

# Issues in the Development and Operation of a Digital Library

Sarantos Kapidakis

Institute of Computer Science, FORTH,  
Heraklion, Crete, Greece, GR 71110

`sarantos@ics.forth.gr`

and

Operations Research Center, MIT,  
Cambridge, MA 02139-4307, USA

`sarantos@mit.edu`

**Abstract.** This paper briefly describes both organizational and technical issues and approaches involved in creating an operational digital library at the University of Crete, found at <http://dlib.libh.uoc.gr>. We investigate and describe our approaches and experiences, the last few years, on setting in operation a Digital Library with many collections.

We had to analyze the library goals and user needs, to select appropriate software, to make flexible design for the additional functionality needed, to adapt and extend the selected software to make it applicable to the current demands, to install and configure the software, to improve it using feedback, and to interact with document authors and librarians to make the digital library friendly, usable and easily maintainable, and even to collect and digitize the library material. The final system is operated by current library personnel.

The main technical issues are related to the design, implementation and application of features of digital libraries, such as multilingual storage and interface, generalization of the software to permit searching on heterogeneous collections, adding support for the Z39.50 protocol and tools that simplify the configuration, administration and data insertion to the digital library, as well as tools to input or modify the metadata and to upload data, when submitting new documents in the digital library.

## 1 Introduction

Although the spread of digital libraries is increasing [12,7,8], most related issues only have occasional working approaches. To find appropriate approaches for many issues and to put them together on a real operational environment is a hard job.

The digital libraries hold material online, on a digital form, and provide advance ways of searching and material retrieval, access and presentation. With the term *material* we mostly refer to both data and metadata (or “data for data”). The digital libraries support distributed collections of digital data all over the world. Users of the libraries can recall from their computer the data

they are interested in, and study digital copies of them without ever having to visit the building of the library itself. The digital library can contain text, picture, sound, video, etc objects, or mixtures of them.

We first overview the operation of a digital library, we will use DIENST as a specific example. DIENST [4,5] is the digital library system used in the Networked Computer Science Technical Report Library (NCSTR), and developed in Cornell University and is used as a search tool for distributed digital libraries.

Every object that is registered under a DIENST server is available in some digital formats and has metadata (like title, creator and abstract) associated with it. A user can browse through all objects registered on a server, where a list of descriptions with links to the documents is presented to the user. DIENST is also using the metadata on user queries, to locate the relevant objects, and provide the user with descriptions and links to them. If the user selects the link to an object, all available information (metadata) about the object is presented to the user, together with a list of all available digital formats that the object is available. The user can select specific formats and retrieve them locally, to access the object.

The formats that DIENST provides consist of either a single file, or multi files, which may describe text, pictures, sound, etc. Each file may represent either the full object, or parts of it (e.g. pages of a document). DIENST can also easily be augmented to support additional formats, using simple configuration descriptions of the formats.

When a request is sent to a DIENST server, the server examines the request, and if the reply refers to objects stored on other servers, too, the corresponding parts of the requests are forwarded to these servers (using the DIENST protocol) and all their answers are collected. Then, the merged answer is sent back to the user that issued the query.

DIENST is only using one language (English) for storage and data, metadata and form presentations. It is also oriented towards a specific type of collection, with the metadata fields hardwired into the software. It is using its own server communication protocol, which is open, but cannot interact with other applications. Finally, it includes no tools for easily adding new documents into the digital library.

On the next section we will present the most important organizational issues that we faced, and on the following sections we will examine the main technical issues that we faced and on which we contributed interesting solutions: multilinguality, interoperability among heterogeneous collections, interoperability with other systems, and creator and administrator interface. Our approaches are general enough to be applied to most other similar situations that we envision that digital libraries will be used on the near future.

## 2 Organization

As a part of library infrastructure project, we wanted to build a digital library at the University of Crete. This library would give remote access, using WWW, to

university material, such as master and doctoral theses, departmental collections, such as an archive of old maps, other university collections such as audio archives with talks on special events, etc. As this was a big project, many people were, part-time, involved, and work was shared with other projects. The coordination and interconnection of these people and activities was quite a complex job.

The organization of a project of that scale is in fact harder than solving specific technical problems. We will present the design choices we faced and our final solutions. Most of these problems could be approached with far too many ways, and many researcher groups may work on each one of them. We do not try to supersede their work, but to simply provide an adequate working solution, that performs well in our case. Where possible, we followed established and proposed standards. Then, our digital library can be used as a working example for study of the applicability of the specific solutions to each of these problems.

## 2.1 Requirements

In order to build a digital library for the university material, a lot of problems had to be solved. We first had to decide on the functionality of the whole system, the software to be used, and even the schedule of our activities and the initial content of the digital library. The University has a lot of useful material, that would like to make available on the digital library, but even the order of adding it on the digital library may be of strategic importance: some people do not easily accept the idea of exposing their personal work that widely. To help accumulating the material, the university society should see and appreciate the advantages of creating and using the digital library. Thus, we had to demonstrate a working environment quite fast.

The University departments are on many different locations, even cities, and the distributed aspect of the material of the digital library is essential. Also, we wanted to use software with open standards, so that we can convert our digital material, data and metadata, to other formats at anytime, for use with more modern software, that will surely appear in the next few years, and we are not tied to products of a specific software vendor.

