

AstroDAS: Sharing Assertions across Astronomy Catalogues through Distributed Annotation

Rajendra Bose¹, Robert G. Mann², Diego Prina-Ricotti³

¹ U.K. Digital Curation Centre and School of Informatics, University of Edinburgh

rbose@inf.ed.ac.uk

² Institute for Astronomy, University of Edinburgh

rgm@roe.ac.uk

³ Dipartimento di Informatica e Automazione, Università di Roma Tre

dpricott@inf.ed.ac.uk

Abstract. As diverse scientific data collections migrate online, researchers want the ability to share their assertions regarding the entities that span these disparate databases. We focus on a case study provided by the astronomical community’s Virtual Observatory effort to investigate the use of annotation to record and share the celestial object mappings asserted by different research groups. The prototype for our Astronomy Distributed Annotation System (AstroDAS) complements the existing OpenSkyQuery tools for federated database queries, and provides web service methods to allow clients to create and store mapping annotations as relational database tuples on annotation servers. We expect the mechanisms for creating and querying annotations in AstroDAS can be extended to assist with tasks other than entity mapping, in other domains with relational data sources.

1 Introduction

Research activity in a number of scientific domains includes organizing and analyzing content culled from diverse collections of online data. Usually, portions of different databases are extracted and placed into personal or research computing environments within a particular organization or group (which may include autonomous relational or XML databases, spreadsheets, and so forth) where the analysis is conducted. We focus on a case study from astronomy, where a group analyzes data from different source databases, and then *asserts* mappings between entities in the data sources. Others who access the source databases may benefit from receiving these assertions as feedback.

Central to the astronomical community’s concept of a global “Virtual Observatory” (VO) is the ability to identify records in databases of astronomical observations as referring to the same celestial object: for example, a typical VO use case might start by matching entries from an optical, an X-ray and a radio database, so that an astronomer could analyze the multi-wavelength properties of a particular class of galaxy. The nascent VO already includes web services which implement the matching of catalogues by spatial proximity alone, but the general catalogue matching problem is

more difficult—and more computationally expensive—than that. This problem motivates our investigation of recording, through *annotations*, the assertions made regarding the associations between entries in different databases, such that they can be re-used by third parties.

In our work, we build on the ideas introduced by the Distributed sequence Annotation System (DAS, later BioDAS) [1]. BioDAS is specifically designed to facilitate genome sequence annotation, however, so while the concepts it introduces are valuable, the approach and means of implementation for this system are not suitable for sharing assertions in other scientific domains. Thus we first introduce a general framework to help clarify and compare other types of annotation systems. Following this, we describe our design of a system for distributed annotation in the context of OpenSkyQuery, a prototype implementation of the International Virtual Observatory Alliance (IVOA) SkyNode specification for nodes in the federated VO. We discuss the topic of matching celestial objects in astronomy catalogues, and describe how the prototype for our Astronomy Distributed Annotation System (AstroDAS) complements existing OpenSkyQuery tools for federated database queries. We close with a brief discussion of related work.

2 A Framework for Annotation

The BioDAS protocol for genome sequence annotations is not directly applicable to the process of annotation in other areas of science. In order to contrast other types of annotation systems we suggest an informal framework that consists of the following basic components:

An *annotation* is some set of data elements that is added to an existing *base* or *target* that possesses structure; the base or target structure can be described by some reference system; and the *point* or *location of attachment* of an annotation can be described using this same reference system.

These various components of BioDAS are summarized in column (2) of Table 1: the annotation target is a (conceptualized or idealized) genome, the target structure consists of a linear sequence of nucleotides (base pairs), and the reference system for the target is a linear coordinate system of base pair numbers. Annotations are usually the identification of particular genes or their products (such as proteins), and the location of annotation attachment is designated by start and stop nucleotide positions (base pair numbers). In this context, the purpose of annotations is to collect and interpret the results of numerous biological experiments or computer algorithms.

Consider the very different example for the annotation of medical images described in [2] (Table 1, column (3)). Here the annotation target is a Human Brain Project (HBP) image, the target structure is a 2D array of pixels, the reference system for the target is a 2D coordinate system of X,Y pixel values, annotations are concepts (for example, controlled medical vocabulary terms), and the location of attachment is a 2D region of interest described by reference to the coordinate system. Other systems with a similar purpose exist, including the Edinburgh Mouse Atlas [3].

