**Digital Library technology in J-ISIS : concept, implementation and comparison with GSDL.**

by H. H. Berhe (Vrije Universiteit Brussels) and E. de Smet (Universiteit Antwerpen) [[1]](#footnote-1)[[2]](#footnote-2)

*Abstract*

Although the new J-ISIS software from UNESCO, based on Berkeley DB and Lucene technology, complies with some of the technical requirements as seen necessary for digital library applications, an easy way of building collections so far was not available. In this paper a report is given on some necessary DL technological requirements, such as the capability to deal with any metadata structure and alphabets and full-text indexing of documents of any length, and how J-ISIS can deal with these, as well as on the production of the DL interface for digital library applications based on Tika technology.

A brief comparison is made with a well established DL software, i.e. GreenStone Digital Library, regarding the concepts and performance. While using a quite different architecture and approach, the test shows that J-ISIS can process the documents faster and with more economical storage efficiency, inviting UNESCO to invest more into it in order to allow incorporation of some more advanced features like Greenstone’s capability to process intra-document segments and images, but also to allow for new exciting features for digital libraries such as interactivity.

Table of Contents :

Introduction : J-ISIS as UNESCO’s new addition to the ISIS-software family

Some Digital Library technological requirements

Testing J-ISIS technology for Digital Library applications

Production of a new Digital Library collection building feature

Comparison of concept and performance with GSDL

Conclusions and recommendations

# Introduction : J-ISIS as UNESCO’s new addition to the ISIS-software family

In spite of the fact that in the last few years UNESCO’s ‘General Information Programme’, once closely linked to CDS/ISIS as its most successful software tool in terms of downloads and high usage esp. in developing environments and Latin America, has dropped its direct input into the software development, still a major effort by mr. J-C Dauphin (who retired from UNESCO in 2010) was supported to develop a quite new, in some ways revolutionary version of the software to replace the popular WinISIS : J-ISIS. J-ISIS (J for Java) claims to replicate the full functionality of WinISIS but at the same time taking away its technological limitations, such as :

* Windows only, by using the Java-platform which is known for its radical platform independency;
* Database and record-size, by substituting the proper ISIS storage technology of the MST/XRF-files completed by a B-Tree indexing engine with the technology of
  + the embedded no-SQL database ‘Berkeley DB’ (at this moment owned by Oracle but maintained as FOSS) with limitations only defined by hardware;
  + Lucene-based full-text indexing and searching;
* ASCII/ANSI coding, now replaced by the UNICODE capability of Java.

However, in order to maintain the ‘core’ of the ISIS-technology, the following concepts are still fully present in J-ISIS :

* Schema-less records, i.e. the field-structure remains individually assigned to each record, therefore representing the ‘documentary’ approach of a database : each record can be a document with its own structure.
* Variable length fields, subfields and occurrences of fields : as a consequence of the previously mentioned feature, records store mostly textual values of any length in fields and subfields and can have 0, 1 or more occurrences, reflecting the original ISO-2709 standard on which e.g. MARC as a bibliographical format is based. In J-ISIS the ISO-2709 standard is maintained for compatibility purposes (e.g. to allow easy migration from the many existing ISIS-databases) but supplemented by XML-based structures, since the ‘key-value’ approach of Berkeley DB as a ‘no-SQL’ database leaves the structure of the ‘value’ part to the application itself.
* A powerful ‘parser’ to mould, format and extract values from the database, i.e. the ‘ISIS Formatting Language’ ; this language acts as an intermediary layer to allow the system managers (mostly the librarians themselves, often working in a context – typical for Southern developmental conditions – of very low IT-support) to define to a large extent how their application will look like (appearance) and will work (functionality). E.g. to illustrate this last aspect of functionality, the FL defines not only which elements of databases will be shown and how, but also which will be used for sorting and indexed, but also exported (therefore offering possibilities for conversion to almost any format).

Esp. due to this last feature, i.e. the continued use of the ISIS FL, we claim that J-ISIS is still a real ISIS-software family member.

