From Databases to Information Systems – Information Quality Makes the Difference

Felix Naumann
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
felix@almaden.ibm.com

Abstract: Research and business is currently moving from centralized databases towards information systems integrating distributed and autonomous data sources. Simultaneously, it is a well acknowledged fact that consideration of information quality—IQ-reasoning—is an important issue for large-scale integrated information systems. We show that IQ-reasoning can be the driving force of the current shift from databases to integrated information systems.

In this paper, we explore the implications and consequences of this shift. All areas of answering user queries are affected – from user input, to query planning and query optimization, and finally to building the query result. The application of IQ-reasoning brings both challenges, such as new cost models for optimization, and opportunities, such as improved query planning. We highlight several emerging aspects and suggest solutions toward a pervasion of information quality in information systems.

1. Information Quality

The development of the Internet—especially the World Wide Web—has made it possible to access a multitude of data sources on almost any given topic. Web directories guide users to these sources, search engines let users discover sources previously unknown to them, and a huge number of Web sites act as data sources and provide the actual data. Most often, a user may choose between many alternative sources and source combinations to obtain the desired information item. This choice is advantageous but also time-consuming. It is advantageous to choose the most renowned, the fastest, or the most accurate sources. But it is time-consuming to come to this choice through trial and error. And it is even more time-consuming to access several sources in a row if the desired information is not provided by a single source, but is spread across those sources.

Consider search engines as data sources. Most users have chosen their favorite search engine, possibly based on personal experience in response time, relevancy of the results, ranking method, usability, etc. However, users might miss just the right Web page, simply because that page was not yet indexed or ranked sufficiently high by the search engine of choice. Meanwhile, other search engines might have already indexed this Web page. The user might turn to one of these

and may eventually find the wanted Web page. A meta-search engine solves this problem by simultaneously querying multiple search engines with the user's keywords. The results of the different engines are integrated to a combined response to the user. The drawback is that a quality-unaware meta-search engine uses engines of mixed quality, creating an inferior result.

Integrated access to data that is spread over multiple, distributed, autonomous, and heterogeneous data sources is an important problem for information consumers in many areas. In this paper we argue that the user goal of finding data is changing with the move from databases to integrated information systems: Users demand not the *correct* answer but are satisfied with approximate answers. Not the *complete* answer is necessary, but the answer should be relevant. Users demand not a *full* answer with all attributes, but are content with missing values. We also show that the optimization goal of finding a complete answer as quickly as possible has shifted to its dual problem of finding the best possible answer within a given cost/time constraint.

The emergence of Web-based information systems has amplified the known problems of poor information quality, but at the same time has reached an audience with a new requirement profile.

- **Technologies**: Due to the abundance of information sources on the Web, and due to many new technologies and architectures to fuse multiple sources to appear as one, source selection, information integration, and information filtering are important tasks to shield information consumers from data overflow, data errors, or simply low quality data.
- **Users**: The Web has made information available to a much broader audience. The vast majority are casual users who do not have a high stake in the outcome of the query, so that the answer must not be of highest quality. Additionally, users of Web-based systems are more aware of IQ problems and in consequence reduce their expectations. Integrated information systems can take advantage of lower expectations by reducing the amount of resources spent to answer a query.

Not having to or not being able to respond to user queries with maximal quality demands a comprehensive model of information quality.

1.1 Databases vs. Information Systems

Reasoning about information quality (IQ-reasoning) comes in two flavors: IQ-reasoning for database management systems (databases) and IQ-reasoning for information systems. Both the way of measuring and improving the quality of query results and the users expectations toward quality differ widely.

Information quality reasoning for databases differs from information quality reasoning for information systems. For illustration, we exaggerate the characterization of the two: Databases provide storage for structured, well-defined data and full query access to the data. In particular, data in databases is either gathered by the users of the database themselves[1], or the users are able to update or delete the data. In essence, the control of the data lies with the users.

[1] Or people working for the same company gather it.

Information systems, on the other hand, are collections of structured, semi-structured, or unstructured information items such as text, tables, images etc. Integrated information systems gather this information (logically or materialized) from multiple, possibly autonomous information sources. Users of such information system have no control over the information it provides. We will argue in Section 2 that answering queries in these two types of systems differs, and that this difference is best described by an information quality model, and is best addressed by IQ-reasoning.

1.2 Scenarios

We point out several exemplary scenarios of data usage that can be enhanced through IQ-reasoning.

Search. Searching is one of the most frequent used methods to gain information on the Web, and for inexperienced users the simplest way to pose queries. To search over multiple sources, meta searching techniques that distribute a search term to multiple sources are employed.

Meta-search engines are the most prominent example. However, search engines differ widely in the number of Web pages they have indexed, the amount of information they return for each page, their response time etc. An IQ-aware meta-search engine could improve results by taking such quality scores into account when deciding which individual sources to send the query to.

Other search-scenario examples are Web-based telephone and email directories, which can be integrated to increase the chance of finding a person (white pages) or company (yellow pages). Reasoning about their quality can identify large sources, sources with much additional data about a person like fax-number and email-address, source with up-to-date data, etc. Integration of the sources using this metadata can greatly enhance the final result and help filter out duplicates.

