A Service-Oriented Infrastructure for Early Citation Management

José H. Canós[†], Manuel Llavador[†], Eduardo Mena[‡,] Marcos R. S. Borges[¥]

[†]Dept. of Computer Science (DSIC), Technical University of Valencia,
Camino de Vera s/n, E46022, Valencia, Spain

[‡]Dept. of Computer Science and System Engineering, University of Zaragoza,
María de Luna 1, E50018 Zaragoza, Spain

[¥]Graduate Program in Informatics
NCE & IM, Federal University of Rio de Janeiro, Brazil

{jhcanos|mllavador}@dsic.upv.es, emena@unizar.es, mborges@nce.ufrj.br

Abstract. Citation analysis needs an in-depth transformation. Current systems have been long criticized due to defects such as lack of coverage and low accuracy of the citation data. Surprisingly, incorrect or incomplete data are used to make important decisions about researchers' careers. We argue that a new approach based on the collection of citation data when they are actually generated (that is, during the edition of papers) can overcome current limitations, and propose a new framework in which the research community as a whole is the owner as well as beneficiary of a Global Citation Registry characterized by high quality citation data. The registry will be accessible for all the interested parties and will be the source over which the different impact models can be applied.

Keywords: Citation management, Service-Oriented Architecture.

1 Introduction

The impact of publications is used by the research community worldwide as the main indicator of scientific quality. Many grants, tenures and employments in the academic world require candidates with excellent research records including a number of "high impact" papers. Such impact-based evaluation has had remarkable side effects; for instance, the phenomenon known as *publish or perish*[1] has produced in the last decade a shift in researchers' work towards productivity (measured strictly in quantitative terms) that, in turn, has made the number of conferences and journals grow substantially.

[1] see e.g., http://www.physicstoday.org/vol-57/iss-3/p61.html

On the other hand, the relevance of impact measures based on citation counts has made citation analysis become a research topic by itself, leading research groups, companies and associations to the development of models, algorithms and tools to deal with the main asset, which is the huge amount of citation data available.

Citation data are acquired in different ways. Manual approaches use human assistance to build citation records, whereas the automatic ones build citation records on the results of automatic harvesting processes. The former are very expensive, which leads to select some citation sources to the detriment of others, whereas the latter are error-prone and thus less reliable. Despite the unsatisfactory performance of current systems, the increasing number of publications and the cost of manual approaches irremediably drives to automated citation data collection. Hence, there is a clear need for automated solutions that overcome the limitations of current tools and enhance, if possible, their benefits.

A common pitfall of both manual and automatic approaches is that *citation data are collected after a work has been published*, loosing relevant citation information that is produced during the edition of papers. This is, in our opinion, the origin of the problems of current systems. In this paper we show how the early collection of citation information brings a universal solution to the problem of citation data quality. The definition of a global citation management workflow, starting at paper edition time, can make citation data flow through the different stages of a paper's lifecycle (edit, review, publish) and thus remove the need for post-publishing harvesting. In short, we propose the adaptation of word processing systems to become *citation-aware*, so that citation information can be captured at edition time, attached to the document as metadata and reused at the publication time. A Global Citation Registry stores the citation information and should be freely accessed by client applications for different purposes. The registry, created and maintained by the scholarly community, will be accessed by different feeding and querying programs via a service-oriented interface.

The paper is structured as follows. In section 2, we describe the different current citation counting-based approaches. In section 3, we outline our proposal using a scenario of citation information management. Such a scenario can be realized in terms of a framework for automatic citation management that we introduce in section 4. The framework's generic service-oriented architecture is defined in section 5. To validate the feasibility of our proposal, we have developed a prototype that is described in section 6. Some conclusions and an outline of the future work conclude the paper. A demonstration of the features of the prototype will be presented at the conference.

2 State of the Art

Citation counts are at the core of most impact models such as the one used by the Thomson's Web of Science (WoS) [1], the Hirsch index (also known as h-index) [2], and others. Researchers in the Bibliometrics field have studied the different approaches to citation counting and analysis for years, with the aim of assessing the quality of citation data provided by the different systems. A study performed by

Bosman et al. [3] defined quality of citation data as a combination of the following factors:

- Publication coverage: number and type of publications analyzed.
- Temporal coverage: temporal range covered for a given publication.
- Up-to-dateness: most recent year covered for publications.
- Information richness: variety of data collected for each publication.

