Gulliver in the land of data warehousing: practical experiences and observations of a researcher

Panos Vassiliadis

National Technical University of Athens,

Department of Electrical and Computer Engineering, Computer Science Division, Knowledge and Database Systems Laboratory, Zografou 15773, Athens, Greece

pvassil@dbnet.ece.ntua.gr

Abstract

The gap between researchers and practitioners is widely discussed in the IT community. The purpose of this paper is towards showing the issues which occupy both research and practice, and the extent to which these issues have any overlap, in the field of data warehousing. To achieve this goal we first present the current status and tendencies in data warehouse research. Then we list several practical problems as they appear in the relevant literature, based also on our personal experience. Finally, we try to give the relationship of research and practice into a unified big picture.

1. Introduction

The gap between researchers and practitioners is widely discussed in the IT community. The situation regarding data warehousing seems to follow the general pattern where practitioners complain that their practical problems are overlooked by research and researchers are generally unsatisfied by the acceptance of their ideas in industry. Let us quote some abstracts from the results of the previous DMDW workshop [GJSV99]: «Although many solutions were developed for interesting subproblems... combining these partial and often very abstract and formal solutions to an overall design methodology and warehousing strategy is still left to the practitioners...», «... the influence of the research results on the commercial stream of data warehouse products is very limited...», «The gap between data warehouse practice and research became obvious ...». The purpose of this paper is towards showing the issues which occupy research and practice, and the extent to which these issues have any overlap. The ultimate goal is to show possible new areas of research, based on practical problems and at the same time to give an idea of how practice could benefit from research results which seem to be rather ignored.

To this end we will divide the paper in three parts. The first part appears in Section 2, where we present the «good news» for data warehousing and more specifically, the current status of the data warehouse industry in terms of profit and sales, as well as the status of research. To present the status of the research we have listed and classified the papers relevant to data warehousing in three major database conferences during the last five years and tried to show the tendencies of the research based on this study. The second part of the paper deals with problems and failures during data warehouse projects and appears in Section 3. The discussion is based both on the relevant literature (which is surprisingly small) and on the author's personal experiences. Based on the problems which we detect in the previous paragraphs, we then proceed to relate the data warehouse lifecycle with potential problems and solutions proposed by the research community. Finally, we give some concluding remarks on the reasons for the gap between the research and practice communities.

2. The Good News: Money and Research

There are good news for the data warehouse field: sales are increasing with high rates and research is achieving a standard focus on the field. We will briefly summarize the importance of the field by mentioning the financial figures in subsection 2.1 and quickly proceed to subsection 2.2 where we discuss the main subject of this

Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)
Stockholm, Sweden, June 5-6, 2000
(M. Jeusfeld, H. Shu, M. Staudt, G. Vossen, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/

section, which is the status and the tendencies of the research in data warehousing.

2.1 The Money

Selling products related to data warehousing is a business making money. As mentioned in a report by Merril Lynch at the end of 1998 [ShTy98], the estimation was that the data warehousing market was going to expand in the next few years. The numbers are surprisingly large: the data mart market was expected to have a 40% compounded annual growth rate (CAGR) and the RDBMS sales for data warehouse purposes a CAGR of 25%, reaching total sales of $2.2 billion dollars. The OLAP report [Pend00] mentions that the sales have reached $2.5 billion dollars for OLAP tools (including implementation services) and they are expected to grow with 20% rate in 2000 and a CAGR of 19% for a five-year period. Fig. 1 shows the estimated sales, along with the CAGR for six categories of tools.

	1998	1999	2000	2001	2002	CAGR (%)
RDBMS sales for DW	900.0	1110.0	1390.0	1750.0	2200.0	25.0
Data Marts	92.4	125.0	172.0	243.0	355.0	40.0
ETL tools	101.0	125.0	150.0	180.0	210.0	20.1
Data Quality	48.0	55.0	64.5	76.0	90.0	17.0
Metadata Management	35.0	40.0	46.0	53.0	60.0	14.4
OLAP (including implementation services)*	2000	2500	3000	3600	4000	18.9

Fig. 1 Estimated sales in millions of dollars [ShTy98] (*estimates are from [Pend00]).

2.2 The Research

Research in the field of data warehousing is flourishing. Sessions dedicated to data warehousing have appeared in most of the major conferences of the data management discipline. Several workshops have appeared [GJSV99, DOLAP] and there is even a dedicated conference for data warehouse issues [DaWaK].

To obtain an overview of the tendencies of research in the past five years we have selected three prestigious database conferences, namely PODS, SIGMOD and VLDB and classified their papers which are relevant to the data warehouse area. We included any papers we found relevant to data warehousing, except for the ones relevant to data mining (to retain a clear-cut separation between the two fields). We restricted ourselves to just three conferences, since our goal is to give a general feeling of the situation in the research field, rather than conduct a thorough survey of the topic. Based on the content of the papers, we classified them to several categories, shown in Fig. 3. For reasons of better presentation and understanding, we group these categories to larger groups, referred to as “super-categories”. Of course, several papers could fit in more than one categories; still we followed a naïve approach and attributed each paper to only one category. Naturally, we do not claim to be perfect: it is possible that some papers can be left out of our study, or classified under a category which was not the most suitable. We apologize in advance for any such occurrences, although we scrutinized the proceedings to avoid this kind of problems. Also, it is possible that the contribution of a paper in one category, could be accompanied by results in another “correlated” category. We believe that the results which we present are not far from the ones which could be produced from a more elaborate categorization of the paper, which would take this issue into consideration. Still, there is no proof for this statement and the issue remains open (although we believe it is outside the scope of this paper).

As one can see in Fig. 2, the number of papers seems to reach stability. Although the research interest is rather young (only 5 years old) we anticipate that the tendency is to keep a standard number of papers in the major conferences. The drop in the number of papers in 1998 could be easily justified due to the strange explosion in the number of papers relevant to data mining during that particular year. It is very interesting to see that during the last five years there have been 99 relevant papers relevant to data warehousing, which makes 20 papers per year on average.

We have identified 22 categories of research fields where the interest of the researchers has been drawn. In the sequel, we list the most popular out of them (Fig. 4).

- *Data warehouse design*: the problem lies in detecting the set of views to materialize in the data warehouse, in order to achieve the optimal operational cost (i.e., the combined cost of querying and refreshing the contents of the warehouse).
- *Query rewriting*: the problem lies in reusing existing views, to rewrite a query posed over the sources. An alternative name for the problem could be ‘Answering queries using views’.
- *Integration*: this is a wide area covering several issues. The general context is that several sources containing operational data exist in the environment of the data

warehouse and a unique interface must be provided in order to query / update them. The problem of integration is definitely larger than the area of data warehousing, especially with the current advances in the Web technology. Note that in our survey we excluded all papers on integration that seemed clearly oriented towards semi-structured or Web data.

time. One can see a dropping interest in the view technology issues, which is rather normal since people originally thought of data warehouses as collections of materialized views. Although we believe that this attitude is still present in the research community, there seems to be a level of saturation in the problems regarding view technology.

