkontext

Die Arbeit am Projekt Muminav begann im Mai 2002 im Rahmen einer Lehrveranstaltung an der TU-Berlin¹. Es handelt sich dabei um die Realisierung eines Teils des ebenfalls an der TU-Berlin laufenden Projektes „Multimediale Mathematikausbildung für Ingenieure“² [Mumie].

1.1.1 Dezentrale Systementwicklung am Beispiel GNU/Linux

Die Lehrveranstaltung „Dezentrale Systementwicklung am Beispiel GNU/Linux“ [DESE], die am Institut für Elektrotechnik und Informatik³ angeboten wird, soll den Teilnehmern einen Einblick in die dezentrale Entwicklung von Softwaresystemen verschaffen. Der Schwerpunkt liegt auf der Betrachtung von Open Source Software.

Der theoretische Teil umfasst Themen wie die Geschichte und Philosophie von Open Source Software (im Speziellen GNU/Linux), Parallelen zwischen Wissenschafts- und Entwicklungstheorien zum Ent- und Bestehen von freien Softwaresystemen, die Organisation und Eigenheiten entsprechender Projekte und die damit verbundenen Anforderungen an die Entwickler, Hilfsmittel, die die dezentrale Entwicklung unterstützen und einige andere mehr.

Für den praktischen Teil bearbeiten die Teilnehmer ein Projekt, um selbst Erfahrungen auf dem Gebiet der dezentralen Systementwicklung zu sammeln. Es werden vom Veranstalter Projektphasen vorgegeben, zu denen Vorträge und Ausarbeitungen den Stand der Entwicklung widerspiegeln sollen. Die Entwicklung soll durch die Nutzung von in diesem Bereich bekannten Tools (z.B. Mailinglisten, CVS, Homepage etc.) unterstützt werden und es wird die Wiederverbenutzung schon bestehender Software angeregt. Ziel ist es, zum Abschluss der Veranstaltung einen Prototypen des Systems vorzustellen.

Die Inhalte der Projekte stehen den Teilnehmern völlig frei. Sie suchen sich selbst ein Thema oder arbeiten mit einem externen Auftraggeber zusammen. Dafür werden auch einige Kontakte vom Veranstalter hergestellt und angeboten.

¹<http://www.tu-berlin.de>

²kurz: MUMIE

³<http://cs.tu-berlin.de>

1.1.2 Das Projekt Mumie

„Multimediale Mathematikausbildung für Ingenieure“ ist ein vom Bundesministerium für Bildung und Forschung gefördertes Projekt, dass an der TU-Berlin in Kooperation mit 3 weiteren deutschen Universitäten entwickelt wird.

Das Ziel liegt in der Entwicklung einer WWW-basierten, modularen Umgebung, die einerseits dem Lernenden den Zugang zur Mathematik und andererseits dem Dozenten die Vermittlung von Wissen durch den Einsatz moderner, multimedialer Techniken erleichtern soll. Das System soll aus mehreren Modulen bestehen:

- Darstellung mathematischer Inhalte mit interaktiver Multimedia-Unterstützung
- Stoffnachbereitung, Wiederholungsunterstützung angeleitete und kommentierte Übungsaufgaben
- Selbstkontrolle durch individuelle Testumgebungen
- Einführung in mathematische Standardsoftwarepakete
- individuelles Trainingscenter weiterführender Inhalte
- Informationsplattform in der und vile und in die der es auch noch kein wer, was, wie, wo
- Kommunikations- und Austauschangebote

Zusätzlich werden Schwerpunkte auf eine individualisierbare Oberfläche und eine intelligente Benutzerführung gesetzt, um den Benutzern einen einfachen Umgang mit dem Stoff zu ermöglichen.

1.2 Ziele

Ein Dozent soll die Möglichkeit haben, aus einer Auswahl an Elementen ⁴, die in einer Datenbank gespeichert sind, die Zusammenstellung für einen Kurs zu erstellen. Dies soll durch einfaches Ziehen und Ablegen von Elementsymbolen auf einer Zeichenfläche erfolgen. Zusammenhänge zwischen Elementen werden dabei durch Verbindungslinien dargestellt, die der Verfasser positionieren kann. Zusätzlich kann es zu den Elementen Subelemente ⁵ geben. Es entsteht durch die Zusammenstellung eine Repräsentation der mathematischen Zusammenhänge mittels eines gerichteten Graphen. Der Dozent kann zur Darstellung des Kursverlaufes eine Art „roten Faden“ festlegen, der allerdings nicht entlang der angelegten Verbindungen laufen muss ⁶.

