A Distributed Catalog for Digitized Cultural Heritage

Marinkovic Bojan, Luigi Liquori, Vincenzo Ciancaglini, Zoran Ognjanovic

A Distributed Catalog for Digitized Cultural Heritage

Bojan Marinković[1], Luigi Liquori[2], Vincenzo Ciancaglini[2], and Zoran Ognjanović[1]

[1] Mathematical Institute of the Serbian Academy of Sciences and Arts, Serbia
{bojanm,zorano}@mi.sanu.ac.rs
[2] Institut National de Recherche en Informatique et Automatique, France
{luigi.liquori,vincenzo.ciancaglini}@sophia.inria.fr

Abstract. Peer-to-peer networks have emerged recently as a flexible decentralized solution to handle large amount of data without the use of high-end servers. In this paper we present a distributed catalog built up on an overlay network called “Synapse”. The Synapse protocol allows interconnection of different overlay networks each of them being an abstraction of a ”community” of virtual providers. Data storage and data retrieval from different kind of content providers (*i.e.* libraries, archives, museums, universities, research centers, etc.) can be stored inside one catalog. We illustrate the concept based on the Synapse protocol: a catalog for digitized cultural heritage of Serbia.

Keywords: Peer-to-peer, Distributed databases, DHT-based Overlay networks, Information retrieval, Digitized cultural heritage.

1 Introduction

1.1 Context

Digitization is an important step towards preservation and promotion of heritage. It safeguards cultural diversity in the global environment and offers a rich treasure to the world-wide public of the Web. Usually, digitization can be seen as a collection of activities, including digital capture, transformation from analogue to digital form, description and representation of heritage objects and documentation about them, processing, presentation and long-term preservation of digitized content, etc.

The document [13] states that current digitization practice in SEE is still not matching the priorities communicated on the EU-level and that the rich cultural content of the region is still underrepresented in the electronic space. One of the main principles accepted by the participants states that ”It is recognized that knowledge of the cultural and scientific heritage is essential for taking decisions concerning its digitization and for interpreting the digitized resources. For this reason, inventorying and cataloging should precede or accompany the digitization of cultural and scientific assets.”

At the moment, there is no widespread meta-data standard for describing digitized heritage in Serbia. Actually, although most of the institutions caring about national heritage have started the digitization process, there is no meta-data standard formally accepted at the state level. Because of that we are faced with something that can be called *the meta-data problem*. Different providers of heritage resources (libraries, museums, archives, some research institutions) use international standards appropriate for

M. Gusev and P. Mitrevski (Eds.): ICT Innovations 2010, CCIS 83, pp. 176–186, 2011.

their specific fields, or ad-hoc methods, or old procedures for describing cultural assets in classical format (formulated in 1980s or early 1990s). In fact, some providers are still waiting for some solution of the meta-data problem and do not do anything related to digital cataloging. This means that digital catalogs in Serbia, if they exist at all, cannot help in communication between different kinds of providers and users.

On the other hand, at the international level, there are plenty of meta-data standards for describing heritage resources, for example: Dublin Core [9], EAD [10], MARC [12], TEL AP [15], FRBR [14, 11], etc.

Given all of the aforementioned, the Committee for digitization of the UNESCO commission of Serbia has recognized the meta-data problem as the most sophisticated one in the cataloging phase of digitization. During the past years, some efforts were made in the field of standardization, which resulted in the development of the recommendation for the meta-data format described in [7], but this recommendation has not, still, been accepted as a formal national standard.

There were also some efforts directed towards developing technology for storing these meta-data documents, but there is still no widespread application. Recent attempts to create digital repositories, such as, for example, Europeana [16], are mostly based on centralized architectures. Here we consider an alternative, decentralized approach, based on overlay networks.

Overlay networks have recently been identified as a promising model to cope with the Internet issues of today, such as scalability, resource discovery, failure recovery, routing efficiency, and, in particular in the context of information retrieval. Many disparate overlay networks may not only simultaneously co-exist in the Internet, but can also compete for the same resources on shared nodes and underlying network links. This can provide an opportunity to collect data on various kind of digitized documents which are, by their nature, highly distributed resources, while keeping backward compatibility, efficient searching, failure resistance, etc. One of the problems of the overlay networking area is how different overlay networks may *interact* and *cooperate* with each other. Overlay networks are heterogeneous, and basically unable to cooperate with each other in an effortless way, without merging, an operation which is very costly since it is not scalable and not suitable in many cases for security reasons. However, in many situations, distinct overlay networks can take advantage of cooperating for many purposes: collective performance enhancement, larger shared information, better resistance to loss of connectivity (network partitions), improved routing performance in terms of delay, throughput and packets loss, by, for instance, cooperative forwarding of flows.