# Some Digital Library technological requirements

When looking at the needs of digital libraries, esp. as a means to avail electronic documents of all sorts in a specific, often local context – we bypass here the very big Digital Library initiatives such as Google Books or the EU’s ‘Europeana’ -, we note the following requirements :

* Independency from fixed metadata formats : while – as is the case with the growing ‘Semantic Web’ – most information structures are based on XML and Dublin Core, for many reasons still many additions are made to the standard, resulting in a multitude of metadata sets which allow locally adapted and specific structures; this requires a database-setup which is not imposing its built-in schema onto the applications.
* Independency from specific scripts and alphabets : again due to the nature of often local projects and collections, digital libraries – esp. the ones in the Eastern part of the world - have to deal with several scripts and not only the latin scripts; for this reason UNICODE is the established standard. Technically this requires I/O techniques which can cope with multi- and variable bit-length of the stored values.
* Independency from document length/size and format limitations : electronic documents come in many varieties : PDF, MS Office formats (old and new), Open Document Formats, RTF, XLS, email-formats etc., varying from short notes (e.g. e-mails) over typical professional and academic documents (articles, papers) to full books and theses. All of these need to be dealt with by the Digital Library technology chosen for the purpose.
* Full-text indexing, ranking and searching : different from classic traditional libraries, digital libraries not only use the added-value ‘metadata’, which describe the documents in several ways (contents, preservation, digital rights…), but also the full text of the documents themselves for retrieval. This is a very strong asset as against printed collection implementations, but requires the technology to do full-text indexing and, related to that, to present the results in a relevance-based ‘ranking’.
* OAI-compatibility : in order to promote inter-operability of collections – each of these using their own formats, software… - the Open Archives Initiative provides a protocol for application-independent metadata-harvesting (PMH), allowing information seekers to retrieve from a world-wide network of ‘repositories’ the metadata of digital library documents.
* Interactivity : this is a revolutionary additional dimension made possible by the relatively new web-based interactivity which made the WWW so popular nowadays. The idea here is to elevate ‘passive’ readers of documents to active contributors, adding value to other readers with their comments, additions… into the documents but preserving their authenticity.

# Testing J-ISIS technology for Digital Library applications

Based on the above listed requirements, we can assess now J-ISIS on each of the requirements.

