

Documentations for DIA2

The DIA2 team

Documentations for Deep Insights Anytime, Anywhere (DIA2)

(Based on the iKNEER Prototype)

PI: Dr. Krishna Madhavan (Overall Project Lead and R&D Architect)

Co-Pis: Drs. Aditya Johri, Mihaela Vorvoreanu, Alan Wang, Niklas Elmquist¹

Supporting authors: Hanjun Xian (Ph.D. scholar, Software Group Leader), Xin Chen (Ph.D. Scholar, Research on Personalization), Sanjeev Singh (MS, Database Engine Optimization)

Delivered: Monday, January 9, 2012

¹ This report characterizes the system engineered as part of iKNEER and the authorship here does not reflect the actual authorship in NSF Grant-1123108. We list contributions to the technical pieces only.

Executive Summary

Purpose of this report: This report aims to summarize the design, implementation, and deployment of DIA2 from both technical and non-technical perspectives. It includes the conceptual design and the software architecture of DIA2. It also contains detailed explanations of major components in the data management level, the computational engine level, and the interface level. We also present technical elements such as class diagrams and Web APIs in this report.

Intended readers: Researchers and program officers who are interested in DIA2 may get a comprehensive understanding of what the current version of DIA2 is capable of doing and find an effective way to explore the large repository of knowledge products. Developers who plan to incorporate DIA2 into their projects or extend DIA2 can find technical details in this report to help them quickly understand the underlying implementations of DIA2 and the available web services.

Organization of this Report

Chapter 1 describes the hardware and software infrastructure that DIA2 is based on and the fundamental workflow of DIA2. DIA2 implementations comprise three major levels: data management, computational engine, and web UI/API. Chapter 2 elaborates the data management modules by presenting the database design, the data acquisition process, and the name disambiguation system. Chapter 3 discusses the computational engine with particular focuses on how to produce keywords/keyphrases to documents without author-supplied keywords and how to design the social networking components. Chapter 4 introduces major UI features and possible scenarios of use. It also includes a list of available web services offered by DIA2 for other researchers and developers to query the DIA2 database. The appendix contains the technical documentations for DIA2.

Deployment and Workflow

Documentations for DIA2

- Deployment & Workflow
 - Data Management
 - Computational Engine
 - Web UI and API

Hardware and Software Infrastructure

Server: 10 Mac Mini (Intel Core 2 Duo 2.66GHz, 4GB RAM, 1TB HDD)

Operating System: Mac OS X Server 10.6

Service/Software: Apache 2.2.14, PHP 5.3.4, MySQL 5.0, and Java 1.6

Deployment

Table 1. Server usage and deployment

Function	Development	Production	Production backup
Web Server	1. Apache, PHP	4	7
Computation Server	2. Java, PHP	5	8
Data Server	3. MySQL	6	9



Figure 1. Scaffolding infrastructures for DIA2

Nine servers are divided into three groups, each of which contains a web server, a computation server, and a data server. One server group is used for developing and testing new features, whereas the other two are publicly available to the users. The corresponding servers in each group have been configured with the same software and settings. The tenth server hosts the team and project website for DIA2. Figure 1 shows the hardware scaffoldings for our servers in the machine room.

Workflow

In general, DIA2 works as follows:

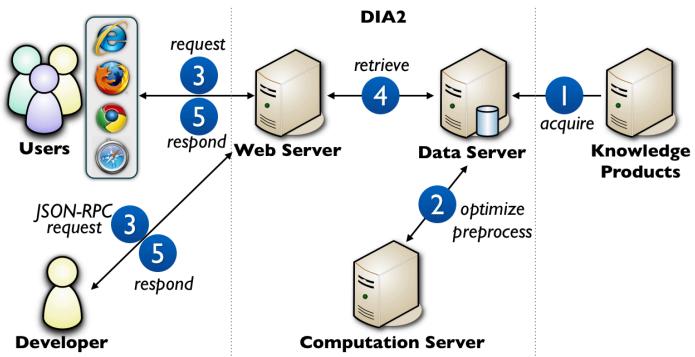


Figure 2. DIA2 architecture

- (1) The data server collects knowledge products such as academic articles and grant proposals from a variety of sources periodically using well-known crawling strategies.
- (2) The computation server optimize the data to cache statistics and visualizations. It also analyzes the archived data to generate new relationship.
- (3) The web server processes user operations on the web site or requests via JSON-RPC.
- (4) The web server authenticate requesters' identities and retrieves results from the database.
- (5) The result will be then represented in a visual and interactive form and a portion of the page will be updated to reflect the changes.

Data Management

Documentations for DIA2

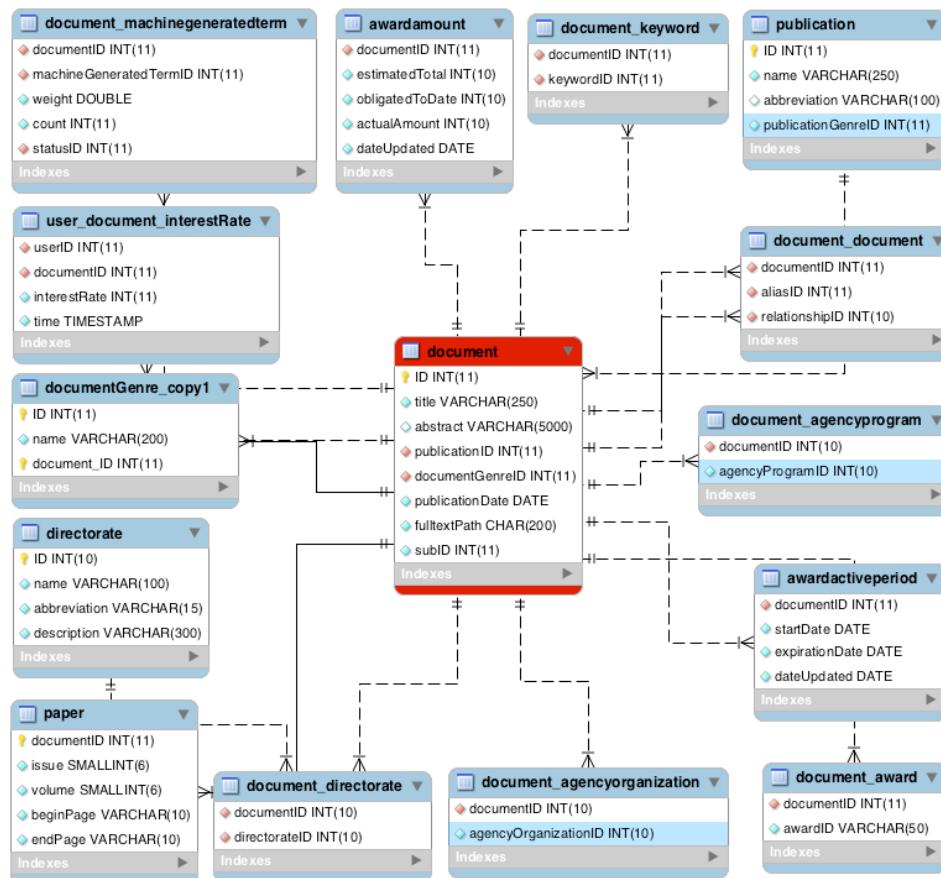
Deployment & Workflow
▪ Data Management
Computational Engine
Web UI and API

Database Entity-Relationship Diagram (ERD)

