Managing Source Quality Changes in Data Integration Systems

Adriana Marotta[1], and Raul Ruggia[1]

[1] Universidad de la República, Facultad de Ingeniería, Instituto de Computación,
Julio Herrera y Reissig 565, Montevideo, Uruguay. Fax: (598 2) 7110469

{amarotta, ruggia}@fing.edu.uy

Abstract. Data Integration Systems (DIS) integrate information from a set of heterogeneous and autonomous information sources and provide this information to the users. We consider a system where quality properties are taken into account. At the sources there are actual values of the quality properties and at the integrated system there are expected values of these properties. In this kind of system, regarding the possible large quantity of sources and their autonomy, a new problem arises: the changes in the quality of sources. We are interested in the consequences that source quality changes may have on the system quality, and even on the design of the DIS. We analyze these consequences taking ideas from the different possibilities for managing the source schema changes. We also study two quality properties in particular; freshness and accuracy, and we define strategies for managing source changes in these properties.

1 Introduction

Data Integration Systems (DIS) integrate information from a set of heterogeneous and autonomous information sources and provide this information to the users through a mediator schema. These systems basically consist of: (a) a set of autonomous sources, (b) a Mediator, which may have materialized or virtual data, and (c) the definition of a transformation process, which is applied to the information extracted from the sources. We represent the transformation process as a graph where each node is an activity and each edge is the data flow from one activity to another, following the idea of the *calculation dag* defined in [1]. For us an activity may represent a relational algebra operation or a more complex procedure, like a cleaning task or any data transformation.

We assume that the retrieval of information from the sources to the mediator is started simultaneously at all the sources. In the case of a virtual mediator, all the sources involved in each view definition are queried simultaneously at the moment of the user query execution.

We consider a system where quality properties are taken into account [2]. At the sources, *actual values* are associated to each relation for each quality property. At the mediator, quality requirements are considered. They are the *expected values* (or *re-*

quired values) and are associated to the mediator's relations for each quality property. In Figure 1 we show the described system.

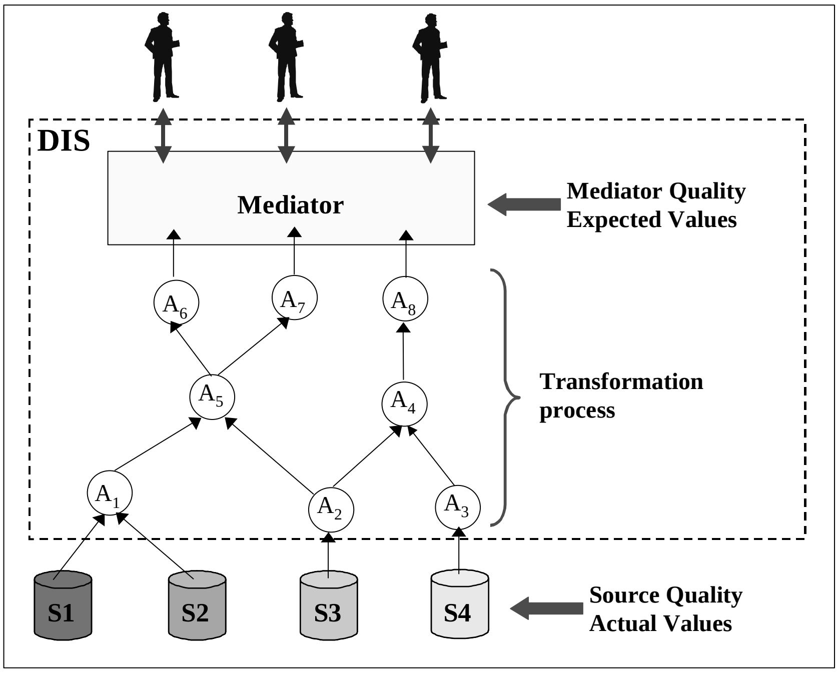
DIS
Mediator
Mediator Quality Expected Values
A6
A7
A8
A5
A4
Transformation process
A1
A2
A3
S1
S2
S3
S4
Source Quality Actual Values

Figure 1: System architecture

The sources' actual values are the quality values that are provided by the sources and that determine the ones provided by the mediator to the users. From the expected values of the mediator, expected values or ranges of expected values of the sources can be deduced. From the combination of the actual and expected values, very important design decisions can be made pointing to the improvement of the quality of the DIS (such as the ones related to source selection, or view implementation).

In this kind of system, regarding the possible large quantity of sources and their autonomy, a new problem arises: **the changes in the quality of sources**. Actual values of the source elements can change very frequently and in a non-predictable manner. The values of the quality properties in general do not evolve going in certain direction, as we can imagine when considering evolution in schemas. It is for this reason that we do not use the term "evolution" for source quality properties.

