Special features of the advanced loans module of the ABCD integrated library system.

Egbert de Smet

Author: Egbert de Smet is Professor of Information and Library Science, Antwerp University, Belgium. E-mail: egbert.desmet@ua.ac.be

Abstract :

Purpose : The ‘advanced loans’ module of the relatively new library software, ABCD, is presented by explaining why it is there in the first place (on top of the normal loans module) and the special extra functionalities it offers as a ‘generic transaction decision-making engine’. The module requires extra installation effort and parameterisation, so this article tries to explain to the many potentially interested libraries, especially from the target-group of ABCD users, why (or why not) and how to use the module in comparison with the basic loans module.

Approach : After some brief history the technical concepts of the software and the required technical skills to use them are described.

Findings : The article concludes that libraries with specific conditions (external user registration, high transactional volumes, multi-policy) the extra burden of installing the software might be worth the effort.

Practical implications : The installation and configuration of the software are briefly described and the possible gains offered by the software are highlighted.

Originality/value : Potential users of the ABCD software will get useful information for helping to decide on whether or not to use the additional Advanced Loans module.

Keywords : Library automation; CDS/ISIS software; Circulation systems; Free and Open Source Software (FOSS); Integrated library systems; Library management systems; Third World libraries

Category : General review

Word length: 3,398

1. About the ABCD software

At the end of 2009 the ABCD software was officially presented in its first release (1.0) in Sao Paulo, Brazil after an intensive two-year development phase led by BIREME (the health information branch of the World Health Organization in Latin America) and the DOCBIBLAS project at the Flemish Interuniversity Council in Belgium.

ABCD stands for 'Automatisación de Bibliotécas y Centros de Documentación' or equivalents in other European languages, e.g. in English, Automation of liBraries and Centres for Documentation, and in Spanish, Automatización de Bibliotecas y Centros de Documentación. It is actually a suite of relatively independent, but co-operating, modules dealing with database administration (catalogues, users, loan objects and so on), acquisition, cataloguing (in any bibliographic format but MARC and CEPAL are pre-defined), loans, serials management, OPAC and a web portal. For the housekeeping of the data the UNESCO text-retrieval software CDS/ISIS is used. Some modules, e.g. the OPAC , Portal and Serials Management, have been developed and used for some time by BIREME in its large bibliographic applications like the Scientific Electronic Library Online - SciELO (http://www.scielo.br), LILDB, LILACS and most of the Virtual Health Library(Biblioteca Virtual em Saúde – BVS). So these are solid proven technologies. To make it a complete library management package the ABCD 'Central' suite member was produced, allowing all CDS/ISIS for Windows functionalities to create, edit and maintain ISIS databases in a web environment and adding specific library management functions like acquisitions, loans and statistics within the same integrated environment.

ABCD is now being implemented in small as well as large university libraries, mostly in Latin America but also East Africa. For instance, in Cuba both the ministries of Higher Education and Health have selected ABCD to run in their institutions. ABCD is described more detail by de Smet (2009).

2. EmpWeb : the advanced loans module of ABCD

2.1 Some history

During the development phase of ABCD at the end of 2008 the development leaders discussed some technical limitations of the CDS/ISIS database technology in view of the transactional characteristics of the loans function. Whereas CDS/ISIS is very powerful for text retrieval and semi-structured data manipulation, especially through its Formatting Language, which allows librarians to take full control of most functions of their data without having to program, the software lacks, by definition, some virtues of transactional (relational) databases. Advantages and disadvantages of both relational and non-relational databases have been discussed more in-depth since the so-called 'No-SQL' movement gained grounds in computer science (see amongst many others : http://**nosql**.mypopescu.com*)*.

The technical concepts of CDS/ISIS , although being old as they stem from the 1970s (which can be thought of as pre-history in Computer Science terms!), are still very healthy and advanced, but have drawbacks when it comes to relatively simple but high volume and dynamic transactional data. So we came to the conclusion that for larger organisations, with higher and more critical-mission circulation needs, additional technology would be preferable for the loans module of ABCD.