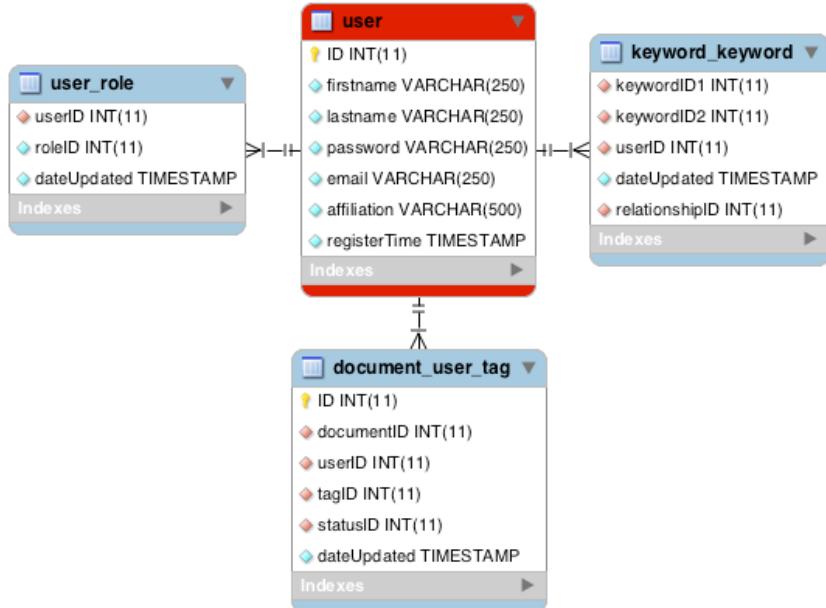
Major modules in the database

For more clear representation, we divide all tables into modules and introduce each of them in more details in the following discussion. These modules are: document, user, person, keyword, and address. A complete ERD can be found in Appendix C.