Table 1. Annotation framework and system comparison

(1) Framework components	(2) BioDAS	(3) HBP image annotation	(4) AstroDAS
Annotation target:			
what	genome	brain image	celestial object in astronomy catalogue (RDBMS tuple)
structure	sequence of base pairs	pixel array	catalogue (RDBMS) schema
reference system	linear coordinate system of base pair numbers	2D coordinate system of pixel X,Y values	catalogue+celestial object id (tuple key)
Annotation:			
what	gene or gene product	domain-specific concept	mapping to a celestial object in a different catalogue
location of attachment	start, stop base pair numbers	2D shape	specific catalogue +celestial object id
purpose	collect and interpret research results on genome	link Web-accessible images to, and query on, concepts	share assertions of celestial object matches across different astronomy catalogues

Column (4) of Table 1 summarizes the components of the AstroDAS annotation system prototype described in the remainder of this paper; these components are somewhat similar to the relational annotation work discussed in [4, 5]. In this case, the annotation target is an entry (tuple) in an astronomy catalogue implemented with a relational database management system (RDBMS), the target structure is given by the catalogue RDBMS schema, and the reference system for the target is provided by catalogue and database object ids (tuple keys), possibly combined with the name or location of specific attributes. Annotations are other tuples that map one catalogue object to another catalogue object, and the location of attachment is a specific catalogue and object id combination. In our work we use relational database framework components because we focus on providing annotation over the federated relational databases in the OpenSkyQuery system, described in the following sections.

3 Matching Celestial Objects in Astronomy Catalogues

Over the past several decades, collections or *catalogues* of celestial object observations, recorded by disparate telescopes and other instruments over various time periods, have migrated online. Astronomy catalogues have different schemas but are usually organized according to a star schema consisting of a primary relation of celestial objects that serves as the basis for most joins. As expected, each tuple in the pri-

mary relation contains a key or *celestial object identifier* (id) that is unique within a specific catalogue. Most astronomy catalogues are updated through large data releases, for example every three to six months, but we assume here that all ids within catalogues are persistent and stable.

Newer astronomy instruments are designed to provide *sky surveys* of large portions of the celestial sphere. While older catalogues consist of hundreds or thousands of objects, these new instruments generate on the order of hundreds of gigabytes of data per day, contributing to catalogues (also known as survey archives) that record the observations of hundreds of millions of celestial objects. The recorded location of a celestial object may vary slightly from catalogue to catalogue due to unavoidable measurement error at the instrument level [6].

The OpenSkyQuery system aims to federate astronomy catalogues and archives through the use of web services and a wrapper-mediator architecture [7]. Federated queries are executed by a portal application that communicates to registered *sky nodes* (wrappers for astronomy catalogues) through a standard web service interface (See lower right of Figure 1. The three sky nodes shown correspond to the catalogues for the Sloan Digital Sky Survey (SDSS), the Two Micron All Sky Survey (TWO MASS), and the U.S. Naval Observatory USNO-B1.0 catalogue (USNOB).)

The queries are expressed in ADQL (Astronomical Data Query Language) [8], which is based on a subset of SQL92 with two extra keywords: *Region* and *XMatch*. The *Region* keyword allows the user to specify the spatial extent of the search using celestial coordinates. The *XMatch* keyword lets the user specify a measure of probabilistic uncertainty *sigma*; for a given *sigma*, an X-match query result consists of two or more columns of potentially coincident celestial object ids (and possibly other attributes) [6]. The locations of the resulting celestial object matches essentially cross catalogues (that is, exist across two or more catalogues) within a specified *sigma*, and therefore match. In this case, however, any further assessment or analysis of X-match results will probably occur in a personal or research computing environment separate from the federated OpenSkyQuery system.

The simple spatial proximity match performed in X-match is inadequate in the general case, where differences in the angular resolution of the data in the two catalogues may mean that a large number of objects from one catalogue (A) will lie within the positional error ellipse of each source in the other (B). In that case, spatial proximity alone cannot judge between the potential counterparts from A of each source in B, and it is necessary to either introduce prior astrophysical knowledge to help identify the most likely match, or use a machine learning algorithm [9, 10]. Machine learning algorithms deduce relationships between the properties of the sources in the two catalogues and can aid the finding of associations between them.