ABCD also faces another major challenge : how to combine the needs of small libraries, with very limited financial resources as well as limited manpower and technical skills resources, with the aim of ABCD being advanced and also powerful enough to serve big institutional libraries like university libraries (the target group of the Flemish Interuniversity Council) ? De Smet (2010) explores this theme further. The 'normal' built-in loans system of ABCD would try to make it as easy as possible for small libraries and documentation centres to automate their limited circulation administration, but would be considered insufficient for the larger more complex organisations. For example, the basic loans module allows, next to the use of a dedicated loan-objects database, the copy-information to be included in the catalogue records (as repeated fields) as is still the practice in many small libraries, but not advisable in more developed circulation environments.

Based on this reasoning it was decided to add an 'advanced' loans module to the basic one in order to cope with the sometimes conflicting technical and user characteristics.

A small company (Kalio from Uruguay) with 'inherited' interest in CDS/ISIS technology meanwhile had already produced a solution for the Catholic University of Chile combining ISIS with other technologies. The original software, known as EMP, was produced in a DOS environment, and the web version was known as EmpWeb. After some negotiations, based on goodwill, it was decided to free the software as open source and to incorporate EmpWeb into ABCD as a suite-member and hire some expertise to re-vamp the software to the ABCD-style and formats (for example using the same user-database and catalogues).

2.2 Technical background

From a technical point of view EmpWeb is rather complicated as it involves the following technologies :

- ISIS-databases for the catalogues and loan-objects
- ISIS-databases for the patrons (loan system users)
- SQL-databases for the patrons running in parallel with the ISIS users-database
- SQL-databases for the transactions
- JAVA as the programming language, involving JDK, a java-servlet server Jetty and a scripting language GROOVY for definition of the circulation regulations, rules and policies
- Web-services technology to link up the above mentioned parts
- The MODS standard to create and exchange bibliographic data in between the different parts.

A basic concept of EmpWeb is the 'pipeline' : all transactions are the result of a series of conditions (rules, parameters defined by the loans policies) which are queued for checking and, if accepted, for passing on to the next stage in the pipeline, until either the pipeline is prematurely exited due to a condition not being met, or a transaction finally is granted at the end of the pipeline.

This concept allows EmpWeb to present itself as an 'agnostic' general-purpose transactional engine: feed the engine with whatever type of parameters and actions, it will process all parameters and decide on granting the transaction (storing its data in the transactions database) or not.

In the case of ABCD the parameters are characteristics of users, loan-objects, rules (how many books can be taken for how long by a user of type X ?) and policies (which can be different for each substituent part of the circulation system, for the season and so on) and of course the transactions are the typical library loan-transactions : issuing, returning, reservations, renewals.

2.3 Differences with the ABCD central loans module

In addition to the obvious and typical loan-system functionalities like issuing, extending and returning objects, management of objects (taken from different catalogues) and users, definition of the circulation parameters and reports, all of these being present in the basic ABCD Central loans module, EmpWeb offers the following extra features :

- Reservation of objects (with the possibility to define pre-emptive reservations) from the web-based OPAC
- The 'MyLibrary' function : users can check their user status online
- User data can be obtained from external SQL-based databases in addition to the built-in ISIS-users database
- Distributed servers can be combined into the system
- Policies can be distributed as well, i.e. be different for different members of the circulation system (libraries or branches)
- Rules can be added and positioned anywhere in the pipelines applied, to give full control of the logical checks leading to a transaction.

As an example for this last feature we can point to the possibility of adding a third dimension on top of the classical two-dimensional space where loan transactions are defined by object-categories (e.g. books can be loaned for a longer period than, say, videos) and user-categories (e.g. professors can take more books than undergraduate students). Such a third dimension could also be the 'season', i.e. during examination period the parameters can be different from summer-time vacation conditions and so on.

