Monitoring an Institution's Research Activities

Magda Foti, Elvis Papa, and Manolis Vavalis

***Abstract*—The objective of this paper is to design and implement a web based information system for monitoring and disseminating the research activities of an institution. Our system purpose is to provide the capabilities of searching and to retrieve and present qualitative and quantitative data about the research activities. We consider as a case study the School of Engineering at the University of Thessaly. Each department may categorize and manage its continuous flow of raw data in order to a) produce information that leads to specific qualitative and quantitative evaluation metrics b) present both the raw data and the added valuable data and information in an intelligent and effective way adapted to the background and the interests of the viewer.**

***Index Terms*—Bibliometric analysis, scientometrics, data visualization.**

I. Introduction

Continuous development and progress of a society is directly connected with scientific research. The recording and study of scientific results is undoubtedly an important and valuable information. This information enables us to observe which research communities are making progress, how their research production amount evolves and how worldwide knowledge networks are formed. Through this study of scientific results and activities each institution will have the opportunity to detect the fields of expertise, but also point out the weakness fields of research and put some effort to strengthen them in the near future. Also the access to this kind of information will help each institution to improve its collaboration network.

The question that arises is how and where we can gather all the information needed and gain all the above mentioned benefits. This can only be done by developing and using the appropriate scientific research systems that will allow us to record the research activity in an easy and efficient way and produce this valuable information for any organization, institution and researcher.

In current work we develop a scientific research system. We use University of Thessaly as a case study and try to visualize its research activity using data visualization techniques. Through the developed system, every department will have access to valuable information concerning research activity. Necessary information is collected real time, from big data repositories using crawling techniques.

Manuscript received December 5, 2015; revised March 14, 2016. This work was supported in part by the University of Thessaly Research Committee through an "Applied Research Grand" entitled "A Web Observatory for Research Activities at the University of Thessaly".

The authors are with the University of Thessaly, Department of Electrical and Computer Engineering, Volos, 38221 Greece (e-mail: mafoti@uth.gr, elvis.papa89@gmail.com, mav@uth.gr).

Our main goal is to use this information to visually represent the scientific profile of each department, its scientific productivity, its evolution and its collaborations with other universities, institutions and researchers around the world. Also we want to add a personalization perspective to our system and visualize the information adapted to the background and interests of each viewer.

The rest of this paper is organized as follows. In the next section we study theory and fundamentals of data visualization. In Section III we study web crawlers. We present the usefulness and the structure of a web crawler and the optimal techniques to build an efficient and intelligent web crawler. In Section IV we discuss about the term "bibliometric analysis" and refer to some of the most famous repositories and search engines on research activity information. In Section V we describe the web based information system implemented. Finally our synopsis and future prospects can be found in Section VII.

II. Data Visualization

The World Wide Web can be described as a growing universe of interlinked pages and applications containing huge amount of data. Web is growing exponentially containing nowadays more than half a billion of active websites. This amount of data has turned traditional data processing methods insufficient and this is how the new medium of big data was born. According to Gartner's 2012 definition: "*Big Data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization*." [1]. All this complexity of data led to the need for new data visualization methods.

Data visualization is the pictorial or graphical representation of abstract data [2] and helps people recognize patterns within the large volume of data. In other words, abstract data are translated into physical attributes understandable by human perception, such as length, position, size, shape, color etc.

Data visualization is derived from three main concepts. Visualization, interactivity and knowledge & wisdom.

Visualization is a valuble tool information representation because it benefits from the considerable abilities of houman vision. Visualization makes use of software tools in order to improve understanding of large amounts of data by transalting them into graphical representations. Human vision is a powerful channel for information transmition in human brain with greater capabilities compaired to other senses. This is why people say "*a picture is worth a thousand words*" which describes the power of data visualization.

Interactivity is usefull when multiple static views are needed and when all visual elements cannot be presented on

the same surface at once. Interactive visualizations let people explore data themselves.

Knowledge and wisdom are aquired when reasoning judgment and decision making are enhanced by available information.

Selection of the apppropriate visualization method, highlighting the valuable information is not a trivial task. Factors such as the number of dimensions of the problem to be solved and type of data structure to be visualized must be considered. The main visualization patterns used can be summarized with the following categories:

- Independent quantities (Bar charts)
- Continuous quantities (Line graphs, Stacked area charts)
- Proportions (Pie charts, Ring charts)
- Correlations (Scatterplots, Bubble charts)
- Networks (Diagram maps)
- Hierarchies (Tree diagrams)
- Cartographic (Maps)

There is a great variety of data visualization tools, which can be divided in four main categories, browser based tools, JavaScript libraries, software platforms and programming languages.