Over all of these features, accuracy of the collected data is a key aspect that must be especially enforced.

Manual or human-supervised systems are those in which citation records are built with human participation, in the whole or in part. The most representative system of this class is by far the WoS. Recently, new systems have appeared, being Elsevier's Scopus [4] among the most relevant ones. These systems produce very accurate citation data [5]. However, human participation has high cost, which has several consequences. First, the increasing number of publications in recent years is making things more expensive as more human participation is needed; as result, not all the existing journals are indexed in such databases, yielding low coverage rates. Second, several authors have also criticized these systems because only journals are indexed, ignoring conference proceedings and other types of scholarly materials, which are very relevant in fields like Computer Science. And third, up-to-dateness is hard to achieve. As of March 14, 2008, the newest data at the WoS's Journal Citation Reports correspond to 2006.

Other freely accessible systems have arose which automatically extract citation information from online documents; this has led to a significant increase of publication coverage over manual systems, as well as (at least in theory) to enhanced up-to-dateness. Citeseer [6] was one of the pioneer systems, followed years later by Google Scholar [7] and Microsoft's Windows Live Academic Search [8]. Unfortunately, due to the automatic nature accuracy in these systems is much lower than in manual systems, and errors are handled in different ways, not all of them appropriate. Citeseer, for instance, requests the collaboration of publications' authors to fix incorrect data. In other cases, users are simply warned of possible errors, as in the ACM Digital Library, where the bibliography lists are automatically built from the results of a full text optical character recognition process. In the worst cases, errors are just ignored and wrong information is shown to users [9].

It is not clear which of the above approaches is the best one. In fact, one can find studies that demonstrate the supremacy of either one [11, 12]. Most of these studies, however, neglect what we believe is the root of the problems of current citation counting systems: citation metadata are collected *after* papers have been published so problems like author identification, etc. arise and decrease automatic methods benefits. However, citation metadata is generated earlier in the paper lifecycle: it is a paper's author who decides, at writing time, which papers to cite. Notice that, from this viewpoint, a citation record is created every time a new reference is inserted into a document. So, why such metadata are systematically discarded and re-discovered later by citation counting systems?

A smarter use of citation metadata along the paper's lifecycle can solve most of the problems of current citation counting systems, as the scenario described in the following section illustrates.

3 A Motivating Scenario

In this section, we describe a typical scenario for enhanced citation management. The text in italic typeface describes actions, not performed in current citation management environments, that lead to high quality citation records.

Jane is writing a paper about citation management that she plans to submit to the next edition of the ECDL Conference. The word processor used by Jane has a Bibliography Manager (BM) associated that helps her to prepare the bibliography of the paper. When Jane wants to insert a citation to a given paper, she looks for its metadata in her personal bibliographic collection using the BM interface. If the paper is not found on her personal collection, the BM can send a federated search request to find the paper in some bibliographic collection like DBLP [13]. Once the BM has retrieved the cited paper's metadata, a citation is inserted into the text at the cursor position. When the paper is almost finished, Jane asks the BM to automatically generate the bibliography list, normally at the end of the document, following a given bibliographic style. *At the same time, Jane's BM generates an XML file (the so-called* ***citation file****) containing all the citations included in the paper. A key-based mechanism ensures that a given citation file corresponds to a given version of the source document.*

Jane submits the paper using the conference's Web-based submission management system, and several weeks later an email message from the conference program chair containing the notification of acceptance arrives to her mailbox. Jane is happy to know that her paper has been accepted, and starts preparing the camera-ready version. *When Jane uploads the camera-ready copy, she is requested by the submission management system to upload both the paper and the citation file. Actually, no paper is accepted for publication without its corresponding citation file.*