2.4 Installation and configuration issues

From the previous description it might have become clear that all this doesn't come for free, but as should be the case in the Free and Open Source Software (FOSS) philosophy to which ABCD fully adheres, the costs are related to skills and time for implementation, not finances.

While the installation of ABCD, which comes as either an installer or a package to be simply unzipped with its own PHP and Apache folders and configurations, involves the installation of ISIS (currently the wxis and CISIS utilities of BIREME), PHP (version 5) and Apache (version 2), the EmpWeb module adds, as explained above, some extra software like Java and MySQL (or other SQL environments supported by Java JDBC like PosGreSQL and Oracle).

In reality the Java Development Kit (JDK) has to be installed on the computer system on which ABCD is running. . Since Java servlets are compiled at their first run, existing installations cannot be mixed with different JDK versions, and the correct version has to be referenced to in the launching script (empweb.bat in Windows and empweb.sh in Linux) in the variable 'JAVA_HOME'. Also the Jetty-servlet server comes with the EmpWeb package and for database management, MySQL server (or another similar SQL-server like PosGreSQL) has to be installed, preferably along with a graphical MySQL-interface like phpMyAdmin.

Installation of the Java components, although being large files (and therefore not that obvious for many typical CDS/ISIS users in remote areas with limited Internet bandwidth), is mostly straightforward and doesn't require any special skills.

Installation of the MySQL server can be cumbersome at times despite the availability of 'meant-to-be-easy' installers, especially when something goes wrong (such as in the definition of the paths, or the password) and all kinds of traces hidden deep into the Operating System need to be removed for re-installation. Some courage and skill will be necessary here. But MySQL is of course used on many computers, often also in pre-combined packages with Apache and PHP like EasyPHP or WAMP (for Windows) or XAMP for Linux, so in the end it should work. ABCD can be installed into such combined packages because all paths and parameters are definable in text-files, but good knowledge of Apache configuration with virtual servers and so on will be needed.

After installation of the MySQL server, a demonstration transactions and users database structure is to be dropped into MySQL using an SQL-file which defines all structures (tables, fields) and some demonstration data. The command 'source' can be used with the command-line MySQL client or using phpMyAdmin also allows running such a SQL-script for creation and filling of tables.

Another complication of EmpWeb, as far as installation is concerned, is related to the use of several ports for the different web services in addition to the application port itself for Jetty (which by default would be port 8080). The default port for ABCD is 9090, so we use tricks like 'ProxyPass' parameters in Apache to call EmpWeb from port 9090. But other ports are also used and all of these of course should be available for use by EmpWeb. All of these are defined in XML-structured files of the EmpWeb system and some deep-digging into the EmpWeb folders and files will be needed. At a later stage simplification at the installation stage is envisaged.

Language versions for EmpWeb can be added/removed/re-sequenced in the file 'gui.properties' in the EmpWeb folder \ABCD\empweb\gui\WEB-INF\conf\ewi18n\core\.

The ports used by EmpWeb mostly are defined in the three Jetty-related XML-files in the folder \ABCD\empweb\common\etc\.

A successful installation of EmpWeb, whether in Windows or Linux, can be tested in the following steps :

- Testing the EmpWeb URL : http://127.0.0.1:8080/empweb/ : this should display the welcome page in the typical ABCD-style
- Testing the ProxyPass with port 9090 : http://127.0.0.1:9090/empweb/ : if working the same page will be displayed as in the first step, if not this option can be dropped as it is only for convenience.
- Initialising the transaction database : in the 'Administration' part of EmpWeb there is an option 'databases' which allows initialising the transa-database (re-setting all data to 0, so deleting all existing transactions); if successful , the link to MySQL from EmpWeb works well – if not : all settings in the XML-files referring to the MySQL environment (e.g.

user/password, name and path of the transa-tables, ports used…) need to be checked and corrected.

