Data Integration: The Teenage Years

Alon Halevy
Google Inc.
halevy@google.com

Anand Rajaraman
Kosmix Corp.
anand@kosmix.com

Joann Ordille
Avaya Labs
joann@avaya.com

1. INTRODUCTION

Data integration is a pervasive challenge faced in applications that need to query across *multiple* autonomous and heterogeneous data sources. Data integration is crucial in large enterprises that own a multitude of data sources, for progress in large-scale scientific projects, where data sets are being produced independently by multiple researchers, for better cooperation among government agencies, each with their own data sources, and in offering good search quality across the millions of structured data sources on the World-Wide Web.

Ten years ago we published "Querying Heterogeneous Information Sources using Source Descriptions" [73], a paper describing some aspects of the Information Manifold data integration project. The Information Manifold and many other projects conducted at the time [5, 6, 20, 25, 38, 43, 51, 66, 100] have led to tremendous progress on data integration and to quite a few commercial data integration products. This paper offers a perspective on the contributions of the Information Manifold and its peers, describes some of the important bodies of work in the data integration field in the last ten years, and outlines some challenges to data integration research today. We note in advance that this is not intended to be a comprehensive survey of data integration, and even though the reference list is long, it is by no means complete.

2. THE INFORMATION MANIFOLD

The goal of the Information Manifold was to provide a uniform query interface to a multitude of data sources, thereby freeing the casual user from having to *locate* data sources, *interact* with each one in isolation and *manually combine* results. At the time (the early days of the web), many data sources were springing up on the web and the main scenario used to illustrate the system involved integrating information from multiple web sources. This collection of sources became known later as the *deep web*. For example, the system was able to answer queries such as: *find reviews of movie directed by Woody Allen playing in my area.* Answering this query involved performing a join across the contents of three web sites: a movie site containing actor and director information (IMDB), movie playing time sources (e.g., 777film.com) and movie review sites (e.g., a newspaper).

A related scenario that is especially relevant today is searching for used cars (or jobs, apartments) in one's area. Instead of the user having to go to several sources that may have relevant postings (and typically, there are 20-30 such sites in large urban areas), the system should find all the postings for the user.

The main contribution of the Information Manifold was the way it described the contents of the data sources it knew about. A data integration system exposes to its users a schema for posing queries. This schema is typically referred to as a mediated schema (or global schema). To answer queries using the information sources the system needs mappings that describe the semantic relationships between the mediated schema and the schemas of the sources. These mappings are the main component of *source descriptions*.

The Information Manifold proposed the method that later became known as the Local-as-View approach (LAV): an information source is described as a *view expression* over the mediated schema. Previous approaches employed the Global-as-View (GAV) approach, where the mediated schema is described as a view over the data sources (see [69, 72] for a detailed comparison of the two).

The immediate benefits of LAV were:

- Describing information sources became easier because it did not involve knowing about other information sources and all the relationships between sources. As a result, a data integration system could accommodate new sources easily, which is particularly important in applications that involve hundreds or thousands of sources.
- The descriptions of the information sources could be more precise. Since the source description could leverage the expressive power of the view definition language, it was easier to describe precise constraints on the contents of the sources and describe sources that have different relational structures than the mediated schema. Describing such constraints is crucial because it enables the system to select a minimal number of data sources relevant to a particular query.

Beyond these contributions, the Information Manifold and its contemporary data integration projects (e.g., [5, 6, 20, 25, 38, 43, 51, 66, 100]) had the following effects.

First, they led to significant research and understanding of how to describe information sources and the tradeoffs, such as expressive power and tractability of query answering. Examples of these issues include the completeness of data sources [1, 39, 71], binding-pattern restrictions on accessing data sources [42, 97, 98], and leveraging data sources that could answer more expressive queries [74, 105]. Later

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permission from the publisher, ACM.
VLDB '06, September 12-15, 2006, Seoul, Korea.
Copyright 2006 VLDB Endowment, ACM 1-59593-385-9/06/09.

work on certain answers and its variants [1, 50] further clarified the semantics of query answering in data integration systems and related the problem to that of modeling incomplete information. The advantages of LAV and GAV were later combined in a mediation language called GLAV [45]. Finally, these languages formed the foundation of data exchange systems [65]. Data exchange systems took a similar approach to mediation between data sources, but instead of reformulating queries these systems materialize a canonical instance of the data in a related source, and queries over that source are answered over the canonical instance.

Second, the progress on studying source descriptions separated the question of *describing* sources from the problem of *using* those descriptions. The process of translating a query posed over the mediated schema into a set of queries on the data sources became known as the problem of query reformulation. With LAV the problem of reformulating a query boiled down to the problem of answering queries using views [26, 29, 37, 67, 90, 92, 94], a problem which was earlier considered in the context of query optimization [24, 68, 103, 112], but started receiving significant attention due to its additional application to data integration (see [53] for a survey). The important difference is that before LAV, reformulation was already built in to the descriptions, making them less flexible and harder to write.