III. Web Crawling

Due to the dynamic nature of the Web extracting valuable information has become a difficult task. A web crawler is a software agent that collects information from the web in a systematic and automated way [3]. The web crawler is given as input one or more lists containing URLs and downloads the pages corresponding to each URL. Afterwards, any hyperlinks contained in these pages are extracted and recursively the coresponting pages are downloaded [4]. The whole proccess is shown in Fig. 1.

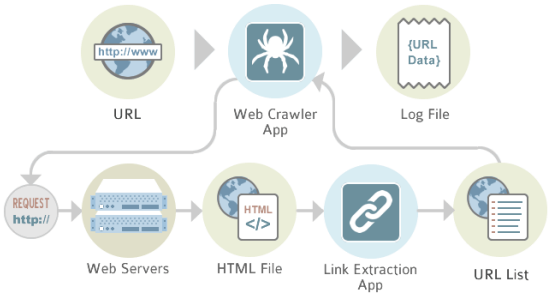


Fig. 1. How a web crawler works.

A web crawler simply sends HTTP requests for documents to web servers. It is the same procedure that is followed by a web browser when the user clicks on a link. Then the crawler handles the result (HTML files) by filtering its content (depending on its policy) and stores it in a file (Log File). There are two main reasons for a task of crawling to become very difficult:

- the amount of web pages to be downloaded
- the rate of change on these web pages

The amount of pages to be downloaded is a problem when the crawler is unable to download all the amount of data so the pages must be prioritized and the crawler has to be smart enough to do that prioritization. The rate of change is a problem when a webpage changes very frequently and the crawler is unable to maintain up-to-day information.

The crawler's behavior is defined by the policies chosen to face the above mentioned problems.

A. Downloading Policy

There are quite a few algorithms proposed for deciding which pages must be downloaded. Here we will see two main types:

- path-ascending crawling
- focused crawling

Path-ascending crawling will execute extensive crawling in the URLs given in the input list. For example, given the URL http://website.com/a/page.html crawler will attempt to crawl /website.com, /a/ and /page.html as shown in Fig. 2. The effectiveness of path-ascending crawler is in finding isolated resources.

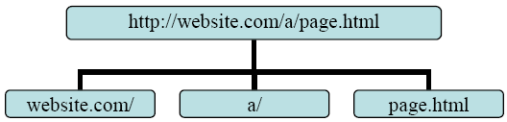


Fig. 2. Path — Ascending crawler.

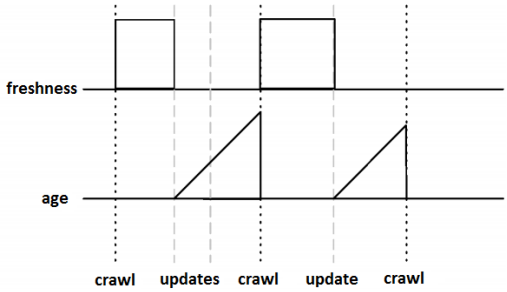


Fig. 3. Freshness vs age.

Focused crawling algorithm calculates the significance of a page. The significance, to the crawler is the similarity between the page content and the driving query. Using this algorithm we can implement focused or topical crawlers. Features like URL and anchor text are used to enable the crawler to calculate the similarity before actually downloading the page [5]. Crawler uses a text classifier to decide whether a page is on topic.

B. Re-visit Policy

From the time the crawler has downloaded a copy of a webpage until the crawler has completed its task many events could have taken place. Such events include creation of new pages, deletions or updates. Two metrics quantifying the effectiveness of the web crawler are freshness and age [6].

Freshness is a binary indicator, which indicates whether the copy saved in crawler's repository is up-to-date. Freshness of page p is calculated using (1).

$$Fp(t) = \begin{cases} 1, & \text{p is equal to the local copy at time t} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Age is a metric counting how outdated is the local copy. Age of p is calculated using (2):

$$Age_p(t) = \begin{cases} 0, & \text{if p is not modified at time t} \\ t - modificationTime, & \text{otherwise} \end{cases} \quad (2)$$

In Fig. 3 we can compare the behavior of the two metrics.

