Universidade de Évora

Master's Thesis in Computer Science Engineering

An Integrated Library System on the CERN Document Server

Author
Joaquim Jorge Rodrigues Silvestre

Supervisors
Luis Arriaga da Cunha (Universidade de Évora)
Jean-Yves Le Meur (CERN IT-UDS-CDS)
Tibor Šimko (CERN IT-UDS-CDS)

April, 2010

Universidade de Évora

Master's Thesis in Computer Science Engineering

An Integrated Library System on the CERN Document Server

Author
Joaquim Jorge Rodrigues Silvestre

Supervisors
Luis Arriaga da Cunha (Universidade de Évora)
Jean-Yves Le Meur (CERN IT-UDS-CDS)
Tibor Šimko (CERN IT-UDS-CDS)

April, 2010

Resumo

Um Sistema Integrado para Bibliotecas no CERN Document Server

O CERN – a Organização Europeia para a Investigação Nuclear – é um dos maiores centros de investigação a nível mundial, responsável por diversas descobertas na área da física bem como na área das ciências da computação. O CERN Document Server, também conhecido como CDS Invenio, é um software desenvolvido no CERN, que tem como objectivo fornecer um conjunto de ferramentas para gerir bibliotecas digitais. A fim de melhorar as funcionalidades do CDS Invenio foi criado um novo módulo, chamado BibCirculation, para gerir os livros (e outros itens) da biblioteca do CERN, funcionando como um sistema integrado de gestão de bibliotecas. Esta tese descreve os passos que foram dados para atingir os vários objectivos deste projecto, explicando, entre outros, o processo de integração com os outros módulos existentes bem como a forma encontrada para associar informações dos livros com os metadados do CDS Invenio. É também possível encontrar uma apresentação detalhada sobre todo o processo de implementação e os testes realizados. Finalmente, são apresentadas as conclusões deste projecto e o trabalho a desenvolver futuramente.

Palavras-chave: CERN Document Server, Invenio, Sistema Integrado para Gestão de Bibliotecas, Biblioteca, BibCirculation;

Abstract

An Integrated Library System on the CERN Document Server

CERN – The European Organization for Nuclear Research – is one of the largest research centres worldwide, responsible for several discoveries in physics as well as in computer science. The CERN Document Server, also known as CDS Invenio, is a software developed at CERN, which aims to provide a set of tools for managing digital libraries. In order to improve the functionalities of CDS Invenio a new module was developed , called BibCirculation, to manage books (and other items) from the CERN library, and working as an Integrated Library System. This thesis shows the steps that have been done to achieve the several goals of this project, explaining, among others aspects, the process of integration with other existing modules as well as the way to associate the information about books with the metadata from CDS Invenio. You can also find a detailed explanation of the entire implementation process and testing. Finally, there are presented the conclusions of this project and ideas for future development.

Keywords: CERN Document Server, Invenio, Integrated Library System, Library, BibCirculation;

Agradecimentos

Esta tese de mestrado é o resultado de vários meses de trabalho, mas tal não teria sido possível sem a ajuda e o apoio de várias pessoas.

Em primeiro lugar, gostaria de agradecer ao meu orientador no CERN, Jean-Yves Le Meur, que me escolheu para participar neste projecto, dando-me a possibilidade de conhecer uma realidade que muito me fez crescer enquanto pessoa. Ao Tibor Simko que me deu a conhecer o projecto CDS Invenio e com o qual troquei várias ideias acerca do meu trabalho. Aos restantos membros da secção CDS, o meu obrigado pela forma como me receberam, em especial os meus colegas de gabinete que sempre me ajudaram. O meu obrigado ao Samuele Kaplun, ao Jerôme Caffaro e ao Marko Niinimaki. Aos membros da biblioteca do CERN com quem pude trocar várias ideias e aprender mais acerca dos sistemas de automação de bibliotecas. Em especial, para o Jens Vigen, o Tullio Basaglia e a Anne Gentil-Beccot, o meu muito obrigado por tudo aquilo que me ensinaram. Gostaria de expressar também a minha gratidão ao meu orientador da Universidade de Évora, Prof. Luis Arriaga, que sempre me encorajou neste meu trabalho e me ajudou quando estava a escrever esta tese.

À minha namorada, Ana, que sempre me apoiou com o seu carinho e amor, nesta étapa em que estivemos distantes, bem como à sua família, em especial à Maria-João, ao Rui e à Bárbara.

Finalmente, gostaria de agradecer aos meus pais, Isabel e Francisco, que sempre me apoiaram nesta grande aventura e me ajudaram a tornar-me na pessoa que hoje sou.

Acknowledgements

This master's thesis is the result of several months of work, but this would not have been possible without the help and support from several people.

In first place, I would like to thank my supervisor at CERN, Jean-Yves Le Meur, who chose me to participate in this great project, giving me the opportunity to discover a new reality who made me grow as a person. Also a special thank to Tibor Simko, who guided me through CDS Invenio and with whom I exchanged many ideas about my work. The other members of the CDS section, my thank for the way I was received and for their friendship, especially my office colleagues who always helped me. My thanks to Samuele Kaplun, to Jerome Caffaro and Marko Niinimaki. To the members of the CERN library with whom I could exchange ideas and learn more about the library automation systems. In particular, for Jens Vigen, Tullio Basaglia and Anne Gentil-Beccot, thank you very much for everything you taught me. I would like also to express my gratitude to my supervisor at the University of Évora, Prof. Luis Arriaga, who always encouraged me in my work and guided me when I was writing this thesis.

To my girlfriend, Ana, who always supported me with her love, when we were distant, and her family, especially to Maria-João, Rui and Bárbara.

Finally, I would like to thank my parents, Isabel and Francisco, who always supported me in this great adventure and helped me to become the person I am today.

"A problem that seems difficult may have a simple and unexpected solution."
Martin Gardner

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Context

This thesis describes the work developed at Information Technology Department of the European Organization for Nuclear Research (CERN), for the CERN Technical Student Programme.

During 18 months, I worked on the CERN Document Server (IT-UDS-CDS) section, on the development of CDS Invenio, more precisely on the creation of a new CDS Invenio module, called BibCirculation.

1.2 CERN

CERN[1] - The European Organization for Nuclear Research is one of the world's largest laboratory for scientific research [1]. Its main field of work is fundamental physics, trying to find out what the Universe is made of and how it works. At CERN, the most complex scientific instruments, like particle accelerators and detectors, are used to study and investigate the basic constituents of matter - the fundamental particles. By studying the collision of particles, physicists can learn and discover more about the laws of Nature.

Founded in 1954 by 12 countries, CERN includes now 20 member states. CERN is located near Geneva on the Franco-Swiss border, it employs 3000 persons. Also, some 6500 visiting scientists, half of the world particle physicists, come to CERN for their research. They represent 500 universities and over 80 nationalities. Since 1954, CERN has made several important discoveries for which scientists have been distinguished with prestigious awards, including Nobel prizes[2].

In March 1989, Tim Berners-Lee, a CERN scientist, wrote a proposal[3], to prevent the problem of *"losing information at CERN"*. This proposal was the beginning of the World Wide Web (WWW). The WWW was also originally developed as the answer to the demand for automatic information sharing, between

[1] See `http://www.cern.ch`

Figure 1.1: CERN - Meyrin Site.

the scientific commuty working in different universities and institutes all over the world.

1.3 CDS Invenio

CDS Invenio[2] is an integrated digital library[4] system conceived and developed at CERN, by the CERN Document Server (CDS) section, in the User and Document Services (UDS) group. With CDS Invenio it is possible to provide the framework and tools for building and managing an autonomous digital library server. The development of CDS Invenio started in 1993 primarily for internal needs as an institutional repository. From 2000, CDS Invenio provides support for multimedia files and OAI-PHM. Nowadays, it represents a suite of applications, used in several places outside CERN such as École Polytechnique Fédérale de Lausanne (EPFL), for general administration of documents, institutional repositories or large-sized library system.

[2]Formerly CDSware

CDS Invenio is available for free and licensed under the GNU General Public Licence[3] (GPL). From a technical point of view, CDS Invenio runs on GNU/Unix systems, using a MySQL database server and an Apache/Python web application server. The software is mainly written in the Python programming language, with some ad hoc modules and functionalities developed in Common Lisp and C.

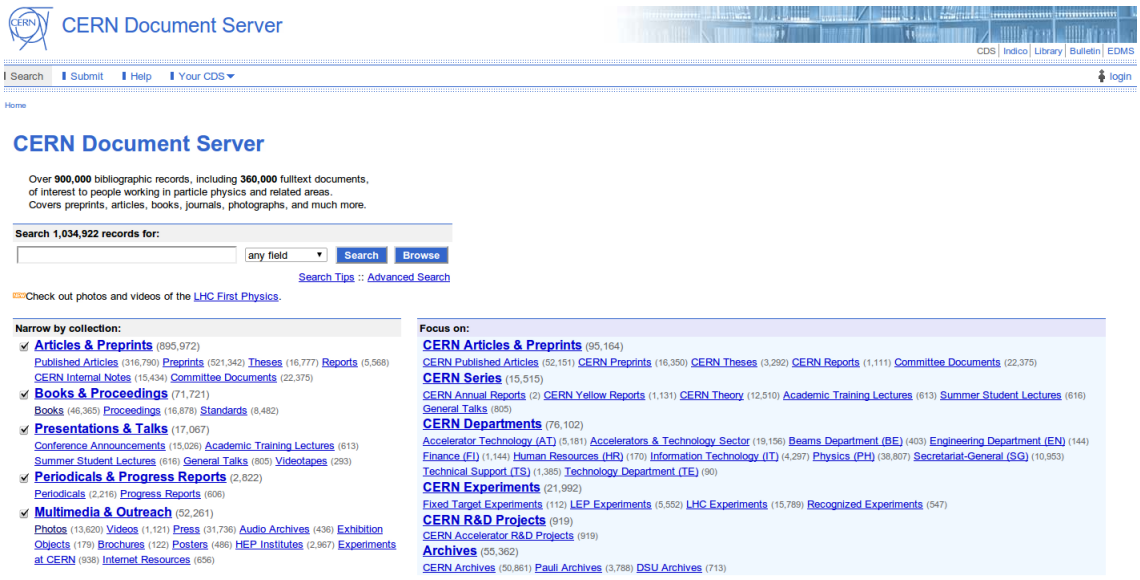

Figure 1.2: CERN Document Server – home page.

The solutions proposed by CDS Invenio cover all the management requirememts of a digital library. It supports the Open Archives Initiative metadata harvesting protocol (OAI-PMH) and uses MARC 21 as its bibliographic standard. The flexibility and the performance of CDS Invenio, make it a attractive and realable solution for the management of document repositories of moderate to large size.

At CERN, CDS Invenio responsable for the management of over 1 million bibliographic records and 500000 fulltext documents, serving 20000 users per month and issuing over than 8000 queries per day. CDS Invenio is not only running at

[3]See `http://www.gnu.org/licenses/gpl.html`

CERN. Currently, it is installed and in use by over a dozen scientific institutions all over the world.

1.3.1 Modules Overview

CDS Invenio is composed by several modules (figure 1.3), having each one a specific functionality. Modules may have different prefixes, in their names. In general, the prefix *"Bib"* is used for modules related with bibliographic data and the prefix *"Web"* is related with modules who work more with the web interface.

- **BibCheck** allows to administrators and cataloguers to automate a variety of tests on the metadata to see whether the metadata comply with quality standards. This offers also the possibility to fixe some kind of errors.
- **BibClassify** permits the automatic extraction of keywords from fulltext documents. The extraction process is based on the frequency of specific each terms, who are taken from a controlled vocabulary.
- **BibConvert** permits metadata conversion from any structured or semi-structured proprietary format into any other format. This conversion is typically to MARC XML[4], that is natively used in CDS Invenio.
- **BibEdit** allows the edition of metadata via a web interface.
- **BibFormat** is responsible for the formatting of the bibliographic metadata, having several types of outputs.
- **BibHarvest** represents the OAi-PMH compatible harvester. It allows the repository to gather metadata from other OAi-compliant repositories and is also in charge for OAi-PMH repository management.
- **BibIndex** is in charge of the indexation of metadata, references and full text files.
- **BibMatch** permits the input filtering of XML files, avoiding doubly-inputted records.

[4]See `http://www.loc.gov/standards/marcxml/`

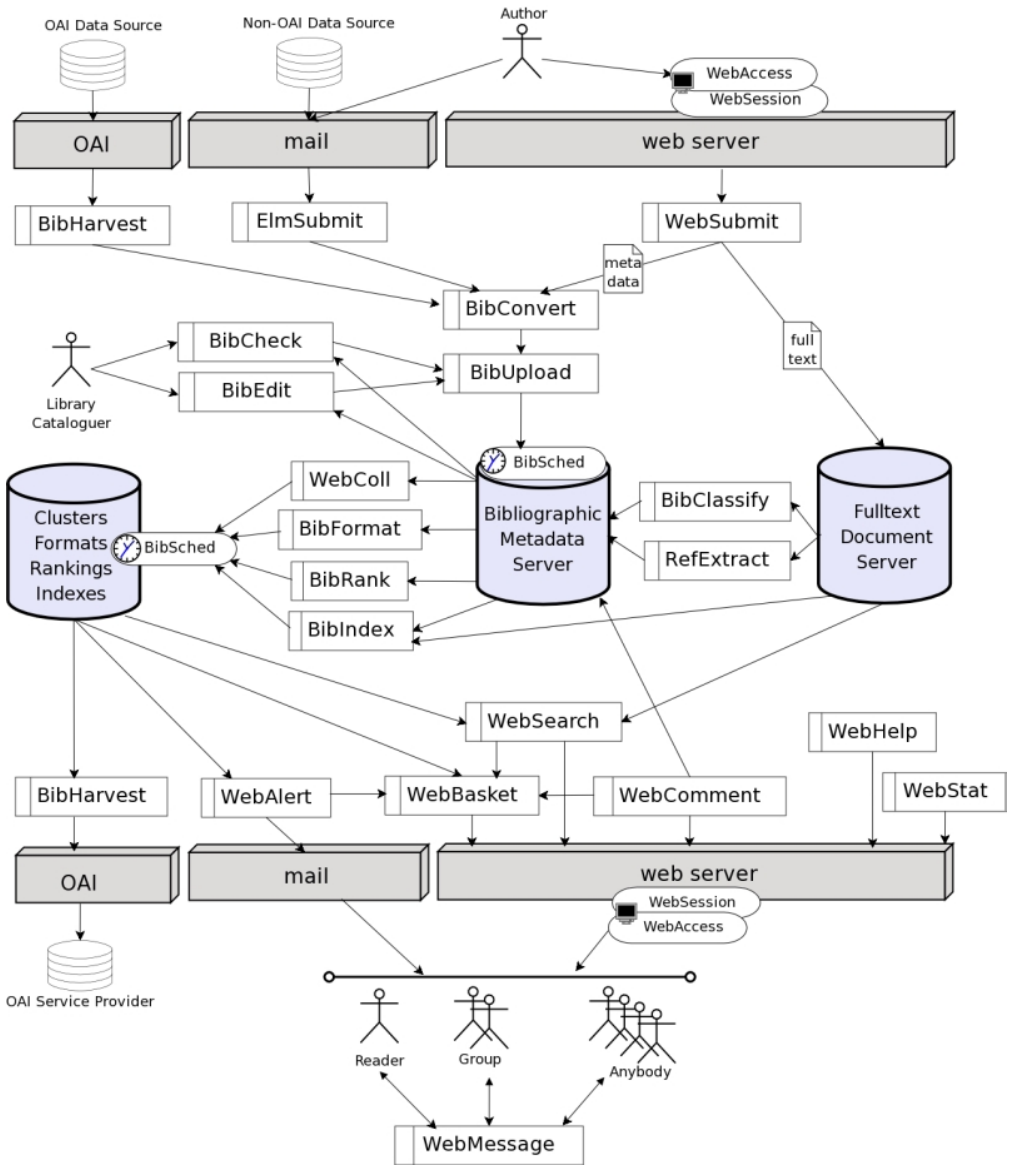

Figure 1.3: CDS Invenio Modules Overview.

- **BibRank** permits to set up a variety of ranking criteria that will be used later by the search engine.
- **BibSched** is the central unit of CDS Invenio. It allows all other modules to access the bibliographic database in a controlled way, preventing sharing violation threats and assuring the coherent execution of the database update tasks.
- **BibUpload** permits to load the new bibliographic data into the database.
- **ElmSubmit** is an email submission gateway who permits automatic document uploads from trusted sources via email.
- **MiscUtil** is a collection of miscellaneous utilities that can be used by the other modules.
- **WebAccess** is the module in charge for granting access to users, in order to perform various actions within the system.
- **WebAlert** permits, to the end user, the reception of alert messages, each time a new document matching his personal criteria is inserted into the database.
- **WebBasket** enables the end user to store the documents who he is interested, in a personal basket or a personal shelf.
- **WebComment** provides a community-oriented tool to rank documents by the readers or to share comments on the documents.
- **WebHelp** shows user-level, admin-level and hacker-level documenation on CDS Invenio.
- **WebMessage** permits the communication between users via web message boards.
- **WebSearch** handles user requests to search for a certain words or phrases in the database.

- **WebSession** is a session and user management module that allows to differentiate users.
- **WebStat** is system that allows to get statistics about the health of the server, the general usage of the system and some other particular system features.
- **WebStyle** permits to define the look and feel of CDS Invenio pages.
- **WebSubmit** is a submission system, that permits authorized individuals (authors, secretaries and repository maintenance staff) to submit individual documents into the system.

1.4 BibCirculation

In this section, I will explain what is this new module of CDS Invenio called BibCirculation. I will also explain the origin and the ideas behind its creation and what was my role, in this project. Finally, I will give a overview about the goals of this new module.

1.4.1 The origin

When I started this project, it was proposed to me, to created a new module for CDS Invenio. My job was to create and develop, several functionalities for this new module from scratch. I was responsible for all the process of development, from the requirements analysis until the "production". The idea for this project was to find a way to manage all the physical items of the CERN library and automate several library functions. I was told that there was already a software used for this purpose. The software used by the CERN library was ALEPH 500. It is a librarian software very popular and used world wide. When I arrived at CERN, all the library functions were manage by ALEPH 500, but there was a link with CDS Invenio. The Online Public Access Catalog (OPAC) was provided by CDS Invenio. That means when a borrower wanted to request, for example, a book, the process was started in CDS Invenio, in the collection "Books" or in the main

search interface. And in the result list it was possible to select a link "CERN Library copies" and see all the available copies. The process was finished with the register of the new request in ALEPH 500.

This type of situations was quite common. For example, to search for a book or an article, the best way to do it was using the search engine provided by CDS Invenio (witch is quite good and efficient). But the process request was treated by ALEPH 500. Several librarian functions, like lending books, were managed by ALEPH 500. The usage of two different application for the same process, was not very efficient and it was common to have problem with consistence of the data. It was necessary to keep always the information of both system synchronized.

To avoid all these situations, it was decided, by the CDS section, to created a new module in CDS Invenio. This new module should be able to automate several functions, avoid all these synchronization procedures and being perfectly integrated with the other modules of CDS Invenio. The idea was to provide library automation tools, like the tools that is possible to find in an Integrated Library System (ILS). CDS Invenio would be a digital library who would able to provide the functionalities of an ILS. This was the beginning of BibCirculation. But why BibCirculation? It was a strange name... Indeed, but it was following the same rule as the other modules of CDS Invenio. And what about Circulation. Well... Circulation systems are one of the earliest examples of the application of data processing technologies in libraries. Circulation is the process that involves, in first place, the relation between borrowers and books (these two entities are the most relevant in a Circulation system) and, in second place, the status of mentioned pair in the system. Circulation is one of the most common modules provided by an ILS. This module was created with the intent of automate all the tasks involved in the lending of material to borrowers (also called patrons). This type of module must offer a very simple and efficient ways to deal with routine transaction like loans, renewals, overdue notifications and returns.