- Doing a query on users : the EmpWeb installation comes with four demonstration users in the MySQL table whereas the ABCD-installation comes with three ISIS users predefined for demonstration purposes. If successful, seven users, combining the results of both different user databases (ISIS and MySQL) will be listed. If the ISIS users are missing the references to the ISIS database has to be checked both in the EmpWeb installation folders (e.g. isis_ABCD_usersconfig.xml in /ABCD/empweb/dbws/WEBINF/conf) and if the MySQL-users are missing other references need to be checked, e.g. also in the ABCD Central subfolder 'bridge' (config.inc.php) where a bridge is laid in between the ABCD-PHP environment and MySQL.
- Doing a query on objects : the EmpWeb system uses exactly the same ISIS database as the Central Loans module, i.e. the database 'loanobjects' of ABCD, where for each bibliographic entity in any of the defined catalogues the unique identifier (referring to the catalogue as 'control number') is given and for each copy of the document in a repeatable subfielded field : the identifier (barcode), location (library and shelf) and possibly other identifier (volume, copy). So when running a 'query' in EmpWeb actually an ISIS search is performed since EmpWeb uses an ISIS-DLL (or SO for Linux) derived solution which is compatible with ISIS. Therefore the results should be the same as for running the same search in ISIS (or ABCD Central). If not again the parameters in the EmpWeb XML-configuration files (e.g. isis_ABCDmarc_objectsconfig.xml) should be corrected.

Once EmpWeb is running well, the proper localisation of EmpWeb (in the 'administration' part of the software) involves the following steps :

- Definition of the libraries in the system with their opening hours (and making sure the libraries are defined with the same identifiers as used in the loan-objects database)
- Definition of operator-profiles with connection parameters (IP-addresses limited or not, time-table limited or not, library limited or not etc.) and functional authorisations to define which operations a member of this profile is allowed to do or not
- Definition of the profiles and policies : a profile is a combination of an object - with a user-category (e.g. books for undergraduate students) for which each time lots of parameters can be defined : e.g. 'Fine amount for an expired reservation when the object has no reservations'. The set of profiles is a 'policy' and different policies can be defined, e.g. for different libraries or periods. This involves a lot of work, both in designing good policies and profiles, and in implementing them into the system.
- Calendars should be defined along with opening hours for each library.
- Pipelines : the rules and their sequence (which can be re-ordered) available in the pool can be created/edited. EmpWeb comes with a large set of pre-defined rules, e.g. 'is the user active or not at this moment ?' or : 'are all conditions met for renewal of a loan?'.

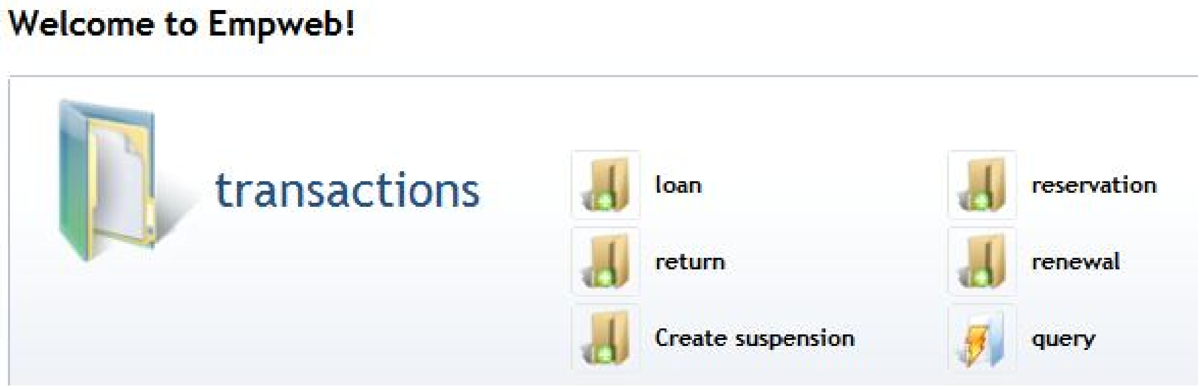
As can be seen from the above, installing EmpWeb is not obvious and since every transaction involves ALL conditions to be met (the URL to be correct, Jetty working at the defined port, the web-services ports to be available, the data being available in both ISIS and MySQL, and then all 'pipe-lined' conditions to be fulfilled), many hurdles need to be overcome. But once a transaction is granted, one can be sure of fast and secure handling with many possibilities and built-in flexibility.

