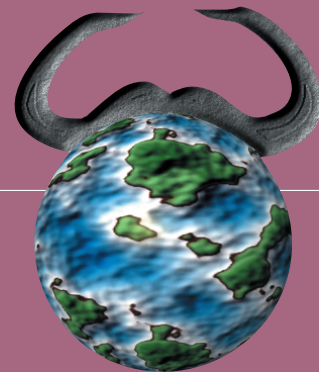


Brave GNU World



Diese Kolumne berichtet aus der Perspektive von GNU-Projekt und FSF über Projekte und aktuelle Geschehnisse im Umfeld freier Software und versucht Einblicke in die zugrunde liegende Philosophie zu vermitteln. In dieser Ausgabe: das Projekt Hyrex und Möglichkeiten, freie Softwareprojekte zu unterstützen. Georg C. F. Greve



Abbildung 1: Wer einen Bug fängt, kann freien Software-Projekten mit detaillierten Reports helfen.

Willkommen zu einer neuen Ausgabe der Brave GNU World, die diesmal an die Vorstellung von Doodle im letzten Monat anknüpft. Auch in dieser Ausgabe also wieder ein Projekt für die Katalogisierung von und Suche nach Informationen. Die Suche und Gewinnung von Informationen, auf Englisch Information Retrieval (IR) genannt, ist nicht auf Suchmaschinen beschränkt. Diese sind nur ein Spezialfall.

Grundsätzlich geht es um das Gewinnen von Informationen aus Dokumenten. Vor etwa 30 Jahren, als dieses Feld aufkam, verstand man unter Dokumenten im Allgemeinen Texte: Zeitungsartikel, Bücher oder wissenschaftliche Publikationen, mittlerweile zählen dazu Audio-dateien, Bild- und Videodokumente und Kombinationen daraus.

Man spricht deshalb auch von Audio-IR, Music-IR oder Image Retrieval. Das Projekt Hyrex [5] (Hypertext Retrieval En-

gine for XML) bewegt sich auf der Trennlinie zwischen Datenbank und Informationsgewinnungs-System. Es erlaubt die Gewinnung von Informationen aus XML-Dokumenten.

Information Retrieval aus XML-Dokumenten

XML wurde deshalb gewählt, weil es erlaubt, Dokumente als Mischung aus Fakten und Prosa zu präsentieren. Eine Sammlung von XML-Dokumenten zur Beschreibung von Büchern kann Fakten wie den Herausgeber, den Autor oder den Preis enthalten, aber auch den gesamte Text des Buches. Eine typische Anfrage wäre dann beispielsweise die Recherche nach günstigen Büchern zu Multimedia-Datenbanken.

Die Baumstruktur von XML erlaubt es, ein Buch durch Kapitel, die Kapitel durch Sektionen und diese wiederum durch Untersektionen zu repräsentieren. Doch auch die Mischung sehr unterschiedlicher Inhalte ist innerhalb dieser Struktur möglich. Hyrex implementiert dabei eine Sprache zur Formulierung von Anfragen, die XPath ähnelt. Allerdings erweitert Hyrex XPath in Bezug auf Unterstützung von Unschärfe und Unsicherheit sowie für unterschiedliche Datentypen. Grundsätzlich ist Hyrex sinnvoll für alle Fälle, in denen Dokumente durchsucht werden werden sollen, die sich in XML abbilden lassen – ein Modul zur Konvertierung von E-Mail ist bereits dabei.

Der Stand der Entwicklung setzt dem Einsatz allerdings zurzeit noch deutliche Grenzen. Die Autoren Norbert Govert und Kai Grossjohann haben Hyrex mit Unterstützung einiger Studenten als

Doktorarbeit an der Universität Dortmund entwickelt. Das Projekt begann im Jahr 2001 und ist noch nicht abgeschlossen. Geschrieben ist Hyrex hauptsächlich in Perl, Zeit-kritische Teile sind in C implementiert. Kai Grossjohann ist gerade dabei, mit gelegentlicher Unterstützung das Projekt aufzuräumen und schneller zu machen. In der Forschungsphase waren den Entwicklern Konzepte wichtiger als Performance.

Hyrex lässt sich also derzeit noch nicht schnell und unkompliziert verwenden, die gesamte Basisfunktionalität ist aber bereits implementiert. Konzeptionell gehört es zu den modernsten Ansätzen – wenn ein sehr aktuelles Projekt reizt, ist herzlich dazu eingeladen, sich mit Kai über [5] in Verbindung zu setzen.

