BIBEX: a Bibliographic Exploration Tool based on the DEX Graph Query Engine

Sergio Gómez-Villamor Gerard Soldevila-Miranda Aleix Giménez-Vañó
Norbert Martínez-Bazan Victor Muntés-Mulero
Josep-L. Larriba-Pey

DAMA-UPC
Computer Architecture Department
Universitat Politècnica de Catalunya
Jordi Girona 1-3, Campus Nord-UPC, Mòdul C6-S103, E-08034 Barcelona
e-mail: {sgomez,gsoldevi,agimenez,nmartine,vmuntes,larri}@ac.upc.edu

ABSTRACT

In this demonstration we show the Bibliographic Exploration tool BIBEX. BIBEX is based on the graph database query engine DEX and integrates both the Citeseer and DBLP databases. BIBEX can be found in our web site at www.dama.upc.edu/bibex.

BIBEX allows for complex bibliographic search and shows the results of its queries as a combination of graphs and text for two types of scenarios: scientists who want to do complex bibliographic search, and PC members and journal editors who want to have an aid to search best suited reviewers that do not have conflicts of interest with the authors of papers submitted for review.

The tool allows to explore the relation between authors, keywords and papers in such a way that a user may perform a complex bibliographic search to investigate the *who is who* in specific areas of research.

EDBT'08, March 25–30, 2008, Nantes, France.

1. INTRODUCTION

The need for powerful bibliographic databases and search tools has motivated the existence of DBLP [3], Google Scholar [4] or Citeseer [2]. The importance of those devices has been evident and many scientists are using them regularly to do bibliographic keyword search. However, the graph nature of the data stored in those databases has not been exploited at its most. Such nature, would allow to place complex queries leading to answers with an added value to the scientific community.

In this paper we present BIBEX, a tool that is based on the use of a graph query engine, DEX [18], that allows to place such complex queries over a set of bibliographic databases. BIBEX stands for Bibliographic Exploration Tool and has the objective to answer pre-programmed queries over the data sets at hand. An answer to a query with BIBEX is basically a graph of relations plus a set of descriptive and/or statistical information that can be obtained from the answers to the query or the graph database itself.

BIBEX is a powerful tool that is able to process large quantities of data coming from different bibliographic databases such as Citeseer and DBLP at present, but could be extended to other bibliographic databases such as the ACM Digital Library [1] or IEEE Explore [5]. In order to create a full graph repository, BIBEX links the records of all those databases on the basis of the relationships among them. The relationships between records are explicit within each database, or artificially created among different databases using the names of the authors, the keywords of the papers or the names of the papers.

BIBEX allows to perform queries under two different flavors: (i) that of the researcher who intends to search for papers and their references, and authors and their relationships in specific areas of research, and (ii) that of the journal editor or PC member who search for good reviewers for a paper, i.e. non interest-conflicting researchers.

There are different graph database query engine proposals in the literature that could be used for our purpose [15, 19]. However, DEX has some interesting characteristics that make it suitable for the bibliographic context: it allows to deal with multiple data sources in different formats, e.g. relational, XML, CSV; it is oriented to deal with large data sets and it provides fast response times.

In this paper, we explain BIBEX, the queries that are included in the BIBEX demo and the type of answers that BIBEX expresses in its user interface. In order to make the paper self content, we also give a short overview of DEX to understand the internals of the query engine used for BIBEX.

2. BIBEX

All BIBEX data is stored in a unique graph which integrates data from different data sources. This graph is managed by the DEX framework [18]. Original data sources may be multiple and include RDBMSs, CSV files, Bibtex references, XML files and others. At this moment, BIBEX only integrates data from Citeseer and DBLP, which are both available as XML files, but we plan to integrate more data sources in the future.

The tool allows the user to explore bibliographic information accessing the integrated data stored in the graph. In all the cases, BIBEX allows for the extraction of useful information from nodes and relationships between nodes.

When a query is launched, BIBEX either explores the graph database starting from one node, or it proceeds with all the nodes of a type. For example, it explores the graph database starting from an author or it proceeds with a set of nodes that fulfil certain keyword parameters imposed by the user.

2.1 Interface

The interface of BIBEX is very simple, including six main panels as shown in Figure 1. The top left panel lists all the available queries that may be launched. Nowadays, three queries are available. Once a query is selected, the same panel allows to set the parameters for the specified query. The bottom left panel shows a tree with the history of executed queries. The central panel shows the main answer to a query in the form of a graph. The right hand side panels show specific information for a selected component (node or edge) from the graph of the central panel. Specifically, the top right panel shows the information integrated from different data sources or the selected component, the middle right panel shows inferred papers from the query, and the bottom right panel shows statistics gathered during the execution of the query.