A circulation modules usually has a database to manage information about booksAlso called items or holdings and borrowers. Often, they provide also the ability

to register new items when items (books, articles, etc...) are not already available on the system. Many libraries impose fines when materials are not returned on schedule. It is common to have circulation modules verifying and managing fine accumulations, payments, creating notices and performing related functions. The requirements of a circulation modules are not always the same, they change from library to library. For example, academic libraries often need the possibility to manage a set of books placed on reserve for a course, imposing rules much more strict in terms of loan period. Public libraries usually have to deal with a much heavier volume of transactions. Circulation modules have to process, usually, a high volume of operations. To become the job of library staff easier, these systems have specialized input, like barcode readers to scan information without need to type.

1.4.2 Project Goals and Overview

BibCirculation is new module of CDS Invenio, developed during this project, completely from scratch. Like all the other modules of CDS Invenio, the new BibCirculation module will be embedded in the Open Source spirit. More precisely, behind the creation and the development of BibCirculation there was three main goals:

1. **Create a new integrated module, using all the advantages of CDS Invenio** such as a powerful search engine and treatment of metadata, in order to be used by the CERN Library. BibCirculation should be able to respond to all the needs and demands of the CERN library, automating several operations and bringing new solutions and features to improve the services provided by the CERN library.

2. **Provide association between books information and metadata**. This association is extremely important. It will be possible to search for a book, not only based on the bibliographic data, stored in the medata, but using also a new type of relevant information such as the due date of a book, its status or even its single barcode.

3. **Make CDS Invenio an application more attractive and interesting for new potential users/clients**. With this new module, CDS Invenio can explore new areas and 'markets'. The addition of such kind of tool, can increase the popularity on CDS Invenio and become a reference in the Open Source context.

More specifically, in terms of implementation, BibCirculation should also:

- **Provide an efficient and user-friendly GUI, for library staff and borrowers**. This aspect is extremely important, because it can represents the difference between the acceptance and the rejection of an application. If the GUI is not good enough, our software won't be selected, because it won't be attractive in terms of design and usability.

- **Permit the management of borrowers, items, vendors and libraries**. BibCirculation has to provide a full set of functionalities like search for borrower, item or library; add a new borrower, item or library, update information, recall an item and notify a borrower (providing also a full list of templates).

- **Allow the management of requests and loans**. In order to achieve this objective, a new set of features has to be developed, such as create new request or loan, cancel a request or a loan, return a loan and check for new request, recall a loan, change due date of a loan, provide a daemon to send overdue letters and detect expired loans;

- **Provide lists with requests and loans overview**. It is important to know, at any moment, the current status of a library, in terms of loans or requests. For this, BibCirculation will provide different types of lists showing the current loans, the expired/overdue loans, all the items on shelf with request and all the items on loan with request;

- **Provide historical information about requests and loans**. This kind of information is important, because with it, we can see what are the books with more requests, with the biggest number of renewals, etc.

- **Migrate and integrate all the circulation data from ALEPH 500**. For an internal use at CERN, it is important to provide to BibCirculation all the data contained on ALEPH 500, because this data is important for the library staff. To achieve this goal, it will be necessary to create a set of tools to migrate and integrate all the information.

- **Provide support for acquisitions and ILL**. Acquisitions and ILL are usually in separated modules. But since most of the relevant information to acquisitions and ILL will be in BibCirculation, all these library functions will be in the same module of CDS Invenio. It is important to have a tool to track and manage acquisitions. BibCirculation should be able to do such thing. In the case of ILL, it would be good to have a tool with support for the Z39.50 protocol. Nowadays, its not used by the CERN Library, but it will be always an advantage the development of this feature.

- **Provide a complete documentation for library staff**. A correct and explicit documentation is always significant in the adoption of a new application.

1.5 Structure

This section describes the organization and the structure of the present thesis.

Chapter 1 explains the context where was developed all the project and gives a small explanation about what is CERN and its main fields of work. In the Chapter 1 gives also general overview about CDS Invenio and its modules. Finally, it presents the ideas and the goals behind the creation of BibCirculation.

Chapter 2 gives a global overview about what it is an Integrated Library System (ILS), presenting also historical aspects and shows the evolution of ILS since the 60's until nowadays. It presents also a section about digital libraries and their relation with ILS. In this second chapter, I talk about the use of open source ILSes and their advantages, and finally, I give a general overview about several ILSes softwares.

On the Chapter 3 it is shown all the different stages of the development of BibCirculation, since the requirements analysis until the technologies and tools used for the implementation. In the end, Chapter 3 presents external sources of information used for the conception of BibCirculation.

Chapter 4 presents all the all tests done after the implementation of BibCirculation. It describes also the different approach used in terms of tests such as regression tests and integration tests. Finally, it shows the comparison between BibCirculation and its features, and other ILS.

Chapter 5 presents the conclusions of this project and the further work who should be done in order to become BibCirculation a more complete, competitive and mature tool, for library management.

Chapter 2

Integrated Library Systems

In this chapter, I will give an explanation about what is a Integrated Library System. There will be a section about the state of art and an historical perspective about the creation and the evolution of Integrated Library Systems, in the last 50 years. There is also a section about the difference between Integrated Library Systems and Digital Libraries, to avoid misunderstanding about these type of applications. And finally an overview and comparison between different types of Integrated Library Systems and Digital Libraries.

2.1 State of the Art

An Integrated Library System[1] (ILS), is an application system for a public and academic libraries. An ILS is planned, conceived and developed to coordinate and automate several library functions, and also represent and register all the library operations. Usually each function of an ILS, is associate to a specific module[6]. There are several examples of modules. In the following list, you can see the most commons:

1. **Circulation** – used to register lending/receiving of materials from borrowers;

2. **Acquisitions** – used to register ordering, receiving and invoicing materials. Claim and cancel late orders and material not received;

3. **Cataloging** – create records for new material. Used for classifying and indexing materials;

4. **OPAC** – provide public interface for users;

5. **Serials** – used for tracking magazine and newspaper holdings;

6. **ILL** – used for interlibrary loans, supporting Z39.50 protocol[2].

[1] Or Library Management System (LMS). There are different names for this specific kind of application. Some authors call them Integrated Library Management System (ILMS).

[2] Protocol for information retrieval from Bibliographic Databases.

For library staff, integrated library systems are very welcome. This type of application improves the efficiency of all the operations in a library. This kind of software has many advantages because it permits not only the control of library operations such as loans and requests, but it provides also an excellent set of tools to manage books and borrowers. The usage of an integrated library system, usually, requires only a single time entry of data (bibliographic and users).

2.1.1 The history of ILS

To understand what has become an Integrated Library System, we should look to the last 50 years. During the last five decades, many improvements have been done. Sometimes adapting existent technology or even creating new solutions and giving a step further. The evolution of Integrated Library Systems is quiet impressive and plenty of remarkable achievements.

1960's - Experimental systems

In the beginning of the 1960s, many libraries, specially in the United Kingdom and in the United States began to experiment the use of computers to treat and help in the processing of information. The majority of these systems had his origin in the eighty-column punched card[3] data processing systems.

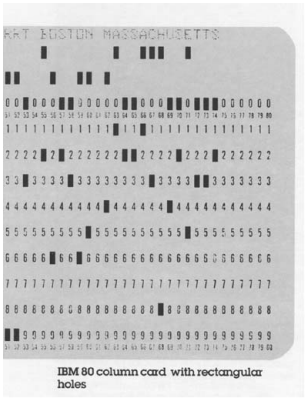

Figure 2.1: IBM 80 column card with rectangular holes[11].

[3]Card designed by IBM in 1928[12].

Those pioneer systems were created by Herman Hollerith[13], in order to aid in the processing of information from the 1890 US Census. But, the idea behind the creation of these cards was given to Hollerith by Dr Billings[14], at this time, librarian of the Library of the Surgeon's General Office.

Nowadays, it's possible to find some articles, or even stories, about the beginning of the usage of processing data systems in libraries. On these documents, it's usual to find also some interesting memories[8] as you can see below:

"When I first started working in libraries, we had a punched card system at Exeter City Library, Castle Street. I left in 1969, but I remember the clunky sorting machine which needed a room of its own (a small room!). It was an innovative system at the time."

In 1968, Dr Ralph Halsted Parker[5], professor at the University of Missouri, was one of the pioneers in library mechanization systems. He used for the first time, the term *"Library Information Systems"*. For him Library Information Systems was not only the *automation* of existing library procedures, such as circulation or cataloging, but also providing access to materials held electronically even by other libraries and information centers all over the world.

In the United Kingdom, Rollo Woods was one of the first people involved with computer systems in the university sector at Southampton University. A paper about the use of an ICT 1907[15] computer for the loans system at Southampton University was the first paper to be published in the journal *Program: news of computers in British university libraries* when it was launched in 1966. This journal was founded by Richard Kimber and in the first issue, he noted that:

"A new have of enthusiasm is sweeping the world of libraries in Britain. Librarians see that it is possible to use computers for most clerical operations in libraries. As a result of the recent Flowers report, more computing machinery will be installed in British universities, and librarians are anxious to stake claims for shares in increased computer time, which will therefore become available... The purpose

of Program is to assist librarians in learning about what is the beginning to be done in this field, to provide a medium for discussion of the problems involved, and to help establish direct personal contact between those working in similar directions."

At the same time, at Newcastle University, Maurice Line was involved, with some colleagues, on the development of an new acquisitions system using a KDF9 computer. The work done by Maurice Line in those times and his involvement with other computer systems at Bath University Library and the British Library, are available in a paper written for the fortieth volume of *Program: electronic library and information systems.*

In the 1960's, also the Liverpool University, worked hard this area, specially on the development of a *"machine-readable catalog"*. This new system was able to find a list containing different types of information such as scientific periodicals, medical periodicals and technical periodicals held in twenty-eight libraries.

In this decade, not only academic libraries were changing and improving their way of work. Several changes were also taking place in the public libraries. An example of this change, was the reorganization of the London boroughs, in 1964. This change provided the opportunity to review several systems used to deal with loans, and also gave the possibility to merge several forms of library catalogs. Between 1965 and 1968, Camden Public Libraries produced a catalog on line-printer paper with input on eighty-column punched cards (with two cards per title). At this time, the other major public library involved in this experimental phase was West Sussex. The work developed by this public library which involved the services of the computer firm, Elliott Automation, who was encharged by the development of a catalog and a location index.

In those early days, there were many challenges and problems to solve for those who were working on the first computer-based systems:

- Computers were large and expensive and were owned by the parent authority;
- Programmers were required to write the appropriate software for each application;

- Programs were often written in machine-code language, i.e. the specific computer language for the particular computer, as general programming languages, such as Algol, Basic, COBOL, Fortran, were all new;
- The computer technology of those days was not always appropriate for the jobs;
- Computer developers thought they knew what library staff required;
- Library staff were not always too sure about what was possible and adequate.

1970's - Local Systems

During the 1970s, integrated library systems were often mentioned as *"library automation systems"* or *"automated systems"*. Those systems have been part of several university computing systems and sometimes they were seen as *"old technology"*.

As we saw, in the previous section, before the emancipation of computers, academic and public libraries were using a card catalog to index their holdings. Computers were very important and useful in order to automate several tasks, like keeping up-to-date the card catalog, validate the checking out and checking in of books, generating statistics and reports, managing the acquisitions and the subscriptions, indexing journal articles and providing interlibrary loans.

In this decade, there were several libraries beginning to use computer systems successfully. There were many reasons for the increase of the usage of this kind of technology:

- Improvement of computer technology and the rise of minicomputers which could be purchased by the libraries;
- Research and development was increased in this new area;
- Improvement of the communication between librarians and software developers;
- Improvements on the system design and management.

Also in the 1970s, the Office for Scientific and Technical Information (OSTI)[4] started to fund research work in this area. The Library of the Southampton University received financial support from OSTI for its developments and it became also the home for the OSTI-funded Library Automation Officer.

In 1971, among several contributions from this officer, one of the most important was the publication of a new journal in the field of computer-based library systems called Vine: very informal newsletter. Between 1973 and 1974, OSTI continued to fund several projects. It spent 762,900 on grants and contracts on computer-related library and information research projects.

With the creation of new library systems, also new and different approaches were developed. For instance, a basic feature of any computer-based circulation system is to record details about the item on loan and details about who has the item loaned. In the United States, eighty-column punched cards were often used for this purpose. Instead, in the United Kingdom appropriate equipment was developed to provide the possibility to register unique numbers for specific copies of books and for borrowers.

In early 1970s, there was two major manufacturers, that had emerged in market of library automation systems, Automated Library Systems (ALS) and Plessey. In 1967, the first ALS system, was developed and marketed by Frank Gurney. This first ALS system was installed at West Sussex County Library, containing details of book numbers and borrower numbers punched on to *cards*. The information contained in the cards was *read* automatically by a reader, provided for this purpose, at the issue counter. Then the information was copied on to a reel of punched paper tape which was then physically transported to a computer for processing.

In 1971, at Sussex University was installed the first "trapping store" system. This system was an electronic storage device capable of holding book numbers. With this system, requested books could be "trapped" on their return to the library.

ALS went on with the idea of create and develop an alternative to the card-based system. The result was a label-based system which comprised a non-metallic

[4]See `http://www.osti.gov`

label mounted in the back of a book. At this time, Plessey had also introduced in the market some new products. The Plessey Library Pen system was the first light-pen based system used in libraries for reading barcoded labels placed in the books and on borrower cards.

In 1972, Camden Public Libraries installed a Plessey system at the Kentish Town branch and other public libraries at Luton, Oxford and Sutton were also early adopters of this system. Some libraries developed very complex numbering systems to enable analyses to be made of stock issues as well as by type of borrowers. For accurate recording of who had what out it was necessary to ensure that the numbers read by these various devices were absolutely accurate and to aid this each number had a Modulus -11 check digit as its last digit.

In the 1970's, there was three ways[9] in which information was processed by a computer system:

- **Batch processing** – where jobs to be carried out by the computer were processed one after another. In this way, there was a linear flow through the system and one job was finished before another was started. This way of processing was suitable for library jobs such as catalog production, production of order notes to send to booksellers, or listing periodicals held in a library.

- **Online processing and time sharing** – where a member of the library staff would communicate directly with the computer via a teletypewriter (or similar machine) and the computer would "share its time" between several online terminals.

- **Remote job entry** – which was a linking of batch and online processing as an online terminal would be used to enter a job into a queue of jobs to be batch processed by the computer. This method saved the physical transportation of data from the library as an electronic link could be made via a suitable network connection.

In the 1970's, the major development that affected computer-based catalog systems was the creation of MARC (machine-readable cataloging). The birth of

MARC is frequently associate to a report about automation of the Library of Congress (LC) in 1963. This report concluded that the bibliographic system within the Library of Congress could be automated in ten years. After the decision of OSTI to finance the development of machine-readable bibliographic record, there was close collaboration between the British National Bibliography (BNB) and the LC in the creation and development of this bibliographic format. Between 1968 and 1974, experimental magnetic tapes holding standardized bibliographic records in the MARC format, about items published in the UK, were available and some twenty libraries received them. In 1974, the BNB had become the British Library Bibliographic Services Division and a number of services based on MARC records were provided. A software package known as MERLIN[8] was under development within the British Library for online book ordering and acquisitions, lending and cataloguing using MARC.

In the end of the decade, a large number of local computer systems developed in libraries were changing, mainly with the help of funds from OSTI. The idea was to create co-operative systems where it would be possible to share resources. In the 1970s, there was several examples of successful library management systems, using typically separate applications for different goal, such as cataloging, circulation control and serials control. But there were spotted also problems like the followings:

- **Hardware** – failure of hardware suppliers to provide the necessary items in working conditions, with the agreed time-scale, the agreed price and being appropriate for their particular function;
- **Software** – several problems found when software was not adequately designed, implemented, tested and documented;
- **People** – the computer systems for the library may not have been designed with the real needs of the library's users. There was also examples of lack of communication between computer developers and library staff, and also

between the person in the library involved with the new computer system – and the rest of the library staff who may not have been so interested in the new system;

- **Financial** – inadequate financial resources for acquiring appropriate hardware, developing software, educating and training staff, planning, designing and implementing system were all possible problem for libraries.

1980's - Turnkey Systems

By the end the of the 1970s and early 1980s there was several developments in terms of computer hardware with minicomputers, specially from manufacturers such as the Digital Equipment Corporation[5], Hewlett Packard, Prime and Texas Instruments as well as microcomputers like Apple, Commodore PET[6] and the IBM PC[7]. In general there was a great decrease in the physical size of this hardware, an increase in processing speeds and storage capacity as well as a decrease in cost.

A particular development resulting from this was the rise of what were known as turnkey systems where the hardware and the software was supplied as an integrated package. Such solutions became common – particularly for circulation control systems. The advantages offered included:

- Little expertise required on the part of the library staff;
- Usually a firm contract price and a predictable delivery date;
- Control of the computer system is within the library;
- More chance of reliable performance as the system would have been tried and tested elsewhere.

[5] http://vt100.net/timeline/1960.html
http://www.computerhistory.org/brochures/companies.php?alpha=d-f&company=com-42b9d67d9c350

[6] http://www.commodore.ca/products/pet/commodore_pet.htm

[7] http://lowendmac.com/orchard/06/ibm-pc-5150-origin.html
http://www.old-computers.com/museum/computer.asp?c=274

Some of the turnkey stand-alone systems were developed by the co-operatives, some by the organizations involved with data collection devices and some by computer compa- nies. Many of these turnkey systems provided a short entry catalogue so that a link could be made between the number of an item being loaned and some bibliographic data for that item. Examples of producers of turnkey systems include:

- ALS – the ALS System 5 was a turnkey system which was first used in Derbyshire County Library in 1979 and subsequently in Hertfordshire County Library.
- BLCMP – the BLCMP developed a stand-alone turnkey system known as CIRCO. Loanable items were usually labelled with Telepen bar-codes and the bibliographic record was a subset of the full MARC record. The first CIRCO system was installed at the City of London's Barbican centre in 1982 with further systems being installed at the polytechnic libraries of Manchester,40 Middlesex and Portsmouth.
- CLSI – the US firm, CL Systems Inc., developed a system known as the LIBS 100 which, by the early 1980s, was being used in about 450 libraries in Canada, Northern Europe and the US. Coventry City Library and Coventry (Lanchester) Polytechnic wished to implement a joint turnkey system in the 1980s and Manson describes the LIBS 100 system that was implemented there.
- Geac – the Geac Computer Corporation of Canada developed a turnkey system which was first used at the university libraries of Guelph and Waterloo in Canada in 1977. Several librar- ies in the UK decided to implement a Geac system in the early 1980s and Young and Stone describe the replacement of the ALS card-based system at Sussex University Library with a Geac system.
- Plessey – in late 1980 Plessey launched its stand-alone turnkey system, known as the Module 4 library management system, having tested a prototype at Calgary Public Library in Canada in the late 1970s. Kent County Library in the UK installed a module 4 system for use in its twenty-six branches in 1982.

With the turnkey systems described previously it was possible for users to search in the library's catalog database in order to verify if a desired item was held in the library. In these systems, the user was informed about the location of the desired book and if the catalog system was linked to the circulation system (as many were), the user would know if the book was currently available for loan or not.

These first generation OPACs were often mentioned as "phrase indexed" or pre-coordinate OPACs. They provided access via author, title (as a phrase), or class mark in a way similar to the COM fiche catalogs of the 1970s.

Derived, or acronym, keys were also used as a search mechanism or a combination of author/title information might be used. These OPACs were good when searching for a known item (i.e. when the author and/or title was known). The next generation of OPACs were based on the information retrieval techniques developed by the online search services, such as Dialog, in the 1970s and were also known as keyword or post-coordinate OPACs.