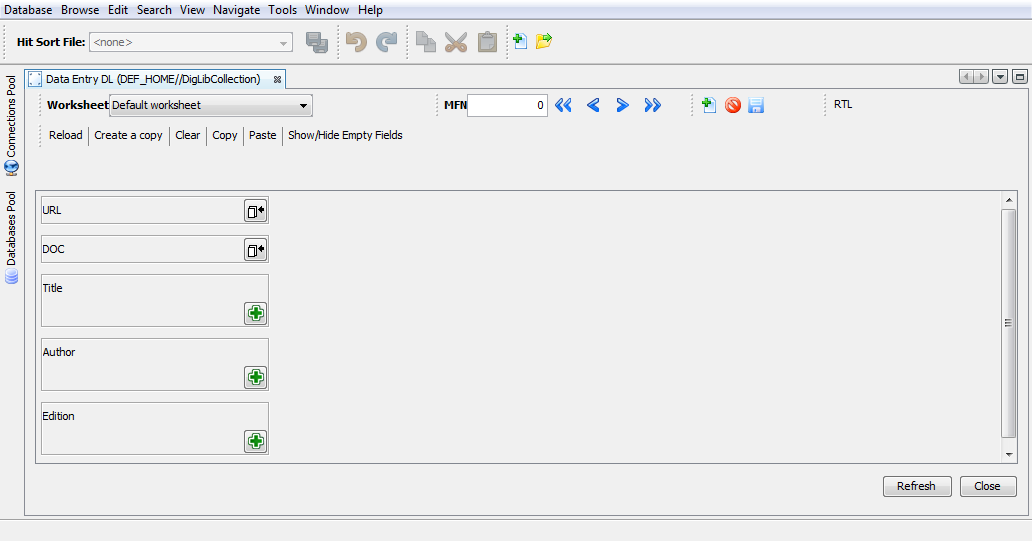
## Independency from metadata structure

Since J-ISIS is a full member of the ISIS-database philosophy, which always has maintained the ‘schema-less’ approach of using records without pre-defined structure, which instead is described in a very efficient and concise way in the ISO-2709 header of each record, librarians and information managers using ISIS can easily define their own structure, even without any stress on efficiency or consistency. ISIS’ built-in flexibility of allowing any field of any length (within the platform’s limits) to be used in any number of occurrences, easily provides the possibility to define all sorts of structures. This is one of the main assets of ‘no-SQL’ databases as opposed to SQL-based applications, in which the efficiency and ‘ACID’-icity is based on a strictly set-up architecture of related tables for which primary keys and links have to be defined in advance, while ISIS can use, e.g. with its ‘REF’ and ‘L’ functions in the Formatting Language (also fully implemented in J-ISIS), a so-called ‘run-time relational capability’ and doesn’t bother about defined but unused fields, even more : about non-defined but used fields !

In other words : it is rather easy in an ISIS-based approach to allow managers to define and alter their metadata sets, e.g. starting from Dublin Core and adding elements as necessary at any time.

E.g. adding now full-text document capabilities into existing ‘bibliographical’ applications (where the bibliographic fields then act as metadata) is no problem at all.

Here is an illustration of a data-entry worksheet in J-ISIS combining classic bibliographic fields with Digital Library fields such as URL and Document Text :



## Independency from document formats and size

The classic ISIS-technology, i.e. ISIS before J-ISIS, has used the concept of variable length fields since its origins (de Smet, 2008) but of course with limitations imposed by the technology of the time. Micro CDS/ISIS e.g. started as a software to run on 2 floppy disks (A: and B:) and obviously this imposed severe limitations. These limitations do not allow the technology to act as a basis for modern Digital Library applications, even if we have proved in our tests and experiments of introducing a Digital Library feature into the ISIS-based ILS ‘ABCD’ (Berhe/de Smet, 2011) that still a lot is possible, esp. when using the ‘extended’ version of ISIS which allows records of up to 1 Mb size each. Most documents after the text-extraction (performed by a tool like PDF2HTML or Tika) will still fit within this limit as we found out.

With the use of Berkeley DB as a storage engine in J-ISIS, these limits are now dropped and we consider this as a major step forward for ISIS, esp. looking at the potential to act in Digital Library applications, which indeed is the very essential aim of this paper.

J-ISIS, as we will see further on, uses Tika for text-extraction, which then will be stored into the database and full-text indexed by Lucene. Tika is one of the many important Apache Foundation projects (see: [www.apachefoundation.org/project/tika](http://www.apachefoundation.org/project/tika) ??), providing a Java library which can deal with most document formats (including the OOXML-based Microsoft formats), but also with image and multimedia files, to extract their text contents along with their metadata.

We tested J-ISIS for this purpose and can report that indeed all documents we tested were successfully converted and stored. In addition, tests of quite large documents (with original size of over 10 Mb and text-only sizes of over 2 Mb) were no problem – however after having coped with some challenges in the underlying Java-coding – and were processed in fact in a remarkable fast way, meaning in reality any document was processed just after a few seconds.

The following table gives some typical pre- and post- text-extraction sizes of some typical document formats, and the time it took in J-ISIS to process them : (??)

## Independency from scripts and alphabets

As we claimed before : our interest is mostly into the many smaller, often local initiatives of digital libraries, certainly within the focus and interest of UNESCO. Many of these, esp. from Eastern Europe or Africa and then further onwards to the East of the globe, use non-Latin scripts and alphabets : from the Greek and Cyrillic, the Ethiopian Amharic and the large community of Arab speaking countries to the major markets of Hindu, Chinese, Japanese and many other Asian scripts.

J-ISIS is fully UNICODE because of being Java-based. The illustration below shows how the WWW-client of J-ISIS (Web-JISIS) can search and display a document in Chinese, while our tests were also positive for e.g. mixed scripts (i.e. Amharic and English) and Arabic.

