An Abstract Processing Model for the Quality of Context Data

Matthias Grossmann, Nicola Hönle, Carlos Lübbe, and Harald Weinschrott

Universität Stuttgart, Institute of Parallel and Distributed Systems,
Universitätsstraße 38, 70569 Stuttgart, Germany
{grossmann,hoenle,luebbe,weinschrott}@ipvs.uni-stuttgart.de

Abstract. Data quality can be relevant to many applications. Especially applications coping with sensor data cannot take a single sensor value for granted. Because of technical and physical restrictions each sensor reading is associated with an uncertainty. To improve quality, an application can combine data values from different sensors or, more generally, data providers. But as different data providers may have diverse opinions about a certain real world phenomenon, another issue arises: inconsistency. When handling data from different data providers, the application needs to consider their trustworthiness. This naturally introduces a third aspect of quality: trust. In this paper we propose a novel processing model integrating the three aspects of quality: uncertainty, inconsistency and trust.

1 Introduction

Applications that process sensed data or integrate data from different independent data providers need to handle data with varying quality. To these applications it is crucial to measure data quality. Moreover, a measurement of quality can be beneficial for both applications and data providers. Applications can use it to exclude data that does not satisfy user needs and data providers could incorporate the quality of the provided data into their pricing policies. Often the quality of data hints at the costs of providing the data. It might be more expensive in terms of energy to provide an accurate, up-to-date data value of a sensor than an imprecise, possibly outdated value. Especially context-aware applications running on resource-limited mobile devices often have to trade quality against resource-consumption. These context-aware applications are the focus of the Nexus project [1].

A lot of research has been done on the subject of data quality. In most cases a metric of a certain quality aspect like uncertainty is used to define quality. In the context of the Nexus project, we have investigated three different aspects of quality: uncertainty, inconsistency and trust. In this paper we integrate all three aspects of quality into a single processing model.

The paper is organized as follows. Sect. 2 introduces the Nexus middleware and motivates the choice of the three quality aspects. Sect. 3 gives an overview of the related work. We define the three aspects of quality, namely certainty,

Published in Proceedings of the 1st International Workshop on Quality of Context (QuaCon'09), pp. 132-143. Stuttgart, Germany, June 2009

http://dx.doi.org/10.1007/978-3-642-04559-2_12

consistency and trust separately in Sect. 4 and we introduce operators used for formulating queries and an example scenario in Sect. 5. In Sect. 6, we explain the reasons for integrating the three quality aspects and present and evaluate a suitable processing model. Section 7 concludes with directions for further research.

2 The Nexus System

In the Nexus project [1], we provide a framework for managing global context models in an open platform, where a multitude of context data providers can integrate and share their context models. Due to the global characteristics and the high number of different context providers, our system is based on a distributed and scalable architecture.

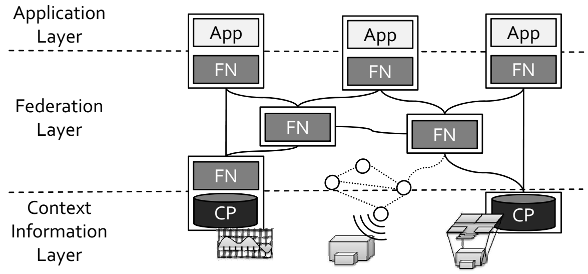

Fig. 1. Architecture of the Nexus system [1]

We depict a simplified three-layer architecture of our system in Fig. 1. The bottom layer, i.e., the context information layer, consists of context data providers (CP) offering context information from various sources ranging from static information to sensor values. Thereby, different context providers may provide data with different levels of detail. In addition, this data can be based on different kinds of sensors. These two characteristics are the reason to specify the *uncertainty* of the data. Moreover, the fact that several context providers may provide data on the same phenomenon is the reason to specify the *inconsistency* of this data. Finally, in this open system, information about the *trust* in context providers is essential to estimate the value of the provided data. We explain the details of these three aspects of quality in Sect. 4.

The middle layer, i.e., the federation layer, is the platform for processing queries on the data provided by the context information layer. Thereby, the federation nodes (FN) on this layer provide the abstraction of a single data source to the applications (App) in the application layer. Processing of data quality is done based on the currently available data at the different providers. This processing is not influenced by limitations through network characteristics, e.g., interpolation mechanisms are incorporated to cope with high network delay.