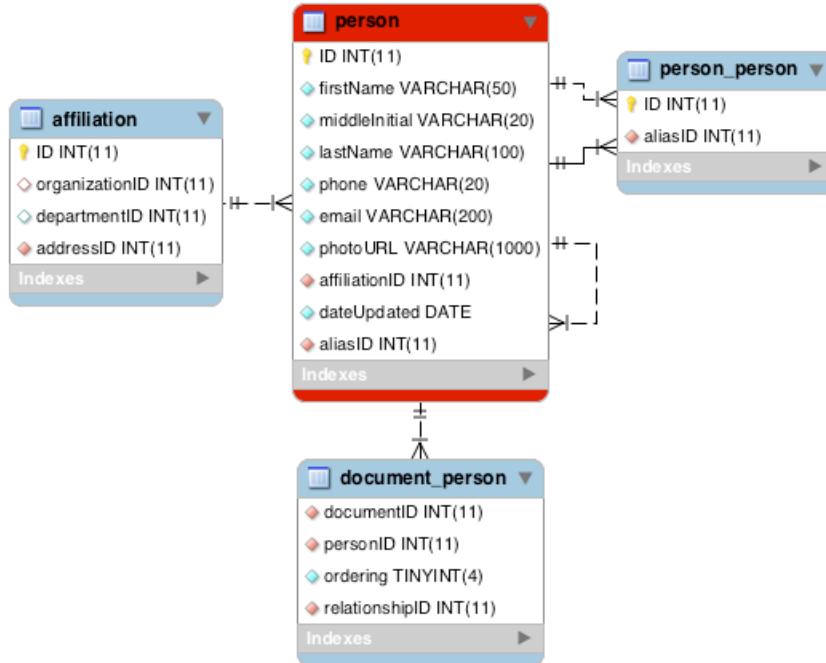
Detail ERD for the document module



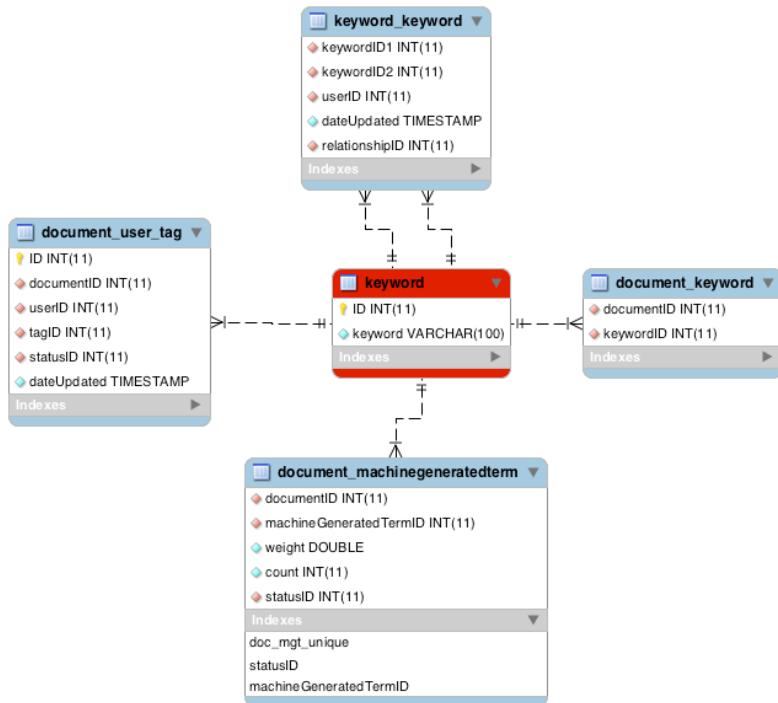
Detail ERD for the user module



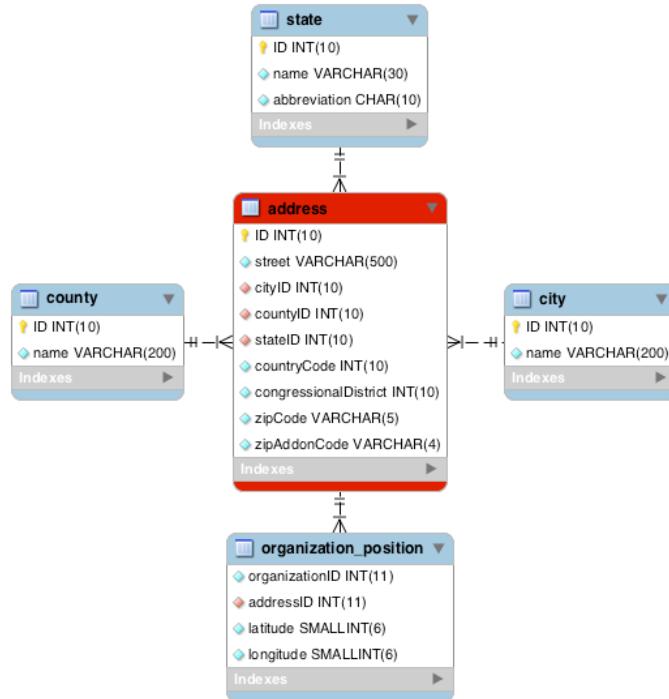
Detail ERD for the person module



Detail ERD for the keyword module



Detail ERD for Address module



Data Acquisition

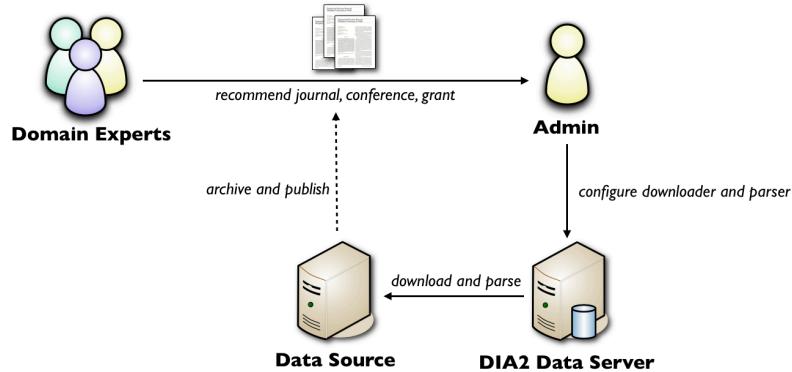


Figure 3. Acquiring knowledge products into DIA2

As described in Figure 3, first, we rely on domain experts to identify relevant knowledge products such as a certain journal, conference, or grant. The DIA2 administrator then configures the downloader and parser and sets up a web crawler to periodically retrieve bibliographic data from the data sources that host the target publications. We have developed various adapters to read metadata files of different formats such as BibTeX, EndNote, and XML. Finally, the downloaded data is archived into the DIA2 data server. Table 2 lists the current DIA2 archives (as of 1/7/2012).

Table 2. Availability of knowledge products archived by DIA2

Publication venue	Type	Num. of documents	Available year range
Advances in Engineering Education	Journal	20	2007-2009
Australasian Journal of Engineering Education	Journal	50	2000-2009
Computer Applications in Engineering Education	Journal	505	1997-2011
EDUCON, IEEE	Conference	461	2010-2011
IEEE Transactions on Education	Journal	3331	1963-2011
Engineering Education	Journal	2292	1969-1991
Engineering Science and Education Journal	Journal	474	1992-2002
European Journal of Engineering Education	Journal	1953	1975-2011
Frontiers in Education Conference	Conference	7082	1982-2010

Publication venue	Type	Num. of documents	Available year range
Global Journal of Engineering Education	Journal	296	2000-2007
International Journal of Continuing Engineering Education and Life-Long Learning	Journal	185	2004-2009
International Journal of Electrical Engineering Education	Journal	1491	1969-2010
International Journal of Engineering Education	Journal	1563	1997-2011
Journal of Chemical Education	Journal	1796	2005-2008
Journal of Engineering Education	Journal	680	1968-2011
Journal of Professional Issues in Engineering Education and Practice	Journal	842	1994-2011
National Science Foundation	Grant proposal	123054	2000-2011

Name Disambiguation

Name ambiguity is a common problem in bibliographic databases. Figure 4 illustrates a typical author name ambiguity example where a researcher has four name variations in different publications. Other ambiguity issues include article title ambiguity, institution name ambiguity, and publication name ambiguity. DIA2 disambiguates entity names by offering users recommendations based on the similarity of information associated with each name. In Figure 4, DIA2 determines the similarity between names based on relevant information such as co-authorship and research area. This supervised disambiguation approach is highly customizable and can be undone and revised easily. Names that are recognized as representing the same entity are viewed as a single unit in computing statistics and visualizations, as shown in Figure 5.

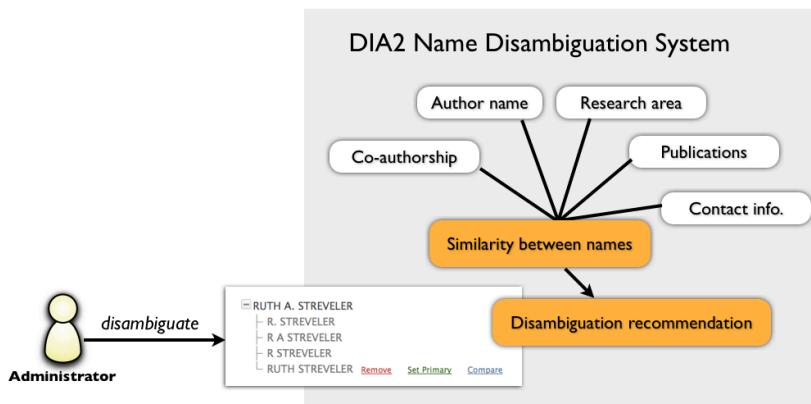


Figure 4. A name disambiguation example in DIA2

Don L. Millard author, PI, co-PI, program officer

Also known as [D. Millard](#), [D L Millard](#), [D Millard](#)

Figure 5. Ambiguous names are grouped and recognized as representing one individual

Computational Engine

Documentations for DIA2

- Deployment & Workflow
- Data Management
- Computational Engine
- Web UI and API

Keyphrase Extraction

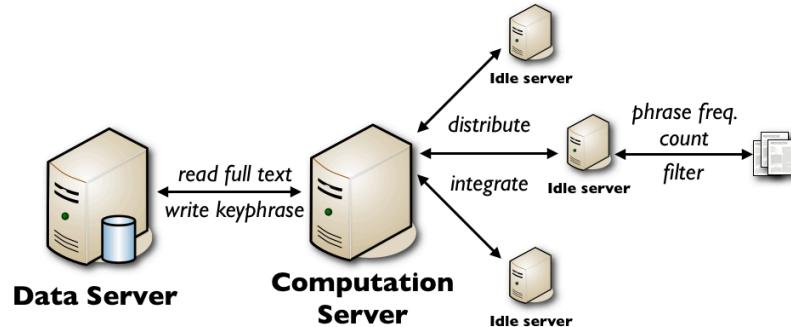


Figure 6. The distributed algorithm for keyphrase extraction

A large number of articles have no author-supplied keywords. To concisely annotate each knowledge product, DIA2 generates a collection of weighted keyphrases using an algorithm called GenEx². Figure 6 demonstrates the keyphrase extraction process: First, the computation server retrieves full text of all documents to be processed and distributes them to idle servers. Each server then counts the occurrence of phrases (up to 4 consecutive words) in each assigned document and the results are aggregated to produce a global phrase-count map in the computation server. Based on how often a phrase appears across all documents, each server filters stopwords such as *the, of, is, that* from each document and compute the weight for each phrase. Weighted phrases are finally inserted into the data server. We are also working on incorporating the topic modeling algorithm developed by the Virginia Tech team to provide better categorization of keywords and keyphrases.

User-Generated Content

Tag, Rate and Comment

Author-provided keywords and machine extracted keywords may not be descriptive enough to the documents. To fully utilize human intelligence and expertise, the DIA2 system enables users to provide keywords to documents, namely tags. Besides, the users can also specify interest ratings and write comments for the documents, as illustrated in Figure 7. These functions are at the testing phase and have not been made publicly available yet.

² Turney, P. D. (2000). Learning Algorithms for Keyphrase Extraction. *Information Retrieval*, 2(4), 303-336.

