A methodology for binary encoding of citation metadata

Marios Poulos*

Department of Archives and Library Sciences,
Ionian University,
Palea Anaktora 49100, Corfu, Greece
Email: mpoulos@ionio.gr
*Corresponding author

Nikolaos Korfiatis

Department of Computer Science,
Institute of Informatics and Mathematics,
Goethe University Frankfurt,
Robert-Mayer-Str. 10, D-60325,
Frankfurt am Main, Germany
Email: korfiatis@em.uni-frankfurt.de

George Bokos

Department of Archives and Library Sciences,
Ionian University,
Palea Anaktora 49100, Corfu, Greece
Email: gbokos@ionio.gr

Abstract: In this paper we describe a methodology for binary encoding of citation metadata which targets the problems of versioning generated from the function of the web agents and/or crawlers on which Google Scholar and Citeseer systems operate. Typically citation indexes require a period of time in order to index the cited articles and update their counters, which is a contingency to allow processing of information. Furthermore, the storage of this information is based on text form, which requires large storage space. The proposed method is an online system based on the existence of a central relational database (in binary format) and the existence of two automated procedures, which carry out the acts of publishing and citation. In particular a detached authority provides a database which stores in the proposed binary coded format all the citations that a new article makes at the time that it is published. The communication between the database and the users is enabled by the use of XML files which can be stored in memory, making computation easier.

Keywords: citation; database; binary format; XML; metadata.

Reference to this paper should be made as follows: Poulos, M., Korfiatis, N. and Bokos, G. (2012) 'A methodology for binary encoding of citation metadata', *Int. J. Metadata, Semantics and Ontologies*, Vol. 7, No. 4, pp.295–301.

Biographical notes: Marios Poulos is an Assistant Professor at the Department of Archives and Library Sciences at the Ionian University in Corfu, Greece. He received his PhD degree from the University of Piraeus in the field of biological information processing. His research interests are in metadata and semantics research with an emphasis on intelligent methods for the extraction of semantic information from various sources.

Nikolaos Korfiatis is a senior researcher (post.doc.) at the group Databases and Information Systems (DBIS) at the Department of Computer Science at Goethe University Frankfurt. He received his MSc in Systems Engineering in 2006 from the Royal Institute of Technology (KTH), Stockholm Sweden and his PhD in 2009 from Copenhagen Business School (CBS) Denmark. His research interests are in the area of information management with particular emphasis on opinion mining and data analytics using novel data management technologies such as Big Data/NoSQL approaches.

George Bokos is Professor Emeritus of Information Technologies in the Ionian University, Corfu, Greece. He was Former Head of the Department of Archives & Library Studies of the University, the founder and Director of the Information Technologies Laboratory and for many years Director of the Department's Postgraduate studies. He had been member of the staff and Director of the National Library of Greece. As a member of both the staff of the National Library of Greece and of the teaching staff of the Ionian University he has participated in many research

activities and projects, in most of them as coordinator. As member of several national and international committees and working groups he has also published numerous books and papers in his main field of interest.

1 Introduction

Citation indexes are of particular importance for researchers and librarians due to the hyperlink based structure they provide which allows for reference tracing across scientific fields as well as scientometric evaluation (Garfield, 2006a). While initially citation indexes were designed for information retrieval purposes, their usage in literature search, analysis of academic literature, and evaluation has turned them into very important tools for research assessments (Harnad, 2009). By definition citations are used in scholarly works to give credit to or to acknowledge the influence of previous works or to refer to authority and provide a link between the current work and the cited articles. The practice of providing citations to previous work is a *de jure* standard of scientific communication and the provision of citations for arguments is required even in practices and industries outside the academia such as for example the patent application system. In the latter is argued that the citation of previous work clearly makes a patent application credible and its useful on the study of how engineering knowledge spreads and develops (Jaffe et al., 1993).