In the context of large scale information retrieval, several overlays may want to offer an aggregation of their resources to their potential common users without losing control of them. Imagine two companies wishing to share or aggregate information contained in their distributed databases, obviously while keeping their proprietary routing and their exclusive right to update it. In terms of fault-tolerance, cooperation can increase the availability of the system – if one overlay becomes unavailable the global network will only undergo partial failure as other distinct resources will be usable. The solution could be found in using a meta-protocol which allows a request to be routed through multiple overlays, where one overlay contains one kind of institutions, even using different routing algorithms, thus increasing the success rate of every request.

The ready-to-market DHT(Distributed Hash Tables)-based technology of structured overlay networks is enriched with the new capability of crossing different overlays through *co-located nodes*, *i.e.* by peers who are, by user's choice, member of several overlays [8]. Such nodes are themselves able not only to query multiple overlays in order to find a match, but also to replicate requests passing through them from one network to another and to collect the multiple results.

1.2 Problem Overview

In the digitization of catalog services, using an Information System (IS) has been shown to be essential in matching the offers, the requests, and the resources. The IS is, in most cases, a front-end web site connected to a back-end database. A classical client-server architecture is usually sufficient to manage those services. In presence of multiple services, for technical and/or commercial reasons, it is not possible to share contents across different providers, despite the evident advantage. In most cases, ISs are not suitable to communicate in any of their features (lookup, search, etc.). Although, in general, this does not affect the correct behavior of an IS, it is clear that interoperability would increase the overall quality of the service. Moreover, the classical shortcomings of client-server architectures make both services unavailable in case both servers are down. Any attempt to make different and disconnected institutional, client-server based architectures, does not foresee any form of service interconnection, with the unpleasant consequence of losing potential matches between offers and requests between users of different communities on the same subject.

As a basic example, let us consider two cultural institutions which contain digital documents inside their databases. One node of the first database stores one volume which is searched for by a node of the second one. Without "inter-network" co-operation, these two databases would never communicate together. But, if these ISs do co-operate with each other, the results can be far more precise and accurate.

As we said above, the digitized documents are, by their nature, highly distributed resources. Because of this, we decided to develop a catalog based on Synapse protocol as a real-life proof-of-concept. Here we analyze how the Synapse protocol [8] can be used as a tool to connect a huge number of content providers.

1.3 Outline

The rest of paper is organized as follows: In Section 2 we summarize some mechanisms proposed in the literature related to distributed systems and describes briefly the interconnection of different distributed catalogs by means of our Synapse protocol. In Section 3 we introduce an idea of a distributed catalog service and show how it is mapped onto a DHT. In Section 4 we show a running example with a proof-of-concept which we have implemented on the base of a real case of study. Section 5 describes the results of the deployment of a client prototype tested over a distributed platform at Mathematical Insitute. In Section 6 we present our conclusions and ideas for further work. [1]

[1] The developed software and all of the results of the tests are available at the web-page `http://www.mi.sanu.ac.rs/~bojanm/synapse`.

2 Related Work and Contributions

2.1 Related Work

The classical viewpoint on distributed databases, as it is described in [1], defines desirable properties for distributed database management systems (DBMS). These properties are: *distributed data independence*, an opportunity to send queries without specifying where the referenced relations are located, and *distributed transaction atomicity*, an opportunity to access and update data at several sites just as they would write transactions over purely local data. Depending on whether the data is distributed over the same DBMS software or not, we distinguish homogeneous and heterogeneous distributed database systems. In situations where these systems are heterogeneous, we have to establish gateway protocols.

In these systems, data is usually distributed by *fragmentation*, when fragments of data are divided over several sites. This fragmentation can be *horizontal*, when each fragment consists of a subset of rows of original relation, and *vertical*, when each fragment consists of a subset of columns of the original relation. Fragmentation is followed with replication of data and several copies of one fragment are stored in the system. We must keep track of how the relations are fragmented and replicated, which can be be complicated. The catalog of this track-keeping can be centralized, in which case the entire system would be vulnerable to failure of the site which contains the catalog, or also distributed, in which case the biggest issue is maintaining data consistency.