The main purpose of this project is to make current and future university material available in digital form. Thus, we would like to avoid making new software that will be used only once, in this project, and will need a constant effort to maintain, and to use as much as possible ready-to-run already-developed code, or at least to share code with projects on progress, improving and extending it as needed. Any new code that will be needed for this project should be designed to be reusable and extendable for many similar requirements and future projects, ours and of others, or for public use.

Soon after the start of the project, we had to show results, set and satisfy milestones and go through evaluation processes. Thus, we should make a quick start, and make use of temporary working replacements for procedures and software that would be developed later on. This complicated our initial work, and produced a lot of otherwise unnecessary load, but was unavoidable in order to satisfy the demands of the project specification. We will only describe the final

approaches here, in the order we had to conclude them, as they depict better the long run picture of the digital library operation.

## 2.2 Workplan

We divided our work in three stages. The first stage advertises the digital library and illustrates it to the university society, with main objectives to make most of the user interface and functionality, to collect initial material, to start investigating the user needs and the existing material, and to make the initial project schedule design. The second stage establishes procedures and finalizes the software, by adding all useful functionality, such as tools for object creators and collection administrators, interoperability of heterogeneous collections and other software. In collections that the documents are submitted directly by the creators, specially defined tools are needed. The third stage completes the content of the digital library, by digitizing material that is in paper and not yet in digital form. At the end of these stages, the digital library should be able to operate with minimal overhead.

We decided to first collect material that will be of high demand, and is easier to start collecting: master and doctoral theses. We also selected the DIENST software, which has a simple and friendly WWW user interface, adequate for most user needs.

One of our first goals was to ensure the completeness of the material of the resulting digital library, as much as possible. It is the completeness of a library that makes the library precious.

We had to make procedures that will ensure that the already written material will be collected, and also no new theses will be lost. We started an effort to collect the completed theses in digital form, and avoid making a lower quality paper-scanned version of them, by contacting the graduated authors.

We had to study the format of the currently available documents, and decide on reasonable and convenient requirements that we should pose on future documents. We proposed to the University the requirements and procedures that were needed for getting digital copies of all new theses, and they were adopted.

We organized the digital material into many separate collections. Each of these collections contains logically correlated material and can be addressed or excluded on user queries, and can also be stored in different computers or locations, or managed independently, by different collection administrators.

Our investigation concluded that there is much more exportable university material, that is candidate for the next offered collections: selected diploma theses, technical reports, working papers, archives and descriptions collected by the department of History and Archaeology, material from the university museum of natural history, video tapes and photographs from various university activities, archive of Cretan literature, archive of old maps, videotaped courses, academic programs of the departments over the years, bibliography of the courses offered over the years, university announcements, incoming and outgoing public paperwork, various journals and magazines published by the university or its alumni unions, electronic profile of university personnel, etc. A more detailed

exploitation, with more personal discussions, will bring into light even more material that is either needed in digital form, or can be easily available in digital form!

We had to finger out the most common document formats, make the software aware of them and make tools to convert other formats into them. More formats were added to the DIENST known format specification. Among them, a new `html` format was developed, that also includes an internal set of files, where the hyper links can refer to, and can incorporate a variety of web format combinations.

We also had to install servers, to gather data and metadata, to cross check the master and doctoral metadata with official university records, to provide user's and reference guides, and to train personnel to use, configure and maintain the software, on its permanent operations phase. During this work, the issues examined in more detailed in the following sections become apparent:

### 2.3 Technical Challenges

DIENST is using the English language for the NCSTRL system - in the Computer Science field, the English language is well established. To handle large collections of documents described into many languages and to increase the applicability and usage of digital libraries on non English speaking countries, a multilingual design is needed [9] - this is much harder than a single language interface.

The library offers many collections, and each one of them has its own metadata fields, so that it can be searched intuitively. A master theses collection may have an *author* field, while a map collection may have *dimensions* fields. Both of them have a *title*, and we should be able to search all collections together when we want to search by *title*, and other common fields, but we should also be able to specify *dimensions* when searching only map collections. As collections may be added to the digital library at any time, we should not need to change or even reconfigure the remote servers, we should have mechanisms to know which are the metadata fields of any collection. This is collection interoperability [1].

We want to provide access to a DIENST digital library via the Z39.50 protocol, a well-established search and retrieval protocol. We map each DIENST collection to a Z39.50 database and use one Z39.50 server for each DIENST server. We directly support different metadata fields per collection, and the metadata fields may contain multilingual information and we provide hyper links to the digital data [13]. Any Z39.50 client can be used to access the digital library.

We built tools with WWW interface that can be used by thesis authors to directly upload the digital documents and submit the associated metadata, and by collection administrator to input or modify the metadata as well as inspect, commit or modify submissions [3].

Among other smaller contributions, we extended DIENST so that individual documents can also be stored on remote locations, and the digital library can provide and use a remote link to the document.

### 3 Multilinguality

The design and operation of the existing multilingual interfaces on WWW is affected by current limitations, and no generic interface approach is implemented, that uses a generalized solution under the existing restrictions from protocols and software. Most current limitations on the WWW-client are related to the limited number of character sets that HTML pages can refer to.

