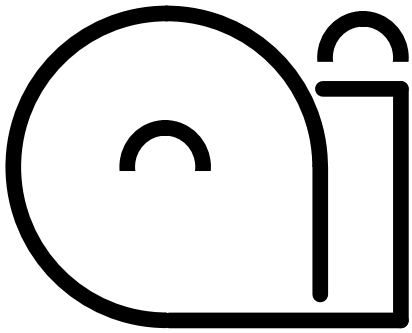


Österreichisches Forschungsinstitut für / Austrian Research Institute for / Artificial Intelligence

TR–2000–28

Christoph Koch, Paolo Petta,
Jean-Marie Le Goff, Richard McClatchey

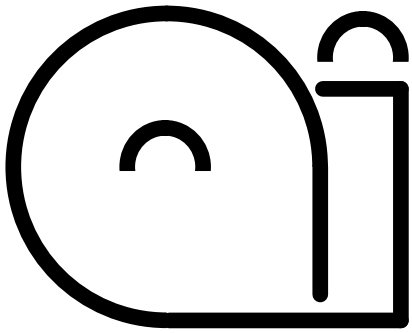
On Information Integration in Large Scientific Collaborations

• Schottengasse 3 • A-1010 Vienna • Austria •

• Phone: +43-1-5336112 •

• mailto:sec@ai.univie.ac.at •

• http://www.ai.univie.ac.at/oefai/ •



Österreichisches Forschungsinstitut für / Austrian Research Institute for / Artificial Intelligence

TR–2000–28

Christoph Koch, Paolo Petta,
Jean-Marie Le Goff, Richard McClatchey

On Information Integration in Large Scientific Collaborations

The Austrian Research Institute for Artificial Intelligence is supported by the Federal Ministry of Education, Science and Culture.

Citation: Koch C., Petta P., Le Goff J., McClatchey R.: On Information Integration in Large Scientific Collaborations. Technical Report, Österreichisches Forschungsinstitut für Artificial Intelligence, Wien, TR-2000-28, 2000

On Information Integration in Large Scientific Collaborations

Christoph Koch[*], **Paolo Petta**[†], **Jean-Marie Le Goff**[‡] and **Richard McClatchey**[§]

Abstract

We discuss the requirements for information integration in large scientific collaborations and arrive at the conclusion that an architecture is needed that follows the declarative paradigm for reasoning completeness, maintainability and reuse of previously encoded knowledge but does not take the classical approach of integrating all sources against a single common "global" information model. Instead, we propose a local-as-view infrastructure that allows to make integrated information from remote sources available to individual (legacy) information systems across multiple different integration models. We discuss our architecture and compare it to previous approaches in the literature.

[*]European Organization for Nuclear Research (CERN), CH-1211 Geneva 23, Switzerland. christoph.koch@cern.ch On leave from Technische Universität Wien, Austria.

[†]Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010 Wien, Austria. paolo@ai.univie.ac.at

[‡]European Organization for Nuclear Research (CERN). jean-marie.le.goff@cern.ch

[§]Centre for Complex Cooperative Systems, University of the West of England, Bristol BS16 1QY, UK. richard.mcclatchey@cern.ch

1 Introduction

There have been two major approaches to information integration, the global-as-view and the local-as-view approach[1] [10]. The mediation process performed by the former is guaranteed to run in polynomial time, while the reasoning problem of the local-as-view approach is presumably very hard even for simple query languages [32, 19, 30]. On the other hand, the latter is more flexible in dealing with mappings of sources, allows to add new sources more easily, and has better completeness properties w.r.t. query evaluation. Also with respect to ease of use and extensibility, the local-as-view approach is usually considered preferable to its alternative.

Previous research on information integration following the local-as-view approach has mainly concerned itself with integrating a number of information sources against a single "global" information model. This is needed for applications such as *data warehousing* or *global information systems* like those used for taming the heterogeneity of the Web. Work on information integration for scientific environments has been so far also no exception [15].

