The ISIS-software family : from ‘Free and Open’ to ‘Free and Open Source Software’.

Egbert de Smet
Univ. of Antwerp, Belgium

1. Some data and history of (CDS/)ISIS software
2. The ISIS-software as an ‘open’ software tool
3. The ISIS-software as open source
4. Consequences for training
5. The future of ISIS-software as FOSS

Abstract
In this article the (CDS/)ISIS software will be discussed as a ‘predecessor’ to the ‘Free and Open Source’ (FOSS) software development movement which is currently gaining importance, also in the library and documentation field. Even though the full adherence of ISIS to this movement is of recent date, we will illustrate how from its beginning – which is very long ago in computer science traditions – the software had concepts of being ‘free’ and even ‘open’ by referring to several technical elements and aspects in this sense. Therefore it is claimed that the software has always been ‘free and open’ but only recently fully moved to the ‘free and open-source’ type [The proper concepts of ‘freeware’ vs. ‘open software’ and the many differences in between Open Source licenses and definitions are not however discussed here.]
Some experiences in teaching students and training system managers on the software will be discussed in the light of this specific context of being ‘open software’.
Finally the current ‘FOSS’ ISIS-projects are briefly described as they will define the software’s future, with a call to the wider community to contribute, in order to make ISIS a real FOSS project.

1. Some data and history of (CDS/)ISIS software

CDS/ISIS is the name of the ‘text-retrieval database’ software which was originally developed at ILO (Geneva) in the seventies and taken over for wider support and free distribution by UNESCO since 1985. UNESCO not only provided the very successful ‘Micro CDS/ISIS’ version for DOS/PC but also the ‘WinISIS’ version for Windows (from 1997) with tens of thousands of (institutional and individual) users all over the world, mainly in developing countries. New versions and applications based on the ISIS-standard(s) have been developed in the environment of other UN-organisations such as FAO (e.g. with the WEBAGRIS and WEBLIS software) and WHO/Bireme (Brazil) with amongst other their very powerful CISIS-tools, a graphical development library ISIS.DLL and webserver softwares (WWWISIS and WXIS).
The ISIS-standards define a storage mechanism based on a format registered as ISO2709 (from which MARC was derived) allowing variable length records with variable field structures (as each record carries its own ‘header’ to define its identity), a B-Tree based indexing mechanism (including full-text word indexing), a Query Language based on Boolean logics and the Formatting Language to define all sorts of processing of data taken from the databases before sending them to a display or printer, to a sorting mechanism, to a validation mechanism during data entry or exporting them with a conversion.
The Formatting Language provided, from the early days of the WWW, the possibility to directly produce HTML-output without a need for an additional scripting language.

As several institutions and individuals have programmed new variants of ISIS based on these same standards (with only minor deviations), we prefer to refer to these as a 'family' of softwares : the ISIS-software family.
Many applications therefore have seen the light, from 'old-fashioned' but very robust DOS- and UNIX-applications to rich graphical (Windows-based) end-user applications and powerful web-based applications (e.g. the earlier mentioned WEBLIS system as an integrated web-based library management software). These applications, along with the basic software, are widely used, especially in Latin-America, but also in non-profit environments in Europe and the South (Asia, Africa) in general.
More details on all this can be found elsewhere in the literature of this particular software, e.g. see the URL http://library.wur.nl/isis/biblio.html .

2. ISIS as an 'open' software tool

Whereas ISIS, from the DOS-version produced and distributed by UNESCO as from 1985, always has been 'free' – i.e. without cost but with a restriction to the not-for-profit sectors only – the software was not 'open' in the strict meaning of the concept as nowadays known as 'Open Source Software' with its different definitions (see http://www.opensource.org/docs/osd) and licenses (e.g. (L)GPL, BSD, Creative Commons..). But in 3 meanings there were, already from this beginning - and therefore long before the FOSS movement began to become really visible –, elements of being 'open' in addition to being free(ware) :