3. BUILDING ON THE FOUNDATION

Given the foundation of source descriptions, research on data integration developed in several important directions.

3.1 Generating Schema mappings

It quickly became clear that one of the major bottlenecks in setting up a data integration application is the effort required to create the source descriptions, and more specifically, writing the semantic mappings between the sources and the mediated schema. Writing such mappings (and maintaining them) required database expertise (to express them in a formal language) and business knowledge (to understand the meaning of the schemas being mapped).

Hence, a significant branch of the research community focused on semi-automatically generating schema mappings [12, 21, 31, 32, 33, 56, 63, 75, 76, 82, 84, 88, 89, 96, 110]. In general, automatic schema mapping is an AI-Complete problem, hence the goal of these efforts was to create tools that speed up the creation of the mappings and reduce the amount of human effort involved.

The work on automated schema mapping was based on the following foundations. First, the research explored techniques to map between schemas based on clues that can be obtained from the schemas themselves, such as linguistic similarities between schema elements and overlaps in data values or data types of columns. Second, based on the observation that none of the above techniques is foolproof, the next development involved systems that combined a set of individual techniques to create mappings [31, 32]. Finally, one of the key observations was that schema mapping tasks are often repetitive. For example, in data integration we map multiple schemas in the same domain to the same mediated schema. Hence, we could use Machine Learning techniques that consider the manually created schema mappings as training data, and generalize from them to predict mappings between unseen schemas. As we describe in Section 4, these techniques are in commercial use today and are providing important benefits in the settings in which they are employed.

A second key aspect of semantic heterogeneity is reconciling data at the instance level [15, 16, 27, 35, 47, 81, 91, 102, 109]. In any data integration application we see cases where the same object in the world is referenced in different ways in data sets (e.g., people, addresses, company names, genes). The problem of reference reconciliation is to automatically detect references to the same object and to collapse them. Unlike reconciling schema heterogeneity, the amounts of data are typically much bigger. Therefore, systems need to rely on methods that are mostly automatic.

3.2 Adaptive query processing

Once a query posed over a mediated schema has been reformulated over a set of data sources, it needs to be executed efficiently. While many techniques of distributed data management are applicable in this setting, several new challenges arise, all stemming from the dynamic nature of data integration contexts.

Unlike a traditional database setting, a data integration system cannot neatly divide its processing into a query optimization step followed by a query execution step. The context in which a data integration system operates is very dynamic and the optimizer has much less information than the traditional setting. As a result, two things happen: (1) the optimizer may not have enough information to decide on a good plan, and (2) a plan that looks good at optimization time may be arbitrarily bad if the sources do not respond exactly as expected. The research on data integration started developing different aspects of adaptive processing in isolation [4, 7, 18, 49, 62, 104, 108], and then came up with unifying architectures for adaptive query processing [59, 61]. It should be noted though that the idea of combining optimization and execution goes even further back to [57].

3.3 XML, XML, XML

One cannot ignore the role of XML in the development of data integration over the past decade. In a nutshell, XML fueled the desire for data integration, because it offered a common syntactic format for sharing data among data sources. However, it did nothing to address the semantic integration issues – sources could still share XML files whose tags were completely meaningless outside the application. However, since it appeared as if data could actually be shared, the impetus for integration became much more significant.

From the technical perspective, several integration systems were developed using XML as the underlying data model [9, 59, 60, 78, 86, 113] and XML query languages (originally XML-QL [30] and then XQuery [23]) as the query language. To support such systems, every aspect of data integration systems needed to be extended to handle XML. The main challenges were typically handling the nested aspect of XML and the fact that it was semi-structured. The Tsimmis Project [25] was the first to illustrate the benefits of semi-structured data in data integration.

3.4 Model management

Setting up and maintaining data integration systems involve operations that manipulate schemas and mappings between them. The goal of Model Management [13, 14, 80] is to provide an algebra for manipulating schemas and mappings, so the same operations do not need to be reinvented for every new context and/or data model. With such an algebra, complex operations on data sources are described as simple sequences of operators in the algebra and optimized and processed using a general system. Some of the operators that have been considered in Model Management include the creation of mappings, inverting and composing mappings [41, 77, 85], merging schemas [93] and schema dif-

ferencing. While we are starting to get a good understanding of these operators, much work remains to be done.

3.5 Peer-to-Peer Data Management

The emergence of of peer-to-peer file sharing systems inspired the data management research community to consider P2P architectures for data sharing [2, 55, 58, 64, 83, 87, 101, 111]. In addition to the standard appeal of P2P architectures, they offered two additional benefits in the context of data integration.

First, it is often the case that organizations want to share data, but none of them wants to take the responsibility of creating a mediated schema, maintaining it and mapping sources to it. A P2P architecture offers a truly distributed mechanism for sharing data. Every data source needs to only provide semantic mappings to a set of neighbors it selects, and more complex integrations emerge as the system follows *semantic paths* in the network. Source descriptions, as developed earlier, provided the foundation for studying mediation in the peer-to-peer setting.