Information integration. Information integration is the process of taking multiple query results and merging them into a single response to the user. IQ-reasoning can enhance the integration of incoming query results in two ways: (i) Conflict resolution benefits from IQ-reasoning and (ii) result tuples can be ranked by their quality.

A data conflict occurs when two sources report different data values about the same real world entity. Resolution functions are employed to resolve these conflicts by deciding which value to include in the final result. Having knowledge about the quality of the sources, resolution functions can favor the value from the qualitatively better source. For instance, when deciding which address to include in a result for a person search, the address of the source with the higher update frequency can be chosen. Further examples are search engines that export a date attribute specifying the last update of the page index. In this case, the more recent data about the Web page should be chosen. Quality-dependent resolution functions enhance the query result by favoring high quality information over low quality information.

The presentation of the final integrated results also profits from IQ-reasoning. The quality determined for a source, a part of a plan, or an entire plan represents the quality of the data generated by the plan. Instead of dropping this information once the data is received, it can be used to rank

the query results. If the user does not specify another order, high quality tuples should be ranked first.

Data mining. Data Mining is the process of extracting previously unknown information from a set of data, such as a data warehouse. Data mining techniques are especially sensitive towards poor data quality [LLLK99]. For instance, outliers, i.e., data points that lie far from the average, severely skew the results of data mining algorithms. Outliers are usually produced where the data itself is generated: Sensors give incorrect output, a human accidentally adds a decimal to a number, etc. Therefore, any data mining method is preceded by a data cleaning technique to improve data quality, before applying the actual mining algorithms [Pyl99].

Also, other aspects of information quality play an important role for data mining. The completeness of the data is of importance so as not to mine on a subset of the available data. If the data, such as consumer behavior data, is obtained from a third party, the reputation and objectivity of the source are an important factor. IQ-aware data mining can improve the quality of the results.

1.3 Conclusion

We define IQ-reasoning as the integration of IQ aspects to the process of planning and optimizing user queries against databases and information systems. IQ aspects include a set of IQ criteria, IQ assessment methods, and an IQ measure. When information sources store data and information about the same real world objects, information quality aspects constitute the main difference between the sources. These observations and others, such as those in [CZW98, Wei99, Wie99, MRV00], give rise to the following axiom:

> *Information quality is the main discriminator of Web data sources, and information quality reasoning should be used to improve integrated query results.*

Or condensed:

> *Information quality is the response time of the Web age.*

The rest of the paper is organized as follows: Section 2 analyses the traditional problem of query answering and optimization, and then describes the changes introduced by query processing over integrated sources. In Section 3 we present several necessary IQ components that enable IQ-reasoning for databases and information systems. Section 4 concludes the paper with an appeal to IQ-aware design and deployment of future information systems.

2. A Problem Shift

Information quality (IQ) is the main discriminator of data and data sources on the Web. As we have seen in the previous section, the autonomy of Web data sources renders it necessary and useful to consider their quality when integrating their data. The information system paradigm shift—from central database management systems (DBMSs) to distributed multidatabase systems and finally to virtual, integrated World Wide Web information systems—has moved attention from *query processing* to what we call *query planning*.

Query processing is concerned with efficiently answering a user query to a single or multidatabase. In this context efficiency means speed. If not the speed of answering one query efficiently, it is the speed of the overall running system that is optimized. Many researchers and developers have designed sophisticated algorithms, index structures, etc., to enhance database efficiency. All those techniques have the same goal: Find the query execution plan that provides users with the correct and complete query result in an efficient manner.

Query planning on the other hand is concerned with finding the best possible answer given some cost or time constraint. Query planning involves regarding many query execution plans across different, autonomous sources that together form the complete result. Research has addressed the problem of determining *all* such plans [LRO96, Les98], but to the best of our knowledge only [NLF99] has addressed the problem of finding the *k* best plans, where "best" is defined through a quality model.

2.1 Query processing in DBMS

Databases store data and let users pose queries against it. The aim of query processing is to answer those queries with the available data. When answering user queries the DBMS assumes that users require correctness of the answer (R.1) and completeness of the answer (R2 and R.3):

- **R.1:** The user expects only *correct* results, i.e., only tuples where all query conditions hold true. For example, a user of a data warehouse asking for departments with revenue of at least $1,000,000 expects in the result *only* such departments.

- **R.2:** The user expects the result to be *extensionally complete*, i.e., to contain *all* correct tuples accessible by the integrated system. Continuing the example above, the user not only expects only departments with the specified revenue, but also *all* those departments (as long as their revenue data is stored in the database).

- **R.3:** The user expects the result to be *intensionally complete*, i.e., to contain all attributes specified in the query and contain non-`null` values in all the attributes. Continuing the example, if the user asked for the department name, its revenue, and its manager, the user expects all this information to be in the result. The user will not accept missing manager data (again, as long as this data is actually stored in the database).

Completeness and correctness in a DBMS are defined with regard to the content of the underlying database. The assumptions toward this database are that it contains only correct data, and that it contains all relevant data (closed world assumption). For instance, corporate users of a customer database assume that all customer data is correct and that data about all customers is actually stored within the database. He/she will not doubt the data provided, and will not turn to other databases suspecting that there is more customer data stored elsewhere.