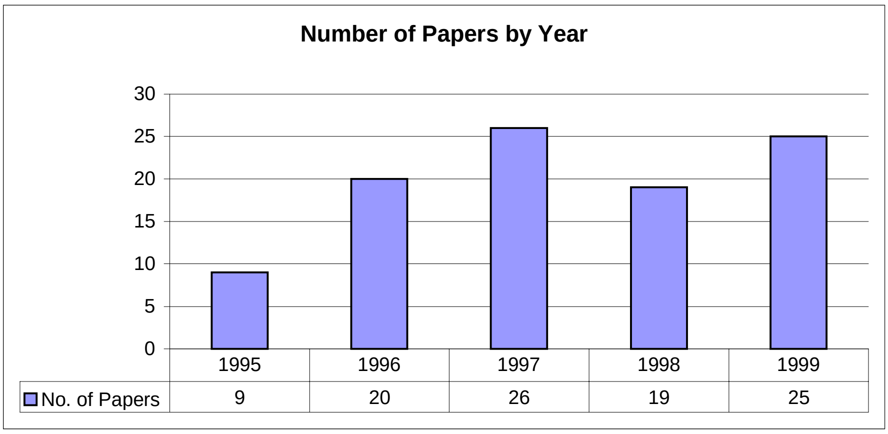


Fig. 2 Number of papers in PODS/SIGMOD/VLDB by year.

- *Processing for relational aggregates*: the area includes structures and algorithms for the efficient processing of aggregate queries. We discriminate this area from query rewriting, in the sense that these papers deal with results that could directly be implemented in a DBMS. We also discriminate the area from the papers involving processing for cubes, which we found more focused in MOLAP databases.
- *View maintenance*: the problem lies in keeping the data warehouse views in accordance with the changes happening in the source data.

The big picture of the area is made clear in Fig. 5, classifying the papers in higher-level super-categories. The classification is based on the grouping of Fig. 3.

The most popular super-categories so far have been *Query Processing*, *View technology*, *Integration* and *Redundancy Exploitation*. Query processing involves all techniques to efficiently process requests and answer queries. It involves six categories and 29% percent of the research performed in the past years. View technology is also a large category, focused on view maintenance techniques as well as the physical data warehouse design process. Integration, which has been previously described, involves producing a single interface for the processing of distributed heterogeneous data, along with query processing techniques for that cause and resolution of conflicts at the schema level. Redundancy exploitation is a field where theoreticians are mostly interested, involving query containment and rewriting.

Probably the most interesting graph is depicted in Fig. 6, grouping the papers by year and super-category. In this figure we see the evolution with respect to the passing of

Category	**Super-Category**
Incomplete information	Incomplete information
Data integration	Integration
Integration in general	
Query processing over integrated data	
Schema integration	
OLAP modeling	OLAP modeling
Caching	Query Processing
Iceberg queries	
Processing for aggregate queries	
Processing for cubes	
Query processing in general	
Top N queries	
Query containment	Redundancy Exploitation
Query rewriting	
Clustering	Storage Management
Indexing	
Storage for cubes	
Storage in general	
Detecting changes in the sources	View Technology
Data warehouse design	
Size estimation for views	
View maintenance	

Fig. 3 Grouping of paper categories to super categories

At the same time, the interest in query processing rises continuously from year to year, probably due to the standard tendency of database researchers towards this field.

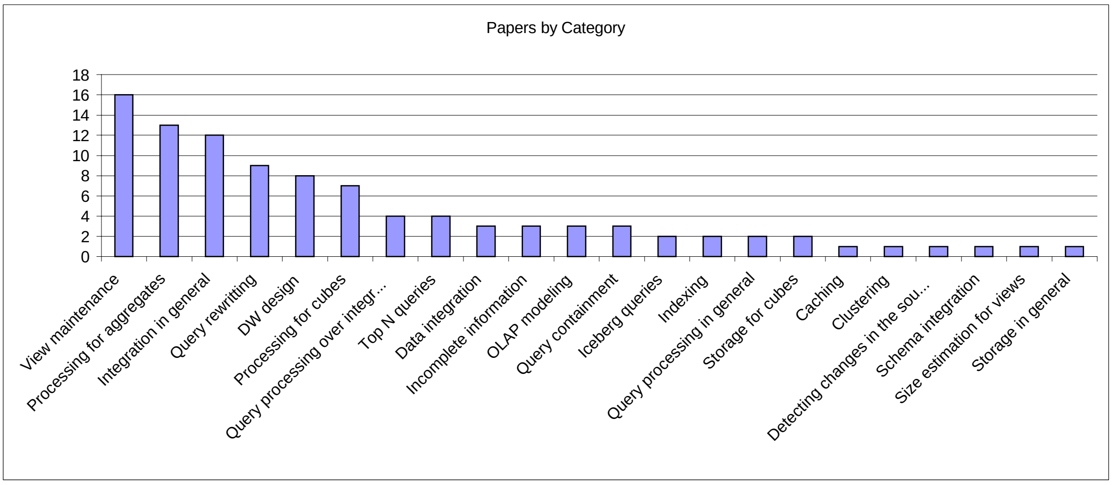
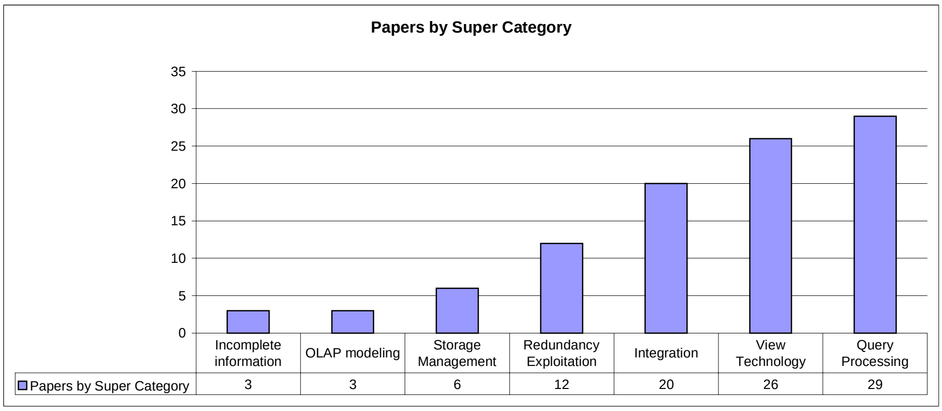
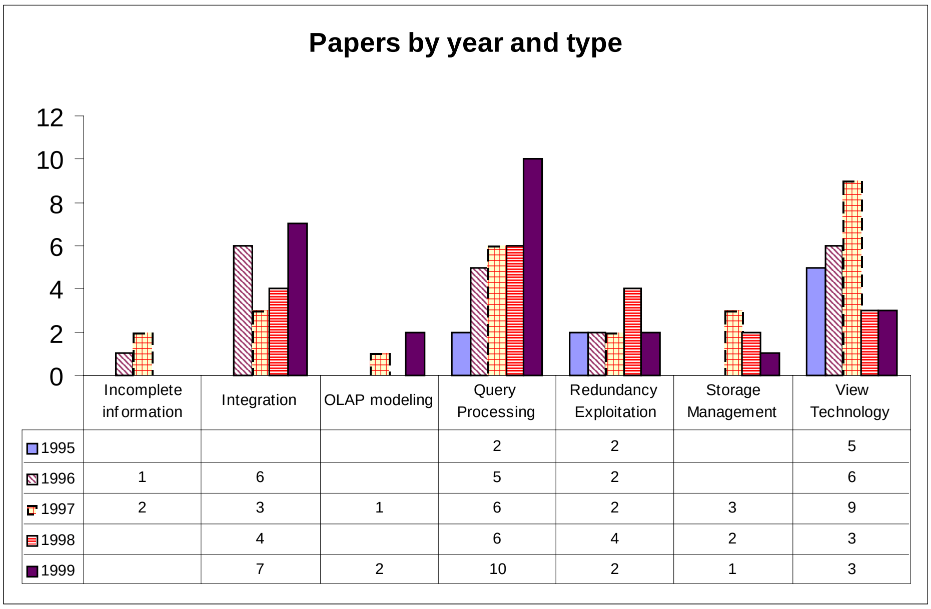


Fig. 4 Number of papers in PODS/SIGMOD/VLDB by category.