We identify two approaches to deal with the problem of source quality changes: the proactive approach and the reactive approach. The first one consists in the actions we can perform before the occurrence of a change in order to prepare the system for the change. We use techniques for informing the sources how their quality values can vary without failing to satisfy the quality requirements of the DIS, and techniques based on probabilistic models and intended to predict source changes and evaluate the system quality reliability. The reactive approach focus on the actions we can take after a source quality change occurs, having as objectives the satisfaction of the DIS quality requirements and the minimization of the impact of the change on the DIS.

This paper focuses on the reactive approach. The goal of this work is twofold: (i) to characterize some situations of source quality changes management, and (ii) to give some preliminary strategies for managing source changes on two quality properties, freshness and accuracy.

The existing work in the area of quality in information systems includes many different approaches and focalizations. We can find definitions and classifications of quality properties, for example in [3] and [4]. In [5], [6], and [7] the problem of quality in a Data Warehouse context is treated, while in [8], [9], and [2] they focus on managing quality in multiple and heterogeneous information systems. There are works that concentrate in one or two quality properties and their impact in the system design, in [10], [11] and [1] freshness and some other quality property are studied in deepness.

As far as we know, the problem of changes in quality values in data integration systems has not been explicitly addressed. Near to this area, we have only found the work in [12], which provides a taxonomy of schema evolution operations and the quality properties that are affected by each of them. However, we found very helpful for applying to our context, the techniques existing in the literature about source schema evolution in this kind of systems. There are many papers that focus on this problem, such as [13], [14], [15], which present analysis and solutions for the view adaptation problem, and [16], [17], which also focus on the source schema evolution and its propagation to the system.

The rest of the paper is organized as follows. Section 2 presents a classification of cases of source quality changes management, in Section 3 there is an analysis of two quality properties: freshness and accuracy, and finally in Section 4 we present the conclusions.

2 Source Quality Changes Management

We believe there is an analogy between source schema evolution management and the way source quality changes can be managed. In this section we show how some ideas and techniques of the former can be reused in the latter.

2.1 Borrowing Solutions from Source Schema Evolution Management

We synthesize the cases of source schema evolution and their solutions in the following three possible situations:

(i) Source schema changes are propagated to the mediator schema generating changes on this schema and also changes on the transformation process. Figure 2 shows an example of this situation. There are three sources providing data to the relation *Doctors* of the Mediator. In source *S3* the relation *Symptoms* is deleted and this change causes the deletion of the attribute *common_symptoms* from *Doctors*, and its associated transformations.

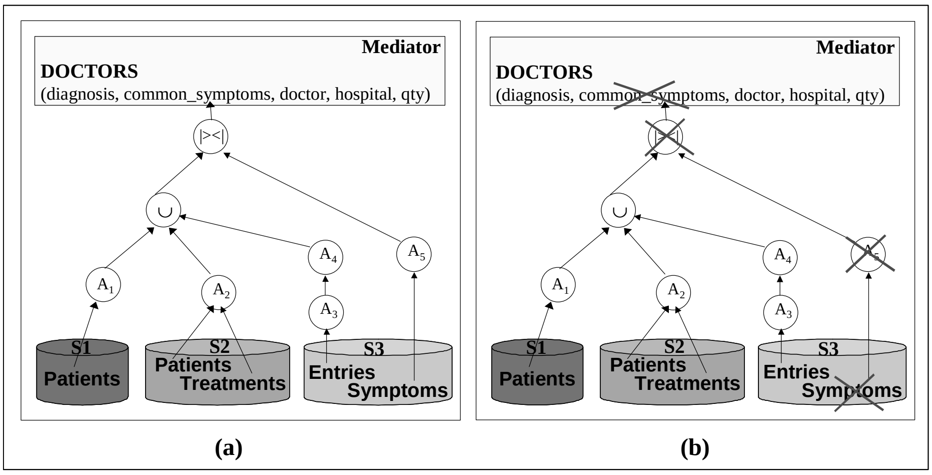

Figure 2: Change on Mediator schema and mappings (i)
(a) Before change **(b)** After change

(ii) Due to a source schema change, the transformation/mappings are modified in a way that they absorb the changes, and the mediator schema is not modified. Figure 3 shows an example of this situation. In this case relation *Patients* of source *S2* is changed and must be eliminated from the system. This causes the modification of the mappings and the elimination of source *S2* from the system (including relation *Treatments*). The schema of the mediator relation *Doctors* is not modified.