The development of citation indexes is a direct result of the practice and development of scientometrics as a quantitative assessment of an individual's or an organisations research profile (Hood and Wilson, 2001). Furthermore the infamous impact factor (IF) is a direct result of the development of a citation index database. The first non-library based citation index was created by the Institute for Scientific Information (ISI), with the first citation database bearing the name Web of Science (WoS) and relied on the theory of Eugene Garfield about the science citation index (Garfield, 2006b).

The WoS database has data from the year 1945. The publishing house Elsevier created its own citation database system in 1994, which is named Scopus and keeps data from the year 1966. After that, in late 2004 the beta version of Google Scholar was introduced mainly targeting preprints and freely available material that was hosted on author's websites in various universities. While Web of Science and Scopus operate with human assistance Google Scholar uses a crawler especially designed to index and tokenise reference based information and perform a citation match. Another fact that assisted the development of Google Scholar can be found on the practice of mainly computer scientists and mathematicians on maintaining BibTex based bibliographies for referencing using the LaTex system for scientific paper editing (Knuth, 1984; Patashnik, 1988). While the automated nature of Google Scholar allows for vast scaling and development of a considerably large citation index in a very short time there are arguments against its use due to the very low metadata quality. A study by Jacsó (2005a) compared the three systems (WoS, Scopus and Google Scholar) in 2005 and found that Google Scholar has a considerable lack of citation medatata quality in comparison with the two other systems. Although Google Scholar does not go into depth about the content of its database, its usefulness as a citation index relies on the fact that is easily accessible from the web and highly responsive (Jacsó, 2005b). Furthermore, for some areas conference proceedings play an important role, something that both WoS and Scopus tend to not index.

An earlier approach from Google Scholar was the CiteSeer citation index (Bollacker et al., 1998) which was also focused on the practice of mining BibTex files as a source of indexing. The main user interaction pattern behind CiteSeer relies using keywords that it gets from the user and starts to locate and download papers in postscript format. Afterwards, it parses the downloaded papers and extracts the abstract, the title, the authors and citations. These citations are followed by crawlers in order to find semantically relative papers to download. This cyclic procedure occurs repeatedly. The biggest disadvantage of Citeseer is that it starts to find papers only after a user requests them and the target papers are in postscript format, while its advantage compared to the three above systems is that these papers might be *invisible* to them or not be indexed yet. On the other hand, in Google Scholar the attention is shifted on incidental problems which are sourced in the distortion of the citation metrics targeted to the individual assessment such as the h-index (Harzing and van der Wai, 2008; Franceschini and Maisano, 2011).

The common characteristic of the current web crawler based citation indexes is the time delay from the time point that a new work is published until the time point the appropriate records are inserted into the citation index. This issue has two negative aspects: (a) citation metadata are not provided in a complete way and (b) with the continuous production of scientific literature and increased popularity by researchers of all fields it becomes costly (in terms of computation and infrastructure) to offer this service for free to the end users.

Furthermore, we consider the following problems for an online automated citation index:

a *The problem of versioning*: While citation metadata are structured in a considerable standardised way, the requirements of a specific citation detail vary based on the citation style that a researcher will use. The later tends to create various versioning problems which are evident on Google Scholar in particular.

b *The cost of retrieval*: Due to the different versions that might arise from (a), a unique identifier for a citation is not always used due to the fact that the nature of the identifier (e.g. DOI) points to the manuscript per se and not on the metadata.

Considering the above issues, we examine a scenario where citation metadata are encoded into a binary format using a proposed procedure. The binary encoding of citation metadata aims on the one hand to provide complete resource identification in order to address versioning issues and on the other hand to tackle with the computational cost of citation retrieval. The proposed method considers an online system using a central relational database and two automated procedures that perform the acts of publishing (storage) and citation provision (retrieval). We invoke the Universal Citation Database (Cameron, 1997), which describes a universal oriented bibliographic and citation schema linking every scholarly work ever written – no matter how it is published – to every work that it cites and every work that cites it. This database would be comprehensive and up-to-date. The act of publishing pertains to the way in which a new article obtains a unique binary identification number, while the act of citation has to do with the way in which they are attached to the cited article.

