

A Pilot Web Environment Implementing Cascading Citations*

Y. Asmanidis¹, D. Dervos¹, G. Evangelidis², N. Samaras²

¹ Dept. of Information Technology, Alexander Technology Educational Institute (ATEI),
Thessaloniki, Greece

Tel: +30 2310791295, Fax: +30 2310791290, E-mail: {ypasm@lib, dad@it}.teithe.gr

² Dept. of Applied Informatics, University of Macedonia, Thessaloniki, Greece

Tel: +30 2310891844, Fax: +30 2310891800, E-mail: {gevan,samaras}@uom.gr

Abstract

In this paper we present a web environment implementing the cascading citations paradigm. The pilot implementation consists of three components, the Universal Author Identifier system (UAI_Sys), the Cascading Citations Indexing Framework system (c²IF_Sys) and the c²IF algorithm. The inner functionality of the c²IF algorithm lies beyond the scope of this presentation, comprising the topic of a separate paper. In this respect, the focus here is on UAI_Sys and c²IF_Sys. The two components are implemented as web applications, and they co-function by utilizing web services. In its implementation, c²IF_Sys utilizes citations data from the ISI Science Citation Index Expanded (ISI SCIE) made available from Thomson Scientific (<http://scientific.thomson.com/>) along the lines of the Cascading Citations Analysis Project (C-CAP, <http://www.ccapnet.org/ccap/>).

Keywords: Author Identification, Citation Analysis, Citation Indexing

1. INTRODUCTION

Nowadays, the scientific community is still in search of a scheme that measures the contribution research publications make in science and technology. Eugene Garfield has been the first to introduce a metric (*Impact Factor*) that could be used to measure the impact of scientific journals over time [3,4,5]. Variations of this proposal have also been introduced; however concern has been expressed about the fairness of such schemes [22,23]. In the Cascading Citations Analysis Project (C-CAP), the citation index paradigm is extended by also considering citations at the (*article, author*), not just citations at the article level [1,2]. In addition, indirect, as opposed to only direct, citations are considered (*ibid.*). An implication of considering citation at the (*article, author*) level is that each one author need be uniquely identified. The name disambiguation problem relates to the existence of homonyms and to having more than one name variants for the same author [19]. In C-CAP, in a way analogous to that of other analogous systems (e.g. [20]), a Universal Author Identifier (UAI) is introduced and maintained by the UAI_Sys web application [6]. UAI_Sys makes possible for each one author to maintain his/her own profile, indicating which fields of the latter are meant to be accessible by the public.

* Research conducted along the lines of the Cascading Citation Analysis Project (C-CAP, <http://www.ccapnet.org>), funded by the Research Committees of ATEI, and the University of Macedonia, Thessaloniki, Greece.

The Cascading Citation Indexing Framework (codenamed: c^2IF_sys), is a web based application that allows each one UAI_Sys registered author to claim/deny authorship on published articles that include his/her name (or any one of his/her name variants) in their authors lists. Its pilot implementation utilizes a subset of the Science Citation Index Expanded bibliographic database (<http://scientific.thomson.com/products/scie/>): currently, years 1999 through to 2005. The latter has been made available from Thomson Scientific (<http://scientific.thomson.com/>), to be used along the lines of C-CAP.

Implementing the C-CAP introduced extended citation analysis paradigm, c^2IF_Sys co-functions with a c^2IF algorithm backend software [7], and calculates a citation standings output for all publications an author has claimed authorship on. The system co-functions with UAI_Sys , making it possible for each one author to register and maintain his/her own metadata. The user (author) establishes access to both systems under the same (UAI_Sys maintained) login credentials.

In section 2 (System Overview), an overview of the c^2IF_Sys/UAI_Sys architecture and selected parts of its functionality are presented. Next, in section 3 (Web Services Approach) interoperability issues are considered, relating to the way c^2IF_Sys and UAI_Sys work together in the network. The scheme may be extended to have more web application co-function with UAI_Sys , the way c^2IF_Sys does. User roles in the combined c^2IF_Sys/UAI_Sys environment are outlined in section 4 (User Roles). Section 5 (Technologies Used) summarizes on the software platforms used. Lastly, section 6 (Conclusion) wraps up on the topic, addressing the ‘What is next?’ question for UAI_Sys , c^2IF_Sys , and C-CAP.