Once all the camera-ready copies are received, the program chair starts the preparation of the program, as well as the conference proceedings. *When the entire program is set, and the proceedings camera-ready copy is published, the program chair (or, alternatively, its agent at the publisher) executes one more task: the registration of the published papers at the Global Citation Registry (GCR), which processes and stores the information obtained from the citation files associated to the accepted papers. Obviously, the update of the GCR (via a web service) is restricted only to the editors of the different publications, who are responsible of the accuracy of the citation information being submitted. Another web service allows citation analysis tools to access to citation data for other purposes like, for example, comparing researchers' curricula vitae.*

4 A Framework for Automatic Citation Management

We introduce a framework that provides a satisfactory solution to the problem of creating and maintaining high quality citation metadata. Our goal is to enforce the quality properties mentioned earlier, as well as to fulfill two additional requirements that will make it viable as a universal solution: to be platform independent, to accept documents created under any technological settings, and to generate trust (and hence acceptance) among the research community. In this paper, we focus on the first requirement. We describe the basic building blocks of the framework, and give clues about its implementation using current technologies.

4.1 Information Infrastructure

The framework is built over three main components: the *Global Citation Registry* (GCR), which stores the citation data; the *Global Author Registry* (GAR), where authors are registered to define a unique authority associated to all their publications, and the *Global Publication Registry* (GPR), an abstraction of a global digital library where research documents and their associated metadata can be retrieved from.

Fig. 1 shows the conceptual model of part of the information infrastructure underlying the framework. It is expressed as a UML class diagram; for simplicity, class attributes are omitted. Classes in the diagram have been associated to three main components of the framework; we are specifically interested in the GCR, so we have included only the details of the GAR and the GPR that are relevant for our discussion.

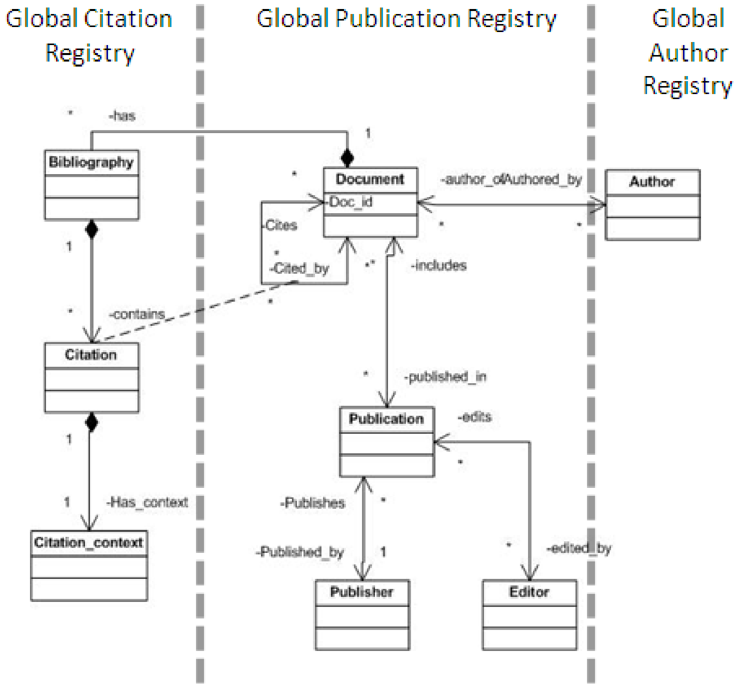

Fig. 1. Simplified conceptual model of the citation management framework.

Author registration services are already available at the portals of some publishers (e.g., the Scopus Author Identifier[2]); when a researcher registers to the GAR, he or she gets an author identifier that can be used in any further publication. This way, deciding whether two author names represent or not the same researcher becomes much easier than it is nowadays.

In our model, the research work (e.g., a research paper) of one or more *authors* is expressed in one *Document*. The document's *Bibliography* is a collection of references to other documents called *Citations*. A citation is characterized by the citing and cited documents, and the citation *Context*, composed of additional metadata that can be used to complement the mere citation count used in current impact assessment systems. Examples of context elements are: the position of the citation in the document, whether authors are citing a document to support or reject the ideas presented in it, or the number of times a document is cited in a given document. Finally, a document is published in a given *Publication*, which has a *Publisher* and one or more *Editors*.

