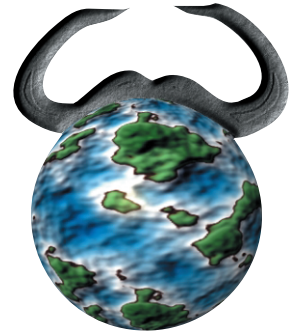


The Monthly GNU Column

Brave GNU World



The HyREX project [5] (Hypermedia Retrieval Engine for XML) is on the borderline between a database and an information retrieval system. HyREX is a tool that lets users retrieve information from XML documents. In this column, I will describe the HyREX project. I'll also respond to a question from a reader who asked what an individual user can do to make a difference within the open source movement.

Information Retrieval with HyREX

XML is a format for organizing data and passing information between applications. An example of an XML document is a file created by a publishing company with information on the company's books. The document could include a description of each book, as well as facts about the author and the price, and even the complete text of the book itself. XML's tree structure allows authors to represent a book as a collection of data elements, including sections and subsections of the text. It is possible to add a wide variety of content types to this structure. HyREX implements a query language similar to XPath to support this. However, HyREX expands on XPath's functionality, adding a fuzzy set framework and uncertainty for various file types. HyREX certainly makes sense in any case where you need to search a

This column explores the world of free software from the perspective of the GNU Project and the

Free Software Foundation (FSF). This month you'll learn about a new tool for information retrieval. You'll also learn how everyday users can play a part in the open source revolution. **BY GEORG C.F. GREVE**

document that can be rendered in XML, And HyREX also comes with an email converter.

Norbert Goevert and Kai Grossjohann developed HyREX at the University of Dortmund, Germany with some support from students as part of their thesis. This said, the project is at an early stage of development. HyREX was launched in 2001 and is still not finished. Kai Grossjohann is currently tidying up and fine tuning with occasional support from other programmers. HyREX is mainly written in Perl, with some performance-critical parts written in C.

While the authors were researching the project, they focused on the design rather than on performance. HyREX is not quick, and not easy to use at present, but all the basic functions are in place. Its design uses an extremely modern approach – and if the prospect of contributing to a state-of-the-art project sounds appealing, Kai would be pleased to hear from you at [5]. The to-do list includes some clean up and code consolidation, especially with regard to speed and memory use; new modules to support additional file types; and a more usable installer. And of course, the program's documentation is not as comprehensive as most users would prefer. There is so much to do.

Free Software Needs Support

Reader Henryk Gerlach sent me several interesting questions. Among other

things, Henryk wanted to know how individuals can provide material support for free software. I'll discuss some of the

ways you can offer financial assistance later in this article, but first, I'll describe some ways of contributing that do not involve money.

If you grew up with proprietary software, you might find it difficult to escape from the confines of acquired passivity and realize you can actually assist with the software development process. At first glance, you might think programming is the only way to help free software. Of course, programming is essential, but quality and usability are also

very important, and non-programmers can assist with improving the quality and usability of free software by volunteering for duties such as the following:

- **Localization** – If you speak a language that isn't supported by an open source product, you can volunteer to help localize the software for the new language. GNU programs often use the gettext application [6] for localization. gettext modifies and updates files with text fragments. After a short learning curve, users without major computing skills can master the art of localizing with gettext. If you want to get involved, you can typically do so without asking for permission. Simply send the file to the program's author when you are finished. Most authors will be happy to receive your work. Of course, it makes more sense to contact the

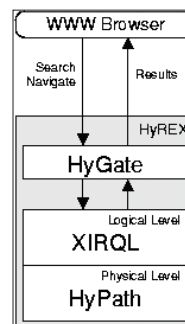


Figure 2:The HyREX project offers an ambitious, XML-based approach to information retrieval.



Figure 1: If you find a bug, you can help a free software project by supplying a detailed bug report.

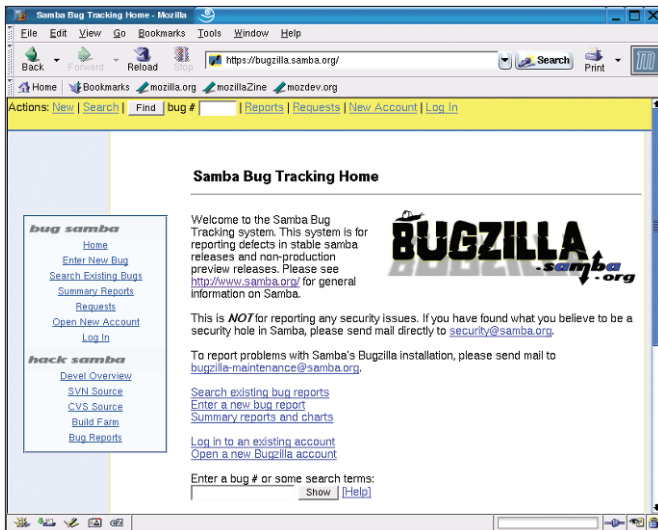


Figure 3: The Samba bug report page.