3 Related Work

Research has been done on quality of context data in various ways, however, distinguishing between different quality aspects is a novel concept. Papers typically talk about quality of data, meaning either certainty or consistency, and treating trust as a different issue. As we will outline in Sect. 4, our concept of certainty is mostly related to the area of sensor data and moving object databases (continuous domain, infinite number of alternatives) and consistency to the area of uncertain databases (discrete domain, finite number of alternatives). In the following, we present some works on quality of context data and show that our models, since they are based on these works, build on accepted knowledge.

Uncertainty is caused, e.g., by the characteristics of update protocols or trajectory simplification for position data or by the impreciseness of sensors. Probability density functions (PDFs) are commonly used to represent uncertain data. E.g., in case of GPS sensors[1], the PDF directly reflects the measuring accuracy. For uncertainty caused by update protocols, it is possible to specify a range and assume uniform distribution [2]. Although PDFs do not provide the most accurate means for representing uncertainty in all cases [3], we will use this well-known uncertainty model in this paper. Some papers in the area of uncertain data only consider simple range or nearest neighbor queries, e.g. [2, 4], while others use more complex composed queries, e.g. [5, 6]. From these two publications, we adopt the idea of composed queries and a one-dimensional (spatial) domain. In addition, for measuring uncertainty, we apply the well-known concept of differential entropy [7] as in [8].

In the area of inconsistent data, i.e., scenarios with a finite number of alternatives for a value, a concept sometimes called the possible worlds model is frequently used, e.g. in [9–12]. The advantage of the possible worlds model is that it formally can be used on top of the relational algebra without modifying the operators. Conceptually, for each possible combination of alternative values, a separate instance of the database (a possible world) is created and the query (consisting of standard relational algebra operators) is evaluated on each instance resulting in a set of alternative results.[2] We also rely on this model, but extend it by allowing uncertain attributes with PDFs. Cheng et al. also use the possible world model together with PDFs [8], but only regarded queries that can be classified either as returning a single value represented by a PDF or as returning entity sets, represented by possible worlds, so no actual integration of PDFs in the possible worlds model is required. For measuring inconsistency, a lot of related works [13, 14] define distance-based metrics, which is also the idea of our model.

For modeling trust in context data providers, we use a simplified variant of the model from [15], which is based on Jøsang's opinion triangle [16].

---

[1] http://telecom.tlab.ch/ zogg/Dateien/GPS_Compendium(GPS-X-02007).pdf

[2] n values with alternatives in a database result in $O(2^n)$ possible worlds, so this model can be used to define the semantics of query processing, but in general not to actually implement it.

To the best of our knowledge, there is no work that tries to define a generic reference model for processing quality of context data, combining the different quality aspects and providing a single expressive interface for applications to quality of data.

4 Three Aspects of Quality of Data

In the following discussion, $o, o_1, o_2, \ldots$ denote objects. Objects are sets of attributes. The P attribute of o_1 is denoted by $o_1.P$. Here, we only regard attributes (called P) with scalar values, representing not only, e.g., temperature or other sensor measurement values, but also – as in the following discussion – position values in a one dimensional space. This is primarily for simplicity, but, depending on the data model, can also practically be used, e.g., for representing positions of cars on a highway [5].

Different data providers can manage the same object. We call the data providers $1, 2, \ldots$, and $o_1^2.P$ denotes the position of object o_1 according to provider 2.

The following definitions are chosen such that greater values correspond to an increase of the named quality aspect, i.e., greater values mean more *un*certainty, more *in*consistency and more trust.

4.1 Uncertainty

Imprecise sensors like GPS are the reason for uncertainty. So sensor values in general are not given exactly, but through a range of values. How this range is given depends on the sensor. In the following we assume that a probability density function (PDF) is given, but in the future, we plan to integrate more flexible representations.

We assume that data providers specify a normal PDF, however, due to the way we handle data of not fully trusted providers when fusing the quality aspects (cf. Sect. 6), we want to be able to express that, with some probability, we are not sure or do not know the value.