Access points of this second generation of OPACs were words from the title, subject headings or author fields. Search statements could be compiled by linking the search terms using boolean operators. Some models of the second generation of OPACs had two levels of user interaction: a simple one for inexperienced or novice searchers and another more advanced for more experienced searchers.

OPAC's became very popular in libraries. A special edition of *Program*, in 1986, was dedicated to OPACs and included papers with relevant developments in Australia, North America as well as in various libraries in the UK. This edition provided also a general overview of online catalogs and user reactions to them.

The major development associated to OPACs in the 1980's was the creation, by the Computer Board for the Universities and Research Councils, in 1984, of the UK's Joint Academic Network (JANET). Using this network it was possible to search an OPAC remotely. It was produced a booklet giving details about OPACs in the UK that were available via JANET. This booklet was updated periodically in the 1980's, at Sussex University Library.

Since the early 1980's, microcomputers had been used in libraries. A series of six papers about the usage of microcomputer in the library was published in the journal *The Electronic Library* (which was released in 1983 and published by Learned Information in Oxford). The fifth of these series was about the usage of microcomputers for circulation control and serials control. At this time, the basic requirements for any library management system to be described as "integrated" were:

- Provide consistency and integrity of data across all applications. For example, changes of data in a catalogue record would be reflected in the databases supporting the circulation and acquisitions systems;
- Transaction, such as placing an order or recording a loan should update the "status" of the item – which would be viewed through the OPAC;
- There should be easy to move between the different functions of the system.

In 1986, a buyer's guide about integrated library management systems was produced by Juliet Leeves. This guide was compiled under the "sponsor" of the Centre for Catalogue Research at Bath University with funding from the British Library Research and Development Department[8] (BLRDD) and in collaboration with the Library Technology Centre[9] (LTC) from the Polytechnic of Central London. The LTC had been established in 1982, with funding from the BLRDD, with the followings purposes:

- provide demonstrations of the wide range of software systems that might be used in the libraries;
- answer specific enquiries and provide advice for library staff;
- run relevant workshops and seminars;
- disseminate information via the journals Vine (which it had taken over from Southampton University) and Library Micromation News.

[8] the successor of OSTI.
[9] later known as the Library and Information Technology Centre (LITC)

By the end of the 1980's, integrated library management systems were available for a variety of housekeeping function using different types of computer, including microcomputers. These systems typically provide modules for:

- Cataloguing materials (some of them using MARC records imported from an external source);
- Providing access to the catalogue for users – OPAC;
- Circulation control;
- Acquisitions and order processing;
- Serials control (possibly);
- Interlibrary loans (possibly)

From 1990s until nowadays...

Since the end of 1980's, several improvements have been done on the area of Integrated Library Systems, specially in terms of usability. Instead of having separate applications, for different tasks, library staff can use a single application with multiple functional modules. During the 1990's, emerged the linkage between bibliographic citations and the content of what they represent.

With the growing up of the WWW, ILS vendors offered more web-based functionalities. ILS systems has now available web-based portals where borrowers can log in to view their account, renew their loans and be authenticated to use online databases.

In one of the first papers about ILS to be published during the 1990's, in the UK, J.A. Arfield[8] describes the environment at Reading University Library and wish to "turn-off" a system shared between different libraries and move to an integrated library system, controlled locally. Reading University Library was a member of SWALCAP[10] which provided shared cataloguing and circulation services to several academic libraries in the UK since 1979.

[10]originally standing for the South Western Academic Libraries Cooperative Automation Project.

However, equipment was becoming very unreliable and staff at Reading University Library felt that the SWALCAP service was unable to support the increasing number of terminals that were necessary for the users. This situation was replicated in other academic and public libraries in the beginning of the 1990's. Many libraries moved over, or migrated, to integrated library management systems. The decline in the number of customers of the shared services resulted in the decision by SLS (SWALCAP Library Services) to withdraw this service.

Most ILS are now integrated system. The data is only held once by the system and is available to be used by all modules and functions. This procedure has an obvious advantage, the result of searching with an OPAC, can inform the user about the number of copies of each title who are held by the library, where they are located, if they are out on loan or not, and if so when they should be returned. The libraries in the beginning of the 1990's, academic or public, dealt primarily with printed materials such as books, reports, scholarly journals and with "non-book" materials such as films, videos and CDs.

However, by the end of the 1990's the huge impact of the Internet and the World Wide Web meant that staff in libraries were involved not only in the management of collections stored physically in shelfs of their library but were also involved in providing access to a very large range of digital information sources with potential relevance to their users. This mixture of providing access to printed material and digital collections was referred as an hybrid library.

At this time, for many library staff, ILS were their first experiences with computers. In order to learn how to use these new system, library staff had to follow a training "Information and Communications Technology". This training was part of Electronic Libraries Programme (eLib) in the UK s academic libraries. With this programme, library staff became more qualified to work with ILS.

For majority of the libraries the big challenge related with ILS was not necessarily the choice of a new system, but the migration from one system to another one. Graeme Muirhead explain on his book *"Planning and implementing successful system migration"* a number of case studies written by library staff from different

types of libraries, describing their experiences about system migration[20].

By the end of the 1990's, various improvements had been done in the automation of library operations. The following list shows some of the most relevant achievements[21] in the development of ILS in the 1990's:

- **Technological developments** – It was common for the first ILS to be developed with their own operating systems. However, during this decade several suppliers decided to change their strategy and started to create and develop systems that ran on Unix. Commonly, like for the case of operating systems, several of the first ILS were designed with a database management systems, only for their own. During the 1990's, there was a move away from these development strategies to relational database management systems. Examples of this are Ingres (used by Galaxy 2000), Informix (used by Unicorn), Oracle (used by ALEPH and Olib) and Sybase (used by Horizon and Talis). Also in the 1990's, there was another relevant development: the adoption of the client-server architecture by ILS. We this new model it was possible to split operations between client and server, improving the quality of ILS.

- **Self service** – The installation of self-issue and self-renewal machines in libraries was one of the most relevant development, during the decade of the 1990's. With these machines it was possible for borrowers to check in and check out their own books. 3M developed this type of systems, and one of these first system was installed in the library at the University of Sunderland. For the successful implementation of this system, it was fundamental four P's: preparation, publicity, position and persuasion[22]. A edition of *Vine* was published in 1997 about the usage of self-service systems in libraries. At this time, library staff accepted the benefits of the new system. On busy days queues had reduced quietly. However, when it was normal day, a quiet day, borrowers preferred the human approach to issuing and returning materials.

- **Messages to users by e-mail or text** – With the growing of the web and some web-based technologies, many borrower started to have access to email.

So several ILS manufacturer decided to incorporate on their systems, the possibility to notify borrowers, using these new technologies. This use very useful for sending overdue letters, alerts about reserved books or other type of information. One of the first ILS to provide this functionality was ALEPH 500, witch had the possibility to store several address, in the borrower record, including e-mail address.

- **Improvement of accessibility via the OPAC and use of the Z39.50 protocol** – The design of OPAC's has always been done focused on final users, borrowers. During the 1990's, the development went from menu-based systems to forms filled in web pages. All these developments have been intended to be straightforward to use. To improve the efficiency of OPAC's, specially when a search was perform, it was necessary to find a solution. The answer was MARC. The 856 field of MARC allows the inclusion of a URL into the bibliographic record. By the end of the 1990's some OPACs were using this to provide links to digital objects. Another important development related to OPAC's was the Z39.50 protocol. By the definition of Dempsey[23] it is *"a retrieval protocol which allows client programs to query databases on remote servers, to retrieve results and to carry out some other retrieval-related functions"*. This protocol is quiet common in libraries. It allows to shared material between several libraries, in order to respond to the borrowers demand.

- **Catalogue record provision** – The majority of ILS provide the possibility to import of bibliographic records, usually in MARC format, records from external sources. Although not all ILS use MARC format for internal processing of records. Usually, in these cases, they include the possibility to input or output records in MARC format. In the UK, some cooperatives of libraries developed large databases containing MARC records. Many of these records have now been incorporated into the OCLC database in the US and made available internationally.

As conclusion of this section, it is possible to say:

"Today's ILS is a multi-function Web-based multimedia content information management system, generally built on a standard relational database structure. While the system architecture remains grounded in bibliographic citations presented via structured indexes, the basis of these indexes is moving beyond the MARC[11] *fields designed for text information to include metadata descriptions for multiple digital file formats and content"*[7].

2.1.2 Integrated Library System and the Open Source

An ILS can be considered as key piece in terms of infrastructure in a library[24]. It offers to libraries the possibility to provide a catalog and also manage several workflows, related with the different operations present in libraries. Libraries are constrained in the amount of investment they can make on new systems. The marketplace is dominated by a restricted number of major vendors and once acquired, systems are retained for a considerable period. Although few would characterize their current automation system as perfect, libraries rarely leave current systems out of dissatisfaction with support or functionality. Migrations are just too costly. Library system companies pulled out all stops to retain customers and entice them to migrate to their replacement systems. These replacement systems need to offer tools to both help manage the electronic content they purchase and create content from digital products.

Whilst vendors are trying to change their systems to meet these demands there is also a questioning about whether one ILMS can offer all these functions. Certainly add-on products are increasing in their use. Integration, metasearching, open source software and the Internet are all pushing the ILS in new directions. Investment in standalone products for linking and digital management accounted for nearly 13ILS market, in 2002[24].

Nowadays, with all the economical and financial problems all over the world, the open source software can be a viable and efficient solution, in several areas of

[11] The MARC formats are standards for the representation and communication of bibliographic and related information in machine-readable form. See `http://www.loc.gov/marc/`

our society. In the case of libraries management, for those who have budgetary constraints, it is possible to find a complete open source alternative, available for all types of operating systems. Although, even using open source software, libraries may have to spend some money on training staff in how to use this new kind of technologies, or sometimes they may need to hire a developer to implement some specifics requirements. In overall, a library can save money, in terms of software costs and licensing fees.

Several public libraries have been investigating "open source" ILSes. However, the percentage of libraries that would seriously consider implementing an open source ILS is still small. Nowadays, it is difficult to say exactly how many libraries are using an open source ILS. Several libraries that have downloaded the software don't to use it. The number of libraries and consortia that have selected open source ILS software is estimated at more than 500 worldwide. Part of these libraries have contracted a commercial company for support services. The motivation of libraries to consider an open source ILS has two big reasons. First, the financial interest and the possibility to reduce the charges with the maintenance of this kind of application. Second, the desire to have a system more close to their requirements.

2.2 Integrated Library Systems and Digital Libraries

It is common, for some people, to confuse Integrated Library Systems with Digital Libraries. This kind of confusion happens specially, when an ILS is mentioned as a Library Management System (LMS). A Digital Library is a type of information retrieval systems, typically used to manage collections of documents in a digital format. CDS Invenio, Greenstone, Dspace and Eprints, among others, are examples of digital libraries.

According with the Digital Library Federation[12], "Digital libraries are organizations that provide the resources, including the specialized staff, to select, structure, offer intellectual access to, interpret, distribute, preserve the integrity of, and ensure the persistence over time of collections of digital works so that they are readily and economically available for use by a defined community or set of communities".

With the development of BibCirculation, CDS Invenio will become an application 2 in 1, being a digital library providing, at the same time, ILS functions, like we have nowadays, for example, in Greenstone. This can be a big advantage for CDS Invenio in comparison with other digital library softwares. To have a better idea about digital libraries, let's take a look to some examples.

2.2.1 Dspace

In the open source community, DSpace is a very popular system for digital libraries. It is written in Java and JSP, using the Java servlet API and supporting PostgreSQL and Oracle. The development of Dspace was started by the Massachusetts Institute of Technology (MIT) and Hewlett-Packard. The first release of Dspace was in 2002. The development and error reporting are hosted by SourceForge (sourceforge.net). Nowadays, there are several universities and institutions, from countries all over the world, giving their contributions for this project. Its development is financially supported by DSpace Foundation. DSpace permits to create digital repositories. These repositories can contain several types of documents from institutions. The data is stored in the system with a unique identifier that contains metadata. Dspace has support for the metadata scheme Dublin Core and it uses the Corporation for National Research Initiatives (CNRI) system to assign the persistent identifiers. It supports the OAI-PMH 2.0 and OpenURL. With Dspace it is possible to export data to XML format or to the Metadata Encoding and Transmission Standard (METS) format.

[12]See `http://www.dlf.org`

2.2.2 Eprints

Eprints is a complex system, based on web technologies, widely used all over the world, developed by the University of Southampton in UK, and available under the GNU license. Its primary purpose is to build institutional repositories for various types of documents such as common literature, but its primary focus is on scientific data. The whole system is easy to configure and it also offers paid services, such as training, management of implementation project and technical support. In terms of standards Eprints supports the system EPrints OAI-PHM and the metadata have their own inner format. Eprints allows the importation of data from documents on XML format and some external resources such as PubMed XML. The export of data is also possible in several formats, XML, RSS, DublinCore, METS.

Eprints provides administration of user accounts, but assigning of user rights is not very elaborated because initially it was aimed only at publishing scientists. It enables search using the interface as well as in the metadata. Eprints indexes text files and other common formats such as PDF. It also allows browsing the logical tree structures, its intro structure is the same as in the Library of Congress, but it can be modified. The interface further enables registration of new users, to inform them about news and to provide them with feeds and e-mail alerts to keep them up to date. Administrator interface allows configuration and control of the whole system. Eprints is very sophisticated system but the upload of individual items is very complex, sometimes difficul to use and time consumed is usually hight.

2.2.3 Fedora

Fedora[13], like the two digital libraries mentioned before was also created in the university environment – at Cornell University and University of Virginia. It all started as a research project in 1997 which result was published on the web of Cornell University in 1998. In 2001 both universities started to cooperate and received financial contribution for further development from the Melon Foundation with the assignment to develop a universal digital library on the basis of the web

[13]Flexible Extensible Digital Object Repository Architecture

services and XML. In 2007 both universities established an organization Fedora Commons, which now takes care of the development of the joint system.

The Fedora system supports various standards, OAI-PHM, exports to METS formats and its own internal format FOXML, the descriptive metadata are stored in the Dublin Core format. Although the core of Fedora system is very advanced, at present it is not a complex library system ready-to-use. It is only a platform that further has to be programmed at quite a higher cost and with great effort. When operated, the higher cost must be taken into account due to the platform independence of the system, because it is more demanding for hardware sources than other systems.

2.2.4 Greenstone

Greenstone is a software, published under the GNU/GPL license, for constructing and presenting collections. Each collection may have thousands or millions of documents, with different types: text, images, audio and video. Usually digital library created using Greenstone will contain many collections, individually organized. The maintenance process is easy and each collection can be augmented and rebuilt automatically.

There are many ways to find information in Greenstone collections. It is possible to search for particular words that appear in the text, or within a section of a document. It is also possible to browse documents by title or by subject. To see several examples of Greenstone collections, you can visit The New Zealand Digital Library website (www.nzdl.org). Greenstone constructs a full-text indexes from the document text. Indexes can be searched for particular words, combinations of words or phrases. The results are ordered according to how relevant they are to the query.

In the majority of collections, associated with each document, we can find descriptive data like author, title, date and keywords. This descriptive information is called metadata. Metadata is used as the raw material for browsing indexes. It must be either provided explicitly or derivable automatically from the source doc-

uments. The Dublin Core metadata scheme is used for most electronic documents, however, provision is made for other schemes.

Greenstone creates automatically all index structures from the documents and supporting files. If a new document, with the same format, become available, it can be merged automatically into the collection. For several collections this is done by processes that awake regularly, search for new material and rebuild the indexes. Documents come in a variety of formats and are converted into a standard XML form for indexing by "plugins". Plugins distributed with Greenstone can process different types of formats: plain text, html, word and pdf documents, and e-mail messages. New ones can be written for different document types. To build browsing structures from metadata, an similar scheme of "classifiers" is used. These create browsing indexes of different kinds: scrollable lists, alphabetic selectors, dates and arbitrary hierarchies.

Unicode, which is a standard scheme for representing the character sets used in the world's languages, is used on Greenstone. This allows any language to be processed and displayed in a consistent way. Grenstone collections are accessed over the Internet or published, in precisely the same form, on a self-installing Windows CD-ROM. Compression is used to compact the text and indexes. A Corba protocol supports distributed collections and graphical query interfaces.

2.2.5 Comparison between Digital Libraries

The following gives an overview and compare the different functionalities and specifications about the digital libraries mentioned before.

2.3 Integrated Library Systems - Overview

In this section, I will show different examples of ILS, some of them commercial and other open source.

	Invenio	**DSpace**	**Eprints**	**Fedora**	**Greestone**
Year of creation	1993	2002	2000	1997	1997
Support provided	Yes	Yes	Yes	Yes	Yes
Development organisation	CERN, Switzerland	DSpace Foundation, Massachusetts, USA (MIT/HP)	University of Southampton, UK	Cornell University, University of Virginia, USA	University of Waikato, New Zealand
Programming language	Python (PHP, Common LISP)	Java	Perl	Java	Perl, GDBM, MG
Operating system and server	Unix, Apache	Unix and Windows, Apache	Unix, Apache	Unix, Apache	Unix and Windows
Database	MySQL	PostgreSQL, Oracle	MySQL	MySQL, Oracle	Not necessary
OAI-PMH	Yes	Yes	Yes	Yes	Yes
Z39.50	No	No	No	No	Yes
Metadata format	MARC21	Dublin Core	Dublin Core	Dublin Core	Dublin Core
Identifiers	their own	CNRI Handle	their own	their own	their own

Table 2.1: Comparison between digital libraries

2.3.1 KOHA

Koha is the first Open Source Integrated Library System in the world. It is distributed under the GNU General Public License. Koha was initially developed, in 1999, in New Zealand, by Katipo Communications and his first deployment took place in January of 2000, for the Horowhenua Library Trust. Its development, is currently maintained by a strong community of software developers and libraries, who are working togheter, in order to achieve, their goals. Koha is written in PERL and requires MySQL database, Apache HTTP Server and can run with Linux or

Windows. It provides diferent functional modules like acquisition, cataloguing, serial control, OPAC and circulation. It is also possible to find other features like MARC support, Z39.50, barcode, RSS feeds, web interface and multi branch library support. In 2006, Koha was updated 3 times with significant changes. User support for Koha is available on the documentation website (www.kohadocs.org), Wiki, mailing lists and open source vendors. Koha has nowadays more than 100 users registered.

2.3.2 OpenBiblio

OpenBiblio is an automated library system written in PHP, using the LAMP stack. It provides several functional modules such as OPAC, circulation, cataloging and staff administration and support for UNIMARC. It is also possible to find an online demo of OpenBiblio. The last release was in 2007, since then, there was no significant development. OpenBiblio needs the contribution from users and developers, to assure the survival of the project.

2.3.3 Emilda

Emilda is developed, since 2000, by CompanyCube[14], a Finish software company, under the GNU General Public License. The initial system was conceived and developed, in PHP, with the assistance of many school libraries. Since 2003, Emilda is supporting tipical standards, including MARC and Z39.50 protocol for ILL. It is XML-based and can be run on Windows and Linux. The circulation module and patron access catalog modules were introduced on June 29, 2005. Emilda uses the Zebra Server from Indexdata as a backend server. The source code and documentation are available online in English. It is also possible to experiment an online demo. Emilda was in use at 14 finish school libraries in 2008.

2.3.4 PMB - PhpMyBibli

PMB (PhpMyBibli) was created in France, in 2002, by Francois Lemarchand. It provides several modules like circulation, acquisition, cataloguing with UNIMARC

[14]formerly Realnode Ltd

support,OPAC and a SDI[15] system. The installation process and the maintenance of PMB is easy in Windows and Linux in comparison with other open source ILS. It is written in PHP, using Apache HTTP Server and MySQl database. PMB provides also, for the library staff, an friendly graphical interface for database back up, system maintenance and import and export of bibliographic records. With this complete set of tools, it is possible to librarians to maintain the ILS without the help of system administrator. Other important features are the import and export of bibliographic records using different formats, Z39.50 support for ILL, barcode generator, serial control, multi-language support and detailed documentation for users and administrator.