[1]Local-as-view is also called the *declarative* approach to information integration, while global-as-view is sometimes considered *procedural*.

However, in a number of application areas (such as large-scale science projects) the premises of this approach do not hold, so different architectures are required. In particular, it is often the case that one encounters *residual heterogeneity* which does not allow to create a single "global" model against which all sources can be integrated. This may happen for several reasons:

- The modeling task needed for building an appropriate global model may be too hard.
- A large monolithic "global" model may be infeasible because of the steady change of the environment.
- Several legacy systems may be in need of integrated information.
- Autonomy and decentralized control of organizations or groups of people operating the different information systems to be integrated do not permit the use of a "global" integration model.

We go into the details of this problem, which can be encountered in large scientific collaborations as well as with collaborating (virtual) enterprises (e.g. when after a merger or acquisition a number of data warehouses have to be integrated) or any large organization that follows a pragmatic (and minimalistic) approach to making its information systems interoperate. We support our standpoint by putting it into the picture of the Large Hadron Collider (LHC) project [21], a large scientific effort in high-energy physics carried out by a huge and heterogeneous collaboration of research institutions in which CERN is taking part.

Furthermore, we discuss another important characteristic of information integration particularly encountered when scientific and engineering information systems have to interoperate within a greater whole: The hardness of certain data transformation tasks, which makes it necessary to relinquish the declarative framework of logical views in one's favorite flavor of query language and to use special-purpose *data transformation procedures* instead.

We propose an approach to information integration based on the local-as-view paradigm which has the following main features:

- No attempt is made to integrate all data sources against a single "global" integration model. Instead, there may be *many* such destination models. Our goal is to provide middleware that allows each (legacy) information system to make all the outside information available to itself that it needs.
- A *declarative approach* is promoted, allowing the system to automatically find and reuse existing views and mappings between information models for new integration requirements where possible. This is paramount as scientific collaborations may face hundreds of legacy information systems ("islands of information") that may each require data from other systems.

- We will provide a mechanism for smoothly integrating *data transformation procedures* into the system. Such procedures are needed because many data transformation requirements found in scientific or engineering settings cannot be expressed in query languages for which both query answering and query rewriting and containment reasoning have acceptable complexity bounds.

Section 2 summarizes requirements from and observations made in a large scientific project. In Section 3 we review previous work on information integration. Section 4 describes our proposed architecture which will be the basis of an information integration project about to be started at CERN. Finally, in Sections 5 and 6, we discuss our approach and give conclusions.

2 The Setting: A Large Scientific Collaboration

As pointed out in the introduction, a number of issues stand in the way of building a single unified "global" information model (as they exist for data warehouses or global information systems) for a large science project.

Heterogeneity

Heterogeneity lurks behind many corners in a scientific project as there are existing legacy systems as well as largely autonomous groups that build more such legacy systems. Information integration is an issue since many of these individual information systems ("islands of information") require integrated data to be provided from other information systems in order to work.

Scientific collaborations become more and more common due to the fact that nowadays cutting-edge scientific research in areas such as nuclear physics, the human genome or aerospace has become extremely expensive, and has gone global. Such collaborations consist of a number of largely autonomous institutes that independently develop and maintain their individual information systems. This lack of central control fosters creativity and is necessary for political and organizational reasons. However, it leads to problems when it comes to making information systems interoperable. In such a setting, heterogeneity is due to many reasons. Firstly, no two designers would conceptualize a given problem situation in the same way. Furthermore, groups of researchers have fundamentally different ways of dealing with bodies of knowledge, due to different (human) languages, professional background, community or project jargon, teacher and curriculum, or "school of thought". Several subcommunities independently develop and use similar but distinct software for the same tasks. As a consequence, one can assume similar but slightly different schemata. In an environment such as the LHC project at CERN, we can estimate the number of individual information systems to be involved along its lifetime to go into the hundreds. This is the case because even for

the same task, sub-collaborations or individual institutes working on subprojects may independently build several systems.