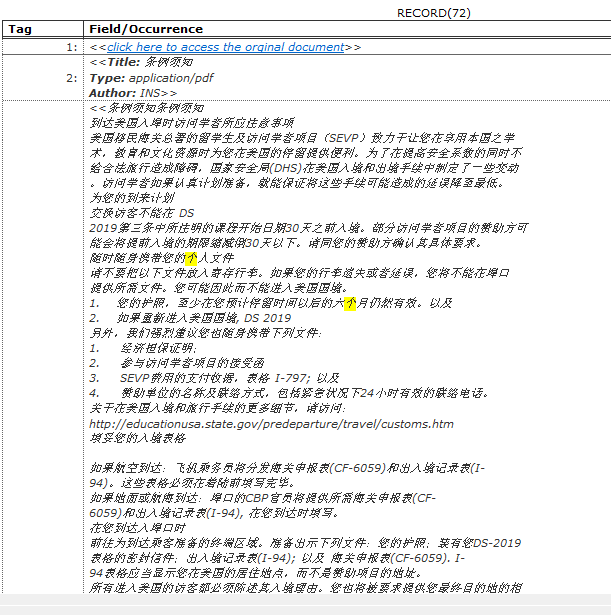


Figure 2 : Chinese document in Web-JISIS after having done a search with highlighted search-keys.

## Full-text indexing, ranking and searching

One aspect in which Digital Library technology by far exceed the capabilities of normal, traditional libraries, is the capability of dealing with the full contents of the documents provided.

In J-ISIS this is covered by the use of Lucene, which is a well-known engine (again of Apache Foundation) used in many applications and products. An example is GSDL, where Lucene can be activated as an alternative engine to their own ‘MGPP’ indexer. Also existing library automation systems, e.g. the one of the University of Antwerp (Brocade, based on Caché database technology) can and actually do use Lucene.

So, this capability was already fully present in J-ISIS before we started thinking about its Digital Library capability, and is now accepted with gratitude for this purpose in J-ISIS.

The ranking feature of Lucene might still not be fully implemented or cleared out, and does not include advanced concepts such as used in Google, but a simple test of creating some records in which one specific word occurs resp. once, twice, three or more times, showed us that – after elimination of the coding line which makes J-ISIS, for compatibility with its predecessor WinISIS, sort the search results by their Master File Record number by default - ranking indeed can be used in J-ISIS, which would be a strong feature for Digital Library applications.

## OAI and interactivity : two more features on our ‘wish-list’

A full ***OAI-PMH compatibility*** is to be offered at the general level of the database system of J-ISIS, i.e. not only for digital library applications but also for others like catalogs. Therefore OAI-compatibility is or should be part of the general J-ISIS development plan. We only emphasize this, by repeating the need for inter-operability, allowing non-ISIS users to also have access to records in J-ISIS collections using one or more of the (limited set of) basic OAI-operators.

We mentioned ***interactivity*** as a Digital Library feature, based on some authors claiming that Digital Libraries can also create added value by allowing readers of documents to add their own comments, insights, additions to existing documents. In fact the whole ‘social media’ interactivity of the modern Web 2.0 could be mentioned here as the context. While traditional libraries also and since longer practice such ‘community’ ideas, e.g. by organizing readers’ evenings where certain books were discussed and commented, the idea is not even that much revolutionary, except for the extent to which this can be practiced now, using current interactive WWW-technology.

Since J-ISIS is fully database-based, and with the features of repeated variable-length fields being already available, adding such functionality into J-ISIS should not even impose a major challenge, in our view. E.g. the WWW-JISIS interface, using JSP with Struts-support to make it quite interactive (e.g. when typing search keywords, a list is dynamically provided adjusting after each added character), only needs in principle some worksheets allowing end-users to add contents and store it with the original documents for viewing them on request.

We consider this as a major candidate for further development on J-ISIS for Digital Libraries.

# Production of a new Digital Library collection building feature

The database management part of J-ISiS, allowing the creation and editing of records, is based on a ‘worksheet’ approach, in which the contents of fields are available in interactive ‘fields’ of the interface (whether in the Java-client or in the WWW-browser). Such worksheets are based on pre-defined lists of fields (taken from the Fields Definition Table) with their own characteristics, and any combination of fields can be used to create a worksheet by the system manager.