We had to design the multilingual data storage and handling, and the multilingual client interface design. Our multilingual extensions to DIENST can support queries and browsing of the collections of a digital library, in many different languages, and have been designed to be easily extendable to accommodate new languages. Our multilingual design is reusable - most of its components do not depend on DIENST, and may be appropriate in other similar designs.

All digital information, the data and especially the metadata, should normally be available in all languages of interest, so that they can be used both for searching and locating the appropriate information, according to the user specification, and for presenting it to the user. If translations are not available in some languages, the document displaying would be affected, but the retrieval functionality should not be restricted.

For example, if a search is based on a creator name written in English, documents that have no English translation of their creator names will probably not be included in the results. On the other hand, a listing of documents presented to the user in the English language should not omit documents from the list just because there is no available translation of their title in English.

Each document in the digital library can be stored in different formats, corresponding to different detail (e.g. resolution) or standards, and can either be expressed on a human language, e.g. text and speech, or can be independent of any language, e.g. picture and music. In the first case, multiple representations of the document, one for each language, are desired. When the user asks to see the document, he will usually select one of its available representations, according to his preferences. Thus, these different representations are handled just as different formats of the document, that can appear even in single language designs.

The contents of the document is not the only way of specifying searching criteria for it. More information that describes it is used, the *metadata*. To indicate the close relation of a document, all of its available formats, and its associated metadata, we call them together a (searching) object.

The metadata fields usually contain text, and must be available in each language of interest, to provide full searching and display functionality for the document. Even on documents that are independent of any language, their metadata are not, and all multilingual problems apply to the metadata, too.

In a WWW interface, the usual WWW forms are written in HTML, and are limited by the HTML directives. The fonts that are specified for the display are constructed to correspond to a specific character set. Each font and character set can represent up to 256 characters. The characters are grouped into sets according to usage, so that most languages (like the western European languages) can be represented by one character set. If the display must contain more than 256

characters, or characters from more than one character sets, the above scheme is inadequate.

Multilingual user input is also needed. In most cases, user input is expressed in one language that can be covered by a single character set: contains ASCII characters and specific characters of some languages. Using a user selectable character set and font, the user is able to provide multilingual input. In more complex cases, where characters from more than one character sets must be input at the same time, different techniques must be used.

### 3.1 The Storage Structure

For multilingual operation, multilingual information has to be stored separately not only in the data, but in the metadata and program configuration area as well. Depending on the storage structure that will be used, the handling code must be extended accordingly.

As each metadata field is usually stored separately, a major decision is how to store the new, multilingual, metadata. From the two approaches that seemed to be more appropriate, the introduction of new, separate metadata fields for the multilingual information, one for each language, is cleaner, as all data and their translations are clearly distinct, and seems to be more appropriate for the design of a new system, but would require a lot of modifications on an existing scheme.

We selected to use the same metadata fields as in the single language case, but to encode the contents of these fields in a way that is easy to be split into the contained translations of the different languages.

Instead of merely using the value of a metadata field, the field contains a *multilingual string*, which includes a substring for each available translation (for some or all of the desired languages). We can always use string operations to get the substring that corresponds to the desired language. Such multilingual strings can be used both in the data and in the permanent program configuration information.

A simple encoding seems to be the most convenient solution for the multilingual string: A selected character (or a combination of characters) that does not appear in the contents of the strings is used as a delimiter to separate the translations, one for each language. A predefined order is used for storing the different translations, and missing translations are denoted by an empty substring. The advantages of our approach are:

- It is simple to isolate the desired translation (substring operation).
- If the desired translation is missing, a translation to a different language can be easily used, based on preconfigured priorities.
- The full multilingual string can be propagated to the interface, and the language separation can be made there. Thus, the user can change the desired language of display without needing the results of a new query.
- The multilingual string is split into its translations and every translation is used without any processing, when constructing the indexes to the metadata

field, to produce an all-language common index, so that the user can search with keywords on these fields without specifying explicitly the language of the keywords.

- It is easy to add new languages on a running system.

### 3.2 The User Interaction

Users can change language any time they want to, and the current language specifies in which language (and character set) the user input is: we have added a user selectable option for the language to be used for the display of this page in every HTML page of the multilingual interface. When the user selects a language, this language will be used for every subsequent HTML page for the display and possible user input, until a new language is selected.

The multilingual interface is used for both searching and browsing. The current language is used for providing language specific selection (e.g. choice on the range of names) and language specific sorting (e.g. on sorting by name). Missing translations on the browsing key are handled by using browsing keys from other languages.

In the interface, we can add a new language with the least possible modifications. Still, concurrent display of many languages, from many character sets, is not possible under this kind of interface. More complex display solutions, with either java or unicode, must be applied to give a more general solution to the problem.

Thus, we also developed and provide a unicode-based digital library interface to the DIENST digital library software, without the use of Java. The unicode interface can be activated at the first DIENST screen, as its use is normally a matter of available resources, and not a personal choice. When unicode is active, a string, before being displayed to the user, is first transformed to the corresponding unicode string. With the unicode interface, the user is able to access documents and see results with text from many different languages in the same page.