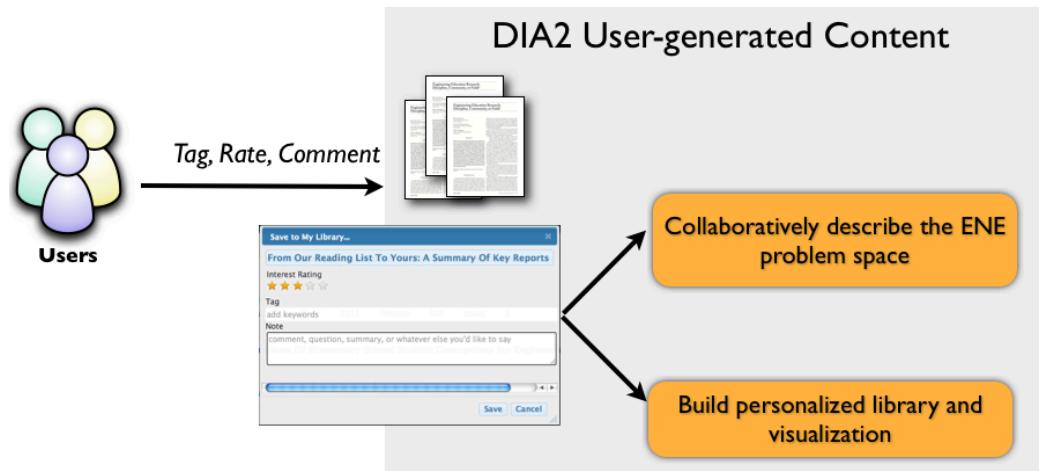


Figure 7. User tag, rate and comment documents

Keyword Grouping

Many keywords may have the conceptually similar meaning and therefore should be viewed as one entity. For example, *problem-based learning* and *PBL* are usually considered as identical, whereas *project-based learning* and *problem-based learning* are not. It is difficult for the machine to make this kind of judgment because it requires domain knowledge. The DIA2 system allows users to group keywords with similar meaning together and the aggregated user input helps produce a keyword folksonomy, as illustrated in Figure 8. This function is not published yet.

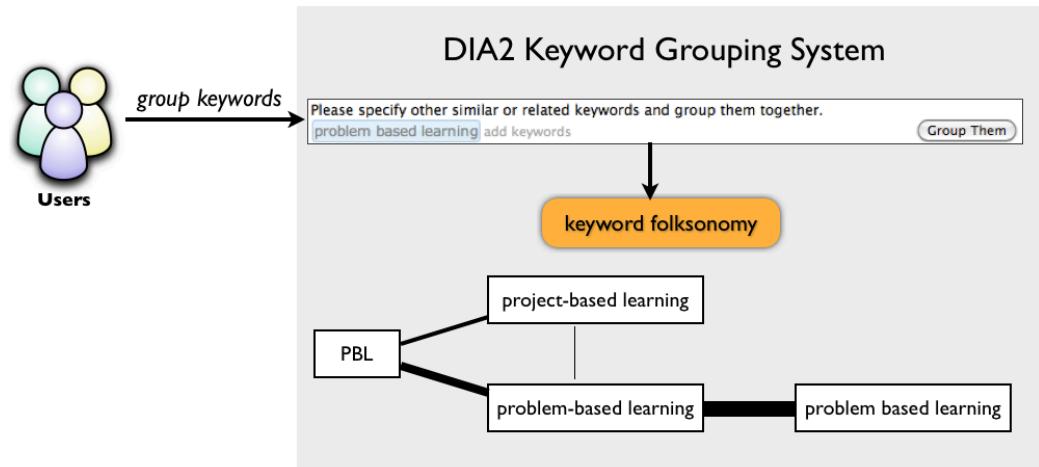


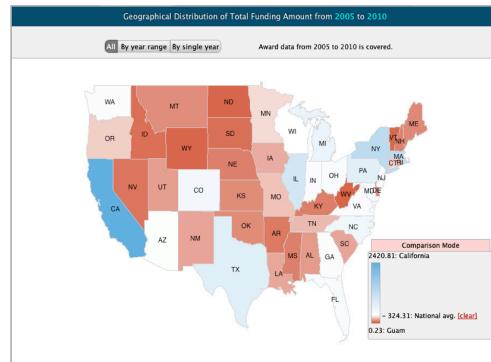
Figure 8. The keyword grouping feature

Web UI and API

Documentations for DIA2

- Deployment & Workflow
- Data Management
- Computational Engine
- Web UI and API

Major Features on the Web Interface



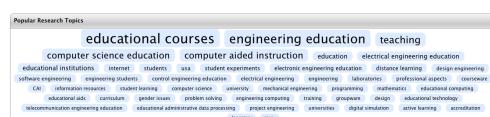
GeoMap

Function: Demonstrate the geographical distribution of a given variable, e.g. number of NSF awards, total amount of awards, and average amount per award.

User interaction: Users can hover each state to see detailed statistics about a state, control what color represents, and change time range.

Dependent infrastructure: protovis

Popular Research Topics



Function: Show the popular research topics based on the keywords/keyphrases of all archived documents.

User interaction: Users can click a topic to view its profile.

Dependent infrastructure: None

Statistical Overview

Num. of academic papers: 71
 Num. of grant proposals: 24
 Num. of authors: 54
 Num. of distinct Pls: 23
 Total amount: \$11.45m
 Avg. amount per award: \$477k

Archive Overview

Function: Display an overview of a set of documents meeting a specified condition such as articles related to *problem-based learning*

User interaction: None

Dependent infrastructure: None

You are searching for problem-based learning (1,21 seconds)					
Displaying 95 documents					
Article: Problem-based Learning In An Industrial Computers Course					
Authors: Julia Avel-Romero Publication: International Journal Of Engineering Education Year: 2011 Volume: 27 Issue: 1 Keyword: problem-based-learning , industrial computer , computers , pl , lessons , pedagogical , problem-based , evaluations , during , theoretical , applied , projects					
Article: Problem-based Learning In Wind Energy Using Virtual And Real Setsups					
Authors: D'Santos-martins , J.Almeida-martinez , J.Say-garcia Carrasco , S.Araújo Publication: Education, IEEE Transactions On Year: 2011 Volume: -- Issue: 99 Keyword: virtual wind-energy , tools -real , problem-based , real , wind , results , both , wind turbine					
Article: A Cochlear Implant Signal Processing Lab: Exploration Of A Problem-based Learning Exercise					
Authors: P.T.Bhatt , J.H.Medlin Publication: Education, IEEE Transactions On Year: 2011 Volume: -- Issue: 99 Keyword: signal processing , lab , exploratory , decomposition , PBL , problem-based , exercise , tested , discussed , cochlear implant					
Proceedings Paper: Special Session – Not All Problems Are Created Equal: From Problem-based Learning Theory To Research On Complex Problem Solving And Implications For The Engineering Classroom					
Authors: O.Pereira , H.Warren , R.Kayler , J.Russell , R.Anderson Publication: Frontiers In Education Conference Year: 2010 Volume: -- Issue: -- Keyword: engineering education , problem solving , undergraduate curriculum , educational courses , engineering classroom , complex problem solving , problem based					

Publication List

Function: Display a list of publications given a certain condition.

User interaction: Users can click on the author names, keywords, and publication venues to see their profile pages.

Dependent infrastructure: DataTables

Krishna Madhavan PI

Email cm@purdue.edu

Affiliation [Purdue University](#)

Num. of journal papers: 0

Num. of conference proceedings papers: 0

Num. of awards: 4

Award amount received as PI: \$ 1,981,666

Award amount received as co-PI: \$ 0

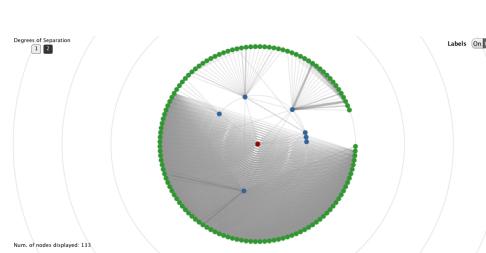
Award amount received as PI and co-PI: \$ 1,981,666

Scholar Overview

Function: Overview a scholar's publication production and funding status.