The system allows for simultaneous live queries. This is important since it may be necessary to have multiple simultaneous answers for a specific bibliographic investigation. The tool also allows to obtain printed reports that reflect a condensed summary of a query. With this information, the search process may be easier to be carried.

Finally, the system allows to navigate and obtain the papers from the bibliographic sources when a url exists and the user is granted access to those sources. It is also possible to access the web pages of the authors if the information is included in the databases.

3. QUERIES

In this section we describe three queries that can be launched with BIBEX and are related to authors, keywords and papers. At this moment, our tool allows to execute those three but the set and functionalities will grow in the future. In any case, other queries can be imagined and implemented easily.

We divide the queries into two different sets, those for the plain researcher, and those for the editor/PC member.

3.1 Researcher queries

Query 1. **Partners/keywords.**

This query generates a graph with all the authors that have collaborated with one or more authors specified in the query. The graph shows author names in the nodes with links between their collaborators and tags in the links with the number of papers written by them.

The query allows to include topic or paper keywords as parameters. At this moment we use the paper title keywords for this purpose but the search could be enriched with real keywords present in other sources. The use of keywords allows to restrict the search to specific topic areas for the author or authors analyzed. A query generates a view with all the subgraphs that respond to it, or one view for each of the subgraphs generated.

Parameters:
$<$ author-name$_1$, author-name$_2$, ..., author-name$_N$ $>$
$<$ keyword$_1$, keyword$_2$, ..., keyword$_N$ $>$

Options: On mouse left button click on a node or edge of the answer graph, BIBEX generates information in the right hand side panels:

- The stored data for the node (top panel).
- The papers related to the edge or node (middle panel).
- The statistics gathered for the query (bottom panel).

Also, a right button click on an author node allows for the execution of chained queries.

On mouse left button click on a paper name in the middle right hand side panel, the authors of that paper are highlighted in the main graph.

Furthermore, the links to the home pages of the authors or the urls of papers are included in the answers if they are available in the original data sources. They appear in the answer graph (central panel) and in the lists of papers

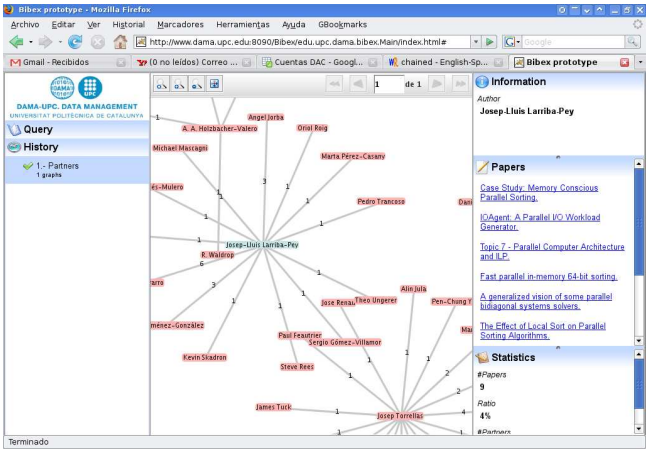

Figure 1: BIBEX window with different panels showing the answer to a query.

generated on the names of the authors (middle right hand side panel).

Use of the query: This query may be used to investigate authors and their relation to topic areas, the papers they have written and who they have collaborated with. This is useful for researchers who are new in an area like PhD students, or those who want to know new papers in a topic.

Query 2. **Keyword recommender.**

Based on a set of authors and keywords, the query searches for the keywords that have not been specified in the query, but are present in the papers involved in the query. Among those keywords, it gathers the groups of keywords that are above a certain support threshold. Finally, it searches for the papers that contain the groups of supported keywords and shows the relation among the authors who have written them.

The result for this query shows a graph for each set of keywords with a minimum support, showing the authors related to those keywords. An example for the answer graph of this query is shown in Figure 2.

Parameters:

$<$ author-name$_1$, author-name$_2$, ..., author-name$_N$ $>$

$<$ keyword$_1$, keyword$_2$, ..., keyword$_N$ $>$

Options: The mouse options for this query are similar to those in query 1, and follow the same logics. For these reasons we avoid explaining them.

Use of the query: This query may be used to search for papers in areas of research when the user does not have a certain knowledge of the keywords in the area. Also, when one wants to know about the papers published close to a certain area.