IV. Scientific Research

Some of the most widespread bibliometric databases are:

- Scopus
- Web of Science
- Microsoft Research Academic
- IEEE Xplore
- Google Scholar
- Research Gate

Web of Science (WOS) and Scopus are considered the most commonly used bibliometric databases and are frequently used for searching the literature [7]. Scopus is the largest research literature database. It is an abstract and citation searchable database of research literature and selected web sources published after 1966. Also, it is continually updated and expanding [8].

A. Bibliometric Analysis

Bibliometric analysis, or sometimes called Scientometrics, is the main tool nowadays used by science to conduct quantitative analysis on its activities. Bibliometric analysis is the process of recording and statistical processing of data related with scientific publications and the calculation of bibliometric indicators.

Some of the main bibliometric indicators, which we will also use in our system, are:

- Number of publications. This indicator shows the amount of publications produces by an entity, which may be a specific researcher, a university or a specific field of research.
- Number of citations. It indicates the number of publications that refer to the specific publication under consideration. It indicates the recognition and influence of the publication.
- Citation impact count the average number of citations per publication. It is the ratio between the number of references to the total number of publications in a certain period.
- H-index was introduced by the physicist Hirsch [9]. h-index is calculated as follows: "*a scientist has index h if h of his/her Np (published papers over n years) papers have at least h citations each and the other (Np - h) papers have ≤h citations each*".

B. Researcher Registries

ORCID[1], which stands for Open Research and Contributor ID, was firstly introduced in 2012. ORCID is an open, non-profit and community-driven platform. Its goal is to "*create and maintain a registry of unique researcher identifiers and a transparent method of linking research activities and outputs to these identifiers*". ORCID is available to individual researchers free of charge.

V. Related Efforts

Some Universities are already managing and monitoring their research activities. We have selected two such efforts to present them here.

[1] http://orcid.org/

The University of Pittsburgh has built an integrated dashboard based on the PlumX and the ORCID (Fig. 4). It has make full use of the PlumX and the ORCID functionalities.

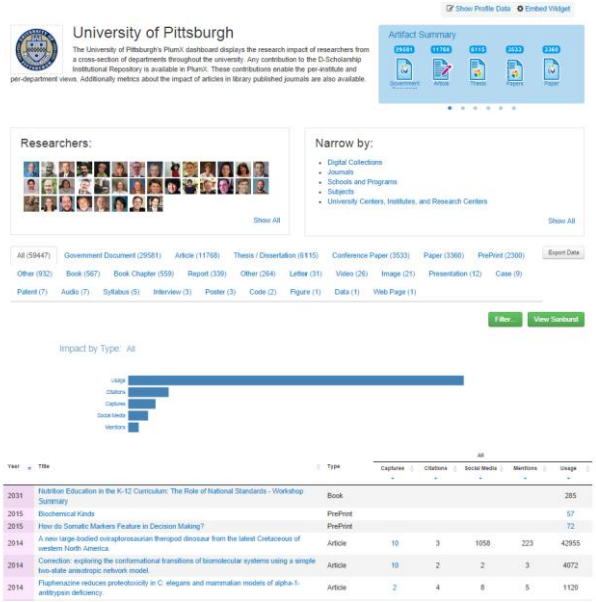


Fig. 4. University of Pittsburg research monitoring.

The Aalborg University has also a research monitoring system called VBN which registers the university's publications, research projects, research activities and press cuttings. The statistics page demonstrates the university's number of publications per year and publications per type per year (Fig. 5).