Of course, DBMSs may also contain incorrect data; of course DBMSs may also not have all available data. However, compared to Web data sources, the owner of a DBMS has the power to change this situation. If there are inaccurate data, one can correct them, if data is missing, one can insert it. If the overall quality of the system is low, one can take measures to increase the quality aspects that are amiss. Web data sources on the other hand are autonomous. If complete-

ness and correctness or the overall information quality is not satisfying, there is usually nothing the integrating system can do about it.

The query processing component of a DBMS tries to answer a given query as cost-efficiently as possible, where cost-efficiency is usually defined as *response time*. Response time is the time a user must wait after submitting a query until reception of the complete result. A DBMS predicts response time using a cost model, which calculates the cost of database operations, such as join or selection operations, on different relations. In particular, the optimization component of a relational DBMS solves the following (simplified) problem:

> *Given a set of relations, a user query against them, and a cost model, find the most cost-efficient order to access and combine the relations.*

The problem definition becomes more complicated for multiple parallel processors, multiple queries and multiple DBMSs. The basics however remain the same: The desired result (and hence, also its quality) is fixed—the aim of query processing is to generate this fixed result as efficiently as possible.

2.2 Query Planning in Integrated Information Systems

Query planning in information systems reverses this paradigm, as we will see: In general, the completeness and correctness assumptions about the underlying database do not hold for Web data sources in an open world—quite the contrary: A search engine will never have indexed *every* available Web page on the World Wide Web; stock information systems do not provide data on every stock; Web-based telephone directories only store data about some people, but never cover all telephone networks. That is, Web data sources are usually not complete. Correctness is also never guaranteed: Web pages may change after a search engine has indexed them; stock information systems purposely return delayed and thus outdated stock quotes; etc.

Further, typical users and Web servers have resource constraints: There might be technical constraints, such as a limited network bandwidth or limited access to the underlying data sources. Users may have constraints, such as a limited budget or limited time. Finally, users might have non-technical constraints, such as an unwillingness to browse a large result set. For example, a meta-search engine does not need to download all hits from all search engines it uses; instead, integrating the top ten hits usually suffices.

Knowing about incompleteness and limited correctness of Web sources, and having limited resources in terms of time and money, users of Web-based information systems make three concessions (C.1 – C.3) corresponding to the three requirements (R.1 – R.3) of the previous section:

- **C.1:** Users accept tuples where attribute values are incorrect but *close* to their selection condition. For example, a user querying for cars with a price lower then $10,000 might also find cars for $10,500 agreeable in the result. Allowing plurals or synonyms of search terms can extend the results of a search engine.

- **C.2:** Users accept *extensionally incomplete* answers in the presence of constrained resources. If, for any reason, the extensionally complete answer cannot be returned, the best

possible answer should be returned. A user of a search engine usually does not demand the entire result set but is satisfied with, say, ten Web pages. However, the result should consist of the Web pages best matching the keywords of the query.

- **C.3:** Users accept *intensionally incomplete* answers or answers with *missing values*—a partial answer is better than no answer. A user of a stock information service asking for companies whose stock quotes have risen more than 10 percent today along with a company profile is at least partially satisfied if the result contains companies without the profile information. Of course, those tuples for which the profile *is* available should be listed first, but others might still be a helpful part of the result.
  Integrated information systems should not reduce their information offer to the lowest common denominator of the participating sources, in effect throwing away information. For instance, a meta-search engine like MetaCrawler offers only title, description, and URL of a Web page, even though it queries several sources that offer much more information, such as language, size, etc.

In short, users cannot and do not expect the same type of results from a query to a Web-based and integrated information system as they do from a DBMS[2]. Hence, the problem of query processing is reversed:

> *Given a set of relations/sources, a user query against them, a quality model, and a cost limit, find the highest quality combination of the relations/sources within the cost limit.*

Like a cost model, a quality model should be able to predict the quality of the result, retrieved from different sources and combinations of sources (see Sec. 3.2). The problem is reversed, because now the cost/efficiency is fixed, while the quality of the result is optimized. Cost can be fixed for several reasons:

- Users might not wait indefinitely for a result, but abort a query after a few minutes. For instance, a meta-search engine will not waste time by waiting for all search engines to return a result. Rather, it will integrate all results that have been returned within the first few seconds. In effect this fixes the time the information system has available to find some (best) answer to the query.

- If systems charge money to access the information, users might specify a spending limit. The higher the limit, the better the result is expected to be.

- To deal with large number of users, the information system itself might wish to spend only a certain amount of bandwidth or time for each query. This may limit the number of sources to access and the amount of data to be retrieved from each source for any given query.

[2] In fact, due to varying availability and frequent changes of sources, user cannot even expect two identical queries to produce the same result.

2.3 Conclusion

Improved technology has given rise to a new type of information system, which covers much more information at the cost of diminished quality. Simultaneously this technology is available to many more people, who have lower expectations toward the quality. IQ, as the main discriminator of these new systems, should play an increasingly important role in the systems design and deployments. The new paradigm of planning queries across multiple information sources provides quality-driven challenges throughout the integrated system:

- Design a **quality measure** with a set of IQ criteria and a way to measure them.
- Design a **quality model** to determine the quality of combinations of sources.
- Design **optimization algorithms** finding only a few best answers and dealing with quality model properties.
- Design **information integration** techniques that enhance the quality of the result.