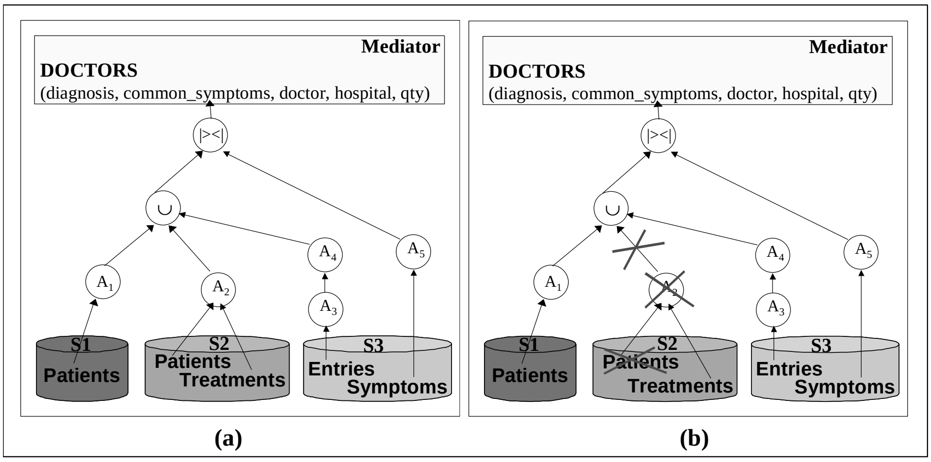

Figure 3: Mappings absorb the change (ii)
(a) Before change **(b)** After change

(iii) A source schema change causes a change in the schema of other source relation. Figure 4 shows an example of this situation. In this example the attribute *hospital* of relation *S1.Doctors* is deleted. This attribute was the join attribute with the relation *S2.Hospital*. In order to maintain the mediator schema without alterations the source *S2* is changed, adding the attribute *doctor* to relation *Hospitals* of *S2*. Obviously the transformation process is also modified changing the join attribute.

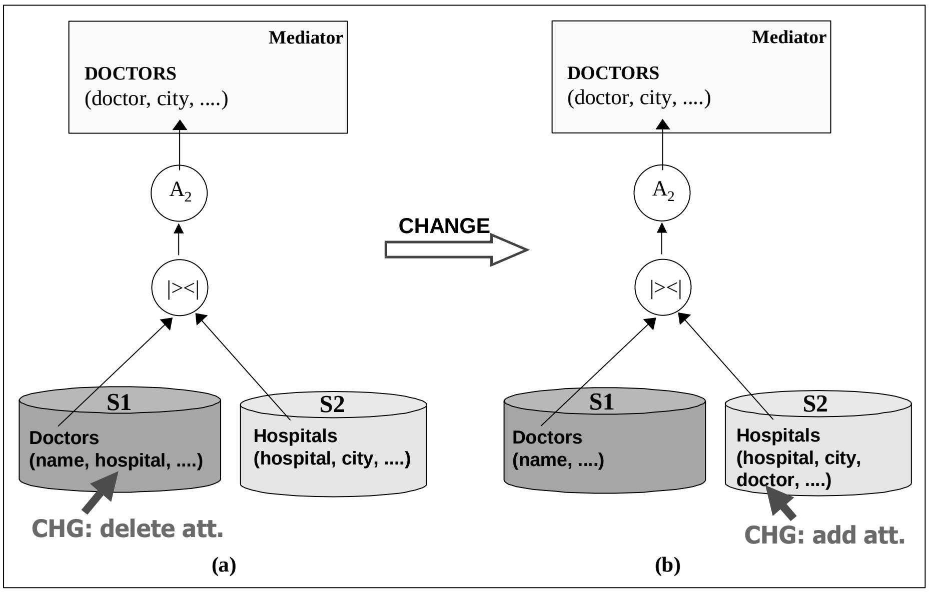

Figure 4: Change on other source (iii)

2.2 Dealing with Source Quality Changes

We investigate the problem of source quality changes and its possible solutions making the analogy with the previously presented situations.
The propagation of source quality changes may be treated by applying two different but not excluding strategies. The first one consists on the propagation of the source quality changes affecting only the quality values of the system. In the second one, this propagation includes modifications on the system schemas and transformation processes, i.e. on the system design. This propagation is more complex if the new source quality value causes that the mediator required quality values are no longer satisfied.

In the following two sections we present these strategies and their possible applications, referring to items (i), (ii) and (iii) of the previous section.

2.2.1 Strategy 1: Propagation affecting only quality values of the system

When a source quality change occurs, the first strategy we may consider is to re-accommodate the quality values of the system without affecting its design. We find three possible situations that are analogous to the ones presented for source schema evolution.

Situation 1:

In [2] it is proposed a mechanism to propagate quality actual values to the mediator in order to evaluate the system quality. Following the same idea, source quality changes can be propagated, modifying the mediator quality actual values. This situation is analogous to the situation (i) of source schema evolution. See the example of Figure 5. In this example quality actual values (QAV) for the property *accuracy*, are shown. The value in source *S1* changes from *0.8* to *0.7*. After the propagation to the mediator, the actual value in the mediator changes from *0.7* to *0.6*.