User interaction: None

Dependent infrastructure: None



Egocentric Collaboration Network

Function: Show a researcher's collaboration network by degree of separation.

User interaction: Users can enable/disable collaborators' names, change the degree of separation displayed, and navigate to a collaborator's profile page.

Dependent infrastructure: Javascript InfoVis Toolkit

Name	Num. of publications
Lisa McNair	12
Naren Ramakrishnan	12
Marie Paretti	11
P.K. Imbrie	9
Aditya Johri	8
Vinod Lohani	8
O Pierakkos	8
Mary Besterfield-Sacre	8
Gary Downey	7
R Anderson	7

Showing 1 to 10 of 95 entries

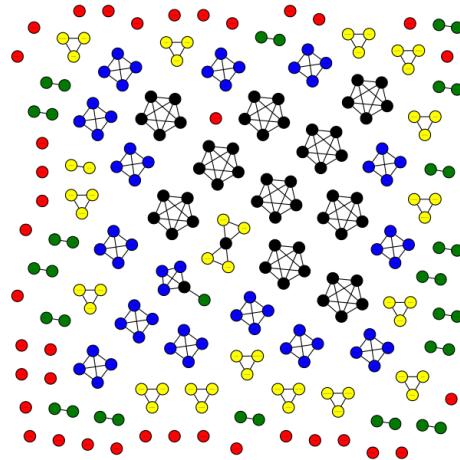
First Previous [1](#) [2](#) [3](#) [4](#) [5](#) Next Last

Collaborator List

Function: Show a researcher's collaborators.

User interaction: Users can sort and search the list and navigate to a collaborator's profile page.

Dependent infrastructure: DataTables

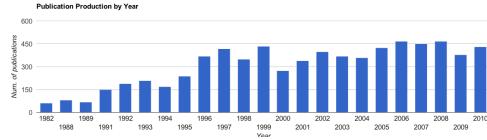


Awardees' Collaboration Network

Function: Show the collaboration network among NSF awardees.

User interaction: Users can click to navigate to a collaborator's profile page.

Dependent infrastructure: Graphviz



Publication Production per Year

Function: Show the publication production of a journal/conference, an author, or a keyword.

User interaction: Users can hover and view the exact number in each year.

Dependent infrastructure: Google Chart API

You may be interested in these authors		
Name	Num. of publications	Search:
Mary Anderson-rowland	38	
Karl Smith	24	
Barbara M. Olds	23	
Jeffrey Froyd	22	
Cynthia Atman	21	
Larry Shuman	21	
Ronald Miller	21	
M M Ciampi	20	
Mary Besterfield-Sacre	20	
D D Budny	19	

Showing 1 to 10 of 5,990 entries
[First](#) [Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#) [Last](#)

Popular Author List

Function: Show the most productive researchers in a publication venue or in a certain research area.

User interaction: Users can click to view a researcher's profile.

Dependent infrastructure: DataTables

Name	Num. of awards	Total award amount / mill. \$
BIO	15756	6267
CSE	14375	5258
EHR	10985	9023
ENG	21454	6347
GEO	16060	8148
MPS	25671	11994
O/D	5063	2607
SBE	11800	2217
OPP	3107	2040
NCO	7	26
BFA	64	8
IRM	94	29
NSB	37	13
NNCO	1	0

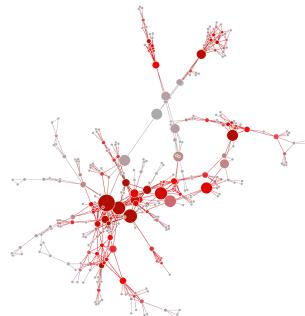
NSF Directorates

Function: Show the list of NSF directorates and their number of awards and total amount of awards.

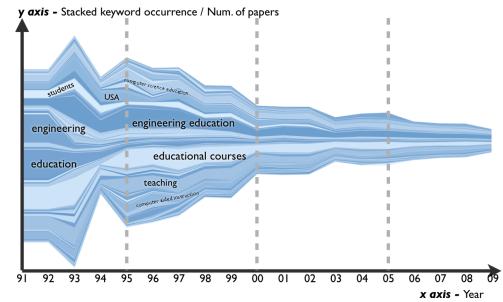
User interaction: Users can click to view a directorate's profile.

Dependent infrastructure: Google Chart API

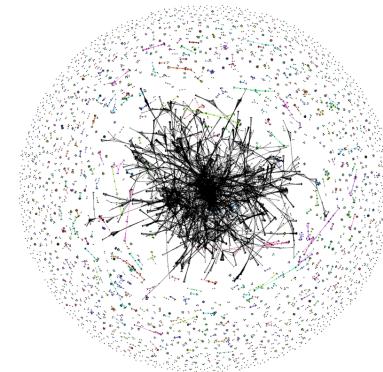
Other Offline Features



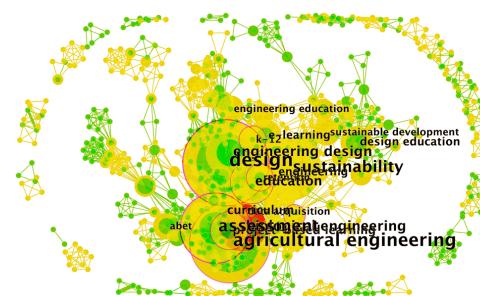
The largest collaboration network among CCLI awardees
(2000-2011)



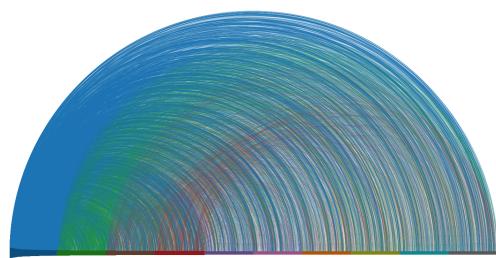
The evolution of research topics in FIE (1991-2009)



The collaboration network among FIE authors (1991-2009)



Popular research topics in IJEE (2004-2007)



The intensity of collaboration between productive scholars
and less productive ones



Geographical distribution across congressional districts

Usage Scenarios

The usage scenarios below demonstrate how DIA2 may be used to address different users' needs.

Scenario 1: Sarah is a first-year PhD student in Engineering Education. She wants to explore the area related to *mentoring*. She would like to find the trends of research around this topic, grants information on this topic and related papers on *mentoring* to read.

Solution: Sarah searches for *mentoring* from the search box at the home page. She gets a list of grants and papers including the keyword *mentoring*, and a statistical overview shown below. By clicking the keyword *mentoring*, she goes to the keyword profile page, where she finds the trends in grants and publications are both growing in recent years. Most published papers around this topic are from the Frontier in Education conference, and there are many grants from the NSF. She also gets the list of authors who actively publish in this area.