The benefits derived from this novel method are that due to the binary format, data disk reads are faster and less memory is used and consequently parsing time and transmission bandwidth can be saved, as fewer bits are used than the bits used in the textual format. Thus, storage space is kept intact, which is very important since the Universal Citation Database specification demands size, speed, reliability and flexibility. Another benefit is that it can be used for dynamic citation counts, scientometric analysis and other research evaluation tasks which will be easier to compute due to the proposed encoding procedure.

To this end this paper is structured as follows: Section 2 underlines the methodology related with the creation of the binary reference format. Section 3 discusses an interoperability scenario using an XML – Database. The paper concludes on Section 4 with directions for future research and possible applications.

2 Method

As mentioned in the introduction, the coding of a new published work's citation metadata is encoded in a binary format and doesn't encapsulate the publication's content per se. Before this is analysed, a database schema definition is required in order to provide an input to the binary encoding process. The following entities and attributes are outlined below.

The common ancestor entity is the 'Publishing Authorities' which stores the Publishers' records. In a relational database management system (RDBMS) implementation it will have two required attributes. The first will be an identifier (autonumber and primary key) and the publication name (text).

The next entity is the 'Identification Serial Numbers', which has the 'Publishing Authorities' ancestor. Its fields will be the ID (autonumber and primary key), the Pub_ID (foreign key from the ancestor) and the Serial_Number (integer). The next entity is the 'Volume', which has the 'Identification Serial Numbers' ancestor. Its fields will be the ID (autonumber and primary key), the SerialNumber_ID (foreign key from the ancestor) and the Volume_Number (integer).

The subsequent entity is the 'Issue', which has the 'Volume' ancestor. Its fields will be the ID (autonumber and primary key), the Volume_ID (foreign key from the ancestor) and the Issue_Number (integer). The next entity is the 'ArticleOrder' which has the 'Issue' ancestor. Its fields will be the ID (autonumber and primary key), the Issue_ID (foreign key from the ancestor) and the Order_Number (integer), which states the order that the work has in the current issue. If the work is published only electronically then this number will be equal to one.

The next entity is the 'Article', which has the 'ArticleOrder' ancestor. Its fields will be the ID (boolean and primary key), the Order_ID (foreign key from the ancestor), the Title (text), the Authors (text), the Abstract (text) and the Body (text). The next and final entity is the 'References', which has the 'Article' ancestor. Its fields will be the Article_ID (foreign key from the ancestor), and the Cited_Article_ID (foreign key from the ancestor). The two fields together make the primary key. All these entities and the hierarchy are shown in Figure 1.

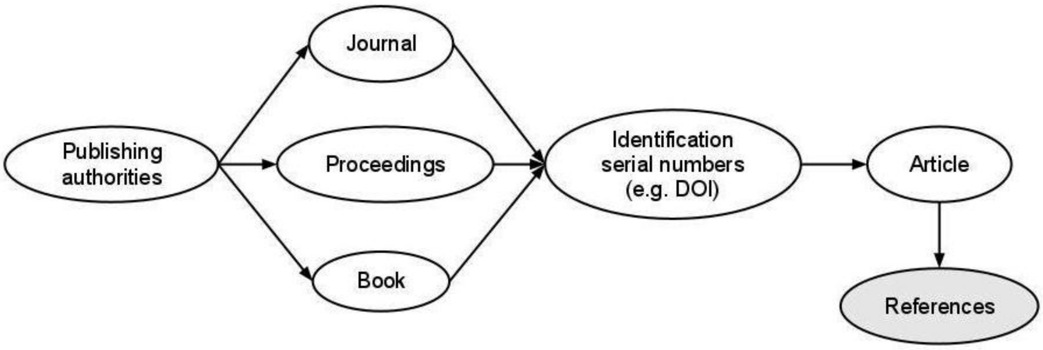
The database provider will independently provide the repository storage and the foundation for two separate procedures, namely: (a) the publication encoding procedure and (b) the citation retrieval procedure. We discuss these procedures on the sections that follow.