Adding sophistication to the matching algorithm also adds computational cost, and one of the main motivations behind investigating the AstroDAS system is the desirability of having a means of recording, and making available for re-use, associations made by algorithms more complex than the simple X-match function implemented in OpenSkyQuery. This has been studied by Taylor [11], who has implemented as web services a number of cross-matching algorithms which make use of non-spatial attributes as well as simple proximity matching. The design of AstroDAS was influenced by the desire to store the results of these services.

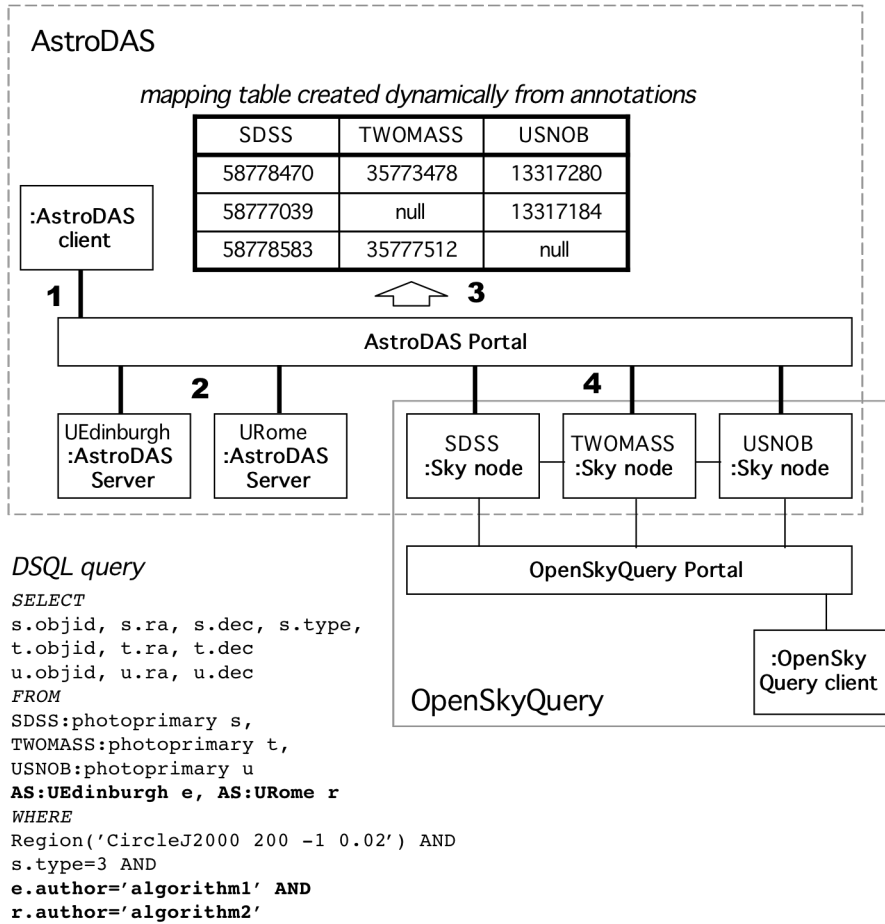


Fig. 1. OpenSkyQuery and AstroDAS architecture

Thus, to provide astronomers with the ability to share their assertions about matching celestial objects directly with their colleagues, we investigate prototypes for AstroDAS, a distributed annotation system that supports queries on annotation, and that is compatible with the federated architecture of OpenSkyQuery.

4 AstroDAS: Asserting Celestial Object Matches using Distributed Annotation

AstroDAS considers the identification of entities across sky nodes to be a database integration problem. Its solution, inspired in part by BioDAS, features an annotation database with a web service interface to store and query annotations. The use of web services provides interoperability with the rest of the Virtual Observatory. Our proto-

type system resolves queries on astronomy catalogues using mapping tables that are dynamically constructed from annotations of celestial object matches.

4.1 Celestial Object Matching as Database Integration

Storing assertions of celestial object matches is essentially the problem of *entity identification* that occurs in traditional database integration scenarios. Entity identification is the ability to find “object instances from different databases which correspond to the same real-world entity” [12]. In database integration, *mapping tables* are commonly employed to store the keys of corresponding tuples that reside in different databases [13].

We make use of mapping tables to express celestial object matches across federated astronomy catalogues. Although the schemas of sky node catalogues differ, we mention in Section 2 that each catalogue contains one primary relation with unique celestial object ids. We thus consider a specific celestial object instance, which we call a *database object*, as the ordered pair: <sky node name, celestial object id>. In our case, mapping tables consist of tuples of database objects that map to one another.