2. System Overview

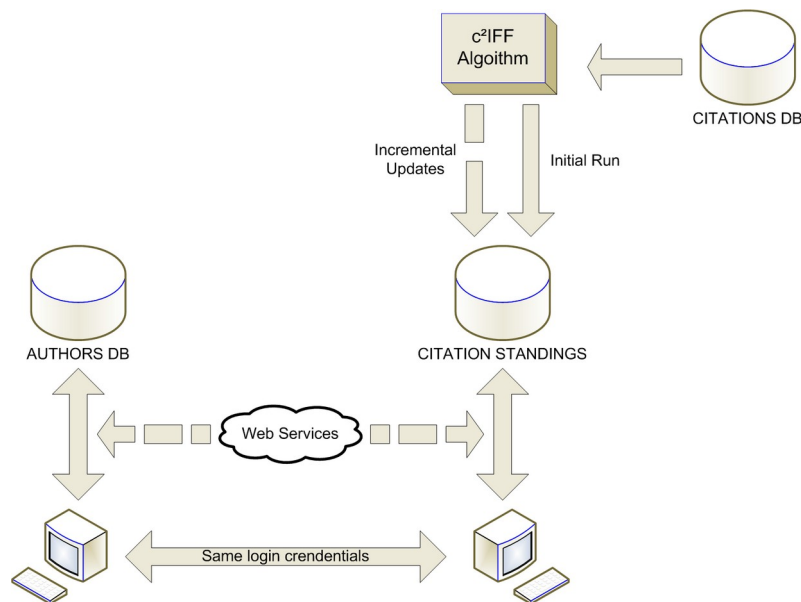


Figure 1 UAI_Sys/c^2IF_Sys architecture

As it is shown in Figure 1, c²IF_Sys co-functions with both UAI_Sys and the c²IF algorithm backend. This web-services based co-functionality makes it possible for each one author to both maintain his/her own metadata in UAI_Sys, claim authorship on the articles s/he has published, and obtain the corresponding citation standings output.

A UAI_Sys registered author is expected to have his/her account be authorized by a UAI_Sys agent (privileged UAI_Sys account, please refer to section 4, below). The authorization process is carried out only once, per UAI_Sys account. It involves the verification of the individual's identity, and enables the author in question to proceed and claim/deny authorship on Science Citation Index Expanded registered articles (at the level of the c²IF_Sys environment, not SCIE, of course). The author supplied (own) metadata are stored in the UAI_Sys controlled relational database schema.

The c²IF_Sys authentication method is using web services to verify the login information from the UAI_Sys users database. While in c²IF_Sys, the user can access the authorship menu, consisting of four options. From these options s/he can browse/claim/deny ownership on articles that appear to match his/her name or name variants(s), as registered with UAI_Sys. Having done so, the author effectively categorizes all articles (co-)authored by his/her name or name variant, as follows:

- **Open Articles** : articles that: (a) appear to have a variation of the author's name in their authors list, (b) have not claimed by another author whose name or name variant is a homonym to the name, or to a name variant of the author in question. In this respect, any one member of the open articles set may be claimed/denied to have been (co-)authored by the author in question.
- **Claimed Articles** : articles that initially belonged to the 'open articles' category, claimed to have been (co-)authored by the author in question. This list is further divided into two sub lists. The "List of Claimed Articles with Citations" and the "List of Articles with no Citations". The first list contains articles that have at least one citation in the c²IF_Sys database. The second one contains articles with no citation at all. The first list displays the articles that have been processed by the c²IF algorithm.
- **Taken Articles** : articles that: (a) appear to have a variation of the author's name in their authors list, (b) have already been claimed by another author whose name or name variant is a homonym to the name, or to a name variant of the author in question. Taken articles are displayed to the author in question, in case s/he wants to file a petition with the system administrator, claiming authorship on an article that has already been claimed by another author.
- **Denied Articles** : articles that: (a) appear to have a variation of the author's name in their authors list, (b) the author in question has denied (co-)authorship on.