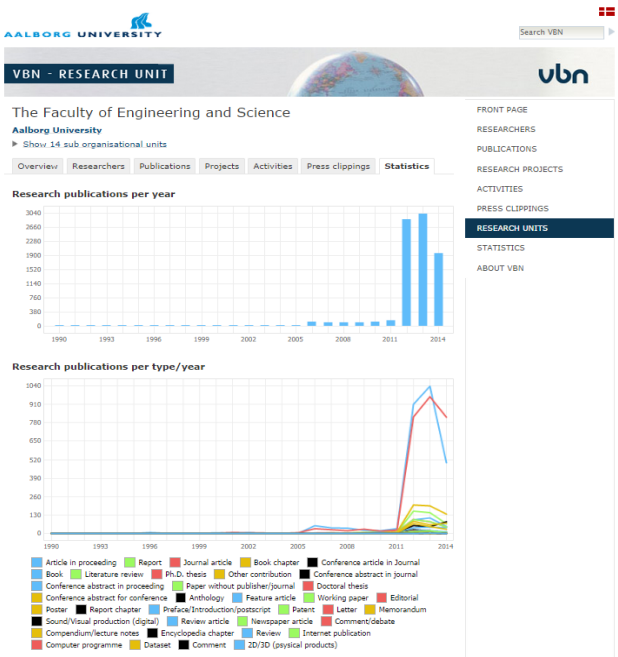


Fig. 5. Aalborg University research monitoring.

VI. Design and Implementation

A. Platform Architecture

During platform implementation task, our main goal was to implement a flexible and scalable system, easy to maintain. We developed our system using Django Framework which uses MTV pattern. MTV pattern effectively separates

communication with the database (Model) with user interface (Template) and the communication between the client and the server (View).

Using University of Thessaly as a case study we will examine the platform architecture. In Fig. 6 we see university's platform architecture. The components inside each vertical rectangular represent a department of the university.

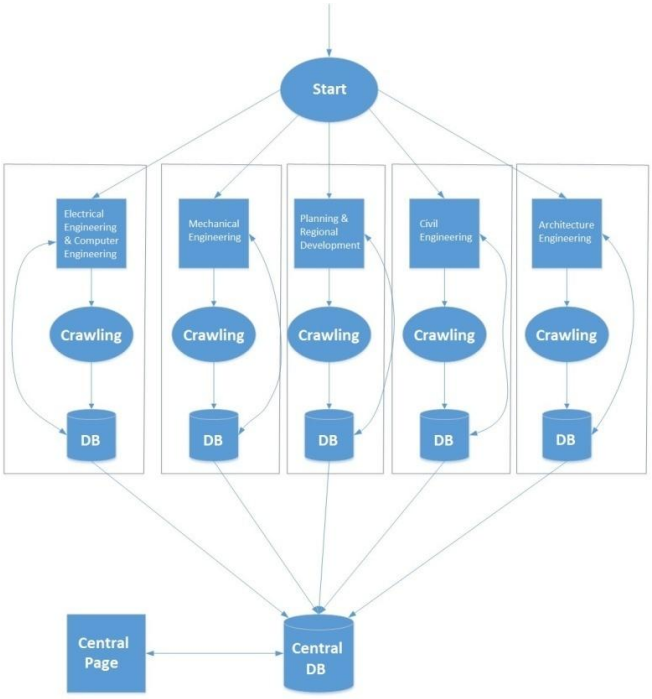


Fig. 6. Platform architecture.

The whole process starts by crawling for each department and saving the results in its own database. This process is the same for each department, since we crawl the same three sources, Scopus, IEEE Xplore and Microsoft Research Academic.

When a process of crawling ends, it stores in the central database some selected results. Again, this is done with the same way from each department. These results, include all the necessary information so we will be able to have a total image that represents the research output of the whole University by combining the output of each department.

The internal architecture of each component inside the vertical rectangular as displayed in Fig. 6, is described in Fig. 7.

As we see in Fig. 7 our system consists of 6 components. Start crawling component is responsible for sending a signal to crawler manager. This functionality needs to be automated without the need of our interference. To implement start crawling component we created a custom management command in Django. A custom management command is a command that let us register our own actions to our Django application beyond the existing commands that Django offers. In our case, we register our command that starts crawling for each author of the department, from three sources, Scopus, IEEE Xplore and Microsoft Research Academic. Additionally we installed Cron to create a Cron job which executes the script that sends the signal to crawler manager.

Crawler Manager is responsible for the functionality of crawling. Firstly, for each input received from the Start crawling component it calls every crawler. For the Scopus crawler module, it send as input the Scopus id which is found by name. Similarly, for the Microsoft crawler the input is the author's Microsoft id. For the IEEE crawler the input is only the author's name since on the IEEE each author is not represented by a unique id. Finally, after calling each crawler for every author the component work ends.

