OPEN SOURCE TECHNOLOGIES IN LIBRARIES:
THE EXAMPLE OF THE FACULTY OF PHILOSOPHY IN ZAGREB

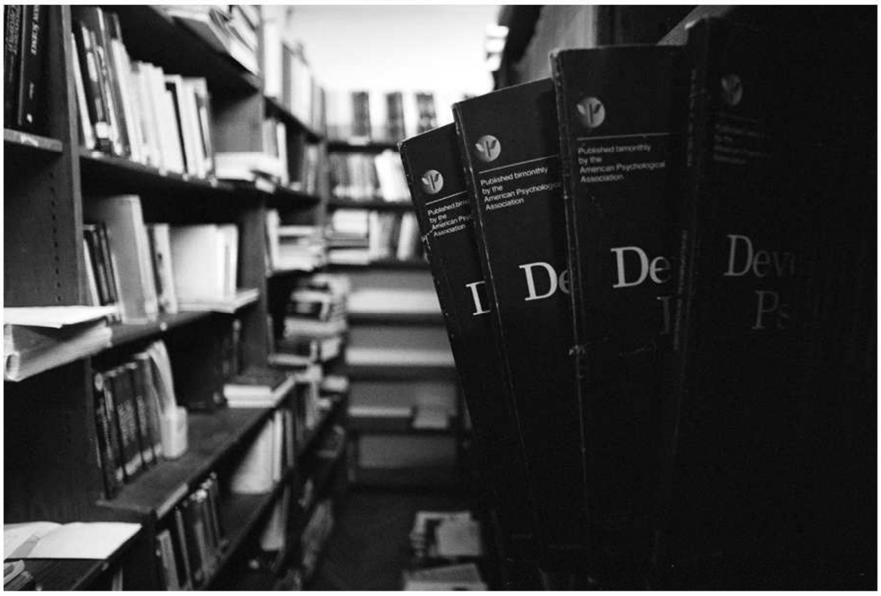
Marijana Glavica, mglavica@ffzg.hr * Marijana Mišetić, mmisetic@ffzg.hr * Dobrica Pavlinušić, dpavlin@rot13.org

Who we are?

FF Libraries

People and environment

- 19 different solo-libraries (each belonging to department, all belonging to one faculty)
- without central management or central service
- financed by faculty with limited money for books; little or no independence - have to negotiate with department authorities for extra financing
- insufficient communication (on all levels)
- similar problems (space, working conditions, communication with head persons)
- no work plan, no goals, no team work
- not experienced in joint projects
- dealing with information overload
- direct contact with users on a daily basis
- working still with old and obsolete tools together while trying to comprehend new tools



Level of automatization

- almost no automatization; starting efforts to turn catalogues in digital form - entering data (supported by SZI project) using partially functional solution for cataloguing - UNESCO CDS/ISIS
- some data in spreadsheets (dissertations, lists of journals)
- database in CDS/ISIS was designed following UNIMARC rules)

What was our main goal (or problem) in this project?

GO ONLINE!

enhance visibility of library collections
make the library records searchable through the web
develop online services

http://knjiznice.ffzg.hr/

Choosing the solution

Possible approaches to informatization of a number of small libraries belonging to a large institution

- waiting for integrated solution
  we were sick of waiting and believing in integrated solutions
- buying complete WebPAC solution
  implied expensive adaptation; free solutions inappropriate and hard to implement; mostly based on proprietary format; no use trying to improve that
- combination of existing Open Source solutions and development of customized parts for our needs

that is our choice!

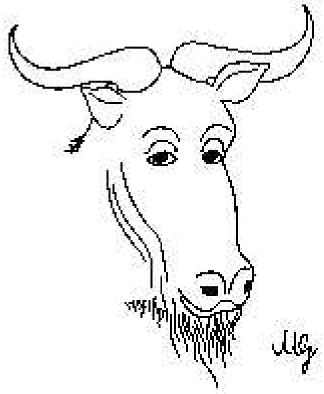
Free Software and Open Source

"Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech" not as in "free beer".

GNU GPL
General Public license

- require that all the released improved versions be free software
- GPL allows author to sell copies of the program for money
- author (copyright holder) transfers part of its rights to users

[http://www.gnu.org/]



Four kinds of freedom
for the users of the software

- to **run** the program, for any purpose (freedom 0)
- to **study** how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- to **redistribute** copies so you can help your neighbour (freedom 2)
- to **improve** the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

"**Open source** promotes software reliability and quality by supporting independent peer review and rapid evolution of source code. To be certified as open source, the license of a program must guarantee the right to read, redistribute, modify, and use it freely."