2.1 Publication encoding procedure

The first Publisher that will be stored in the appropriate table will automatically obtain a number k for the ID and the user must insert the name only into the other field. The next Publisher will obtain the number $k+1$ for ID and so on. The insertion of values in the table of identification Serial Numbers is done in the same way but here it must be clarified to which Publisher it refers by writing the Publisher's ID value in the Pub_ID field. The insertion of values in the table of Volume is done in the same way but here it must be defined to which Serial Number it refers by writing the Serial Number's ID value in the SerialNumber_ID field.

Subsequently the insertion of values in the table of Issue is done in the same way but here it must be described to which Volume refers by writing the Volume's ID value in the Volume_ID field. The insertion of values in the table of ArticleOrder is done in the same way denoting also to which Issue it refers by writing the Issue's ID value in the Issue_ID field.

Figure 1 The entities and the hierarchy on the database schema



If the new work–article was published by the Publishing Authority that has ID equal to *k*+1 then the first part of the Article's ID will be the binary sequence "01". If it was published by the Publishing Authority that has ID equal to 7 then the first part of the Article's ID will be the binary sequence "0000001". In this sequence, on the right is always 1 and on the left is (are) (ID-1) 0 bit(s).

In the previous examples, when ID=2, the zero is 2–1=1 and when ID=7, the zeros are 7–1=6. In the same way, the second part of the Article's ID is created depending on the value of the Identification Serial Number's ID. If this equals 10 then the second part will be "0000000001" and it will be added at the right of the first part. If the first part was "01" then the two parts together would be "010000000001". The third part of the Article's ID is created depending on the value of the Volume's ID. The fourth part of the Article's ID is created depending on the value of the Issue's ID. The fifth and final part of the Article's ID is created depending on the value of the ArticleOrder's ID. The number of "1" in the sequence must always be five, one per each part. The creation of the Article's ID is shown in Figure 2.

In essence the citation encoding is provided by traversing the automatic identifier definition of the RDBMs by making it easier to compute as the last node on a sequence of indexes. This binary encoding can be done automatically by a stored procedure of the database. Then the Article should be inserted into the "Article" table. The ID field takes its value from the stored procedure's output and the other complementary fields may also be inserted by the user.

Having described the codification procedure we proceed on providing the citation retrieval procedure which is based on the reverse path of the codification graph discussed above.

2.2 Citation procedure

The citation procedure considers the problem of assigning an interlink which is derived from the publication codification procedure. From the new article's reference page, records will be inserted into the "References" table. All the cited articles are already inserted in the database and each one has a unique ID, which is the binary sequence of the "Article" table. Figure 3 illustrates the citation provision procedure.

When an article is cited for the first time then a "path" is created, where the main path is the article's ID and the child path is the ID of the article that cites it. For example, the new article with ID = "01001110001" cites the old article with ID = "0001000100110001" for the first time. The path of the old article is [[0001000100110001]]–[01001110001].

In a later time point a new article with ID = "000010010010100001" cites the same old article. Then the train of the old article is [[0001000100110001]]–[01001110001] – [000010010010100001] and so on.

It follows that a user can derive the information about the time sequence of citations that a specific article has. The last part of the path is always the latest citation for the article of the first part. The path creation can also be automatically addressed by a stored procedure in the database. This can also insert the appropriate values into the "References" table.