⁴Motivation, Definition, Theorem, Lemma, Algorithmus, Anwendung

⁵Herleitung, Beweis, Motivation, Bemerkung, Historisches, Visualisierung, Beispiel, Tabelle

⁶Aktueller Stand, Änderungen möglich

Die Aufgabe für das Projekt MUMINAV besteht darin, dieses vorgegebene Netz mit Hilfe eines Java-Applets in einem Navigationsframe darzustellen und somit den Zugriff auf die Inhalte zu ermöglichen. Die Darstellung der Elemente erfolgt durch unterschiedlich farbige, leicht dreidimensional angedeutete Kästen. Subelemente werden nach vorgeschriebenen Regeln an diesen Kästen angeordnet und sind standardmäßig nicht mit Unterscheidungsmerkmalen versehen.

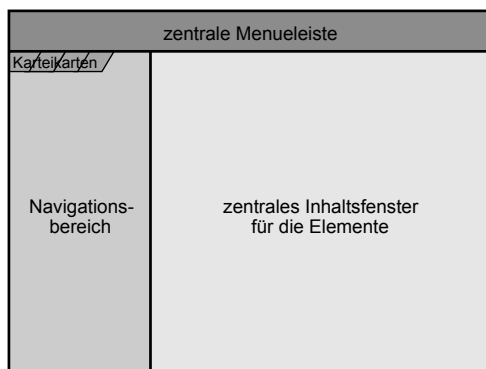


Abbildung 1.1: Layout der Hauptansicht

Die Abbildung des logischen Netzes geschieht, wie vorher beschrieben, durch Verbindungslinien zwischen den Elementensymbolen. Es wird ein „roter Faden“ entsprechend der linearen Anordnung der Kursinhalte gelegt.

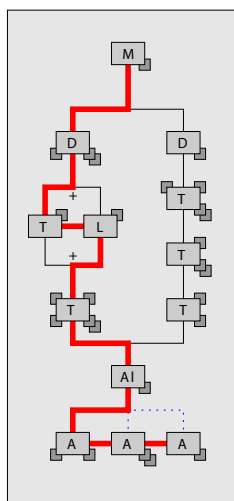


Abbildung 1.2: Kurspfade und roter Faden

Die Darstellung soll auf mehrere Maus-Aktionen reagieren. Wird die Maus über ein Element bewegt, soll dieses vergrößert dargestellt werden und die Subelemente werden

unterscheidbar durch Beschriftung. Bei Klick mit der linken Maustaste soll der Inhalt des jeweiligen (Sub-)Elementes im zentralen Inhaltsfenster dargestellt werden. Dieselbe Aktion auf der mittleren Maustaste führt zu einem Öffnen des Inhalts in einem externen Fenster und schließlich ist es zukünftig vorgesehen, mit der rechten Maustaste eine Liste von Optionen anzubieten.

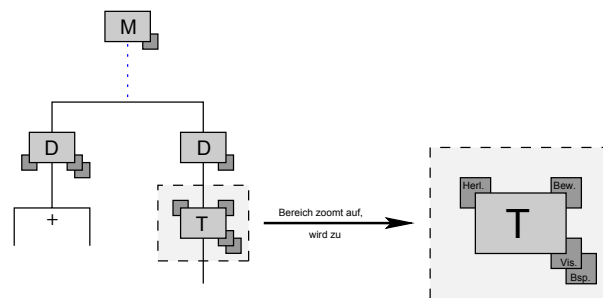


Abbildung 1.3: Detailansicht bei Zoom

2 Organisation des Gesamtprojekts

An MUMINAV haben im Rahmen der Veranstaltung *Dezentrale Systementwicklung am Beispiel GNU/LINUX* mitgewirkt: Jörg Küster, Matthias Erche und Michael Glässel. Alle, Studenten der TU-Berlin im Fach Informatik.

2.1 Meilensteine

Die Termin- und Sachzielverfolgung wurde mit Hilfe von 4 Meilensteinen sichergestellt.