[**http://www.opensource.org**]

Open Source in libraries

[http://www.oss4lib.org/]

Main information source for those interested in Open Source in libraries.

- news
- mailing list
- readings
- bibliography
- projects
  (list of software)

What do librarians do?
mediation between users and information

- collect information
- search for information
- make information available
- develop tools for information searching

should have influence on developing policies
to support freedom of information!

How libraries benefit from Open Source?

- more control over the development of services and collections
- informal review process (in contrast to competitive marketing approach)
- resources sharing, knowledge sharing
- open source data

Component architecture

What is so special about component architecture?

- allows changing of any component if needed (because of insufficient functionality, user preference or bugs)
- find the correct solution the first time - **is hard!** (so, we'll do it step-by-step; or component-by-component)
- implementation of component architecture can have significant overhead compared to monolith (but enables incremental development in which users can participate during development and not only at the end)
- reasoning for component approach is flexibility without disturbing other parts of the system - **supports changing requests**

Which components do you need?

Our situation can serve as example:

- *fixed:* CDS/ISIS for cataloging
- *optional:* WebPAC for search engine and UI
- *planned:* circulation

Solution:

> **integration of individual parts which allows incremental development**

This solution is different than reasoning of Koha [http://www.koha.org/] team which produced another nice integrated solution called Koha which is a full catalogue, OPAC, circulation and acquisitions system.

