ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

A. Avenali, C. Batini, P. Bertolazzi, P. Missier

A FORMULATION OF THE DATA QUALITY OPTIMIZATION PROBLEM IN COOPERATIVE INFORMATION SYSTEMS

R. 598 Ottobre 2003

Alessandro Avenali – Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", via Buonarroti 12 - 00185 Roma, Italy. Email: avenali@dis.uniroma1.it.

Paola Bertolazzi – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email: bertola@iasi.rm.cnr.it.

Carlo Batini – Dipartimento di Informatica e Sistemistica, Università di Milano Bicocca, Milano, Italy. Email: batini@bicocca.mi.it.

Paolo Missier – School of Computer Science, The University of Manchester, UK. Email: Missier@cs.man.ac.uk.

This work was partially done under financial support of MIUR - FIRB project "MAIS"

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

The execution of inter-organization business processes involves the exchange of large amounts of structured data across distributed and cooperative information systems. Since the quality of those processes is directly affected by the quality of the data they acquire from other processes, we introduce a model whereby processes may (i) state their requirements on the types and quality of incoming data (data and data quality demand), and (ii) advertise the type, quality and the cost of the data they can supply to other processes (data and data quality offer). Using this model, we formulate the problem of optimizing the cost of data exchange under quality requirements constraints. This is formalized as an Integer Linear Programming problem that outputs an optimal matching of data providers to data consumers. We also show how a quality monitoring service may provide the necessary input to the problem in a realistic setting.

1. Introduction

The problem of measuring and improving the quality of information resources is of particular interest in the area of Cooperative Information Systems (CIS in the following). In previous papers [**?**, **?**, **?**], we have proposed a model for describing both the structure of a cooperative system and its *Information Quality Profiles* (IQP). According to this model, every organization that is part of a CIS is described as a set of processes that transform input *information flows* into output flows that carry typed information items. In this way, each organization appears both as an information *supplier* that provides its own information *offer*, and an information *consumer* that has its own information *demand*. Additionally, the IQP associated to a flow defines quality measures along various quality dimensions, both for elementary data items (eg a salary figure) and for aggregated items (eg the average over a set of salaries) carried by the flow. By extension, the set of IQPs associated to the set of output flows of a supplier describes the overall data quality offered by that supplier. Thus, in addition to information demand, we can also model the information *quality* demand of a consumer, defined as a set of predicates over the quality of data items acquired from some supplier. These predicates describe the quality requirements of input data, which are normally determined by the needs of the data consuming process (this is termed *fitness for use* in [**?**]).

We make a few key assumptions. First, we assume that information has a cost, which is determined in part by its quality. As we show in a later example, realistic quality requirements do not ask for perfect information (eg 100% correctness, completeness, etc.), because attaining those levels is often impractical and expensive. We can offer two informal explanations. First, The typical goal of record linkage tools used in traditional data cleansing is to perform the largest possible amount of automatic record reconciliation, under constraints on the max acceptable number of false positives and false negatives. As these systems leave a set of undecided records for clerical review, we may informally depict the cost curve for the entire cleansing process has a flat line for the automated portion of data, which then skyrockets for the remaining manual portion. In this case, a reasonable cost-quality trade-off would limit the quality requirements to those corresponding to the knee of the curve. The approach described in [**?**] provides a recent example of such a cleansing tool. Another reason for sub-optimal requirements is that sometimes processes can deal with imperfect data. The literature on data quality in OLAP environments is a good source of examples.

We also assume that information is described using the relational model. Specifically, each organization manages a local relational schema, and the collection of local schemas is integrated into a single global schema. In this setting, information demand is defined as a query on the global schema, which can be decomposed into multiple local queries using global-to-local schema mappings. Information quality demand is also expressed on the global schema, while quality offer and corresponding cost are associated to the local relations.

In this work, we address the problem of matching information demand to information offer under quality constraints, minimizing on the cost.

In general, a local schema may contain *relation fragments* that represent a partial answer to the global query, and furthermore, those fragments may be offered by different suppliers with different quality levels and at a different cost. In order to satisfy the entire demand, it is necessary to collect fragments from multiple local schemas, possibly selecting among equivalent fragments with different quality and cost, in such a way that the quality requirements on the resulting whole are satisfied. Our technical approach consists of two steps. First, a query decomposition algorithm selects feasible fragments from the local relations, given a global query. Then, we formulate an Integer Linear Programming optimization problem that uses the selected fragments and produces a cost-optimal bag of fragments that satisfy the entire demand.