2.3.5 EverGreen

Evergreen is an Integrated Library System, licenced under the GNU General Public License, and initially developed and maintained by the Georgia Public Library Service for the PINES[16] Program, a consortium of 270 public libraries. The develoment of Evergreen started in 2005 and it appears as the answer for the specific needs of PINES. At this time, any kind of ILS (proprietary or open source) was good enought to cover all the requirements and achieve all the objectives of PINES. For that reason, Evergreen was one of the first Open Source Library Automation System conceived from scratch for a large-scale deployment in a public library consortium. Nowadays, Evergreen is maintained by Equinox Software, a company formed by the original development team of Evergreen. This new company provides services like development, migration, support, training and consultation. Evergreen provides several modules like cataloging, circulation, statistical reporting and OPAC. Modules for acquisitions, reserves and serials are on development. It has also support for Z39.50 and MARC. Evergreen is mainly written Perl and some few sections were rewritten in C. The OPAC module was developed using JavaScript and XHTML and the interface (for library staff and users) was written in Mozilla XUL (XML + JavaScript). Python was also used for the internationalization. It runs on Windows and Linux, using PostgreSQL as

[15]Selective Dissemination of Information is mechanism used to keep an user informed about new resources related specific topics.

[16]Public Information Network for Electronic Services

database.

2.3.6 GNUteca

GNUTECA is an Open Source ILS, published under the GNU General Public License, and developed since 2001. It is highly popular among public and academic libraries, in Brazil. It has modules for circulation, cataloguing, serial control, ILL and OPAC. GNUTECA supports MARC21 and CDS/ISIS[17] conversion. The documentation of this project is available on portuguese and french. GNUTECA runs only Linux, using Apache HTTP Server, PHP and PostgreSQL.

2.3.7 ALEPH 500

ALEPH500 is integrated library system created by ExLibris. It is a market leader in the automation of libraries and research centers providing an the efficient, user-friendly tools and workflow support they need. Based on industry standards such as OpenURL, XML, OAI, LDAP, ISOILL, and RFID, offers the ultimate in resource-sharing capabilities, full connectivity, and seamless interaction with other systems and databases. Built on an Oracle database, ALEPH 500 offers full Unicode support, employs system-wide XML technology, and offers third-party integration through an XML gateway as well as standard protocols such as Z39.50 and ODBC.

2.3.8 Comparison between ILSes

The following table shows the comparison between several ILSes and gives an overview about their main functionalities and specifications.

[17] CDS/ISIS is a software package for generalised Information Storage and Retrieval systems developed, maintained and disseminated by UNESCO.

Table 2.2: Comparison between ILSes

	KOHA	OpenBiblio	Emilda	PMB	EverGreen	GNUteca	ALEPH 500
Operating Systems	Linux	Linux	Windows and Linux	Windows and Linux	Windows and Linux	Linux	Windows
Database	MySQL	MySQL	Qual	MySQL	PostgreSQL	PostgreSQL	Oracle
Programming language	Perl	PHP	PHP	PHP	Perl, C and Python	PHP	-
Metadata	MARC	UNIMARC	MARC	UNIMARC	MARC	MARC21	MARC
Developed by	Katipo Communications	-	CompanyCube	François Lemarchand	Equinox Software	-	ExLibris
Year of creation	1999	-	2000	2002	2005	2001	-
Circulation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Acquisitions	Yes	No	No	Yes	Yes	No	Yes
Serial control	Yes	No	No	Yes	Yes	Yes	Yes
Cataloguing	Yes	Yes	No	Yes	Yes	Yes	Yes
OPAC	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ILL (Z39.50)	Yes	No	Yes	Yes	Yes	Yes	Yes

Chapter 3

BibCirculation Development

In the chapter 3, I will explain the process of creation and development of BibCirculation. This chapter is fundamental to understand my options and my implementation strategy for this project. It describes the technologies used, the software engineering model followed during the development, the requirements analysis and all the work developed in terms database and Graphical User Interface (GUI) design. This chapter gives also an overview of all the implemented features of BibCirculation, explaining the role of each one and the interaction with the other modules of CDS Invenio. Finally, there is also an explanation about the synchronization process with the ILS used nowadays by the CERN Library and a overview of external sources of information used to help the implementation and development of BibCirculation.

3.1 Development model and strategy

The development of BibCirculation is based on a software engineering development model. A software engineering development model is a virtual structure, more or less flexible, who is created with the purpose to guide the development of a software application. It can be also called software life cycle or software development process. There are several models for this kind of processes, each one describing a different approach to a variety of tasks, problems or activities, who are contained in the process.

On the development of BibCirculation, the followed development model was the *Waterfall Model*. This model is, usually, composed by the following different steps[17]:

- Requirements analysis
- Design
- Implementation
- Integration
- Testing
- Installation

- Maintenance

After the end of each step, the process goes to the next one. In the following figure 3.1, you can see the relation between each step and the respective transitions.

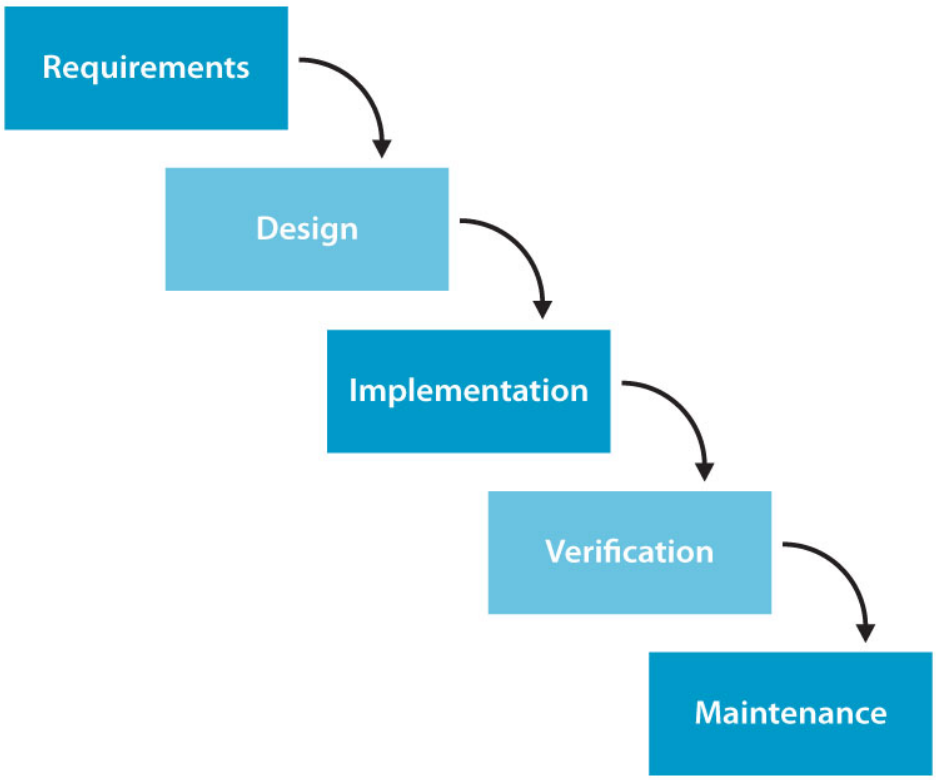

Figure 3.1: The Waterfall model[16].

3.2 Requirements Analysis

The requirements analysis is the first phase of the Waterfall model. Once BibCirculation is following this development model, the creation started by investigating and collecting information about our problem. This first step is perhaps one of the most important step in a software development process. This step should be static, in order to be the base for all the development of the project. To understand its importance, we can say this first step will be the difference between the failure and the success. If we don't pay enough attention when we are doing the requirements analysis, the result can be catastrophic, or even, with no way back, in the future. Sometimes requirements analysis errors are just detected when an application is installed on production. This situation will increase the total cost (not just in terms of budget, but also in terms of workforce) and the duration of the project, creating a delay on the delivery schedule.

The requirements analysis should be saw like an environmental analysis. We need to understand what kind of parameters, needs and variables will be surrounding our application. For BibCirculation, it is really important to not forget that this new module will interact we other modules of CDS Invenio and will be available not only for CERN, but also for other external users and institutions, with a different type of needs. In order, to be able to answer to all kind of requests, BibCirculation should be the most generic possible, but keeping always, a high level of efficiency and performance.

In the particular case of BibCirculation, I started by creating a division. A division between the borrowers and the library staff. This division is very important for the requirements analysis, because each actor will be in a different environment, with different actions, rules and needs. There will be off course a relation between borrowers and library staff, but in terms of software development, they are not together, because they have different specifications.

To understand the needs of the future users and administrators of BibCirculation, we have to watch and study, very carefully, what has been used in the last

years, in the previous system. It is also necessary to understand the usual behavior of users and an administrators, when they are interacting with the system. BibCirculation will replace the system currently used by the CERN library, so we need to keep the good ideas and replace/improve what is not good and useless.

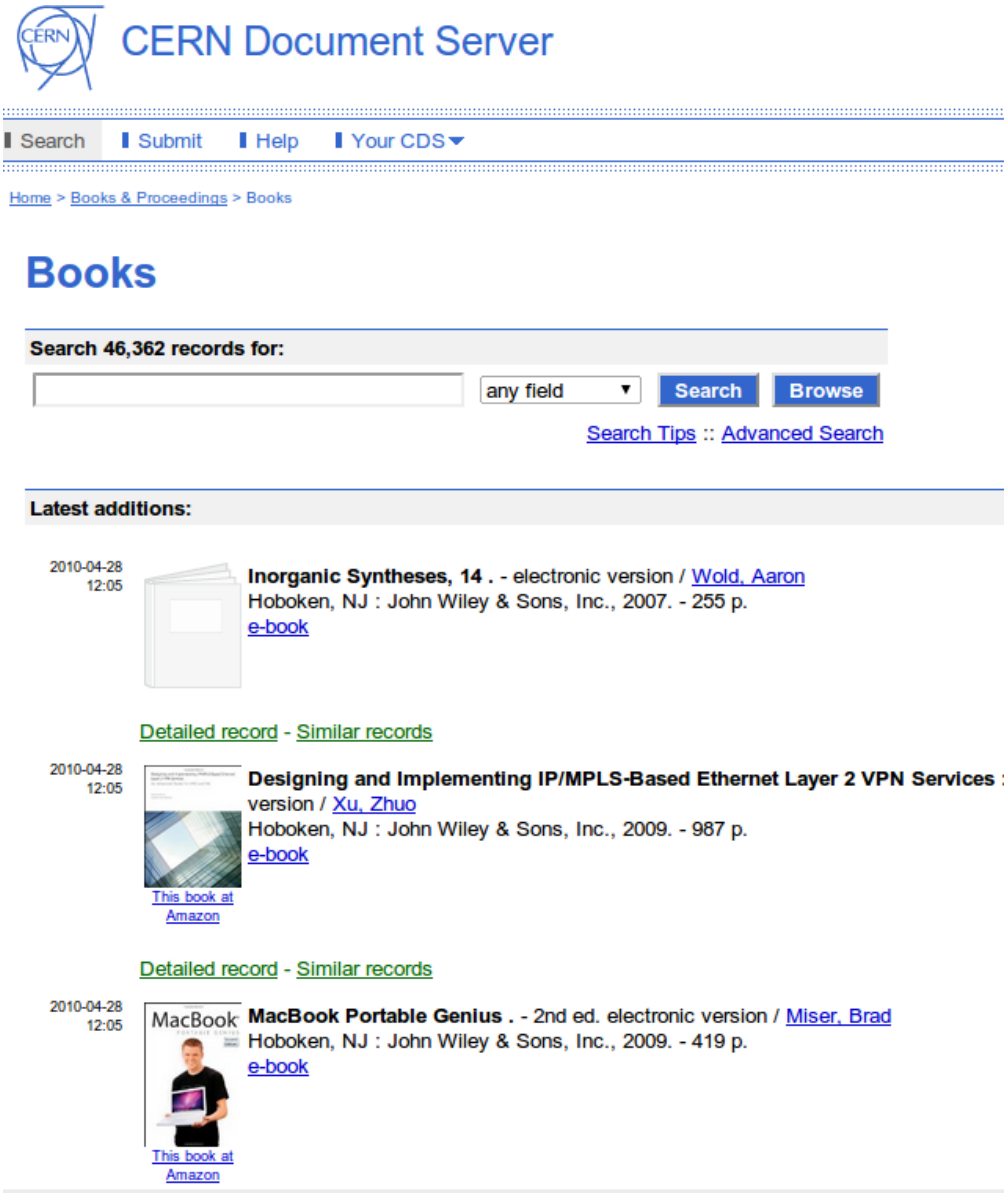

Figure 3.2: OPAC of CDS Invenio – searching books.

With this project, it will be created several functions to automate library operations, like it happens with an ILS. As I mentioned before, an ILS as several modules. With BibCirculation, it will be developed functions for circulation, serials, acquisitions and ILL. Functionalities such as cataloging or OPAC, are already provided by CDS Invenio. Cataloging is provided by BibIndex and BibUpload. OPAC is provided by WebStyle, WebSession, WebAccess and WebSearch.

3.2.1 Understanding librarians needs and demands

When we are creating a new software, we should have two things in mind. First, our software will handle a problem or a set of problems and it will try solve them. Second, our software will be used by someone (an individual person or a company). This subsection will be focused on the second point – the people who is going to use our software. Personally, I guess this is a very important step in the development of a software. We need to spend sometime trying to understand the needs and the demands of the final user. Part of the success of a software application is in this very particular and specific point.

To understand the needs of the CERN library staff, I spend several weeks studying the system used by the library and having meetings to discuss the goals to be reached in this project. All this process is quiet important, because it allowed to me to see how works a library staff and what type of routines they have. If possible, it is an advantage to keep in a new software application similar behaviors and action. This is very helpful for the people who will use the software in the future.

3.2.2 Entity-Relationship (ER) Model

After the requirements analysis, the design of the database – the ER model – is a very relevant component in a software engineering process. In software engineering, the ER model is a conceptual and abstract way to represent data. This model is a database modeling method used to produce a type of conceptual schema or a semantic data model of a system. Our database ER model will contain all the information related with the activities of BibCirculation. For this reason, his

conception and his design is very important and have to be done very carefully. The created ER model will be added to the ER model who already exists in CDS Invenio. The link with the rest of the system will be done by the table `bibrecord`. This table contains the ID of each record present on CDS Invenio. These IDs will be used in different BibCirculation tables such as `crcITEM`, `crcLOANREQUEST` or `crcLOAN`. This second phase has an strong relation, with the requirements analysis. The database ER model is mainly based on the work developed in the previous step. The integration of BibCirculation with all the other modules of CDS Invenio, starts here. The relation with the table `bibrecord` is the beginning of the integration process.

In order, to give a correct answer to all needs, of an application like BibCirculation, an application who will be responsible for library management, it is necessary to create an ER model, who will be able to handle with a specific set of requirements. In this second step, like for the first one, we have to keep in mind, the idea of a generic tool – a tool who will able to deal with different situations in different libraries. The figure 3.3 shows the final BibCirculation ER model.

3.2.3 Graphical User Interface

When we are creating a new application, the Graphical User Interface (GUI)is perhaps one of most relevant parts in the creation process. The layout, is often, a subject who creates many discussions between developers, designers and clients. For BibCirculation, I tried to provide a *clean* interface, without many buttons or many options, just something simple and clear. For this new application, it is necessary to consider two different parts or two different interfaces. One interface for borrowers and another for library staff. Below, I will explain the creation and the development of these two interfaces.

Borrowers GUI

The development of the borrowers interface was based in the previous system and also in comments and improvements suggested during the requirements analysis. For this specific interface, it was kept all the advantages from the previous system,

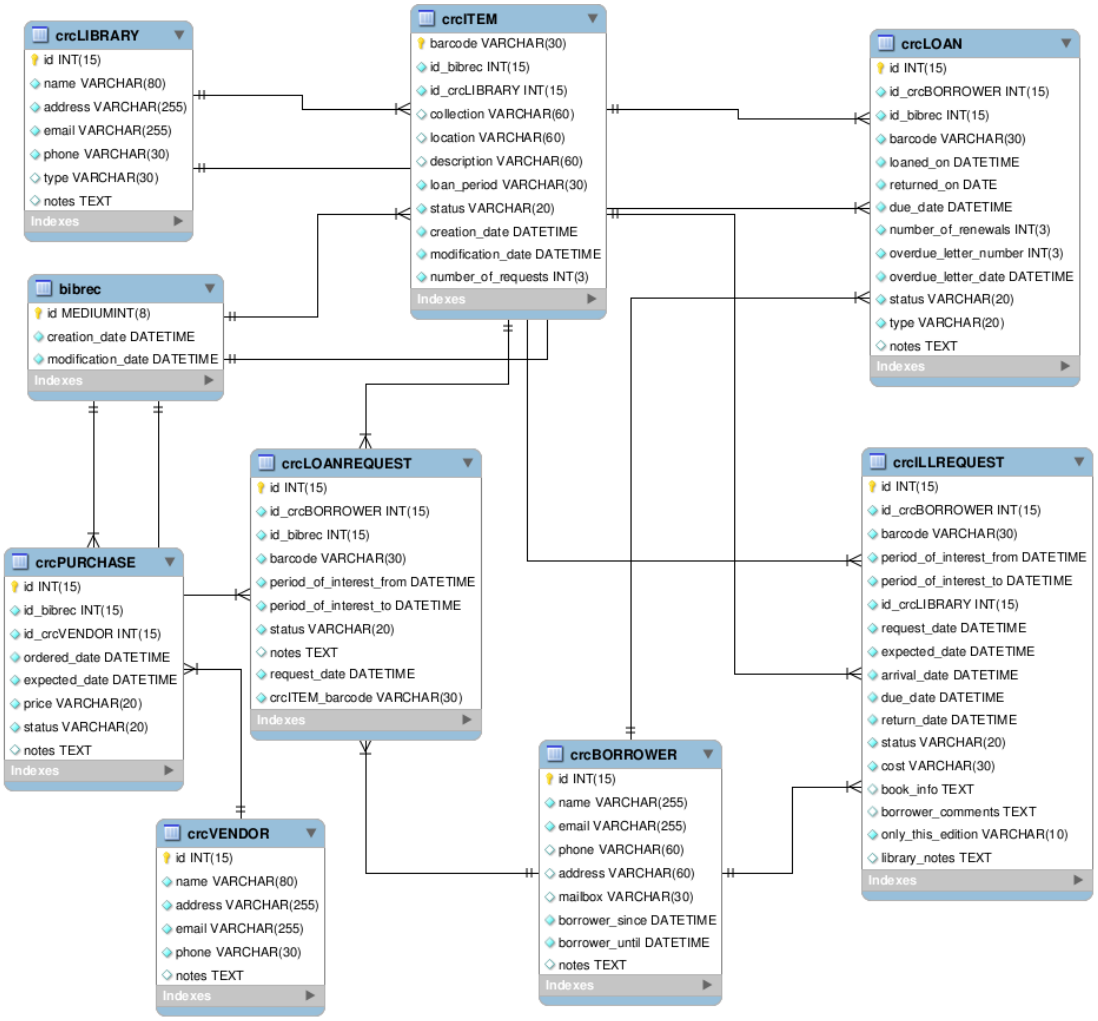

Figure 3.3: BibCirculation ER Diagram.

in order to have the same logic and the same behavior. From the usability point of view, this fact is really important for the future users, because it will be easier and faster to understand the action, they have to do, to reach their goal.