Second, it is not always clear that a single mediated schema *can* be developed for a data integration scenario. Consider data sharing in a scientific context, where data may involve scientific findings from multiple disciplines, bibliographic data, drug related data and clinical trials. The variety of the data and the needs of the parties interested in sharing are too diverse for there to be a single mediated schema. With a P2P architecture there is never a single global mediated schema, since data sharing occurs in local neighborhoods of the network.

3.6 The Role of Artificial Intelligence

Data integration is also an active research topic in the Artificial Intelligence community. Early on, it was shown that Description Logics, a branch of Knowledge Representation, can be used to describe relationships between data sources [22]. In fact, the idea of LAV was inspired by the fact that data sources need to be represented declaratively, and the mediated schema of the Information Manifold was based on Classic Description Logic [17] and on work combining the expressive power of Description Logics with database query languages [10, 70]. Description Logics offered more flexible mechanisms for representing a mediated schema and for semantic query optimization needed in such systems. This line of work continues to recent days (e.g., [19]) where the focus is on marrying the expressive power of Description Logics with the ability to manage large amounts of data.

Research on planning in AI also influenced the thinking about reformulation and query processing in data integration systems beginning with earlier work on the more general problem of software agents [40]. In fact, the idea of adaptive planning and execution dates back to earlier work in AI planning [3, 8].

Finally, as stated earlier, Machine Learning plays a key role in semi-automatically generating semantic mappings for data integration systems. We predict that Machine Learning will have an even greater impact on data integration in the future.

4. THE DATA INTEGRATION INDUSTRY

Beginning in the late 1990's, data integration moved from the lab into the commercial arena. Today, this industry is known as Enterprise Information Integration (EII). (One should not underestimate the value of being associated with a three-letter acronym in industry). The vision underlying this industry is to provide tools for integrating data from multiple sources *without* having to first load all the data into a central warehouse as required by previous solutions. A collection of short articles by some of the players in this industry appears in [54].

Several factors came together at the time to contribute to the development of the EII industry. First, some technologies developed in the research arena matured to the point that they were ready for commercialization, and several of the teams responsible for these developments started companies (or spun off products from research labs). Second, the needs of data management in organizations changed: the need to create external coherent web sites required integrating data from multiple sources; the web-connected world raised the urgency for companies to start communicating with others in various ways. Third, the emergence of XML piqued the appetites of people to share data. Finally, there was a general atmosphere in the late 90's that any idea is worth a try (even good ones!). Importantly, data warehousing solutions were deemed inappropriate for supporting these needs, and the cost of ad-hoc solutions were beginning to become unaffordable.

Broadly speaking, the architectures underlying the products were based on similar principles. A data integration scenario started with identifying the data sources that will participate in the application, and then building a mediated schema (often called a *virtual schema*) which would be queried by users or applications, and building semantic mappings from the data sources to the mediated schema. Query processing would begin by reformulating a query posed over the virtual schema into queries over the data sources, and then executing it efficiently with an engine that created plans that span multiple data sources and dealt with the limitations and capabilities of each source.

Some of the companies coincided with the emergence of XML, and built their systems on an XML data model and query language (XQuery was just starting to be developed at the time). These companies had to address an additional set of problems compared to the other companies, because the research on efficient query processing and integration for XML was only in its infancy, and hence they did not have a vast literature to draw on.

Some of the first applications in which these systems were fielded successfully were customer-relationship management, where the challenge was to provide the customer-facing worker a *global view* of a customer whose data is residing in multiple sources, and digital dashboards that required tracking information from multiple sources in real time.

As with any new industry, EII has faced many challenges, some of which still impede its growth today. The following are representative ones.

Scaleup and performance: The initial challenge was to convince customers that the idea would work. How could a query processor that accesses the data sources in real time have a chance of providing adequate and predictable performance? In many cases, administrators of (very carefully tuned) data sources would not even consider allowing a query from an external query engine to hit them. In this context EII tools often faced competition from the relatively mature data warehousing tools. To complicate matters, the warehousing tools started emphasizing their *real-time* capabilities, supposedly removing one of the key advantages of EII over warehousing. The challenge was to explain to potential customers the tradeoffs between the cost of building a warehouse, the cost of a live query and the cost of accessing stale data. Customers want simple formulas they could apply to make their buying decisions, but those are not available.

Horizontal vs. Vertical growth: From a business per-

spective, an EII company had to decide whether to build a horizontal platform that can be used in any application or to build special tools for a particular vertical. The argument for the vertical approach was that customers care about solving their *entire* problem, rather than paying for yet another piece of the solution and having to worry about how it integrates with other pieces. The argument for the horizontal approach is the generality of the system and often the inability to decide (in time) which vertical to focus on. The problem boiled down to how to prioritize the scarce resources of a startup company.