Although solving this problem is equivalent to solving its dual, namely maximizing on the quality at a fixed cost, we adopt a process-centric perspective in which minimal quality requirements driven by the organizations' processes determine the constraints.

The rest of the paper is organized as follows. A discussion of related work is presented in the next Section. In Section **??**, we provide formal definitions both for Information Demand and Offer, as well as for the Quality Model, and introduce our running example for a specific information demand and offer scenario. We also address how the quality features associated to attributes and relations may be

predicted from a time series of historical quality values observed in the course of a quality monitoring process. In Section **??** we present the main problem formulation. We address limitations of this work and open research issues in Section **??**.

2. Related work

The relation fragments that each information supplier contributes to the overall demand, correspond to horizontal fragments in standard distributed query processing. The problem of computing local fragments that contribute to the result set of a global query has been studied extensively in the literature. A general survey can be found in [**?**], where different algorithm and cost estimation models are presented, especially for optimization of processing, I/O and communication costs of distributed queries in homogeneous distributed data bases, both replicated and non replicated. Query processing in heterogeneous databases environments is surveyed in [**?**], where theoretical issues and algorithms are discussed both for the Global-As-View and the Local-As-View models [**?**]. The Bucket algorithm for query answering with materialized views, described in [**?**], is specifically relevant to our problem of computing local fragments that contribute to a global query.

Economic models for distributed query processing are provided in [**?**] for client server systems, where servers bid to execute parts of queries, and in [**?**] where a class of problems is discussed in which a function over a set of inputs is computed, where each input has an associated price. Also, a survey of several economic models can be found in [**?**]. Literature on economic aspects specifically related to Data Quality is not wide-spread. We mention two papers by Ballou[**?**], [**?**]. In [**?**], the economic relevance of data quality is taken into account and ongoing measurement of the quality of data produced and delivered to customers is established, in order to create a Total Quality Management System for data. In [**?**], the trade-off between the data quality level and the cost incurred in achieving it are investig ated, and an Integer Linear Programming (ILP) problem is formulated that allows to define quality-improving projects that target specific data sets while maximizing the *Information Utility*.

In our paper, we investigate the aspect of procurement of data with a given quality level at minimum cost. To the best of our knowledge, the optimization cost model proposed here has not been previously addressed in the literature. However, Naumann [**?**] presents an extension of a traditional algorithm for distributed query planning over heterogeneous sources, that adopts a mediator approach to integration and takes into account data quality features associated to local individual data items. Given a global query, the algorithm computes all semantically correct query plans on the local schemas, and then ranks those plans according to the estimated quality of the returned data, selecting the top-N plans for execution. In particular, quality estimation for a join relation is computed from the quality of the join operands. Our approach differs from Naumann's in that instead of producing multiple query plans, our algorithm identifies individual feasible fragments from the local relations, and then computes an optimal choice of those fragments using ILP.

The ILP approach itself takes inspiration from the mathematical model underlying the Facility Location Problem (see [**?**]). Note however, that information differs from physical commodities in that the "overlapping" of their sources represents an opportunity for creating a marketplace by diversifying offer (eg by cost and quality), rather than a potential for wasting resources.

3. Definitions

3.1. CIS Data Model

We model the structure of a CIS as a collection of organizations whose processes are able to exchange relational data among each other through information flows (Figure **??**).

To model data exchange, we assume that each organization manages a local relational schema, and that the collection of all local schemas has been integrated into a single global schema. The global schema represents the overall information content that suppliers offer to the consumers. Information flows carry relations that are the result of queries on the local schemas. A flow is determined by a data supplier and a data consumer *process*, plus the schema for the data carried by the flow.

Figure 1: Simplified CIS Model

Note that, in principle, this simple model may yield as many different flows as there are potential queries. In practice, however, the queries are defined by the system processes managed by each organization, which in many real business environments are stable over time, resulting in a limited number of pre-defined flows. In fact, one can envision an implementation model for the CIS in which organizations define stable service interfaces through which data of known types are exchanged. Such model has been adopted by the Italian Public Administration, to provide a number of e-Government services.