	Incomplete information	OLAP modeling	Storage Management	Redundancy Exploitation	Integration	View Technology	Query Processing
□ Papers by Super Category	3	3	6	12	20	26	29

Fig. 5 Number of papers in PODS/SIGMOD/VLDB by super category.



	Incomplete information	Integration	OLAP modeling	Query Processing	Redundancy Exploitation	Storage Management	View Technology
□ 1995				2	2		5
▨ 1996	1	6		5	2		6
▦ 1997	2	3	1	6	2	3	9
▤ 1998		4		6	4	2	3
■ 1999		7	2	10	2	1	3

Fig. 6 Number of papers in PODS/SIGMOD/VLDB by year and super category.

There are areas like incomplete information and storage management which seem to lose interest as time passes. Redundancy exploitation keeps a standard interest due to its dedicated audience of theoreticians. Integration and OLAP modeling seem to gain interest at the same time. The probable reasons for the former are due to the criticism against the materialized nature of data warehousing. As for the latter, it is possible that the lack of a standard OLAP model plays its role to the increasing interest in this category.

3. Data Warehouse Problems and Failures

An objective observer facing the facts of the previous section would directly conclude that the area of data warehousing thrives and the potential for further growth is more than probable. Although this seems to be a quite accurate description of the situation, we argue that a data warehouse project is a great risk and is definitely endangered by several factors. We intend to back up this statement by concrete arguments based both on our personal practical experience in the field and relevant literature.

Category of Factors	**Factors**
Design Factors	Lack of metadata management
	Problematic data engineering
	Unrealistic schema design
	Client tools are neglected or dominate the design
	No design method is used
Technical Factors	Choice of wrong components
	Vendor claims are not tested
	No examination of volume of queries, data sets and network traffic
Procedural Factors	Improper project scope
	Bad use of pilot projects
	User communities are not involved in the design
	No test of new management requirements
	Lack of training for stakeholders
Sociotechnical Factors	Data warehouses cross organizational treaty lines
	Data ownership and access are reconsidered due to the presence of a data warehouse
	The work practices of user communities are affected

Fig. 7 Factors affecting the failure of data warehousing projects [Dema97].

A very good discussion on the problems of data warehousing projects is found in [Dema97]. The paper mentions the logical fact that nobody really speaks about data warehousing failures and goes on to group the reasons for the failure of a data warehousing project into four categories, namely design, technical, procedural and sociotechnical factors (Fig. 7).

According to [ShTy98], the average time for the construction of a data warehouse is 12 to 36 months and the average cost for its implementation is between $1 million to $1.5 million. Data marts are a less risky expenditure, since they cost hundreds of thousands of dollars and take less than a year to implement. Still, if a project of such nature is dependent on so many factors in order to succeed, then the self-contemplating statements on the state-of-the-art on data warehouse management are rather unrealistic. In the sequel, we will take a short look to the particular factors of failure for data warehouse projects. As far as the *design factors* are concerned, there is an obvious deficit in the part of a “textbook” methodology for the design of a data warehouse. There are no standard, or even widely accepted, metadata management techniques[1] or languages, data engineering techniques or design methodologies for data warehouses. Rather, proprietary solutions from vendors, or do-it-yourself advice from experts seem to define the landscape. If we look to the relevant research papers, the picture is disappointing: the three major conferences on data management are not really concerned with issues like metadata management or design methodologies for data warehouses. There exist, though, relevant areas such as the research on the physical data warehouse design and the integration issues. Still, a closer look will reveal that the research seems to target problems not really close to the practical ones. For example, the assumptions made for the design problem are rather unrealistic (knowledge of user queries, their sizes and frequencies) with respect to practical cases. Also, the integration problem is definitely oriented toward a uniform API to distributed sources, i.e., to languages and mechanisms that enable the querying of data. Still, problems like extraction, transformation and cleaning which can take up to 80% of the time spent in the development of a data warehouse [Dema97], seem to be ignored by the research community.

The *technical factors* also reveal the absence of research in the confrontation of practical problems. There exist, of course, standards for the evaluation of software components, but there is a gap in the evaluation and choice of hardware components. As one can see in Fig. 8, hardware costs up to 60% of a data warehouse budget (disk, processor and network costs). Critical software (DBMS and client tools) which is purchased (and not developed in-site) take up to 16% of the budget. There are no papers to our knowledge that deal with issue of hardware/software selection for data warehouse environments. As for the estimation of the sizes of queries, data sets and network traffic, a closer look to the

[1] [ShTy98] reports that the lack of a common metadata standard (despite the existence of the MDIS standard at the end of 1998) is the basic source for concern for metadata management tools.

appendix will reveal only one (!) paper on the estimation of view sizes [SDNR96]. The fact that the average size of data warehouses increases year by year makes the problem even tougher. Back in 1996 the average data warehouse size was estimated to be around 250 GB. In today's data explosion there is even talk about scientific data warehouses of 40 TB [SGKT00]. This means that despite Moore's law and the drop in the cost of storage units, size is still a problem for data warehousing. The increasing number of users increases the complexity of the problem. [ShTy98] mentions the case of a data warehouse involving 20.000 users with an annual increase of 2.000 users per year. Obviously, estimating the size of materialized views or user queries is of great importance, in this context.

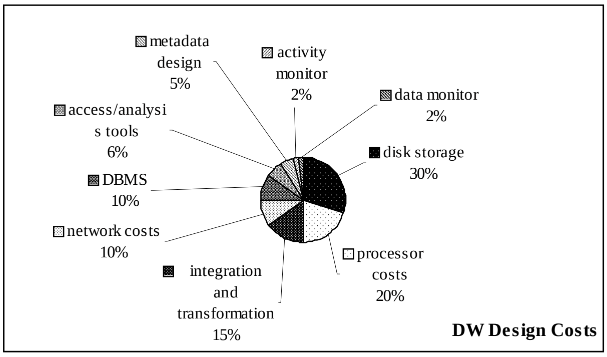


Fig. 8 Data warehouse design costs according to Bill Inmon [Inmo97]

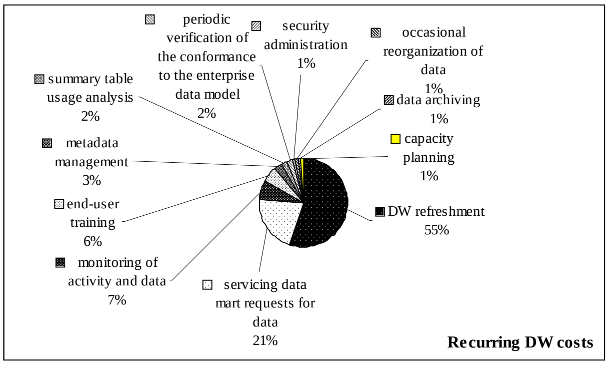


Fig. 9 Data warehouse recurring costs according to Bill Inmon [Inmo97]

The procedural and sociotechnical reasons are not really technical reasons with which we should expect the research society to deal with. We mention them for reasons of completeness and in order to show how sensitive a project like the construction of a data warehouse is. The *procedural factors* involve reasons for deficiencies concerning the deployment of the data warehouse. Apart from classical problems in IS management, it is important to notice that the role of user communities is crucial: the end-users must be trained to the new technologies and included in the design of the warehouse. We refer the interested reader to [Gree00, Dema97] for further probing on this very interesting issue.