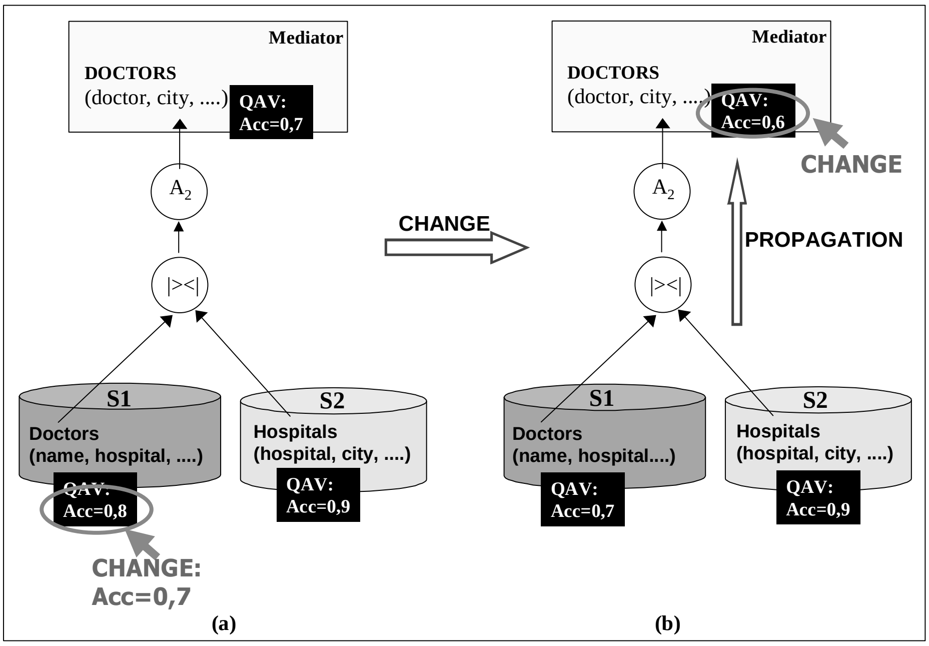

Figure 5: Propagation of quality change from source to mediator (i)

Situation 2:

In some cases, depending on the quality property, changing some quality values of intermediate nodes of the transformation, avoids the change in the mediator quality values, analogously to (ii). In fact, if the mediator quality values change, but they continue satisfying the mediator quality requirements, it is also an analogous situation, where the transformation "absorbs" the quality change.

Consider, for example, the *freshness* quality property. Intuitively, the freshness actual value of a mediator relation is calculated summing the actual freshness of the source relation plus the sum of the processing costs of the activities plus the synchronization delays between activities, for each path that goes from a source relation to the mediator relation. The maximum of these sums (or path costs) is the actual freshness of the mediator relation. See [1] for technical details.

Figure 6 shows an example of this case. The freshness expected value of the mediator relation *Doctors* is *24* and the maximum value of freshness that the source relations can have for satisfying the expected value is *20*. The relation *S2.Patients* changes its actual value from *19* to *21*. The consequence of this change is that the activity A_3 changes its cost from *3* to *2*, modifying the synchronization delay existent in the path of *Patients* from *3.5* to *2.5*, and assuring the satisfaction of the freshness requirements.

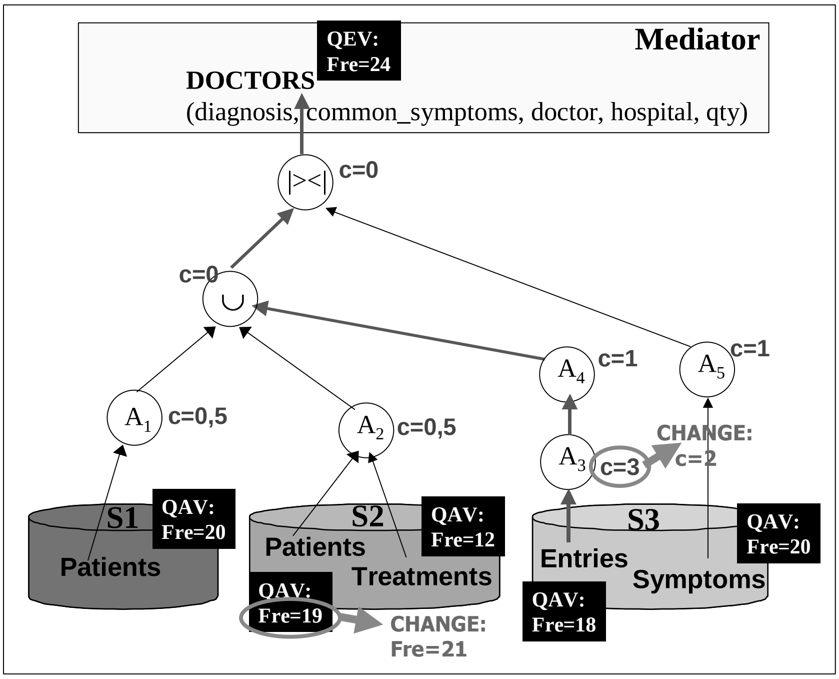

Figure 6: The transformation quality values absorb the change (ii)

Situation 3:

Analogously to (iii), as a consequence of a source change another source may be affected, i.e. the quality values of some other source may be forced to change for compensating the previously occurred change. In Figure 7 we show an example similar to the one of Figure 5, but where the value of the mediator must not be changed. It is supposed that the change in the value of source *S1* is unavoidable, while source *S2* may change its value. Based on the expected accuracy value existent in the mediator, the minimum accepted value for source *S2* is calculated (following the propagation strategy of [2]).