For borrowers, it will be important to provide a nice and usable interface, where they can find some new options in comparison with the previous system. For example, using the circulation module provided by the CERN library it was not possible, for a borrower, to cancel or delete a request. Sometimes this situation

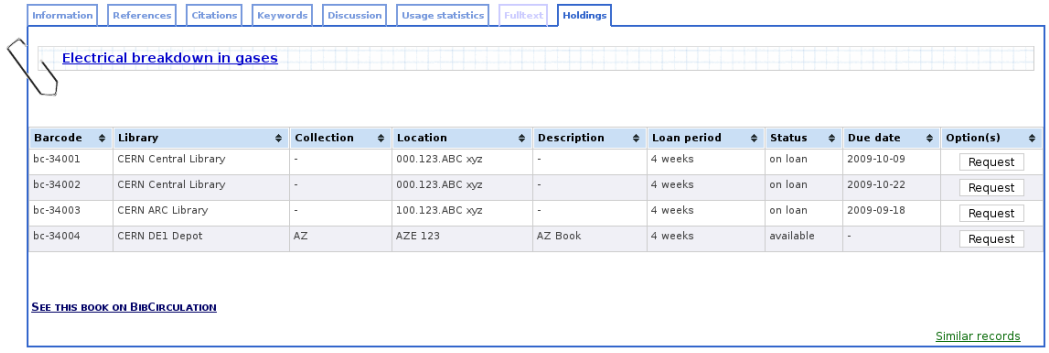

Figure 3.4: Borrower interface - Holdings tab.

was annoying for the borrower. It was necessary to notify the library by phone or email in order to cancel the wrong request.

Library staff GUI

The development of the library staff interface was done very carefully. The design, like for borrowers, was based on comments and recommendations of the CERN Library, and also by studying the system used in the last years. BibCirculation has to provide several functionalities to the library staff. These functionalities has to be organized in an efficient way, in order to avoid waste of time for the person who is trying to perform an action. To create this interface, I tried to understand the needs of the library staff and also the way they work. At the library desk, members of the CERN library explained to me how they performed several actions, giving their opinion about possible improvements and what should kept in BibCirculation. The following pictures shows one of the first versions of the interface designed for library staff. As it is possible to see in the figure, the interface is perfectly integrated with CDS Invenio.

3.2.4 Main features of BibCirculation

As I explained in Chapter 1, the main purpose of BibCirculation is to deal with processes that involve the association between books (items) and borrowers, their relation with the system and support the management of acquisitions and ILL,

Figure 3.5: Graphical user interface for library staff.

providing features usually present in ILS. To understand what is this project, let's talk a little bit more about BibCirculation showing the ideas behind this module and explaining each features if it was an *informal use-case*.

The circulation process is based on the interaction of data from different places – files or dabatases – in the system. In this type of system, the most important information, or base for all the rest, is the information about books (items) and borrowers. Usually, the information about borrowers contains common information like name, address, email, etc and some properties, that can be relevant for the library, like privileges or statuses. These properties can be used by the system to applied, for example, different kinds of lending privileges. The information about books usually contains bibliographic details and also some additional information such as the type of material or the location in the library.

The additional information mentioned previously is defined by librarians. When a new book arrives to the library, the library staff add to this new item, information like the barcode (witch is unique), the lending conditions, the loan period and some other information that depends on the each library.

Management and registration of borrowers

People who wants to borrow material from a library, or to have access to electronic resources like e-books, or research information from a database must first to register as a borrower. Borrowers include usually people with a library's membership or other organizations or institutions that can borrow material. A single record for each borrower is stored in the database. There are three methods used to add borrower records to the database. These methods usually depend on the library and in the capabilities of the library system:

- The borrower can supply information personally or by filling a paper form, which the library staff can create a new record on the system;
- The borrower register online using a web form;
- Files containing borrower data are loaded into the system from external databases.

Data stored in a borrower record include commonly name, address, telephone, email, driver's license or another unique ID number. If necessary or considered relevant, libraries may add for statistical usage, information fields such as age, gender, language or level of study. Similarly to other system records, a borrower record contains fixed-length fields for data such as codes and dates, and a variable-length fields for names, addresses, numbers and notes. The system allocates a unique number to each borrower record as it is saved on the system. Library staff use the number to search for and retrieve a record and the system may use it to create relations with another entity.

With BibCirculation, we can register a new borrower. This is one most basic features of an ILS, but it is also one of the most important. Usually, libraries receive new borrowers. It is essential to collect and register all the information about the new borrowers. This information is highly relevant, because it will be associate with books information, to create request and loan procedures.

In the particular case of BibCirculation, there are three ways to register new borrowers. The first one, and also the most common, is to register someone one in the library desk, when a person demands a book for the first time. The library staff just need to fill a form with the borrower's information and it's done. The second one, appends when a borrower is requesting a book online, using CDS Invenio. When a borrower is requesting a book, BibCirculation will verify if it's the first request of this person. If it's true, BibCirculation will register this person has a new borrower. To achieve this operation, BibCirculation use the borrower's information (CERN id, name, email) contained in the session. The missing information will be retrieve from CERN LDAP. This operation is transparent and the new borrower doesn't know anything about the registration process. The third way merges the two first ways. A borrower arrives to the library desk with a book. The library staff enter the borrower's name or scan the borrower card reader. At CERN this can be done using the CERN card, witch has a barcode. If it is the first loan of this borrower, the borrower's information is retrieved and he is registered on BibCirculation.

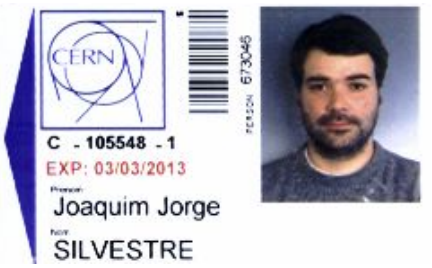

Figure 3.6: CERN card with barcode.

Management and registration of books

A book can be defined by different names in an ILS. Item or holding record are also common names for books. With BibCirculation, I thought that the term books would the best way to explain what we are talking about. Usually, an item is associated to a copy of a title that is in circulation. The item record links directly to a bibliographic record that contains several descriptive fields like title, author, publisher and ISBN. A single bibliographic record can have several

hundred of copies linked to it. For instance, issues of a journal that circulate need different item records. A copy has multiple fixed-length fields that store permanent information like:

- The usual location of the copy (the shelf);
- A code representing a format of a copy, e.g., book, journal.
- A price for assessing replacement costs;
- Statistical codes defined by the library.

Fixed-length fields also contain information generated by the system, for example, coming from updates when a books is returned or a loan is renewed. These information may include the followinf types:

- Date of check out (or return);
- Date item is due;
- ID of the borrower who has checked out an item;
- ID of the previous borrower and check in date;
- Circulation status of the copy;
- Number of renewals within a loan period and total of renewals;
- Number of overdue letters or emails sent;
- Date of the last overdue letter or email sent;
- Total number of loans;
- Checkouts during a statistical time period.

Usually, the system is able to collect fixed-field statistical information to generate reports. Some fields are also editable by the library staff. The fields who are not editable are generated and updated by the system it self. Copy variable-length fields include:

- barcodes;
- volume details for serials;
- local call numbers;
- historical or circulation notes added by the library staff;
- system generated information recording information from transactions.

The following figure shows the interface (viewed by the library staff), create to display items details.

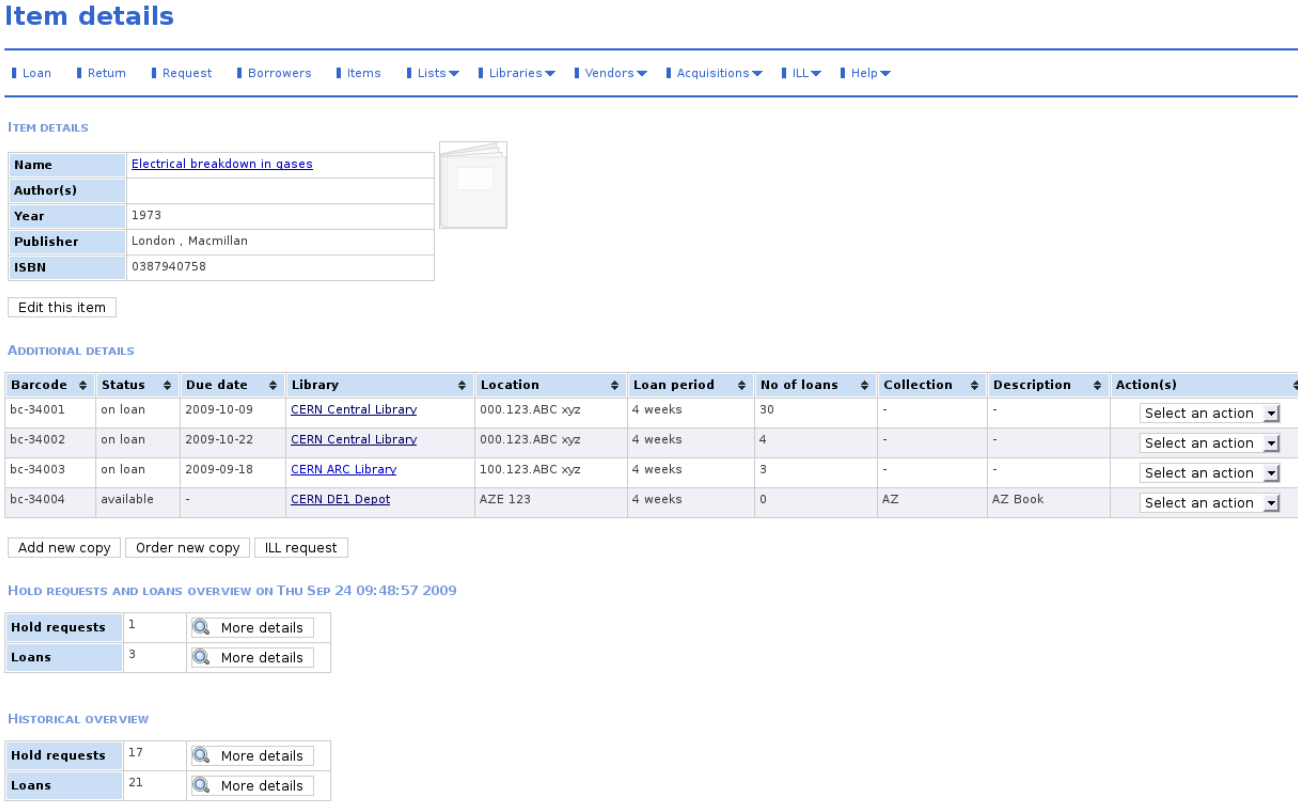

Figure 3.7: Interface containing items informations.

Barcodes Barcodes are a great achievement in terms of ILS. They can provide a quick retrieval of an copy from the database. To retrieve a specific barcode, the library staff can scan the barcode, using a barcode reader, linked to the computer where the system is running. In libraries, the barcodes use the same black-and-white-striped format used in shops or supermarkets.

Figure 3.8: Example of barcode present in bools.

A standard barcode format is the Codabar[1] or Code 39 design, which has 14 digits. It includes item data, institution or library data and the check digit, which is the final digit calculated from the previous digit in the barcode. An alternative, to barcodes is the Radio Frequency IDentification (RFID). An RFID tag contains a microship and an antenna and is programmed electronically. The RFID tag functions in a similar way as a barcode, but is read by radio frequency technology, not scanned like a barcode. Using RFID, library staff can check in or out a stack of eight to ten copies just in a single movement. Other RFID applications are available for inventory, self-checkout and security. Systems providing RFID are more expensive to implement than systems that support only barcodes, because the purchase of a set of additional hardware such as readers and sensors. However, the cost of this type of technology shall decrease as it becomes more widely used.

Like for borrowers, the registration of books can be considered a basic operations, but it's also an essencial functionality on an ILS. To register books, BibCirculation has two different ways to deal with them.

The first case, it's when a book arrive for the first time in the library. This book will be register using BibRecord. Now, all the metadata of the new book is

[1]See `http://www.mecsw.com/specs/codabar.html`

contained on CDS Invenio (it's now available for search) and it has record_id. With the given record ID, we can use BibCirculation to complete the missing information and create a new copy (in this case, the first one) on the table crcITEM. This operation has the purpose to associate the record ID with relevant information for librarians such as the barcode, the loan period, the location of the new book in the library, etc... For more details about the complementary information associated to the record ID, see the details of the table `crcITEM`, on the chapter DB or in the appendix. This complementary information need to be register on this way, because when the new book is registered with BibRecord, there is no MARC field for information like barcode or loan period. So, to solve this problem, it was decided to create a table, `crcITEM`, to assemble all the information together.

The second case, appends when a book already exist in the library and had been registered, at least, once before. In this situation, the process is simple. Like for the first case, we use BibCirculation to add the missing information and register a new copy. In this case, we don't need to use BibRecord, because the book has already been registered and the relevant MARC information has been collected in the past. The library staff just has to go the item's page and select the options *add new copy*.

Registration and management of (sub)libraries

As I mentioned before, sometimes it is important to associate to the different books additional informations. One example of additional information is the library where the book is stored. In the particular case of BibCirculation, it is stored two type of information: the library or sublibrary, because at CERN, the CERN library is divided in different subjects, so there are many sublibrary, and the location, witch give us, the shelf in the library.

This feature is important for the management of different sublibraries, like we have at CERN, and also to keep the information of other libraries such as external libraries. It is common to have requests of books witch are already on loan or doesn't exist in the library. In these cases, libraries demands for the desired book to an external library. This why it is important to register information about external

libraries. BibCirculation provides also the possibility to update the information related with all libraries and write notes about each one.

Registration and management of ILL requests

This subsection is related with the previous one. I talked about requests to external libraries. This type of request or demand is knew as Interloan Library. The idea of this service is to provide the lending of material among different libraries. With BibCirculation, when a borrower is looking for a book and this one is not available or doesn't exist in the library, BibCirculation will show a form where the borrower will perform an ILL request. This request will be treated by the library and the book will be requested from another library.

This functionality is not common for a circulation module. In ALEPH 500, like in many ILSes, there is a module only for ILL. But since an ILL request can be treated like a normal request, it was decided to have this feature in BibCirculation instead of created a new module just for ILL.

In order to manage ILL requests efficiently, it was created in CDS Invenio two new collections: ILL Books and ILL Articles. When a new ILL request is registered for a book or an article who doesn't exists at CERN, some of the bibliographic information is stored in CDS Invenio. This process is done by using BibUpload, witch creates a new bibioghraphic record. The new record for the requested book or article contains all the relevant information stored using MARC21 format. After this step, our new record is in the correct format to be handled by the search engine of CDS Invenio.

Management and Registration of acquisitions

When we talk about acquisitions, we are talking about a kind of support – for financial activities – for the process of adding new items to a library collections. This type of module usually has a database or a table in a database, to register the information about the vendors that the library use to make new purchases. An acquisition module has also a financial system to register all the purchase and

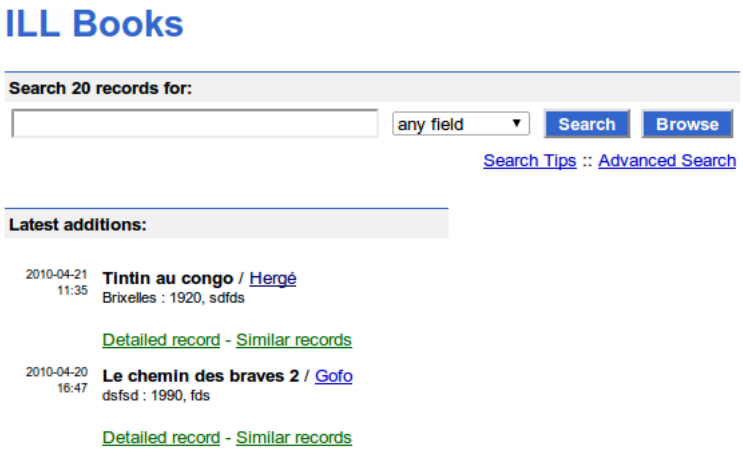

Figure 3.9: Collection containing ILL Books.

```
001149672 001__ 1149672
001149672 020__ $$a
001149672 100__ $$aGofo
001149672 245__ $$aLe chemin des braves 2
001149672 250__ $$a
001149672 260__ $$adsfsd$$b1990$$cfds
001149672 980__ $$aILLBOOK
```

Figure 3.10: MARC record of an ILL book.

allocate the funds according with the available budget. There are specific tasks automated by this type of module such as management of approvals plans based on the policy of the libraries, processement of invoices for new items who have been received and approvement of payments. Some libraries, depending on their size, can implement, in collaboration with their vendors, a transfer of bibliographic data. Amazon provides Amazon WebServices (AWS) where is possible to retrieve this kind of data. This may simplify the management of the information in the library side. It is also common to use, for this data sharing, a protocol called Electronic Data Interchange[2] (EDI).

EDI is the computer-to-computer interchange of strictly formatted messages that represent documents other than monetary instruments. EDI implies a sequence of messages between two parties, either of whom may serve as originator

[2]See also `http://www.unece.org/trade/untdid/directories.htm`

or recipient. The formatted data representing the documents may be transmitted from originator to recipient via telecommunications or physically transported on electronic storage media[19]. The process involved with the acquisition of new material for a library is quite complex in terms of tasks automation. Specially large libraries have an extensive use of acquisitions functionalities and it is common to involve several exchange of data with financial systems. For small libraries, this type of module is not so used like in bigger libraries. Sometimes they manage their purchases using a simple spreadsheet.

Like for the previous functionality, the registration of purchases is not a typical process associate to a circulation module. Usually, there is an independent module to manage this type of operation. But like for ILL requests, it was decided to keep, also, this operation inside BibCirculation, instead of develop a new module just for this single purpose. BibCirculation give us the possibility to manage the purchase of new items. This functionality was not in the list of features to be created, when the development of BibCirculation started. But after some meetings and some tests with the CERN Library Staff, it was decided to implement this additional functionality. I think it is a good improvement, because it provides a feature witch very important for librarians. It is quite common to order new material in a library, so if we have all the management options in the same application it is an advantage.

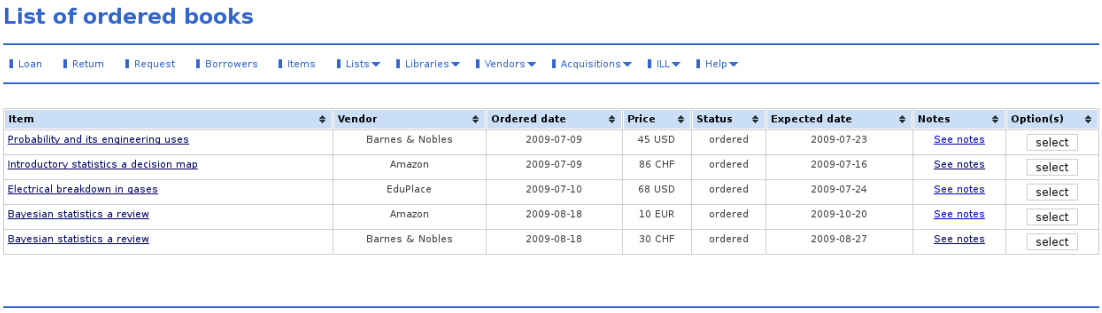

Figure 3.11: List with ordered books.

With this new feature, it is possible to track the material who has been ordered. At any time, it is available a complete overview of all the requests, where we can see the different statuses, the places where the material has been ordered or the expected delivery date. When a new item arrives, it is possible to update the information about the acquisition and finish the procedure. If the acquisition is a copy of an existent book of the library, the new copy will be automatically associate with a record ID.