We model the relationship between global and local schemas by providing mappings for entities in the global schema into entities in each of the local schemas, in such a way that global queries posted against the global schema can be translated into a set of queries, each addressing one of the local schemas, using the mappings provided [**?**]. Formally, we are given a Global Schema:

$\texttt{GS} = \{R_1(A_{11}, \ldots, A_{1n_1}), \ldots, R_k(A_{k1}, \ldots, A_{kn_k})\}$

a set $LS_1, \ldots, LS_N$ of N local schemas:

$LS_j = \{R_1^{(j)}(A_{11}^{(j)}, \ldots, A_{1n_1}^{(j)}), \ldots, R_h^{(j)}(A_{h1}^{(j)}, \ldots, A_{hn_h}^{(j)})\}$

and a set of mappings from the global to the local schemas. Each mapping is an expression of relational algebra, that defines one relation $R_i^{(j)}$ of LS_j in terms of relations of GS. In the following, we adopt a schema-prefixed notation for local relation schemas, i.e, we are going to write $LS_j.R_i$ to indicate relation $R_i^{(j)}$ in local schema LS_j.

In our running example, for simplicity we are going to consider a global schema consisting of a single relation, called `Resident`, three local schemas and the mapping from the global to the local schemas. The example concerns records of the resident population of the Italian Region Lazio, comprising the three provinces of Rome, Latina and Rieti[1].

Note that the point of the example is not to show distributed query planning, but rather, to illustrate the selection of suitable relational fragments from each of the local schemas, without the use of joins.

GS consists of the single relation:

R = `GS.Resident(SSN, Name, Address, Province, Region, WS, EI)`[2].

The three local schemas are defined as follows:

Local schema LS1:

`LS1.Resident(SSN,Name, Address, Province, R, EI)`

This relation contains all and only the residents of the provinces of Rome and Rieti, and its schema contains a subset of the attributes in `GS.Resident`. Therefore, the mapping view from GS into LS1 is simply:

`LS1.Resident` = $\pi_{SSN,Name,Address,Province,Region,EI}$
($\sigma_{Province="Rome"\,or\,Province="Rieti"}$(`GS.Resident`))

Local schema LS2 for the provinces of Latina and, again, Rieti:

`LS2.Resident(SSN, Name, Address, Province, Region, WS)`

The mapping view from GS into LS2 is the following:

`LS2.Resident` = $\pi_{SSN,Name,Address,Province,Region}$
($\sigma_{Province="Latina"\,or\,Province="Rieti"}$(`GS.Resident`))

[1] In the Italian administrative legislation, Regions are partitioned into Provinces. Furthermore, we are simplifying for the sake of understanding. Lazio really has four provinces, not three.

[2] SSN is the Social Security Number, WS stands for "working status" and can be either "working" or "retired", and EI is the Estimated Income.

Finally, local schema LS3:
LS3.Resident(SSN,Name, Address, Province, EI).
This relation contains all and only the residents of Region Lazio. Therefore, the mapping view from GS into LS3 is the following:

LS3.Resident = $\pi_{SSN,Name,Address,Region,EI}(\sigma_{Region="Lazio"}$(GS.Resident))

We also note that the following *inter-schema properties*, i.e., additional semantic rules regarding the mappings, are defined:

Province = 'Rome' $\Rightarrow$ Region = 'Lazio'
Province = 'Latina' $\Rightarrow$ Region = 'Lazio'
Province = 'Rieti' $\Rightarrow$ Region = 'Lazio'
Province $\neq$ 'Latina' and Province $\neq$ 'Roma' and Province $\neq$ 'Rieti'$\Rightarrow$ Region $\neq$ 'Lazio'

These rules allow us to derive the following containment relationships among local schema relations:

$\pi_{SSN,Name,Address,Province,Region}$(LS1.Resident) $\subseteq$
$\pi_{SSN,Name,Address,Province,Region}$(LS3.Resident),
$\pi_{SSN,Name,Address,Province,Region}$(LS2.Resident) $\subseteq$
$\pi_{SSN,Name,Address,Province,Region}$(LS3.Resident),
$\pi_{SSN,Name,Address,Province,Region}$(LS1.Resident) $\cup$
$\pi_{SSN,Name,Address,Province,Region}$(LS2.Resident) =
$\pi_{SSN,Name,Address,Province,Region}$(LS3.Resident).