## 4 Interoperability of Heterogeneous Collections

The DIENST software had to change substantially, to support *heterogeneous* collections, with different metadata: The metaserver, a separate program that informs the servers for the existing collections and the places they are available, must provide more information now, like the metadata fields of each collection and their translations to all supported languages. The query protocol had to be augmented to return all matching metadata, in a known order. Furthermore, for more query flexibility, we augmented the protocol to accept queries with values that can match any metadata field in the collection. This allows users to pose simple queries, without addressing a specific metadata field, or to combine this field with others. DIENST has two ways of searching: simple and fielded search.



## 4.1 Searching Anything

In the *simple searching* of DIENST, the user specifies a single value, that is matched against any object metadata field information. In the original DIENST code, this mode of operation would formulate a query where all field are or-connected. Now that the collections have different metadata fields, the query formulation would be collection dependent and much more complex.

We now use the *anything* fields for this functionality, for both performance and simplicity in query formulation. Although there is no actual such field in the metadata, the DIENST server uses this field like any other field, even on queries involving other fields, and actually matches its value against any metadata field. The indices of this field are the indices of all other fields together.

Also, as the simple searching is the most common user searching method, used in more than 60% of the searches (see [2]), our method provides more efficient query evaluation, by providing this extra indexing. Finally, this new field is available together with the other fields, in the fielded search form, augmenting the query expressiveness.

## 4.2 Fielded Search

In the *fielded searching*, searching is done based on the values specified by the user to match specific metadata fields. The search mechanism tries to match these values with the indices held for the corresponding fields.

When searching more than one heterogeneous collections, the user first selects the collections that he wants to search and then gives values for some of the fields that are common to all selected collections. If no other common fields exist, then only the *anything* field is available!

For example let us assume we have three collections. The first has the *title*, *author* and *abstract* metadata fields, the second has *title* and *author* and the third has *title* and *abstract* (figure 1). When searching only the first two collections, the user interface provides entries for values corresponding to the *title*, *author* and *anything* fields only, the common fields of the collections. It would be meaningless to search the second collection based on a value for the *abstract* field. Similarly, when searching all collections we can specify values only for the *title* and *anything* fields and when searching the first and third collections, the *title* and *abstract* and *anything* fields are available.

A new configuration file is used, which describes the local collections, with their names and descriptions. It also includes translations of the collection names and metadata field descriptions to all supported languages, and the order of reporting the fields in the different types of printouts. Finally, default values for metadata fields of collections not explicitly mentioned can be provided, simplifying the configuration.

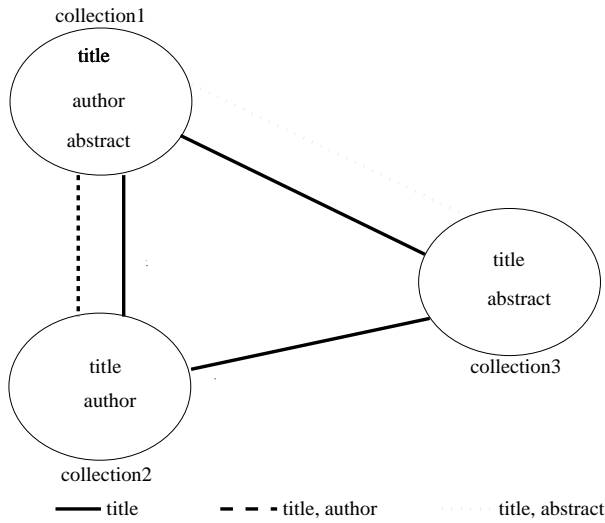


Fig. 1. Cooperation of three DIENST servers with different metadata fields

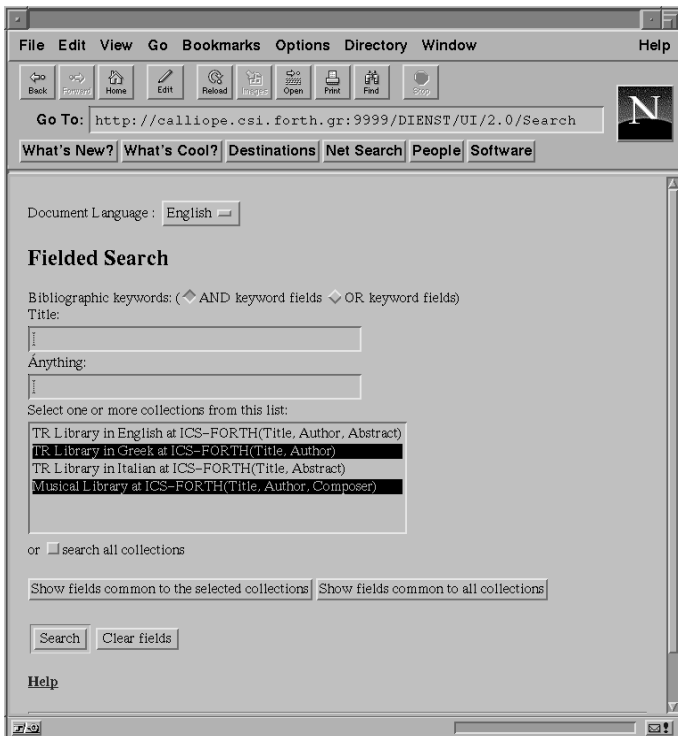


Fig. 2. New DIENST interface with the two additional buttons

### 4.3 Fielded Search Interface

In order to support collection interoperability, we extended the fielded search interface with two additional buttons, *Show fields common to the selected collections* and *Show fields common to all collections*, as is shown in figure 2.