3. Use of EmpWeb

The actual use of EmpWeb, as should be the case with any high-frequency daily-use software, is a ‘piece of cake’. Figure 1 provides a screenshot of the transactions menu page.

Take in Figure 1

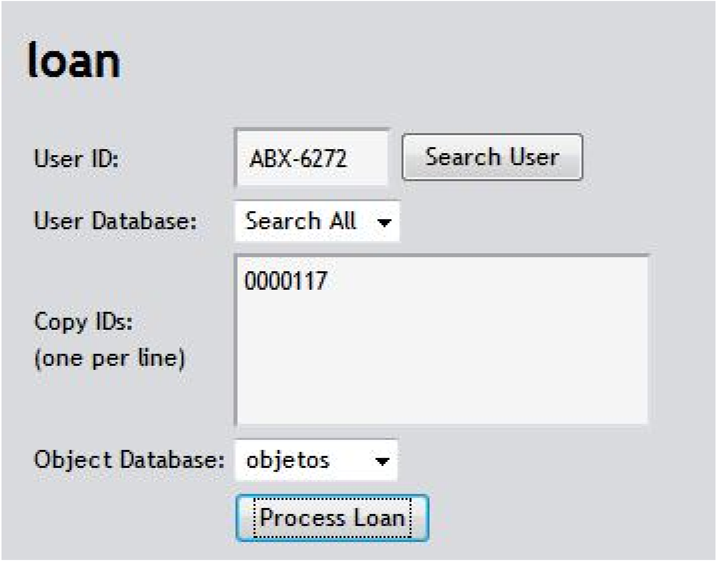
Figure 1 Transactions menu page



Creating a loan involves scanning the identifiers (barcodes) of a user and one or more objects as shown in Figure 2.

Take in Figure 2

Figure 2 Screen for creating a loan



loan

User ID: ABX-6272

User Database: Search All

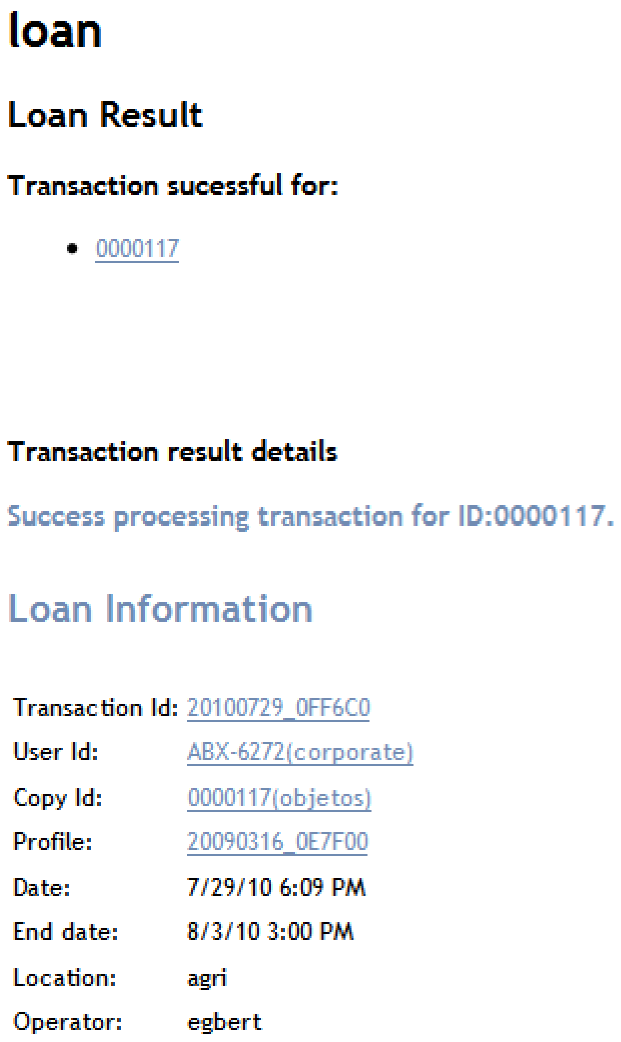
Copy IDs:
(one per line)
0000117

Object Database: objetos

The resulting screen is shown in Figure 3 ,with all details.