The authorship menu includes one more option which allows each one author to initiate the calculation of the citation standings output for all articles s/he has claimed authorship on. The citation standings output is stored in a c²IF algorithm controlled relational database schema. Upon completion of the citation standings construction stage, the author is notified by an email notification. The author has the option of checking the status of his/her citation standings snapshot, namely the (claimed) articles awaiting to be processed by the

c²IF's algorithm, and the articles already processed, i.e. the ones present in the citation standings table.

Both systems maintain a complete/detailed log of all user initiated update operations. The latter may be used to trace critical operations whereby a UAI_Sys user updates his/her own metadata, or when a UAI agent initiates privileged administrative operations on UAI_Sys accounts (e.g. a reset password operation), etc.

Last but not least, the content of UAI_Sys need be searchable, both by the public user as well as by the registered one. As it stated earlier, it is to each one author's discretion which of his/her profile metadata fields are to be accessible by the public (UAI_Sys registered, or public users).

3. Web Services Approach

The World Wide Web Consortium (W3C, <http://www.w3.org>) defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network. Web services are usually web APIs that can be accessed over a network, such as the Internet, running on a remote system hosting the requested services [21].

From the C-CAP perspective, provision is taken for the two modules to incorporate web services facilitating communication with third party applications. More specifically, UAI_Sys exposes own code and author name variants (aliases) to c²IF_Sys. This way, it becomes possible for the user/author to claim/deny authorship on selected publications, via the c²IF_Sys interface. This functionality is open to future extensions for web-based co-functioning with any third party software that utilizes web services. In this respect, it becomes possible for UAI_Sys to make available to other applications selected subsets of author relating metadata (e.g. UAI code, authorization status, etc.), in a transparent way, over the Internet.

Utilizing analogous web services functionality, it becomes possible for c²IF_Sys to receive bibliographic (citation) data from other applications, over the Internet. In return, the citation standings output of c²IF algorithm can be broadcasted to remote applications. In the current pilot implementation, the c²IF algorithm software backend co-functions with c²IF_Sys communicate by sharing database tables in the RDBMS residing database schema. The c²IF_Sys backend calculates the increments of the citation standings (tabular) output, while c²IF_Sys provides the user interface and queues-in all new incoming requests (Figure 1).

Concluding with the interoperability of the system, it is noted that it has been designed to extend beyond the context of the C-CAP project; the potential is there for web based co-functioning with applications involving bibliographic data, for example: institutional repositories.

4. User Roles

In the case of c²IF_Sys, there exist two user roles: the administrator, and the author. UAI_Sys, on the other hand, involves one extra role: the UAI agent. In addition, it is also the public user, namely one who accesses UAI_Sys and retrieves the registered author and UAI agent data.

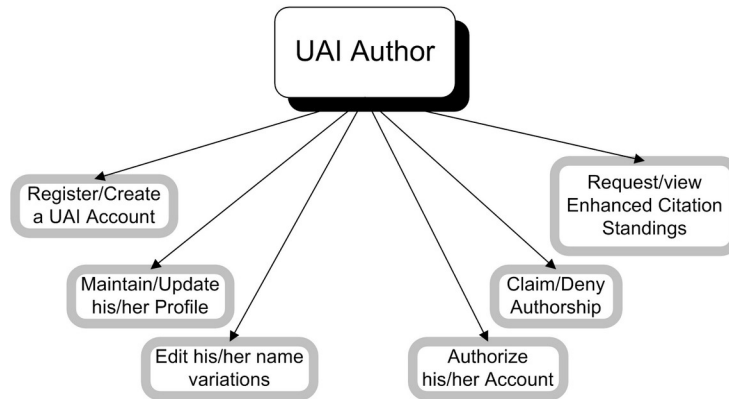


Figure 2. The ‘author’ role system functionality

Figure 2 summarizes on the system supported functionality for the ‘author’ user role. It is noted that the UAI_Sys relating tasks are not differentiated from the c²IF_Sys tasks. Having registered him/her/self with UAI_Sys, the user accesses c²IF_Sys to carry out tasks like claiming or denying authorship on publications having one of his/her name variants appear in their authors lists, and requesting updated versions of his/her citation standings output. A prerequisite for this c²IF_Sys functionality is for the user to have his/her account be authorized by a UAI agent. The UAI_Sys system need then be accessed only when the author wishes to update his/her own profile (metadata) content.