The model in Fig. 1 is minimal, in the sense that more properties could be added to the different classes. For instance, in heterogeneous settings, different metadata schemas can be used to specify documents and publications properties. However, a document identifier is mandatory, as it will be used to link documents by citation relationships.

4.2 Global Citation Management Workflow

During the edition process, researchers often use bibliography managers to control the insertion of citations and the automatic generation of the bibliography lists in documents. Examples of bibliography managers are the BibTeX macros used with the LaTeX typesetting system, or EndNote, ProCite, Refworks and many others used with Microsoft Word (indeed, the 2007 version of Word includes a built-in bibliography manager).

As citations can be added or removed at different moments during the edition of a document, the mere insertion of a citation cannot be considered a relevant fact on itself. Some operation is required to consolidate the bibliography, after which no citations can be added or removed to the document. It seems natural that a check-out operation, which cannot be undone, must be executed at the document's publication time. In order to perform such an operation, the document's citation metadata must be made explicit, either embedded in the document or in the form of a companion citation file, which can be uploaded to the GCR when a document is registered as published. Although most document formats can include embedded metadata, having a separate citation file hides the heterogeneity of word processing systems, and hence eases the GCR feeding process[3].

[2] http://info.scopus.com/etc/authoridentifier/

[3] CrossRef (www.crossref.org) has launched a service called "Forward Linking" which requests publishers to upload citation files in order to be able to find papers citing a given one. However, this service is only optional and citation data are not public and therefore not usable by authors.

The GPR can be seen as a global, federated search service which gets documents along with their metadata from different digital libraries or bibliographic collections; also, it holds information about the different publications, such as the editor, publisher, volume and number of a journal issue, edition of a conference, and so on. From the citation management perspective, the association "published_in/publishes" links a document with the publication where it appears. For our purpose, citations are created and inserted in the GCR once a document has been *officially* published, and not otherwise. By *officially* we mean that the editor responsible of the publication of a paper (e.g., the editor in chief in the case of a journal, or the program chair(s) in the case of conference proceedings), acting on behalf of a publisher, or the publisher itself, execute the check-out operation mentioned earlier.

The key for the success of the framework is its universal adoption and use, which requires a twofold effort: on one hand, the development of a software infrastructure supporting the overall workflow; on the other hand, such infrastructure must be supported by a general agreement of the research community to accept the model and populate the GCR with quality data.

5 A Service-Oriented Software Infrastructure

The most important requirement for a universal solution is platform-independence, which means that any word processing system, content management system, or any other application can use the framework services. Thus, service oriented architecture seems the right option at the current technological settings. The framework is structured in the layered architecture shown in Fig. 2.
The storage layer includes the different registries involved (GCR, GAR and GPR).

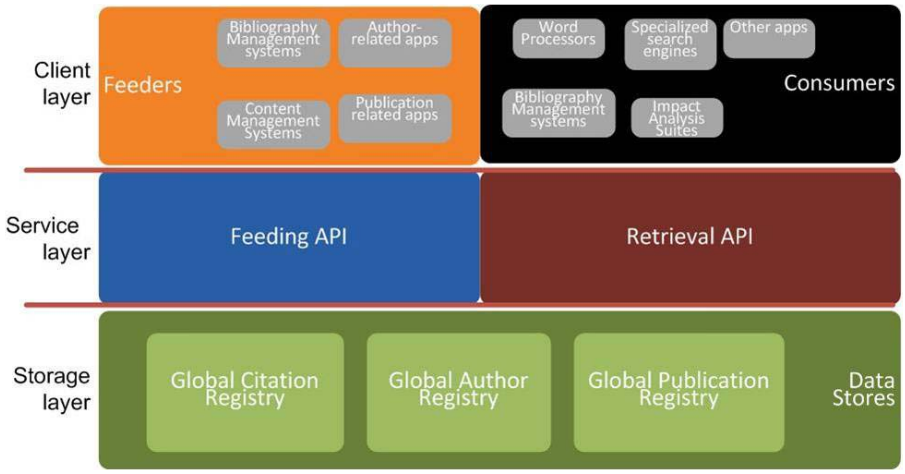

Fig. 2. Generic architecture of the software infrastructure supporting the framework.