Figure 9. Statistical overview of search result from *mentoring* and author list around the topic *mentoring*

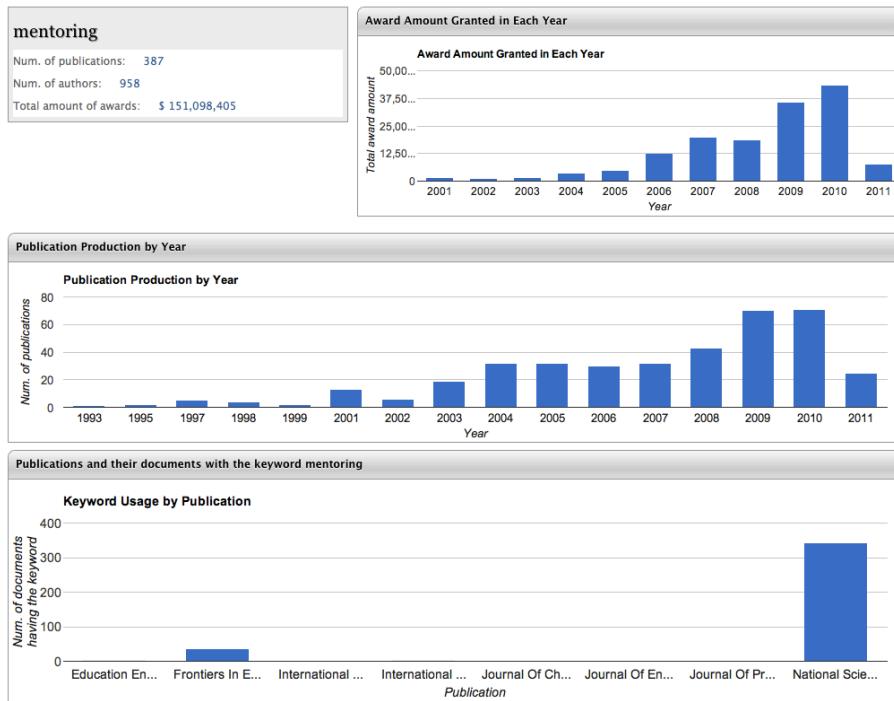


Figure 10. Trends of mentoring on the mentoring keyword profile page

Scenario 2: Mary is an engineering education faculty member. She wants to look at her own profile on DIA2 and look at her collaborators and further identify potential collaborators.

Solution: Mary searches her own name from the search box and goes to her own profile page. She gets her own collaboration network. By clicking the dots on this network graph, she goes to her collaborators' profile pages, where she can find research interests and publications of their collaborators. Thus she can identify potential collaborators.

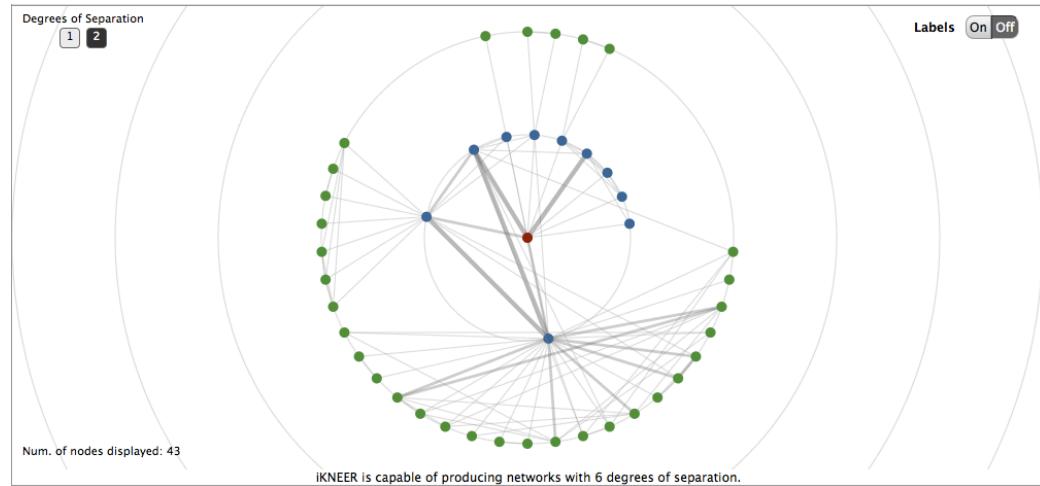


Figure 11. Egocentric collaboration network of the author

Scenario 3: Peter is a NSF program officer. He wants to look at the geographic view of grant distribution of the Education & Human Resources directorate. He is also interested to see the collaboration network among PIs and Co-PIs who have awards approved by him.

Solution: From the homepage publication list, he clicks the last entry: National Science Foundation. He gets the geographic view of all NSF awards archived on DIA2. From the directorate list on the right, he clicks EHR (stands for Education & Human Resources), so he gets the geographic view of grants under this directorate. Clicking “compare” at the right bottom, he can compare the grant amount of each state with the national average amount. By searching his own name, he goes to his own program officer profile page, where he can see the collaboration network of the PIs and Co-PIs who have awards approved by him.

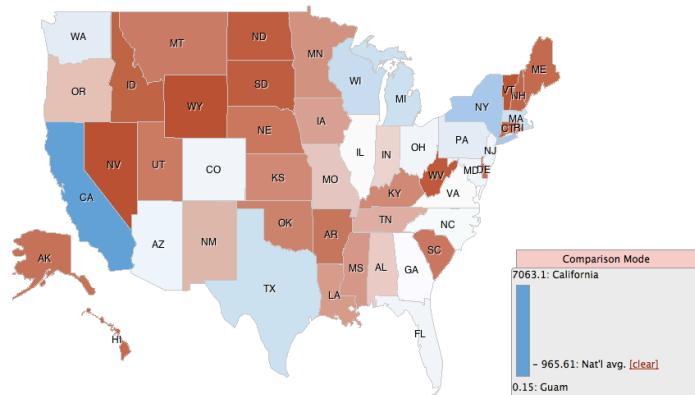


Figure 12. Geographic view of grant distribution in the compare mode

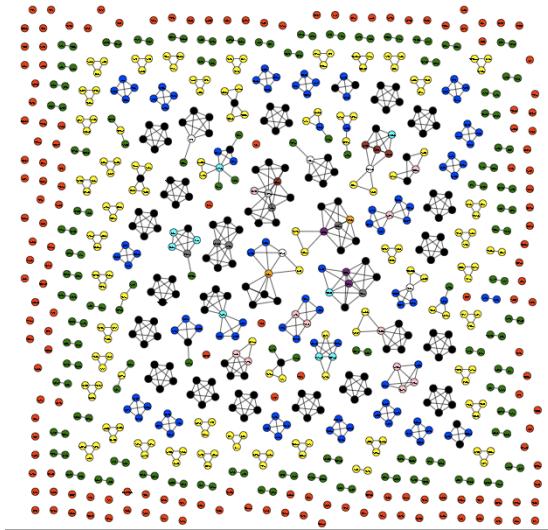


Figure 13. Collaboration network of the PIs and Co-PIs who have received awards approved by the program officer

Scenario 4: Jim is a PhD student in Engineering Education. He wants to look at his advisor's profile on DIA2, and find who has collaborated with his advisor, so he may be able to ask his advisor to introduce him to those other professors.

Solution: Similar as in scenario 2, Jim searches his advisor's name, goes to the profile page of his advisor, and finds the egocentric collaboration network of his advisor. From this network graph, he is able to find the collaborators of his advisors. If he is interested in some of them, he can ask his advisor to introduce him to them, thus he can start to build a professional network for his research career.

Web Client Implementations

We implement the client end using HTML and Javascript designed in an Object-Oriented manner. The visual elements on the page are highly configurable and reusable. For example, on a person's profile page like Figure 14, the overview, research interests, and the search dialog are controlled by different *Window* and *WindowContainer* objects. Each of these Javascript classes can process user interaction and invoke the corresponding JSON-RPC to retrieve data and update the visual representation.