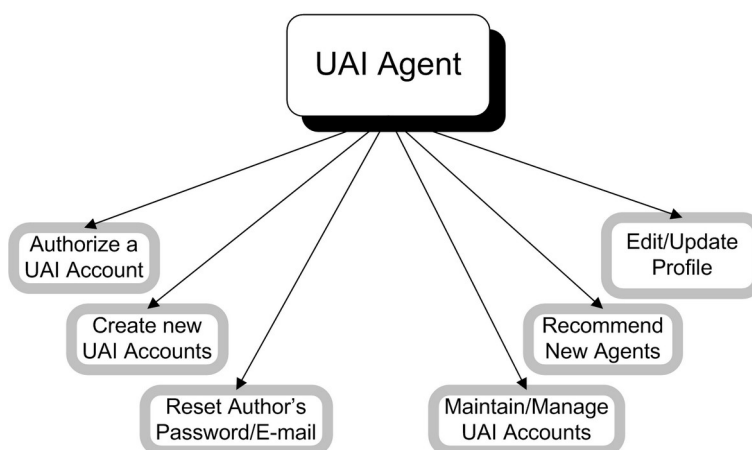


Figure 3. The ‘UAI Agent’ role system functionality

Figure 3 summarizes on the UAI Agent role system functionality. Clearly, the latter is restricted to the UAI_Sys environment. UAI Agent accounts are meant for parties that produce/manage bibliographic data (e.g. libraries, and publishers). The UAI Agent has access to administrative operations that focus on serving the authors in many levels, thus s/he must be a trustworthy entity in the context of the UAI_Sys. The UAI author turns to the UAI agent nearest him/her in order to: (a) have his/her UAI account be authorized, (b) have his/her email address and/or password reset, (c) obtain assistance in having his own profile (metadata) be updated, etc. The UAI agent is also able to create new UAI_Sys user (author) accounts, either in batch or in one-at-a-time mode. Once a new author account has been created by an agent, the latter has the option to continue being the user who maintains/updates the account in question. Such a system functionality is expected to be handy in cases whereby authors prefer to have their local agent be in charge of maintaining their own UAI account/profile (thinking of cases where, for example, the author does not have access to the Internet). As stated above, UAI agents are trustworthy entities. In this respect, it makes sense to have an agent recommend a new UAI Agent account to be created (library consortia members, for example).

5. Technologies Used

Both UAI_Sys and c²IF_Sys comprise J2EE applications [8], utilizing EJB 3.0 components and POJOs (Plain old java objects) to organize the business logic and JSF (Java Server Faces) [9] for the presentation layer and render JSFs as valid XHTML pages [10]. JBoss SEAM [11], a contextual component introduced by JBoss, allows to inject/outject EJB 3.0 components in and out of the presentation layer. The applications are deployed in the JBoss application server [12]. The Hibernate engine [13] is utilized, which is the default object/relational persistence and query service that runs with the JBoss enterprise Middleware platform, to communicate with the RDBMS. The latter is implemented in Postgresql [14].

Web services are utilized in order to co-operate with external systems, utilizing the JBoss technology [15]. They are fully JAX-WS, JAX-RPC, J2EE/JEE web services stack compliant. The application code was generated with the JBoss-ide for eclipse [16]: a series of Eclipse plug-ins to support the development of JBoss applications.

Lastly, the system operates in the Linux operating system environment, using the Slackware distribution with a kernel of the 2.6 series [17, 18].

6. Conclusion

The UAI_Sys/c²IF_Sys pilot application environment has been developed along the lines of the Cascading Citations Analysis Project (C-CAP). The system involves two main components: UAI_Sys and c²IF_Sys. The latter co-function in the Internet to enable authors obtain a unique universal author identifier (UAI) and, having done so, proceed to claim/deny authorship on published research articles whereby one of the author's name variant is listed with the corresponding (article) author lists. Next, the author can proceed to request a (personal) citation standings output, reporting citation info in accordance with the C-CAP introduced (extended) citation indexing paradigm (i.e. including indirect citations and chords targeting each one of the author's articles).

The two web applications support role-based access and management. In its current pilot implementation the system operates on a subset of the Science Citation Index Expanded (SCIE) dataset: years 1999-2005. The latter is made available from Thomson Scientific (<http://scientific.thomson.com/>) in order to be used for research purposes along the lines of C-CAP. The dataset registers 7,364,211 research article records involving a total of 165,822,522 (direct) citation instances.