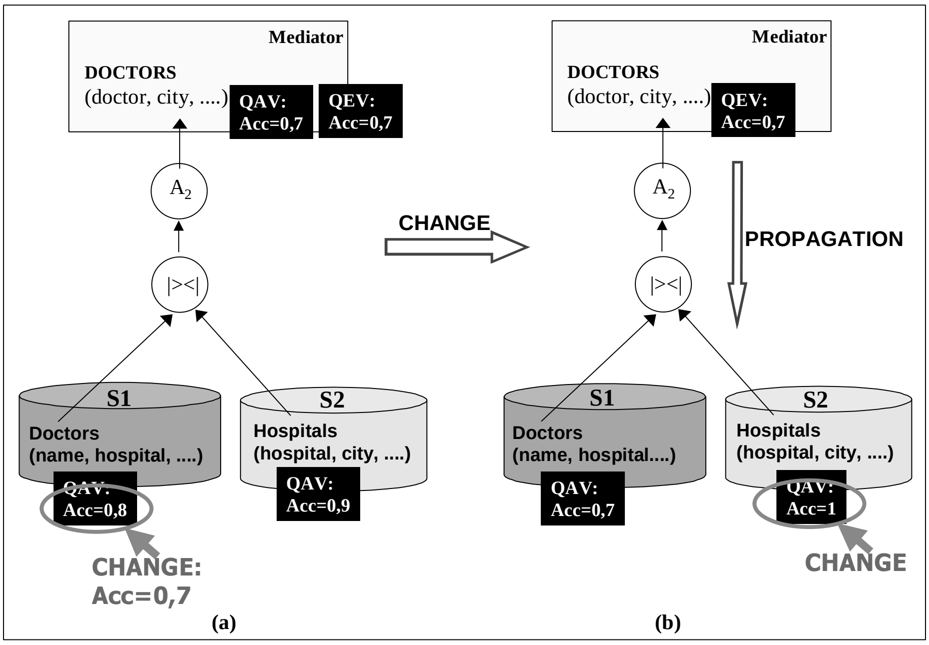

Figure 7: Change in a source affects values in another source (iii)

2.2.2 Strategy 2: Propagation affecting system design

When the application of the first strategy is not sufficient to propagate the source quality change, maintaining the satisfaction of the DIS quality requirements, the second strategy may be applied. In this one, changes on the transformation and/or on the mediator schema are generated. The possible propagations, in these cases, always involve one of the following: (a) the changed source element is eliminated from the system, or (b) the changed source element stays in the system but it is processed differently, in a way that the quality values are compensated.
The cases that involve (a) derive in the situations (i), (ii) or (iii) of source schema evolution. For instance, consider the examples of Figures 2 and 3. In them, the elimination of the source relations could be caused by a quality change that could not be propagated satisfactorily to the system. Analogously, in the example of Figure 4, the deletion of the attribute *hospital* in source *S1* could be caused by the same reason. However, the problem of the propagation of the source schema change is more complex when we consider quality issues. We must find a new configuration of the system that, not only minimizes the mediator schema changes and the loss of information from the sources, but also satisfies the quality requirements.
The cases that involve (b) are included in (ii), since in them the transformation process absorbs the change. See an example in Figure 8. In this example the value of the *accuracy* of *S1.Doctors*, changes from *0.8* to *0.7*. The propagation in this case is done by adding an activity *Cleaning* to the transformation process, which is applied to the information coming from *S1.Doctors*, improving its *accuracy* value.