Since we wanted, and this is a crucial concept of our implementation, to preserve the qualities of J-ISIS as an ISIS-based software by allowing system managers to create any database structure, in which also classic bibliographic fields (or now : metadata) can be combined with the new Digital Library types of fields, i.e. URL for providing a link to the original document and DOC to provide a pre-view of the document-contents, our digital library implementation in J-ISIS is based on the following steps :

* Introduction of 2 new field types ‘URL’ and ‘DOC’ : this is done at the stage of FDT creation in J-ISIS, simply adding (into the coding) the types along with the pre-existing 15 types of fields
* Parsing the FDT whenever opening a worksheet to check if any of these Digital Library fields are present, and if so : showing a special icon for uploading (the URL and document-text resp.) – the figure 1 above illustrates this special icon in the worksheet.
* When clicking on the special icon, a file-selector (taken from the rich Java set of libraries) allows to identify and upload the document and store the file-name and document text-contents into the resp. fields of the worksheet.’
* When storing the record in the worksheet into the database, immediately also Lucene will perform its duty of indexing the new fields, if included into the ISIS Field Selection Table, i.e. again respecting the general approach of ISIS allowing full control by system managers through such tables (FDT, worksheets and FST).

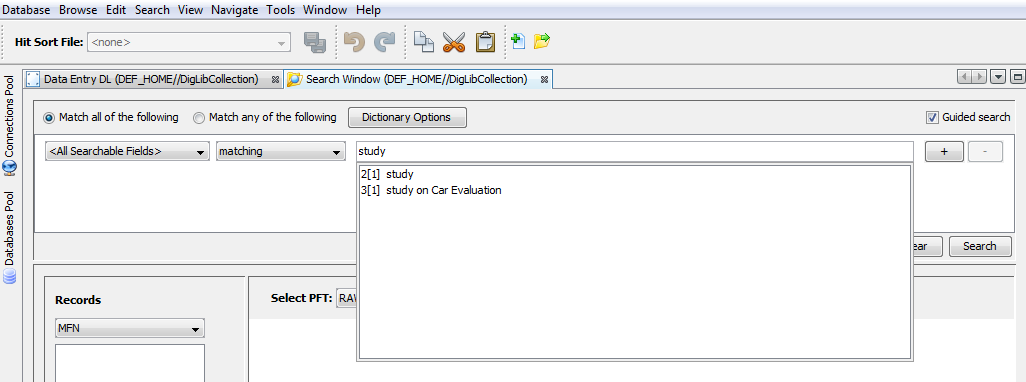
While coding this new feature into J-ISIS, it was found out that some special care had to be taken when doing such coding in order to cope with the challenges of quite big strings. We experimented, reacting on ‘out-of-memory’ problems, with e.g. writing the Tika-extraction to a file on the hard-disk first instead of into-memory, but found out that this works only well (i.e. fast) on small-up-to-medium-size documents, necessitating careful coding of the writing- and reading methods in J-ISIS. The default limit of Tika, truncating document-text extraction to 100Kb had to be eliminated with e.g. the following line of coding :