3.2. Information offer and demand

Information demand is simply defined as any query on the global schema. In our proposal, a global query is handled by a Global Query Processor (GQP for short), which translates it into a set of local queries, plus one relational algebra expression defined on the results sets returned by each local query, such that the relation resulting from the evaluation of the expression is a valid result for the global query. The GQP uses the global-to-local mappings to determine the local queries. We illustrate this informally, through the following example global query Q: *find the SSN, Name and Address of all residents of Rieti and Latina*:

Q: $\pi_{SSN,Name,Address}(\sigma_{province="Latina" or province="Rieti"\}}$(GS.Resident))

Q admits multiple decompositions of Q into local queries. According to standard distributed database theory, a horizontal relational fragment is the result of selection operations on base relations of a local schema LS, of the form $V = \sigma_C(R(\mathcal{A}))$ where C is a condition on the values of attributes in $\mathcal{A}$ (for simplicity, we only consider single-attribute conditions). Furthermore, an *elementary fragment* (EF) is a horizontal fragment in which the attributes mentioned in condition C are restricted to a subset of those appearing in the demand query condition. Intuitively, in our optimization problem, we first compute a set of candidate elementary fragments, possibly from different providers, and then select a subset of those fragments whose union yields a complete result set with minimal cost.

In our example, ERs represent residents of each of our three provinces, plus the combination "Rome" and "Rieti". We see that the following contributions are available:
For schema LS1:

LS1.EF1 = $\pi_{SSN,Name,Address}(\sigma_{Province="Rieti"}$(LS1.Resident))

(note that Latina cannot be offered)
For schema LS2:

LS2.EF1 = $\pi_{SSN,Name,Address}(\sigma_{Province="Latina"}$(LS2.Resident)),
LS2.EF2 = $\pi_{SSN,Name,Address}(\sigma_{Province="Rieti"}$(LS2.Resident)),

For schema LS3:

LS3.EF1 = $\pi_{SSN,Name,Address}(\sigma_{Province="Latina"}$(LS3.Resident))
LS3.EF2 = $\pi_{SSN,Name,Address}(\sigma_{Province="Rieti"}$(LS3.Resident))

The complete result relation R may now be obtained in a number of different ways:
R = LS3.EF1 ∪ LS3.EF2
R = LS1.EF1 ∪ LS2.EF1
R = LS3.EF1 ∪ LS2.EF2
R = LS2.EF2 ∪ LS3.EF1
and more.

In the rest of this paper, we are going to refer to this decomposition into elementary RFs.

3.3. CIS Quality Model and Quality Monitoring

As defined in the previous section, RFs represent the basic units of information offered by a supplier organization O in response to a specific demand. We now want to characterize their quality features, so that information consumers may define quality requirements along with their queries. Then, we are going to show how the quality of the information provided in response to a global query, can be computed in a bottom-up fashion from the elementary quality features associated to the RFs that are part of the complete result. This will lead us to our core problem, namely finding cost-optimal flows that connect suppliers with consumers under data quality constraints.

Extending the simple CIS data model introduced earlier along the temporal dimension, we can think of an information flow as a *temporal sequence* of RFs, each having a timestamp. We call each instance of RFs returned by a local query a *data event*. It is important to note that the same set of records contained in a EF, generated at different times (possibly, by the same query), are different data events.

For example, given a schema $R(Address)$, an instance of Address would be the set of all addresses of a certain population, that is sent by process p_1 to process p_2 at time t_i. If the same data set is sent again at a later time, according to our definition it originates a new data event. Thus, a data event is uniquely identified by its timestamp. Also, note that the same EF is represented by different data events when it is carried by different flows.

Our approach to computing quality features is to associate quality features to data events, much in the same way as quality is measured for individual product units coming out of a manufacturing production process. In the case of information, the same EF may exhibit different quality at different times because of changes in the production process, or in the input data, or in the characteristics of the data storage used to maintain it. The idea is then to collect time series of quality features for individual data events, and to use the historical data to predict the expected quality of the next data event. This is the quality that we are going to use to match consumer requirements.

We start from the widely adopted definitions of multiple *quality dimensions* [**?**, **?**], and assume that the observations are limited to a well-defined set of such objective dimensions[3], for instance syntactic and semantic data correctness, data obsolescence, and data completeness. A generic data event *de* is a relation of the form $r(a_1, \ldots, a_n)$. To each attribute a_i we associate a quality vector $qual(a_i) = \mathbf{q_i}$ of size m, with one element $\mathbf{q_{ij}}$ for each representative quality dimension.