Also, at any time, the displayed fields, where the user can specify values for searching, are always these common to all collections in the collection browser. Initially, all registered collections are available in the browser for user selection.

When the user wants to search in a subset of the available collections, using fields that are common to these collections, but not to all collections currently in the browser, after selecting the collections, he can use the *Show fields common to the selected collections* button, as in figure 3, to display all their common metadata fields. At the same time, the available set of collections on the browser is narrowed to only these that have all the displayed fields (which can be wider than the current collection selections), so that the user can still change collection selection - and specify any displayed fields.

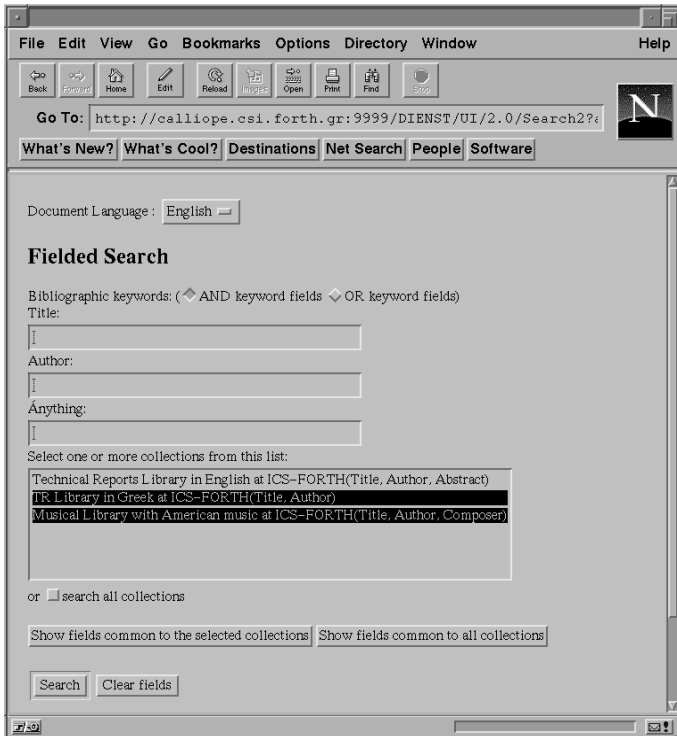


Fig. 3. A restriction on the displayed collections increases the displayed fields

The user can subsequently change collection selection, and possibly select *Show fields common to the selected collections* again to get access to more fields,

or specify values for searching. By selecting *Show fields common to all collections*, he can return to the initial state with all registered collections and entries for the globally common metadata fields. The selected collections do not change by changing the displayed fields, and all selected collections are always present on the collection browser.

In most cases, the users avoid using the fielded searching. As a consequence, searching on heterogeneous collections, and its interface, is even less popular: Whenever possible, it is better to define collections with always the same fields.

#### 4.4 Distributed DIENST Operation

The DIENST software is a client-server application where the users contact and query the DIENST server using their web browsers. The results are returned to the client application and are then displayed to the web browser. Each object of the digital library should be “registered” in the DIENST server, so that the server is able to know about its existence and access it when necessary.

When two or more DIENST servers cooperate in a distributed system then the search process dispatches many DIENST copies to query the distributed digital library servers in parallel. Each server searches its local collection and returns the matching items to the requesting server, to merge the results and reply to the user. For distributed operation, a metasever is needed to provide information about the configuration of all other servers.

To make DIENST able to search in digital libraries with different metadata fields, we have to add the names of the metadata fields we operate on as a new parameter to its search machine, and to extend the metasever protocol, so that it can also give information about the metadata fields of each server.

A server can consult its collection configuration file, to find the metadata fields of the local collections, and the metasever, to find the metadata fields of the remote collections. In distributed operation, the queries formed and the merging of the returned results make use of these fields.

We implemented a DIENST metasever (the DIENST distribution does not include one) that, in addition to its normal protocol, provides one more request, to report information about the metadata fields of each collection to the cooperating distributed servers, using the same configuration information with the DIENST servers.

## 5 Interoperability with Other Protocols

DIENST has a built-in mechanism for distributed search. We take advantage of the functionality of a DIENST digital library and the features of the Z39.50 protocol without modifying the code of either the DIENST or the Z39.50 server.

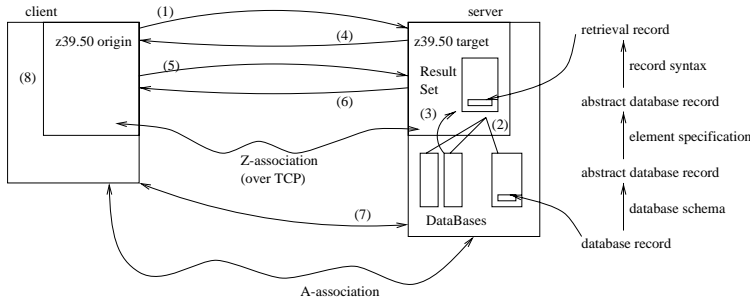
The benefits from providing Z39.50 access to a DIENST digital library in summary are: flexibility on the definition of the metadata fields (the fields can have a type); flexibility on the retrieved format of a registered object, we may make information available in many variants and we may also provide various

levels of structuring, from flat organization to arbitrary DAGs; transparent access to diverse databases, e.g. concurrent access to DIENST and other Z39.50-compatible library systems.