Given the collection of decentralized, read-only sky nodes, and the lack of a central authority to resolve disputed mappings, we use the concept of distributed annotation to provide autonomous research groups with the means to store their own celestial object matches locally. In our approach, a group stores *mapping annotations* that refer to two or more sky node entries (database objects) on their own annotation server. A group’s assertions of celestial object matches can then be shared by the wider community if the group provides online access to their annotation server web service interface. Once this is done, queries on specific sky nodes can include parameters for annotation.

4.2 Storing and Querying Mapping Annotations in AstroDAS

Storing and querying mapping annotations in AstroDAS are achieved through the mechanism of web services. Web services are compatible with the OpenSkyQuery infrastructure adopted by the IVOA; they also facilitate client implementation and promote system interoperability. Similar to OpenSkyQuery, AstroDAS follows the wrapper/mediator architecture whereby a client connects to a portal (mediator) that accesses data on both annotation servers and sky nodes through wrappers implemented with a web service interface (Figure 1). The wrappers provide a common data model and query language, and hide the potential heterogeneity of the different data sources.

Note from Figure 1 that AstroDAS is separate from OpenSkyQuery. The loose coupling of web services means that it has been possible to make AstroDAS interoperable with the OpenSkyQuery system without modifying OpenSkyQuery’s existing code or requiring any action by its developers and maintainers. AstroDAS web service methods exist to allow clients to create and store mapping annotations as relational database tuples on AstroDAS annotation servers (Figure 1, label number 2).

AstroDAS includes the custom query language DSQL (Distributed SQL), similar in purpose to MSQL [14] but more limited in scope. DSQL was designed to retain a syntax as close as possible to ADQL, but *from* clauses in DSQL can specify a list of annotation servers to access annotations from, and *where* clauses in DSQL can specify constraints on mapping annotations.

A DSQL query example is shown in the lower left of Figure 1. The structure is identical to an ADQL query, with the addition of constraints unique to DSQL shown in bold. The hypothetical query shown requests the value of attributes for celestial objects that match according to either the results of “algorithm1” stored on the University of Edinburgh AstroDAS annotation server, or the results of “algorithm2” stored on the University of Rome AstroDAS annotation server.

Methods to retrieve mapping annotations in the form of a mapping table are available. One aspect of the AstroDAS portal design concerns the use of an *inference algorithm* to simplify mapping tables by inferring new mappings from existing ones. Because mappings are transitive, the simplified mapping table to the right of Figure 2 can be inferred from the mapping table to the left, for example.

The execution of a DSQL query similar to the example shown in the lower left of Figure 1 proceeds as follows: A client sends a web service request to the AstroDAS portal mediator with a DSQL query as a parameter (Figure 1, label number 1). The portal parses the DSQL query, generates a query for each annotation server and executes them (Figure 1, label 2). The portal then receives the mapping table results of the annotation queries, and dynamically combines the separate mapping tables into one if necessary (Figure 1, label 3). The mapping table shown in Figure 1, label 3 could have resulted from, for example, the UEdinburgh server storing the annotations:

```
<SDSS, 58778470> maps to <TWOMASS, 35773478>
<SDSS, 58778470> maps to <USNOB, 13317280>
<SDSS, 58777039> maps to <USNOB, 13317184>
```

and the URome server storing the annotation:

```
<SDSS, 58778583> maps to <TWOMASS, 35777512>
```

The inference algorithm is executed on the resulting mapping table if a specific keyword is included in the DSQL query. The mapping table is used to generate the queries for the astronomy catalogues in order to retrieve the actual data. The portal performs the queries by contacting the sky node web service wrappers (Figure 1, label 4). With the query result relations retrieved from the astronomy catalogues, the portal uses the mapping table to combine mapped entities; that is, the data corresponding to the same celestial object is concatenated in the same row of the resulting table. Finally, the portal returns the result to the client (Figure 1, label 1).