Registration and management of vendors

Vendors are extremely related with the featured mentioned before. It is, in my opinion, very important to have information about the suppliers of a library. In order, to provide this type of feature, in was created in BibCirculation the possibility to register vendors. It is a place where is possible to keep the relevant information about vendors and also associate notes. If necessary, it is also possible to update the information. For more details about the information collected about vendors, you can see the ER model of BibCirculation or see the SQL script present in the appendix.

Request Workflow

When a request is performed, it can happen in two different ways. To understand how they work, the process has been called request workflow. The idea it's to explain the different steps of a request. As I said, there two different kind of request. A request can be done either online, accessing the web interface of CDS Invenio, or in the library desk.

Request online To perform a request online, a borrower needs to go to CDS Invenio webpage. After this, he can search for several type of informations. CDS Invenio provides a huge set of collection, each one related with a specific category like articles, books, presentations, periodicals, multimedia and much more things. Let's take for example a book request. Our borrower is looking for a book. He can find it by using the search engine of CDS Invenio. He will get a list containing the result of its search query. After to choose the desired item, he will see all the

details about it. In order to request one copy of the book, he needs to go to the tab "holdings" and select one of the available copies. At this moment, BibCirculation will verify if our borrower is logged in or not. If he is not, he will see the login page of CDS Invenio. After the login, he will able to define the period of interest for his request. After this the request process will be done. The borrower will see a new page with a success message and a link for a section called *"Your loans and requests"*. In this section, he will see his loans, requests and also an historical overview. It will be also possible to cancel a request in this section. There are two important details in this step. If it is the first time that our borrower is requesting a book, he will registered as a new borrower, in BibCirculation. The information for the registration process is retrieved from the session and some complementary information is retrieved from CERN LDAP. The second details is related with the information of our borrower. If, by chance, the information about is address or office is not available, the borrower will receive a message informing that is not possible to perform the request because there is no information in the database, about the place where the book should be sent. In the message, the borrower will see, a link (or an email address), where it will possible to update his personal information.

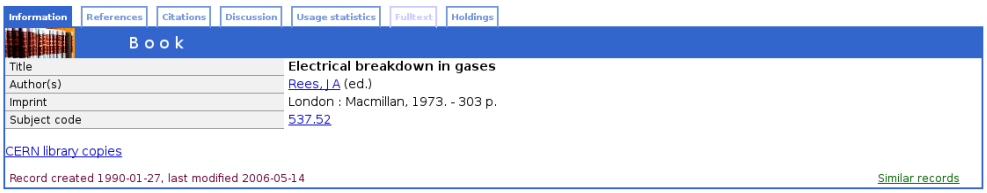

Figure 3.12: Request online: detailed record of a book.

In meanwhile, the new request was registered in BibCirculation DB and it will be treated by the library. In the *admin* interface of BibCirculation, the library staff will be able to verify all the request witch are pending or waiting. If the request status is "pending" that means the book is available in the library. If the request status is "waiting" that means that the requested copy is already on loan and when it will be back, the returned copy will be associate to request.

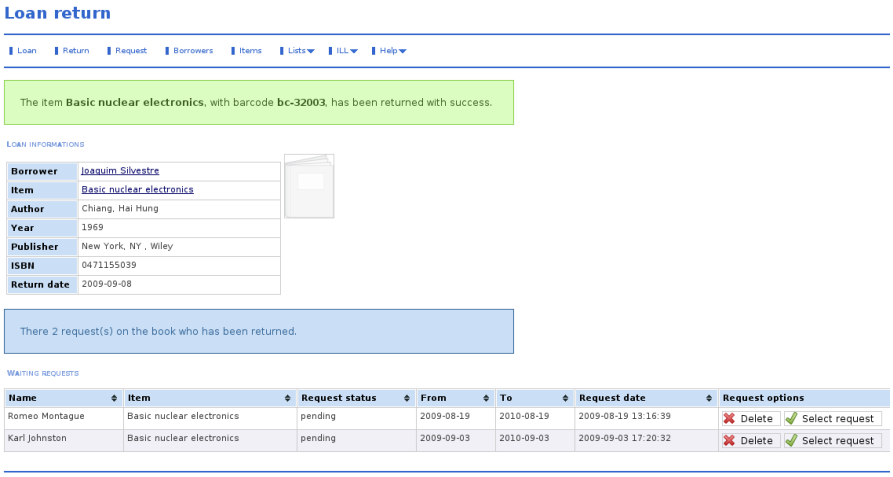

Figure 3.13: Returning a book from loan.

Each time a request is treated there is always an association with a copy of a book, or more specifically with a barcode of a book. For instance, if I am requesting a book, with more than one copy, I am interested in the book, not in the copy. I mean, I just want the book "A" and I don't care if I receive the copy number 1 or the copy number 2. BibCirculation is able to manage this situation. When the request is registered, it is associate to the request, the record ID of the book and the barcode of the selected copy. In the library, the librarian just goes to shelf where the book is and pick one copy. The librarian doesn't care about the barcode. In this case, the barcode, it's just an additional information that can be used for statistics. When the librarian returns to the desk, he will associate the picked copy with the borrower. In the database, the request status will be update do "done" and a new loan will be created in the table `crcLOAN`. Also the `crcITEM` will change. The row corresponding to the barcode will be updated to the status *on loan*. With this last operation, the request workflow, for requests online, it's done.

Request in the library desk The request in the library desk has many operations in common with the request online, mentioned before. This type of request is very traditional, because before the massification of the WWW, it was not possible to request a book online. This is for sure, the most common way to get books

Figure 3.14: Associate a barcode to a borrower.

from libraries. People goes to the library to get a book. At CERN the majority of the request are done, also, in this way. When someone arrives to the library desk with a book for lending, BibCirculation will start a different process from that we have with an online request. First of all, it will be introduced in BibCirculation an information to identify the borrower. In this first step, we can use a borrower id, an email address or the name. For example, if there is a borrower who doesn't remember his borrower id and doesn't have an email address, the librarian will be able to search by the borrower's name. BibCirculation will give a list with names that match the name who has been introduced. After this, the librarian just need to select the correct name and go to the second step. In the second

step, the librarian will see the complete information about the borrower (name, address, email, phone, etc). If all the information is correct, he goes to the third step. In the third step, the librarian will associate the barcode (or the barcodes) of the desired book with the borrower. In the next step, the fourth, the librarian will see a complete overview about the requested book(s). BibCirculation will present information about the return date and also a warning message, if one of the requested books, is already under request. At this moment the librarian is free to decide about what to do. He can just cancel the process or ignore the warning message. This will depend on the rules and policy of each library. If there is no warning message the librarian can finish the request process. At this moment it will happen exactly the same, as I mentioned before, for the request online. The different tables of BibCirculation will be updated and request workflow is finished.

Searching...

In the library staff interface of BibCirculation, it is possible to find several options and functionalities. One of them is to search for something. BibCircularion give us the possibility to search for different types of information such as books, borrowers, libraries and vendors. This kind of functionality is quite important because sometimes with need to find a specific information, but we doesn't know where it is. For example, if I want to search for a book, I have a form to do this operation. I just need to write the name of the book and I will get the result. To perform this operation, specially in the case of book, BibCirculation uses directly the search engine of CDS Invenio (from module WebSearch). That means we can search for a book using the same sintax that we use when we search in CDS Invenio. This is very good for librarians because they will be able to write very complex queries, using for example MARC syntax.

In the case of borrowers, libraries and vendors, BibCirculation uses the search functionalities provided by MySQL. It is maybe not very optimized, but it is , for sure, a good and simple way, to implement this type of feature. In all the different cases of search, the result is always a list containing the information. Each element on the list contains a link to page where we can see the desired information. For instance, if I am looking for a borrower, I will be sent, to the

Figure 3.15: Searching book (Admin Gui).

borrower information page. In that page I will see all the relevant information of a borrower, like personal information and also the loans and the request performed by the selected borrower. If when we are searching, and the result is only one, BibCirculation will display directly the page containing the information. There will be no list with the result. It doesn't make sense, to show a list with a single result, when we can see directly desired information.

Management of requests and loans

In BibCirculation, the librarian has in the toolbar, different menus. One of them is called *Lists*. Inside *Lists*, we can find different list showing information about different *subject*. In the case of the CERN Library, it was defined four different types of lists: Current loans, Overdue loans, items of shelf with holds and items on loan with holds. The first list show us, all the books witch are on loan and the respective borrower. It is possible to see more information such as loan date, return date, barcode, number of renewals, overdue letters and loan notes. It is also possible to perform an action called *Claim return*. This action has the purpose of send a message (an email) to a borrower about one of his loans. The message can be written by the librarian or it can be defined in the config file of BibCirculation. The second list, overdue loans, show us all the loans where the return date has expired. In general, this list has exactly the same information that the previous one and also the same action. During the development of BibCirculation, the CERN Library ask to have a tool to manage automatically the overdue loans. It

was created a BibCirculation daemon. The goal of this tool, was to run every day, at a specific time, and search for all the overdue loans. This daemon was also responsible to send overdue letters for all the borrowers where the return date had already expired and update the database with information related with all the performed operations. The third list, Items on shelf with holds, correspond to all the book witch are in the library and are under request. Before, I mentioned this situation calling it *pending* request. The expression *"items on shelf with holds"* was choose by the CERN Library.

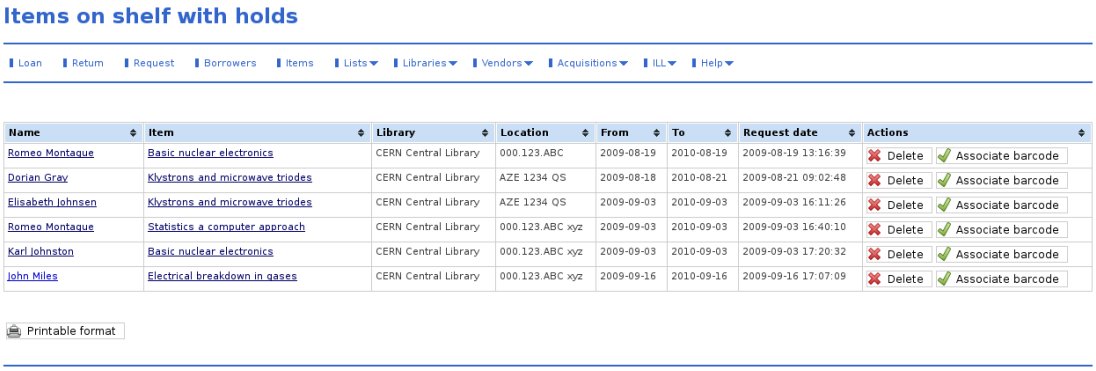

Figure 3.16: List showing *"item on shelf with holds"*.

That means for internal operations, BibCirculation is using the term *pending*, but to keep the same name, of the previous system, in the library staff interface, like it has been, the name of the list is *"item on shelf with holds"*. This list shows to the librarian the borrower name and the requested book. It shows also the location of the book in the library, the period of interest and the request date. It provides also the possibility to perform two action: *Delete* a request and *associate a barcode* to the request. The fourth list, *"items on loan with holds"*, has this name for the same reason has the previous list. Internally, BibCirculation uses the term *waiting* witch corresponds to this kind of situation. In this list, it is possible to see the borrower name, the requested book, the book location (shelf) in the library, the period of interest, the request date and also two options, like in the list mentioned before. *Delete* and *Associate barcode*.

Informations about books, borrowers, libraries and vendors

BibCirculation is based in four different entities, or more precisely, in a set of informations from four different types. We have informations about borrowers, books, libraries and vendors. These are the base of BibCirculation. The relations between these different entities are the origin of a set of secondary entities such as requests, loans, ILL requests and acquisitions. It is very important to manage efficiently all these informations. For librarians it is also fundamental to have the access to all these types of informations in an efficient way. In BibCirculation, we have different pages (for library staff) where it is shown all the information related with the four entities mentioned before. We have a page for borrower informations. In this page, it is possible to see borrower personal information like email, address, phone, etc, and also information about loans, request and ILL request. It is also shown an historial overview and there are four action that can be performed: *New loan*, *New request*, *New ILL request* and *Notify this borrower*.

For books informations, BibCirculation provides also a page, very similar with the page mentioned before. We have a short list containing bibliographic details and there is also a table containing all the copies and several informations about each one. In the last section of the page, there is an historical overview about loans and request, like in the borrower's information page. In the books information page, we have also a button called *Edit this record*, witch is a link to another module of CDS Invenio called BibEdit. This module provides the possibility to edit the different MARC fields contained in a record.

We have more two information pages. One for libraries and another for vendors. They are quite similar and much more simple than the two information pages provided for books and borrowers. For libraries and vendors, both pages show a set of relevant information such as address, phone, email. There is also the possibility to verify notes written previously and also write additional notes. Also in both cases, it is available the possibility to update the information give before. It is just necessary to use the button *Update* and an editable form will appear in order to write the new information about libraries or vendors.

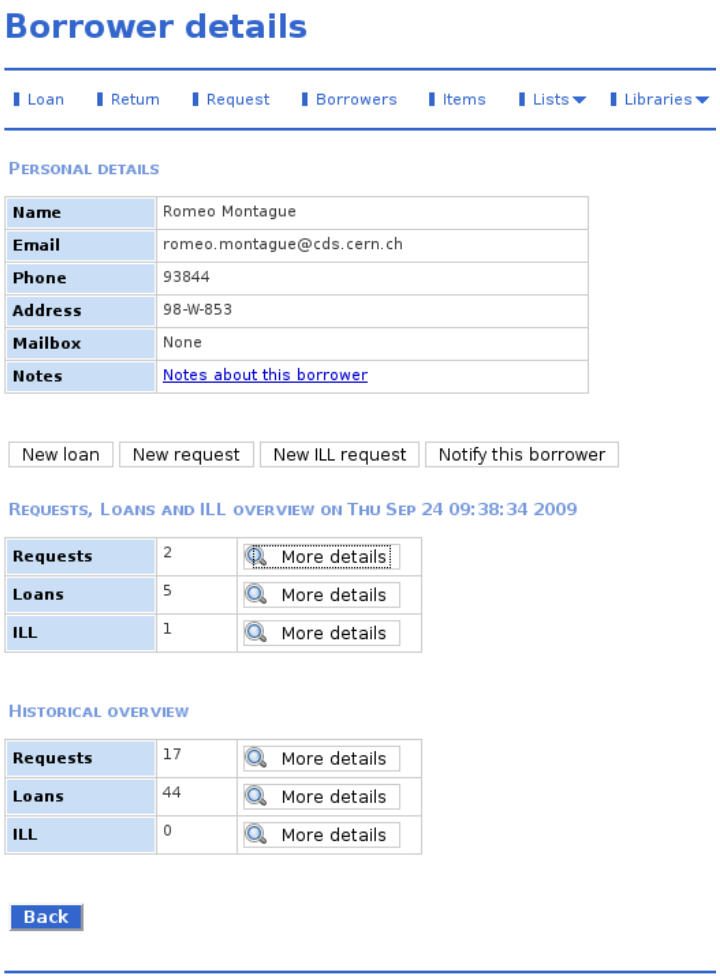

Figure 3.17: Borrower details page.

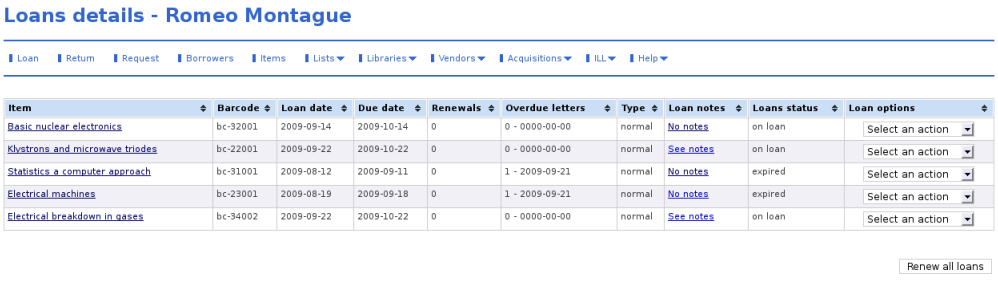

Figure 3.18: Borrower loans details page.

Complementary features

BibCirculation has also some complementary features. These features are not related with the typical functionalities of an ILS. They were implemented in order to help and improve other features. I mentioned before, the possibility to send overdue letters when a loan has expired. This work is done by a daemon. This daemon was created and runs everyday, finding all the expired loans and sending notifications for the different borrowers. Each mail is based on the different templates present in BibCirculation config.

Another functionality related with the previous one, is the notification by email. It was developed a system who allows the library staff do sent emails to the borrowers. Each borrower's page has an option *Notify this borrower*. The librarian will see a form where is possible to write a message or just load a template, like for BibCirculation daemon.

BibCirculation provides another interesting functionality. It is possible to define in the config file, the holidays of the library or other days where the library is not open. This is information is very important and it is used when a new loan is done. BibCirculation will calculate the return date based on this configuration and using the loan period stored for each book in the database. This avoid the possibility to have a return date on a Saturday or during Christmas holidays.

3.3 Implementation

In this section, I will explain what was done in terms of implementation, giving special importance to the technologies and development tools used in this process. In this phase, it will applied all the knowledge collected about the problem in the previous. At this point, we know what are the needs of the library staff and how they will be implemented.

3.3.1 Technology and development tools

I will present all the technologies and tools used on the development of BibCirculation. I will give a brief explanation of each one, with special focus on technical details, trying always to show reason behind the choice.

Python

Python is a general-purpose programming language, created in 1990 by Guido van Rossum[25]. Python is now a mature language highly dynamic, object-oriented, interpreted and interactive. It can be used for many kinds of software development and offers high productivity for all steps of the software life cycle. Python offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality and more maintainable code.

When I started the development of BibCirculation, Python was already being used. In the documentation[26] of CDS Invenio it is possible to understand some of the reasons about the usage of Python:

"Python is highly dynamic language with many redefinition capabilities, very well suited to test-driven development and rapid prototyping. Many people like the "batteries included" aspect of Python. It is very easy to learn. On the other hand, it has got several drawbacks. One of them is the slowness. Another is the lack of a language standard which could become a problem in maintaining programs in 10-15 years span"

MySQL

MySQL is a RDBMS[3], licensed under the GNU General Public License. It is *"the world's most popular open source database because of its consistent fast performance, high reliability and ease of use"*. For these reasons, MySQL is used

[3]Relational Database Management System

by several world's largest companies such as Yahoo!, Alcatel-Lucent, Google and Nokia.

MySQL also become the main choice for the applications developed on the LAMP stack[4]. MySQL can run on more than 20 platforms including Linux, BSDs, Windows, OS/X, Solaris, Symbian, HP-UX, AIX and Netware.

The use of MySQL on the development of BibCirculation is related with the own development of CDS Invenio[26]. It make sense to keep the same database, specially because BibCirculation needs to interact with other tables, who already exist on CDS Invenio.

"Initially, at around 1998, we choose to use MySQL for a simple CDS application, because the inherent simplicity of the problem did not require usage of heavy and complex systems such as Oracle. It would have been an overkill. In the course of years, MySQL has proven very stable, scalable, and capable of dealing with very complex tasks, so its usage at CDS has spread".

Apache HTTP Server/mod_python

The Apache HTTP Server[5] is an open source software, licensed under the Apache License 2.0, and developed in order to create a robust, commercial-grade, featureful and freely-available HTTP Web server. Apache has been, since many years, very important in the expansion of the WWW. In 2009, Apache became the first web server who cross the barrier of 100 million web site. Since is initial release, in 1995, Apache appears as the first reliable alternative to the old Netscape Communications Corporation web server[6].

The Apache HTTP Server project is part of the Apache Software Foundation and it is maintained by a strong and world wide community of developers. Apache is available for a wide number of operating systems, like Unix, FreeBSD, Linux,

[4]LAMP is a term to define how MySQL is used in conjunction with Linux, Apache, and either Python, Perl or PHP

[5]See `http://httpd.apache.org/`

[6]Nowadays know as Sun Java System Web Server

Solaris, NetWare, Mac OS X and Microsoft Windows. Since April 1996, Apache has been the most popular HTTP server on the World Wide Web. In March 2009, Apache served over 46% of all websites and over 66% of the million busiest.