1. *the standards were open and published*. In the 'CDS/ISIS Reference Manual', written by its founding father Gianpaolo Del Bigio (working for ILO then UNESCO), the technical details were published in the annexes, allowing others to program their own versions of ISIS using the same compatible standards. E.g. in Slovakia Marek Smihla had programmed executables (e.g. ADEM for data-entry) which ran independently from the ISIS-executables from UNESCO and could write and read ISIS-records. Bireme in Sao Paulo, Brazil, did something similar : they programmed writing, reading and indexing tools with lots of advanced features (e.g. joining databases, linking them as relations etc…) in the C-language (therefore CISIS) which are still the basis for their other ISIS-related software : the DLL and the webservers (WWWISIS, WXIS) and which now have expanded capacity, e.g. 4 Gb max. database size, 1 Mb record size, 60 character-index keys. Co-operation was then set up with UNESCO, e.g. allowing the 'CDS/ISIS for Windows' to become a mix of UNESCO-programmed and Bireme-programmed modules.
2. *an open, adjustable interface* : the software itself was presented as a very flexible environment, with three main features which were used heavily all over the world not only to change its 'interface' but also the functions and features.
    a. An open menu-structure : Micro-CDS/ISIS was fully based on menus which could be produced and changed by using the software itself, including the definition of 'actions' to be invoked by each menu option and allowing hierarchical sub-menus as well as dropping/adding options.
    b. An open message system : all messages were/are based on small ISIS-databases which can be edited (each language having its own message-database) and expanded. This not only allowed (often together with the previous feature of open menu's) creation of rather different conformations of the software – taking into account also colors and screen-features which could be changed – but also expansion and

introduction of parameters (which could then be 'read' as messages) for additional software running inside ISIS (see further : ISIS/Pascal add-ons), as amply used e.g. by the cataloguing interface 'ODIN' and OPAC 'IRIS' (by the author of this article).

c. A programming tool 'ISIS/Pascal' which acted as an 'API' (with published calls for functions and their parameters) inside CDS/ISIS. ISIS/Pascal programmes, varying from a few lines to thousands of lines for sophisticated applications, could be included into the program either as 'format exits' (to expand the functions of the already very rich Formatting Language) or as 'menu exits' to expand the functions of the menus, allowing almost independent interfaces to 'take over' the CDS/ISIS environment in the creation and manipulation of its databases. One feature illustrating the 'openness' was the possibility of adding a parameter in the 'Syspar.par' initialization file to automatically invoke a menu and its option, therefore allowing the menu-interface to be skipped and immediately presenting the new ISIS/Pascal interface. In this way full OPAC (e.g. IRIS using a welcome-screen which could be invoked by a time-out mechanism after a previous session was left) and CD-ROM search modules (HEURISKO is an example) were written, loan-systems for libraries and thesaurus-management tools were produced.
d. *Last but not least : the 'open character' of the Formatting Language.* The Formatting Language is a grammar used to define in a detailed way how elements of the database-data, taken from repeatable fields and subfields, also from other records in the same or other databases (therefore resembling relational approaches) and with navigation links, will be 'processed' in some output (for display, sorting, printing, exporting). It was largely expanded with graphical features in the Windows version (RichText but also images and extra text- and image-boxes). Together these strong 'data-processing' and 'presentation' features of the Formatting Language have allowed the production of rather new 'identities' of the software, e.g. as a Library Management software with OPAC and Loans System (e.g. PURNA from India). In current applications, based on web-technology, the Formatting Language is still gracefully used to produce HTML-elements (e.g. links but also tables), even if more dedicated tools for that, e.g. PHP, are now added to the power of the own ISIS Formatting Language.

3. ISIS as full open source software

Already in 2001 UNESCO decided to embark on this relatively new approach of not only providing the software for free but also making the source codes in principle 'open', i.e. publicly available (see : http://portal.unesco.org/ci/en/ev.php-URL_ID=13803&URL_DO=DO_TOPIC&URL_SECTION=201.html). This has finally lead to a framework of its wider 'Free and Open Source Portal' approach promoting the idea and adding other softwares, e.g. Greenstone, into their 'basket' of supported and promoted softwares for better professional development also in the Southern and transitional countries. UNESCO's FOSS Portal can be found at : http://www.unesco.org/cgi-bin/webworld/portal_freesoftware/cgi/page.cgi?d=1, with interesting links to discussions of the FOSS history, licenses and case studies.
In reality however the source codes for existing ISIS-software are to be requested from UNESCO, but the new softwares will be fully available on public websites.