In [3], the authors have developed a multi-ring model based on Chord, in which each shared resource is described by one or more keywords. Nodes are organized in multiple keyword rings, and each node in a keyword ring contains the list of nodes that host resources matching a certain keyword/value pair. A new keyword ring is created only when the number of queries or registered resources for the keyword rises above a certain threshold. To enable keyword rings to be found, a Super Ring is used to host a ring of nodes which contain pointers to other rings. One major drawback of the model is that it heavily depends on the bootstrap node.

ML-Chord, presented in [6], is a multi-layered P2P resource sharing model which introduces overlay layers of categories. The number of these categories depends on the number of categories for a specific domain or ontology. Also, two types of nodes are introduced : *normal peers*, which can be associated with one or several layers and *bridge peers*, which are peers with better capabilities, linked to all categories and themselves form a category as well. The problem with this approach is that it is not possible to simply encapsulate a new system into an existing one because all of the Chord layers share the same hash function. Although this system is scalable and efficient, it is not possible to easily introduce a new category during the system lifetime. The developers suggest that one node should be linked to only one layer for better performance. So, if a node with good capabilities has not become a bridge peer at the start of the lifecycle of the system, it will remain a normal node, and its beneficial capabilities will be lost.

2.2 Interconnecting Different Overlay Networks

As said in the introduction, co-operation of different ISs implemented via overlay networks is a challenging problem which make pragmatical benefit in the context of this paper, namely catalog digitization. In the context of large scale information retrieval, several overlays may want to offer an aggregation of their information/data to their potential common users without losing control of it. One may perceive that having a single global overlay has many obvious advantages and is the *de facto* most natural solution: unfortunately it appears unrealistic in many real cases. In some optimistic case, different overlays are suitable for co-operation by opening their proprietary protocols in order to build an open standard; in many other pessimistic cases, this opening is simply unrealistic for many different reasons (backward compatibility, security, commercial, practical, etc.).

The catalog we present in this paper is based on *Synapse* [8], a scalable protocol for information retrieval over the inter-connection of heterogeneous overlay networks. The protocol is based on co-located nodes, also called *synapses*, serving as low-cost natural candidates for inter-overlay bridges. In the simplest case, where overlays to be interconnected are ready to adapt their protocols to the requirements of interconnection, every message received by a co-located node can be forwarded to other overlays that node belongs to. In other words, upon receipt of a search query, in addition to its forwarding to the next hop in the current overlay, according to their routing policy, the node can possibly start a new search, according to some given strategy, in some or all other overlay networks it belongs to. This obviously implies that a Time-To-Live value has to be provided and detection of already processed queries implemented, so as to avoid infinite looping in the networks, as it is the case in unstructured peer-to-peer systems. Applications of top of Synapse see those inter-overlays as a unique overlay.

Experiments and simulations, we run, showed that a small number of well-connected synapses is sufficient in order to achieve almost exhaustive searches in a "synapsed" network of structured overlay networks. We believe that Synapse can give an answer to circumventing network partitions; the key points being that:

- several logical links for one node lead to many alternative physical routes through these overlay, and
- a synapse can retrieve keys from overlays that it does not even know, simply by forwarding the query to another synapse that, in turn, is better connected.

For more details on the Synapse protocol, see [8].

2.3 Contribution

Our intention is not to give a final solution for a new concept of DBMS, but to show an idea that it is possible to establish a DBMS with desirable properties, which can connect heterogenous DHTs in a homogenous way and which can be easily expanded, without real fragmentation of data, while keeping the property of load balance and good performance of a whole system. Also, we will show that this system is applicable to real-life situations.

3 Application Principles

One of the main features of a distributed catalog is to assist researchers and members of the wider community in retrieving information concerning some fact of interest, information which can be provided from different kinds of sources. As mentioned before, digitized documents, by their nature, are highly distributed resources. By connecting different kinds of data providers into one system, the quality of the resulting information can be increased.

In this paper, we consider a distributed catalog which contains only meta-data on digital documents which follows a part of the Recommendation for the meta-data format for describing digitized heritage, described in [7]. One of the main reasons for this is the intellectual property rights issue. Simply, some institutions do not wish to outsource control over their digital repositories, and, instead, choose only to publish the information that they are in possession of a certain document. The digital documents themselves can be retrieved with one of the meta-data fields which contains information on their actual remote location.

A user can connect to one or more communities which he is member of (i. e. he has been invited to or his request has been accepted). Two operations are then available, namely: (i) storing a new record and (ii) finding a record which contains some information.