- **8. Mai** Festlegung auf eine Projektaufgabe
- **29. Mai** Bestandsaufnahme von relevanten Projekten
- **19. Juni** Vorlage des Lösungskonzept
- **1. Oktober** Zwischenbilanz und Prototyp

2.2 Workshops

Beim Workshop, der am 10. Juli stattfand, ging es um die Nachbearbeitung, Analyse und Ergänzung der gesammelten Erfahrungen. Dazu wurden Gastredner eingeladen, die sich seit längerer Zeit in diesem Bereich betätigen und die mit ihrem Wissen dazu beitragen können, die persönlichen Erfahrungen der Teilnehmer in einem allgemeineren und objektiverem Zusammenhang zu sehen.

2.3 Gruppenaufteilung

Die Entwicklungs- und Implementierungsarbeit hat sich wie folgt aufgeteilt:

- **Matthias Erche** Entwerfen des Designs und der Inhalte der Projekt-Homepage, XML-Treeparser, Datenaustauschformat
- **Jörg Küster** Zoomfunktionalität, Dokumentation und Skins

- **Michael Glässel** Entwicklung des Grundgerüsts, Mouseoververhalten und Tool-tips

Zu jedem der ersten drei Meilensteine hat ein Gruppenmitglied einen Vortrag gehalten.

3 Muminav – Das Navigationstools

Beispiel Programm, welches zeigt wie man Muminav einbinden kann.
relevante andere Projekte - hoch spezifizierte Anforderungen

3.1 Lizenz

Eine Vorgabe für die Entwicklung war, die gesamte Arbeit als Open Source Projekt durchzuführen. Dies schließt ein, die Projektergebnisse unter einer Open Source Lizenz [OSI2002] zu veröffentlichen.

Zur Diskussion standen zwei Lizenzen. Die GPL [GPL1991] (GNU General Public License), als eine der meistverwendeten Lizenzen im Open Source Bereich, sieht vor, eine Software für jeden frei verfügbar, nutzbar und veränderbar zu machen. Hinzu kommt der Aspekt, dass Programme, die GPL lizenzierte Software benutzen (linken) oder die durch die Weiterentwicklung dieser entstehen, wieder unter der GPL stehen müssen. So wird verhindert, dass man von der Arbeit anderer profitiert, ohne eine Gegenleistung zu erbringen.

Eine Lockerung dieser Politik sieht die LGPL [LGPL1999] (Lesser General Public License oder auch Library General Public License) vor. Ursprünglich eingeführt, um das Linken von frei verfügbaren Bibliotheken aus proprietären Systemen heraus zu ermöglichen, ohne Einfluss auf die ursprüngliche Lizenz zu nehmen, wird sie heutzutage überall dort eingesetzt, wo freie Software auch in kommerziellen und nicht freien Umgebungen eingesetzt wird.

Da der Ursprung von MUMINAV in der Entwicklung eines Navigationssystems für die modulare Lernumgebung MUMIE liegt, musste bei der Wahl der Lizenz darauf Rücksicht genommen werden. Zum Entstehungszeitpunkt dieses Dokumentes wird zwar für die MUMIE ebenfalls eine Open Source Lizenz vorgesehen, doch ist in Zukunft ein kommerzieller Einsatz nicht ausgeschlossen.

Um eventuelle Vorhaben in dieser Richtung nicht zu verhindern, haben wir uns deshalb dazu entschlossen, MUMINAV unter die LGPL zu stellen und somit die freie Verfüg- und Benutzbarkeit zu sichern und trotzdem den Entwicklern der MUMIE alle Möglichkeiten offen zu lassen.

3.2 Entwicklungsumgebung

3.2.1 Entwicklungswerkzeuge

Für das Projekt wurden eine Reihe von Entwicklungswerkzeugen und Technologien verwendet, welche für Open-Source-Projekte charakteristisch sind:

- **Mailinglisten** um mit den Entwicklern und allen, die sonst noch Interesse an dem Projekt haben zu kommunizieren. Wobei damit auch gleichzeitig eine Dokumentation des Projektverlaufs über die Mailinglisten-Archive entsteht.
- **CVS** Concurrent Version System [FOGEL2000]. Hat uns ermöglicht dezentral an den selben Quelldateien zu arbeiten. Über die CVS-Log Einträge lässt sich auch nachträglich die Entwicklungs-Historie nachvollziehen und stellt somit auch zugleich eine zusätzliche Dokumentation des Projekts dar [?].
- **eMail** Als standard-Medium zum direkten persönlichen Informationsaustausch.
- **Instant-Messaging** hat bei Arbeiten, die ein hohes Maß an Absprachen bedürfen, nicht die Nachteile, die ein asynchrones Medium wie eMail und Mailinglisten haben.
- **Projekt-Homepage** Unter der URL: <http://muminav.berlios.de> haben wir eine Projekt-Homepage angelegt, die die Öffentlichkeit und die Teilnehmer des Projekts mit allgemeinen Informationen versorgt.
- **Newsgroups** Hier haben wir uns Anregungen und Informationen für die Planung des Projekts und bei Problemen, die in der täglichen Arbeit aufraten besorgt.

Für die Entwicklung der Quellen haben wir folgende Software eingesetzt.

- **JavaTM 2 Platform, Standard Edition**
- **Apache Ant** des JAKARTA PROJEKT von Apache, als make-Tool [ANT2002].

Leider ist der Javacompiler, den wir eingesetzt haben nicht als Open-Source veröffentlicht worden. Aus Kompatibilitätsgründen mussten wir uns jedoch für diesen Compiler entscheiden, damit das Applet unter möglichst unterschiedlichen Umgebungen läuft ohne, dass der Benutzer weitere Software installieren muss.

Als Alternativen zum Java-Compiler von Sun wäre im Open-Source-Bereich z. B. JIKES von IBM [JIKES], der an der Universität von Boston entwickelte ESPRESSO [ESPR] oder der GCJ der FREE SOFTWARE FOUNDATION [GJC], denkbar. Da sich aber jeder bei uns die Java Quellcodes herunterladen kann, bleibt es jedem selbst überlassen, welchen Compiler er einsetzt [J2SE] .

3.2.2 Projekthoster

Bei der Wahl des Projekthosters kam für uns SOURCEFORGE *nicht* in Frage, da sich im Laufe der Zeit die Lizenz-Politik [SFORGE] des Hosters immer mehr gegen die Ideale der Open-Source-Gemeinde richten. Das SAVANNAH Projekt der Free Software Foundation kam nicht in Frage, da die Aufnahme eines Projektes strengen Restriktionen unterliegt, die es zum Beispiel schwer machen, eine Software, die mit dem Java-Compiler von Sun entwickelt wird, bei diesem Hoster anzumelden.

Aus diesen Gründen haben wir uns für den Berliner Projekthoster BERLIOS [BerliOS] entschieden, der uns mit den meisten Diensten unterstützen konnte.

3.3 Probleme und ihre Lösungen

XML-Parser mitliefern - Größe - Entscheidung für Java 1.4 und Mozilla

3.4 Verwendete Komponenten und Ressourcen

3.5 Programm Aufbau

Muminav ist ein Java Applet und besteht aus folgenden Klassen:

Muminav	die Applet-Klasse (JApplet)
MuminavPanel	das Panel auf dem gezeichnet wird. (JPanel)
MuminavToolTipManager	steuert das Verhalten der Tooltips.
TreeBuilder	liest die XML-Dateien und erzeugt einen Baum mit den Elementen aus der Datei
Part	eine Abstrakte Klasse von der Jedes Element eines Skins abgeleitet wird
NavNet	
DrawLib	enthält diverse Zeichenmethoden
UnknownElement	
Connector	Verbindungselement aus dem Beispiel-skin des Mathe-Thema
Mainelement	Hauptelement aus dem BeispielSkin des Mathe-Themas