Anstehende Aufgaben sind die Bereinigung und Konsolidierung des Codes, speziell in Bezug auf Geschwindigkeit und Speichernutzung, die Erweiterung für andere Datentypen über Module und die Vereinfachung der Installation. Und natürlich ist auch die Dokumentation des Programms für normale Nutzer noch nicht umfangreich und verständlich genug. Es gibt also viel zu tun.

Freie Software braucht Unterstützung

Der Leser Henryk Gerlach hatte eine Menge interessanter Fragen. Er wollte unter anderem wissen, wie Einzelne freie Software sinnvoll materiell unterstützen können. Die Antwort soll aber für einen Moment zurückgestellt werden, da es viele Möglichkeiten der unmittelbaren immateriellen Unterstützung gibt. Wer mit proprietärer Software aufgewachsen ist, tut sich oft schwer, aus

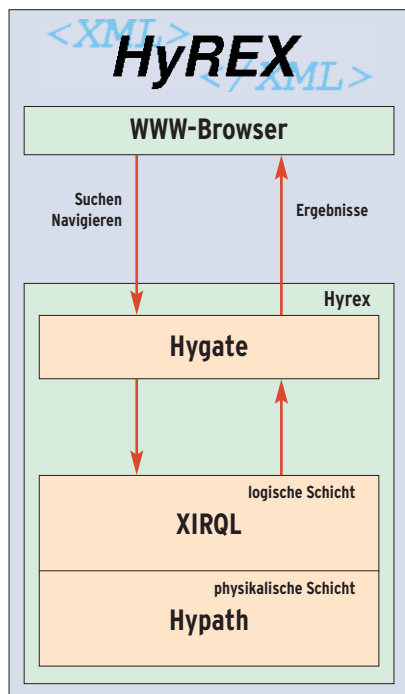


Abbildung 2: Das Hyrex-Projekt ist ein ambitionierter Weg des Information Retrieval per XML.

der anerzogenen Passivität auszubrechen und zu erkennen, dass er selbst aktiv werden kann. Selbst wenn die Erkenntnis vorhanden ist, fehlt zum Teil das Wissen darum, was zu tun ist und wie man es anstellt.

Auf den ersten Blick scheint es, dass Programmieren der Weg ist, um freie Software voranzubringen. Das ist natürlich essenziell. Aber es kommt nicht auf die Anzahl von Zeilen Programmcode an, wichtiger sind Qualität und Benutzbarkeit. Gerade hier kann freie Software die Nase vorn haben, auch wegen der Mitwirkung von Menschen, die selbst keine Programme entwickeln.

Mehr Sprachen

Da ist zum einen die Anpassung an andere Sprachen – bei freier Software selten ein technisches Problem, meist kommt hier das in Ausgabe 36 der Brave GNU World [6] vorgestellte GNU-Programm »gettext« [7] zum Einsatz. Es dient dazu, Dateien mit Textfragmenten entsprechend anzupassen und zu aktualisieren. Nach einer kurzen Einarbeitungsphase ist dies auch ohne größere Computerkenntnisse möglich. Wer hier aktiv werden möchte, kann es zumeist selbstständig tun und schickt

später nur die Datei an den Autor des Programms, der sich normalerweise sehr darüber freuen wird. Besser ist es jedoch, vorher kurz Kontakt mit dem Autor aufzunehmen und ihn über das Vorhaben zu informieren, um Mehrfacharbeit zu vermeiden.

Dokumentation und Feedback

Das Schreiben und Aktualisieren von Dokumentationen erfordert meist mehr Kontakt mit den Entwicklern und bedeutet auch mehr Arbeit, ist jedoch extrem wertvoll. Die meisten Entwickler schreiben ungern Dokumentation und unterschätzen oft deren Bedeutung. Eine gute Dokumentation kann aber enorm zur Verbreitung eines Programms beitragen. Hilfe dabei erlaubt es also nicht nur den Entwicklern, sich auf die Entwicklung zu konzentrieren, sie macht ihre Arbeit auch mehr Menschen verfügbar.