Definition 1. *An uncertain position P is represented by a special PDF $p : \mathbb{R} \to \mathbb{R}_0^+$ with $0 \leq \int_{-\infty}^{\infty} p(x)\, dx \leq 1$. With the probability $1 - \int_{-\infty}^{\infty} p(x)\, dx$, the value is unknown (NULL).*

Besides representing uncertain positions, we also require a means for measuring how uncertain a position is. For this, we adopt the concept of differential entropy from [7], which was already used for measuring quality of data in [8]. To be able to use this definition, we restrict the position PDF to have a lower bound l and upper bound u, with

$$p(x) \begin{cases} > 0, l \leq x \leq u \\ = 0, \text{otherwise} \end{cases} .$$

Definition 2. $u(P) = -\int_l^u p(x) \log_2 p(x)\, dx$ *is the uncertainty of position* P*.*

This definition restricts the form of the PDF and may not be adequate for cases where the probability for the value being NULL is greater than 0, however, as shown in Sect. 6, we only apply this definition to values directly retrieved from data providers, where these limitations are reasonable.

4.2 Inconsistency

Inconsistency occurs when different data providers offer the same datum, e.g., different sensors measure the same datum, or the buildings in a town are modelled from different organizations. This leads to a finite number of alternatives for one value. For measuring the inconsistency of two positions, we use the arithmetic mean of the smallest possible distance and the largest possible distance between the positions:

Definition 3. *The smallest and largest possible distance between two positions* P_1 *and* P_2 *are* $d_{\min} = \max(0, \max(l_2 - u_1, l_1 - u_2))$, $d_{\max} = \max(u_1, u_2) - \min(l_1, l_2)$*. The inconsistency of the two positions is*

$$i(P_1, P_2) = \frac{d_{\min} + d_{\max}}{2} \quad .$$

4.3 Trust

We consider data providers to be differently reliable. The reliability of a data provider cannot be constituted globally, because it depends on the user and its preferences. In the Nexus project, we model trust as a triple (belief, disbelief, ignorance), where the three values are from the interval [0,1] and their sum is 1. In the following discussion we use a simplified version, where the disbelief value is always 0. In this case, it is sufficient to specify the belief value b (we trust the provider), the ignorance value (we cannot decide) is $1 - b$.

Definition 4. *The trust value of data provider* i *is given by* $b(i)$, $b : \mathbb{N} \to [0, 1]$*.*

5 Query Processing

As mentioned in Sect. 3, we use the possible worlds approach as basis for the query processing, but need to be able to represent uncertainty, so we have to extend the model to support an infinite number of possible worlds. This is subject to ongoing research, but for queries with simple selection predicates, the approach shown in Fig. 2 is reasonable. In addition to enumerate a finite number of possible worlds (boxes in Fig. 2), we allow uncertain attributes in a possible world (grey circles representing positions), so that a possible world in our model can represent an infinite number of exact possible worlds (shown in the bottom part of Fig. 2). In contrast to the original possible world model, we need to adapt operators for our approach. Fig. 2 shows a range query, which only a part of the o_1s represented by $PW1$ fulfills, so the result of applying the query to $PW1$ is an empty possible world ($PW3$), and a possible world with a modified position for o_1 ($PW2$).

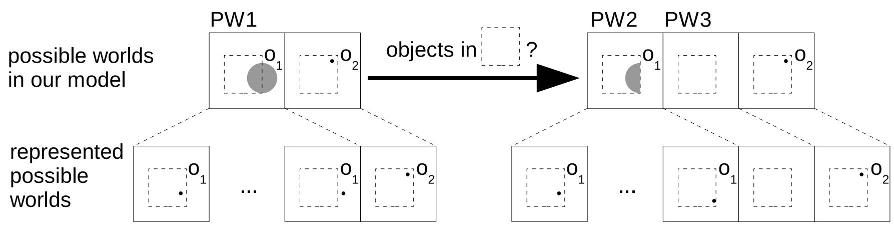

Fig. 2. Extending the possible worlds model to support uncertainty

The Nexus system is not only able to simply retrieve objects, but can also process more complex queries. It provides a set of generic operators, which is similar to the relational algebra. The precise definition of the complete set is beyond the scope of this paper, but we briefly describe the operators used in the example scenario. Note that these operators only handle uncertainty, we explain in Sect. 6, why this is sufficient.