In a previous work [**?**] we have provided further details on the quality models. In the same paper, we have shown how quality vectors carrying actual quality measurements can be associated to data events in practical situations, and how they can be collected into a quality monitoring data mart. For each EF, the data mart contains the quality vectors for each data event in which that EF has been observed over an information flow. We have also shown how the multidimensional data model chosen for the data mart can be used in an OLAP environment to compute various aggregations over the historical time series of the quality vectors. For instance, one may query the model to find the average correctness of any EF produced by organization O within a given time window, or to determine the minimum currency values for all Address attributes produced across any organization. This work is extended and applied here to the problem of matching quality offer, and demand, as explained in the next section.

[3]Objective dimensions are those for which the measure does not depend on the information customer's perspective, as it would eg for "understandability".

3.4. Information Quality offer and demand

To each individual attribute A_i of a local base relation schema we associate a vector $q_i \in \Re^k$ of quality features, where k is the (fixed) number of the quality dimensions of interest for the cooperative system. The quality vector associated to each attribute that appears in a EF is simply the quality vector from the same attribute in the corresponding base relation. The quality features associated to the entire query result are computed as a suitable aggregation function, eg by taking the average values for each quality dimension, on the quality features carried by the corresponding attributes of the composing RFs. Thus, the quality vector associated to a result set that is the union of n EF horizontal fragments, is obtained as the average of the quality vectors, element by element, of each EF in the union.

Note that, in order to keep the mapping simple, we have avoided RFs that result from the join of multiple local relations. While this has no impact on the problem formulation or the optimization model, it does affect the post-processing of the result. In this work, we are not going to investigate the use of joins in the model.

The choice of aggregation by averaging is only a simple illustrative example that is chosen as representative of a class of non-monotonic aggregation functions. Non-monotonicity is important in our approach because, as described later, it prevents the optimization algorithm from filtering out low-quality fragments. Although these fragments, when taken individually, do not satisfy the quality requirements, they may very well participate in a feasible solution once averaged with higher quality fragments. Note that individual quality vectors can be combined in more sophisticated ways, i.e., using quality merging functions as suggested in [**?**]. The issue of quality merging operators, however, is beyond the scope of this paper.

In addition to the quality vectors, an organization also associates a cost to each EF, which can be thought of as a "market value" attached by the supplier to the information fragment.

With this setup, we may better state the notions of offer and demand for information supplier and consumer organizations. Each organization (corresponding to each of the three local schemas in our example) may act either as a supplier of RFs and/or as a consumer of query results. Consumers state their demand as a pair of the form $(Q, [\hat{q}_1, \ldots, \hat{q}_l])$, where Q is a relational algebra expression on the global schema `GS`, with projection attributes $\mathcal{D} = \{d_1, \ldots, d_l\}$, and each $\hat{q}_i$ is a constant associated to $d_i \in \mathcal{D}$. Symmetrically, EF suppliers offer tuples of the form $(\mathrm{EF}_i, q_i, p_i)$. The quality offer q_i for EF_i is given by the set of quality vectors, one for each attribute, associated to EF_i, and p_i is the cost of offering the EF to a consumer.

An atomic *quality requirement* is an inequality expression at the individual attribute level, of the form $\hat{q}_i \geq q_i$, on the requested $\hat{q}_i$ and offered q_i quality vectors, respectively. Vector inequality is defined as the conjunction on the pairwise inequalities for each vector element, that is, separately on each quality dimension used in the Quality model. A quality requirement for the entire demand is the conjunction on the l atomic requirements.

4. Minimal Cost Supplying Problem Formulation

We now formulate the problem we are going to solve. Given demand $(Q, [\hat{q}_1, \ldots, \hat{q}_l])$, quality features q_i on the local schemas attributes, a set of RFs (horizontal fragments) whose union includes the complete result set, and a cost associated to those RFs, find the cost-optimal complete result set consisting of a subset of those RFs, that satisfy all quality requirements. Note that different RFs may belong to different supplier organizations. We call this the *minimum cost supplying* problem, or MCS for short.

The algorithm for solving MCS consists of two phases, informally sketched as follows. The main assumption is that quality demand can be satisfied by buying fragments from different providers, when this leads to lower costs. Hence, in the first phase, a set of local RFs and associated price tag and expected-quality vector is produced. Specifically, the global query processor identifies the elementary RFs that may contribute to the result of the global demand query, and local organizations are required to associate the price tag to each fragment they are willing to supply. Computing the elementary fragments can be accomplished using algorithms presented in [**?**]. The expected-quality vectors for each attribute in each of the fragments are computed by querying the quality monitoring data cube.