Eine andere wichtige Möglichkeit, die freie Software-Entwicklung voran zu bringen, ist Feedback. Entwickler sitzen zumeist nicht neben jedem Anwender und können daher nur versuchen die Programme so zu entwerfen, wie sie ihnen schlüssig erscheinen. Dabei geschieht es, dass Programme scheinbar nicht nur von, sondern auch für Entwickler geschrieben werden. Hier ist es wichtig, vernünftiges Feedback zu geben, was aber nicht bedeutet, sich nur undifferenziert zu beschweren.

Sachliche Kooperation der Beteiligten

Auch wer beispielsweise eine Linux-Distribution kauft, ist nur gegenüber dem Distributor Kunde, von dem er die GNU/Linux-Distribution bezogen hat. Die Entwickler der Software werden aber in den meisten Fällen nicht vom Distributor bezahlt. Bei dem mag seinen Unmut abladen, wer Anlass dafür hat, er verbessert die Situation damit aber meist nicht – zumal leider oft völlig falsche Personen oder Organisationen als Adressaten herhalten müssen.

Es gehört zur Kultur freier Software, dem anderen eine gewisse Grundintelligenz zuzuerkennen. Ebenso gehört der konstruktive Vorschlag zum guten Ton: An-

statt sich also beispielsweise über die Anordnung der Fenster zu beschweren, ist ein konstruktiver Gegenvorschlag sinnvoller. Da der Anwender zumeist nicht Kunde des Entwicklers ist, ist er auch nicht dessen König. Beide können aber zum gegenseitigen Nutzen miteinander kooperieren.

Ganz besonders gilt dies für Problem- und Fehlermeldungen. Die Qualität einer Fehlermeldung kann den Unterschied zwischen nahezu unmittelbarer Behebung des Fehlers und jahrelanger Suche bedeuten. Doch was macht eine gute Fehlermeldung aus? Sie ist erstens so konkret und spezifisch wie möglich. Sie beschreibt möglichst genau, unter welchen Umständen welcher Fehler auftritt. Sie informiert außerdem, ob sich das Problem reproduzieren lässt und ob es beispielsweise immer oder nur sporadisch auftritt.

Ebenso wichtig ist es, den Fehler einzugrenzen. Wann tritt das Problem nicht mehr auf? Wo scheint die Grenze zu liegen? Und dann sollte ein Blick auf die Webseite und in die Archive der Mailingliste nicht fehlen, um zu sehen, ob das Problem bereits bekannt ist.

Oft ist also für eine gute Fehlermeldung eine gründliche Experimentierphase nötig. Im Anschluss daran sollte der Entdecker des Fehlers das Problem möglichst vollständig beschreiben und dabei die Umstände, also Rechner, Betriebssystem, Distribution und vor allem die betroffenen Versionen klar angeben. Das Angebot, bei Bedarf zusätzliche Informationen zu liefern, rundet die gute Fehlermeldung ab. Wer jetzt denkt, dass dies nach ziemlich viel Arbeit klingt, hat es verstanden. Gute Fehlermeldungen schreiben ist Arbeit, und zwar eine sehr nützliche Arbeit.

Zeit für gute Fehlermeldungen

Die Fähigkeit, gute Fehlermeldungen zu schreiben, wird oft nicht hoch genug eingeschätzt. Sie entwickelt sich wie alle Fähigkeiten durch Übung. Angst vor Fehlern beim Melden von Fehlern ist dabei fehl am Platze. Wer sich ehrlich bemüht, wird meist mit freundlicher Aufnahme rechnen und im Dialog sehr viel lernen können. Für das technische Ge-

lingen eines Projekts ist die Rückmeldung von Fehlern ebenso wichtig wie das Entwickeln selbst. Daher lohnt es sich, auch einmal relativ junge Projekte auszuprobieren, denn die Rückmeldung kann dann einen deutlich spürbaren Einfluss auf die Entwicklungsgeschwindigkeit haben. Übrigens bedeutet das für die Projekte im Umkehrschluss, dass sie gutes Feedback ähnlich hoch schätzen sollten wie guten Code.

Weitere Möglichkeiten der Unterstützung bieten sich in der Hilfestellung für andere, sei es durch die Beantwortung von Fragen auf Mailinglisten, in Onlineforen oder im realen Leben. Das entlastet jene, die aktiv an der Entwicklung teilnehmen – Entwickler, Autoren von Dokumentation, Übersetzer, Tester –, und sorgt zudem dafür, dass mehr Menschen sich in Zukunft auch selber aktiv beteiligen werden.