Suppose we wish to store the following information on a digital object:

```
<digitalObject>
  <title>Title</title>
  <creator>Name</creator>
  <location>link</location>
  <relatedAsset>
     Realife object</relatedAsset>
  <note>
    <src lang ="
     language of the value">value</src>
  </note>
  <archivalDate>date</archivalDate>
  <mimeFormat>mime type</mimeFormat>
  <digitalObjectOwner>
     Owner</digitalObjectOwner>
</digitalObject>
```

XML metadata record

Table 1. Different data structures stored in the distributed catalog DHT for each entry

No.	Key	Value
1	$title\sharp Title$	$hash(\maltese)$
2	$creator\sharp Name$	$hash(\maltese)$
3	$relatedAsset\sharp realifeObject$	$hash(\maltese)$
4	$mimeFormat\sharp mimeType$	$hash(\maltese)$
5	$digitalObjectOwner\sharp Owner$	$hash(\maltese)$
6	$hash(\maltese)$	$\maltese$

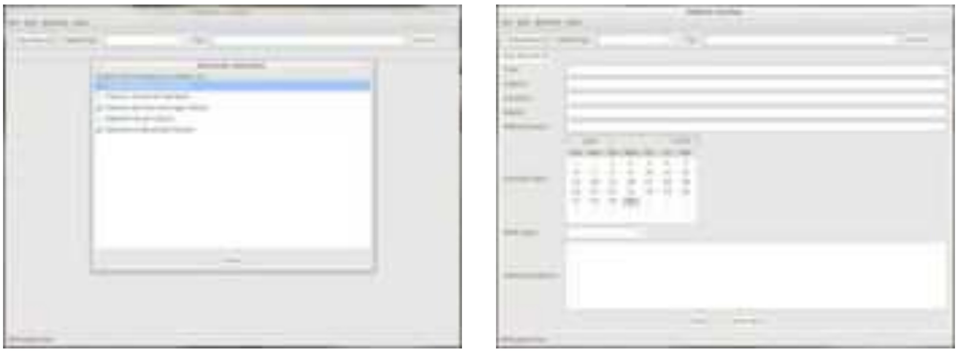
where $\maltese$ represents the full meta-data record on one digital document

If we were to decide to make the catalog searchable for the values in the fields: *title*, *creator*, *relatedAsset*, *mimeFormat* and *digitalObjectOwner*, then we would store segments in accordance with Table 1.

More precisely:

1. For every field of a meta-data record which we choose to be searchable, the hashed value for the current overlay of the entire meta-data record as value with the key which contains information about the field and its value is stored (rows 1 to 5 in table 1).
2. The entire meta-data record as a value with the corresponding key that contains its hashed value for the current overlay is stored (row 6 in table 1).

Note that all of the keys are stored with their hashed values. With this in place, the search mechanism has two phases. During the first phase, we attempt to find the hashed value



(a) Connecting to an overlay (b) Record insertion form

Fig. 1. Connecting and Inserting

of the meta-data record (the first kind of entries) and then, during the second phase, to find the entire meta-data record (the second kind of entries) only in the overlays which contain the first kind of entries. Although we have multiple copies of data, so as to accomplish failure resistance of the system, the storage space is of the same complexity as for a standard DBMS with indices. If N and M are the number of overlays and the number of nodes per overlay, respectively, then the time complexity of a search, in the worst case, is $O((N+1) * (\textit{time to search an overlay with M nodes}))$.

4 Case Study

Institutions which are interested in sharing meta-data information on their digital documents can be connected in different overlays by their nature. So, all archives may be a part of one overlay, all libraries of the other, and similarly with museums, research centers, universities, etc. These overlays can be connected by institutions which contain various kind of content, like research centers with important libraries or research centers which are part of the universities, etc. All of these institutions would run the same application.

The following proof-of-concept is a simple application to store and retrieve records from one or multiple overlays. It offers the following three functionalities, arranged in a Graphical User Interface developed in Java, for cross-platform compatibility:

- **Joining** of a new network,
- **Storing** of a new record,
- **Searching** for records.

The application is designed using a tabbed organization of different forms. This is to allow the user to easily perform multiple operations at the same time (*e.g.* doing multiple queries and comparing the results). Furthermore, it constitutes a familiar usage environment, resembling, in the approach, most of modern Internet browsers (multiple tabs, address/search on a top bar). Basic editing features, like saving and loading a record to and from an XML file, copying/pasting and printing the XML raw data, are provided.