Integration with EAI tools and other middleware: To put things mildly, the space of data management middleware products is a very complicated one. Different companies come at related problems from different perspectives and it's often difficult to see exactly which part of the problem a tool is solving. The emergence of EII tools only further complicated the problem. A slightly more mature sector is EAI (Enterprise Application Integration) whose products try to facilitate hooking up applications to talk to each other and thereby support certain workflows. Whereas EAI tends to focus on arbitrary applications, EII focuses on the data and querying it. However, at some point, data needs to be fed into applications, and their output feeds into other data sources. In fact, to query the data one can use an EII tool, but to update the data one typically has to resort to an EAI tool. Hence, the separation between EII and EAI tools may be a temporary one. Other related products include data cleaning tools and reporting and analysis tools, whose integration with EII and EAI could stand to see significant improvement.

Despite these challenges, the fierce competition and the extremely difficult business environment after the internet bubble burst, the EII industry survived and is now emerging as an indispensable technology for the enterprise. Data integration products are offered by most major DBMS vendors, and are also playing a significant role in the business analytics products (e.g., Actuate and Hyperoll).

In addition to the enterprise market, data integration has also played an important role in internet search. As of 2006, the large search companies are performing several efforts to integrate data from the multitude of data sources available on the web. Here, source descriptions are playing a crucial role: the cost of routing huge query volumes to irrelevant sources can be very high. Therefore it is important that sources are described as precisely as possible. Furthermore, the *vertical search* market focuses on creating specialized search engines that integrate data from multiple deep web sources in specific domains (e.g., travel, jobs). Vertical search engines date back to the early days of the Web (e.g., companies such as Junglee and Netbot). These engines also embed complex source descriptions.

Finally, data integration has also been a significant focus in the life sciences, where diverse data is being produced at increasing rates, and progress depends on researchers' ability to synthesize data from multiple sources. Personal Information Management [95, 48, 34] is also an application where data integration is taking a significant role.

5. FUTURE CHALLENGES

Several fundamental factors guarantee that data integration challenges will continue to occupy our community for a long time to come. The first factor is social. Data integration is fundamentally about getting people to collaborate and share data. It involves finding the appropriate data, convincing people to share it and offering them an incentive to do so (either in terms of ease of sharing or benefits from the resulting applications), and convincing data owners that their concerns about data sharing (e.g., privacy, effects on the performance of their systems) will be addressed.

The second factor is complexity of integration. In many application contexts it is not even clear what it means to integrate data or how combined sets of data can operate together. As a simple example, consider the merger of two companies and therefore the need for a single system to handle their different stock option packages. What do stock options in one company even mean in the context of a merged company? While this example seems like a business question (and it is), it illustrates the demands that may be imposed on the data management systems to accommodate such unexpected complexity.

Because of these reasons, data integration has been referred to as a problem as hard as Artificial Intelligence, maybe even harder! As a community, our goal should be to create tools that facilitate data integration in a variety of scenarios. Addressing the following specific challenges could go a long way towards that goal.

Dataspaces: Pay-as-you-go data management. One of the fundamental shortcomings of database systems and of data integration systems is the long setup time required. In a database system, one needs to first create a schema and populate the database with tuples before you receive any services or obtain any benefit. In a data integration system, one needs to create the semantic mappings to obtain any visibility into the data sources. The management of dataspaces [44] emphasizes the idea of pay-as-you-go data management: offer some services immediately without any setup time, and improve the services as more investment is made into creating semantic relationships. For example, a dataspace should offer keyword search over any data in any source with no setup time. Building further, we can extract associations between disparate data items in a dataspace using a set of heuristic extractors, and query those associations with path queries. Finally, when we decide that we really need a tighter integration between a pair of data sources, we can create a mapping automatically and ask a human to modify and validate it. A set of specific technical problems for building dataspace systems is described in [52].

Uncertainty and lineage. Research on manipulating uncertain data and data lineage has a long history in our community. While in traditional database management managing uncertainty and lineage seems like a nice feature, in data integration it becomes a necessity. By nature, data coming from multiple sources will be uncertain and even inconsistent with each other. Systems must be able to introspect about the certainty of the data, and when they cannot automatically determine its certainty, refer the user to the lineage of the data so they can determine for themselves which source is more reliable (very much in spirit with how web search engines provide URLs along with their search results, so users can consider the URLs in the decision of which results to explore further). Imbuing data integration systems introspection abilities will widen their applicability and their ability to deal with diverse data integration settings. A recent line of work in the community is starting to address these issues [11, 28, 101, 107].

Reusing human attention. One of the principles to achieving tighter semantic integration among data sources is the ability to reuse human attention. Simply put, every time a human interacts with a dataspace, they are indirectly giving a semantic clue about the data or about relationships between data sources. Examples of such clues are obtained

when users query data sources (even individually), when users create semantic mappings or even when they cut and paste some data from one place to another. If we can build systems that leverage these semantic clues, we can obtain semantic integration much faster. We already have a a few examples where reusing human attention has been very successful, but this is an area that is very ripe for additional research and development. In some cases we can leverage work thats users are doing as a part of their job [32], in others we can solicit some help by asking some well chosen questions [79, 99, 106], and in others we simply exploit structure that already exists such as large numbers of schemas or web service descriptions [36, 56, 76].