Whether they are centralized or distributed is not relevant at the moment, and will be a design decision driven by non functional requirements. On top of the storage layer, the service layer offers two basic Application Program Interfaces (APIs); on one hand, the Feeding API includes services that provide data to the different registries; on the other hand, the Retrieval API offers data retrieval services to its clients. Both APIs can invoke services of the three registries of the storage layer. Finally, at the client level we can find all the applications that create and/or use author, publication or citation data. As with the services APIs, we group in the Feeders category those applications that somehow provide data to any of the registries through the Feeding API. Typical feeders are the content management systems used by publisher to upload citation files, as explained earlier. Consumer clients are those applications that can make use of the information via the Retrieval API, such as impact analysis suites.

The infrastructure must be completed by introducing some adapters to some of the client applications to make them “citation aware”. Specifically, word processing systems with built in bibliography management utilities (such as Word), or third party bibliography managers (e.g., EndNote, RefWorks, etc.), should include functionality to generate citation files, or at least export citation data in a format easily transformable into a common citation file format. If possible, such a functionality should be provided in the form of plug-ins that users interested in the framework can download and install. Publishers should modify the content managers’ workflows to include the feeding of the registries with citation, author and publication information. From the retrieval perspective, applications can use citation data with other purposes than impact analysis. Bibliography managers, for instance, can offer to their users advanced search facilities that exploit the citation relationships (e.g., “find papers on Digital Libraries published at the ECDL proceedings and cited by at least two other papers in the last two years”).

6 Implementation of a Prototype

To validate the proposal, we have instantiated the reference architecture of Fig. 2 in a prototype, whose architecture is shown in Fig. 3.

At the storage layer, we use DBLP-DOI as GPR, a database that stores the records of the DBLP collection that have a DOI [14] associated (around 250,000 items out of

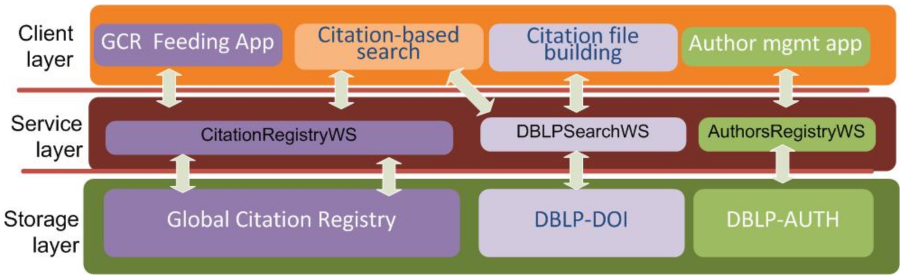

Fig. 3. Architecture of the prototype

one million). Each bibliographic record (each document, in terms of our conceptual model) is represented by a *BibItem* that has, among other attributes:

- the *DOI*;
- the *type*, whose value can be any of the eight types of publications supported by DBLP: Article, Inproceeding, Proceeding, Book, Incollection, PhdThesis, MastersThesis and Www; and
- *FullText*, an XML text containing the full BibItem's DBLP metadata record (Author, Editor, Title, Booktitle, Cite, Year, Page, Journal, Address, Publisher, Number, Volume, Url, Month, Chapter, Cdrom, School, EE, Serie, ISBN, Crossref and Notes). The attribute *FullText* allows performing searches over the full record.

The GCR has been implemented as a database that registers citations among documents. For each citation, we store the DOI of the citing and cited documents, as well as the citation context. At the moment, we have represented context as a textual description, which will be replaced by a more sophisticated context model in further versions of the prototype. The database has been populated with 1,500,000 citation records resulting from a process that generated randomly around 10 citations per each BibItem in DBLP-DOI.

Finally, the GAR has been implemented as a database that records author information (name, surname, address, institution, etc.), and co-authoring relations. We have extracted author data from the DBLP's authors information records.