Mod_python is used on CDS Invenio because it allows to the developer to create web-based applications in Python that will run much more faster than a traditional CGI[7] and provide access to Apache's core system. Using mod_python, we have also the advantage of Python Server Pages (PSP), a strategy to embed Python code into HTML pages, like in ASP, PHP and Java Server Pages (JSP).

PyLint

Pylint is a great tool, used to help the development of CDS Invenio since several years. This tool helps developers to improve their code and give suggestions about good practices. Basically, Pylint checks if a module satisfies a coding standard. Pylint is very similar to PyChecker but provides more features such as checking line-code's length, checking if variable names are well-formed according to your coding standard, or checking if declared interfaces are truly implemented. Pylint offers also the possibility to configure and customize different options, adding personal feature.

3.3.2 Interaction with the other modules of CDS Invenio

To achieve one of the goals of this project, BibCirculation should be perfectly integrated with the other modules of CDS Invenio, and use all advantages and functionalities already created. The following figure shows the interaction and the integration of BibCirculation with several modules of CDS Invenio.

WebSearch

This module is used by BibCirculation when a search involving metadata is performed. WebSearch provides to BibCirculation the capability to support the search syntax generally used by librarians (queries with MARC tags). It is one of

[7]Common Gateway Interface

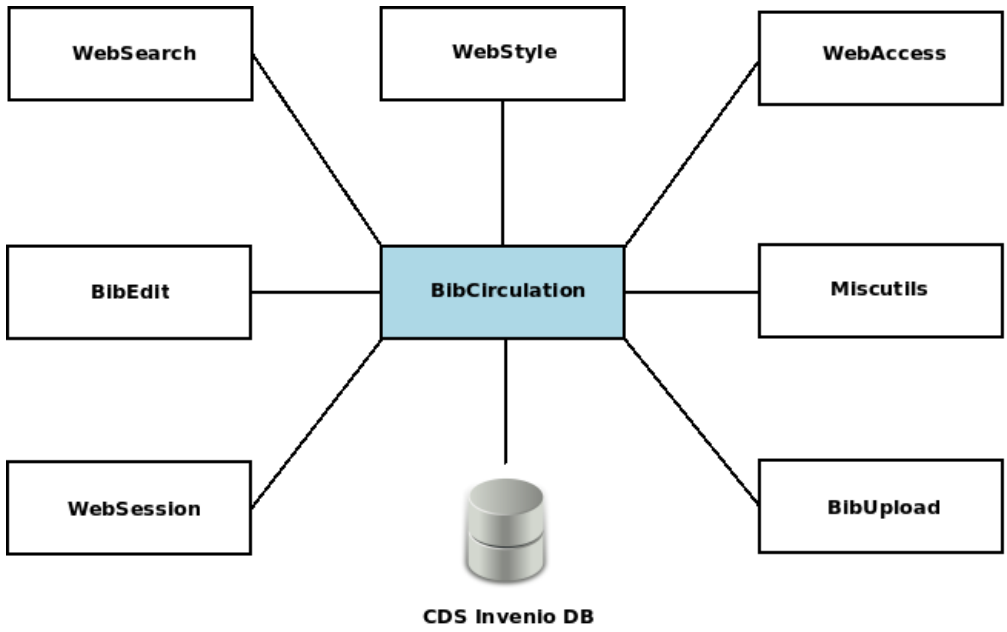

Figure 3.19: Integration and interaction with other modules

the most important module of CDS Invenio. But it provides also some options in terms of interface. For example, when the tab *Holdings* is created in the "Detailled Record" page, this is done by WebSearch.

WebSession

Like for other modules of CDS Invenio such as WebBasket or WebMessage, it is necessary to provide a tool to manage sessions and allow different functionalities depending if a user is logged in or not. WebSession is responsible for this type of verification process. In the case of BibCirculation, if a borrower is not logged in, it will be not possible see the section *Your Loans and requests*. To have this behavior, it is just necessary to configure WebSession with the adequate parameters of each module.

BibUpload

BibCirculation interacts with BibUpload when the treatment of ILL request, about a book or an article, who doesn't exists in CDS Invenio, is being done.

That means, a new MARC record will be created. BibCirculation produces a MARCXML[8] file. This new file will send to BibUpload who will create a new MARC record on CDS Invenio.

```
<record>
    <datafield tag="100" ind1=" " ind2=" ">
        <subfield code="a">Doe, John</subfield>
    </datafield>
    <datafield tag="245" ind1=" " ind2=" ">
        <subfield code="a">On The Foo And Bar</subfield>
    </datafield>
</record>
```

Figure 3.20: Example of a MARCXML record.

Miscutils

Miscutils provides a lot of tools for several types of operations. It is like a swiss-army knife. In the case of BibCirculation, Miscutils is used to provide access to the database.

BibEdit

There is a link between BibCirculation and BibEdit. This is very important, because it allows to the library staff, when they are in the book details page, to go directly to BibEdit and insert or update the MARC fields of a record.

WebStyle

This module provides the *look and feel* for the users of BibCirculation. All the pages or sections provided to borrowers such as *Your loans and requests* are managed by WebStyle.

[8]The core of the MARC XML framework is a simple XML schema which contains MARC data.[**?**]

WebAccess

WebAccess is responsible to provide the access of users to CDS Invenio, managing different levels of roles and different types of authorizations. This module is very important for the development and implementation of BibCirculation. It allows the possibility to distinguish, for example, borrowers from library staff. It permits also create different roles for the library staff. For instance, it is possible to create a role for a person who will responsible to manage ILL requests. With the creation of this new role, only the person (or the persons) associate to this role will be able the perform such kind of action.

3.3.3 Synchronization with ALEPH 500

As mentioned previously, in the Requirements Analysis (section 3.2), BibCirculation should keep all the relevant information contained in ALEPH. This is extremely important, because when the migration will come, we will have books on loan, requests waiting for an answer and a lot of historical information about borrowers and books. In order to migrate all the relevant information of ALEPH in the database of BibCirculation, I have developed a synchronization tool to perform this operation. To understand the functionalities of this synchronization tool, let's see how it works.

To have access to ALEPH data, we need to connect before to an Oracle Database, where the ALEPH information is stored. We can have access to Oracle using the Python class OracleConnector.py. This class was created specially to handle the connections with Oracle Databases. For this procedure, we need also to know that the important data is spread in 4 different databases: CER00, CER01, CER20 and CER50. Each one contains different information. For this synchronization process, it is also important to know, what is the relevant data we will have in our new module BibCirculation. In order to use BibCirculation successfully, without any problems, we need to retrieve information about current loans, requests, borrowers, holdings and historical information.

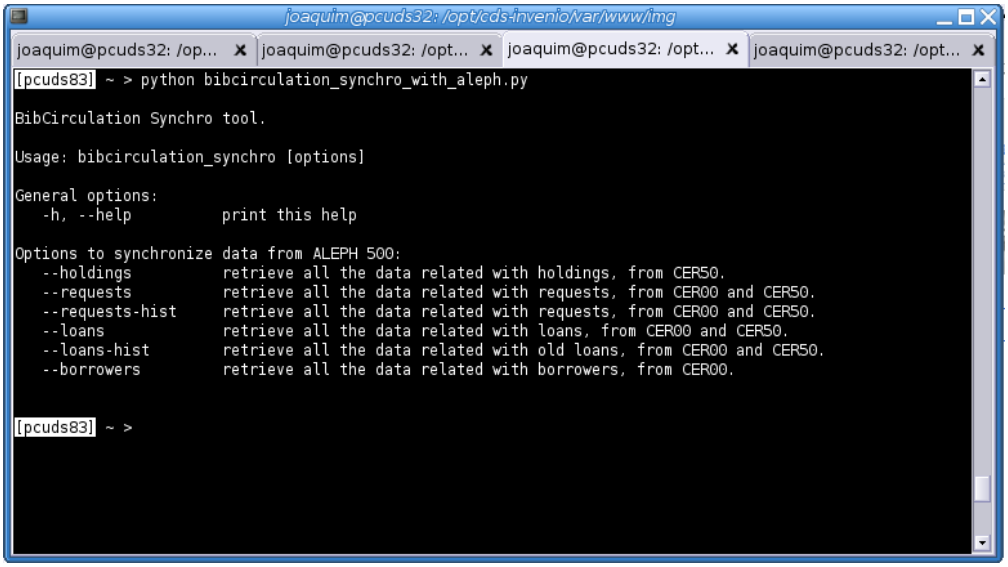

Figure 3.21: Synchronization command line tool.

3.4 External sources of information

To have the necessary information for some tables of BibCirculation such as `crcITEM` and `crcBORROWER`, it was necessary to retrieve information from different external sources of information. In this section, I will explain how was done this work.

3.4.1 CERN LDAP

In order to have additional information about borrowers it was necessary to find a way to get this important information. At CERN there are several services, connected to different databases, where is possible to find information about CERN users such as name, phone, address, office, email, etc... This information was necessary to populate the table `crcBORROWER`.

3.4.2 Amazon Web Services

Amazon Web Services (AWS) are a collection of web services available on the web by Amazon.com, since July 2002. It provides on-line services for other web sites or for other client-side applications. The main purpose of AWS, was to

provide a set of functionalities that developers can use. In June 2007, Amazon claimed that more than 330000 developers had subscribed to use AWS. AWS can be accessed over HTTP using REST and SOAP protocols.

In the development of BibCirculation, the AWS were a great tool. With them, it was possible to retrieve several bibliographic information and specially on the GUI for library staff, it was possible to get all the book's covers. It was very simple to use it, it was just necessary to parse the XML received from the AWS. With Python this task was very easy.

During the development of BibCirculation, it was created an *ah hoc* functionality, who can be used, in the future associated with WebSubmit. This functionality retrieve all the bibliographic information, when a new book is inserted on CDS Invenio. Nowadays, this information has to be filled manually, on the WebSubmit module, but with this new functionality, the submission form will be completed automatically.

Chapter 4

Tests and Comparative Analysis

In the chapter 4, I will describe all the tests done during the development of BibCirculation. This step is very important, because it will verify if our requirements analysis was correct and if our goals have been achieved with success. I did different kind of tests, with different goals. Some of then, for instance, regression tests, are related with the code quality and the improvements (in terms of features) of BibCirculation. But, it was also important to test the usability of BibCirculation with real data and real users. So, it was decided to put BibCirculation running on development/staging machine – CDS DEV – a server who has the data same data as CDS WED, the production server used at CERN.

In this chapter, I am going to provide, also, a comparative overview between our new software, BibCirculation, and other ILS. This comparison is quite important, because it will show if BibCirculation will be a good alternative for other ILS such as ALEPH 500.

4.1 Regression Tests

Regression Tests are a type of software testing that aim to avoid errors after a program has been modified. This kind of software testing is typically used when a developer wants to unsure that its last code changes (like new improvements or a bug fix) have not created bugs in the features previously implemented with success.[**?**]

"Regression testing identifies when code modifications cause previously-working functionality to regress, or fail, ultimately allowing you to catch regression errors as soon as they are introduced. Most organizations verify critical functionality once, and then assume it continues to work unless they intentionally modify it. However, even routine and minor code changes can have unexpected side effects that might break previously-verified functionality." [**?**]

In the case of BibCirculation, these kind of tests were used because they are really relevant for a correct process of software development, but also because

they have been used, since long time ago, in the development of the different CDS Invenio's modules. Since, there was a significant set of functionalities on BibCirculation, it was decided to write the first regression tests. Like I wrote before, the regression tests were very useful because they gave a guaranty about the correct behavior of the features created in first place. For the development of huge and complex system like BibCirculation, this kind of tools are essentials and fundamentals to reach our goals.

4.2 Tests with Selenium - Firefox plugin

BibCirculation is mainly a graphical software application. I mean, we have a graphical user interface to interact with the software. So, it would be great to run tests using the GUI provided by BibCirculation, it would be much more realistic. To achieve this idea, it was selected a very good tool for our purpose – Selenium. Selenium is a plugin for Firefox. It provides an integrated development environment for testing, where it is possible to record, edit and debug tests[**?**].

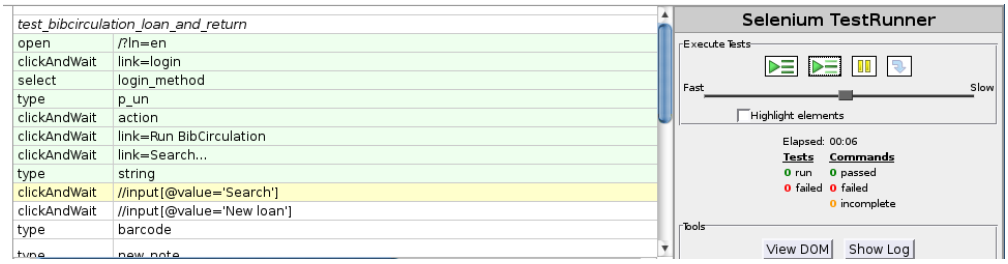

Figure 4.1: Example of Selenium test running.

Regression tests have been very useful, but with Selenium it was possible to test BibCirculation like for real. It was just necessary to record all the desired tests and ran then, each time that it was required. With Selenium, it is possible to pass different kinds of parameters, like usernames, passwords, field values and other kind of information. We can create tests and simulate many types of situations that can append for real. This tool is a great achievement in terms of software testing and improve, for sure, the quality of any software. It was done several

tests and the result was very good. Selenium gave the opportunity to test the integration of different components of BibCirculation and helped also fixing some small bugs.

Inside BibCirculation, it is possible to find the different tests created with Selenium. For all those, want to contribute with new ideas and functionalities for this software, the recorded Selenium tests will be, for sure, a great help.

4.3 Deployment and tests on CDS DEV

Like I said before, in the introduction of Chapter 4, to understand if BibCirculation was created and developed correctly – I mean, being a good alternative for other application with the same goals – it was necessary to test it, with an environment similar to the production environment. To do that, BibCirculation was deployed on CDS DEV. CDS DEV it is a server, witch as the same specifications (operating system, RAM, database information and specially, the same version of CDS Invenio) of CDS WEB – the production server used at CERN.

The tests performed were done by me and more two or three persons of CDS Section and by several members of the CERN library. During this phase several tests were done, trying to verify if all the goals were achieved. The first tests were more *visual*. They were focused on the graphical user interface, checking if it was easier to use and if it was understandable. With these tests and the feedback of the library staff, it was possible to improve the graphical user interface of BibCirculation, creating a very simple, clear and *comfortable* interface. After these tests, our attention was focused on the features. Several cases and situations were simulated and tested. The request cycle, witch includes registration of borrowers, requests and loans (several updates in the database), was tested exhaustively several times, giving the opportunity for some improvements. The final result was quite good. All the features tested were working fine and all the bugs were fixed.

In order to have a good feedback and discuss possible improvements, there was one or two meetings per week. With this strategy it was possible for me to under-

stand how were going the test and what were the questions and the comments of the library staff. After several weeks of tests, the feedback was very good. There was a problem with the synchronization of ILL request stored in ALEPH, but apart from this all the other functionalities were approved by the CERN library.

Two weeks before the end of my contract, I was invented to present my work to all the members of the CERN library. This was very important because I had in this presentation, all the future users of BibCirculation. It was a great opportunity have the feedback of several persons who are dealing since several years with ILS. The result was very positive. They gave me interesting suggestions and comments, in order to keep going with the good job.

4.4 Comparison with other systems

The development and the tests of BibCirculation are done. It is time to compare the final result with other similar systems, trying to understand if the work done in the last months produced good results. The following table give us an overview of several systems and compare their features and specifications with BibCirculation.

Table 4.1: Comparison between different systems

	KOHA	OpenBiblio	Emilda	PMB	EverGreen	ALEPH 500	BibCirculation
Operating Systems	Linux	Linux	Windows and Linux	Windows and Linux	Windows and Linux	Windows	Lunix
Database	MySQL	MySQL	Qual	MySQL	PostgreSQL	Oracle	MySQL
Programming language	Perl	PHP	PHP	PHP	Perl, C and Python	-	Python
Metadata	MARC	UNIMARC	MARC	UNIMARC	MARC	MARC	MARC21
Developed by	Katipo Communications	-	CompanyCube	François Lemarchand	Equinox Software	ExLibris	CERN
Year of creation	1999	-	2000	2002	2005	-	2008
Circulation	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Acquisitions	Yes	No	No	Yes	Yes	Yes	Yes
Serial control	Yes	No	No	Yes	Yes	Yes	Yes
Cataloguing	Yes	Yes	No	Yes	Yes	Yes	Yes[a]
OPAC	Yes	Yes	Yes	Yes	Yes	Yes	Yes[b]
ILL (Z39.50)	Yes	No	Yes	Yes	Yes	Yes	Partially

[a]Provided by other modules of CDS Invenio
[b]Provided by other modules of CDS Invenio

Chapter 5

Conclusions and further work

5.1 Conclusions

The work reported in the present thesis was for me a big and great challenge. From the technically point of view, all the goals defined in the beginning of project were, in the majority, reached with success. CDS Invenio has nowadays available a new integrated module to manage physical items and automate operations from libraries. Considering the goals defined in the section 1.4.2, let's have an overview on the way of each was implemented.

- The new module, BibCirculation, **should be perfectly integrated, like the other modules, and take advantage of all the great tool of CDS Invenio such as the search engine or the treatment of metadata**. This actually happen with BibCirculation. The new module is perfectly integrated with the rest of the system. Its interaction with other modules is quite common. BibCirculation, to perform some operations, needs the support of other modules such as WebSearch, BibEdit, WebSession, BibUpload, WebAccess, Miscutils and WebStyle.

- This second goal, is in my opinion the most important development of this project. It was essential **to provide association between books information and metadata.** This never happened before in CDS Invenio. There was no association between metadata and the items information which is not present in the metadata fields, such as barcodes or statuses. It is possible to store a lot of information with MARC21, several tags are provided to organize the bibliographical data of a book or an article. It is also possible to store the number of copies of an item and identify each copy by a number. But this number doesn't mean anything, it just identify the copy internally. For normal usage, with need barcodes, and this type of information is not possible to store in MARC21 fields. To solve this problem, it was created a new called `crcITEM`. This table provide the association between a record ID and one or more barcodes, depending on the number of copies. With this relation, library staff can search for an item just using the barcode. In the result there will be all the information contained in the MARC21 fields and the complementary information stored in `crcITEM`.

- Another goal of BibCirculation was **to make CDS Invenio more attractive and interesting for new potential users.** With the new module, CDS Invenio has now new features, which are not really usual on a digital library. With the creation of BibCirculation, CDS Invenio can be a 2 in 1 application. It is no more necessary to have a digital library or repository and an ILS. Now it is possible to have everything together. This is a great advantage for new potential users. They will have a new tool where two different systems, which are usually separated, are now integrated in the same application. They can reduce cost in terms of support and maintenance, and they can manage digital and physical items with the same application.

I had also mentioned, in the section 1.4.2, some other goals. Those goals were more specific and related with the implementation of several functionalities. Let's also have an overview about the way of those goals were reached.

- When we are creating a new application one of the first things we have in mind it's the look, the interface of our application. In BibCirculation, **the GUI should be efficient and user-friendly, for library staff and borrowers**. To reach this goal, the design of the GUI was based on the needs of library staff and borrowers. For library staff, we have an interface with a menu containing the main features. Each feature is reachable in one or two clicks. For borrowers, the interface of BibCirculation is integrated with the general interface of CDS Invenio. When a borrower finds a books, he just has to select the tab *holdings* to see all the available copies and perform a request. It is clear and simple.