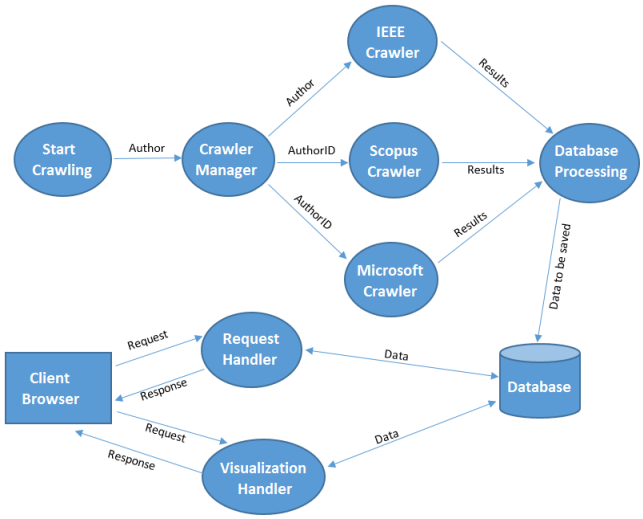


Fig. 7. Component architecture.

Crawlers where implemented using Python. Since nor Scopus, nor Microsoft, nor IEEE provide a good API for retrieving data we had to make requests and after retrieving the HTML page to parse it. For parsing we used Beautiful Soup library[2]. Each crawler returns a list of publications, each publication has the JSON structure shown in Fig. 8.

```
publication = {'title': 'pub_title', 'url': 'pub_url', 'venue':
'pub_venue',
'date': 'pub_date', 'type': 'pub_type', 'citations': 'pub_citations',
'pub_cited':'[(publication title_1),(publication url_1),
(publication title_2),(publication
url_2)]','authors_affiliations' :[(author_name_1,author_affilaitio
n_1,author_profile_url_1),( author_name_2,author_affilaition_2,
author_profile_url_2)….], 'keywords': 'pub_keywords', 'doi':
```

Fig. 8. Publication structure.

The role of database processing component is to collect all publications retrieved by crawlers and save them in the database by communicating with the corresponding models.

Request handler is responsible for business logic connecting client with server. For each client's request, the Request Handler processes this request by executing specific tasks that the request demands or querying the database if necessary, after communicating with the corresponding models. The Request Handler, after collecting all data, builds the response and then loads this response into the appropriate template which contains all the logic of data presentation.

Visualization handler is responsible for the majority of the visualizations we use in our system. More specifically, for each visualization we make a request to our database through Ajax requests to retrieve all the information which is necessary for the visualization. Next, having that information

[2] http://www.crummy.com/software/BeautifulSoup/

we can draw the visualization. With this technique, all the process of visualization is done asynchronously and the page has not to be refreshed.

B. User Interface

In the University's home page (Fig. 8) we visualize University's research activities, which arises from the collection of each department's research output. More specifically, we can see the total number of publications, citations, collaborations, countries and research areas. Also, the two bar charts visualize the number of citations that the University's publications have received in time, and the type of publications produced by the University in time. At the bottom of this page we can see University's departments. If we hover one of them we will see the number of publications and citations of that department and if we click the image we visit the department's page.

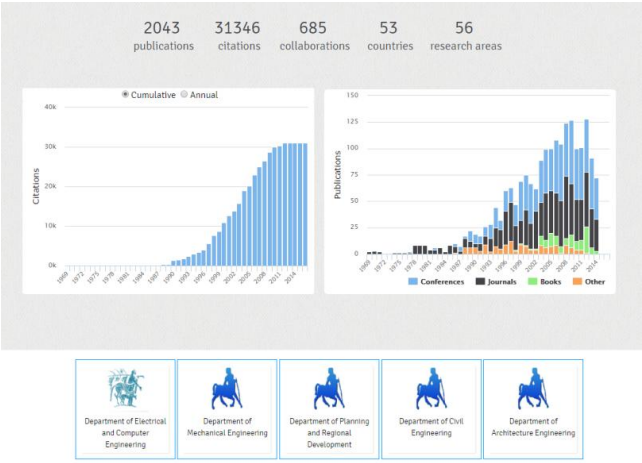


Fig. 8. University's homepage.

Fig. 9. Collaboration network.

In Fig. 9 we see the visualization of information about a department's collaboration network. On the right side we see the map with all affiliations. On the left side of the page, we see all the affiliations, with each one to consist of its name and a number referring to the times that this affiliation has collaborated with the department. Also, we can click on an affiliation and navigate to the map to see where the affiliation is located.