At Bireme/OPS/WHO a similar decision was taken in 2006/7. No longer would the institute charge a small fee for their software (as was the case before, e.g. 150 USD for official registration as a user with support rights) and therefore make it 'free', but also the sources have been and are still being prepared for publication of all their software, including the basic CISIS-modules. Their new ISIS-generation software, called 'ISIS-NBP' (Network Based Platform) will follow FOSS-methods (including a 'community' with possibilities to contribute, discuss and download sources at the URL http://reddes.bireme.br) to show their firm commitment to FOSS.
Also their newest full-fledged application, an integrated library management system called 'ABCD' (Automatisación para Bibliotécas y Centros de Documentación or : Automation for Libraries and Documentation Centres), will be fully published as open source, even if the original development is still centrally managed by Bireme and its own programmers, as the project is now also supported by the Flemish Interuniversity Council (VLIR) with specific requirements to present it as a full competitor to other library systems (including the FOSS –brothers KOHA and NewGenLib) and to this end needs some more central control for specific purposes. The advantage of becoming fully open source – for all software - lie in the fact that users, certainly (programming) skilled ones, can fully check on the internal mechanisms and propose/make changes if so desired. One example: WinISIS has a slightly different way of sorting values taken by the 'VAL'-function (i.e. removing padding 0's first) which is not a bug as such and therefore doesn't 'need' to be corrected by the software provider; with access to the source codes one could change this however. As is always the case with open source software, it would be best not to make such changes without consulting/informing the 'developers' community'.

4. Consequences for training

The very 'open' nature of the main ISIS-representations, i.e. Micro-ISIS and WinISIS, had consequences for training and educational use. We think these were both positive and negative.

- As positive consequences we would like to mention the fact that the software can be presented, due to its openness, as a 'tool-box' for all text-related database applications and problems. The formatting Language, for example, has allowed the author of this article (and many others) to produce advanced conversions without the need of any real programming tools or expertise. Students could easily be impressed by explaining such conversion techniques, for which many libraries using commercial software had – or would have – to pay huge amounts of money, conversion mostly being 'tailor-made' work and requiring higher technical skills. I used ISIS to demonstrate how strongly textual data could be manipulated and processed, once the data were converted into ISIS (for which many tools are available, see e.g. Piet De Keyser's website on conversion techniques for ISIS [http://library.wur.nl/isis/keyser] and more recent tools such as IsisAscii and ImpExp, available from UNESCO's webpage on ISIS).
  The toolbox idea became even stronger – allowing courses taught to be elevated from mere 'software training' (i.e. 'how to use a software', which buttons to press to get which pre-defined result) to capacity building – by the possibility to show how, if one ISIS-interface (e.g. Micro-ISIS) could not do the trick, another could be used, e.g. from the large set of tools or utilities in CISIS.
- An adverse consequence we need to admit, however is that students were/are relatively easily confused by the (too) many conformations and features of the

software. From a mere didactic point of view it is easier – a lot easier in fact – to just explain how a certain closed software works : specific actions have known results, the software always has the same “looks ‘n feeling”, but not so with ISIS. Students expect a database software to be ‘user-friendly’, which mostly means ‘easy to use, requiring a minimum of effort’, whereas the alternative definition of user-friendliness, as ‘being functionally rich and offering many solutions to the users, takes some getting-used-to.
Many students have had basic training on e.g. ‘Microsoft Word’ (which is mostly presented as ‘Word Processing’, like Microsoft Excel is taken for ‘Spreadsheets’, which in my view is wrong) and therefore are not used to considering software as an open toolbox.
Some incompatibilities within the family, e.g. in the Formatting Language - the syntax for linking databases in WinISIS e.g. is ‘REF->otherDB(MFN,PFT)’ whereas in CISIS it is REF[‘otherDB’](MFN,PFT) -, are also confusing of course (and should have been avoided indeed) but when different teams are working on the same software in different tracks, some differences are to be taken into the balance.

Mentioning the ‘balance’, tone can rightfully state that in the end the balance is quite positive : the training courses using (more than ‘about’) ISIS invariably have resulted in lots of enthusiastic new users, mostly from African universities and institutes, for all kinds of purposes, i.e. from scientific reference systems with advanced contents disclosure, to community information systems in public libraries (see: LIBRI 45: 36-44 , Mar. 1995, de Smet, Egbert ‘Evaluation of a computerised community information system through transaction analysis and user survey using GIDS-system in a public library in Belgium’).

5. The future of ISIS-software as FOSS

At this moment (early 2008) the ISIS software is being developed along different tracks, but all of them now are fully ‘open source’, albeit that with strong interests existing (e.g. Bireme runs all their databases, including very big ones like the 13.000.000+ records Medline provides online, using ISIS) most of the (initial) development is done in-house by their own programmers. This, by the way, is not uncommon as it is also the case with most other major FOSS-projects : MySQL is actually a company (now taken over by Sun) and so of course are Sun and IBM, two major defenders of Open Source.
We see 3 major developments now :

- UNESCO is developing a general-purpose ISIS-DBMS running on Java (J-ISIS) which will not only allow ISIS to run in a graphical stand-alone environment also on Linux/UNIX, but also it will be full UNICODE (which is quite important in such an internationally aiming project) and it will have dropped the currently existing limits re record and database-size. Indexing will be based on Lucene, meaning larger keys can be defined and techniques such as ranking are possible. As an important innovation we have to mention the use of a separate ‘back-end’ database, in this case Berkeley DB for the storage management of the fields in the records. The idea is to maintain compatibility with other new ISIS-applications using the same (BIREME-derived) API to link the application with the storage layer. FOSS development will be facilitated after the initial stage by the use of NetBeans technology and quite modular approaches – allowing developers to focus on just one module independent of the others.