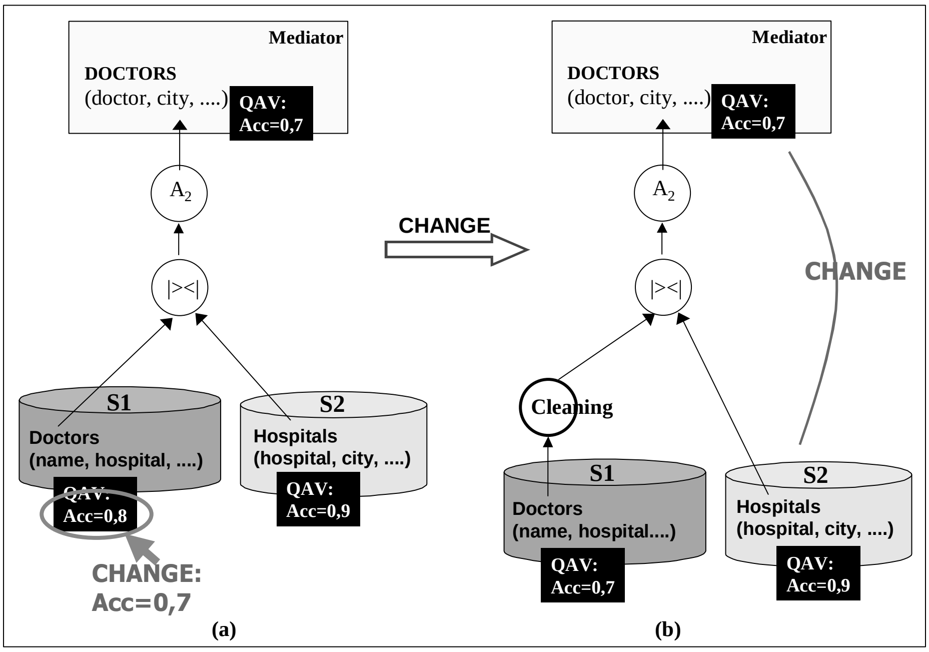

Figure 8: Change in a source causes a change in the transformation process.

3 Two Particular Cases: Freshness and Accuracy Quality Properties

In the previous sections we presented general classifications for the management of a source quality change, which allow having an overview of the whole universe of situations that may appear after the change. However, it is difficult to give behavior patterns or rules, and detailed solutions, applicable to the majority of the quality properties. Therefore we believe that for studying more deeply the problem of changes it is necessary to start considering one property at a time, since each property has a particular behavior, and affects the system differently.

In this section we focus on two quality properties: freshness and accuracy. We give their particular characteristics, and we present some ideas for managing source changes in each case. We also show how they affect each other when their values change. We have chosen these properties because they have very different characteristics, such as the form of propagation in the system or the parameters that affect them.

3.1 Properties Description

For each property we present the definition, propagation, and behavior. The definition is selected from the ones existing in the literature. The propagation is the mechanism we use to calculate the actual values of the property in the mediator from the actual values of the property in the sources. The behavior we describe refers to how the features of the different components of the system affect the values of the property in a source and vice versa. The definition of the properties and their propagation mechanism are not our focus, we choose them so that they are simple but interesting. We concentrate on studying the behavior with the perspective of defining strategies for change management.

In the following paragraphs we use the terms *required* (or *expected*) *values of the sources.* As said before, these values can be obtained from the expected values of the mediator. In the case of **freshness**, the required value in a source relation is the **maximum** value it can have in order to satisfy the DIS quality requirements. In the case of **accuracy** this value is the **minimum** value.

Freshness

Definition: Time elapsed between the production of the data and the moment it is read. [8] [1]

Propagation: Considering the graph of activities defined in [1], actual freshness of a mediator relation R is calculated by the following expression:

Freshness(R) = max {Freshness(A_0) + PathCost([A_0, A_1, …, A_p]) / [A_0, A_1, …, A_p] is a path from any source node A_0 to relation R},

where: A_0 represents a source relation.

PathCost([A_0, A_1, …, A_p]) = $\Sigma_{i=1..p}$ (cost(A_i)) + $\Sigma_{i=1..p}$ (sync(A_{i-1}, A_i)),

cost(A_i) is the processing cost of the activity A_i, and

sync(A_{i-1}, A_i) is the time that A_i must wait after the execution of A_{i-1} finishes, in order to synchronize with the other activities whose output is an input for it (for A_i).

Behavior:

- For each mediator relation we consider several paths, each of which goes from a source relation to the mediator relation [1]. One of them is the *critical path*, which fixes the minimum freshness actual value that can be reached by the mediator relation. For us, the *critical path* is the one who has the maximum $\Sigma_{i=1..p}$ (cost(A_i)), where A_1… A_p are all the activities that belong to the path. The other paths have their synchronization delays fixed in function of the critical path activities costs. Therefore the pathcost is the same for all the paths. This means that decreasing the cost of activities that do not belong to the critical path, does not lead to an improvement in the freshness actual value of the mediator relation. In Figure 6 it can be noted that the critical path is the one which goes from the source relation *Entries* to the mediator relation *Doctors*.

- The cost of the paths, which is determined only by the costs of the activities that belong to the critical path, affects the required values of the sources. In the example of Figure 6, the expected freshness value of the relation *Doctors* is *24*. As the cost of the paths is *4*, the required freshness value at the sources is *20*. If the cost of the paths was *8*, the required value at the sources would be *16*.
- The actual value of a source does not affect the required value of other source. For example, in Figure 6, when the actual value of relation *S2.Patients* changes, the required value in the other sources continues being the same. Moreover, we cannot change the required value of *S2.Patients* by changing the actual value of some other source.

Accuracy

Definition: The accuracy of a relation is the percentage of correct tuples over the total of the tuples of the relation [8]. We express these values with a number between 0 and 1, which corresponds to the percentage divided by 100.