The following sections highlight some of the necessary changes to meet the challenges.

3. New Components for IQ Pervasion

General definitions for information quality are "*fitness for use*" [TB98], "*meets information consumers needs*" [Red96], or "*user satisfaction*" [DM92]. These definitions are just as non-operational as Pirsig's: "*Even though quality cannot be defined, you know what it is*" [Pir74]. Rather, we conceive quality as an aggregate value of multiple IQ-criteria. With this definition, information quality is flexible regarding the application domain, the sources, and the users, because the selection of criteria can be adapted accordingly. Also, assessing scores for certain aspects of information quality and aggregating these scores is easier than immediately finding a single global IQ-score.

3.1 An IQ Measure

Information quality is defined as a catalog of IQ-criteria. Several research projects have put together such general catalogs [Bas90, CZW98, JV97, Red96, Wei99, WS96] or compiled multiple catalogs [NR00, EW00]. These catalogs are proposals formulated in the most general way to allow for different interpretation depending on applications, data sources, and users. Many criteria are not independent and typically not all criteria should be used at the same time. Rather, an application specific selection of criteria helps to identify qualitatively good data and simultaneously reduces assessment cost. Information quality assessment is the process of assigning numerical values (IQ-scores) to IQ-criteria. An IQ-score reflects one aspect of information quality of a set of data items. Usually this set represents an entire data source, but it might be useful to assign scores to certain parts of data sources as well. We are aware of the difficulties of numerically expressing certain criteria. Because not the absolute IQ-scores are of importance, but rather their relative values, we believe that a numerical approach is reasonable. One of the major challenges is to make IQ-assessment feasible.

IQ-assessment is rightly considered difficult, and there have been only few research approaches addressing it. In [EW00] Eppler and Wittig observe that most existing assessment methods *solely* rely on users to provide IQ-scores [BMY99, WSKL99], even though many criteria can be assessed automatically (e.g., AVAILABILITY), or semi-automatically (e.g., COMPLETENESS) [NR00].

When assessing IQ-scores, it is necessary to observe the tradeoff between precision and practicality.

Below, we highlight two criteria that play an especially important role for both databases and integrated information systems (RESPONSE TIME and ACCURACY), and two exemplary criteria that emphasize the need for a broader definition of IQ for integrated information systems (COMPLETENESS and RELEVANCE).

RESPONSE TIME. Traditionally, the quality of a database is determined by its ability to respond quickly to queries, i.e., its RESPONSE TIME. The cost models of database optimizers, which have only speed as their goal, reflect this quality measure[3]. While this goal remains important for integrated information systems, methods of achieving low RESPONSE TIME have dramatically changed: In traditional databases much query processing time is spent in CPU-bound tasks such as the optimization algorithm itself, sorting a large set of values, or processing a join operator. Because of the distribution of sources in integrated information systems, this time is by far outweighed by network-bound tasks, such as retrieving a result set over a network, or waiting for a server response. In consequence, cost models of optimizers should adapt to this new situation. The key ability of Web-based information systems is not to answer queries quickly, but to answer them well.

ACCURACY. Recently, ACCURACY[4] has found more attention among database users and has been subject of several research projects, such as [HS98, MWS98, GFSS00]. Data quality is a quality measure for the relative amount of erroneous data stored in the database. Integrating multiple information sources is both a source for low ACCURACY and an opportunity to increase ACCURACY.

Autonomous information sources are a source for inaccurate data, or more precisely, a source for data with unknown and unalterable ACCURACY. In a centralized database the consumer of data typically owns the data. Insufficient ACCURACY is created by the consumer and can be remedied by the consumer. This is not the case for autonomous sources, such as sources on the Web.

On the other hand, the ability to access multiple sources to obtain information about the same real world object gives systems the opportunity to combine the data to a more accurate overall representation of the object (see Section 3.4).

COMPLETENESS. For many data sources and many application domains, size is everything: The more tuples and the more attributes a source provides, the more attractive it is to users. For instance, users typically prefer large search engines, i.e., search engines that have indexed a large number of Web pages, over small search engines. The rationale is that the larger a search engine is, the higher the probability is, that the result the user is looking for has been indexed by the search engine (and therefore appears in the result). Also, users prefer search engines that return more attributes than others, e.g., knowing the *byte size* of a Web page before clicking on the link is advantageous.

[3] In multi-user environments, some DBMS optimize for throughput, sacrificing response time of individual users for overall fast responses to all users. Essentially, the optimization goal remains time-based.
[4] ACCURACY is also known as “data quality”, as opposed to the more general term “information quality”.

Determining the "size" of a data source has only recently become a problem, when such meta-data became desired for autonomous sources of unknown size, such as typical WWW information sources. There are yet few projects striving to model or determine the size of Web data sources [BB98]. Chen and associates, who address query processing in the WWW, mention the quality criteria "size of result" and "number of documents accessed", but they neither define them, nor point out the difference between the two [CZW98]. Motro and Rakov define a completeness criterion, counting the tuples in a source [MR98].