Figure 14. Part of a person's profile page on DIA2

The following Javascript classes are the major components in the client end. A detailed class diagram can be found in Appendix A.

Table 3. Primary Javascript classes implemented in the client end

Javascript class	Function
AuthorListWindowContainer	Display a sortable table of authors and their number of publications
AwardeeListWindowContainer	Display a sortable table of NSF awardees
ProgramOfficerListWindowContainer	Display a sortable table of program officers
CollaboratorsWindowContainer	Display a sortable table of authors that collaborate with the specified author
DirectorateListWindowContainer	Display a sortable table of NSF directorates and their number of awards and total amount
PublicationListWindowContainer	Display a sortable table of archived publications, their number of publications, and their available year range
PersonEgoCollaborationNetworkWindowContainer	Visualize an interactive author's egocentric network with specified degree of separation
AwardAmountByYearWindowContainer	Show the yearly distribution of NSF award amount
PublicationByYearWindowContainer	Show the yearly distribution of academic publications
GeoWindowContainer	Show the geographical distribution of a variable such as number of awards across states in the U.S.
SearchWindowContainer	Draw the search dialog
TagCloudWindowContainer	Draw a word cloud to represent a scholar's research interests
POTagCloudWindowContainer	Draw a word cloud to represent a program officer's funding area
SearchResultWindowContainer	Show a list of publications with titles, abstracts, authors, publication date, and other metadata
PersonCollaborationNetworkWindowContainer	Show a collaboration network in a certain context

JSON-RPC

To open our database to other researchers, we provide remote procedure calls formatted in JSON (JSON-RPC) for accessing our data via the programming interface. In total we have 74 web services available to fetch the data from our database. For example, a developer can pass the JSON packet to request overall details of an author. Other procedure calls include computing co-author networks, keyword trends, and papers written by a given author. The table below lists part of the JSON-RPC available on DIA2 and more technical details can be found in Appendix B.

Table 4. JSON-RPC methods offered by DIA2

Method	Function
getAuthorEgoNetwork	Get data to build author collaboration network.

Method	Function
getAuthorList	Get detail of authors associated with a keyword.
getAuthorOverview	Get the details of a specific author.
getAwardAmountByYear	Get award amount & year associated to a keyword.
getAwardeeList	Get details of a program officer.
getContactInfo	Get contact info of a person.
getKeywordOverview	Get number of publication, award amount, authors against a keyword.
getNodesWithin3Degrees	Get info of collaborator.
getOverview	Get overview of the knowledge base.
getPersonCollaborationNetwork	Get data to build collaboration network.
getPOOview	Get award details of a person.
getProgramOfficerListSP	Get info of officers and award against a keyword.
getPublicationDistribution	Get publication name and keyword count against a keyword
getPublications	Get info of all the publications.
getTagCount	Get top 50 tags.
getYearlyDistribution	Get number of publication and year against a keyword.

Appendix

A. Documentations for the Web Client (Javascript)

The following is the class diagram for the web client (Javascript). *WindowContainer* (colored in grey) is the base class for all other window containers (colored in red). So all other window containers “extends” the *WindowContainer* class. *PlainWindow* (in white) is the base class for all windows. All other window classes (colored in yellow) extend *PlainWindow*. Two other basic window types derived from *PlainWindow* are *TableWindow* and *ListWindow*. Every window container usually contains one or more different types of windows, shown using the diamond-shape sign in the class diagram.

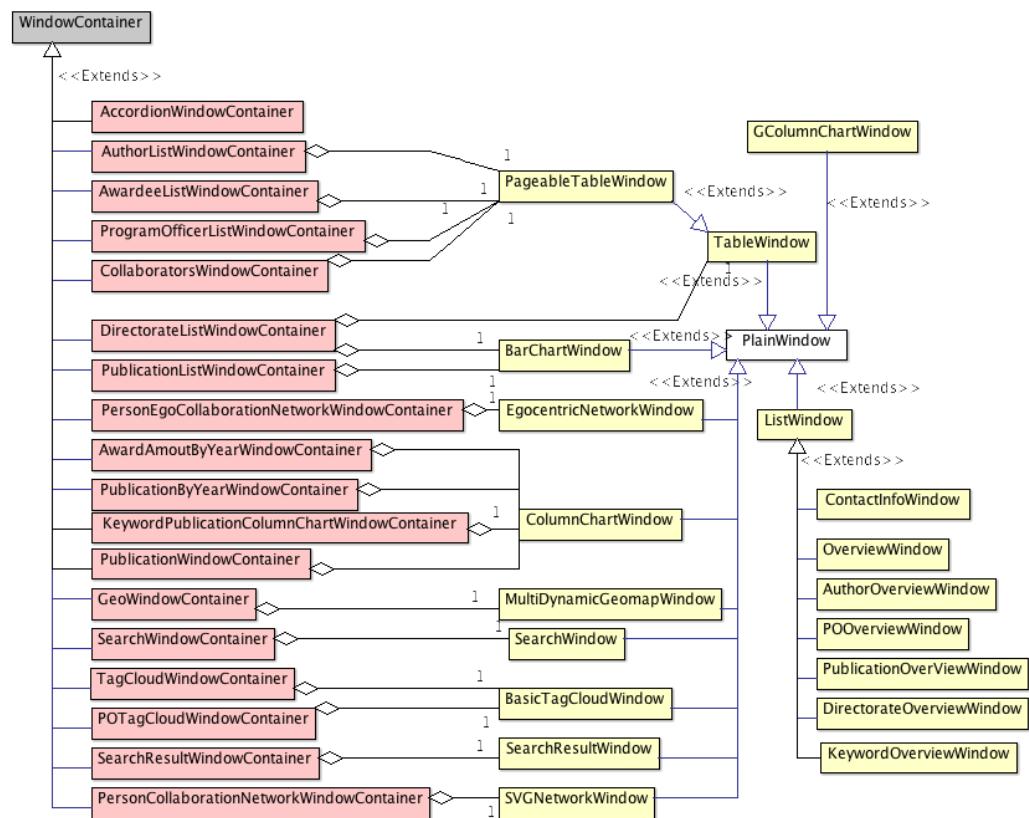


Figure 15. DIA2 web client class diagram

The following shows the windows or window containers the profile pages use on the DIA2 web site. For example, the publication profile page (publication.php) uses *PublicationOverviewWindow*, *SearchWindowContainer*, *PublicationByYearWindowContainer*, *TagCloudWindowContainer*, *AuthorListWindowContainer* and *SearchResultWindowContainer*. These are examples of the most important profile pages, and do not include every page.

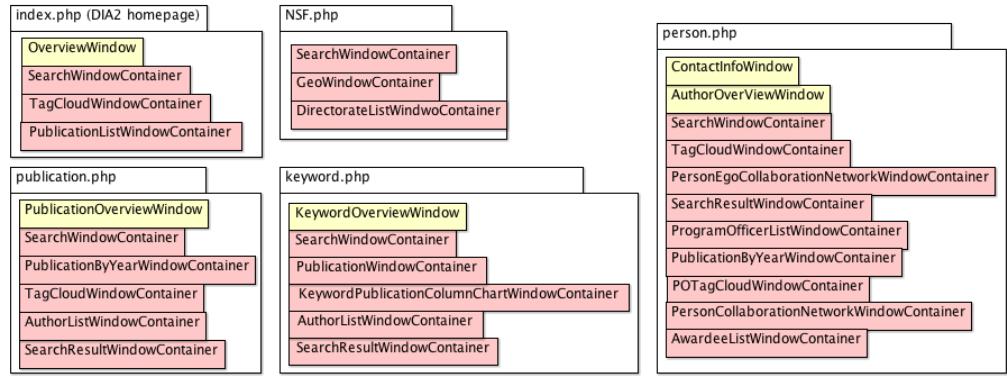


Figure 16. Profile web pages and their corresponding window and window container objects

B. Documentations for the API (JSON-RPC implemented in PHP)

The following list gives an in-depth detail of web-services that can be evoked remotely to get useful information.