We have built a Z39.50 server that implements this functionality. The Z39.50 server is an enhanced version of CNIDR Isite; it uses DIENST metadata to build a Z39.50 database, accepts a query in order to perform a search in the digital library and provides URLs to the digital library objects identified by the query; last but not least, the Z39.50 server has the capability to search multiple Z39.50 servers concurrently and merge the results.

### 5.1 DIENST and Z39.50

Storage organization of a DIENST digital library is hierarchical: the library is distributed among various DIENST servers, each DIENST server provides access to a number of collections, each collection comprises of registered digital library objects. We map each DIENST collection to a Z39.50 database and use one Z39.50 server for each DIENST server. Our terminology is based on the ANSI/NISO Z39.50-1995 document [10], that fully describes the standard. The protocol interaction is depicted in Fig 4. Our work was the implementation of a mapping between a DIENST digital collection and the Z39.50 abstract model.



**Fig. 4.** Search and Retrieval using *Z39.50*

Typically, a Z39.50 server accesses a database via an API defined by the server (not by the Z39.50 standard), which we implemented. A different, generic, implementation would translate Z39.50 queries to DIENST requests, having the advantage of using the DIENST distributed searching capability.

However, the fact that DIENST collections have a moderate size and can be considered read-only — they change very infrequently, and not by the DIENST protocol — greatly simplifies our implementation and permits concurrent access to the registered objects without danger for damaging the consistency of the digital library. As a result, we can bypass the DIENST server and avoid translating from Z39.50 to the DIENST protocol, and the two servers need not be aware of each other. This approach is simpler to implement but requires to launch

an independent Z39.50 server and needs additional effort to provide distributed searching; this effort is not significant.

One of the design decisions was not to change the Z39.50 server protocol code. This decision permits the easy incorporation of additional functionality when new Z39.50-1995 facilities are implemented by Z39.50 servers; e.g. automatic configuration of the client according to the DIENST metadata associated with each object (*explain* facility).

The Z39.50 server that provides access to a DIENST digital library can be accessed by a generic Z39.50 client but there is a trend of Z39.50 clients towards the web. This trend was the motivation for enhancing a gateway so that it generates the search form *on the fly*. A file that contains global information as well as a description of the attributes which can be used to formulate a query is used by the gateway in order to modify an HTML template and generate the search form.

## 6 Submissions and Administrator Interface

The submission tools are normally sharing configuration files with DIENST. They provide functionality that will cover even rare requirements and offer many optional features, through configuration choices. For example, confirmation messages may be desired for destructive actions, or notification messages may be sent on specific asynchronous events.

The functions of the tools can be subdivided into three distinct but also self-complementary categories: metadata manipulation, data (digital format) handling and repository management. We will analyze each of the above separately.

### 6.1 Metadata Manipulation

No data are accepted, unless their metadata are already submitted. Especially for DIENST, the metadata file follows the protocol described in RFC 1807 [6], which proposes a generic format for organizing metadata.

This format is simple to understand. However, it can be quite difficult to consistently follow for users with little computer familiarization, as in fact are the majority of the people that are expected to submit an object in a library. Moreover, it can often be frustrating for collection administrators to maintain a digital library by editing such files by hand. Furthermore, as a digital library provides means to be accessed from the WWW, the submission of metadata is done in a similar manner, as in figure 5.

Also the extensions to manage multilingual objects and to handle heterogeneous collections require different and more complicated metadata. Our implementation manipulates metadata in a way that conforms to these extensions, providing at the same time a multilingual user interface, and the code that depends on the specific protocol in which metadata are stored forms a distinct module and can be easily modified to manipulate metadata for different systems.

Submit Metadata to the Database

Form Language English

Collection:	ICS-FORTH
Document Title:	
Submission and administration tools ...	English
	Greek
Author:	
Greg Karvounarakis, Sarantos Kapidakis	English
	Greek
Abstract:	
Submission and database management tools ....	
	English
	Greek
Phone Number:	326420
E-mail:	gregkas@ics.forth.gr
Publication Date:	March 1999 (Month Year)
Comments to the Librarian (if any):	

Fig. 5. Metadata submission by user

Moreover, RFC 1807 supports multi valued fields attributes in the form of repeated attribute value pairs. Thus, if an object has two or more *creators*, multiple lines for this field should be created in the metadata file. By default, such fields can be filled in as comma-separated lists. The user can enter, for example, the value *Greg Karvounarakis, Sarantos Kapidakis* as a value for the field *creator*. All these internal transcriptions are transparent to the user who wants to submit an object to the library, requiring only the filling of a form with the appropriate fields for each collection for the supported languages.