Figure 2 The publication procedure: Article's ID generation (see online version for colours)

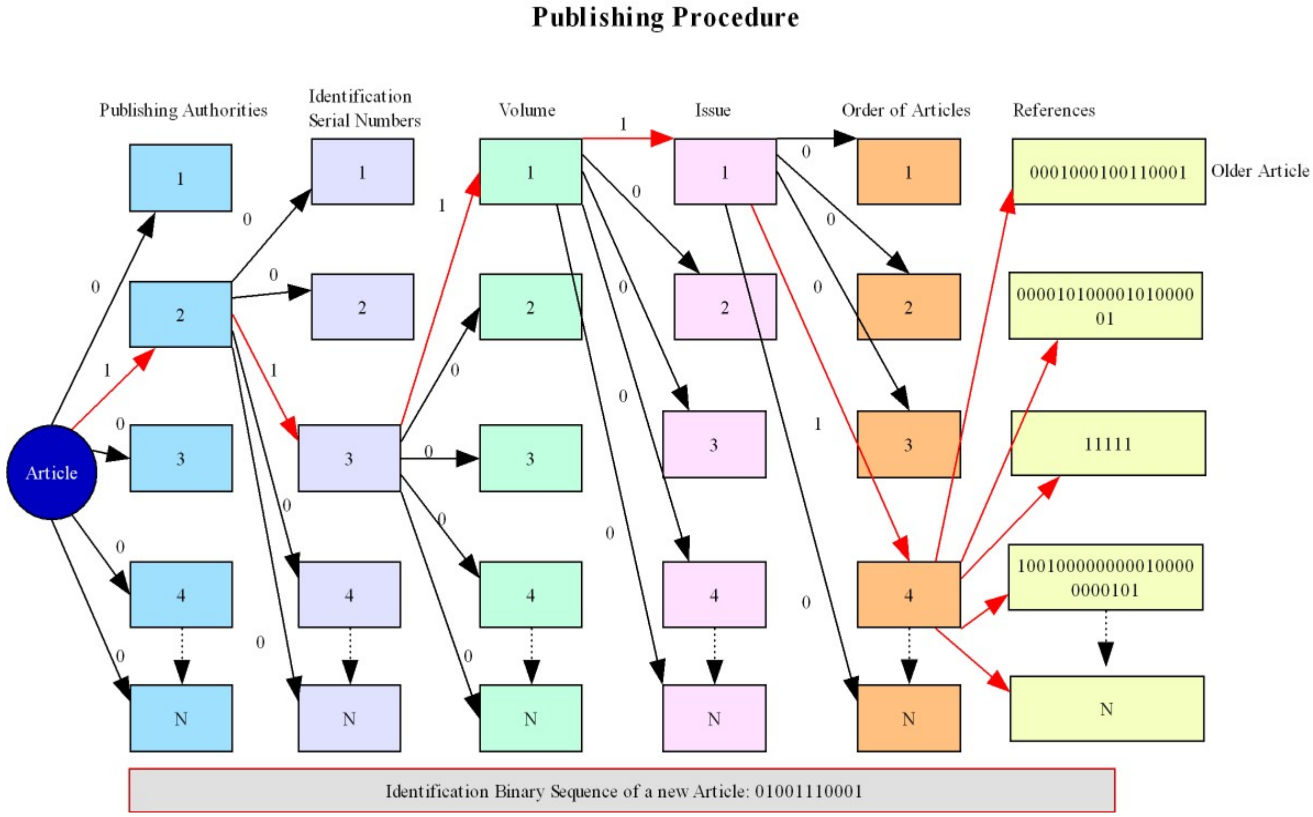
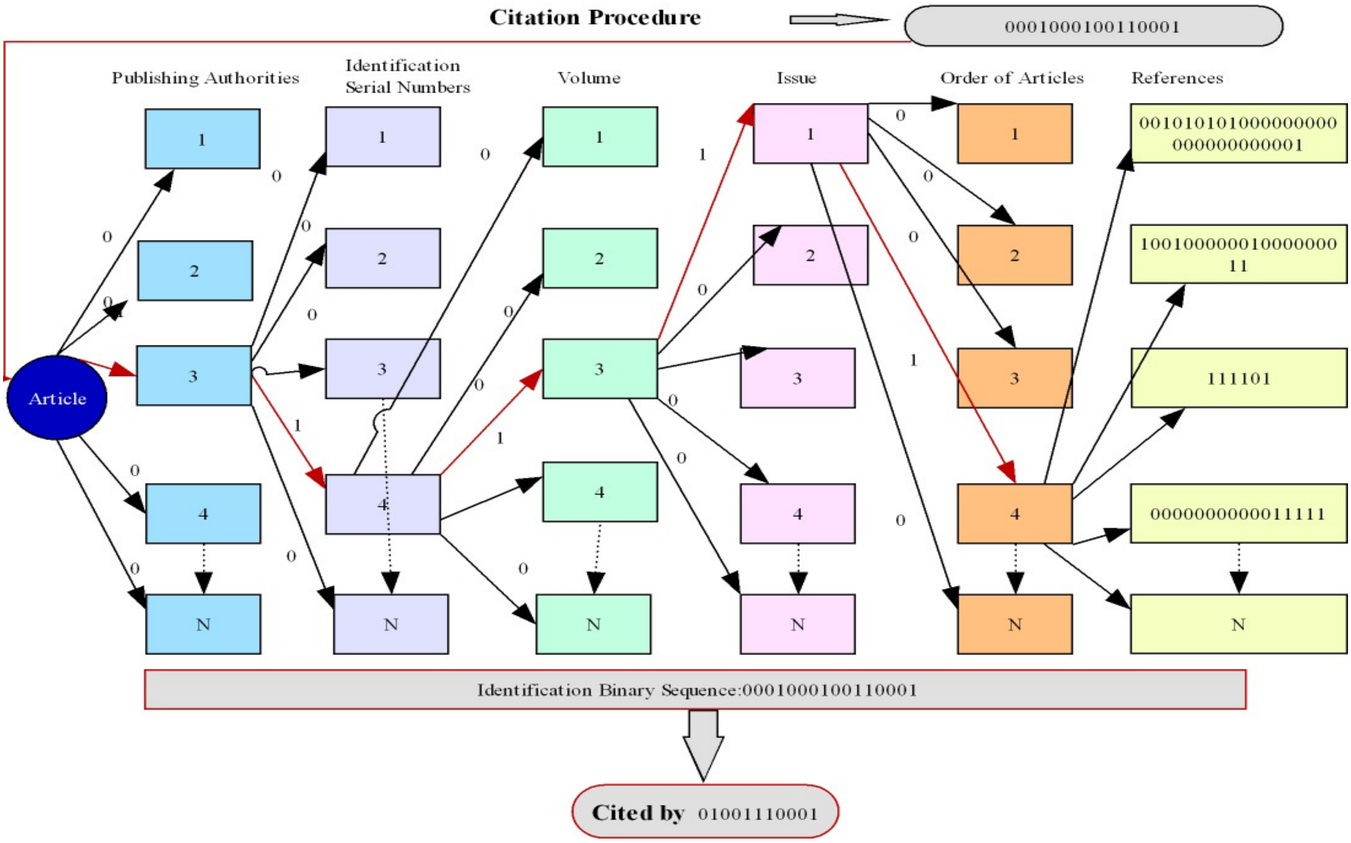