6. CONCLUSION

Not so long ago, data integration was considered a nice feature and an area for intellectual curiosity. Today, data integration is a necessity. Today's economy, based on a vast infrastructure of computer networks and the ability of applications to share data with XML, only further emphasize the need for data integration solutions. Thomas Friedman [46] offers additional inspiration with his motto: The World is Flat. In a "flat" world, any product or service can be composed of parts performed in any corner of the world. To make this all happen, data needs to be shared appropriately between different service providers, and individuals need to be able to find the right information at the right time no matter where it resides. Information integration needs to be part of this infrastructure and needs to mature to the point where it is essentially taken for granted and fades into the background like other ubiquitous technologies. We have made incredible progress as a community in the last decade towards practical information integration, and now we are rewarded by even greater challenges ahead!

Acknowledgments

We would like to thank Omar Benjelloun, Anhai Doan, Hector Garcia-Molina, Pat Hanrahan, Zack Ives and Rachel Pottinger for discussions during the writing of this paper. We thank the VLDB 10-year Best Paper Award Committee, Umesh Dayal, Oded Shmueli and Kyu-Young Whang for selecting our paper for the award. Finally, we would like to acknowledge Divesh Srivastava, Shuky Sagiv, Jaewoo Kang and Tom Kirk for their contributions to the Information Manifold.

7. REFERENCES

[1] Serge Abiteboul and Oliver M. Duschka. Complexity of Answering Queries Using Materialized Views. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, 1998.

[2] P. Adjiman, Philippe Chatalic, François Goasdoué, Marie-Christine Rousset, and Laurent Simon. Distributed reasoning in a peer-to-peer setting. In *ECAI*, pages 945–946, 2004.

[3] Jose Ambros-Ingerson and Sam Steel. Integrating planning, execution, and monitoring. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 83–88, 1988.

[4] Laurent Amsaleg, Michael J. Franklin, Anthony Tomasic, and Tolga Urhan and. Scrambling query plans to cope with unexpected delays. In *Proc. of the Int. Conf. on Parallel and Distributed Information Systems (PDIS)*, pages 130–141, 1996.

[5] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu, and Craig A. Knoblock. Retrieving and integrating data from multiple information sources. *International Journal on Intelligent and Cooperative Information Systems*, 1994.

[6] Yigal Arens, Craig A. Knoblock, and Wei-Min Shen. Query reformulation for dynamic information integration. *International Journal on Intelligent and Cooperative Information Systems*, (6) 2/3:99–130, June 1996.

[7] Ron Avnur and Joseph M. Hellerstein. Eddies: Continuously adaptive query processing. In *Proc. of SIGMOD*, 2000.

[8] Greg Barish and Craig A. Knoblock. Learning value predictors for the speculative execution of information gathering plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3–9, 2003.

[9] Chaitanya K. Baru, Amarnath Gupta, Bertram Ludäscher, Richard Marciano, Yannis Papakonstantinou, Pavel Velikhov, and Vincent Chu. Xml-based information mediation with mix. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, pages 597–599. ACM Press, 1999.

[10] Catriel Beeri, Alon Y. Levy, and Marie-Christine Rousset. Rewriting queries using views in description logics. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pages 99–108, Tucson, Arizona., 1997.

[11] Omar Benjelloun, Anish Das Sarma, Alon Y. Halevy, and Jennifer Widom. Uldbs: Databases with uncertainty and lineage. In *Proc. of VLDB*, 2006.

[12] Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[13] Philip A. Bernstein. Applying Model Management to Classical Meta Data Problems. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2003.

[14] Philip A. Bernstein, Alon Y. Halevy, and Rachel Pottinger. A vision of management of complex models. *SIGMOD Record*, 29(4):55–63, 2000.

[15] Indrajit Bhattacharya and Lise Getoor. Iterative record linkage for cleaning and integration. In *Workshop on Data Mining and Knowledge Discovery (DMKD)*, 2004.

[16] Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems Special Issue on Information Integration on the Web*, September 2003.

[17] Alex Borgida, Ronald Brachman, Deborah McGuinness, and Lori Resnick. CLASSIC: A structural data model for objects. In *Proceedings of the ACM SIGMOD Conference*, pages 59–67, Portland, Oregon, 1989.

[18] Luc Bouganim, Franoise Fabret, C. Mohan, and Patrick Valduriez. A dynamic query processing architecture for data integration systems. *IEEE Data Eng. Bull*, 23(2):42–48, 2000.

[19] Diego Calvanese, Giuseppe De Giacomo, Domenico

Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proceedings of KR*, pages 260–270, 2006.

[20] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Information integration: Conceptual modeling and reasoning support. In *CoopIS*, pages 280–291, 1998.

[21] S. Castano and V. De Antonellis. A discovery-based approach to database ontology design. *Distributed and Parallel Databases - Special Issue on Ontologies and Databases*, 7(1), 1999.

[22] T. Catarci and M. Lenzerini. Representing and using interschema knowledge in cooperative information systems. *Journal of Intelligent and Cooperative Information Systems*, pages 55–62, 1993.