Some metadata fields (e.g. *submission date* or *CS-TR-version*) are automatically filled with default values. Furthermore, some of the fields have extra restrictions: for example, the field *id* denotes the collection in which the object is going to be placed and also its identification as a library object. While the collection name should be chosen by the user among a list of the existing collections, it is preferable that the *id* is automatically generated, to ensure uniqueness. Thus, we create an *id* based on the date of the submission, a serial number and a random code (serving as a temporary password, for security). Also, mandatory fields must be completed for at least one of the languages supported by the system. Finally, only the administrator is allowed to modify specific metadata fields, if so configured.

New submissions of metadata and data are normally placed in a temporary repository with the same structure as that of the DIENST repository, rather than on the permanent repository. It is also possible to have a hierarchy of

temporary repositories and collection administrators. Non administrators can only access and modify objects in this temporary repository. When a user tries to modify an object which has been placed in the permanent repository and has no entry in the temporary repository, a new entry is created to store the modified information (based on the original entry in the repository), associated with its permanent repository entry. New or modified submissions are later inspected by the collection administrator and, if approved, are placed in the repository. The creator can also attach contact information and comments about the submission, for the collection administrator.

## 6.2 Manipulation of Digital Formats

The tools can even handle complex digital formats like ones that are physically formed by a hierarchy of files. A degenerate such hierarchy is the scanned pages of a paper document. Apart from the validity of each file separately, we also check the ability to put these files in a logical order, and for missing pieces, in order to display them correctly. We handle file configurations that have an *obvious*, unambiguous order.

The digital library objects may be represented in several digital instances (formats). For example, a picture may be in different resolutions, and possibly according to different image standards, or a document may be accompanied by many translations. Several issues arise regarding the way these formats are uploaded to the library as well as the way they are organized in the repository.

**The metadata you sent were successfully saved**

**Registration ID : 1999\_03\_09-311-34762**

---


**File Format:**

**Local File Name:**

**Comments to the Librarian (if any):**

---

 **NCSTRL**  
 This server operates at Test Ics Collection.  
 Send email to [gregkar@e-si.forth.gr](mailto:gregkar@e-si.forth.gr)

**Fig. 6.** Submission of a data file by a user



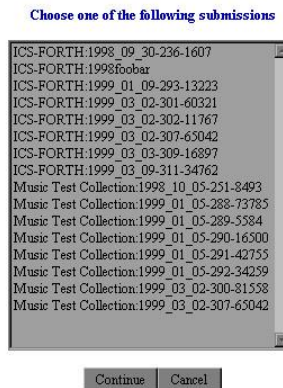
After submitting the metadata, the user can repeatedly select data files to submit and their type (or select auto detection of the file type) in a form such as this shown on figure 6. To upload the files, we use the proposed standard RFC 1867 [11]. There are ways to send not only more than one digital formats, but to change an uploaded format with a newer version or add new formats some time later, as well.

The description of the supported digital formats is also read from the DIENST configuration files. The submission process handles both metadata and digital formats, and stores them in a way compatible with DIENST, on a temporary repository. The collection administrator need only move the files to their permanent space. This will usually be done using our WWW interface offering a minimum effort procedure to inspect and discard or approve and commit to the repository each digital format separately or whole submissions altogether.

### 6.3 Repository Management

The administrator interface supersedes the creator interface and can be used to overwrite creator entries, or to create new entries, especially in massive object entry. Additionally, there are some more actions that an administrator is able to perform:

- Browsing, to see and select a submission on the temporary repository as in figure 7.



**Fig. 7.** Selection of a submission by the collection administrator

- Inspecting — approving a submission: He can inspect submitted objects to see if they are approved and placed in the digital library. This action actually consists of two parts.

First, the administrator has to inspect and possibly modify the metadata describing the object, to ensure completeness and validity as in figure 8. This is actually similar to the function of the modification of the metadata of a submission, and therefore a similar interface is used.

**Check Submitted Metadata**

Form Language: English

Collection: ICS-FORTH		
Document Title:		
<input type="text" value="Submission tools for digital libraries"/>	English	
<input type="text"/>	Greek	
Author:		
<input type="text" value="Greg Karvounarakis, Sarantos Kapidakis"/>	English	
<input type="text"/>	Greek	
Abstract:		
<input type="text" value="Submission and administration tools ..."/>	English	
<input type="text"/>	Greek	
Keywords:		
<input type="text" value="submit digital libraries tools administration"/>		
Publication Date :	Entry :	Registration ID :
<input type="text" value="March 1999"/>	<input type="text" value="March 09 ,1999"/>	<input type="text" value="1999_03_09-187-43263"/>
Comments to the Librarian (if any):		
<input type="text" value="PHONE NO: 226420"/>		
<input type="text" value="EMAIL: gregkar@ics.forth.gr"/>		

**Fig. 8.** Cross check of a submission by the collection administrator

Then, the administrator checks what digital formats have been submitted for this object and, if possible, whether they are valid (e.g. not corrupted). At the end, the administrator should either approve the submission, and commit it to the digital library as a whole or in part or reject it by erasing it from the temporary storage space, or leave it in the temporary space to be modified or inspected at a later time.