Als Autor der Brave GNU World würde ich mich auch sehr darüber freuen, wenn mehr Leser gerade Entwicklerinnen dazu ermuntern würden, eigene Projekte der Brave GNU World zu melden [1], oder selbst aktiv werden und die Projekte vorschlagen. In der Vergangenheit hat sich dies zum Teil als sehr effektiv erwiesen, um Projekte einer größeren Öffentlichkeit zu präsentieren.

Materielle Unterstützung

Zusätzlich oder alternativ zu diesen Aktivitäten kann man freie Software natürlich auch materiell unterstützen – was sich, wie in Brave GNU World #46 beschrieben, auch als Bereitstellung von „gefrorener Zeit“ in Form von Geld betrachten lässt [8]. Damit ist der Bogen zurück zu der Frage von Henryk Gerlach geschlagen, den im Wesentlichen zwei Dinge bewegen: Die schon fast unüberschaubare Zahl an Möglichkeiten, mit Spenden Gutes zu tun, und die oft mangelhafte Transparenz.

Die Möglichkeiten beginnen mit Händlern, die häufig Produkte anbieten, bei denen ein Teil des Verkaufspreises als Spende an bestimmte Projekte geht, und reichen über direkte Spenden an Projekte bis hin zu Spenden an Organisationen wie die FSF.

Bei den Spenden über Händler gibt es in der Tat häufig wenig Transparenz und

Überblick, ob das Geld jemals den gewünschten Empfänger erreicht – da im Zweifelsfall nur der Empfänger selbst darüber informieren kann. Henryk fordert hier Transparenz – eine Forderung, die vom Autor dieser Kolumne voll und ganz unterschrieben wird. Die Free Software Foundation Europe (FSFE) [9] hat ihre gesamten Budgets seit Anbeginn offen gelegt [10] und unterhält eine Liste ihrer Spender [11].

So lässt sich grob feststellen, in welchem Ausmaß Spenden beispielsweise vom Online-Buchportal Bookzilla.de [12] tatsächlich an die FSFE geflossen sind. Auch wenn Transparenz zum Teil viel Arbeit bedeutet, kann eine ähnliche Vorgehensweise nur allen ans Herz gelegt werden. Potenziellen Spendern und Spenderinnen sei im Gegenzug empfohlen, im Zweifelsfall nach Transparenz, Glaubwürdigkeit und Vertrauen zu entscheiden, wer Spenden vermutlich verantwortungsvoll behandelt.

Bei direkten Spenden an einzelne Projekte sei den Empfängern dringend angeraten, zumindest halbwegs transparent darüber Buch zu führen, wie viel Geld von wo eingenommen und für welche Arten von Ausgaben es verwandt wurde. Niemals sollten Projekte und Organisationen aus dem Blick verlieren, dass sie mit Spendengeldern von Menschen und Unternehmen handeln, die auf einen sorgfältigen Umgang mit ihrer Spende vertrauen.

Die FSF in den USA veranstaltet jährliche Treffen, zu denen alle Teilnehmer

des Fundraising-Programms eingeladen sind. Dort berichten die Angestellten und der Vorstand der Free Software Foundation USA über ihre Aktivitäten. Dazu kommen noch die Mailings auf Papier und per E-Mail sowie eine Community-Seite [13].

Die europäische FSF informiert auf der Website über aktuelle Projekte [14] sowie seit kurzem mit einem monatlichen Newsletter [15], der die Zweijahresberichte [16] aktuell ergänzt. Zusätzlich plant die FSFE im Moment ein Fellowship-Programm ähnlich der Associate Membership der FSFUS.

Spenden-Dachverband?

Henryk schlug vor, im Rahmen der FSF(E) einen Dachverband für Spenden an freie Softwareprojekte zu schaffen. Das erscheint jedoch schwierig. Einerseits erfordert es eine Verwaltung, die ebenfalls durch die Spenden finanziert werden müsste, zum anderen müsste dieser Dachverband die Verwendung der Gelder in den einzelnen Projekten dann aufwändig überwachen. Es ist wahrscheinlich viel effektiver, wenn die einzelnen Projekte mehr Transparenz beim Sammeln und Verwenden ihrer Mittel walten lassen.