As for the *sociotechnical issues*, it is also very interesting to briefly discuss the relevant factors, since there is very little reference to this kind of problems in the literature. According to [Dema97], breaking the organizational treaties is a consequence of the fact that the data warehouse may reorganize the way the organization works and intrude the functional or subjective domain of the stakeholders. For example, imposing a particular client tool to the users invades the users' desktop, which is considered to be their personal "territory". The problems due to the data ownership and access are grouped in two categories. First, data ownership is power within an organization. Any attempt to share or take control over somebody else's data is equivalent with loss of power of this particular stakeholder. Secondly, no division or department can claim to possess 100% clean, error-free data. The possibility of revealing the data quality problems within the information system of the department is definitely frustrating for the affected stakeholders. Finally, the invasion in the work practice reduces to the psychological reason that no user community seems to be really willing to shift from gut feeling or experience to objective, data driven management (see [Dema97] for a broader discussion). To top the entire skepticism about the non-technical problems and reasons of failure, *ethical considerations* can be added to the big picture of data warehousing. In [Smit97] several such thoughts are presented: Is it fair to use customers' data to harm their relationships with their suppliers/customers? Is it fair to use such data to intrude your customers' know-how? Is it fair to use customers' data to change the structure of your organization in a way that is detrimental to your customers? Is it fair to use personal data of individual customers without any prior notice?

Most of the aforementioned reasons for failure are backed up from other testimonial literature (e.g., [Paul97], [ShTy98]).

3.1 Personal Experience

The author has been involved in both research and practical data warehouse projects, during the last six years. Our research experience was mainly the European basic research project "DWQ: Foundations for Data Warehouse Quality" [JaVa97]. Obviously, some of the criticism and comments in this paper are influenced by the research conducted in this project. We apologize for this clear bias; still, since this paper presents the author's personal judgments we believe that we should make clear what has possibly influenced our opinion.

The author has also been involved in three rather small practical data warehouse projects. The first involved loading data from all the health centers (i.e., hospitals, provincial medical centers and other special kinds of centers) in Greece into an enterprise data warehouse. The loading of data was performed annually and the querying was supposed to be performed mostly by pre-canned reports. Still, quite a lot of flexibility was provided to the user to filter, roll-up drill-down and drill-through the data. The data warehouse was rather small and its construction took around 12 months. The major problems encountered were not technical, since (a) the size of data was not so big, (b) the refreshment window was not a problem and (c) there was no real problem in reconciling the source data. Still, there were major problems with the administration team of the legacy system due to the following reasons:

– *Lack of training of the target administration team*. The people administering the legacy COBOL-based system were the ones who would administer the new system, too. Still, this was their first experience with the relational technology and this was definitely a cultural shock for them.
– *Involvement of the administration team of the legacy system in the design of the new system*. Although it is clear that no data warehouse can be built without the involvement of the source administrators, our personal experience suggests that this should be limited to the construction of the data warehouse enterprise model (or even only to the reverse engineering of legacy data). Any attempt to include people without the proper background in a process they do not really understand, seems to jeopardize the while effort, rather than train / accustom them to the new system.
– *Poor quality of legacy data*. The toughest problem in this particular problem was the cleaning of data. Each circuit in the schema seemed to be a sui generis situation. Most important, we faced big difficulties trying to convince the administrators of the legacy system for the poor quality of their data. *Another big problem was the detection of which sources were reliable*. In a COBOL system there is too much redundancy, since each application uses its own data store. Every now and then, the different COBOL files are synchronized, although this is not always 100% successful. When building the data warehouse, it is a hard task to determine the quality of each candidate data source.
– *Data warehouse evolution*. The business rules for the data warehouse are likely to change even during the construction of the warehouse itself. The problem is hard, since it (a) brings the whole project back in schedule and cost, (b) it psychologically frustrates the development team and (c) the lack of a metadata management repository makes it almost insurmountable to detect which part of the database or the applications has to be synchronized with the new situation. Imagine, for example, the case where the primary key of a fact table has to change a couple of weeks before completing the project. In our case, we had to detect and evolve around 50 pre-canned reports as well as all the refreshment processes of the fact table and the materialized views that used it. It was only the consistent naming of all the software components that helped us perform this task.

Note that the project experienced no political problems. The data warehouse was requested by the same department that previously owned the data. The new system would still be under the control of this particular department and would thus synchronize and clean the information they provided to higher management. Note also that we never came to direct contact with the end-users: this was supposed to be a task undertaken by the administration team of this particular department. Thus, we have no knowledge for the real success of this project.

In a second occasion, we had to build a data warehouse with pension data. The data were to be updated monthly and used by pre-canned reports. The size of data involved a few million rows per month. The source data relied again on a COBOL-based legacy system. The project lasted nine months and could be characterized more as the construction of a data mart rather than the construction of a full data warehouse. In this case, *the major problem was of political nature*: different departments were involved in the ownership of the information. The people administering the legacy system were definitely affected by the construction of the warehouse. These people

– would lose the full ownership of the information (which translates to sheer power in the IT department);
– would have to take care of the transportation and conversion of the data in their own system (which means extra workload for both people and systems) and
– any deficiencies of the information they produced would be revealed (a fact of enormous importance and effect in the public sector).

Bearing all this in mind, it quite straightforward to understand the difficulties raised. Moreover, it was interesting to see that the higher management, although committed to the idea of constructing the data warehouse, was unable to force things to happen and had to take an approach that peacefully resolved any problems that occurred, in order to salvage the project from total failure.

Another problem we had to face in this project was *the difficulty in constructing the extraction and cleaning software*. The extraction of data from the legacy systems is a highly complex, error-prone and tiring procedure. To give an idea of the problem, let us mention the case where the problem involved detecting relevant data from a COBOL file, converting EBCDIC to ASCII format, unpacking the packed numbers, reducing all address fields to a standard format and loading the result into a table in the data warehouse. Apart from the standard tool offered by Oracle for these purposes (SQL*Loader) we did not use any commercial tool for these tasks. This seems to be the tactics followed by the majority of data warehousing projects. According to [ShTy98] most of the companies contacted for their survey, estimate that more than 1/3 of the cost and time are spent to ETL tasks during the development process. Still, in spite the obvious importance of this process, the vast majority of them developed their own application instead of using a tool to facilitate the process. [ShTy98] also reports that data quality products are expensive and hard to use. Based on the problem of time and budget constraints for a data warehouse project, [ShTy98] estimates that such products are going to modestly foster in the next few years (with the almost the lowest CAGR of all the product categories).