In the second phase, these RFs are used as input to formulate an Integer Linear Programming problem. Solving the ILP for minimal cost yields the best matching offer for the demand.

We illustrate the algorithm on our running example. Consider query `Q` above, with projected attributes `GS.SSN, GS.Name, GS.Address`. The consumer organization associates three quality requirement vectors $\hat{q}_{SSN}$, $\hat{q}_{Name}$, $\hat{q}_{Address}$ to these attributes. As we have noted in Section **??**, `Q` may be computed as the union of various combinations of elementary RFs. Since quality requirements expressed on individual attributes are oblivious of fragments, we must now map those requirements onto the possible fragments combinations. In our example, take for instance the union `R = LS1.EF1` $\cup$ `LS2.EF1`, and requirement: $q_{GS.Resident.SSN} \geq \hat{q}_{SSN}$.

Given quality vectors $q_{LS3.Resident.SSN}$ and $q_{LS2.Resident.SSN}$, the actual requirement can be expressed using a simple average function:

$\hat{q}_{SSN} \geq avg(q_{LS1.Resident.SSN}, q_{LS3.Resident.SSN})$

where the average is taken element-wise on the vectors[4]. Similarly, we combine quality vectors for each of the other attributes in the demand.

In the rest of this section, we formalize the second phase of MCS. Observe that, w.l.o.g, to have the global optimal solution we can solve many single optimization problems, one for each consumer, since the EF's are unlimited resources and hence the optimum for one consumer does not affect the optimum for the others. Therefore, our formulation is limited to the one-consumer, many-suppliers scenario.

Let $O_1, \ldots, O_n$ be supplier organizations, and for a given demand `Q`, let $S_1 = \{EF_1, \ldots, EF_{s_1}\}$, $S_2 = \{EF_{s_1+1}, \ldots, EF_{s_1+s_2}\}$, ... be the set of their candidate elementary RFs identified in the first phase. Recalling our earlier informal definition, a EF is a view on single relations of a local schema. For convenience, we now label the attributes in each view as c_i, and w.l.o.g. we index those attributes consecutively. Thus, EF are relations of the form:

$EF_1 = \{c_1, \ldots, c_{u_1}\}$, $EF_2 = \{c_{u_1+1}, \ldots, c_{u_1+u_2}\}$ etc.

In our previous example, we would have c_1 = `LS1.Resident.SSN`, c_2 = `LS1.Resident.Name`, and so forth. Let $\mathcal{C} = \{c_i, i = 1, \ldots, m\}$ denote the entire set of such attributes. For each c_i, the vector q_i of the supplied quality levels for the different quality dimensions, and the associated cost p_i are given.

On the consumer side, let $R = (d_1, \ldots, d_l)$ be the schema for the relation returned by a query, with $\mathcal{D} = \{d_1, \ldots, d_l\} \subseteq \mathcal{C}$.

For each attribute d_i, the consumer states a quality requirement $\hat{q}_i$, that is the minimum level of quality desired, for each value of that attribute and for each dimension. Regarding the result set for a query, we may have the following situations:

- one or more suppliers can fulfill the query and produce the entire result set, for each d_i, with the required quality level;
- no single supplier can fulfill the query, and hence the complete result set for a d_i can only be obtained using multiple horizontal fragments of d_i;
- at least one supplier fulfills the query, and additionally, other suppliers supply horizontal fragments for the attribute values.

As noted earlier, in the second and third scenarios, the overall value of quality dimension j for attribute d_i is computed as the mean value of quality values over all horizontal result fragments, possibly weighted by the cardinality of the fragments.

We formulate MCS as an Integer Linear Programming problem, using a bipartite graph to represent the information offer and demand scenario.

Let $G = (V, W, E)$ be a bipartite graph, where $W = W_1 \cup W_2$ and $E = E_1 \cup E_2$. V nodes represent the consumer side, while W nodes represent available fragments. As we will see later, $W_1 = \{w_i\}$ is the set of vertices that represent fragments provided by suppliers, $W_2 = \{\bar{w}_j\}$, is the set of vertices that are associated to the attributes in $\mathcal{C}$, E_1 (E_2) is the set of edges connecting W_1 (W_2) with V.

[4]Alternatively, we may weight the average using the relative cardinalities of each of the fragments. We are not going to discuss the pros and cons of this alternative in this work.