Selection σ_{pred}**:** The selection operator is equivalent to the selection operator of the relational algebra. It takes a list of objects as input and outputs a list containing all objects from the input list, which fulfill the predicate *pred*. When applied to uncertain data, objects fulfill the predicate with some probability, and objects are included in the result list with this probability, i.e., σ can create several alternative results (possible worlds) and is an entity-based non-aggregate operator according to the classification in [8]. As previously explained, it may be necessary to modify uncertain attribute values. NULL values are handled as in SQL: When *pred* evaluates to *unknown*, the object is not included in the result.

Sorting $sort_{expr}$**:** The sorting operator sorts a list of objects. *expr* is an expression based on attributes of an object. It is evaluated for each object in turn, and the objects are sorted according to the results. Like the selection operator, sorting can create several alternative results when applied to uncertain data. The probability of a result list is determined by the probability that evaluating *expr* in sequence on all objects of this list results in a sorted list. Sorting is an entity-based aggregate operator.

Fetch $fetch_n$**:** The fetch operator just cuts a list of objects to the first n objects. It does not evaluate attributes like the other two operators do, thus does not require an adaption to handle uncertain data. We use the fetch operator in conjunction with sorting to implement a nearest neighbor query.

5.1 Example Scenario

Fig. 3 shows the example scenario. Two providers 1 and 2 store two objects o_1 and o_2. For o_2, each provider offers a representation, these two representations are different.

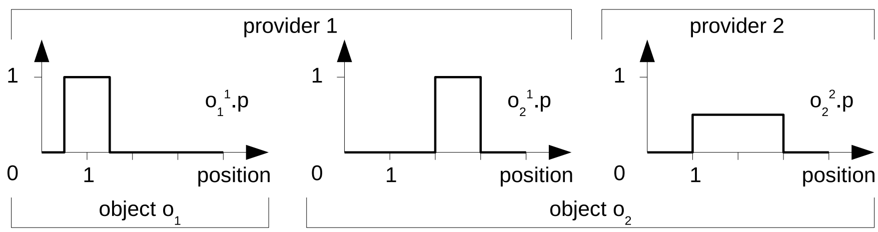

Fig. 3. Example scenario: PDFs of the positions of o_1 and o_2

The uncertainties of the positions are $u(o_1^1.P) = u(o_2^1.P) = 0, u(o_2^2.P) = 1$, the inconsistency of $o_2.P$ is $i(o_2^1.P, o_2^2.P) = 1$. We want to answer the query, which of the objects located between the positions 1 and 3 is closest to position 0, more formally

$$fetch_1(sort_{dist(0,o.P)}(\sigma_{1 \leq o.P \leq 3}[o_1, o_2])) \ .$$

dist calculates the distance between its arguments. As in this scenario, the first argument is the position 0, the result has the same PDF as the second argument.

As explained above, it may be necessary to adapt the PDF for the position during selection. For a selection of the form $\sigma_{l \leq P \leq u}$, we do this the by narrowing the range, where the PDF is > 0, to the interval $[l, u]$ and multiplying the resulting function with a constant factor, so that the integral equals to 1:

$$p'(x) = \begin{cases} \frac{p(x)}{\int_l^u p(x)\,dx}, & l \leq x \leq u \\ 0, & \text{otherwise} \end{cases}$$

For evaluating *sort*, we must calculate the probability that a distance D_2 is greater than an other distance D_1. When D_1 and D_2 are represented by two PDFs d_1 and d_2, the probability for $D_2 > D_1$ is[3]

$$\int_{-\infty}^{\infty} \int_{x_1}^{\infty} d_1(x_1) d_2(x_2)\,dx_2\,dx_1 \ . \tag{1}$$

In the given scenario, we cannot be sure if o_1 actually fulfills the selection predicate, and – according to the data of provider 2 – there is a chance that

[3] $d_1(x_1)d_2(x_2)$ is the combined PDF for D_1 and D_2. To derive the probability for $D_2 > D_1$, we need to integrate over the area where $x_2 > x_1$.

o_2 is closer to 0 than o_1. Obviously, the probability for o_1 to be closer to 0 is much higher than for o_2. However, the probability for o_1 to fulfill the selection predicate is only 0.5, so we expect the probability for o_2 being the result of the query to be only a little bit above 0.5.