- BIREME/OPS is preparing its ISIS-NBP (Network Based Platform), using a different storage back-end (ZopeDB) but also integrating it closer with a CMS (content management system), i.e. Plone, and with the intention to move all their ISIS-based applications to this new platform as from 2009. All technology used therefore is FOSS.
- BIREME/OPS with the support of others (e.g. the Flemish Interuniversity Council) is developing the earlier mentioned ABCD software (see 3.) at this moment still based on the current (expanded-capacity) WXIS ISIS-server but transferred to the new ISIS-NBP as one of the first early in 2009. The Flemish Interuniversity Council will support this project to strengthen it and elaborate it for university libraries in the South (e.g. using MARC21, including other standards such as METS, OAI/HP etc. but also expects the software explicitly to be presented as fully FOSS.

With this last project, ISIS will be able to offer a software which can be compared to other major players in the FOSS Library Automation like e.g. KOHA and NewGenLib. This will hopefully also eliminate misunderstandings about the nature of ISIS as an open textual database environment, which in itself is not a library system (but library systems can be based on it), and avoid the inappropriate comparisons found in the literature, such as 'KOHA is more suited for your library automation than ISIS' (see e.g. Étude comparative de Koha et CDS-Isis, Tristan Müller et Daniel Caissy, Canada, http://bibliothequeglobale.org/sigb_libre/FBG_OSIGBL_Etude2007.pdf), a study done for eIFL in 2006. After comparing MySQL to WEBLIS (another web-based integrated library system based on ISIS) one could easily conclude that 'WEBLIS is more suitable for your library automation than MySQL' – this makes exactly as much or as little sense.

The next step : more active involvement of the ISIS-community as developers in the open-source ISIS-projects…

UNESCO, after having decided to fully move to FOSS, still faces lots of difficulties in deploying all necessary elements for this. E.g. it is quite difficult to 'publish' sources which were not written with this aim, therefore have different (or 'no') styles of documenting the code as only internal programmers needed to be able to 'read' them. The same goes for Bireme as the other main source of code for ISIS-software. They are currently working on this problem, trying to make the sources more consistent and readable, but that is a major effort and will take some time to finish. So at the moment of writing this, not everything is available as open-source in reality, but the intentions are clearly there and publicly announced.

One of the factors which has made full deployment of 'open source' development of ISIS software more difficult than in other more typical software environments, is the relatively poor level of technical skills of the typical ISIS-users (mostly small institutes and organizations in the South), meaning they are not ready or don't feel sufficiently knowledgeable to actively contribute to new developments. The OpenISIS team (working on OpenISIS in between 2003-2005) e.g. was quite disappointed by the poor response level to their projects on SourceForge – and ended up turning away from ISIS, while nevertheless still using the old ISIS-ideas and principles for their database work in their current projects of Selene and Malete (see http://www.malete.org).
Also the new ISIS-softwares (e.g. Bireme's 'Network Based Platform' and ABCD) are mainly being 'fed' by their own developers while contributions from others will be welcome to expand the team and expertise in the near future. While programming input

(‘producing code’) is certainly not the only input needed in an open source project, other contributions by the ‘community’, e.g. user-feedback and suggestions, documentation and training materials, are very well possible. Only then ISIS will have fully developed from ‘free and open’ to ‘free and open source’ and this remains a major challenge to secure the future of ISIS.

List of Acronyms used :

API	Application Programming Interface
BSD	Berkeley Software Distribution
CDS/ISIS	Central Documentation System/Integrated Set of Information Services
DLL	Dynamically Linked Library
FAO	Food and Agriculture Organisation
FOSS	Free and Open Source Software
GPL	GNU General Public License
ILO	International Labour Organisation
LGPL	GNU Lesser General Public License
MFN	Master File Number
OPAC	Online Public Access Catalogue
PFT	Print Format Table
UN	United Nations
UNESCO	United Nations Education, Sciencd and Culture Organisation
WHO	World Health Organisation