Let l be the number of elementary fragments found by the GQP (i.e., 5 in our example). Note that not all fragments can be combined to produce a complete result; in our example, it makes no sense to take the union of two `Rieti` fragments (by construction, each such fragment contains all the required residents of that town). Thus, node matching must take into account the *types* of the fragments, let them be $\mathcal{FT} = \{ft_1, \ldots, ft_m\}$. We recall that this information comes from phase I and from the QGP.

For each $d_k \in \mathcal{D}$ and fragment type $ft_h \in \mathcal{FT}$, we introduce node $v_{kh} \in V$, corresponding to one attribute and one fragment type for that attribute. In the example, we have two nodes for $d_k =$ `SSN`, namely v_{k1} for `"SSN / Rieti"` and v_{k2} for `"SSN / Latina"`.

$W_1 = \{w_i\}$ is the set of vertices that represent fragments provided by suppliers, where w_i is associated to a whole fragment EF_i, and $W_2 = \{\bar{w}_j\}$, where each vertex $\bar{w}_j$ is associated to the c_j-th attribute in $\mathcal{C}$.

To each vertex v_{kh} is associated the quality requirement vector $\hat{q}_k$ associated to attribute d_k, while vector q_j of offered quality is associated to each vertex $\bar{w}_j$.

Note that, on the supplier side, there is one vertex for each fragment of each requested attribute that can produced, and in particular, if attribute d_i can be offered by multiple suppliers, then multiple attribute nodes appear in W_2, one for each supplier.

Edge $(v_{kh}, w_i) \in E_1$ iff organization O_i supplies a fragment EF_i with attribute d_k, and EF_i is of fragment type h. Edge $\{v_{kh}, \bar{w}_j\} \in E_2$ iff the attribute c_j of the supplied fragment EF_i coincides with d_k, and EF_i is of fragment type h.

Finally, to each vertex $w_i \in W_1$ the cost p_i that O_i requires to supply EF_i is associated.

Given the graph $G = (V, W, E)$ the MCS problem is formulated as a ILP problem, associating one binary variable x_i to each vertex $w_i \in W_1$ that is equal to 1 if the associated fragment EF_i is sent to O, and 0 otherwise. Moreover, a variable y_j is associated to each vertex $\bar{w}_j \in W_2$, that is equal to 1 if the corresponding attribute is chosen. The problem can be formulated as follows. Let $|V| = l$ $|W_1| = n$ and $|W_2| = m$.

$$min\ z = \sum_{i=1}^{n} p_i \times x_i$$

subject to the following constraints:

$$y_j \leq x_i \qquad \forall c_j \in EF_i \tag{1}$$

$$\sum_{j:(v_{kh}, \bar{w}_j) \in E_2} y_j = 1 \qquad \forall v_{kh} \in V \tag{2}$$

$$\sum_{j:(v_{kh}, \bar{w}_j) \in E_2, \forall h: ft_h \in d_k} q_j y_j \geq \hat{q}_k \times \sum_{j:(v_{kh}, \bar{w}_j) \in E_2, \forall h: ft_h \in d_k} y_j \tag{3}$$

$$x_i \in \{0, 1\} \tag{4}$$

$$y_j \in \{0, 1\} \tag{5}$$

Constraint (**??**) means that if an attribute from a EF is chosen, then the entire EF is chosen.

Constraint (**??**) says that each fragment request must be satisfied by exactly one supplied attribute Constraint (**??**) means that if the instance of a required attribute can be obtained as a merge of horizontal fragments of the instance itself, provided by different suppliers, then the overall quality of the merge is given by the (possibly weighted) medium value of the horizontal fragments' qualities, where F_t is the number of suppliers that provide the horizontal fragments. Observe that constraint (**??**) must be replicated for each quality dimension.

To illustrate, in Figure **??** the bipartite graph for our running example is shown, with respect to the SSN attribute, and a subset of the constraints of our ILP formulation is shown. In particular, constraints 2 and 3 are shown only for the fragment of Rieti, and the quality constraint is written only for one dimension.

Figure 2: Construction of the bipartite graph for the running example

Observe that the problem is NP-hard as the underlying location problem is NP-hard and constraint (**??**) has a similar role to that one played by capacity constraints in the capacitated location problem. However the size of real cases is usually such that a branch and bound algorithm can solve them in reasonable time.