► getAuthorEgoNetwork

Description: Get data to build author collaboration network. Pass the person_id and get the adjacent edges for the network.

Usage:

1. Request

```
{"params":{"personID":122937,"depth":2}, "method":"getAuthorEgoNetwork"}
```

2. Response

```
{"exeTime":0.083745002746582, "result":{"stats":{"depth":2}, "status":"OK", "data":[{"name":"Eduardo Sontag", "weight":1, "root":false, "id":50667, "adjacencies":[{"id":50668, "count":1}, {"id":33698, "count":1}, {"id":37156, "count":1}, {"id":78552, "count":1}, {"id":82708, "count":1}]}], "id":null}
```

► getAuthorList

Description: Get detail of all the authors associated with a keyword. Pass the keyword_id in the request and get author specific information.

Usage:

1. Request

```
{"params":{"keywordID":2051}, "method":"getAuthorList"}
```

2. Response

```
{"exeTime":0.019109010696411, "result":{"data":[{"personID":39058, "numPublications":3, "name":"Richard Finno"}, ...], "status":"OK"}, "id":null}
```

► getAuthorOverview

Description: Get the details of a specific author. Pass person_id to get details : number of conference papers, number of awards etc. .

Usage:

1. Request

```
{"method":"getAuthorOverview", "params":{"personID":82708}}
```

2. Response

```
{"exeTime":0.010731935501099, "result":{"data":{"numJournalPapers":0, "numConferencePapers":0, "numAwards":2, "awardAmountAsPI":349149, "awardAmountAsCoPI":0, "awardAmountAsPIInCoPI":349149}, "status":"OK"}, "id":null}
```

► getAwardAmountByYear

Description: Get award amount & year associated to a keyword. Pass the keyword_id and get yearly amount information.

Usage:

1. Request

```
{"params":{"keywordID":2051}, "method":"getAwardAmountByYear"}
```

2. Response

```
{"exeTime":0.0046341419219971, "result":{"data":[{"year":2001, "amount":989361}], "status":"OK"}, "id":null}
```

► getAwardeeList

Description: Get details of a program officer. Pass program officer id and PO name and amount information.

Usage:

1. Request

```
{"params":{"POID":33305,"method":"getAwardeeList"}}
```

2. Response

```
{"exeTime":0.72132897377014,"result":{"data":[{"id":45900,"amount":3208731,"name":"Xiaochun Li"},...],"status":"OK"},"id":null}
```

► getContactInfo

Description: Get contact info of a person. Pass person_id and get the complete contact information.

Usage:

1. Request

```
{"method":"getContactInfo","params":{"personID":33305}}
```

2. Response

```
{"exeTime":0.014854907989502,"result":{"data":{"name": {"firstName":"George", "lastName":"Hazelrigg", "middleInitial":"A."}, "contactInfo":{"phone":"","email":""}, "affiliation": {"ID":"1", "organization":"NVA"}, "aliasNames":null, "role":["programOfficer"]}, "status":"OK"}, "id":null}
```

► getKeywordOverview

Description: Get number of publication, award amount, authors against a keyword.

Usage:

1. Request

```
{"method":"getKeywordOverview","params":{"keywordID":2051}}
```

2. Response

```
{"exeTime":0.011595964431763,"result":{"data":{"name":"finite element", "numPublications":182, "awardAmount": 30981535, "numAuthors":294}, "status":"OK"}, "id":null}
```

► getNodeWithin3Degrees

Description: Get info of collaborator. Pass the person_id, depth and receive name and number of publication of a collaborator.

Usage:

1. Request

```
{"params":{"personID":82708,"depth":6,"method":"getNodesWithin3Degrees"}}
```

2. Response

```
{"exeTime":0.081821918487549,"result":{"data":[{"name":"William Levine","ID":"55687","numPublications":1},...],"status":"OK"}, "id":null}
```

► getOverview

Description: Get overview of the knowledge base.

Usage:

1. Request

```
{"method":"getOverviewSP","params":{}}
```

2. Response

```
{"exeTime":0.78313589096069,"result":{"data": {"numPapers":23181,"numAwards":117882,"numAuthors":134583,"numPls":100315,"numKeywords":262671}, "status":"OK"}, "id":null}
```

► getPersonCollaborationNetwork

Description: Get data to build collaboration network. The data is returned in a SVG format carrying the information for a dynamic two dimensional vector graphics.

Usage:

1. Request

```
{"params":{"POID":33305,"method":"getPersonCollaborationNetwork"}}
```

2. Response

```
{"exeTime":1.5413630008698,"result":{"status":"OK","data":"services\tempGVs\1324062686.64.svg"}, "id":null}
```

► getPOOverview

Description: Get award details of a person.

Usage:

1. Request

```
{"method":"getPOOverview","params":{"personID":33305}}
```

2. Response

```
{"exeTime":0.0057821273803711,"result":{"data":{"approvedAwardAmount":121010468,"approvedNumAwards":485,"status":"OK"},"id":null}
```

► getProgramOfficerListSP

Description: Get info of officers and award against a keyword.

Usage:

1. Request

```
{"params":{"keywordID":2051}, "method":"getProgramOfficerListSP"}
```

2. Response

```
{"exeTime":0.03731107711792,"result":{"data":[{"personID":38634,"numAwards":1,"name":"Mary Lynn Realf"},...],"status":"OK"},"id":null}
```

► getPublicationDistribution

Description: Get publication name and keyword count against a keyword

Usage:

1. Request

```
{"params":{"keywordID":2051}, "method":"getPublicationDistribution"}
```

2. Response

```
{"exeTime":0.010564088821411,"result":{"data":[{"cnt":4,"publicationName":"Computer Applications In Engineering Education","publicationID":"4"},...],"status":"OK"},"id":null}
```

► getPublications

Description: Get info of all the publications.

Usage:

1. Request

```
{"method":"getPublications"}
```

2. Response

```
{"exeTime":0.0025718212127686,"result":{"data":[{"ID":3,"publicationName":"Advances In Engineering Education","minYear":2007,"maxYear":2009,"publicationGenre":"Journal","numDocuments":20},...],"status":"OK"}, "id":null}
```

► getTagCount

Description: Get top 50 tags.

Usage:

1. Request

```
{"params":{"numTags":50,"method":"getTagCount"}
```

2. Response

```
{"exeTime":0.0068650245666504,"result":{"data":[{"ID":4756,"name":"educational courses","weight":3773},...]}
```

► getYearlyDistribution

Description: Get number of publication and year against a keyword.

Usage:

1. Request

```
{"params":{"keywordID":60123,"method":"getYearlyDistribution"}
```

2. Response

```
{"exeTime":0.025763988494873,"result":{"data":[{"publicationYear":1998,"cnt":1},...],"missingYears": [1999,2000,2002],"status":"OK"}, "id":null}
```

C. A Complete ERD