3.2 Editor/PC member queries

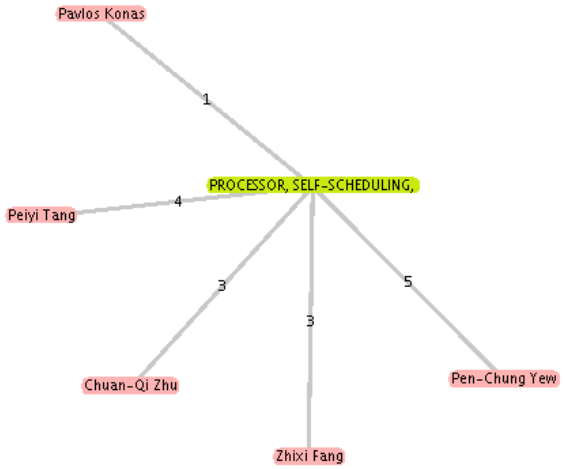

Figure 2: Answer graph for Query 2, showing two keywords and authors related to them.

Query 3. **Reviewers for a paper.**

Based on a set of keywords and author names, this query recommends a set of reviewers that do not have any conflict of interest with the authors. Information related to authorship is used to discard researchers that have collaborated with the authors stated in the query. The query returns a graph with researchers that could review a paper related to the set of keywords introduced for the query, and their relations with other researchers.

Parameters:

$<$ author-name$_1$, author-name$_2$, ..., author-name$_N$ $>$

$<$ keyword$_1$, keyword$_2$, ..., keyword$_N$ $>$

Options: The mouse options for this query are similar to those in query 1, and follow the same logics. For these reasons we avoid explaining them.

Use of the query: This query, may be used by editors and PC members to find reviewers for papers. It may be also helpful for researchers looking for groups that work on similar topics in a competitive way.

4. DEX

In order to make the paper self content, we explain the main characteristics of DEX and its use with a bibliographic database.

DEX is based on a graph database model, that is basically characterized by three properties: data structures are graphs or any other structure similar to a graph; data manipulation and queries are based on graph-oriented operations; and there are data constraints to guarantee the integrity of the data and its relationships. For further details, we refer the reader to [18].

4.1 Graph Model

The most basic logical data structure in DEX is a labeled and directed attributed multigraph $G = \{T, N, E\}$, where T is the collection of labels, N is the collection of nodes and E is the collection of directed edges. A *labeled* graph has a label for each node and edge, that denotes the object type. A *directed* graph has all edges with a fixed direction, from the *tail* or source node to the *head* or destination node. An *attributed* graph allows a variable list of attributes for each node and edge, where an *attribute* is a value associated to a name, simplifying the graph structure. Finally, a *multigraph* allows multiple edges between two nodes. This means that two nodes can be connected several times by different edges, with the only restriction that it is not allowed to have two edges with the same tail, head and label. A directed multigraph helps to represent the richness of the multiple relationships between objects as this is a usual situation into complex social networks.

A label or object type $t \in T, t = \{typename, A\}$, has a unique name *typename* and a collection of attribute definitions, where each attribute $a \in A$, $a = \{attrname, domain\}$ has a unique name *attrname* and is restricted to a *domain* (for instance strings, numbers or timestamps). Thus, the objects of a certain type can only contain values within the same set of strong-typed attributes.

A node $n \in N, n = \{key, type, depth, text, V\}$ has an optional unique *key* that can serve as a unique identifier for the original data source, like a ROWID in a relational database or an URL into the WWW. It belongs also to a $type \in T$, and

has a *depth* that represents its level in the hierarchy of nodes. It can also have an optional *text* to store a user-defined string: a description, an HTML page, the text of an XML element, etc. Each node has a collection $V = \{(a, v)\}$ of values for the attributes of its type, where $a \in A(type) \wedge v \in domain(a)$. An edge $e \in E, e = \{tail, head, type, depth, V\}$ has a *tail* or source node, a *head* or destination node, and like the nodes, it belongs to a $type \in T$, has the *depth* into the graph hierarchy and a collection $V = \{(a, v)\}$ of values for the attributes of its type.

4.2 Data Representation

DEX deals with two different types of graphs: the *DbGraph*, a single graph that contains the schema and data extracted from external data sources and stores them in a persistent storage; and the *RGraph*, the result of a DEX query, which is stored in the temporal storage. The information contained in the DbGraph is used as source data by DEX queries to obtain the results in the form of one or more RGraphs.