Calculation or prediction of join result sizes is an important technique for cost-based query optimization in DBMS [Ros81, GP89, SS94]. Mannino and associates give a survey on the suggested statistical values to store, how to maintain them, and how to use them to predict the result sizes of various database operations [MCS88]. Florescu and associates attempt to describe quantitatively the content of distributed autonomous document sources using probabilistic measures [FKL97].

All approaches have in common that they aim to predict the number of tuples/objects in the result, but none consider the amount of information returned per tuple. One source might provide rich information about the objects, another only a few attributes. In [NL00a] we propose to combine these measures with a density measure, which takes this aspect into account and also counts the frequency of null-values in the tuples—a common phenomenon in Web-based information sources.

RELEVANCE. RELEVANCE is the degree to which the provided information satisfies the users need. RELEVANCE is a standard criterion in the field of information retrieval [SM83]. There, a document or piece of data is considered to be relevant to the query, if the keywords of the query appear often and/or in prominent positions in the document. That is, word-counting techniques guide the relevance measure [GGMT99].

The importance of RELEVANCE as a criterion depends on the application domain. For instance, for search engines RELEVANCE is quite important, i.e., returned Web page links should be as relevant as possible, even though this precision is difficult to achieve. For instance, a query for the term "jaguar" to a Web search engine retrieves document links both for the animal and the automobile. If the user had the animal in mind, the links to automobile sites should have been considered as not relevant. The use of ontologies can help solve such problems to some extent. In other application domains, RELEVANCE is implicitly high. For instance, a query for IBM stock quotes in an integrated stock information system only returns relevant results, namely IBM stock quotes. The reason for this discrepancy is the definition of the domain: Search engines have the entire WWW as a domain and thus provide much data that is of no interest to the user. The domain of a stock information system is much more clear-cut and much smaller, so a query is less likely to produce irrelevant results.

For our purposes we reduce the RELEVANCE criterion to a correctness criterion. If a result is correct with respect to the user query, we assume that it is also relevant. If it is not relevant, the user query was either incorrect with respect to what the user had in mind, or it was not specific enough.

3.2 An IQ Model

An information quality model for integrated information systems takes on the role of cost models in DBMS. Given the quality of the participating sources (using the quality measure of the previous section), a quality model determines the quality of the query result. In a DBMS, the optimizer component explores different alternatives of executing a query (query execution plans), applies the cost model to each alternative and chooses the cheapest one. In an information system, the planner also considers different alternatives of executing a query (different combinations of sources) and applies the quality model to determine the best of those plans.

Given IQ-scores for all sources in all criteria, two problems must be solved: (i) IQ aggregation to determine the IQ-score for a plan in each criterion. (ii) IQ ranking to rank sources according to those multiple, aggregated IQ-scores.

IQ Aggregation. We propose merge functions as a method to determine IQ criterion scores of multiple sources. A merge function has a different interpretation for each criterion, reflecting properties of the underlying IQ-measure. For instance, the merge function for a PRICE criterion is the SUM function, because the price of each participating source in a plan must be paid. RESPONSE TIME has MAX as merge function, assuming parallel access to all sources in a plan. Merge functions can be quite complex, such as for the COMPLETENESS criterion [NF00].

Merge functions must be commutative and associative, so that a change of the execution order has no effect on its IQ-score. This property is desirable, as the user perceives the quality of the query result and not the quality of how the query result is obtained. The result of IQ aggregation for each combination of sources (plan) is a vector of IQ-scores with one dimension per criterion.

IQ Ranking. Given the IQ-vectors for a number of plans, we want to find a—possibly complete—qualitative ordering of them, to decide which one to execute. Methods to solve this problem are called ranking methods or Multi-Attribute Decision-Making methods (MADM). These face three general problems:

1. The range and units of the IQ-scores of the criteria varies. *Scaling methods* solve these problems.

2. The importance of the criteria varies in the eyes of a user. *User weightings* specified as a weight vector solve this problem.

3. The IQ-scores place the data sources into a multi-dimensional space with one dimension per IQ-criterion. Because there is no natural order on a multi-dimensional space, the *ranking methods* determine an ordering among the sources or combinations of sources (for an overview see [Nau98]).

After scaling and weighting the IQ-vectors, ranking methods map them to single scalar IQ-scores, which determine the rank among them. In a simple scheme, the best plans are subsequently queried, until the cost limit is reached. The following section describes more sophisticated approaches.

3.3 IQ Optimization

Query answering on the Web can be enhanced both in effectiveness and efficiency by using IQ-reasoning. As argued before, in the presence of resource constraints it is often not possible to execute all plans for a query. When not all plans can or should be executed, it is beneficial to restrict execution not to arbitrary plans, but to the best plans according to a quality model.