Thus, through annotation retrieval, DSQL queries similar to the given example provide a means to perform *entity joins* [15] across different sky nodes. Successive prototypes of AstroDAS have been implemented by the authors and tested by as-

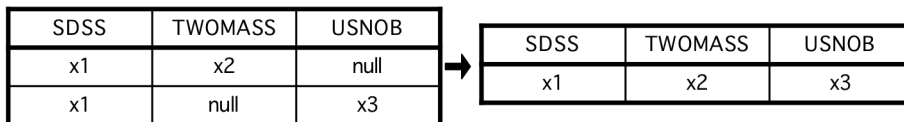


Fig. 2. Applying inference to a mapping table

tronomers from the University of Edinburgh associated with AstroGrid, the UK's VO development project (www.astrogrid.org). The AstroDAS portal and annotation server web service interfaces are implemented through Apache Axis2; the annotation server uses a PostgreSQL or IBM DB2 database, accessed by the web service interface through JDBC.

4.3 AstroDAS v2: Towards Peer-to-Peer Annotation

AstroDAS version 2 is still in development; this successor to AstroDAS is an evolution of the initial architecture to explore the replacement of annotation servers (see Figure 1) with *annotation peers*—nodes in a peer-to-peer (P2P) network of databases. Each peer shares its own set of annotations and cooperates with other nodes for retrieving query results. A distributed annotation system based on a P2P structure is expected to be more scalable than other architectures because the computational load is spread across the peers; the issue of scalability becomes important as the number of research groups sharing their annotation increases.

We refer to the same example DSQL query shown in the lower left of Figure 1, but describe its execution with a network of annotation peers rather than annotation servers. As before, the client sends a web service request to the AstroDAS portal mediator with a DSQL query as a parameter (Figure 1, label number 1). The portal parses the DSQL query, and generates a *single* query for the P2P annotation network that now takes the place of the two annotation servers shown by Figure 1, label 2. The portal then sends a web service request to one annotation node of its choice with the single query as a parameter. This annotation node is called the *coordinator* and is responsible for interacting with the other nodes in order to answer the query. The coordinator returns the mapping table result to the portal, and the DSQL query execution continues as previously described.

The AstroDAS portal logically views the P2P annotation network as a single global mapping table, which integrates all the local mapping tables in the network. Thus the portal only issues a single query on this global mapping table to the coordinator. A typical query could ask for mapping annotations that include: celestial objects belonging to a particular region of the sky; data held on an arbitrary number of skynodes; one or more specific authors; and some minimal degree of reliability. The coordinator determines the annotation peers that are involved with the query and sends an asynchronous web service request to those nodes to start the computation of their local mapping table. The coordinator then determines an execution plan in order to reduce network traffic and starts the execution that contacts the nodes in sequence to integrate the global mapping table from the local ones.

Furthermore, AstroDAS v2 provides a distributed inference algorithm, related to the one described in [13], that computes a global inferred mapping table. The computational load of this algorithm is distributed across all the nodes involved with the query, rather than executing only on the portal (as in the previous AstroDAS version).

5 Related Work

The annotation management system for relational databases presented in [4] and affiliated papers assume that source databases are already populated with annotations, such as location tags for all attributes in every tuple. Although annotations are propagated through the system, direct queries on annotation values are not possible. The Mondrian prototype [5] introduces a system for creating annotations for subsets of tuple attributes, as well as the associations between multiple values; this project includes the design of mechanisms to "query values and annotations alike (in isolation or in unison)." Image annotation projects [2, 3] use relational databases and also offer the ability to make these same types of queries. Annotations for the entity mapping in our work are different from [2, 3, 5]: we create annotations that are both external to their targets of relational tuples and distributed among different groups. Like these projects, however, we share the goal of providing the ability to query on a mixture of data and annotation values.

6 Conclusion

The ultimate aim of AstroDAS is similar to the goal of the earlier BioDAS: to record and share scientific assertions with a wider community. Whereas biologists use annotation to interpret the reference map of a genome, however, astronomers seek to share the mapping of entities derived from their research across established scientific databases. Specifically, astronomers want to be able to share their identification of matching celestial objects within the existing federation of disparate catalogues.

The contribution of our work is to demonstrate how distributed annotation can be used beyond the domain of bioinformatics to assert entity mappings across databases. Successive AstroDAS prototypes suggest the use of annotation servers that complement the existing OpenSkyQuery data access system. They also demonstrate how dynamic mapping tables constructed from stored annotations can serve as the means to map entities across disparate databases. We expect that continuing work will show that AstroDAS mechanisms for creating and querying annotations can be extended to assist with tasks other than entity mapping, in other domains with relational data sources.