When it comes to types of heterogeneity that may be encountered in such an environment, it has to be remarked that beyond heterogeneity that is due to discrepancies in conceptualizations of human designers (including polysemy, terminological overlap and misalignment), there is also heterogeneity that is *intrinsic* to the domain; for example, in the environment of high-energy physics experiments (say, a particle detector) as they are carried out at CERN, *detector parts* will be necessarily conceptualized differently depending on the kind of information system in which they are represented. For example, in a CAD system that is used for designing the particle detector, parts will be spatial structures; in a construction management system, they will have to be represented as tree-like structures modeling compositions of parts and their sub-parts, and in simulation and experimental data taking, parts have to be aggregated by associated sensors (readout channels), w.r.t. to which an experiment becomes a topological structure largely distinct from the one of the design drawing. We believe that such differences also lead to different views on the knowledge level, and certainly lead to different database models.

Hardness of Modeling

Apart from the notion of intrinsic heterogeneity that we have given rise to in the previous paragraph, there are a number of other issues that contribute to the hardness of modeling in a scientific domain. Firstly, overall agreement on a conceptualization of a large real-world domain cannot be achieved. Whenever new requirements are discovered or a better understanding of a domain is achieved, there will be an incentive to change the current model. Such change may go beyond pure extension. Instead, existing parts of models will have to be revisited, also invalidating mappings for information integration that rely on these models. Global modeling also fails because of the sheer size of such a scientific domain. In fact, in a project that involves the collaboration of several thousand researchers and engineers, to be able to model the domain would require to have access to all the knowledge in the heads of all the people involved, and this knowledge to be stable. This, however, is an unrealistic conjecture, all the more so in a highly experimental environment.

The Project Lifecycle

It is important to note that large science projects have a lifecycle much like industrial projects; that is, they go through stages such as design, simulation, construction, testing, calibration, deployment, decommissioning, and many more. Such steps have some temporal overlap in practice, but there is a gross ordering. Large science projects persist for large time spans. For example, the LHC project at CERN is expected to be carried on for 20 years. As a consequence, the information systems for some steps of

the lifecycle will not be built until many years from now.

In such an experimental setting, full understanding of the requirements for later information systems can often only be won once that information systems for the current work have been implemented. Nevertheless, since some information systems are already in need of information integration now, one either has to build a global model today which might become invalid later, leading to serious maintenance problems of the information infrastructure (that is, the logical views that map sources), or an approach has to be followed that goes without such a model. Since it is impossible to preview all the requirements of a complex system far into the future, one cannot avoid the need for change through proper a priori design.

The nature of the project lifecycle can also be seen to ease the maintenance problem. The progression of the lifecycle during the lifetime of a project means that at certain points individual stages will end, thus not evolve anymore. When that happens, mappings between information systems of such stages and other, still evolving systems may continue to change. Mappings between systems in stages that have ended, though, will have stabilized as well and may not contribute to maintenance problems anymore.

3 Approaches to Information Integration

As mentioned, research efforts in information integration in the past have mostly focussed on two major directions, the global-as-view approach (GAV) and the local-as-view approach (LAV). We will discuss these two approaches in terms of relational databases, while particularly the first approach has also been evaluated using object-oriented or semi-structured data models [12].

In GAV, global relations are expressed in terms of source relations. Given a global relation $p(\overline{X})$ and sources $p_1, \ldots, p_n$, p might for example be expressed as a (conjunctive) view

$$p(\overline{X}) \ :- \ p_1(\overline{X}_1), \ldots, p_n(\overline{X}_n).$$

(using datalog notation). *Mediators* [31], middleware that provides data from sources for a number of "global" predicates, may themselves access other mediators to fulfill their jobs; i.e., there may be hierarchies of mediators, in which lower-level mediators act as virtual sources to higher-level mediators. Given a query posed in terms of the "global" schema, the query answering process is simple, as it reduces to a simple view unfolding process (that is, views used in a query are replaced by their definitions iteratively until no views remain). However, this simplicity comes with the major drawback that the knowledge about how to map sources to the global predicates to be used in queries has to be provided by the human designer, and requires manual redesign when

data sources are changed or added, or when the *kinds of queries to be expected change* [30].