Figure 3 The citation provision procedure (see online version for colours)

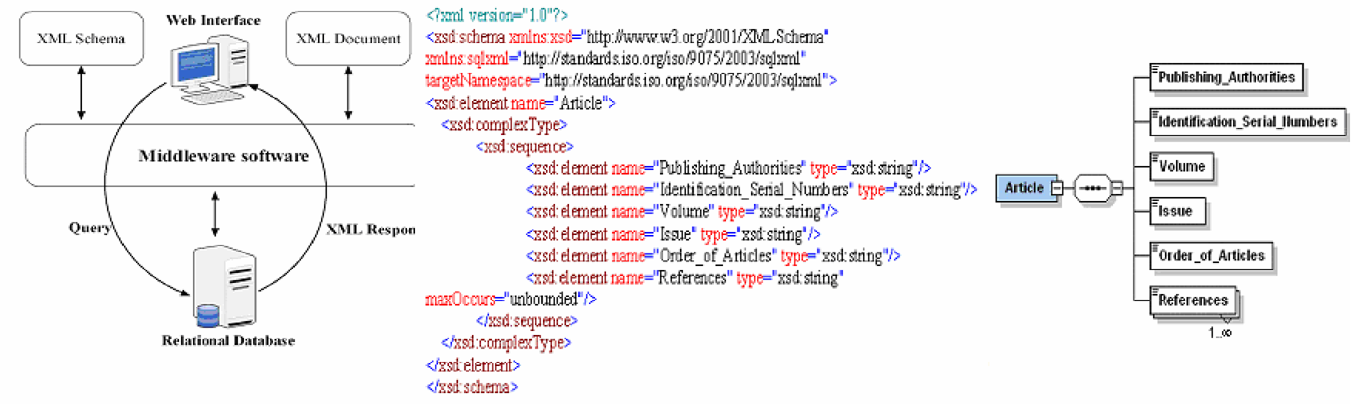


This citation retrieval process has a major advantage on relying on the RDBMs' ID assignment without requiring a re-computation of the identifier part but solely querying the identifiers of each entity which only corresponds to a unique path making the computation time minimal and scalable.

3 Interoperability scenario XML – database

Having described the binary codification and retrieval of the citation metadata we continue on an example application scenario using an xml response as a translator between the binary citation format and the user. Conceptually at the higher level, there is a single user web interface environment for interaction and querying that allows the user to access the bibliographic records and their citations and make them available for searching, retrieval and analysis. Via the web user interface authors are able to register their new articles online with the database. When a query to the database is submitted by a user, then the database performs the automated stored procedure of the citations and sends an XML Response via the web browser to the user (see Figure 4).