[23] Don Chamberlin, Daniela Florescu, Jonathan Robie, Jerome Simeon, and Mugur Stefanascu. XQuery: A query language for XML. Technical report, World Wide Web Consortium, February 2001. Available from http://wwww.w3.org/TR/xquery/.

[24] Surajit Chaudhuri, Ravi Krishnamurthy, Spyros Potamianos, and Kyuseok Shim. Optimizing queries with materialized views. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 190–200, Taipei, Taiwan, 1995.

[25] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Hammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. The TSIMMIS project: Integration of heterogeneous information sources. In proceedings of IPSJ, Tokyo, Japan, October 1994.

[26] Chandra Chekuri and Anand Rajaraman. Conjunctive query containment revisited. *Theor. Comput. Sci.*, 239(2):211–229, 2000.

[27] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWEB*, pages 73–78, 2003.

[28] Nilesh N. Dalvi and Dan Suciu. Answering queries from statistics and probabilistic views. In *Proc. of VLDB*, pages 805–816, 2005.

[29] Jan Van den Bussche. Two remarks on the complexity of answering queries using views. *To appear in Information Processing Letters*, 2000.

[30] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. A query language for XML. In *Proceedings of the World-Wide Web 8 Conference*, pages 1155–1169, 1999.

[31] Hong-Hai Do and Erhard Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proc. of VLDB*, 2002.

[32] AnHai Doan, Pedro Domingos, and Alon Y. Halevy. Reconciling Schemas of Disparate Data Sources: A Machine Learning Approach. In *Proceedings of the ACM SIGMOD Conference*, 2001.

[33] Anhai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Learning to map between ontologies on the semantic web. In *Proc. of the Int. WWW Conf.*, 2002.

[34] Xin Dong and Alon Halevy. A Platform for Personal Information Management and Integration. In *Proc. of CIDR*, 2005.

[35] Xin Dong, Alon Y. Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *Proc. of SIGMOD*, 2005.

[36] Xin (Luna) Dong, Alon Y. Halevy, Jayant Madhavan, Ema Nemes, and Jun Zhang. Similarity search for web services. In *Proc. of VLDB*, 2004.

[37] Oliver M. Duschka and Michael R. Genesereth. Answering Recursive Queries Using Views. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, 1997.

[38] Oliver M. Duschka and Michael R. Genesereth. Query planning in infomaster. In *Proceedings of the ACM Symposium on Applied Computing*, pages 109–111, San Jose, CA, 1997.

[39] O. Etzioni, K. Golden, and D. Weld. Sound and efficient closed-world reasoning for planning. *Artificial Intelligence*, 89(1–2):113–148, January 1997.

[40] Oren Etzioni and Dan Weld. A softbot-based interface to the internet. *CACM*, 37(7):72–76, 1994.

[41] Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang Chiew Tan. Composing schema mappings: Second-order dependencies to the rescue. In *Proc. of PODS*, pages 83–94, 2004.

[42] Daniela Florescu, Alon Levy, Ioana Manolesu, and Dan Suciu. Query optimization in the presence of limited access patterns. In *Proceedings of the ACM SIGMOD Conference*, 1999.

[43] Daniela Florescu, Louiqa Raschid, and Patrick Valduriez. Using heterogeneous equivalences for query rewriting in multidatabase systems. In *Proceedings of the Int. Conf. on Cooperative Information Systems (COOPIS)*, 1995.

[44] M. Franklin, A. Halevy, and D. Maier. From databases to dataspaces: A new abstraction for information management. *Sigmod Record*, 34(4):27–33, 2005.

[45] Marc Friedman, Alon Levy, and Todd Millstein. Navigational Plans for Data Integration. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999.

[46] Thomas Friedman. *The World is Flat: A Brief History of the Twenty-First Century*. Farrar, Straus and Giroux, 2005.

[47] Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, and Cristian-Augustin Saita. Declarative data cleaning: language, model, and algorithms. In *VLDB*, pages 371–380, 2001.

[48] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits: Fulfilling the memex vision. In *ACM Multimedia*, 2002.

[49] G. Graefe and R. Cole. Optimization of dynamic query evaluation plans. In *Proceedings of the ACM SIGMOD Conference*, Minneapolis, Minnesota, 1994.

[50] Gosta Grahne and Alberto O. Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 332–347, 1999.

[51] Laura Haas, Donald Kossmann, Edward Wimmers, and Jun Yang. Optimizing queries across diverse data sources. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, Athens, Greece, 1997.

[52] Alon Halevy, Michael Franklin, and David Maier. Principles of dataspace systems. In *Proc. of PODS*, 2006.

[53] Alon Y. Halevy. Answering Queries Using Views: A Survey. *VLDB Journal*, 10(4), 2001.

[54] Alon Y. Halevy, Naveen Ashish, Dina Bitton, Michael J. Carey, Denise Draper, Jeff Pollock, Arnon Rosenthal, and Vishal Sikka. Enterprise information integration: successes, challenges and controversies. In *SIGMOD Conference*, pages 778–787, 2005.