In the LAV approach, the content of "local" sources is expressed in terms of the predicates of the "global" model. This is achieved through *logical views* (thus the term local-as-view). That is, given "global" predicates $p_1, \ldots, p_n$ and a source v, a *logical view* can be defined in the form

$$v(\overline{X}) \ :- \ p_1(\overline{X}_1), \ldots, p_n(\overline{X}_n).$$

Assuming a query over global predicates $p_1, \ldots, p_m$, this query can be automatically rewritten by the system to contain only source predicates (such as v) instead of the global predicates. This is a hard reasoning problem; however, it spares the human designer to carry out this task by hand. Also, if a new source is to be added to the system or a definition is to be changed, only a single logical view is affected. Furthermore, there are concrete theoretical results (e.g., [19]) about the completeness of querying integrated data; given the same sources, the query rewriting process in LAV may produce queries that have better completeness properties than mediated queries under GAV.

The core theoretical problem in the local-as-view approach is the one of answering queries using logical views. It basically decomposes into the task of finding a rewriting s of an input query q in terms of the sources and checking query containment $s \subseteq q$, that is, that every result tuple of s is also a result of q, for any possible database [32]. These two problems are individually NP-complete already in the simple case of conjunctive queries [7, 19] (i.e., queries that support selections, projections and joins, the basic operations of relational algebra); however, the combined problem remains in NP [19]. Algorithms for solving the query rewriting problem in this case have been proposed in [20, 13, 23, 18]. For more expressive classes of query languages, the problem is harder or even undecidable [19, 29, 9, 27]. However, there is an efficient (linear-time) algorithm for deciding query containment of conjunctive queries if no predicate appears more than twice in any of the two queries [26]. [3, 4, 2] treats the case of query containment given that it is known that the database underlies a certain number of constraints.

Traditionally, under a declarative approach, a number of sources have always been integrated against a common "global" model. Differently, the restriction to integrate against a single information model only is *not* a principal requirement of the global-as-view approach. Rather, mediators can easily be defined to provide integrated information to a number of models within the same system, or to serve as sources to other, higher-level mediators [31].

Some classical implemented research prototypes using the global-as-view approach are TSIMMIS [12], Garlic [6], and HERMES [14]. The systems that follow the local-as-view paradigm include Information Manifold [20], InfoMaster [13], and SIMS [1]. We want to mention a third class of systems that includes various AI systems that follow a declarative approach using a single "global" integration model but which is different from LAV, and usually based

on mappings to or between ontologies based on various formalisms. Systems belonging to this category include OBSERVER [22] and InfoSleuth [11] (and its predecessor CARNOT [28]).

4 Architectural Principles

In the computing environment of a large scientific collaboration there may be a large number of individual information systems. As a consequence, there is a problem in specifying the mappings between them (e.g., for integrating a federation of n databases, $\approx n^2$ mappings are needed). This has profound consequences on the design and maintenance of an information integration environment. Clearly, a declarative approach in which as much of this work as possible can be automatized is strongly advisable so as to limit manual work to cases where new knowledge that has previously been missing has to be put into the system.

Change in our settings leads to problems not previously addressed in research on data warehouses [17] or global information systems because there it is usually safe to assume a clearly specified goal that the integration system to be built has to satisfy, and a fairly complete understanding of the domain to be modeled. Our situation is different. Following a declarative approach to information integration is the only way to avoid the need to revisit all the specified mappings between models every time a change occurs.

Furthermore, in the scientific computing domain, optimal *recall* is needed, that is, all data that could be returned for a given query and a number of source descriptions should be returned. This again requires a local-as-view approach, as theoretical completeness independent of pure design decisions cannot be ascertained in a GAV approach (for an example see [30]).