Political problems were apparent in a third case where the project failed. The organization possessed four legacy systems, all of different kind (COBOL, Excel and dBase files as well as a relational system). A pilot data mart involving a subset of one of the legacy systems had already been successful and the management was enthusiastic about the whole idea. Still, the project failed, before it even started. As we had also observed in the previous case, it seems to be a common phenomenon that the people administrating the legacy system take a little time until they understand what is politically happening to them once a data warehouse is built. In this particular case the reaction was quick and absolute: no data were to be given from the largest legacy system, since its administrators simply refused to provide them. The project was thus canceled. The lesson we learnt in this case is that it takes more than an enthusiastic management and a successful pilot for a data warehouse to succeed. Later, we learned that the warehouse project started again, still we have no knowledge for the fate of this new effort.

3.2 Relationship between Practical Problems and Research Issues

In this section we would like to relate the data warehouse lifecycle with potential problems and solutions offered by technology to tackle this particular problems. The first problem in this task is the lack of a concrete "textbook-style" methodology. Reading the two classical books on data warehousing [Inmo96, Kimb96] one gets the feeling that they provide tips and solutions for fragments of the whole process, rather than a concrete methodology for the data warehouse practitioner. We use as a template methodology the one proposed in an Appendix of [Inmo96] and try to relate it to potential problems and technological solutions offered by research. We list only the aforementioned problems and research categories. Again, we do not claim that either list is exhaustive, but rather indicative.

As we can see in Fig. 10 there are areas where research has contributed a lot to the practical problems. For example, several issues of the view technology super-category are (or at least, can be) somehow used by practitioners in data warehouse design and implementation. Also, several topics of the integration super-category can be exploited in practical cases.

Apart from these successes, there are two issues that clearly depict the gap between research and practice. On the one hand, *there is an unclear picture with respect to the extent that practice has exploited the results of research*. Query processing and storage management are two research fields aiming to empower the technology providers (i.e., the software and hardware vendors) with better techniques for the storage and acquisition of information. To our knowledge, it is not clear to which extent have this results been incorporated in commercial products. The extent to which results in the field of incomplete information and redundancy exploitation can be exploited is another pending issue. The former seemed to be a rather promising research field but the lack of research interest in the later years seems to be discouraging for its further exploitation. The latter is a clear field but we believe that its practical exploitation will take time to be implemented. As far as the data warehouse designer is concerned, the cases where the determination of the intentional subsumption of two data stores is useful is rather limited. Instead, it is the extensional properties of the data source that count (an issue not really apparent in database research). Finally, OLAP modeling could be very useful in the logical definition of the data warehouse, but the lack of a standard multidimensional hierarchical model seems to drive designers to ad-hoc, proprietary solutions. Still, the relational counterpart, in the form of the ER diagram and the relational model, seems to be a promising precedent.

On the other hand of course, there seem to be rather big gaps in the table of Fig. 10, with respect to *steps in the data warehouse lifecycle which are not supported by the conducted research*. The data model analysis could be clearly helped by improved techniques of metadata management (and standards) as well as by data engineering methods that enable the designer to understand and model data and processes better. Breadbox analysis and technical assessment are clearly under-estimated by the research community. Techniques to analyze data volume, network traffic, relevance and quality of software components would greatly be appreciated by data warehouse designers. The extraction process is also suffering from lack of help from the research community: as already mentioned, most research performed has been dedicated to *what* should be extracted (instead of *how* this extraction is performed). The practical aspects of extraction are clearly neglected (e.g. declarative languages and visual interfaces for the management of the extraction process, automation of the extraction programs, etc.). The problem is vast due to the sui generis nature of each kind of source (ASCII data are different from ISAM or database data) and of each particular source itself. The peculiarities of the conversion process are also –more or less- neglected.

Phase	Lifecycle step	Description	Potential Problems	Solutions offered by the research
	Decision to built the warehouse		Improper project scope	
			Bad use of pilot projects	
			Data warehouses cross organizational treaty lines	
			Data ownership and access are reconsidered	
			The work practices of user communities are affected	
Design	Data Model Analysis	Conceptual and logical model	No design method is used	OLAP modeling
			User communities are not involved in the design	
			Lack of metadata management	
			Problematic data engineering	
			Lack of training of the target administration team	
			Excessive involvement of the administration team of the legacy system in the design of the new system	
	Breadbox Analysis	Size estimation for the data	No examination of volume of queries, data sets and network traffic	Size estimation for views
	Technical Assessment	Definition of technical requirements	No test of new management requirements	
	Technical Environment Preparation	Definition of network, storage, OS, software components, etc.	Client tools are neglected or dominate the design	
			Choice of wrong components	
			Vendor claims are not tested	
	Subject Area (per subject)	Decision which subject area to populate		
	Source System Analysis (per subject)	Identification of proper source for the data and reverse engineering of the selected source	Difficulty in determining which source is appropriate, due to quality problems	
	Data Warehouse Database Design	Physical database design for the data warehouse	Unrealistic schema design	Physical DW design, Indexing
DW implementation	Program Specifications (per subject)	Formalize the interface between source data and warehouse	Data warehouse evolution	View Maintenance, Data & Schema Integration
	Programming (per subject)	Construction of the appropriate software for ETL purposes	Poor quality of legacy data	Detecting changes in the sources
			Difficulty in constructing the S/W correctly	
	Population (per subject)	Load the warehouse with data	Difficulty in using the data quality tools	
Report Implementation (per report)	Determine data needed	Decide which part of the data warehouse covers the data for the report		
	Program to extract data	Write a program to get the data from the DW		
	Customize the data	Customize the data for the user's intuition		
	Refine the analysis	Is the report suitable for what it was intended?		
	Usage	Use the reports	Lack of training for stakeholders	
	Institutionalize	Should the report be institutionalized?		

Fig. 10 Data warehouse lifecycle steps, potential problems and solutions offered by the research community

We believe that a turn in the interest of the research community from the virtual querying of distributed heterogeneous data sources and the intentional reconciliation to practical aspects of extraction of materialized data could benefit the practitioners a lot. Finally, it seems to be unclear, to which extent procedural and sociotechnical factors (involved mostly at the beginning and the end of a data warehouse project) could benefit from the use of new technology, suggested by research results. This fuzziness alone, is a very good reason for research from the part of academia. As reported in [SJSV99] significant contribution could also be made from business administration sciences, e.g., in the way the data warehouse in introduced in the corporation.

4. Conclusions

Normally, this is the place for an optimistic message, or the ringing of the bell. For a change, we will do neither. There are two issues, though, we would like to touch, as concluding remarks. First, is it really the case, that research and practice are so much apart? In our humble opinion, the answer is negative. Although research has targeted only a fraction of the possible areas where practitioners could need assistance, the technological contribution of the research society is significant. For example, let us mention the case of data warehouse refreshment. Despite the problems in the extraction step, which we have already mentioned, the refreshment process is of significant importance for the proper operation of the data warehouse. The recurring costs for data warehouse refreshment come up to 55% of the overall cost for running a data warehouse (Fig. 9). Still, the contribution is only in areas where the existing technology could be enhanced, without any methodological results or groundbreaking research in new fields.