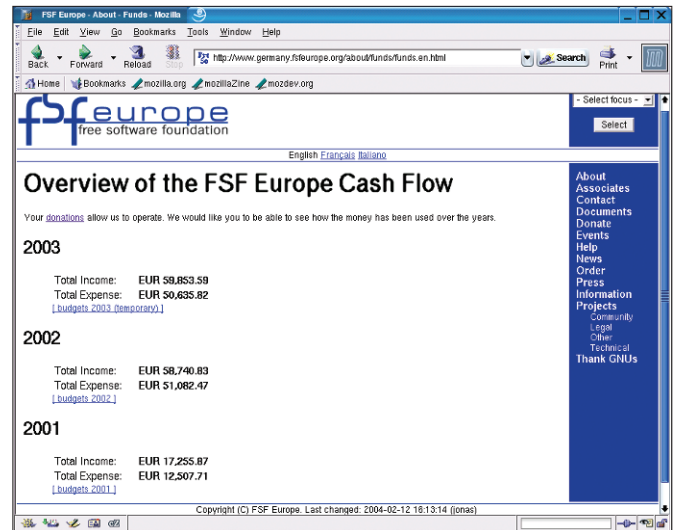


Figure 4: FSF Europe discloses budget information at its public website.

author first to find out if someone else is already working on the localization to your language.

- **Documentation** – All programs need some form of documentation. Most developers hate writing documentation, and they often underestimate how important documentation is. Good documentation can really help to propagate a program by making the software more accessible and easier to use. Helping out with documentation also allows the programmers to concentrate on new features.
- **Feedback** – Developers can't sit next to every user; all they can do is develop a program the way they think it should be. This leads to programs looking like they were written by and for developers. Good feedback is vital because it helps the programmer understand the needs of the user. But simply complaining is not the same as providing good feedback. Constructive criticism is a matter of courtesy. Instead of just complaining about a minor error such as a faulty window position, place your feedback in the form of a constructive suggestion. Your feedback could be a simple email message to an address listed at the program's web site, or it could be a detailed bug report. The quality of a bug report can make a difference in how quickly the bug is fixed. A good bug report should be factual and specific, describing the circumstances when the error occurs as clearly as possible. And don't forget to check out

the website and the mailing list archives to determine if someone else has already reported the same bug.

- **User Support** – You can assist less experienced users by answering questions in mailing lists, contributing to on-line forums, or tutoring users in real life. Any time you spend helping end users with their questions offloads some of the work from people involved in active development – programmers, documentation authors, translators, and testers.

In addition to these cost-free activities are the many ways you can donate money to support free software. As I said in a previous column, money is like "frozen time." And that takes us back to Henryk Gerlach's questions. Henryk is concerned about two things: the confusingly large number of options for donating and the typical lack of transparency.

Your donation options start with buying products from vendors who donate part of the selling price to a specific project or foundation. Donations via dealers provide very little in the way of transparency. In fact, the vendor is sometimes the only one who sees the whole picture. Henryk would like to see more transparency here – and as the author of this column, I fully support Henryk's demands.

Another approach is to give money directly to an open source project or foundation. Ideally, a direct donation to a non-profit organization provides a bit more transparency. Projects that receive

direct donations should keep records of what funds they have received and how these funds were used. The Free Software Foundation Europe (FSFE) [7] has had a policy of disclosing its budgets right from the outset [8], and it publishes a list of donors [9]. In some countries, this type of transparency is necessary for an organization to maintain its non-profit status. But even if transparency isn't required, I would heartily recommend it for organizations in a similar situation. At the same time, I would suggest that those of you who are interested in making a donation investigate how transparent, credible, and trustworthy the potential recipients are. ■

INFO

- [1] Send ideas, comments, and questions to Brave GNU World: column@brave-gnu-world.org
- [2] GNU Project: <http://www.gnu.org/>
- [3] Georg's Brave GNU World homepage: <http://brave-gnu-world.org>
- [4] "We run GNU": <http://www.gnu.org/brave-gnu-world/rungnu/rungnu.en.html>
- [5] HyREX: <http://www.is.informatik.uni-duisburg.de/projects/hyrex/>
- [6] GNU Gettext: <http://www.gnu.org/software/gettext/>
- [7] Free Software Foundation Europe: <http://www.germany.fsf europe.org/>
- [8] FSFE Budgets: <http://www.germany.fsf europe.org/about/funds/funds.de.html>
- [9] FSFE "Thank GNUs": <http://www.germany.fsf europe.org/help/thankgnus-2004.en.html>