Recently, there has been some research on retrieving only the top *N* answers to a query [CK97, CG99, TGO99], where "top" is not in reference to information quality, but to some similarity measure. For instance, Chaudhuri and Gravano justify the relaxed requirement with a query for houses at a certain price and with a certain number of rooms against a real estate database. Obviously, the user does not expect only houses that *exactly* match the query, rather, the *N* results *best* matching the query should be returned. In an earlier article the authors based the top *N* approach on multimedia repositories, where objects typically match conditions only to a certain degree [CG96]. Therefore, it does not suffice to only return exact matches, nor is it feasible to return all objects that match to even the slightest degree. In their paper, the user must specify a minimum matching degree for result objects. This research amounts to the consideration of concession C.1 for query planning.

Pre-optimization. The potential number of plans for a user query is exponential in the number of relations in the user query and the number of sources. For instance, given 10 search engines, a meta-search engine could answer a user query by accessing any of the 10! = 3,628,800 combinations of them. Therefore, it is desirable to decrease this number before starting to generate these combinations. To this end, we use the source-specific IQ-criteria to "weed out" sources that are qualitatively not as good as others. Our goal is to find a certain number or percentage of best sources independently of any user-specific weighting or preference.

Mihaila and associates recently suggested using IQ-metadata for source selection [MRV00]. To this end, the authors suggest an extension of SQL with fuzzy conditions so that the user can specify the desired quality of the result.

Optimization. Essentially, an optimizer trying to find the best set of sources under some cost constraint must solve the Knapsack problem [GJ79]. The Knapsack problem is proven to be NP-complete, but there are many approximation algorithms that efficiently find near optimal solutions. The Knapsack problem assumes that combining sources has monotone benefit, i.e., adding a source to a combination never decreases overall quality. For general quality model we cannot assume this property. Consider the ACCURACY criterion. Adding an inaccurate source to a combination can decrease overall accuracy. In such cases, more quality-aware algorithms must be employed to guarantee certain optimality [NL00a]. An additional problem arises in a Web-based environment, where sources can fail without warning. Optimization algorithms must be able to dynamically adapt to such situations, for instance, by re-optimizing after each source failure, or by anticipating failures in the plan. Of course, consideration of an AVAILABILITY criterion for each source could reduce source failures in a plan: Unreliable sources will be valued at a lower quality and will less likely enter a plan in the first place.

Post-optimization. The order in which the results arrive from the participating sources is not necessarily the best order to present them to the user. The IQ-scores already obtained can be used to rank the result tuples, presenting the highest quality information first.

3.4 Information Integration

Data about a real world entity may be stored with differing attribute values at different sources. In strict, duplicate removing relational semantics, those tuples would appear individually in the result of any operator. Even in the presence of a unique ID-attribute identifying the entity, a relational operator returns multiple tuples about the same entity. Integration of results is reduced to concatenation of results. It is left to the user to identify and resolve data conflicts. We propose to only represent one result tuple per real world entity. To this end, traditional operators must be enhanced to include resolution functions as presented earlier.

Generally speaking, data sources overlap in two ways: extensionally and intensionally. The extensional overlap between two sources is the set of real world entities that are represented in both sources. The intensional overlap between two sources is the set of attributes both sources provide.

To make use of overlap and to integrate data in a meaningful and useful way, we must recognize identical entities represented in different sources (object identification), and we must be able to resolve any data conflicts between values (conflict resolution). Especially during conflict resolution, IQ-reasoning can greatly improve the result.

Object Identification. Integrating data from different sources requires that different representations of identical real world entities be identified as such [Ken91]. This process is called object identification. Object identification is difficult, because the available knowledge about the objects under consideration may be incomplete, inconsistent, and sparse. A particular problem occurs if no natural IDs exist. For instance, the URL of a Web page is a natural ID for the page. A meta-search engine can use the URL of reported hits to find and integrate duplicates. On the other hand, a used car typically has no natural ID. An integrated information system for used cars has no easy way of finding identical cars being advertised in different data sources.

Object identification in the absence of IDs, which is essentially the same problem as duplicate detection, record linkage, or object fusion [NL00, New88, PAGM96], is typically approached by statistical methods, for instance, using rough set theory [Zia99]. After having identified a set of tuples representing the same real world entity, they must be combined to a single representation. If their data values differ in some attributes, conflict resolution must be applied.

Conflict Resolution. Once different tuples have been identified as representing the same entity, the data about them can be integrated. In general, a result that is integrated from tuples of different sources, contains tuples where

1. some attribute value is not provided by *any* of the sources,
2. some attribute value is provided by *exactly one* source, and
3. some attribute value is provided by *more than one* source.

In the first case, it is obvious how the result is merged: Because the sources do not provide a value, the tuple in the result has no value either (null-value). In the second case, there is also no data conflict; thus, when constructing the result, the one attribute value can be used for the result tuple. Depending on the type of attribute and the type of sources, the fact that the data is missing in some sources can be taken into account as well, when determining the final attribute value.

The third case demands special attention. Several sources compete in filling the result tuple with an attribute value. If all sources provide the same value, that value can be used in the result. If this is not the case, there is a data conflict and a *resolution function* must determine what value shall appear in the result table.

Internal resolution functions are of various types, depending on the type of attribute, the usage of the value, and many other aspects [KCGS95, YM98]. A simple resolution function might concatenate the values and annotate them with the source that provided the value. Especially conflicts in textual attributes may be resolved in this way. Resolution functions need not only depend on the two conflicting attribute values. A resolution function could additionally depend on quality scores like AGE, favoring the more recent data value. In general, resolution functions should include IQ-scores in their decision and favor sources of higher quality.