Figure 4 Representation of the architecture which is based on the proposed XML schema (see online version for colours)



```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:sqlxml="http://standards.iso.org/iso/9075/2003/sqlxml"
targetNamespace="http://standards.iso.org/iso/9075/2003/sqlxml">
<xsd:element name="Article">
  <xsd:complexType>
        <xsd:sequence>
                <xsd:element name="Publishing_Authorities" type="xsd:string"/>
                <xsd:element name="Identification_Serial_Numbers" type="xsd:string"/>
                <xsd:element name="Volume" type="xsd:string"/>
                <xsd:element name="Issue" type="xsd:string"/>
                <xsd:element name="Order_of_Articles" type="xsd:string"/>
                <xsd:element name="References" type="xsd:string"
maxOccurs="unbounded"/>
        </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

An XML schema plays a considerable role in the overall XML architecture. It is a description of a type of XML document and it may be used to verify the integrity of the content, in that case the citation metadata. In particular the XML schema (see Figure 4) that is used has a root element which is the Article and thereafter the following elements in sequence: (a) Publishing Authority (b) Identification Serial Number, (c) Volume, (d) Issue, (e) Order of Articles and finally the (F) References. The elements in essence are derived from the publication metadata encoding procedure.

As can be seen in Figure 4 the application scenario considers the following:

a A user access a web interface of the relational database by using a browser or a REST type of request (Khare and Taylor, 2004) using the HTTP/GET protocol

b A middleware software handles the communication between the user and the database and depending on the HTTP request parameters an xml file is formatted

c If the request is a publication addition then the parameters are encoded in an xml file and the store procedure is invoked.

d If the request is about citation retrieval, the xml file containing the translation of the binary format to the actual metadata is provided.

The benefit of using XML files as a middleware between the user and the RDBMS is that it allows the files to remain on the server's memory which makes computation faster such as for example citation counts. The later can be achieved by using a binary memory translator which also allows for resource optimisation (Xu et al., 2007).

While this interaction procedure is not fully automated from the part of the indexing as it can be in the case of a web crawler from the perspective of gathering publication metadata, it can be applied in several social citation indexing scenarios such as in the case of user generated citation indexes (e.g. Citeulike).

4 Conclusions and future development directions

In this paper a method was described aiming to address the problems of versioning and retrieval for online and user based citation indexes. In the automated citation indexes such as Google Scholar a period of time is required in order for the index to update the cited articles and recomputed their citation counters, which is a '*just-in-case*' processing of information. Furthermore, the storage of this information is based on text form, which requires a large storage space while the proposed binary encoding is translated to the user only on request.

The proposed method requires the deployment of an online system based on the existence of a central relational database and the existence of two automated procedures, which carry out the acts of publishing and citation. In particular, there will be a detached authority where a database will be created. This database will store in binary coded format all the citations that a new article makes at the time that it is published.

However, the implementation of this method presupposes two conditions. The first is that the system must be tested in an experimental environment, for example an appropriately constructed database of a library. The second is the construction of a DTD schema for inference on data views such us search within cited articles (Papakonstantinou and Vianu, 2000). Furthermore, the implementation of this method opens up new possibilities in the computation of scientometric indexes, for which new methods may be developed utilising the possibility of application of Boolean algebra, as the system stores all the data in binary format. Finally, a definite advantage of this method is the way in which the binary storage exploits the non-lost compress possibility of the run length encoding.

References

Bollacker, K.D., Lawrence, S. and Giles, C.L. (1998) 'CiteSeer: an autonous Web agent for automatic retrieval and identification of interesting publications', *Proceedings of the 2nd International Conference on Autonomous Agents*, ACM, pp.116–123.

Cameron, R.D. (1997) 'A universal citation database as a catalyst for reform in scholarly communication', *First Monday*, Vol. 2, No. 4.

Franceschini, F. and Maisano, D. (2011) 'Bibliometric positioning of scientific manufacturing journals: a comparative analysis', *Scientometrics*, Vol. 86, pp.463–485.

Garfield, E. (2006a) 'Citation indexes for science. A new dimension in documentation through association of ideas', *International Journal of Epidemiology*, Vol. 35, No. 5, p.1123.

Garfield, E. (2006b) 'The history and meaning of the journal impact factor', *JAMA: The Journal of the American Medical Association*, Vol. 295, No. 1, p.90.

Harzing, A.W.K. and van der Wai, R. (2008) 'Google Scholar as a new source for citation analysis', *Ethics in Science and Environmental Politics(ESEP)*, Vol. 8, pp.61–73.

Harnad, S. (2009) 'Open access scientometrics and the UK Research Assessment Exercise', *Scientometrics*, Vol. 79, No. 1, pp.147–156.

Hood, W.W. and Wilson, C.S. (2001) 'The literature of bibliometrics, scientometrics, and informetrics', *Scientometrics*, Vol. 52, No. 2, pp.291–314.

Jacsó, P. (2005a) 'As we may search–Comparison of major features of the Web of Science, Scopus, and Google Scholar citation-based and citation-enhanced databases', *Current Science*, Vol. 89, No. 9, pp.1537–1547.

Jacsó, P. (2005b) 'Google Scholar: the pros and the cons', *Online Information Review*, Vol. 29, No. 2, pp.208–214.

Jaffe, A.B., Trajtenberg, M. and Henderson, R. (1993) 'Geographic localization of knowledge spillovers as evidenced by patent citations', *The Quarterly Journal of Economics*, Vol. 108, No. 3, p.577.

Khare, R. and Taylor, R.N. (2004) 'Extending the representational state transfer (rest) architectural style for decentralized systems', *Proceedings of the 26th International Conference on Software Engineering*, IEEE Computer Society, pp.428–437.

Knuth, D.E. (1984) *The texbook*, Reading, Mass.

Papakonstantinou, Y. and Vianu, V. (2000) 'DTD inference for views of XML data', *Proceedings of the 9th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ACM, pp.35–46.

Patashnik, O. (1988) *Designing BIBTEX Styles*, Citeseer.

Xu, C., Li, J., Bao, T., Wang, Y. and Huang, B. (2007) 'Metadata driven memory optimizations in dynamic binary translator', *Proceedings of the 3rd International Conference on Virtual Execution Environments*, ACM, pp.148–157.