[55] Alon Y. Halevy, Zachary G. Ives, Jayant Madhavan, Peter Mork, Dan Suciu, and Igor Tatarinov. The piazza peer-data management system. *Transactions on Knowledge and Data Engineering, Special issue on Peer-dta management*, 2004.

[56] B. He and K. Chang. Statistical Schema Matching across Web Query Interfaces. In *Proceedings of the ACM SIGMOD Conference*, 2003.

[57] W. Hong and M. Stonebraker. Optimization of parallel query execution plans in xprs. *Distributed and Parallel Databases*, 1(1):9–32, 1993.

[58] R. Huebsch, B. Chun, J. Hellerstein, B. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. Yumerefendi. The architecture of pier: an internet-scale query processor. In *CIDR*, pages 28–43, 2005.

[59] Zachary Ives, Daniela Florescu, Marc Friedman, Alon Levy, and Dan Weld. An adaptive query execution engine for data integration. In *Proceedings of the ACM SIGMOD Conference*, pages 299–310, 1999.

[60] Zachary Ives, Alon Halevy, and Dan Weld. An xml query engine for network-bound data. *VLDB Journal, Special Issue on XML Query Processing*, 2003.

[61] Zachary G. Ives, Alon Y. Halevy, and Daniel S. Weld. Adapting to source properties in processing data integration queries. In *Proc. of SIGMOD*, pages 395–406, 2004.

[62] Navin Kabra and David J. DeWitt. Efficient mid-query re-optimization of sub-optimal query execution plans. In *Proceedings of the ACM SIGMOD Conference*, pages 106–117, Seattle, WA, 1998.

[63] Jaewoo Kang and Jeffrey Naughton. On schema matching with opaque column names and data values. In *Proceedings of the ACM SIGMOD Conference*, 2003.

[64] Anastasios Kementsietsidis, Marcelo Arenas, and Rene J Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. In *Proc. of SIGMOD*, 2003.

[65] Phokion Kolaitis. Schema mappings, data exchange, and metadata management. In *Proc. of ACM PODS*, pages 61–75, 2005.

[66] Chung T. Kwok and Daniel S. Weld. Planning to gather information. In *Proc. of the 13th National Conf. on Artificial Intelligence (AAAI)*, pages 32–39, 1996.

[67] Eric Lambrecht, Subbarao Kambhampati, and Senthil Gnanaprakasam. Optimizing recursive information gathering plans. In *Proc. of the 16th Int. Joint Conf on Artificial Intelligence(IJCAI)*, pages 1204–1211, 1999.

[68] P. A. Larson and H.Z. Yang. Computing queries from derived relations. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 259–269, Stockholm, Sweden, 1985.

[69] Maurizio Lenzerini. Data Integration: A Theoretical Perspective. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, 2002.

[70] Alon Levy and Marie-Christine Rousset. Combining Horn rules and description logics in carin. *Artificial Intelligence*, 104:165–209, September 1998.

[71] Alon Y. Levy. Obtaining complete answers from incomplete databases. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 402–412, Bombay, India, 1996.

[72] Alon Y. Levy. Logic-based techniques in data integration. In Jack Minker, editor, *Logic-Based Artificial Intelligence*, pages 575–595. Kluwer Academic Publishers, Dordrecht, 2000.

[73] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 1996.

[74] Alon Y. Levy, Anand Rajaraman, and Jeffrey D. Ullman. Answering Queries Using Limited External Processors. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, 1996.

[75] Jayant Madhavan, Phil Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[76] Jayant Madhavan, Philip A. Bernstein, AnHai Doan, and Alon Y. Halevy. Corpus-based schema matching. In *Proc. of ICDE*, pages 57–68, 2005.

[77] Jayant Madhavan and Alon Halevy. Composing mappings among data sources. In *Proc. of VLDB*, 2003.

[78] Ioana Manolescu, Daniela Florescu, and Donald Kossmann. Answering xml queries on heterogeneous data sources. In *Proc. of VLDB*, pages 241–250, 2001.

[79] R. McCann, A. Doan, A. Kramnik, and V. Varadarajan. Building data integration systems via mass collaboration. In *Proc. of the SIGMOD-03 Workshop on the Web and Databases (WebDB-03)*, 2003.

[80] Sergey Melnik, Erhard Rahm, and Phil Bernstein. Rondo: A programming platform for generic model management. In *Proc. of SIGMOD*, 2003.

[81] Martin Michalowski, Snehal Thakkar, and Craig A. Knoblock. Exploiting secondary sources for unsupervised record linkage. In *IIWeb*, 2004.

[82] R.J. Miller, L.M. Haas, and M. Hernandez. Schema Matching as Query Discovery. In *VLDB*, 2000.

[83] Tova Milo, Serge Abiteboul, Bernd Amann, Omar Benjelloun, and Frederic Dang Ngoc. Exchanging intensional xml data. In *Proc. of SIGMOD*, pages 289–300, 2003.