(a) Search results for a query

(b) Details of a retrieved record

Fig. 2. Searching and Retrieving

Network join. As shown in Figure 1a, upon starting, the program will propose to the user a list of known DHTs to connect to. These represent existing overlays put in place using the same system, which are, therefore, compatible with our software. It is important to notice that, after having connected to a first overlay, a user can choose to further join other available networks. This can be done via the menu entry Network → Join, which will propose the same dialog box as in Figure 1a.

Once being a member of multiple overlays, not only it becomes possible to query all of the overlays simultaneously, but, thanks to the capabilities of the synapse protocol described in Section 2.2, it will also be possible to act as a relay, replicating requests from one overlay to another.

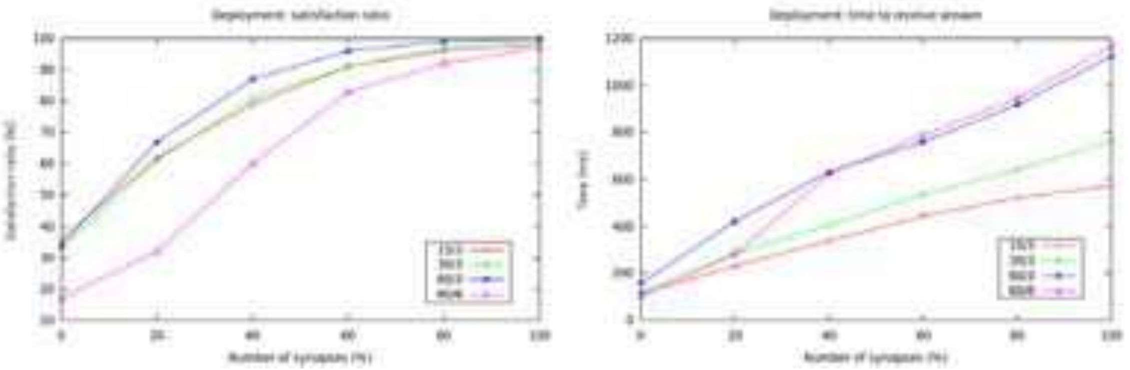
Storing a new record. Figure 1b shows the insertion form for a new record in the DHT.

In this catalog we store the records which follow a part of the mentioned recommendation of the meta-data format:

- **Title** of the digital document (*i.e.* electronic book)
- **Name of the author** who made electronic version
- **Link** of the remote location of the digital document
- **Related object** (*i.e.* hard copy book)
- **Note** or a short description
- **Date** when electronic copy was made
- **Mime type** (*i.e.* pdf)
- **Owner** of the digital object

While some of the fields may be optional, the ones used as search criteria have to be filled before the record can be saved. Therefore, the “Save” button remains disabled until all of the appropriate text boxes are filled.

Record searching. Looking for a record takes place in a way resembling the behavior of most modern Internet browsers: as one can see in Figure 2a, the search type and field are in the upper toolbar. Here the user can choose the type of search to perform (title, author, owner, related object, mime type) and fill the search key. By pressing the “Search” button, a query for the corresponding key is performed in the overlay (or overlays, if synapses are present or the software is connected to multiple networks). A



(a) Exhaustiveness

(b) Latency

Fig. 3. Deploying Synapse

result summary is displayed in a new tab once the query is over, containing the number of records found and a table with all the records.

To display the details of a record the user can double-click on the corresponding row in the table. This (showed in Figure 2b) will open a new tab containing record details.

The details tab is similar to the new record form, except that the text fields cannot be edited (although it is still possible to select and copy the text inside).

The button "Display raw XML" will open a new dialog showing the actual XML data.

5 Experiments

In order to test our inter-overlay protocol, as a ground base of our catalog, we have developed open-synapse software. It is based on the open-chord v. 1.0.5 implementation, developed by Distributed and Mobile Systems Group Lehrstuhl fuer Praktische Informatik Universitaet Bamberg, which is, essentialy, a Java implementation of the Chord protocol. This platform fully implements a Chord-based inter-overlay network, but to achieve the goal of connecting heterogenous overlay networks, we have decided that in our implementation every Chord ring has its own hash function. The experiments were realized on an IBM Beowulf Cluster 1350. During the tests we have started 5 logical nodes at each of up to 12 working nodes. The nodes have been uniformly dispatched over 3 or 6 overlays, and during deployment, the overlays were progressively bridged by synapses (the degree of each logical node was never greater than 2). So, we did tests with 15, 30 and 60 nodes uniformly dispatched over 3 overlays and 60 nodes uniformly dispatched over 6 overlays.