4. Conclusion

The surfacing of Web-based, integrated information systems has altered the way queries can be answered, and it has altered the expectations of users. In both cases, information quality is the main discriminator of the changes: Now, more queries can be answered with a larger underlying information space, at the cost of decreased quality of the answers. With more and more publicly available data and more and more autonomous sources, the problem will increase in the future. Now, more users can access the information sources and the information need of more users is covered. The expectations towards the quality of the answers to such queries are low.

To make full use of the opportunity to integrate large amounts of data from various sources, IQ-reasoning methods must be applied at all levels of the integration process. We hope that our findings about information quality and our IQ-reasoning techniques will find their way into integrated information systems, thereby regaining the ability to deliver high quality query results to users, once lost in the transition from centralized database management systems to systems integrating autonomous information sources.

References

[Bas90] Reva Basch. Measuring the quality of the data: Report on the fourth annual SCOUG retreat. *Database Searcher*, 6(8):18-24, October 1990.

[BB98] Krishna Bharat and Andrei Broder. A technique for measuring the relative size and overlap of public web search engines. In *Proceedings of the International World Wide Web Conference*, Brisbane, Australia, April 1998.

[BMY99] Mónica Bobrowski, Martina Marré, and Daniel Yankelevich. A homogeneous framework to measure data quality. In *Proceedings of the International Conference on Information Quality* (IQ), pages 115-124, Cambridge, MA, 1999.

[CG96] Surajit Chaudhuri and Luis Gravano. Optimizing queries over multimedia repositories. In *Proceedings of the ACM International Conference on Management of Data* (SIGMOD), pages 91-102, Montreal, Canada, 1996.

[CG99] Surajit Chaudhuri and Luis Gravano. Evaluating top-*k* selection queries. In *Proceedings of the International Conference on Very Large Databases* (VLDB), pages 397-410, Edinburgh, Scotland, 1999.

[CK97] Michael J. Carey and Donald Kossmann. On saying ”Enough already!” in SQL. In *Proceedings of the ACM International Conference on Management of Data* (SIGMOD), pages 219-230, Tucson, AZ, 1997.

[CZW98] Ying Chen, Qiang Zhu, and Nengbin Wang. Query processing with quality control in the World Wide Web. *World Wide Web*, 1(4):241-255, 1998.

[DM92] W.H. Delone and E.R. McLean. Information systems success: the quest for the dependent variable. *Information Systems Research*, 3(1):60-95, 1992.

[EW00] Martin J. Eppler and Doerte Wittig. Conceptualizing Information Quality: A review of information quality frameworks from the last ten years. In *Proceedings of the International Conference on Information Quality* (IQ), Cambridge, MA, 2000.

[FKL97] Daniela Florescu, Daphne Koller, and Alon Levy. Using probabilistic information in data integration. In *Proceedings of the International Conference on Very Large Databases* (VLDB), pages 216-225, Athens, Greece, 1997.

[GFSS00] Helena Galhardas, Daniela Florescu, Dennis Shasha, and Eric Simon. An extensible framework for data cleaning. In *Proceedings of the International Conference on Data Engineering* (ICDE), page 312, San Diego, CA, 2000.

[GGMT99] Luis Gravano, Hector Garcia-Molina, and Anthony Tomasic. GlOSS: Text-source discovery over the Internet. In ACM Transactions on Database Systems (TODS), 1999

[GJ79] Michael R. Garey and David S. Johnson. Computers and Intractability. W.H. Freeman and Company, New York, NY, 1979.

[GP89] Daniéle Gardy and Claude Puech. On the effects of join operations on relation sizes. *ACM Transactions on Database Systems* (TODS), 14(4):574-603, 1989.

[HS98] Mauricio A. Hernández and Salvatore J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9-37, 1998.

[JV97] M. Jarke and Y. Vassiliou. Data warehouse quality design: A review of the DWQ project. In *Proceedings of the International Conference on Information Quality* (IQ), Cambridge, MA, 1997.

[KCGS95] W. Kim, I. Choi, S. Gala, and M. Scheevel. On resolving schematic heterogeneity in multidatabase systems. In W. Kim, editor, *Modern Database Systems*, chapter 26, pages 521-550. ACM Press, New York, NY, 1995.

[Ken91] William Kent. The breakdown of the information model in multi-database systems. *SIGMOD Record*, 20(4):10-15, 1991.

[Les98] Ulf Leser. Combining heterogeneous data sources through query correspondence assertions. In *Workshop on Web Information and Data Management*, in conjunction with CIKM'98, pages 29-32, Washington, D.C., 1998.

[LLLK99] Mong-Li Lee, Tok Wang Ling, Hongjun Lu, and Yee Teng Ko. Cleansing data for mining and warehousing. In *Proceedings of the International Conference on Data-*

base and Expert Systems Applications (DEXA), volume 1677 of *LNCS*, pages 751-760, Florence, Italy, 1999. Springer.