The DbGraph includes two subgraphs: the *schema* subgraph, which contains the metadata of the data sources, and the *data* subgraph, which contains the data loaded from the data sources as defined in the schema subgraph. Both are connected through edges between the entity definitions of the schema subgraph and their instance objects in the data subgraph. The schema subgraph contains the following node types:

datasource: contains information of a data source (ODBC, CSV, XML, etc.). It is connected to one or more *datasets*.

dataset: the definition of an entity or collection of data units (rows, XML elements, HTML pages, etc.). It is defined by one or more *attributes*.

attribute: a characteristic or property of a dataset (column, xml attribute, etc.). It has a name and a data type and can be part of one or more *relationships*.

relationship: it can represent either any edge between two instances in any dataset, or the definition of a constraint between two attributes (for example a foreign key).

Figure 3 shows an example of the mapping between a simple bibliographic database stored in a RDBMS, and the corresponding schema subgraph through its DEX representation. Although BIBEX has a similar structure, it is more complex due to the integration of different sources and the more complex database schemas found. Attributes belong to datasets and datasets to datasources, creating a hierarchy that depends on the original data sources. However, relationships may link datasets from a single or across multiple data sources (intra- or inter-relationships, respectively). This goes beyond the classical data models, allowing the DbGraph to connect and relate heterogeneous data sources that share common identifiers, as if they were foreign keys across different data sources. Thus, if there are two distinct databases that have the entity PERSON, and both entities have a public person identifier like the social security number (SSN) as a primary key, a relationship between them can be created within the DbGraph to join all the information of each single person. This provides the illusion of a single and integrated storage. In the future these relationships can also serve as integrity constraints between multiple data sources.

5. RELATED WORK

The need for bibliographic search engines has been partially solved by the use of tools like DBLP [3], Citeseer [2], Google Scholar [4] or Live Academic [6]. However, to the best of our knowledge, there is a lack of work in the area of complex queries, and we cover it with this paper.

On the other hand, there are many graph database models in the literature such as GOOD [11], GRAM [8], GOAL [14], GraphDB [13], GRAS [15], GRACE [19] and RDF [16] [9]. Among these, GRAS and GRACE propose an implementation of a graph database management system for small and medium sized databases. Increasing the semantics of the results in keyword search has been addressed in several approaches, where special-purpose tools, like BANKS [10] or Précis [17], were built to map a relational database into a graph to be returned as a response for a keyword search query. Other interesting works have focussed on the performance of graph querying, like the work presented in [7] and GraphGrep [12].

There are other different graph database formalizations like those found in [8, 9, 11, 13, 14, 16].

6. CONCLUSIONS

The need for complex bibliographic search in the scientific community makes it possible to propose tools that help researchers for such task. Those tools require clear yet complex requirements like the possibility to query various data sources with large data sets in reasonable execution times.

In this paper we have presented BIBEX, a bibliographic exploration tool that allows scientists and editors to perform complex queries on multiple bibliographic databases. BIBEX is a front end that makes use of DEX, a graph query engine developed by DAMA-UPC, that allows complex queries on large data sets in fast execution times. DEX is being used by the Spanish registrars to detect fraud in property transactions, which gives a solid background to make BIBEX a robust front end for the use in complex bibliographic search.

7. FUTURE WORK

The need for the type of tools explained in this paper makes it possible to evolve BIBEX further with the objective to enlarge the number of queries and quality of the answers obtained. With these objectives in mind, we plan for further development of BIBEX in the following directions:

- New queries that are paper centric as opposed to author centric as those explained in this paper.
- Metrics to help users to decide on the centricity of authors and/or papers like the h index.
- Improvement of the reporting tools for the queries offered by BIBEX.
- Use of the logs of user accesses to derive further recommendations. Note that the logs of users may be represented as a graph, and DEX can deal with them as well.
- Accessing the web as a resource to obtain data that improves the answers for different types of queries. These data coming from the web may enrich the graph stored by BIBEX in a permanent way.