Figure (3a) shows the satisfaction ratio when increasing the number of synapses. By the satisfaction ratio we mean the percentige of the succesfull answers for the vaules that are, already, inserted into the system. It can be seen that a quasi-exhaustiveness has been achieved when the synapses are members of only 2 overlays. Note that the satisfaction ratio confirms the simulation results, which are available in [8].

Figure (3b) shows the average elapsed time from the moment a query is sent to some node in the system until the moment when the answer is received. This time was not so short, but this is due to the configuration of the cluster. We suppose that these results could be illustration of the performance in a real-life situation, in which the nodes would not be members of networks which are within the same infrastructure.

6 Conclusion

In this paper we have shown that the Synapse protocol has good potential as a new concept of DBMS. With this concept, it is possible to connect heterogenous DHTs in a homogenous way. We have proven the scalability of this protocol and its applicability to a real-life situation.

Since we cannot guarantee full exhaustiveness of information retrieval, we have decided that the procedure of removing/updating items should currently be out of scope of our research. The reason for this is that the only one who may remove or update items inside the catalog should be the one who inserted them in the first place, thus guaranteeing the highest probability of data consistency. For this, we would first need to implement a User Management System, for instance, by implementing cryptographic technics into our system, as described in [4].

As mentioned before, within this system we can also store the digital documents themselves. We have also decided that in the current phase, this should be out of scope of this paper, but we consider this to be a possible continuation of our research.

As a positive side-effect, we believe that our catalog can lay promising groundwork for a low-cost solution to cultural interconnection of the institutions inside the region.

Acknowledgment. The authors warmly thank Petar Maksimović and Cédric Tedeschi for their precious suggestions during the writing of this paper.

References

1. Ramakrishnan, R., Gehrke, J.: Database Management Systems. McGraw Hill, New York (2000); ISBN: 0-07-246535-2
2. Stoica, I., Morris, R., Karger, D., Kaashoek, M., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup service for Internet Applications. In: ACM SIGCOMM, pp. 149–160 (2001)
3. Antonopoulos, N., Salter, J., Peel, R.: A multi-ring method for efficient multi- dimensional data lookup in p2p networks. In: Proceedings of the 1st International Conference on Scalable Information Systems (2006)
4. Avramidis, A., Kotzanikolaou, P., Douligeris, C.: Chord-PKI: Embedding a Public Key Infrastructure into the Chord Overlay Network Public Key Infrastructure. Springer, Heidelberg (2007); ISBN: 978-3-540-73407-9
5. Liquori, L., Tedeschi, C., Bongiovanni, F.: BabelChord: a Social Tower of DHT-Based Overlay Networks. In: 14th Symposium on Computers and Communications (ISCC 2009). IEEE, Los Alamitos (2009) (short paper)
6. Lu, E.J.-L., Huang, Y.-F., Lu, S.-C.: ML-Chord: A multi-layered P2P resource sharing model. Journal of Network and Computer Applications 32, 578–588 (2009)
7. Ognjanović, Z., Butigan-Vučaj, T., Marinković, B.: NCD Recommendation for the National Standard for Describing Digitized Heritage in Serbia. In: Metadata and Semantics, p. 978. Springer, Heidelberg (2009); ISBN: 978-0-387-77744-3

8. Liquori, L., Tedeschi, C., Vanni, L., Bongiovanni, F., Ciancaglini, V., Marinković, B.: Synapse: A Scalable Protocol for Interconnecting Heterogeneous Overlay Networks. Networking (2010)
9. The Dublin Core Metadata Initiative, http://dublincore.org/
10. Encoded Archival Description, http://www.loc.gov/ead/
11. IFLA Study Group, Functional Requirements for Bibliographic Records (1998), http://www.ifla.org/VII/s13/frbr/frbr.pdf
12. MARC Standards, http://www.loc.gov/marc/
13. Recommendations for coordination of digitization of cultural heritage in South-Eastern Europe, Conclusions of the Regional Meeting on Digitization of Cultural Heritage, Ohrid, Macedonia, March 17-20 (2005); Review of the National Center for Digitization, 2-7 (2005), http://elib.mi.sanu.ac.rs/files/journals/ncd/7/ncd07002.pdf
14. A weblog about FRBR: Functional Requirements for Bibliographic Records (2007), http://www.frbr.org/
15. The European Library, http://www.theeuropeanlibrary.org/
16. Europeana, http://europeana.eu/portal/