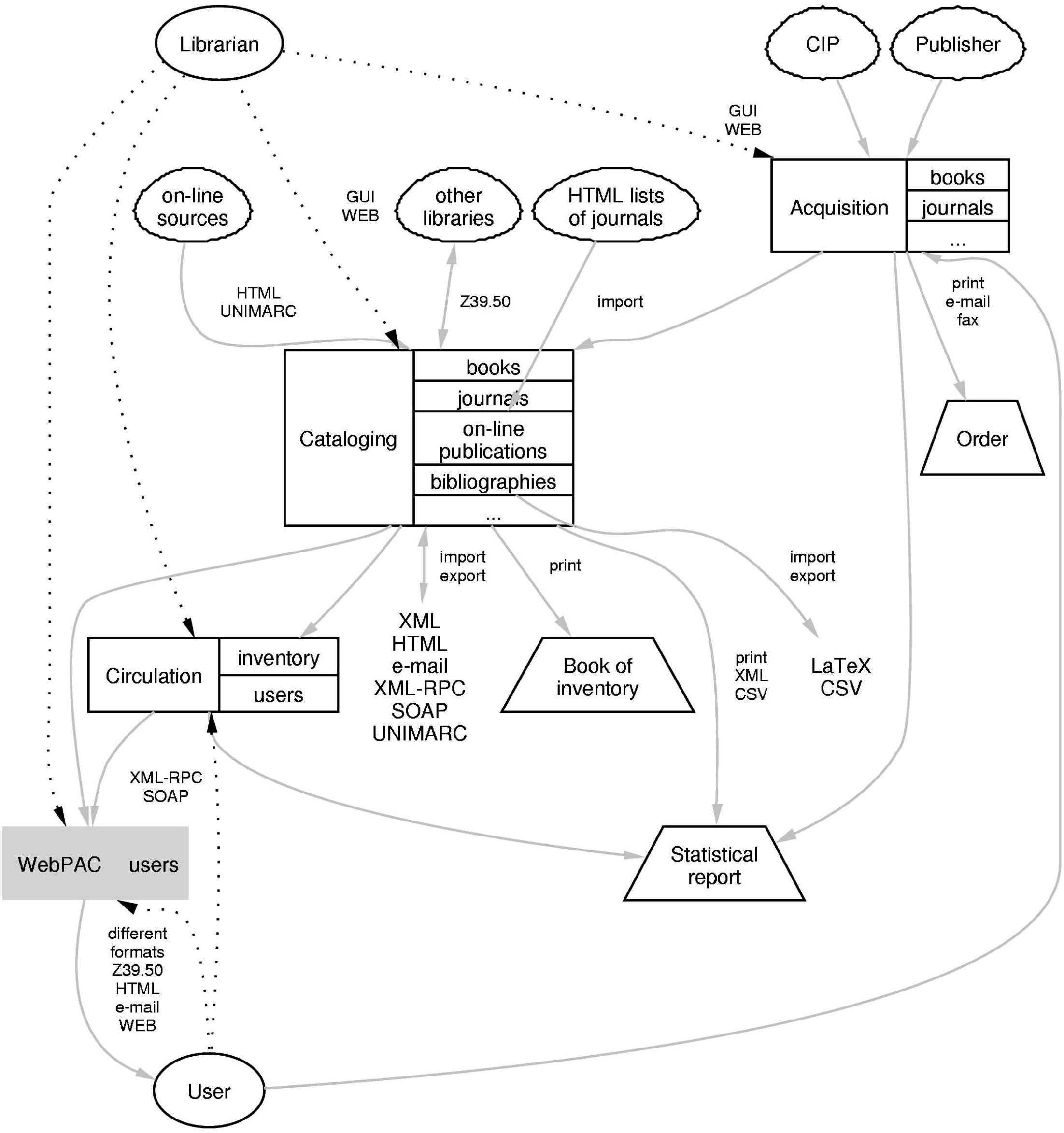
Perfect search engine

Ideas to start with and quest to find one...

- Don't use RDBMS (unsuitable for text entries; too slow and limiting from own experience)
- use existing Open Source technology and improve it
- main concern HCI, not performance

Search through http://freshmeat.net/, http://sourceforge.net/ and finally http://www.oss4lib.org/

We were not alone in our search.
Eric Lease Morgan wrote an article "Comparing Open Source Indexers"
[http://www.infomotions.com/musings/opensource-indexers/]



Component architecture for library automation

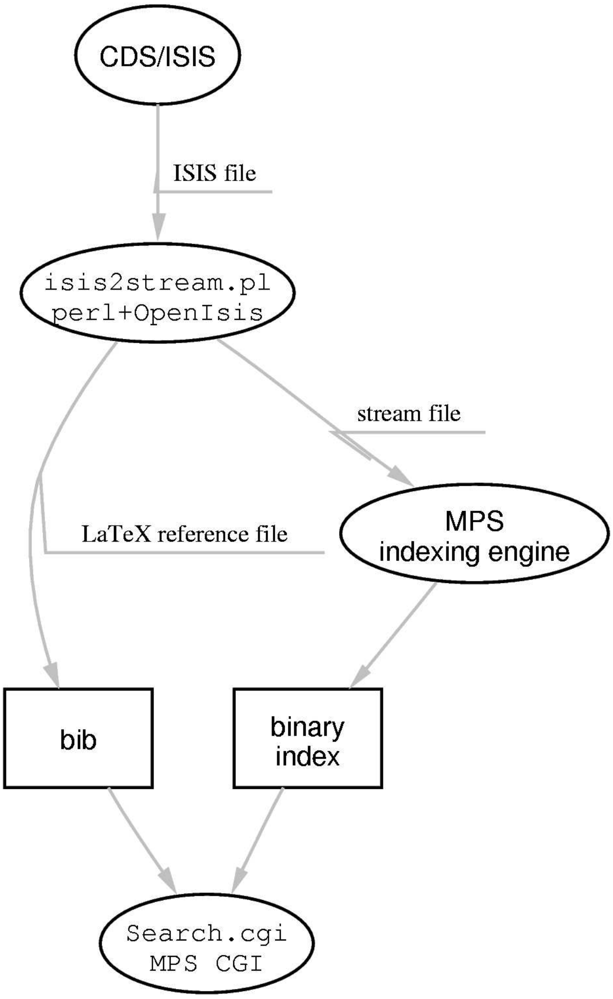
Programming Perl – first try

Prototype implementation of MPS

- isis2stream.pl using OpenIsis [http://www.openisis.org/] libraries to access CDS/ISIS database
- user management using HTTPD::UserAdmin
- modification of Search.cgi UI (JavaScript, CSS)
- most of the time we edited formats of stream and bib files

Problems with MPS

- code hard to modify (it's C-like and not Perl-like)
- html in-between code, without the use of templates (localization problems)
- occasional bugs (ISBN* is a nasty one)
- Open Source but commercial product, *free as a beer, but not as a speech*