- In this type of application it is important **to manage all the different entities present in library operations such as items, borrowers, libraries and vendors**. To reach this objective, it was implemented an information page for the different entities mentioned before. For each entity it possible to perform different actions. For libraries and vendors the number of actions is low and simple, but for items and borrowers, the set of available actions is bigger and more complex. These actions provide the appropriate management facilities to improve the work of library staff.

- Like for entities, it is also important **to manage** the relations between them such as **loans and requests**. For this two different type of relations it was created a section, in the borrower's information page and item's information page, to provide information about loans and request. For each section, there is a link to a page containing detailed information of loans and requests, and where is possible to perform different actions. With these actions it possible to manage loans and requests easily.

- It is important **to provide an overview** of what is happening in terms **of loans and requests**. With BibCirculation it is provided a set of lists, where is possible to see the different loans and requests, in different statuses. For each row on the mentioned lists, it is possible to select different actions. These actions are the same that I mentioned in the point before.

- Another goal in term of functionalities was **to provide historical information about loans and requests**. To do this BibCirculation provides a list with this type of information from item's information page and from the borrower's information page. In these list, it is possible to see, different informations such as number of requests or the number of renewals. This information is also important to statistical reports.

- When we are moving from a old system to a new, it is fundamental to keep in the new one the information stored in the old one. One of the goals of BibCirculation was **to migrate all the circulation data contained in ALEPH and integrate it in order to be used**. To achieve this goal is was developed a synchronization tool, which retrieve the information from ALEPH and write it in the tables of BibCirculation. This goal was almost accomplished. There was a problem to find the information about ILL requests in ALEPH. The documentation is not clear about this point and there was no references to the place (database and tables) where the data is stored. Unfortunately, it was not possible to retrieve this information.

- To enrich BibCirculation and take advantage of its architecture, it was decided **to provide support for acquisitions and ILL requests**. As I

mentioned before, it is common to have this functionalities in separated modules. In BibCirculation, the management of acquisitions and ILL requests is quite easy to use. We have pages containing the informations about each acquisition and each ILL request. The information can be updated and it is possible to write notes. BibCirculation provides also list containing an overview of all acquisitions and ILL requests who are being managed.

- One of the most relevant point in the software development is documentation. **Provide documentation for library staff** was one of the goals in the development of BibCirculation. This goals was not completely finished. There is available some documentation but focused in the synchronization process with ALEPH. This should be improved in the future.

5.2 Further work

After several months of development, and considering the goals of the projects, I think there still having some improvements who need to be done, to improve BibCirculation, in order to become an application more complete. One of the biggest problem in the implementation/installation at CERN was the various problems I had to have access to some data. The migration of data about ILL requests was not finished because there was no way to retrieve that information, and the documentation available was not enough. I guess one day this situation will be solved, and than it will possible to finish the synchronization process at CERN.

I think the work done in terms of development was great but in my opinion it was too much focused on the needs of the CERN library. CDS Invenio is an application used in several institutions, each one with different needs. In my opinion, the next improvements or the creation of new features, should take in consideration this fact. CDS Invenio and BibCirculation, should be a generic tool. I guess it will be extremely positive to have also the feedback and the suggestions of other institutions.

Another improvement concern the Z39.50 protocol. It will be good to implement this protocol "correctly". Nowadays it is not being used as it should be. The CERN library has alternative ways to request books and other type of materials, from other libraries, as I mentioned before. I hope in the future it will be possible to use BibCirculation with Z39.50 protocol working perfectly.

It would be nice to provide to BibCirculation the possibility to treat request using RFIDs. I think this will the future of several libraries in the next years. This will increase the level of automation present in libraries and improve the quality of services provided to the borrowers.

In terms of development, I think with would be an advantage to migrate BibCirculation (and all the other modules of CDS Invenio) to a development framework like Django. I know this is difficult to implement. There are thousands of lines of code to migrate, and with all the requests and demands that arrives everyday to the CDS section this is quite complicated. Anyway, this would be a great improvement, all the code would be easier to maintain.

Another important development would be creation of a new feature to generate automatically barcodes. Nowdays, when a new copy arrives, it is possible to associate a barcode who is written manually. It would be good to automate this function.

Bibliography

[1] The European Organization for Nuclear Research. *CERN in a nutshell*, `http://public.web.cern.ch/public/en/About/`. Accessed on 8th April 2009.

[2] The European Organization for Nuclear Research. *CERN Nobel prizes*, `http://public.web.cern.ch/public/en/About/Nobels-en.html`. Accessed on 8th April 2009.

[3] Tim Berners-Lee, CERN, March 1989. *The original proposal of the WWW.* `http://www.w3.org/History/1989/proposal.html`. Accessed on 20th March 2009.

[4] A. Pepe, T. Baron, M. Gracco, J.-Y. Le Meur, N. Robinson, T. Simko, M. Vesely. *CERN Document Server Software: the integrated digital library.* Presented at the ELPUB 2005 conference, Heverlee (Belgium), 8-10 June 2005.

[5] Indira Koneru. *Integrated Library System: Selection and Design.* DESIDOC Bulletin of Information Technology, Vol. 25, No. 5 6, September & November 2005, pp. 3-9

[6] M.G. Sreekumar. *Open Source Web Content Management Technologies for Libraries.* Center for Development of Digital Libraries (CDDL), Indian Institute of Management Kozhikode.

[7] Marcia Deddens. *Overview of Integrated Library Systems.* University of Cincinnati, 20th August 2002.

[8] Lucy A. Tedd. *Library Management Systems in the UK: 1960's - 1980's.* Aberystwyh University, 2007.

[9] Lucy A. Tedd. *Program 1966-2006: celebrating 40 years of ICT in libraries, museums and Archives*. Emerald Group Publishing Limited, 2006.

[10] Laura Maquignaz, Jane Miller, *The Centrality of the Integrated Library Management System: a Strategic View of Information Management in an E-Service Environment*. Victoria University, 2003.

[11] Gwen Bell. *Stories Behind the Devices*. `http://ed-thelen.org/comp-hist/TheCompMusRep/TCMR-V19.html` Accessed on 21th April 2010.

[12] IBM. *IBM Archives: 1928*. `http://www-03.ibm.com/ibm/history/history/year_1928.html`

[13] Mark Russo. *Herman Hollerith: The World's First Statistical Engineer*. `http://inventors.about.com/gi/dynamic/offsite.htm?site=http://www.history.rochester.edu/steam/hollerith/index.htm` Accessed 24th February 2010.

[14] Mark Russo. *The Influence of Dr. Billings*. `http://www.history.rochester.edu/steam/hollerith/billings.htm` Accessed 24th February 2010.

[15] Hugh Devonald, Derek Eldridge. *How the ICT 1900 series Evolved*. `http://www.cs.man.ac.uk/CCS/res/res16.htm#e` Accessed 24th February 2010.

[16] Jack Beidler. *Waterfall-to-Boehm*. `http://blog.hydro4ge.com/2007/12/16/waterfall-to-boehm/` Accessed 16th February 2010.

[17] Tech Republic. *Understanding the pros and cons of the Waterfall Model of software development*. `http://articles.techrepublic.com.com/5100-10878_11-6118423.html?part=rss&tag=feed&subj=tr` Accessed 14th September 2009.

[18] Rollo G. Woods. *Library automation in British universities*. MCB UP Ltd, 1986.

[19] Katie Wilson. *Conputers in Libraries – An introduction for library technicians*. CRC Press, 2006.

[20] Graeme Muirhead. *Planning and implementing successful system migrations.* Library Association Pub, 1997.

[21] John Akeroyd, Andrew Cox. *Integrated Library Management Systems: overview.* VINE, 1993.

[22] Janet Stafford. *Self issue — the management implications. The introduction of self-service at the University of Sunderland.* Emerald Group Publishing Limited, 1993.

[23] Lorcan Dempsey, Rosemary Russell, John Kirriemuir. *Towards distributed library systems: Z39.50 in a European context.* Emerald Group Publishing Limited, 1993.

[24] Marchall Breeding. *Open Source Integrated Library Systems.* Library Technology Reports, 2008.

[25] Alex Martelli *Python in a nutshell.* O'Reilly, 2006.

[26] Tibor Simko *Tibor Simko's Vademecum and Essays.* `http://simko.home.cern.ch/simko/vademecum/` Accessed 22th September 2009.

Appendix A

Database SQL script

```
-- BibCirculation tables:

CREATE TABLE IF NOT EXISTS bibrec (
  id mediumint(8) unsigned NOT NULL auto_increment,
  creation_date datetime NOT NULL default '0000-00-00',
  modification_date datetime NOT NULL default '0000-00-00',
  PRIMARY KEY  (id),
  KEY creation_date (creation_date),
  KEY modification_date (modification_date)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcBORROWER (
  id int(15) unsigned NOT NULL auto_increment,
  name varchar(255) NOT NULL default '',
  email varchar(255) NOT NULL default '',
  phone varchar(60) default NULL,
  address varchar(60) default NULL,
  mailbox varchar(30) default NULL,
  borrower_since datetime NOT NULL default '0000-00-00 00:00:00',
  borrower_until datetime NOT NULL default '0000-00-00 00:00:00',
```

```
  notes text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcILLREQUEST (
  id int(15) unsigned NOT NULL auto_increment,
  id_crcBORROWER int(15) unsigned NOT NULL default '0',
  barcode varchar(30) NOT NULL default '',
  period_of_interest_from datetime NOT NULL default '0000-00-00 00:00:00',
  period_of_interest_to datetime NOT NULL default '0000-00-00 00:00:00',
  id_crcLIBRARY int(15) unsigned NOT NULL default '0',
  request_date datetime NOT NULL default '0000-00-00 00:00:00',
  expected_date datetime NOT NULL default '0000-00-00 00:00:00',
  arrival_date datetime NOT NULL default '0000-00-00 00:00:00',
  due_date datetime NOT NULL default '0000-00-00 00:00:00',
  return_date datetime NOT NULL default '0000-00-00 00:00:00',
  status varchar(20) NOT NULL default '',
  cost varchar(30) NOT NULL default '',
  book_info text,
  borrower_comments text,
  only_this_edition varchar(10) NOT NULL default '',
  library_notes text,
  PRIMARY KEY  (id),
  KEY id_crcborrower (id_crcBORROWER),
  KEY id_crclibrary (id_crcLIBRARY)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcITEM (
  barcode varchar(30) NOT NULL default '',
  id_bibrec int(15) unsigned NOT NULL default '0',
  id_crcLIBRARY int(15) unsigned NOT NULL default '0',
  collection varchar(60) default NULL,
  location varchar(60) default NULL,
```

```
  description varchar(60) default NULL,
  loan_period varchar(30) NOT NULL default '',
  status varchar(20) NOT NULL default '',
  creation_date datetime NOT NULL default '0000-00-00 00:00:00',
  modification_date datetime NOT NULL default '0000-00-00 00:00:00',
  number_of_requests int(3) unsigned NOT NULL default '0',
  PRIMARY KEY  (barcode),
  KEY id_bibrec (id_bibrec),
  KEY id_crclibrary (id_crcLIBRARY)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcLIBRARY (
  id int(15) unsigned NOT NULL auto_increment,
  name varchar(80) NOT NULL default '',
  address varchar(255) NOT NULL default '',
  email varchar(255) NOT NULL default '',
  phone varchar(30) NOT NULL default '',
  type varchar(30) default NULL,
  notes text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcLOAN (
  id int(15) unsigned NOT NULL auto_increment,
  id_crcBORROWER int(15) unsigned NOT NULL default '0',
  id_bibrec int(15) unsigned NOT NULL default '0',
  barcode varchar(30) NOT NULL default '',
  loaned_on datetime NOT NULL default '0000-00-00 00:00:00',
  returned_on date NOT NULL default '0000-00-00',
  due_date datetime NOT NULL default '0000-00-00 00:00:00',
  number_of_renewals int(3) unsigned NOT NULL default '0',
  overdue_letter_number int(3) unsigned NOT NULL default '0',
  overdue_letter_date datetime NOT NULL default '0000-00-00 00:00:00',
```

```
  status varchar(20) NOT NULL default '',
  type varchar(20) NOT NULL default '',
  notes text,
  PRIMARY KEY  (id),
  KEY id_crcborrower (id_crcBORROWER),
  KEY id_bibrec (id_bibrec),
  KEY barcode (barcode)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcLOANREQUEST (
  id int(15) unsigned NOT NULL auto_increment,
  id_crcBORROWER int(15) unsigned NOT NULL default '0',
  id_bibrec int(15) unsigned NOT NULL default '0',
  barcode varchar(30) NOT NULL default '',
  period_of_interest_from datetime NOT NULL default '0000-00-00 00:00:00',
  period_of_interest_to datetime NOT NULL default '0000-00-00 00:00:00',
  status varchar(20) NOT NULL default '',
  notes text,
  request_date datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  (id),
  KEY id_crcborrower (id_crcBORROWER),
  KEY id_bibrec (id_bibrec),
  KEY barcode (barcode)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcPURCHASE (
  id int(15) unsigned NOT NULL auto_increment,
  id_bibrec int(15) unsigned NOT NULL default '0',
  id_crcVENDOR int(15) unsigned NOT NULL default '0',
  ordered_date datetime NOT NULL default '0000-00-00 00:00:00',
  expected_date datetime NOT NULL default '0000-00-00 00:00:00',
  price varchar(20) NOT NULL default '0',
  status varchar(20) NOT NULL default '',
```

```
  notes text,
  PRIMARY KEY  (id),
  KEY id_bibrec (id_bibrec),
  KEY id_crcVENDOR (id_crcVENDOR)
) TYPE=MyISAM;

CREATE TABLE IF NOT EXISTS crcVENDOR (
  id int(15) unsigned NOT NULL auto_increment,
  name varchar(80) NOT NULL default '',
  address varchar(255) NOT NULL default '',
  email varchar(255) NOT NULL default '',
  phone varchar(30) NOT NULL default '',
  notes text,
  PRIMARY KEY  (id)
) TYPE=MyISAM;
```

Appendix B

Request workflow: online request

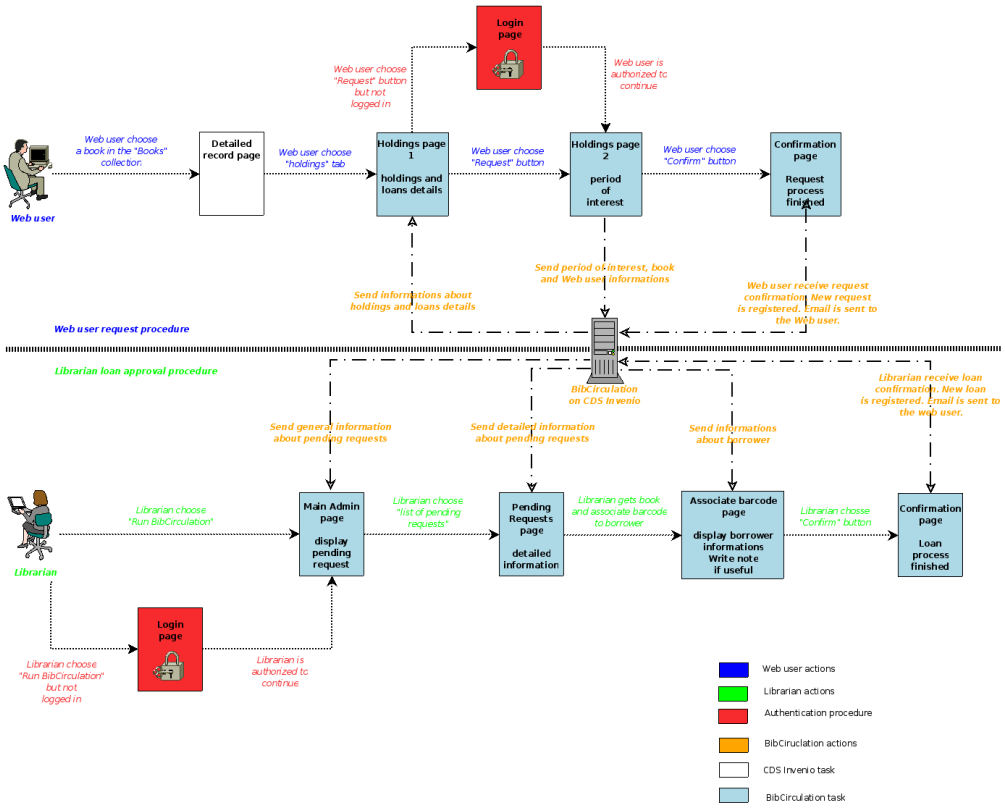

Figure B.1: Request workflow: online request.

Appendix C

Retrieving data from ALEPH

C.1 How to retrieve holdings information

All the information concerning holdings is stored in CER50 on the table z30. To get the correct information, for the synchronization process with BibCirculation, we need to retrieve the following fields:

- sysno (it is the system number used by ALEPH);
- barcode;
- sublibrary;
- collection;
- location;
- description;
- item_status (it corresponds to the loan period);
- process_status;
- nb_loans (number of loans);

C.2 How to retrieve loans information

Like for holdings, all the information about loans is stored on CER50 on the table z36. In order, to get the correct information, for the synchronyzation process with BibCirculation, we need to retrieve the following fields:

- sysno;
- barcode;
- user_id;
- loan_date;
- due_date;
- nb_renewall;
- letter_number;
- letter_date;

To retrieve all the fields listed above, we will need to get also some relevant data from table z30.

C.3 How to retrieve requests information

Like for holdings and loans, all the data concerning requests is stored on CER50 on the table z37. In order, to get the correct information, for the synchronization process with BibCirculation, we need to retrieve the following fields:

- sysno;
- user_id;
- req_status;
- req_date;
- from_date;
- to_date;

C.4 How to retrieve borrowers information

All the information related with borrowers is stored on CER00 on the table z303. From the table z303 with get the rec_key (Z303_REC_KEY). This rec_key represents the ID of a borrower in ALEPH. We can get the rec_key using the following code:

```
list_ids = db_cer00.run_sql("select Z303_REC_KEY from z303")
```

Each rec_key has a relation with the CERN ID. This relation is made on the table z308.

This information is important, because with the CERN ID, we can get all the other informations related with borrowers such as the email, the phone number, the address and the mailbox.

In order to be sure about the quality of the borrower's data, we should retrieve information from CERN LDAP. To do this, we can use the following method:

```
def get_user_info_from_ldap(nickname="", email="", ccid=""):
    """Query the CERN LDAP server for information about a user.
    Return a dictionary of information"""
    try:
        connection = _ldap_connection_pool[get_ident()]
    except KeyError:
        connection = _ldap_connection_pool[get_ident()] = _cern_ldap_login()
    if nickname:
        query = '(displayName=%s)' % nickname
    elif email:
        query = '(mail=%s)' % email
    elif ccid:
        query = '(employeeID=%s)' % ccid
    else:
        return {}
    try:
        result = connection.search_st(CFG_CERN_LDAP_BASE, ldap.SCOPE_SUBTREE, query,
```

```
        if result and nickname:
            return result
        else:
            try:
                return result[0][1]
            except IndexError:
                return {}
    except ldap.TIMEOUT:
        pass
    return {}
```

Examples:

```
get_user_info_from_ldap(email="johndoe@cds.cern.ch")
get_user_info_from_ldap(nickname="John Doe")
get_user_info_from_ldap(ccid="12345678")
```

C.5 Populating BibCirculation database

During the Synchronization process, it is necessary to populate the database of BibCirculation. This operation is done by the same script that is responsible for retrieving the relevant information from ALEPH 500. After retrieved all the information the script will insert in BibCirculation database.

The following table shows, in terms of quantity, the information retrieved from ALEPH 500:

Type of information	**Amount**	**Retrieved in**
loans	2674	00:01:32
loans (historical)	46223	00:20:06
requests	264	00:00:08
requests (historical)	8974	00:03.43
holdings	279579	01:41:34
borrowers	16152	00:05:27