We want to follow some of the major architectural principles *shared* by both the global-as-view approach and the local-as-view approach (e.g., [10, 25]), namely an architecture based on wrappers around sources that cover structural integration (i.e., the main purpose of wrappers is to translate between the query language of the integration system and the source). On top of the wrappers, we need a reasoning system that allows to rewrite queries in terms of the model of the local information system that the client (user or program) is working in, using specified logical views describing outside sources. However, we do not want to assume a single such information system (with its schema being the "global" model): Rather, there may be many other information systems that are in possession of a number of logical views describing the content of outside sources as well.

Extending the Local-as-view Approach across Multiple Integration Models

We now arrived at an altered problem statement for information integration: In our approach, there are many integration models; each of them has a

number of materialized and virtual predicates and a number of logical views expressing the content of sources from other information systems in terms of the materialized or virtual predicates of the local integration model (i.e., following the local-as-view paradigm). Given the requirement that the set of all logical views in the system is nonrecursive, the integration problem can be transformed into the standard query rewriting problem of the local-as-view approach. This is achieved through simple view unfolding s.t. all logical views representing nonterminal sources are mapped into views that really stand for materialized sources (and which may only be indirectly reachable through a path of mappings starting from the integration model against which a query was issued).

We deem the requirement that the set of all logical views be nonrecursive to be natural and thus not a real restriction. In fact, this requirement is automatically satisfied if we always add logical views one by one and each logical view only contains predicates as subgoals that already have been defined before. This constraint is necessary to avoid the need to decide the containment of a datalog program in a conjunctive query, which is known to be a problem complete for doubly exponential time [9].

Data Transformation Procedures

Certain kinds of data transformations between information systems required for integration are very complicated; such problems occur particularly often in scientific and engineering settings as we discuss them in this paper. Such transformations mostly concern graph reformulation problems. An example of such a complicated data transformation requirement is the need to make construction data (which are stored in a tree of parts of an experiment-as-assembled) available to a domain such as physics event reconstruction where sensor data arrive through a number of readout channels of a physics experiment. This requires a complex projection of data in a large tree into a flat topology of readout channels. Such a transformation cannot be encoded in a query language such as SQL3.

Query languages cannot be reasonably extended to have elegant semantics that cover the required reasoning, as their efficiency would drop dramatically, which particularly in scientific environments with their large amounts of data is not acceptable. Also, as requirements develop, so would query languages have to change. However, these very transformations *are* part of the information integration problem.

Instead of extending data models and query languages, it is more appropriate to encapsulate complex data transformation algorithms into procedures that are then plugged into the integration system using external interface and capability descriptions. The necessary theoretical and algorithmic work to make this possible has already been carried out, as procedures can be simply modeled as logical views with binding patterns [24, 8].

5 Discussion

We have discussed some of the most serious problems encountered in information systems in a scientific setting that are related to heterogeneity. An additional important property of systems in this domain is that they often deal with very large amounts of information. The scalability of an information integration approach is not primarily affected by the size of the data that have to be dealt with; the main factors involved in query rewriting are the size and numbers of queries and views. However, a declarative approach to information integration often provides the appropriate means and encoded knowledge for semantic query optimization, which can have a profound positive impact on the performance of a system that has to deal with large amounts of information. This is clearly one thread of further research.

Clearly, the scalability of an information integration approach is paramount for its practical usefulness. The local-as-view approach has been rightfully criticized for its intrinsic computational complexity. However, local-as-view systems have been presented that use algorithms for answering queries using views that scale up to hundreds of sources (e.g. [20]).

We have restricted our discussion to semantic integration and have left aside structural integration – which is equally essential – for the lack of space. The issues to be resolved in this area, i.e. how to build wrappers and which data models and languages to use, are very important and will be a goal of further research. Clearly, while we have presented our approach in the light of relational data models and query languages, particularly so in scientific and engineering domains, systems have to deal with complex objects often stored in databases using different paradigms than the relational, such as the object-oriented. While some major work following the global-as-view approach has been based on object-oriented or semi-structured data models (e.g. TSIMMIS [12]), there has been less emphasis on this under LAV. However, some recent work (e.g. [5] on regular path queries and [3], which uses a data model that can deal with objects) has been going in this direction.