Take in Figure 3

Figure 3 Loan details



loan

Loan Result

Transaction sucessful for:

- 0000117

Transaction result details

Success processing transaction for ID:0000117.

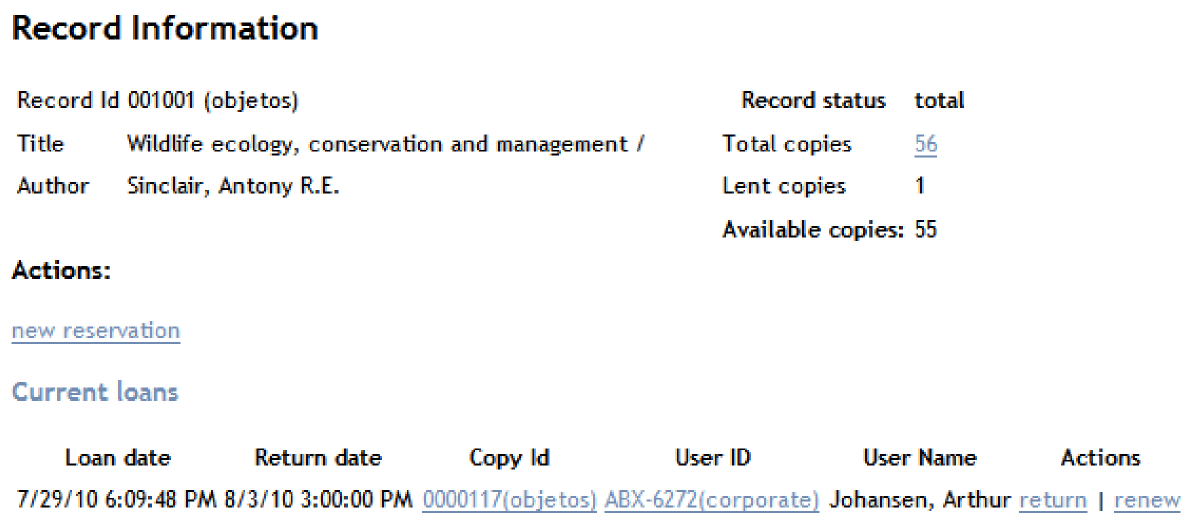
Loan Information

Transaction Id:	20100729_0FF6C0
User Id:	ABX-6272(corporate)
Copy Id:	0000117(objetos)
Profile:	20090316_0E7F00
Date:	7/29/10 6:09 PM
End date:	8/3/10 3:00 PM
Location:	agri
Operator:	egbert

When checking the book in the circulation system, the data are displayed as shown in Figure 4.

Take in Figure 4

Figure 4. Details from circulation system



Record Information

Record Id 001001 (objetos)		Record status	total
Title	Wildlife ecology, conservation and management /	Total copies	56
Author	Sinclair, Antony R.E.	Lent copies	1
		Available copies:	55