Propagation: The formula for calculating the actual values of the mediator from the actual values of the sources, is different depending on the type of activity. The value of accuracy changes after the application of an activity if the activity receives more than one relation as input or if the activity performs a cleaning process. For instance, if the activity is a join operation, S1 |><| S2, that gives as result the relation R:

Accuracy(R) = Accuracy(S1) * Accuracy(S2)

If the activity is a cleaning process with S as input and R as output, which corrects the incorrect portion of the relation in a 50%:

Accuracy(R) = Accuracy(S) + (1-Accuracy(S)) * 0.5

Behavior:

- The accuracy actual values of the input relations, in each activity that has more than one input relation, combine differently depending on the type of activity. For this reason, both the type of activity and the actual values of the input relations affect the accuracy value of the result.
- The actual value of a source affects the required value of other sources. See the example of Figure 7. Here, due to a change in relation *Doctors* of source *S1*, the required value in source *S2* changes from *0,9* to *1*.
- Due to the fact that the actual value of a source affects the required value of other sources, when we search for the required values (minimum) of the sources we find that it exists a *solution space* where each solution is a combination of required values for the different sources. For example, in Figure 7,

it can be noted that one solution of the solution space is <Acc(S1)=0.8, Acc(S2)=0.9> and other solution is <Acc(S1)=0.7, Acc(S2)=1>.

- In cleaning activities the effectiveness of the process affects the accuracy value of the result.
- The effectiveness of the cleaning activities affects the required values of the sources; not only on the source to which the cleaning is applied, but also on other sources. Consider the example of Figure 8, part (b). Suppose that the data coming from relation *Doctors* of source *S1* changes its characteristics and the effectiveness of the cleaning activity decreases, and we need to maintain the accuracy value in the mediator. If the actual accuracy of relation *Doctors* cannot be augmented, then the required accuracy of relation *Hospitals* of source *S2* will augment.

3.2 Change Management

Taking into account the properties' characteristics described in last section, we deduce some techniques for propagating source quality changes to the DIS, minimizing the impact on the system and its quality. We apply the strategy presented in section 3.1, where the propagation affects only quality values of the system.

In the following we consider that the DIS required quality is being satisfied, and at a certain moment a change in one of the sources' quality values occurs, causing the non-satisfaction of this requirement. We describe the possible actions to be taken in order to solve this problem.

Freshness

Scenario of change:

The actual freshness of a source relation $S_1.R_1$ changes and the DIS does no longer satisfy the required freshness of a certain mediator relation. The actual freshness of $S_1.R_1$ is greater than its required value, let *t* be the difference between the actual value and the required value.

Alternative actions:

– The starting time of the retrieving of data from the sources towards a mediator relation is, a priori, the same at all the sources. In the case where $S_1.R_1$ does not belong to the critical path, we may change this starting time only at the source that belongs to the critical path, setting it a time *t* before the rest of the sources. This change decreases the synchronization delays, and therefore the pathcosts, of the rest of the paths, achieving the required freshness at the mediator. This solution is easier to apply in the context of a materialized mediator. In the case of a virtual mediator, we should anticipate the execution of the corresponding critical path for each user query.

– Improve the quality of an activity that belongs to the critical path, decreasing its cost a time *t*. This is possible if there exists an activity in the critical path that can be implemented more efficiently maintaining its semantics.

Accuracy

Scenario of change:

The actual accuracy of a source relation $S_1.R_1$ changes and the required values of the sources are no longer verified by them.

Alternative actions:

– Search for another valid solution (a solution from the solution space) that can be satisfied by all the sources' actual values. One of the following may happen:

- There is one solution that is verified by all the sources' actual values. In this case the required values are actualized for all the sources.
- There is not a solution that is verified by all the sources' actual values, but it is possible to negotiate with certain subset of the sources, asking them to improve their accuracy values. In this case it is possible to achieve a change in some source that allows verifying one of the valid solutions.
- There is not a solution that is verified by all the sources' actual values, and it is not possible to achieve a source change that allows verifying one of the valid solutions.

– Change activities that perform a cleaning task, improving their effectiveness, i.e. augmenting the percentage of information that is corrected. The improved activity may be an activity that processes information from the changed source, $S_1.R_1$, or from another source, $S_i.R_i$. The latter can be useful if data from $S_i.R_i$ combines with data from $S_1.R_1$ somewhere in the activity graph.

Freshness and Accuracy

Freshness and accuracy are not totally independent properties.

It is true that a change on one property does not automatically generate a change on the other one. However, we note they are not independent when we consider the possible consequences of managing a change in one of them. A change in one of the properties may indirectly cause a change on the other property, since when we modify the system in order to improve the values of one property, the values of the other property may get worse.