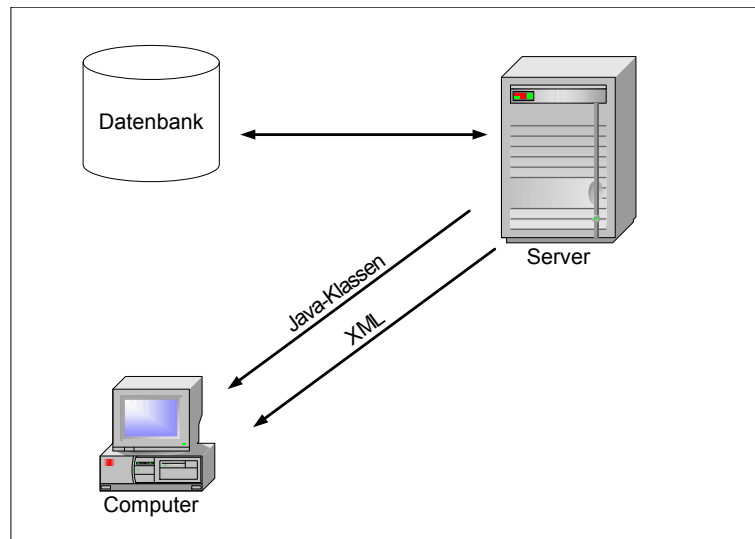


Abbildung 3.1: Kommunikation: Informationsfluss

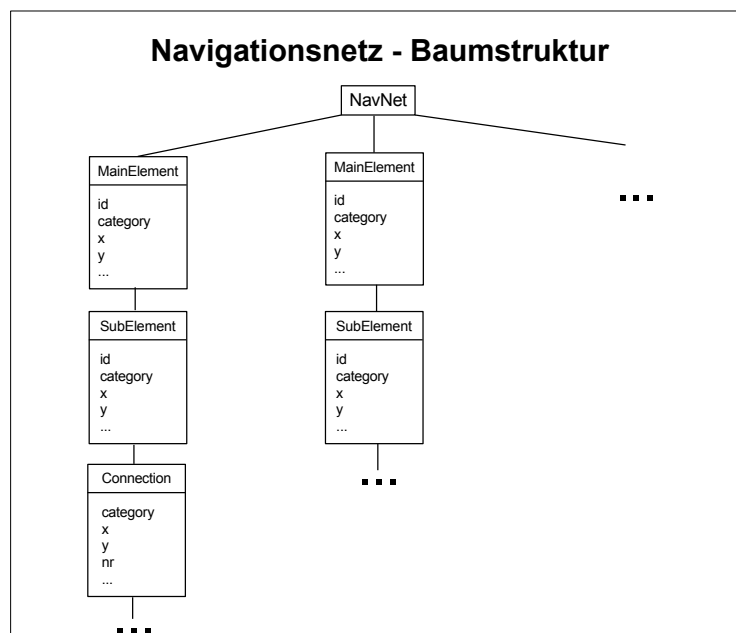


Abbildung 3.2: Baumstruktur

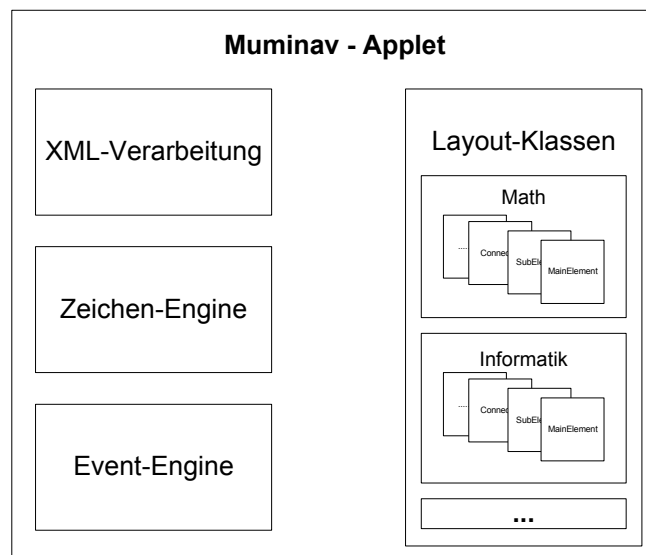


Abbildung 3.3: Aufbau: Hauptteile von Muminav

4 Beurteilung des Projekts

4.1 Diskussion des Status quo

4.2 Einsatzfähigkeit

MUMINAV wurde als hoch spezialisierte Softwarekomponente für den Einsatz im Gesamtprojekt MUMIE ins Leben gerufen. Nichts desto trotz haben wir großen Wert darauf gelegt, die Software so allgemein wie möglich zu gestalten damit sie für möglichst viele andere Open-Source-Projekte interessant wird. Letztendlich wurde, durch den Einsatz von Skins und XML als Datenformat, aus der Aufgabe, ein Java-Applet zu programmieren, welches Navigationsnetze für Mathematische Kurse im Internet dynamisch erzeugt, ein Softwaremodul, welches für die Darstellung jeglicher graphischer Darstellungen geeignet ist. Zum Beispiel ließen sich leicht Skins erstellen, mit denen man Flowcharts, E/R Diagramme oder sogar UML darstellen kann.