Actions:

new reservation

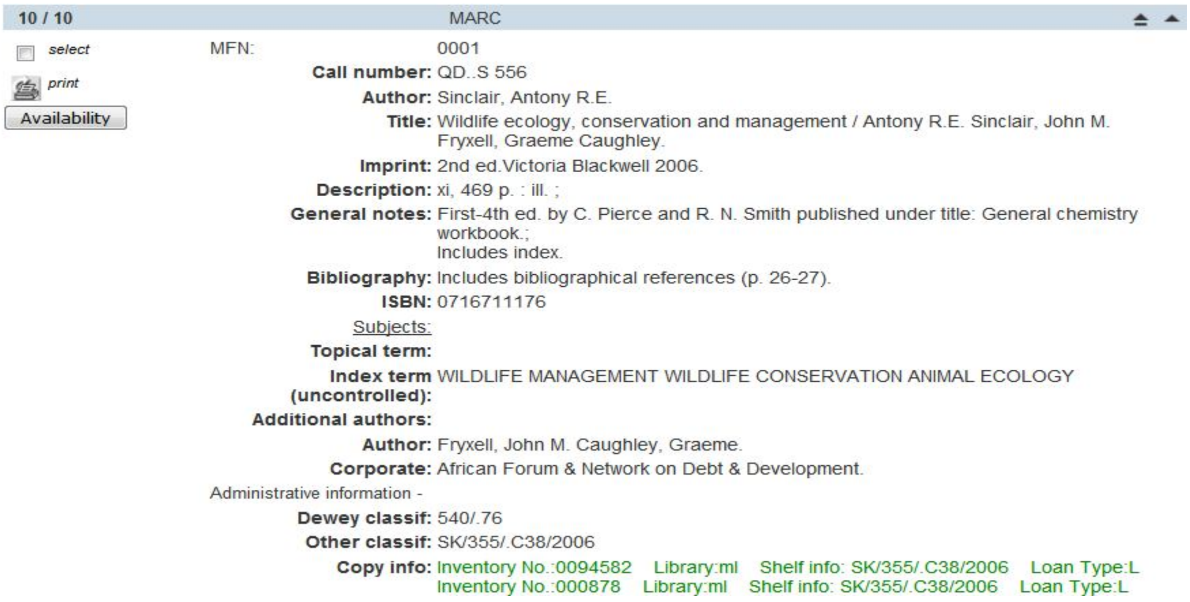
Current loans

Loan date	Return date	Copy Id	User ID	User Name	Actions
7/29/10 6:09:48 PM	8/3/10 3:00:00 PM	0000117(objetos)	ABX-6272(corporate)	Johansen, Arthur	return | renew

An interesting and less-obvious feature is the end-user facility to reserve books from the OPAC or check the user-status by clicking on the ‘availability’ button in an ABCD OPAC search result as shown in Figure 5.

Take in Figure 5

Figure 5 OPAC search result

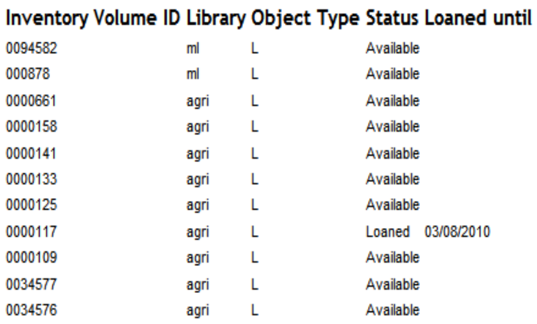


MFN: 0001
Call number: QD..S 556
Author: Sinclair, Antony R.E.
Title: Wildlife ecology, conservation and management / Antony R.E. Sinclair, John M. Fryxell, Graeme Caughley.
Imprint: 2nd ed.Victoria Blackwell 2006.
Description: xi, 469 p. : ill. ;
General notes: First-4th ed. by C. Pierce and R. N. Smith published under title: General chemistry workbook.;
Includes index.
Bibliography: Includes bibliographical references (p. 26-27).
ISBN: 0716711176
Subjects:
Topical term:
Index term (uncontrolled): WILDLIFE MANAGEMENT WILDLIFE CONSERVATION ANIMAL ECOLOGY
Additional authors:
Author: Fryxell, John M. Caughley, Graeme.
Corporate: African Forum & Network on Debt & Development.
Administrative information -
Dewey classif: 540/.76
Other classif: SK/355/.C38/2006
Copy info: Inventory No.:0094582 Library:ml Shelf info: SK/355/.C38/2006 Loan Type:L
Inventory No.:000878 Library:ml Shelf info: SK/355/.C38/2006 Loan Type:L

This would give in a separate window the copy information as shown in Figure 6.