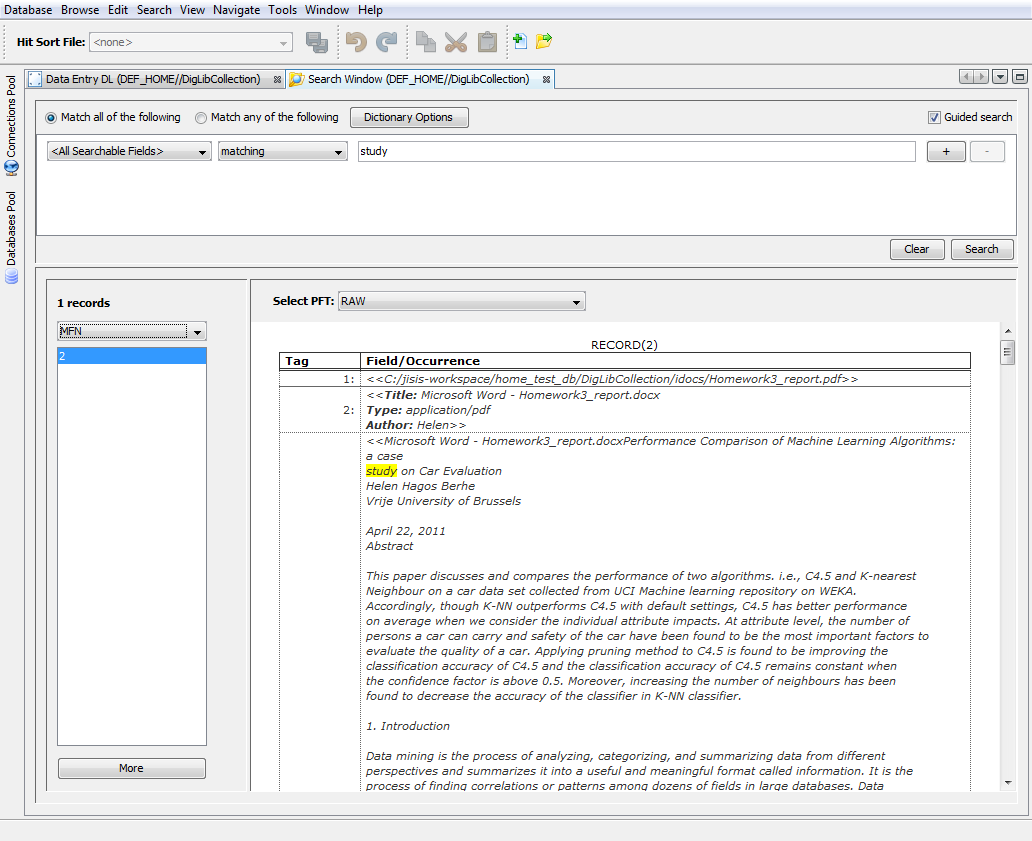
ContentHandler handler = new BodyContentHandler(-1);

Other problems we experienced, again showing how coding for digital libraries requires special attention to large-strings manipulation, were related to the setting of the logging-level (if set too high it would considerably lower the processing speed in larger documents) and the need to avoid the use of a pre-existing Java-class for ‘reading’ strings, which was – undocumented and probably also unwanted – limited to strings of max. 64Kb.

Our tests showed that adding a new document into a collection this way only takes a few seconds, mostly depending on the speed of navigating into the file-system to identify the document. This brings us to the next chapter where some comparison is made with the Greenstone Digital Library technology re architecture and performance.

The following illustration shows how the existing J-ISIS search-interface shows a resulting document.





# Comparison of concept and performance with GSDL

From the well-known and widely used ‘Greenstone Digital Library’ we adopted the following concepts :

* Flexibility of meta-data sets (see supra), where GSDL also allows editing any metadata-set on top of providing the major standards such as DC and SCORM.
* Previewing the document content along with providing a link to the original document.
* Using the embedded capability of the text-parsers (Tika in the case of J-ISIS, the Perl-plugins in the case of GSDL) to extract possibly available metadata.

Different from GSDL however are the following technical ideas in J-ISIS as a Digital Library tool :

* Whereas GSDL uses the file-system as a ‘database’ environment to keep track of the documents (with an organization in a hierarchy of folders with hashed names), J-ISIS is fully database-oriented, which we claim would yield into better performance esp. at higher levels of document numbers.
* GSDL seems to store the original documents (e.g. PDF’s ) three times : in the ‘import’ folder, the ‘archives’-folder and once again in the indexes-folder. On top of that the text-contents of each document are represented in XML-formatted files. J-ISIS in our implementation only stores the original document in the ‘idocs’ sub-folder of the database and the text-contents only once within the database, from where it can be loaded using the ISIS Formatting Language.

As a rather simple but useful first test, we built a collection of 200 documents, a typical mix of mostly PDF with some .doc, docx, .xls, .htm(l) etc. The size of the documents varies from the typical minimum sizes for HTML-document of about 20Kb to one very large PDF of more than 200 Mb. The average size was 857 Kb., whereas the resulting text-sizes vary from min. 1,3Kb up to max. 892 Kb with average of 82 Kb. The correlation in between both sizes (original and after text-extraction) is only 0,31, showing that documents vary rather a lot as to the share in them taken by the text-contents. Very large documents mostly are large because their images-contents, not their text.