Wenn potenzielle Spender die gewünschten Informationen über den Einsatzzweck der Spendengelder nicht finden, sollten sie den Projekten unbedingt mitteilen, was nach ihrer Ansicht verbessert werden kann. Da aber die meis-



Abbildung 3: Manche Händler, beispielsweise das Online-Buchportal Bookzilla, geben einen Teil ihrer Einnahmen als Spenden an freie Projekte weiter.

ten Projekte bereits eine große Vorleistung gebracht haben, indem sie die Software geschrieben und zur Verfügung gestellt haben, hoffe ich allerdings, dass potenzielle Spender – obwohl sie stets kritisch sein sollten – zunächst eher mehr als weniger Vertrauen in die Projekte zeigen.

Aufgaben der FSFE langfristig orientiert

Es gibt noch einen zweiten Grund, warum die Idee eines Spenden-Dachverbands nicht sinnvoll ist. Die Arbeit der Free Software Foundation ist relativ speziell, sie ist unter anderem sehr viel langfristiger orientiert als bei anderen Organisationen oder Projekten. Diese langfristige politische, gesellschaftliche und juristische Arbeit ist für den Schutz und Ausbau der Welt freier Software essenziell, gleichzeitig jedoch abstrakt und schwer vermittelbar. Resultate zeigen sich oft erst nach Jahren.

Ein Beispiel ist die Arbeit im Umfeld des Kartellverfahrens gegen Microsoft [17]. Die FSFE ist bereits seit Jahren in dieses Verfahren einbezogen, die Kommission hat nach eigenem Bekunden große Teile ihres Urteils gegen Microsoft auf den Eingaben der FSFE aufgebaut, die zu diesem Zweck eng mit dem Samba-Projekt [18] kooperiert hat.

Nachdem Microsoft vor dem Europäischen Gerichtshof Einspruch eingelegt hat, ist die FSFE auf Seiten der Europäischen Kommission in den Prozess ein-

getreten [19], wobei sie hier offiziell auch im Namen des Samba-Projekts spricht. Ziel ist es, das Urteil der Kommission zu stärken und vor Gericht die notwendige Auslegung zum Schutz freier Software zu erreichen. Dieser Prozess wird vermutlich wesentliche Auswirkungen auf das Samba-Projekt haben. Neben Softwarepatenten stuft Samba diesen Fall als eine der größten Bedrohungen ein.

Der Prozess wird vermutlich mehrere Jahre dauern. Nur um sicherzustellen, dass die FSFE zum Prozess zugelassen wird, musste Carlo Piana, der zuständige Anwalt der FSFE, kurzfristig einen Tag nach Luxemburg fliegen. Obwohl er sich bereit erklärte, einen großen Teil der Arbeit pro bono zu leisten, sind also bereits jetzt sichtbare Kosten entstanden. Gleichzeitig lässt sich die Summe, die die FSFE bis zum heutigen Tag an Unterstützung durch die Vielzahl der Unternehmen erhalten hat, die Samba einsetzen oder vertreiben und somit unmittelbar vom Fall betroffen sind, relativ leicht einstellig ausdrücken.

Es ist also schwer, Unterstützung für diese Arbeit zu finden. Fast alle finden sie gut und sehr wichtig, aber unterstützen sollen sie doch bitte die anderen. Wer diese anderen sind, findet sich – wie bereits gesagt – online [11]. Würde die FSFE jetzt anfangen, Spenden für andere Projekte zu verwalten, bestünde die Gefahr, dass sie bald keine Ressourcen mehr für ihre eigentliche Arbeit zur Verfügung hat. (uwo/mhu) ■