6 Conclusions

In a large scientific collaboration, hundreds of information systems may coexist, and many of them may need integrated data from a number of other systems. In addition, the requirements of how data may have to be integrated may be highly nontrivial. However, it is not possible to follow the classical global information systems approach as it is impossible to build a single model that fits the needs of all the various client information systems.

We are currently starting an information integration project at CERN. Some of the main principles of the planned system as envisioned today have been summarized in this paper.

We have pointed out the need for following a declarative approach, while at the same time support-

ing an environment that is able to deal with many models that are destinations of integration efforts. Furthermore, we have discussed ways to reason over mappings for reuse. We believe that a declarative approach is the only way in which maintainability under heavy extension and change over long time scales is feasible.

Apart from that, we have explained why we deem it necessary for an information integration infrastructure in a scientific environment to support and smoothly integrate data transformation procedures. While the query rewriting problem using views with binding patterns (and therefore, procedures) is well understood, we still face further design work on how to make procedures available to and interoperable with the integration environment at runtime, on the level of middleware. Possibilities range from collections of procedures in dynamically linked libraries up to a full multi-agent systems approach.

We believe that in large scientific projects it has not been sufficiently realized how important the implications and costs of lack of a principled solution to the information integration problem are. If no investment into an appropriate infrastructure is made, raw manpower will be needed in large quantities instead – and this need will not be recognized to be due to the lack of sophistication in the infrastructure. Information integration problems will only be realized when there are several incompatible databases that users would like to access uniformly, but not if data have to be substantially transformed for use in application areas other than those that they originate from. Much time will be spent on manually translating data; also, similar programs for data transformation will be implemented again and again and again.

This work is meant to lead the way to a solution to this dilemma, which will become critical to the success of science projects as they become ever larger and the problem of getting access to the right information at the right time becomes increasingly complicated.

Acknowledgments

The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry of Education, Science and Culture. The first author was sponsored by the CERN Austrian Doctoral Student Programme.

References

[1] Yigal Arens and Craig A. Knoblock. Planning and Reformulating Queries for Semantically-Modeled Multidatabase Systems. *Proceedings of the First International Conference on Information and Knowledge Management (CIKM 1992)*, Baltimore, MD, 1992.

[2] Catriel Beeri, Alon Y. Levy, and Marie-Christine Rousset. Rewriting Queries Using Views in Description Logics. *Proc. PODS 1997*, pp. 99–108.

[3] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the Decidability of Query Containment under Constraints. *Proc. PODS 1998*, pp. 149–158.

[4] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Daniele Nardi, and Riccardo Rosati. Information Integration: Conceptual Modeling and Reasoning Support. *Proc. CoopIS 1998*, pp. 280–291.

[5] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Answering Regular Path Queries Using Views. *Proc. ICDE 2000*, pp. 389-398.

[6] Michael J. Carey, Laura M. Haas, Peter M. Schwarz, Manish Arya, William F. Cody, Ronald Fagin, Myron Flickner, Allen W. Luniewski, Wayne Niblack, Dragutin Petkovic, John Thomas, John H. Williams, and Edward L. Wimmers. Towards Heterogeneous Multimedia Information Systems: The Garlic Approach. *Proceedings of the Fifth International Workshop on Research Issues in Data Engineering: Distributed Object Management (RIDE-DOM'95)*.

[7] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Conference Record of the Ninth Annual ACM Symposium on Theory of Computing*, pp. 77–90, Boulder, Colorado, 2–4 May 1977.

[8] Surajit Chaudhuri and Kyuseok Shim. "Query Optimization in the Presence of Foreign Functions" In *Proc. of the 19th International Conference on VLDB*, Dublin, Ireland, 1993.