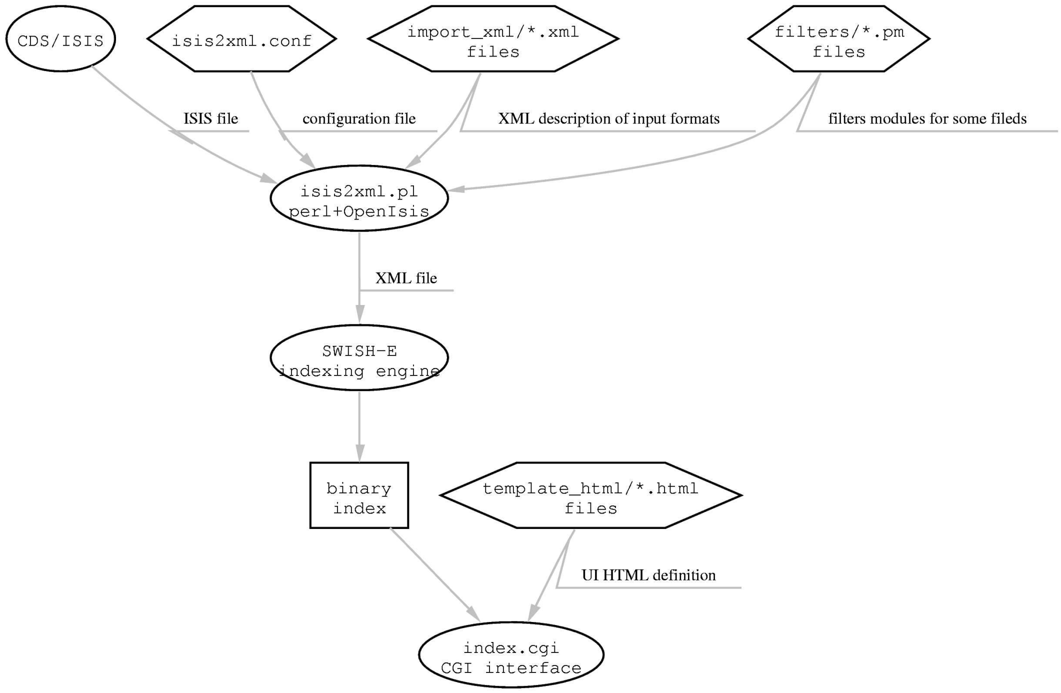


Move to SWISH-E which is **free**, but not the "turn-key" solution.

Programming Perl – second try

Move to SWISH-E was not easy... We had to re-write everything, which wasn't that bad: we had some experience with prototype and knew how to make better system. New architecture include:

- configuration file which defines which input file will use which xml definition file
- flexible xml definition file for mapping between input format fields and fields which are used for indexing and/or display
- filters which can be applied to fields before indexing or display to help normalize data (e.g. Replace language codes to ISO equivalents or similar)
- html template files which define user interface (and enable localization)
- just two perl programs: `isis2xml.pl` and `index.cgi` (which actually uses `WebPac.pm` for all code)



WebPAC and librarian

What you need to know as a librarian...

- users' needs
- structure of your data
- html basics
- css basics

and have some affinity for IT

... to change the following elements of your WebPAC:

- UI design
- a variety of input data formats (had to be structured)
- what is to be searched and how
- what is to be displayed and how
- what is to be shown in index

That is possible through flexible configuration files, described above.

FF WebPAC gives freedom to librarian to make changes in all those elements, not depending on vendor or maintainer.

Project timeline

Planned in August 2002 and tasks completed up to now (May 2003):

- planning and development of circulation using PostgreSQL, perl and XML/RPC
- evaluating different UI (telnet, Windows using XML/RPC)
- ~~implement single common configuration file~~
- ~~replace indexing engine with SWISH-E~~
- ~~publish project on WebPAC.sourceforge.net~~

During that time we also explored and collaborated on various other Open Source projects that we used for WebPAC or which are complementary to ours like MyLibrary (personal library portal) which we helped port to PostgreSQL.

Instead of conclusion

Dispelling the doubts

- software and hardware are the most important elements of IT (expensive equipment first)
- only integrated solutions can be successful
- well known vendors provide security and peace of mind

Some of that might be true, but not necessary for us.

A short note from "ordinary" librarian:

I would like to express our great happiness that we have finally conquered our space on the Web. We have created our WebPAC as a result of shared efforts of both librarians (26 librarians from the Faculty of Philosophy) and one programmer. It is primarily the result of good will and enthusiasm while working with restricted funds, but also of using IT knowledge. All parts of the puzzle were finally put together! But we must point out that without IT knowledge and technology all our efforts would remain fruitless. Our collaboration has shown that shared participation can remove all artificial barriers between different experts. (*M. M.*)

Full source code of SWISH-E based WebPAC and article about first phase of development is available at
[http://webpac.sourceforge.net/]