[84] Tova Milo and Sagit Zohar. Using Schema Matching to Simplify Heterogeneous Data Translation. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 1998.

[85] A. Nash, P. Bernstein, and S. Melnik. Composition of mappings given by embedded dependencies. In *Proc. of PODS*, 2005.

[86] J. Naughton, D. DeWitt, D. Maier, A. Aboulnaga, J. Chen, L. Galanis, J. Kang, R. Krishnamurthy, Q. Luo, N. Prakash, R. Ramamurthy, J. Shanmugasundaram, F. Tian, K. Tufte, S. Viglas, Y. Wang, C. Zhang, B. Jackson, A. Gupta, and R. Chen. The Niagara Internet query system. *IEEE Data Engineering Bulletin*, June 2001.

[87] W. S. Ng, B. C. Ooi, K.-L. Tan, and A. Zhou. Peerdb: A p2p-based system for distributed data sharing. In *ICDE*, Bangalore, India, 2003.

[88] Natalya Freidman Noy and Mark A. Musen. Smart: Automated support for ontology merging and alignment. In *Proceedings of the Knowledge Acquisition Workshop, Banff, Canada*, 1999.

[89] Luigi Palopoli, Domenico Sacc, G. Terracina, and Domenico Ursino. A unified graph-based framework for deriving nominal interscheme properties, type conflicts and object cluster similarities. In *Proceedings of CoopIS*, 1999.

[90] Yannis Papakonstantinou and Vasilis Vassalos. Query rewriting for semi-structured data. In *Proceedings of the ACM SIGMOD Conference*, pages 455–466, 1999.

[91] Jose C. Pinheiro and Don X. Sun. Methods for linking and mining massive heterogeneous databases. In *SIGKDD*, 1998.

[92] Lucian Popa and Val Tannen. An equational chase for path conjunctive queries, constraints and views. In *Proceedings of the International Conference on Database Theory (ICDT)*, 1999.

[93] Rachel Pottinger and Philip A. Bernstein. Merging models based on given correspondences. In *Proc. of VLDB*, pages 826–873, 2003.

[94] Rachel Pottinger and Alon Halevy. Minicon: A Scalable Algorithm for Answering Queries Using Views. *VLDB Journal*, 2001.

[95] Dennis Quan, David Huynh, and David R. Karger. Haystack: A platform for authoring end user semantic web applications. In *ISWC*, 2003.

[96] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[97] Anand Rajaraman. *Constructing Virtual Databases on the World-Wide Web*. PhD thesis, Stanford University, 2001.

[98] Anand Rajaraman, Yehoshua Sagiv, and Jeffrey D. Ullman. Answering queries using templates with binding patterns. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pages 105–112, San Jose, CA, 1995.

[99] S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *SIGKDD*, 2002.

[100] V.S. Subrahmanian, S. Adali, A. Brink, R. Emery, J. Lu, A. Rajput, T. Rogers, R. Ross, and C. Ward. HERMES: A heterogeneous reasoning and mediator system. Technical report, University of Maryland, 1995.

[101] N. Taylor and Z. Ives. Reconciling while tolerating disagreement in collaborative data sharing. In *Proc. of SIGMOD*, 2006.

[102] Sheila Tejada, Craig A. Knoblock, and Steven Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *SIGKDD*, 2002.

[103] Odysseas G. Tsatalos, Marvin H. Solomon, and Yannis E. Ioannidis. The GMAP: A versatile tool for physical data independence. *VLDB Journal*, 5(2):101–118, 1996.

[104] Tolga Urhan, Michael J. Franklin, and Laurent Amsaleg. Cost based query scrambling for initial delays. In *Proceedings of the ACM SIGMOD Conference*, pages 130–141, Seattle, WA, 1998.

[105] Vasilis Vassalos and Yannis Papakonstantinou. Describing and using query capabilities of heterogeneous sources. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 256–265, Athens, Greece, 1997.

[106] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of ACM CHI, Vienna, Austria*, 2004.

[107] J. Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. In *Proc. of CIDR*, 2005.

[108] A. N. Wilschut, J. Flokstra, and P. M. G. Apers. Parallel evaluation of multi-join queries. In *SIGMOD-95*, pages 115–126, May 1995.

[109] William E. Winkler. Using the em algorithm for weight computation in the fellegi-sunter model of record linkage. In *Section on Survey Research Methods*, pages 667–671. American Statistical Association, 1988.

[110] Ling Ling Yan, Renee J. Miller, Laura M. Haas, and Ronald Fagin. Data Driven Understanding and Refinement of Schema Mappings. In *Proceedings of the ACM SIGMOD*, 2001.

[111] Beverly Yang and Hector Garcia-Molina. Improving search in peer-to-peer networks. In *ICDCS*, pages 5–14, 2002.

[112] H. Z. Yang and P. A. Larson. Query transformation for PSJ-queries. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 245–254, Brighton, England, 1987.

[113] Cong Yu and Lucian Popa. Constraint-based xml query rewriting for data integration. In *SIGMOD Conference*, pages 371–382, 2004.