Take in Figure 6

Figure 6 Copy information for a given title

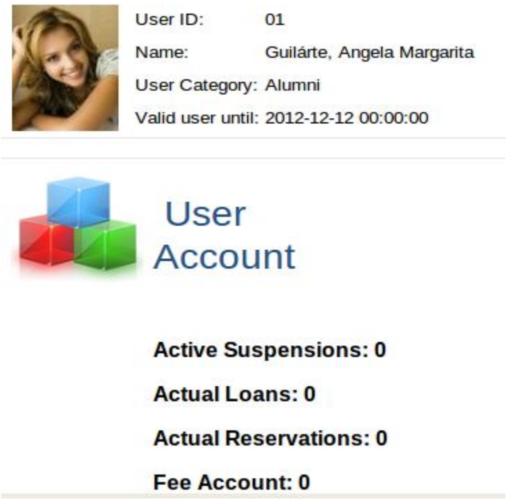


Inventory	Volume ID	Library	Object Type	Status	Loaned until
0094582		ml	L	Available	
000878		ml	L	Available	
0000661		agri	L	Available	
0000158		agri	L	Available	
0000141		agri	L	Available	
0000133		agri	L	Available	
0000125		agri	L	Available	
0000117		agri	L	Loaned	03/08/2010
0000109		agri	L	Available	
0034577		agri	L	Available	
0034576		agri	L	Available	

As can be seen one copy is on loan. When clicking on 'reservation' (at the end of the list, not shown here) the user can login into his/her own page and reserve the book (if no copies are available) or check her/his status as shown in Figure 7.

Take in Figure 7

Figure 7. User account



User ID: 01
Name: Guilárte, Angela Margarita
User Category: Alumni
Valid user until: 2012-12-12 00:00:00

Active Suspensions: 0

Actual Loans: 0

Actual Reservations: 0

Fee Account: 0

4. Conclusions

The EmpWeb Advanced Loans module of ABCD offers interesting added value to the basic loans module, but at a price. The price is to be paid in installation of additional but free software (Java and an SQL-database) and acquiring quite some additional skills, and a lot of effort in implementing the many customisation features of EmpWeb.

In return one gets a very powerful circulation module which can connect to databases of the more traditional type (SQL) most probably used already by other administrations within the institute or organisation like Registrar's Office for students' registration. Transactions volume and speed, even if not really a problem when using the basic loans module in smaller organisations, will be on par with any current standards of performance (e.g. MySQL has many flaws as a 'relational' system but is very fast and efficient for simple setups like in a loans system).

EmpWeb therefore is advised for use for ABCD-users where one or more of the following conditions is met :

- Users-data exist in external SQL-tables
- High transaction volumes are expected
- Distributed hardware setup with several servers
- Good skills are available to write (Groovy-)scripts for defining additional transaction rules and logics.

Unfortunately the ABCD system manager has to make a choice between the Central (basic) loans module and EmpWeb : even if both use the same ISIS-users and loanobjects databases, the transactions are stored in a completely different way and this, unless sophisticated synchronisation would be applied e.g. overnight (which in itself is not impossible), makes them mutually exclusive. Therefore the manager has to take a well-founded decision at an early stage and this article hopefully has brought some useful and necessary elements for such strategic decision.

References

de Smet, Egbert (2009), "ABCD : a new FOSS library automation solution based on ISIS",

Information Development, Vol. 25 No.1, pp 61-67.

de Smet, Egbert (2010), "Some ISIS- software history and technical background on the new FOSS integrated library system ABCD", *Liber quarterly: the journal of European Research Libraries / European Research Libraries* , Vol. 19 Nos 3/4,pp. 324-335