Another interactive visualization a viewer can use on the home page, is the number of subject areas. For this visualization we use the D3.j library. This number indicates the total number of research areas of the department's authors. If we click on it we will see the visualization of Fig. 10. By clicking the subject area will see the department's authors who are working in this area. Also, if we click on an author name we will navigate to his/her profile.

In authors' page (Fig. 11) we can see all the department's authors ordered by name. If we hover with our mouse an author's profile photo we will see an animated window sliding down containing the number of publications, the number of citations for this author and a button which navigates us to profile page. We decided to put the number of publications and the number of citations because these are the most significant bibliometric metrics that describes best a researcher.

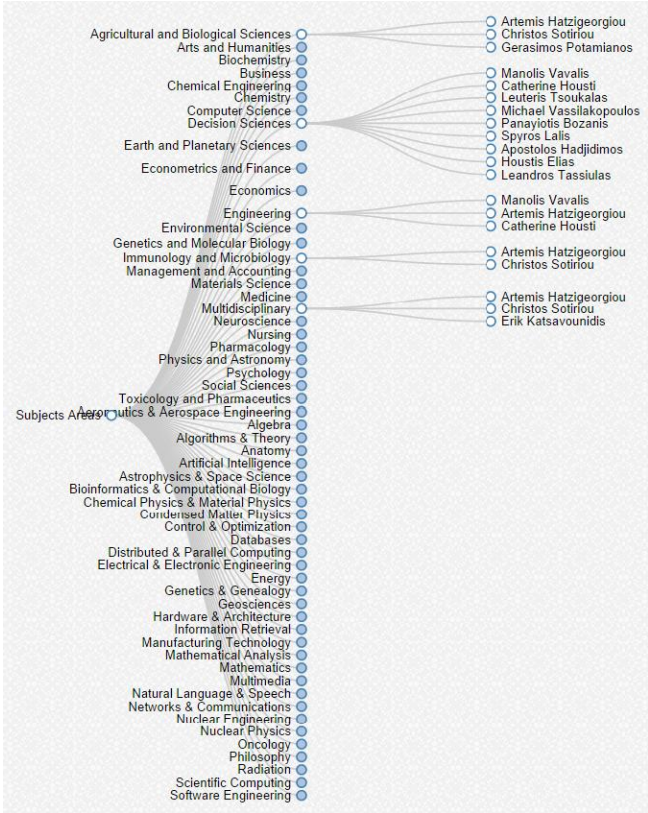


Fig. 10. Research areas.

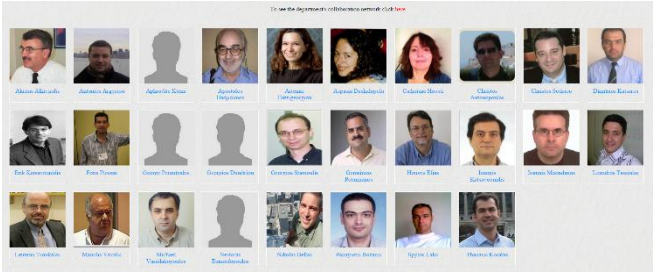


Fig. 11. Authors' page.

By selecting to see the department's collaboration network we will see an animated division to appear (Fig. 13). On this visualization, which was developed using D3.j library, we try to visualize the departments' collaborations among its researchers. As shown in Fig. 13, if we hover with our mouse a researcher name, we will see with bold red and green lines the researchers with whom he/she is collaborating.

In author's page (Fig. 12) each author is described by the following metrics:

- Publications: the total number of publications of the author.
- Citations: the total number of citations the author has received so far.
- H-index: the h-index value of the author.
- Citation impact: the citation impact value of the author. As we described in IV.A this indicates the average number of

citations per publication and is calculated as the ratio of the number of references which are listed in a certain period of time to a total number of publications of the same period. In our case, we calculate it for the past ten (10) years.

- Collaborative authors: the total number of collaborative authors the author has collaborated with so far.