Secondly, why is it that researchers are found away from the practical problems of data warehousing? This is a widely discussed issue (e.g., there is a standard debate in the Communications of the ACM magazine). We point only a few reasons that have come to our attention:

- It is possible that several researchers are not aware of the real-world problems. The major motivation for writing this paper was a discussion with a visiting researcher to our department. This person has devoted too much time, programming and energy to the data warehouse design problem. Still, he believed that the data warehouse is simply a set of “DECLARE VIEW” statements. Clearly, this was a problem of lack of direct contact with practical problems.
- It is not always rewarding, in terms of research, to deal with practical problems. The extraction process of our case study, which we mentioned in Section 3 might give an example for this statement. Which researcher would feel happy to work on such a ‘dirty’ problem, knowing that it will be too hard to make publications out of such an effort. It is not strange, thus, that so much theoretical work has been devoted to view maintenance issues, with respect to *what* should be propagated to the warehouse, while few research efforts have been made as to *how* this extraction and propagation is to be made. We believe that it would be really hard for a paper concerning practical automation techniques for the data extraction task to convince an academic audience. The last Asilomar report [BBC+98] states the need for “groundbreaking” instead of “delta” research; still, it is not clear which practical issues concerning data warehousing are qualified under this definition.
- The rules that govern the behavior of science are applied also in the case of data warehousing. It is commonly agreed that it is the Paradigm that determines the interesting problems and not vice-versa. In our case, the paradigm set by the papers of Codd and Selinger et al., has –more or less- set the landscape for the research in the data warehouse field, too. For example, although too much work has been devoted to query processing for aggregate queries, these queries are still treated in isolation. Still, an OLAP session is a *sequence* of steps, which have some *logical interrelationship*. How many papers do you know dealing with this particular property of OLAP? As another example, we simply remind the technical and design problems mentioned in Section 3, which although being of great importance are not addressed by the research. We believe that one of the reasons for this situation is the non-standard nature of these problems, which puts them outside the scope of the relational paradigm.

As for the future, it is hard to make any predictions. Is data warehousing going to be virtual (making all our comments on the integration problem void, and the research conducted in this field highly useful)? Is there going to be a shift towards methodological issues in data warehouses? Are the gaps in Fig. 10 going to be filled? Although the answer is ‘I don’t know’ –at least from our part- it is a challenging issue to work on these issues, contributing thus, to the closing of the gap between research and practice and making data warehousing an easier and less risky endeavor for practitioners and organizations.

5. References

[BBC+98] P.A. Bernstein, M.L. Brodie, S. Ceri, D.J. DeWitt, M.J. Franklin, H. Garcia-Molina, J. Gray, G. Held, J.M. Hellerstein, H.V. Jagadish, M. Lesk, D. Maier, J.F. Naughton, H. Pirahesh, M. Stonebraker, J.D. Ullman. The Asilomar Report on Database Research. SIGMOD Record 27(4): 74-80 (1998)

[Comp96] ComputerWire Inc. Data Warehouse

Economics: ROI doubts? Data Warehouse Tools Bulletin, November 1996. Available at http://www.computerwire.com/dwtb/free/2112_182.htm

[DaWaK] International Conference on Data Warehousing and Knowledge Discovery (DaWaK). http://www.informatik.uni-trier.de/~ley/db/conf/dawak/index.html

[Dema97] M. Demarest. The politics of data warehousing. Available at http://www.hevanet.com/demarest/marc/dwpol.html

[DOLAP] International Workshop on Data Warehousing and OLAP (DOLAP). http://www.pages.drexel.edu/faculty/songiy/dolap.html, http://www.informatik.uni-trier.de/~ley/db/conf/dolap/index.html

[GJSV99] S. Gatziu, M.A. Jeusfeld, M. Staudt, Y. Vassiliou. Design and Management of Data Warehouses - Report on the DMDW'99 Workshop. SIGMOD Record **28**(4), December 1999. Refers to the International Workshop DMDW'99 at CAiSE'99, Heidelberg, Germany, June 1999. Online version available at http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-19

[Gree00] L. Greenfield. Data Warehousing Political Issues. February 2000. Available at http://www.dwinfocenter.ord/politics.html

[Inmo96] W.H. Inmon. Building the Data Warehouse. John Wiley & Sons, March 1996.

[Inmo97] B. Inmon. The Data Warehouse Budget. DM Review Magazine, January 1997. Available at http://www.dmreview.com/master.cfm?NavID=55&EdID=1315

[JaVa97] M. Jarke, Y. Vassiliou. Foundations of data warehouse quality – a review of the DWQ project. In *Proc. 2nd Intl. Conference Information Quality (IQ-97)*, Cambridge, Mass., 1997. Available in http://www.dblab.ece.ntua.gr/~dwq

[Kimb96] R. Kimbal. The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, February 1996.

[Paul97] L.G. Paul. Anatomy of a failure. CIO Magazine. November 15, 1997. Available at http://www.cio.com/archive/enterprise/111597_data_content.html

[Pend00] N. Pendse, February 24, 2000. The OLAP Report. Available at http://www.olapreport.com/Market.htm.

[SDNR96] A. Shukla, P. Deshpande, J.F. Naughton, K. Ramasamy. Storage Estimation for Multidimensional Aggregates in the Presence of Hierarchies. In Proceedings of 22nd International Conference on Very Large Databases (VLDB), Mumbai India 1996.

[SGKT00] A. Szalay, J. Gray, P. Kunszt, A. Thakar. Designing and Mining Multi-Terabyte Astronomy Archives. SIGMOD Conference 2000. Also available at http://www.research.microsoft.com/~gray/

[ShTy98] C. Shilakes, J. Tylman. Enterprise Information Portals. Enterprise Software Team. November 1998. Available at www.sagemaker.com/company/downloads/eip/indepth.pdf.

[Smit97] J. Smith. Do Data Warehouses Challenge Fair Play? Beyond Computing, **6**(4), May 1997. Available at www.beyondcomputingmag.com/archive/1997/5-97/ethics.html