Future plans include the co-functionality of UAI_Sys with open source institutional repository software, the ultimate goal being the harmonization of the citation standings output obtained from a variety of Internet residing (heterogeneous) institutional repository environments.

References

- 1 Dervos, D., Samaras, N., Evangelidis, G., and Folias, T. (2006). *A New Framework for the Citation Indexing Paradigm*, Proceedings, 2006 Annual Meeting of the American Society of Information Science and Technology (ASIS&T). Retrieved on February 23, 2007 from: <http://dlist.sir.arizona.edu/1714/>
- 2 Dervos, D.A and Kalkanis, T. (2005). *cc-IFF: A Cascading Citations Impact Factor Framework for the Automatic Ranking of Research Publications*. Proceedings of the 3rd IEEE International Workshop on Intelligent Data acquisition and Advanced Computer Systems: Technology and Applications (IDAACS), p. 668-673, Sofia, Bulgaria, 5-7 September, 2005. Postprint version from DLIST, retrieved 15.05.2006: <http://dlist.sir.arizona.edu/1105/>
- 3 Garfield, E. and Sher, I.H. (1963). New factors in the evaluation of scientific literature through citation indexing. *American Documentation* 14(3): 195-201.
- 4 Garfield, E. (1972). Citation Analysis as a tool in journal evaluation. *Science* 178: 471-479.
- 5 Garfield E., (1994). *The Impact Factor*. Retrieved 15.06.2007: <http://scientific.thomson.com/knowtrend/essays/journalcitationreports/impactfactor/>
- 6 Dervos, D., Samaras, N., Evangelidis, G., Asmanidis, Y. and Hyvärinen, J. (2006). *The Universal Author Identifier System (UAI_sys)*. Proceedings of the 1st International Conference eRA 2006, Tripolis, Greece, 15-16 September 2006. Postprint version from DLIST, retrieved 29.07.2007: <http://dlist.sir.arizona.edu/1716/>
- 7 Dervos, D., Samaras, N., Evangelidis, G., and Folias, T. (2007): *Cascading Citation Indexing In Action*, Proceedings of the eRA 2007 Conference, Athens, 15-16 September, 2007
- 8 Java (2007): Retrieved 16.07.2007: <http://java.sun.com/products/ejb/>
- 9 Java Server Faces (2007): Retrieved 20.07.2007: <http://java.sun.com/javaee/javaserverfaces/>
- 10 XHTML - The Extensible HyperText Markup Language(2007): Retrieved 20.07.2007: <http://www.w3.org/TR/xhtml1/>
- 11 Jboss SEAM (2007): Retrieved 16.07.2007: <http://www.jboss.com/products/seam>
- 12 Jboss Application Server (2007): Retrieved 20.07.2007: <http://labs.jboss.com/jbossas/downloads?>

action=a&windowstate=normal

- 13 Hibernate Engine (2007) Retrieved 20.07.2007: <http://www.hibernate.org/>
- 14 Postgresql Relational Database Management System (2007): Retrieved 15.07.2007: <http://www.postgresql.org/>
- 15 Jboss Web Services (2007): Retrieved 20.07.2007: <http://labs.jboss.com/jbossws/>
- 16 Jboss IDE (2007): Retrieved 20.07.2007: <http://labs.jboss.com/jbosside/>
- 17 The Slackware Linux Project (2007): Retrieved 20.07.2007: <http://www.slackware.com>
- 18 The Linux kernel archive (2007): Retrieved 20.07.2007: <http://www.kernel.org>
- 19 Braun, T. (2003). The reliability of total citation rankings. *J. Chem. Inf. Comput. Sci* (43), p.45-46.
- 20 RePeC (Research Papers in Economics Author Service) (2007): Retrieved 26.07.2007: <http://authors.repec.org>
- 21 W3C – the World Wide Web Consortium (2007). Accessed on 28.07.2007: <http://www.w3c.org>
- 22 Glänzel, W., and Moed, H. F. (2002), Journal impact measures in bibliometric research. *Scientometrics*, 53(2), 171–193.
- 23 Moed, H.F. (2005). Citation Analysis of scientific journals and journal impact measures. *Current Science* 89(12), 1990-1996.