Zusätzlich ist die Software so ausgelegt, dass sie nicht auf Applets beschränkt ist. Durch das Packaging in ein Java-Swing-Panel lässt es sich genau so gut in eine Java-application einbauen.

4.3 Ausblick

Für die Zukunft des Projekts haben wir uns vorgestellt, den Abstraktionsgrad weiter zu steigern, damit es immer leichter wird, seine eigenen Skins zu entwerfen.

Angedacht ist auch ein einfacher Skin-Editor, mit dem man sich bequem per Maus seine eigenen Skins entwerfen kann.

Interessant wäre es auch, unsere Quelle, die wir bisher auf der Java-Plattform von SUN entwickelt und getestet haben, auf die Kompatibilität zu Open-Source-Kompilern zu testen und gegebenenfalls anzupassen. Damit könnten wir dann auch den Bytecode unter einer entsprechenden Lizenz veröffentlichen.

Wir werden auch versuchen eine kleine Community ins Leben zu rufen, welche gegenseitig Skins austauscht. Dabei könnte man die MUMINAV-Homepage zu einem öffentlich zugänglichen Skin-Archiv erweitern, in das jeder seine Skin einbringen und bei Bedarf von der Kreativität anderer profitieren kann.

Was eigentlich gar nicht erwähnt werden müsste: MUMINAV, wie auch jedes andere Open-Source-Projekt, ist mit der Fertigstellung der ersten Version nicht abgeschlossen. Vielmehr wird es, solange es aus der Open-Source-Community Interesse und Anregungen gibt, sich weiterentwickeln. Sei es durch entdeckte Bugs oder Vorschläge für neue Features, die durch uns oder Mitglieder aus der Open-Source-Community in das Projekt einfließen.

Literaturverzeichnis

- [Mumie] DOMINIK EBERLEIN, ANDREAS HOFFMANN: Die Mumie - Projekthomepage. <http://www.mumie.org/>
- [DESE] STEFFEN EVERS: Dezentrale Systementwicklung am Beispiel GNU/LINUX. <http://flp.cs.tu-berlin.de/dese>
- [OSI2002] OPEN SOURCE INITIATIVE: The Approved Licenses. <http://www.opensource.org/licenses/> 2002
- [GPL1991] FREE SOFTWARE FOUNDATION, INC: The GNU General Public License (GPL). <http://www.opensource.org/licenses/gpl-license.php> 1991, version 2,
- [LGPL1999] FREE SOFTWARE FOUNDATION, INC: GNU Lesser Public License. <http://www.opensource.org/licenses/lgpl-license.php> 1999, version 2.1,
- [FOGEL2000] KARL FOGEL: Open Source Development with CVS. <http://cvsbook.red-bean.com/> 2000, version 1.2
- [EVERS] STEFFEN EVERS: 2002
- [ANT2002] APACHE SOFTWARE FOUNDATION: Apache Ant 1.5 Manual. <http://jakarta.apache.org/ant/manual/index.html> 2002, version 1.15.2.2
- [JIKES] IBM: Jikes Official Homepage. <http://oss.software.ibm.com/developerworks/opensource/jikes/> 2002
- [ESPR] PROF A J KFOURY: Java Espresso. <http://types.bu.edu/Espresso/JavaEspresso.html> 2002
- [GJC] FREE SOFTWARE FOUNDATION, INC: The GNU Compiler for the Java Programming Language. <http://gcc.gnu.org/java/> 2002
- [J2SE] PROF A J KFOURY: Java Espresso. <http://types.bu.edu/Espresso/JavaEspresso.html> 2002

- [SFORGE] SOURCEFORGE: Terms and Conditions of Use. http://sourceforge.net/docman/display_doc.php?docid=6048&group_id=1 2002
- [BerliOS] BERLIOS: BerliOS Site Documentation. <http://developer.berlios.de/docs/site/> 2002