Appendix

Paper	**Category**
1995 – PODS	
Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, Divesh Srivastava. Answering Queries Using Views. 95-104.	Query rewritting
Anand Rajaraman, Yehoshua Sagiv, Jeffrey D. Ullman. Answering Queries Using Templates with Binding Patterns. 105-112.	Query rewritting
H. V. Jagadish, Inderpal Singh Mumick, Abraham Silberschatz. View Maintenance Issues for the Chronicle Data Model. 113-124.	View maintenance
1995 - SIGMOD	
Ashid Gupta, Inderpal Singh Mumick, Kenneth A. Ross. Adapting Materialized Views after Redefinitions. 211-222.	View maintenance
Yue Zhuge, Hector Garcia-Molina, Joachim Hammer, Jennifer Widom. View Maintenance in a Warehousing Environment. 316-327.	View maintenance
Timothy Griffin, Leonid Libkin. Incremental Maintenance of Views with Duplicates. 328-339.	View maintenance
James J. Lu, Guido Moerkotte, Joachim Schü, V. S. Subrahmanian. Efficient Maintenance of Materialized Mediated Views. 340-351.	View maintenance
1995 - VLDB	
Weipeng P. Yan, Per-Åke Larson. Eager Aggregation and Lazy Aggregation. 345-357.	Processing for aggregates
Ashish Gupta, Venky Harinarayan, Dallan Quass. Aggregate-Query Processing in Data Warehousing Environments. 358-369.	Processing for aggregates
1996 - PODS	
Alon Y. Levy, Anand Rajaraman, Jeffrey D. Ullman. Answering Queries Using Limited External Processors. 227-237.	Query rewritting
1996 - SIGMOD	
Richard Hull, Gang Zhou. A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches. 481-492.	Data integration
Venky Harinarayan, Anand Rajaraman, Jeffrey D. Ullman. Implementing Data Cubes Efficiently. 205-216.	DW design
Leonid Libkin, Rona Machlin, Limsoon Wong. A Query Language for Multidimensional Arrays: Design, Implementation, and Optimization Techniques. 228-239.	Processing for cubes
Sudhir Rao, Antonio Badia, Dirk Van Gucht. Providing Better Support for a Class of Decision Support Queries. 217-227.	Query processing in general
Kenneth A. Ross, Divesh Srivastava, S. Sudarshan. Materialized View Maintenance and Integrity Constraint Checking: Trading Space for Time. 447-458.	View maintenance
Latha S. Colby, Timothy Griffin, Leonid Libkin, Inderpal Singh Mumick, Howard Trickey. Algorithms for Deferred View Maintenance. 469-480.	View maintenance
1996 - VLDB	
Peter Scheuermann, Junho Shim, Radek Vingralek. WATCHMAN : A Data Warehouse Intelligent Cache Manager. 51-62.	Caching
Alon Y. Levy, Anand Rajaraman, Joann J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. 251-262.	Data integration
Alon Y. Levy. Obtaining Complete Answers from Incomplete Databases. 402-412.	Data integration
Wilburt Labio, Hector Garcia-Molina. Efficient Snapshot Differential Algorithms for Data Warehousing. 63-74.	Detecting changes in the sources
Curtis E. Dyreson. Information Retrieval from an Incomplete Data Cube. 532-543.	Incomplete information

Laks V. S. Lakshmanan, Fereidoon Sadri, Iyer N. Subramanian. SchemaSQL - A Language for Interoperability in Relational Multi-Database Systems. 239-250.	Integration in general
Yannis Papakonstantinou, Serge Abiteboul, Hector Garcia-Molina. Object Fusion in Mediator Systems. 413-424.	Integration in general
Mark W. W. Vermeer, Peter M. G. Apers. The Role of Integrity Constraints in Database Interoperation. 425-435.	Integration in general
Damianos Chatziantoniou, Kenneth A. Ross. Querying Multiple Features of Groups in Relational Databases. 295-306.	Processing for aggregates
Sameet Agarwal, Rakesh Agrawal, Prasad Deshpande, Ashish Gupta, Jeffrey F. Naughton, Raghu Ramakrishnan, Sunita Sarawagi. On the Computation of Multidimensional Aggregates. 506-521.	Processing for aggregates
Divesh Srivastava, Shaul Dar, H. V. Jagadish, Alon Y. Levy. Answering Queries with Aggregation Using Views. 318-329.	Query rewritting
Amit Shukla, Prasad Deshpande, Jeffrey F. Naughton, Karthikeyan Ramasamy. Storage Estimation for Multidimensional Aggregates in the Presence of Hierarchies. 522-531.	Size estimation for views
Martin Staudt, Matthias Jarke. Incremental Maintenance of Externally Materialized Views. 75-86.	View maintenance
1997 - PODS	
Ching-Tien Ho, Jehoshua Bruck, Rakesh Agrawal. Partial-Sum Queries in Data Cubes Using Covering Codes. 228-237.	Processing for cubes
Catriel Beeri, Alon Y. Levy, Marie-Christine Rousset. Rewriting Queries Using Views in Description Logics. 99-108.	Query rewritting
Oliver M. Duschka, Michael R. Genesereth. Answering Recursive Queries Using Views. 109-116.	Query rewritting
1997 - SIGMOD	
Joseph M. Hellerstein, Peter J. Haas, Helen Wang. Online Aggregation. 171-182.	Incomplete information
Patrick E. O'Neil, Dallan Quass. Improved Query Performance with Variant Indexes. 38-49.	Indexing
Ching-Tien Ho, Rakesh Agrawal, Nimrod Megiddo, Ramakrishnan Srikant. Range Queries in OLAP Data Cubes. 73-88.	Processing for cubes
Yihong Zhao, Prasad Deshpande, Jeffrey F. Naughton. An Array-Based Algorithm for Simultaneous Multidimensional Aggregates. 159-170.	Processing for cubes
Nick Roussopoulos, Yannis Kotidis, Mema Roussopoulos. Cubetree: Organization of and Bulk Updates on the Data Cube. 89-99.	Storage for cubes
Michael J. Carey, Donald Kossmann. On Saying "Enough Already!" in SQL. 219-230.	Top N queries
Inderpal Singh Mumick, Dallan Quass, Barinderpal Singh Mumick. Maintenance of Data Cubes and Summary Tables in a Warehouse. 100-111.	View maintenance
Brad Adelberg, Hector Garcia-Molina, Jennifer Widom. The STRIP Rule System For Efficiently Maintaining Derived Data. 147-158.	View maintenance
Dallan Quass, Jennifer Widom. On-Line Warehouse View Maintenance. 393-404.	View maintenance
Latha S. Colby, Akira Kawaguchi, Daniel F. Lieuwen, Inderpal Singh Mumick, Kenneth A. Ross. Supporting Multiple View Maintenance Policies. 405-416.	View maintenance
Divyakant Agrawal, Amr El Abbadi, Ambuj K. Singh, Tolga Yurek. Efficient View Maintenance at Data Warehouses. 417-427.	View maintenance
1997 - VLDB	
Dimitri Theodoratos, Timos K. Sellis. Data Warehouse Configuration. 126-135.	DW design
Jian Yang, Kamalakar Karlapalem, Qing Li. Algorithms for Materialized View Design in Data Warehousing Environment. 136-145.	DW design
Elena Baralis, Stefano Paraboschi, Ernest Teniente. Materialized Views Selection in a Multidimensional Database. 156-165.	DW design
Christos Faloutsos, H. V. Jagadish, Nikolaos Sidiropoulos. Recovering Information from Summary Data. 36-45.	Incomplete information