A clear example of this behavior is the following. Suppose that a source changes its value of accuracy and as a consequence, a cleaning activity that exists in the graph is modified achieving more effectiveness. The modified activity is at the same time more expensive and affects the cost of the paths to one of the mediator relations, causing that the required freshness at this relation is no longer satisfied. Finally, in order to satisfy the required freshness, a determined source must decrease its value of freshness, or, which is yet worse, the cost of certain activity is decreased having as secondary effect that the accuracy decreases too. In the latter we probably would enter in an infinite loop.

Considering the previously explained situation, we believe that changes on each of these two quality properties must not be managed independently. It is necessary to take into account how the actions performed on the system as a consequence of a property change affect the other property.

4 Conclusions

In order to have a general view of the problem of source quality changes in DIS, we made a classification of the possible situations that may be generated in this context. We present this classification making an analogy with the situations we find in the problem of source schema evolution. We show examples for illustrating each of the presented situations.

We also show the existence of two different strategies for managing the changes, a strategy where the propagation of the source quality change is limited to modify only quality values of the DIS, and a strategy where the propagation can cause changes on the system design. The first strategy has the advantage of not modifying the system design, however there are cases where it does not achieve the satisfaction of the DIS quality requirements. The second strategy is much more complex, since it manipulates the system structures and processes, but it is capable to correct the system so that it satisfies the quality requirements in spite of the occurred source change.

We study the behavior of two quality properties, freshness and accuracy, focusing on the inter-relations between each property and the system. Based on these, we propose some techniques for the management of the source freshness and accuracy values changes. Finally we analyze how the two properties inter-relate with each other, when the proposed techniques for change management are applied.

The first overview of strategies for managing source quality changes, presented in this paper, shows that dealing with these changes is a complex task. Therefore, while we are working on specifying solutions for this problem, we are also working on the pro-active approach. In this approach the idea is to take actions preventively, taking benefit from existing techniques such as probabilistic methods.

We are also currently working on the prototyping of a framework that allows trying the techniques over different scenarios, by means of modeling the transformation graph of a DIS, the algorithms for the calculation of the quality values at the different layers of the information system, and the quality changes applied to the sources. The framework allows testing with different DIS characteristics and different scenarios of changes.

References

1. V. Peralta, R. Ruggia, Z. Kedad, M. Bouzeghoub *A Framework for Data Quality Evaluation in a Data Integration System*. 19º Simposio Brasileiro de Banco de Dados (SBBD'2004). Brasil, October 2004.
2. A. Marotta, R. Ruggia. *Quality Management in Multi-Source Information Systems.* II Workshop de Bases de Datos. Jornadas Chilenas de Computación. Chile. Nov. 2003.
3. D.M. Strong, Y.W. Lee, R.Y. Wang. *Data Quality in Context.* Communications of the ACM. May 1997/Vol.40, No.5.
4. Y.W. Lee, D.M. Strong, B.K. Kahn, R.Y. Wang. *AIMQ: A Methodology for Information Quality Assessment.* Forthcoming in Information & Management, published by Elsevier Science (North Holland). (Accepted in November 2001)
5. M. Jarke, Y. Vassiliou. *Data Warehouse Quality: A Review of the DWQ Project.* Invited Paper, Proc. 2nd Conference on Information Quality. MIT, Cambridge, 1997.
6. M.A. Jeusfeld, C. Quix, M. Jarke. *Design and Analysis of Quality Information for Data Warehouses.* ER 1998: 349-362
7. M. Helfert, C. Herrmann. *Proactive Data Quality Management for Data Warehouse Systems.* DMDW 2002: 97-106
8. F. Naumann, U. Leser, J.C. Freytag. *Quality-driven Integration of Heterogenous Information Systems.* VLDB 1999: 447-458
9. M. Mecella, M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci, C. Batini. *The DaQuinCIS Broker: Querying Data and Their Quality in Cooperative Information Systems.* J. Data Semantics I 2003:208-232
10. L. Bright, L. Raschid. *Using Latency-Recency Profiles for Data Delivery on the Web*. In Proc. of the 28th Int. Conf. on Very Large Databases (VLDB'02), China, 2002.
11. D. Theodoratos, M. Bouzeghoub. *Data Currency Quality Factors in Data Warehouse Design.* In Proc. of the Int. Workshop on Design and Management of Data Warehouses (DMDW'99), Germany, 1999.
12. C. Quix. *Repository Support for Data Warehouse Evolution.* DMDW'99.
13. Rundensteiner, Lee, Nica. *On Preserving views in evolving environments.* KRDB'97.
14. Nica, Lee, Rundensteiner. *The CVS Algorithm for view synchronization in evolvable large-scale information systems.* EDBT'98.
15. Nica, Rundensteiner. *View Maintenance after View Synchronization.* IDEAS'99.
16. Bouzeghoub, Kedad. *A Logical Model for Data Warehouse Design and Evolution.* DaWaK'00
17. M. Bouzeghoub, B. Farias Lóscio, Z. Kedad, A.C. Salgado. *Managing the Evolution of Mediation Queries.* CoopIS/DOA/ODBASE 2003: 22-37