On top of the storage layer, the feeding and retrieval APIs have been implemented as three web services, one per each component of the storage layer. The GCR is accessible through the CitationRegistryWS[4] web service, which publishes two web methods:

- *InsertCitationFile* is used to feed the GCR from citation files. A citation file is associated to a document and includes all the citations contained in the document (see section 3). A sample citation file appears at Fig. 4. The DOI of the citing document is given as the value of the attribute "id" of the element <Citation_file>. Other identification mechanisms could be used and specified in the "schema" attribute of the same element. The

```
<?xml version="1.0"?>
<citation file id schema="DOI" id="10.1002/asi.10351">
  <Citations>
    <Cited doc id schema="DOI" id="10.1002/asi.10352">
      <context>related work</context>
      ...
    </Cited doc>
  ...
  </Citations>
</Citation_file>
```

Fig. 4. Sample citation file

[4] http://www.bibshare.org/CitationFramework/CitationRegistry/CitationRegistryWS.asmx

citations contained in a document are listed as <cited_doc> elements, and are composed of the cited document's DOI and the citation context.

- *GetCitations* accepts a DOI as argument and returns a list of the DOIs of the BibItems that cite the document.

The DBLP-DOI database is accessible via the DBLPSearchWS[5] web service, which includes two web methods:

- *GetRecords*: it takes as input argument an XML-encoded query and returns the records of the database that match the criteria, also in XML. A query is composed of one or more parts combined with logical connectives (AND, OR). Each part follows the pattern "Attribute like/not like Value". It is also possible to search for a type of publication or a text in the full metadata record. The following XML fragment is an example of a query that retrieves all the papers that the author with name Borgman has published at the ECDL conference:

```
<?xml version="1.0"?>
<AdvancedQuery>
 <SearchConditionFirst
   Field="author" Comparation="Like" Value="Borgman" />
 <SearchCondition
   Condition="AND" Field="booktitle" Comparation="Like"
   Value="ECDL" />
</AdvancedQuery>
```

  We have developed a library for the definition of search criteria that serializes the XML in the format presented above. When a query is sent to the GetRecords web method, it returns an XML where each result has the same format as the bibliographic records of the original DBLP collection, plus an attribute that holds the document's DOI.

- *GetRecord*: it accepts a DOI as argument and returns the full BibItem in the same format as *GetRecords*.

The AuthorsRegistry database can be accessed through the AuthorsRegistryWS[6] Web service, which includes five web methods:

- *GetRecord* and *GetRecords*: accept an id or part of the author's full name, respectively, as argument and return the full author record.
- *InsertRecord, UpdateRecord, and DeleteRecord:* inserts, updates and deletes author records, respectively.

At the client layer, we have developed a web-based interface to access the different framework services (see Fig. 3). To insert citations into the GCR, a web application[7] allows users to upload citation files. When one wants to upload citations of a document that has no citation file associated, another web application[8] allows generating citation files from the DBLP-DOI following the shopping cart metaphor.

[5] http://www.bibshare.org/CitationFramework/DBLP-DOI/DBLPSearchWS.asmx

[6] http://www.bibshare.org/CitationFramework/AuthorsRegistry/AuthorsRegistryWS.asmx

[7] http://www.bibshare.org/CitationFramework/CitationRegistry/Insert.aspx

[8] http://www.bibshare.org/CitationFramework/CitationRegistry/Shopping.aspx

This application is intended for users of word processors without citation file generation capabilities, as well as for integrating legacy citation data into the GCR.

To exploit citation data, we have developed a citation-enhanced bibliographic search application[9] that allows users to retrieve bibliographic records that match a search criteria based on both bibliographic and citation parameters (for instance, to find the papers of an author that have a number of citations over a given threshold). This application uses both CitationRegistryWS and DBLPSearchWS Web services. Finally, a simple application provides basic functionality to handle authors' data.

Currently, we are working on the development of an extension, compliant with the framework, of the Bibshare's bibliography management system [15]. Bibshare is a general purpose bibliography management environment that allows different word processing systems to share bibliography collections, either locally or remotely. Specifically, we are extending the Bibshare client for MS-Word with citation file generation capabilities, as well as with citation-based searches.