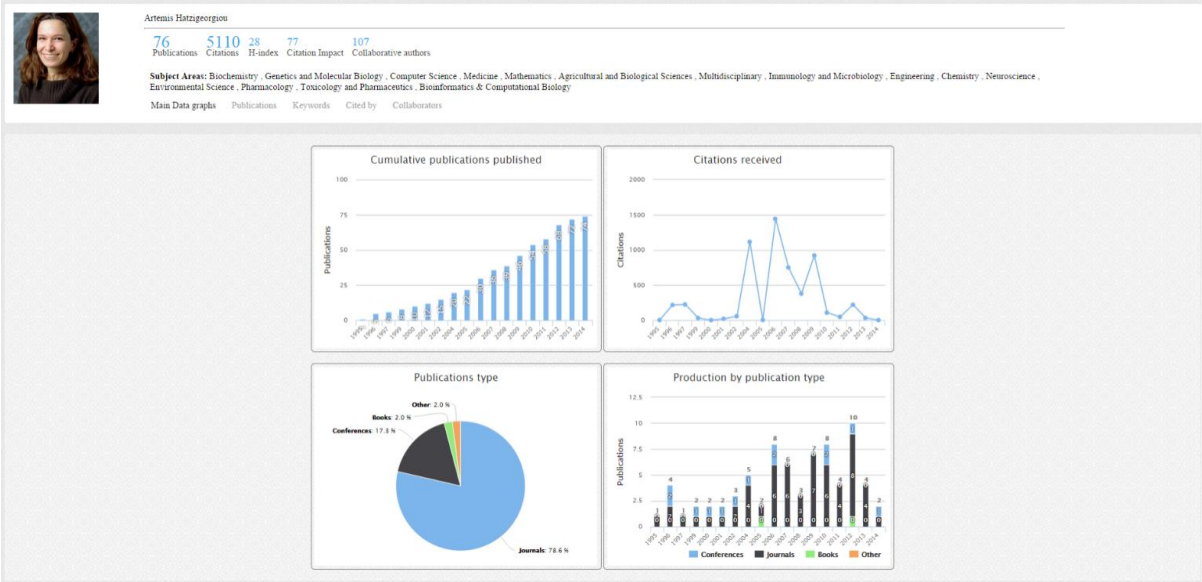


Fig. 12. Author's main page.

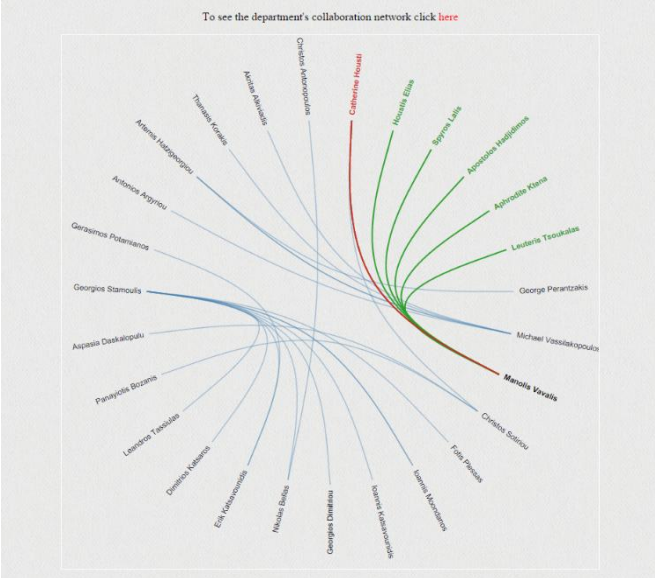


Fig. 13. Collaboration network of department's authors.

Just below the metrics we can see also all the author's subject areas. Next we can see four interactive visualizations, which are designed using Highchart library[3]:

- Cumulative publications published: This bar chart shows the author's publication production in time.
- Citations received: This line chart shows how the author's number of citations has evolved in time. Also, if we hover a specific point-year we can see the number of citations corresponding to this year.
- Publications type: This pie chart is used to show what type of publications the author published. The type can be conference, journal, book or other.
- Production by publication type: This stacked column chart shows us how the author's production type evolves in time. From this chart, an author can get valuable information about what type of publication he/she published.

C. Admin Interface

The first time we will install and deploy our web application we will notice that the content is empty since nor the department's logo nor its authors' names with their ids have been initialized. More specifically, when our crawlers start to crawl, will notice that the author name lists are empty and so they can't start crawling. So to overcome this problem, we make use of the Django admin interface. With this powerful interface each department will be able to manage all its content by adding, deleting or updating content.

For example, if we want to add another author who just came to the department we have to click the "Add" button next to the "Authors" table (Fig. 14). Then we just have to add his/her profile elements such as name, ids, profile url and a profile photo.