Acknowledgements

We would like to thank: John Taylor, Emma Taylor, Martin Hill, and other members of the Wide Field Astronomy Unit, University of Edinburgh; Anastasios Kementsietidis, Floris Geerts and other members of the Database Group in the School of Informatics, University of Edinburgh; and Professor Paolo Atzeni at the Dipartimento di Informatica e Automazione, Università di Roma Tre for their support and help in this work. This project has been supported in part by the UK Digital Curation Centre, which is funded by JISC and the eScience core programme.

References

1. L. D. Stein, S. Eddy, and R. Dowell, "Distributed Sequence Annotation System (DAS) Specification Version 1.53," 21 March 2002. <<http://www.biodas.org/documents/spec.html>>
2. M. Gertz, K.-U. Sattler, F. Gorin, M. Hogarth, and J. Stone, "Annotating Scientific Images: A Concept-based Approach," in *Proceedings of the 14th International Conference on Scientific and Statistical Database Management (SSDBM 2002)*, Edinburgh, Scotland, 2002, pp. 59-68.
3. R. A. Baldock, C. Dubreuil, W. Hill, and D. Davidson, "The Edinburgh Mouse Atlas: Basic Structure and Informatics," in *Bioinformatics: Databases and Systems*, S. I. Letovsky, Ed.: Kluwer Academic Publishers, 1999, pp. 129-140.
4. D. Bhagwat, L. Chiticariu, W. C. Tan, and G. Vijayvargiya, "An Annotation Management System for Relational Databases," in *Proceedings of the VLDB*, Toronto, Canada, 2004, pp. 900-911.
5. F. Geerts, A. Kementsietsidis, and D. Milano, "MONDRIAN: Annotating and querying databases through colors and blocks," in *Proceedings of the ICDE*, 2006.
6. T. Malik, A. S. Szalay, T. Budavari, and A. Thakar, "SkyQuery: A Web Service Approach to Federate Databases," in *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, Asilomar, CA, 2003, pp. 17-26.
7. T. Budavari, A. S. Szalay, J. Gray, W. O'Mullane, R. Williams, A. Thakar, T. Malik, N. Yasuda, and R. Mann, "Open SkyQuery -- VO Compliant Dynamic Federation of Astronomical Archives," in *Astronomical Data Analysis Software and Systems (ADASS) XIII (ASP Conference Series)*, vol. 314, F. Ochsenbein, M. G. Allen, and D. Egret, Eds. San Francisco: Astronomical Society of the Pacific, 2004, pp. 177-180.
8. IVOA VOQL Working Group, "IVOA Astronomical Data Query Language Version 0.91," IVOA Working Draft 2005-02-25, IVOA, 2004-08-19. <<http://www.ivoa.net/internal/IVOA/IvoaVOQL/ADQL-0.91.pdf>>
9. D. J. Rohde, M. J. Drinkwater, M. R. Gallagher, T. Downs, and M. T. Doyle, "Applying machine learning to catalogue matching in astrophysics," *Monthly Notices of the Royal Astronomical Society*, vol. 360, no. 1, 2005, pp. 69-75.
10. A. J. Storkey, C. K. I. Williams, E. L. Taylor, and R. G. Mann, "An Expectation Maximisation Algorithm for One-to-Many Record Linkage, Illustrated on the Problem of Matching Far Infra-Red Astronomical Sources to Optical Counterparts," University of Edinburgh Informatics Research Report (EDI-INF-RR-0318). <<http://www.inf.ed.ac.uk/publications/report/0318.html>>
11. E. L. Taylor, *Ph.D. Thesis*, University of Edinburgh, 2005.
12. E. P. Lim, J. Srivastava, S. Prabhakar, and J. Richardson, "Entity identification in database integration," in *Proceedings of the ICDE*, 1993, pp. 294-301.
13. A. Kementsietsidis, M. Arenas, and R. J. Miller, "Mapping Data in Peer to Peer Systems: Semantics and Algorithmic Issues," in *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD) Conference*, 2003, pp. 325-336.
14. W. Litwin, A. Abdellatif, A. Zeroual, and B. Nicolas, "MSQL: A Multidatabase Language," *Information Sciences*, vol. 49, no. 1-3, 1989, pp. 59-101.
15. W. Kent, "The Entity Join," in *Proceedings of the Fifth International Conference on Very Large Data Bases (VLDB)*, Rio de Janeiro, Brazil, 1979, pp. 232-238.