5. Further work

In this paper, we have proposed an integer linear formulation to solve a problem of cost-optimal provisioning of information in a cooperative environment, with required quality features.

The model does not address a number of issues that are the subject of our current research:

1. Quality propagation. The improvement of information quality in input to the consumers has an impact on the information quality in output, resulting in a progressive improvement of the information assets for each participating organization;

2. Algebra of quality. Fragments requested by consumers can be obtained as a function of elementary fragments of the suppliers, according to a given relational expressions. The quality of the resulting fragment is obtained by composing the quality of elementary fragments. In this paper, we restrict our attention to the composition of elementary horizontal fragments, that is, we only admit relational expressions that contain selections, projections and unions among fragments, and compute the resulting quality by simply averaging on the quality of the fragments. In order to handle more general cases, namely the use of joins and more sophisticated quality composition, we need to develop an algebra for quality composition;

3. A general cost model. The cost model used in this paper assumes a fixed cost for an information fragment with a given quality, independently of the number of requesting organizations. Two more general cost models should be investigated. In the first, we assume that individual organizations may partner with each other in order to aggregate their demand and thus obtain a "volume discount" on cost. In the second, the cost is a function of offered quality.

These three issues introduce different kinds of inter-dependencies among consumers, and create a feedback loop between consumers and suppliers. We are currently investigating alternative mathematical models, such as game theory, to handle the more general problem.

References

[BG99] D. Ballou and G.K.Tayi. Enhancing data quality on data warehouse environments. *Communications of the ACM*, 42(1), January 1999.

[CFG+00] M. Charikar, R. Fagin, V. Guruswami, J. Kleinberg, P. Raghavan, and A. Sahai. Query strategies for priced information. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (2000), Special issue of JCSS for STOC*, 2000.

[CLLR03] A. Cali, D. Lembo, M. Lenzerini, and A. Rosati. Source integration for data warehousing. In M. Rafanelli, editor, *Multidimensional databases: problems and solutions*. Idea Group Publishing, 2003.

[DRHG98] D.Ballou, R.Wang, H.Pazer, and G.K.Tayi. Modeling information manufacturing systems to determine information product quality. *Journal of Management Sciences*, 44(4), April 1998.

[FNSY96] D. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. Economic models for allocating resources in computer systems. In S. Clearwater, editor, *Market based Control of distributed systems*. World Scientific Press, 1996.

[Hal01] Alon Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, December 2001.

[Kos00] D. Kossmann. The state of the art in distributed query processing. *ACM Computing Surveys*, 32(4):422–469, December 2000.

[MAAA03] M.Buechi, A.Borthwick, A.Winkel, and A.Goldberg. Cluemaker: a language for approximate record matching. In *Proceedings of the Eight International Conference on Information Quality (ICIQ-03)*, Cambridge, MA, 2003.

[MB03a] P. Missier and C. Batini. An information quality management framework for cooperative information systems. In *Procs. ISE 2003*, Montreal, Canada, July 2003.

[MB03b] P. Missier and C. Batini. A model for information quality management in cooperative information systems. In *Procs. SEBD03*, Cetraro, Italy, June 2003.

[MB03c] P. Missier and C. Batini. A multidimensional model for information quality in cooperative information systems. In *Proceedings of the Eight International Conference on Information Quality (ICIQ-03)*, Cambridge, MA, 2003.

[MPW+96] M.Stonebraker, P.Aok, W.Litwin, A. Pfeffer, A. Sah, J.Sidell, C. Staelin, and A. Yu. Mariposa: a wide-aread distributed database system. *The VLDB Journal*, 5(1):48–63, January 1996.

[NLF99] Felix Naumann, Ulf Leser, and Johann Christoph Freytag. Quality-driven integration of heterogenous information systems. In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie, editors, *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 447–458. Morgan Kaufmann, 1999.

[RMW83] Francis R.L., L.F. McGinnis, and J.A. White. Locational analysis. *European Journal of Operational Research*, 12:220–252, 1983.

[RMY01] R.Y.Wang, M.Ziad, and Y.W.Lee. *Data Quality*. Advances in Database Systems. Kluwer Academic Publishers, 2001.

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

[Ull97] Jeffrey D. Ullman. Information integration using logical views. In Foto N. Afrati and Phokion G. Kolaitis, editors, *Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 1997.

[WR96] Y. Wand and R.Wang. Anchoring data quality dimensions in ontological foundations. In *Communications of the ACM*, volume 39. ACM, 1996.