7 Conclusions and Further Work

The scientific community uses impact measures to make some of the most important decisions affecting people's careers. However, unlike one could expect, there is a common feeling that the right evaluation model is still to come and current impact assessment systems are used because they provide the "least bad" solution to the problem of evaluating scientific output. Whatever impact metric we use (e.g., productivity, citation ranking, or the h-index), reasonable pros and cons can be stated. Indeed, most systems have been criticized for the low quality of the citation data they manage; surprisingly, no alternative solutions have been provided so far.

We have introduced a simple technological solution based on the early collection of citation metadata that significantly improves the quality of citation information. Instead of discarding such data, as we currently do, we have shown how a global workflow can be defined as the composition of processes that have been considered independent for years. This way, citation data are collected at the very moment of their generation (when editing a document), and moved through the content management systems of publishers to the Global Citation Registry that can be accessed by any system, regardless the impact measurement models used.

The actual implementation of such a workflow depends heavily on the acceptance of the proposal by the research community, which only will be possible if supporting tools are available. We have implemented a prototype to show that our proposal is easily implementable. Three web services provide access to the Global Citation Registry, Global Publication Registry, and Global Authors Registry to all the potential client applications, like bibliography managers, content managers, and others.

Last, but not least, we believe that collaboration among all the actors is crucial as all of them can benefit from this initiative. Authors will trust citation data as their overall quality will be higher than in current systems; the developers of bibliography managers will see how a universal application of the framework will increase the use

[9] http://www.bibshare.org/CitationFramework/CitationRegistry/AdvancedSearch.aspx

of their systems; the management of citation data at the publishers site can provide richer metadata they can offer to their customers; and, finally, impact measurement systems will obviously have access to more and better data than they get now in a pure automatic way.

Further work includes the specification of an XML schema for citation files including a richer context model. Also, the development of plug-ins for the different word processors and bibliography managers that allow the generation of citation files. Finally, there is a clear need for a module to cope with the heterogeneity of document identification systems, though we believe that the universal adoption of the DOI will ease the management of citations.

Acknowledgements. Work of J. H. Canós and M. Llavador is partially funded by the Spanish Ministerio de Educación y Ciencia (MEC) under grants META (TIN2006-15175-C05-01) and CAPES/DGU 2008 (PHB2007-0064-PC), and Generalitat Valenciana (ACOMP07/216). M. Llavador is the holder of the MEC-FPU grant #AP2005-3356. Work of E. Mena is supported by the MEC under grant TIN2007-68091-C02-02. M. Borges' work has been partially funded by grant #169/08 from CAPES/DGU (Brazil-Spain).

References

1. Thomson's Web of Science, http://scientific.thomson.com/products/wos/
2. Hirsch, Jorge E., (2005), "An index to quantify an individual's scientific research output," PNAS 102(46):16569-16572, November 15 2005.
3. Bosman, J., van Mourik, I., Rasch, M., Sieverts, E. and Verhoeff, H. Scopus Reviewed and Compared. *Utrecht University Library* (June 2006); http://igitur-archive.library.uu.nl/DARLIN/2006-1220-200432/Scopus%20doorgelicht%20&%20vergeleken%20-%20translated.pdf
4. Scopus: http://www.scopus.com
5. Adam, D. The counting house. Nature, Vol. 415, February 2002, pp. 726-729.
6. Scientific Literature Digital Library: http://citeseer.ist.psu.edu
7. Google Scholar: http://scholar.google.com
8. Microsoft Live Search Academic: http://academic.live.com
9. ACM Digital Library, http://www.acm.org/dl
10. Jacsó, P. Deflated, inflated and phantom citation counts. *Online Information Review,* Vol.30 No.3, 2006, pp.297-309.
11. Jacsó, P. Google Scholar: the pros and the cons. *Online Information Review* Vol.29, No.2 (2005).
12. Harzing, A.-W. Reflections on Google Scholar (2007), http://www.harzing.com/pop_gs.htm
13. Computer Science Bibliography (DBLP), http://dblp.uni-trier.de
14. The Digital Object Identifier System, http://www.doi.org.
15. Canós, J.H., Llavador, M., Ruiz, E., Solís, C. A Service Oriented Approach to Bibliography Management. *D-Lib Magazine*, Vol. 10, no. 11, November 2004.