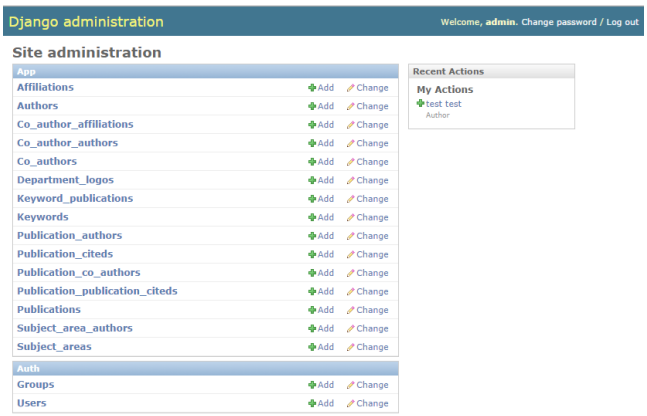


Fig. 14. Administration interface.

VII. Conclusions and Future Work

The objective of this work was the implementation of a scientific research system. A system which gathers data using

[3] http://www.highcharts.com/

web crawling techniques and visualizes information using state of the art visualization tools. The main goal is to provide organizations, institutions and researchers qualitative and quantitative evaluation metrics to monitor their scientific productivity, its evolution and their collaborations.

A public beta version of the platform is available at http://research.e-ce.uth.gr.

Our work may be extended in several directions, including (but not limited to):

- Modification of our crawlers, so as to collect data from ORCID registry using its API. By this modification the problem of duplicate registrations of researchers or publications will be solved.
- Addition of notifications functionality, which would enable users subscribe to specific events, for example receive an email when a publication from a specific author is added to the system.

REFERENCES

[1] M. A. Beyer and D. Laney, "The importance of 'big data': A definition," 21 June 2012.

[2] S. Vaidya and A. Chavan, "Data visualization in graphical format using big data," *International Journal of Scientific Engineering and Technology Research*, vol. 4, no. 14, pp. 2717-2721, 2015.

[3] T. V. Udapure, R. D. Kale, and R. C. Dharmik, "Study of web crawler and its different types," *IOSR Journal of Computer Engineering*, vol. 16, no. 1, pp. 1-5, 2014.

[4] M. Najork, "Web crawler architecture," *Encyclopedia of Database Systems*, Springer, 2009, pp. 3462-3465.

[5] R. Janbandhu, P. Dahiwale, and M. M. Raghuwanshi, "Analysis of web crawling algorithms," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 3, pp. 488-492, March 2014.

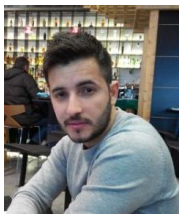
[6] J. Cho and H. Garcia-Molina, "Synchronizing a database to improve freshness," *ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, United States, 2000.

[7] A. Chadegani, H. Salehi, M. Md Yunus, H. Farhadi, M. Fooladi, M. Farhadi, and E. Ale, "A comparison between two main academic literature," *Asian Social Science*, vol. 9, no. 5, pp. 18-26, 2013.

[8] D. Rew, "SCOPUS: Another step towards seamless integration of the world's medical literature," *European Journal of Surgical Oncology*, vol. 36, no. 1, pp. 2-3, 2010.

[9] J. E. Hirsch, "An index to quantify an individual's scientific research output," National Academy of Sciences of the United States of America, 2005.

Magda Foti was born in Veria, Greece in 1985. She received B.Sc. degree and M.Sc. from the Department of Electrical and Computer Engineering, University of Thessaly, in 2009 and 2012 respectively. Currently she is a PhD candidate in the Department of Electrical and Computer Engineering, University of Thessaly. She worked as research engineer at Converge ICT Solutions & Services from 2011 to 2015. Her research interests fall into the broad areas of Power Grids, Game Theory and Machine Learning.



Elvis Papa received his B.Sc. degree from the Department of Electrical and Computer Engineering, University of Thessaly, in 2014. During 2015 he worked as a researcher at Centre for Research & Technology Hellas (CERTH). Currently he works as a software developer at babelforce, Berlin, Germany.

Manolis Vavalis serves as a professor of electrical and computer engineering at the University of Thessaly and as a senior researcher at Centre for Research & Technology Hellas. He is an expert in high performance scientific computing, information and knowledge management and Web science and technology. He has published over 100 technical articles and acted as a reviewer and consultant for public and private organizations.