The table below summarizes some comparison results, but one should mostly take into account the rather different interaction type in building collections : GSDL offers a batch-import (which can be handy in the initial building of a collection) whereas J-ISIS only offers, at this time, one-by-one creation/addition of digital collection records :

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature/Criterion** | **Greenstone Digital Library System** | **J-ISIS** | **Comments** |
| Success coping with test collection documents | 91% | 100% | GSDL: v2.85 with new PDF-Box installed as instructed |
| Total time to build collection | 15 min | 33 min. (10 seconds average per document) | GSDL : batch import possible, J-ISIS : speed highly dependent on file-selection interaction both : all docs in 1 folder available |
| Average time adding 1 document | 9 min | 10 sec | GSDL : minimal rebuild only, no hashing for optimal speed, plugins list optimized to collection profile both : no meta-data added but embedded meta-data processed |
| Indexing full collection | 25 sec | 9 sec | Lucene is used in both cases |
| Total storage for collection | 626 Mb | 204 Mb | GSDL : keeps copy of original files in 2 locations |
| Space occupied by indexes | 223 Mb | 23 Mb |  |
| Web-interface end-users | yes | yes | J-ISIS : Web-JISIS prototype only |
| Possibility to edit text-contents of document | no | yes | e.g. tagging inside document |
| Advanced features e.g. page-browsing, intra-document sections, image processing | yes | no | GSDL, as a dedicated DL software, emphasizes these features but they are not used often except for images |

Brief discussion :

* GSDL did not recognize or accept all documents; this result for sure could be improved by correctly applying the instructions on activating a more recent version of ‘PDF-Box’ and the Word-to-text feature of MS-Word; we didn’t try this but we also wonder if many librarians using GSDL would do so (or be capable of doing that).
* With the new ‘EmbeddedMetadataPlugin’ of GSDL v2.85, it seems that even adding one single document in the collection implies re-applying this plugin into all documents, of course slowing down largely the time needed for adding such one document, even in the ‘minimal re-build’ option.

Based on such comparison, which for sure has its flaws as it should reflect more the architectural differences, we believe we can still conclude the following :

* J-ISIS is certainly as fast, if not faster, in processing documents in comparison with GSDL
* J-ISIS is certainly (much) more efficient in the necessary storage for collections. This will become even more important when larger collections are to be dealt with. However others may claim that storage has become so cheap nowadays that this should be hardly a concern. Still in view of e.g. the need to backup and still-existing limited resources esp. in smaller libraries and projects in the South, we think this could be an important observation.

This comparison for sure does not full right to the GSDL software, as it provides more advanced features such as ‘section’ based indexing (taking into account intra-document boundaries) and ‘classifiers’ for browsing (which in J-ISIS should be done by the use of different sorting mechanisms of search results) or specific nice features such as the ‘PagedImagePlugin’, which allows paging through a collection with mutual links in between images and texts of documents.

On the other hand, using more advanced techniques of ISIS, esp. its Formatting Language, one could imagine also nice more advanced Digital Library features to be already – in principal – being available for further development and testing.

# Conclusions and recommendations

With the new Digital Library feature for easy building collections now being available in J-ISIS, along with the presence of well-proven and positively tested features like structural independency, capacity of dealing with any document sizes and formats, UNICODE and Lucene-based full-text indexing and ranking, we believe that J-ISIS could be a strong new addition to the market for Digital Library solutions, esp. where existing ISIS-data and –skills are available.

We strongly recommend UNESCO, in view of its mandate to support such information management initiatives esp. in the South, to re-activate its support for J-ISIS in view of its potential.

1. With full credits being due to mr. Jean-Claude Dauphin for his invaluable help with the J-ISIS coding challenges. [↑](#footnote-ref-1)
2. This article is based on the MA-thesis ‘Extension with Digital Library Technology of UNESCO’s J-ISIS Database Software’ submitted by H. H. Berhe, January 2012, VUB Brussels, Faculty of Engineering [↑](#footnote-ref-2)