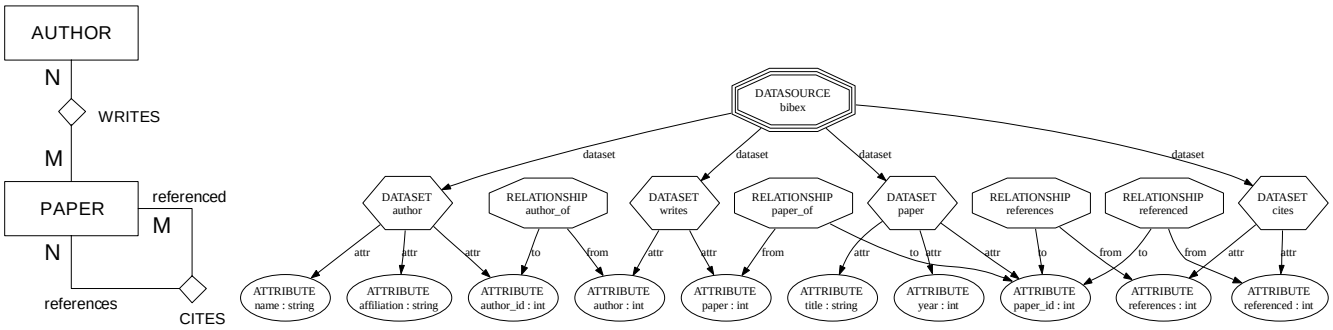

Figure 3: Example of E/R and DbGraph for a Bibliographic Database

- Use of approximate matching and edit distance metrics for the integration of different data sources. In order to distribute the cost of integration, a system could be implemented in the form of a wiki, so that after the recommendation of BIBEX, a user could validate that two or more nodes represent the same author. The system could integrate the information in the BIBEX graph permanently.

We would like to make a call to the community asking for proposals oriented to the evolution of BIBEX. Any suggestion or bug in the code could be addressed to bibex@dama.upc.edu.

8. ACKNOWLEDGMENTS

The authors from UPC want to thank Generalitat de Catalunya for its support through grant number GRE-00352 and Ministerio de Educación y Ciencia of Spain for its support through grant TIN2006-15536-C02-02. Josep L. Larriba-Pey wants to thank Ministerio de Educación y Ciencia of Spain for his I3 grant.

The authors want to thank Javier Castro for his effort to make BIBEX possible.

9. REFERENCES

[1] ACM digital library, http://www.acm.org.

[2] Citeseer, scientific literature digital library, http://citeseer.ist.psu.edu/.

[3] DBLP, computer science bibliography, http://www.informatik.uni-trier.de/ ley/db/index.html.

[4] Google scholar, http://scholar.google.com.

[5] IEEE explore, http://www.ieee.org.

[6] Live search academic, http://search.live.com.

[7] R. Agrawal and H. Jagadish. Algorithms for searching massive graphs. *IEEE Transactions on Knowledge and Data Engineering*, 06(2):225–238, 1994.

[8] B. Amann and M. Scholl. Gram: a graph data model and query languages. In *ECHT '92: Proceedings of the ACM conference on Hypertext*, pages 201–211, New York, NY, USA, 1992. ACM Press.

[9] R. Angles and C. Gutiérrez. Querying rdf data from a graph database perspective. In *ESWC*, pages 346–360, 2005.

[10] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. *In Procs. of ICDE, 2002.*, pages 431–440, 2002.

[11] M. Gemis, J. Paredaens, I. Thyssens, and J. V. den Bussche. Good: A graph-oriented object database system. In P. Buneman and S. Jajodia, editors, *Proc. of the 1993 ACM SIGMOD, Washington, D.C., May 26-28, 1993*, pages 505–510. ACM Press, 1993.

[12] R. Giugno and D. Shasha. Graphgrep: A fast and universal method for querying graphs. *International Conference on Pattern Recognition*, 02:20112, 2002.

[13] R. H. Güting. Graphdb: Modeling and querying graphs in databases. In *Proc. of the 20th VLDB Conference*, pages 297–308, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[14] J. Hidders and J. Paredaens. Goal: a graph-based object and association language. *CISM - Advances in Database Systems*, pages 247–265, 1993.

[15] N. Kiesel, A. Schuerr, and B. Westfechtel. Gras, a graph-oriented engineering database system. *Information Systems*, 20(1):21–51, 1995.

[16] G. Klyne and J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004. *http://www.w3.org/TR/2004/REC-rdf-concepts-2004*.

[17] G. Koutrika, A. Simitsis, and Y. Ioannidis. Précis: The essence of a query answer. In *Proc. of the ICDE'06 Conference*, page 69, Washington, DC, USA, 2006. IEEE Computer Society.

[18] N. Martínez-Bazán, V. Muntés-Mulero, S. Gómez-Villamor, J. Nin, M.-A. Sánchez-Martínez, and J.-L. Larriba-Pey. Dex: High Performance Exploration on Large Graphs for Information Retrieval. In *Proceedings of the CIKM conference, Lisbon*, pages 573–582, 2007.

[19] S. Srinivasa, M. Maier, M. R. Mutalikdesai, G. K. A., and G. P. S. Lwi and safari: A new index structure and query model for graph databases. In *COMAD*, pages 138–147, 2005.