[9] Surajit Chaudhuri and Moshe Y. Vardi. On the Equivalence of Recursive and Nonrecursive Datalog Programs. *Proc. Eleventh ACM Symposium on Principles of Database Systems (PODS 1992)*, pp. 55–66, 1992.

[10] Daniela Florescu, Alon Levy, and Alberto Mendelzon. Database techniques for the World-Wide Web: A survey. *SIGMOD Record* **27**(3):59–74 (1998).

[11] Jerry Fowler, Marian Nodine, Brad Perry, and Bruce Bargmeyer. Agent-based Semantic Interoperability in Infosleuth. *SIGMOD Record* **28**, 1999.

[12] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, J. Widom. "The TSIMMIS approach to mediation: Data models and Languages". In *Journal of Intelligent Information Systems* **8**(2):117-132, 1997.

[13] Michael R. Genesereth, Arthur M. Keller, and Oliver M. Duschka. Infomaster: An Information Integration System. *Proc. SIGMOD Conference 1997*, pp. 539–542.

[14] `http://www.cs.umd.edu/projects/hermes/`

[15] Catherine Houstis, Christos Nikolaou, Spyros Lalis, Sarantos Kapidakis, Vassilis Christophides, Eric Simon, and Anthony Thomasic. Towards a next generation of open scientific data repositories and services. *CWI Quarterly (Centrum voor Wiskunde en Informatica)* **12**(2):111–132, 1999.

[16] Chun-Nan Hsu and Craig A. Knoblock. Reformulating Query Plans for Multidatabase Systems. *Proc. CIKM 1993*, pp. 423–432.

[17] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, 2000.

[18] Chung T. Kwok and Daniel S. Weld. Planning to Gather Information. *Proceedings of AAAI-96*, Portland, OR, August 1996.

[19] Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering Queries Using Views. *Proc. PODS 1995*.

[20] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille: Querying Heterogeneous Information Sources Using Source Descriptions. *Proc. VLDB 1996*, pp. 251–262.

[21] `http://lhc.web.cern.ch/lhc/`

[22] Eduardo Mena, Vipul Kashyap, Amit P. Sheth, Arantza Illarramendi. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. *Proc. CoopIS 1996*, pp. 14–25.

[23] Rachel Pottinger and Alon Y. Levy. A Scalable Algorithm for Answering Queries Using Views. *Proc. VLDB 2000*.

[24] Anand Rajaraman, Yehoshua Sagiv, and Jeffrey D. Ullman. Answering Queries Using Templates with Binding Patterns. *Proc. PODS 1995*, pp. 105–112.

[25] Mary Tork Roth and Peter Schwarz. Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. *Proc. VLDB 1997*.

[26] Y. Saraiya. Subtree Elimination Algorithms in Deductive Databases. Doctoral Thesis, Department of Computer Science, Stanford University, Jan. 1991.

[27] Oded Shmueli. Decidability and Expressiveness Aspects of Logic Queries. *Proc. PODS 1987*, pp. 237–249, 1987.

[28] Munindar P. Singh, Philip Cannata, Michael N. Huhns, Nigel Jacobs, Tomasz Ksiezyk, KayLiang Ong, Amit P. Sheth, Christine Tomlinson, and Darrell Woelk. The Carnot Heterogeneous Database Project: Implemented Applications. *Distributed and Parallel Databases* **5**(2):207–225, 1997.

[29] Divesh Srivastava, Shaul Dar, H. V. Jagadish, and Alon Y. Levy. Answering Queries with Aggregation Using Views. *Proc. VLDB 1996*, pp. 318–329.

[30] Jeffrey D. Ullman. Information Integration Using Logical Views. *Proc. ICDT 1997*, pp. 19–40.

[31] Gio Wiederhold. Mediators in the Architecture of Future Information Systems. *IEEE Computer* **25**(3):38–49, March 1992.

[32] H. Z. Yang and Per-Åke Larson. Query Transformation for PSJ-Queries. *Proc. 13th VLDB 1987*, Brighton, England, pp. 245–254.