[LRO96] Alon Y. Levy, Anand Rajaraman, and Joann J. Ordille. Query-answering algorithms for information agents. In *AAAI National Conf. on Artificial Intelligence*, pages 40-47, Portland, OR, 1996.

[MCS88] Michael V. Mannino, Paicheng Chu, and Thomas Sager. Statistical profile estimation in database systems. *ACM Computing Surveys*, 20(3):191-221, 1988.

[MR98] Amihai Motro and Igor Rakov. Estimating the quality of databases. In *Proceedings of the International Conference on Flexible Query Answering Systems* (FQAS), pages 298-307, Roskilde, Denmark, May 1998. Springer Verlag.

[MRV00] George A. Mihaila, Louiqa Raschid, and Maria-Esther Vidal. Using quality of data metadata for source selection and ranking. In *Proceedings of the ACM SIGMOD Workshop on The Web and Databases* (WebDB), pages 93-98, Dallas, TX, 2000.

[MWS98] Steve Mohan, Mary Jane Willshire, and Charles Schroeder. DataBryte: A proposed data warehouse cleansing framework. In *Proceedings of the International Conference on Information Quality* (IQ), pages 283-291, Cambridge, MA, 1998.

[Nau98] Felix Naumann. Data Fusion and Data Quality. In *Proceedings of the New Techniques and Technologies for Statistics Seminar* (NTTS), Sorrento, Italy, 1998.

[New88] H.B. Newcombe. *Handbook of Record Linkage*. Oxford University Press, Oxford, UK, 1988.

[NF00] Felix Naumann and Johann Christoph Freytag. Completeness of Information Sources. Technical Report HUB-IB-135, Humboldt University of Berlin, February 2000.

[NL00] Mattis Neiling and Hans-Joachim Lenz. Data integration by means of object identification in information systems. In *Proceedings of European Conference on Information Systems*, Vienna, Austria, 2000.

[NL00a] Felix Naumann, Ulf Leser. Cooperative Query Answering with Density Scores. In *Proceedings of the Conference on Management of Data* (COMAD 00), Pune, India, 2000.

[NLF99] Felix Naumann, Ulf Leser, and Johann-Christoph Freytag. Quality-driven integration of heterogeneous information systems. In *Proceedings of the International Conference on Very Large Databases* (VLDB), pages 447-458, Edinburgh, UK, 1999.

[NR00] Felix Naumann, Claudia Rolker. Assessment Methods for information quality criteria. In *Proceedings of the International Conference on Information Quality* (IQ), Cambridge, MA, 2000.

[PAGM96] Yannis Papakonstantinou, Serge Abiteboul, and Hector Garcia-Molina. Object fusion in mediator systems. In *Proceedings of the International Conference on Very Large Databases* (VLDB), pages 413-424, Bombay, India, 1996.

[Pir74] Robert Pirsig. *Zen and the Art of Motorcycle Maintenance*. Bantam Books, New York, 1974.

[Pyl99] Dorian Pyle. Data Preparation for Data Mining. Morgan Kaufman Publishers, San Francisco, CA, 1999

[Red96] Thomas C. Redman. *Data Quality for the Information Age*. Artech House, Boston, London, 1996.

[Ros81] Arnon Rosenthal. Note on the expected size of a join. *SIGMOD Record*, 11(4):19-25, 1981.

[SM83] Gerard Salton and Michael J. McGill. Introduction to Information Retrieval. McGraw-Hill, Inc., New York, NY, 1983

[SS94] Arun Swami and K. Bernhard Schiefer. On the estimation of join result sizes. In *Proceedings of the International Conference on Extending Database Technology* (EDBT), volume 779 of *LNCS*, pages 287-300, Cambridge, UK, 1994. Springer.

[TB98] Giri Kumar Tayi and Donald P. Ballou. Examining data quality. *Communications of the ACM*, 41(2):54-57, 1998.

[TGO99] K.L. Tan, C.H. Goh, and B.C. Ooi. On getting some answers quickly, and perhaps more later. In *Proceedings of the International Conference on Data Engineering* (ICDE), pages 32-39, Sydney, Australia, 1999.

[Wei99] Gerhard Weikum. Towards guaranteed quality and dependability of information systems. In *Proceedings of the Conference Datenbanksysteme in Büro, Technik und Wissenschaft* (BTW), pages 379-409, Freiburg, Germany, 1999.

[Wie99] Gio Wiederhold. Trends for the information technology industry. Technical report, Stanford University under sponsorship of the Japan Trade Organization, October 1999.

[WS96] Richard Y. Wang and Diane M. Strong. Beyond accuracy: What data quality means to data consumers. *Journal on Management of Information Systems*, 12(4):5-34, 1996.

[WSKL99] Richard Y. Wang, Diane M. Strong, Beverly K. Kahn, and Yang W. Lee. An information quality assessment methodology. In *Proceedings of the International Conference on Information Quality* (IQ), pages 258-265, Cambridge, MA, 1999.

[YM98] C. Yu and W. Meng. *Principles of database query processing for advanced applications*. Morgan Kaufmann, San Francisco, CA, 1998.

[Zia99] Wojciech Ziarko. Discovery through rough set theory. *Communications of the ACM*, 42(11):54-57, November 1999.