Infos

- [1] Ideen, Anregungen, Kommentare an die Brave GNU World: [\[column@brave-gnu-world.org\]](mailto:column@brave-gnu-world.org)
- [2] GNU-Projekt: [\[http://www.gnu.org/\]](http://www.gnu.org/)
- [3] Homepage von Georg's Brave GNU World: [\[http://brave-gnu-world.org\]](http://brave-gnu-world.org)
- [4] „We run GNU“: [\[http://www.gnu.org/brave-gnu-world/rungnu/rungnu.de.html\]](http://www.gnu.org/brave-gnu-world/rungnu/rungnu.de.html)
- [5] Hyrex: [\[http://www.is.informatik.uni-duisburg.de/projects/hyrex/\]](http://www.is.informatik.uni-duisburg.de/projects/hyrex/)
- [6] Brave GNU World #36: [\[http://brave-gnu-world.org/issue-36.de.html\]](http://brave-gnu-world.org/issue-36.de.html)
- [7] GNU Gettext: [\[http://www.gnu.org/software/gettext/\]](http://www.gnu.org/software/gettext/)
- [8] Brave GNU World #46: [\[http://brave-gnu-world.org/issue-46.de.html\]](http://brave-gnu-world.org/issue-46.de.html)
- [9] Free Software Foundation Europe: [\[http://www.germany.fsfeurope.org/\]](http://www.germany.fsfeurope.org/)
- [10] FSFE Budgets: [\[http://www.germany.fsfeurope.org/about/funds/funds.de.html\]](http://www.germany.fsfeurope.org/about/funds/funds.de.html)
- [11] FSFE „Thank GNUs“: [\[http://www.germany.fsfeurope.org/help/thankgnus-2004.de.html\]](http://www.germany.fsfeurope.org/help/thankgnus-2004.de.html)
- [12] Bookzilla.de: [\[http://bookzilla.de\]](http://bookzilla.de)
- [13] FSF(US) Associate Membership Portal: [\[http://member.fsf.org\]](http://member.fsf.org)
- [14] FSFE-Projekte: [\[http://www.germany.fsfeurope.org/projects/projects.de.html\]](http://www.germany.fsfeurope.org/projects/projects.de.html)
- [15] FSFE-Newsletter: [\[http://www.germany.fsfeurope.org/news/newsletter.de.html\]](http://www.germany.fsfeurope.org/news/newsletter.de.html)
- [16] 2-Jahres-Bericht der FSFE: [\[http://www.germany.fsfeurope.org/documents/reports/es-2003.de.html\]](http://www.germany.fsfeurope.org/documents/reports/es-2003.de.html)
- [17] FSFE in dem Kartellverfahren gegen Microsoft: [\[http://www.germany.fsfeurope.org/projects/ms-vs-eu/ms-vs-eu.de.html\]](http://www.germany.fsfeurope.org/projects/ms-vs-eu/ms-vs-eu.de.html)
- [18] Samba: [\[http://www.samba.org\]](http://www.samba.org)
- [19] FSFE vor dem europäischen Gerichtshof: [\[http://mailman.fsfeurope.org/pipermail/press-release-de/2004q3/000037.html\]](http://mailman.fsfeurope.org/pipermail/press-release-de/2004q3/000037.html)

Der Autor

Dipl.-Phys. Georg C. F. Greve beschäftigt sich seit etlichen Jahren mit freier Software und kam früh zu GNU/Linux. Nach Mitarbeit im GNU-Projekt und seiner Aktivität als dessen europä-



ischer Sprecher hat er die Free Software Foundation Europe initiiert, deren Präsident er ist. Mehr Informationen finden sich unter: [\[http://www.gnuhh.org\]](http://www.gnuhh.org)

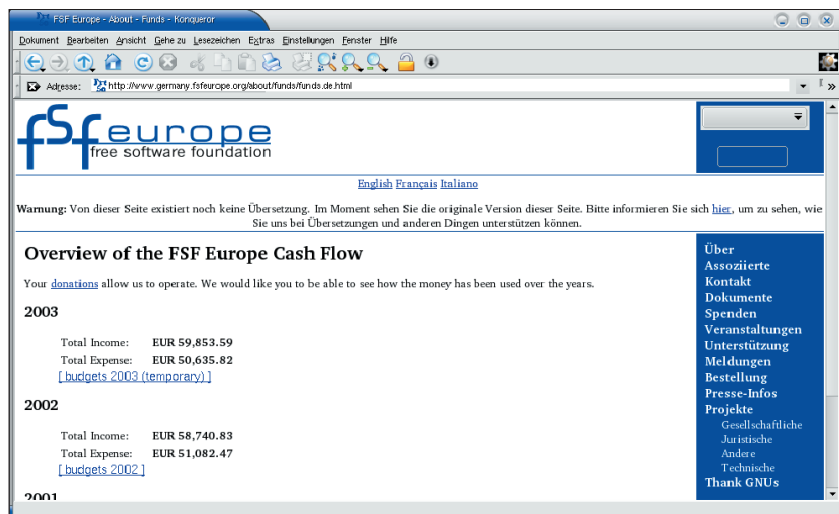


Abbildung 4: Transparenz im Umgang mit Spenden. Die FSFE legt seit Beginn ihrer Tätigkeit ihre Einnahmen und Ausgaben auf der Website offen.