Vasilis Vassalos, Yannis Papakonstantinou. Describing and Using Query Capabilities of Heterogeneous Sources. 256-265.	Integration in general
Mary Tork Roth, Peter M. Schwarz. Don’t Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. 266-275.	Integration in general
Marc Gyssens, Laks V. S. Lakshmanan. A Foundation for Multi-dimensional Databases. 106-115.	OLAP modeling
Kenneth A. Ross, Divesh Srivastava. Fast Computation of Sparse Datacubes. 116-125.	Processing for aggregates
Damianos Chatziantoniou, Kenneth A. Ross. Groupwise Processing of Relational Queries. 476-485.	Processing for aggregates
Laura M. Haas, Donald Kossmann, Edward L. Wimmers, Jun Yang. Optimizing Queries Across Diverse Data Sources. 276-285.	Query processing over integrated data
H. V. Jagadish, P. P. S. Narayan, S. Seshadri, S. Sudarshan, Rama Kanneganti. Incremental Organization for Data Recording and Warehousing. 16-25.	Storage in general
Nam Huyn. Multiple-View Self-Maintenance in Data Warehousing Environments. 26-35.	View maintenance
1998 - PODS	
John R. Smith, Chung-Sheng Li, Vittorio Castelli, Anant Jhingran. Dynamic Assembly of Views in Data Cubes. 274-283.	DW design
Phokion G. Kolaitis, David L. Martin, Madhukar N. Thakur. On the Complexity of the Containment Problem for Conjunctive Queries with Built-in Predicates. 197-204.	Query containment
Phokion G. Kolaitis, Moshe Y. Vardi. Conjunctive-Query Containment and Constraint Satisfaction. 205-213.	Query containment
Werner Nutt, Yehoshua Sagiv, Sara Shurin. Deciding Equivalences Among Aggregate Queries. 214-223.	Query containment
Serge Abiteboul, Oliver M. Duschka. Complexity of Answering Queries Using Materialized Views. 254-263.	Query rewritting
1998 - SIGMOD	
Chee Yong Chan, Yannis E. Ioannidis. Bitmap Index Design and Evaluation. 355-366.	Indexing
Prasad Deshpande, Karthikeyan Ramasamy, Amit Shukla, Jeffrey F. Naughton. Caching Multidimensional Queries Using Chunks. 259-270.	Processing for aggregates
Yihong Zhao, Prasad Deshpande, Jeffrey F. Naughton, Amit Shukla. Simultaneous Optimization and Evaluation of Multiple Dimensional Queries. 271-282.	Processing for aggregates
Jun Rao, Kenneth A. Ross. Reusing Invariants: A New Strategy for Correlated Queries. 37-48.	Query processing in general
Subbu N. Subramanian, Shivakumar Venkataraman. Cost-Based Optimization of Decision Support Queries Using Transient Views. 319-330.	Query processing over integrated data
Renée J. Miller. Using Schematically Heterogeneous Structures. 189-200.	Schema integration
Yannis Kotidis, Nick Roussopoulos. An Alternative Storage Organization for ROLAP Aggregate Views Based on Cubetrees. 249-258.	Storage for cubes
1998 - VLDB	
Amit Shukla, Prasad Deshpande, Jeffrey F. Naughton. Materialized View Selection for Multidimensional Datasets. 488-499.	DW design
Min Fang, Narayanan Shivakumar, Hector Garcia-Molina, Rajeev Motwani, Jeffrey D. Ullman. Computing Iceberg Queries Efficiently. 299-310.	Iceberg queries
Frédéric Gingras, Laks V. S. Lakshmanan. nD-SQL: A Multi-Dimensional Language for Interoperability and OLAP. 134-145.	Integration in general
Fernando de Ferreira Rezende, Klaudia Hergula. The Heterogeneity Problem and Middleware Technology: Experiences with and Performance of Database Gateways. 146-157.	Integration in general
Guido Moerkotte. Small Materialized Aggregates: A Light Weight Index Structure for Data Warehousing. 476-487.	Processing for aggregates

Michael J. Carey, Donald Kossmann. Reducing the Braking Distance of an SQL Query Engine. 158-169.	Top N queries
Hector Garcia-Molina, Wilburt Labio, Jun Yang. Expiring Data in a Warehouse. 500-511.	View maintenance
1999 - PODS	
Howard J. Karloff, Milena Mihail. On the Complexity of the View-Selection Problem. 167-173.	DW design
Sara Cohen, Werner Nutt, A. Serebrenik. Rewriting Aggregate Queries Using Views. 155-166.	Query rewritting
Stéphane Grumbach, Maurizio Rafanelli, Leonardo Tininini. Querying Aggregate Data. 174-184.	Query rewritting
1999 - SIGMOD	
H. V. Jagadish, Laks V. S. Lakshmanan, Divesh Srivastava. Snakes and Sandwiches: Optimal Clustering Strategies for a Data Warehouse. 37-48.	Clustering
Yannis Kotidis, Nick Roussopoulos. DynaMat: A Dynamic View Management System for Data Warehouses. 371-382.	DW design
Kevin S. Beyer, Raghu Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs. 359-370.	Iceberg queries
Ramana Yerneni, Chen Li, Hector Garcia-Molina, Jeffrey D. Ullman. Computing Capabilities of Mediators. 443-454.	Integration in general
Peter J. Haas, Joseph M. Hellerstein. Ripple Joins for Online Aggregation. 287-298.	Processing for aggregates
Arunprasad P. Marathe, Kenneth Salem. Query Processing Techniques for Arrays. 323-334.	Query processing for arrays
Zachary G. Ives, Daniela Florescu, Marc Friedman, Alon Y. Levy, Daniel S. Weld. An Adaptive Query Execution System for Data Integration. 299-310.	Query processing over integrated data
Chen-Chuan K. Chang, Hector Garcia-Molina. Mind Your Vocabulary: Query Mapping Across Heterogeneous Information Sources. 335-346.	Query processing over integrated data
Wilburt Labio, Ramana Yerneni, Hector Garcia-Molina. Shrinking the Warehouse Update Window. 383-394.	View maintenance
1999 - VLDB	
Vanja Josifovski, Tore Risch. Integrating Heterogenous Overlapping Databases through Object-Oriented Transformations. 435-446.	Integration in general
Felix Naumann, Ulf Leser, Johann Christoph Freytag. Quality-driven Integration of Heterogenous Information Systems. 447-458.	Integration in general
Alin Deutsch, Lucian Popa, Val Tannen. Physical Data Independence, Constraints, and Optimization with Universal Plans. 459-470.	Integration in general
Laks V. S. Lakshmanan, Fereidoon Sadri, Subbu N. Subramanian. On Efficiently Implementing SchemaSQL on an SQL Database System. 471-482.	Integration in general
H. V. Jagadish, Laks V. S. Lakshmanan, Divesh Srivastava. What can Hierarchies do for Data Warehouses? 530-541.	OLAP modeling
Torben Bach Pedersen, Christian S. Jensen, Curtis E. Dyreson. Extending Practical Pre-Aggregation in On-Line Analytical Processing. 663-674.	OLAP modeling
Kian-Lee Tan, Cheng Hian Goh, Beng Chin Ooi. Online Feedback for Nested Aggregate Queries with Multi-Threading. 18-29.	Processing for aggregates
Alfons Kemper, Donald Kossmann, Christian Wiesner. Generalised Hash Teams for Join and Group-by. 30-41.	Processing for aggregates
Chee Yong Chan, Yannis E. Ioannidis. Hierarchical Prefix Cubes for Range-Sum Queries. 675-686.	Processing for aggregates
Sunita Sarawagi. Explaining Differences in Multidimensional Aggregates. 42-53.	Processing for cubes
Jianzhong Li, Doron Rotem, Jaideep Srivastava. Aggregation Algorithms for Very Large Compressed Data Warehouses. 651-662.	Processing for cubes
Surajit Chaudhuri, Luis Gravano. Evaluating Top-k Selection Queries. 399-410.	Top N queries

Donko Donjerkovic, Raghu Ramakrishnan. Probabilistic Optimization of Top N Queries. 411-422.	Top N queries