- New submission: This action is similar to that for a creator. However, the administrator is allowed to intervene to the creation of the submission. For example, an administrator should be able to change the *id* of a submission. Therefore, the administrator is allowed to overwrite the automatic *id* and other special fields that are created for the new submission. Moreover, after the metadata have been stored and the digital formats have been uploaded, the administrator is able to immediately commit the new submission to the permanent repository. This way, inspection and approval can be done on one stage.
- Modification of permanent digital library objects: The collection administrator is able to delete, modify or replace by a newer copies permanent

library objects, similarly to the modification of a submission that appears in the temporary space. For repository consistency, all modified or new submissions are first stored in temporary space. Then, the administrator can choose which parts from the older version should be kept, and which should be replaced by the modified, or new ones, as in figure 9.

The following files were found, regarding to submission with Registration ID 1999\_gregkar

Check to add new files (from tmp)	Check to delete old files (from db)
<input checked="" type="checkbox"/> 1999_gregkar.bib	1999_01_09-293-13223.bib <input type="checkbox"/>
<input checked="" type="checkbox"/> 1999_gregkar.text (ascii text)	(ascii text) 1999_01_09-293-13223.text <input type="checkbox"/>
<input checked="" type="checkbox"/> 1999_gregkar.ps (PostScript (v3.0) text)	
<input checked="" type="checkbox"/> contents.html	
<input checked="" type="checkbox"/> README	README <input type="checkbox"/>

Fig. 9. Update of the permanent repository by the collection administrator

It should be obvious that a major security issue emerges, by the introduction of such web-based administrative tools, since they can be used to modify or delete objects from both the temporary and the permanent repository. To cope with this problem, we applied access control to these tools, by using the access control options offered by the web-server. Since the whole application is based on server scripts, this is a simple way to ensure that only the persons that should use these tools will be allowed to access them.

## 7 Conclusions

Our digital library is easily extendable, and all its code can still be freely used by anyone. This is a complete digital library, ready to accept material and to operate, in an efficient and friendly way.

There are still many technical issues that deserve better solutions: we could use better communication protocols, improve the distributed performance of the system, or use a more general purpose multilingual solution and better interfaces. Automatic translations and thesaurus browsing would be a very useful interface feature.

But for a successful digital library, the organizational issues are the ones that play a key role, and also define the priority of the technical problems. We had to solve several problems, from software and interface design decisions to contacting authors for the collection of their material. One must be very close to the library, to set its policy directions and to see and satisfy its real needs

early enough. We hope that our organizational steps, as well as our technical contributions and software, will help others to reach their digital library goals easier and faster.

## 8 Acknowledgements

We want to thank the students that did the programming work on changing or extending the software functionality, as their diploma theses: Panagiotis Alexakos, Greg Karvounarakis, Iakovos Mavroidis and Giorgos Sapunjis, and Antonis Sidiropoulos for the extension for the HTML document format. Also, we would like to thank the supporting digital library team at the University of Crete, Grigoris Tzanodaskalakis, Flora Chryssou, Manolis Koukourakis, Bagelio Anifantaki, Eleftheria Prokopaki and Soula Koumaki, for their comments and feedback on the software and interface and cooperation and support on the digital library operation.

## References

1. Panayotis Alexakos, Sarantos Kapidakis: *Parameterized DIENST* <ftp://crete.csd.uch.gr/pub/thesis/diplom/jalexak/TR.ps>
2. S. Kapidakis, J. Sairamesh, S. Terzis, *A Management Architecture for Measuring and Monitoring the Behavior of Digital Libraries*, Proc. 2nd European Conference on Research and Advanced Technology for Digital Libraries, Crete, p. 95-114, September 1998.
3. Greg Karvounarakis, Sarantos Kapidakis: *Submission and repository management tools for digital libraries, with WWW interface* <ftp://crete.csd.uch.gr/pub/thesis/diplom/gregkar/TR.ps>
4. Carl Lagoze and Jim Davis. *DIENST: An Architecture for Distributed Documents Libraries*, Communications of the ACM, 38(4), April 1995, p. 47
5. C. Lagoze, E. Shaw, J. R. Davis and D. B. Krafft, *DIENST: Implementation Reference Manual*, TR95-1514, Cornell University, May 5th, 1995.
6. R. Lasher, Stanford, D. Cohen, Myricom, June 1995, *RFC 1807: A Format for Bibliographic Records*
7. Library of Congress, *American Memory* <http://memory.loc.gov/>
8. Los Alamos National Laboratory, *Library without walls* <http://lib-www.lanl.gov/lww/welcome.html>
9. Jacob Mavroidis, Sarantos Kapidakis: *Multilingual Extensions to DIENST* <ftp://crete.csd.uch.gr/pub/thesis/diplom/jacob/TR.ps>
10. National Information Standards Organization. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*, NISO Press, Bethesda, MD, July 1995. <http://lcweb.loc.gov/z3950/agency/document.html>
11. E. Nebel, L.Masinter, Xerox Corp, November 1995, *RFC 1867: Form-based File Upload in HTML*, <http://rfc.fh-koeln.de/rfc/html/rfc1867.html>
12. C. Nikolaou, S. Kapidakis and G. Georgianakis, *Towards a Paneuropean Scientific Digital Library*, TR96-0167, Institute of Computer Science - FORTH, May 1996.
13. Giorgos Sapunjis, Sarantos Kapidakis: *Z39.50 Access to a DIENST Digital Library* <ftp://crete.csd.uch.gr/pub/thesis/diplom/sapunjis/TR.ps>