6 Processing Model

In Sect. 4, we presented approaches for representing and measuring uncertainty, inconsistency and trust on the data provider level. To be able to process complex queries like the one presented in the previous section, we need to address two additional questions: how to account for the quality aspects during the processing of queries and how to measure the quality of the final result set.

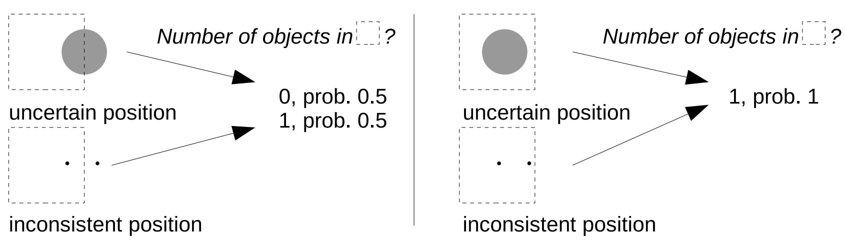

Fig. 4. Measuring the quality of the query result

The straightforward attempt to solve the first problem would be to define separately for each operator, how each quality aspect is handled. When, e.g., the selection operator is applied to an uncertain attribute, the uncertainty selection operator would be invoked, and for an inconsistent attribute the inconsistency selection operator. However, this approach cannot handle information that is both uncertain and inconsistent, like $o_2.P$ in Fig. 3. Therefore, we need a more integrated concept, which can deal with all three quality aspects simultaneously.

To measure the quality of result sets, in some cases, it is possible to directly apply the definitions to query results. When an object with an uncertain position is present in a result set, the uncertainty model presented in Sect. 4.1 can be used to represent its position. Likewise, the inconsistency model from Sect. 4.2 can be used when two different values for the same attribute of an object are in the result set. However, when only one value qualifies for the result set, the inconsistency information gets lost. To use the trust model from Sect. 4.3, the trust value for the provider has to be assigned to each attribute he provides to the result. However, in more complex situations, these definitions are not suitable. Figure 4 shows on the left hand side a situation where we are not sure if the answer to the query *How many objects are located inside the dashed square?* is 0 or 1. This should somehow be reflected by the result's quality, but is unclear if this is uncertainty or inconsistency, because exactly the same result can be caused by uncertainty (top) or by inconsistency (bottom). On the right hand side, we have the same situation with a slightly shifted square for the query. In this case, we can be sure that the result of the query is 1, so the quality of the result should be optimal, although the data used for answering the query is uncertain or inconsistent.

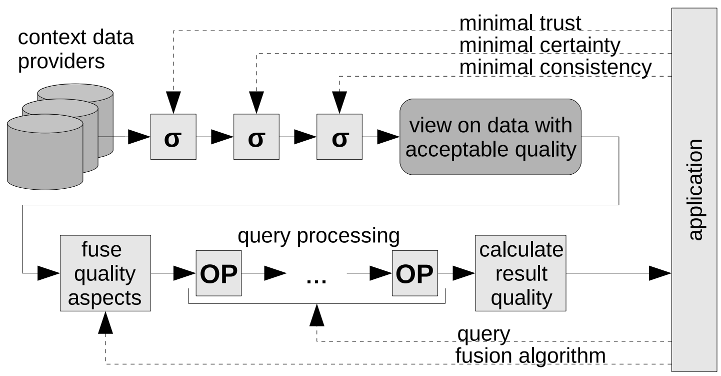

Fig. 5. Processing model

To address these two problems, we are investigating the approach depicted in Fig. 5. The main idea is to combine the three aspects before the actual query processing takes place, and define query processing and the result's quality based on the possible worlds model. In the following, we discuss reasons for choosing this approach.

Viewed from the perspective of query processing, uncertainty and inconsistency describe similar phenomena – there exist several alternatives for one value. In the case of uncertainty, the number of alternatives is possibly infinite, whereas in the case of inconsistency, a finite number of alternatives exist. In that sense, uncertainty is a generalization of inconsistency and both can be expressed by an uncertainty model.

When expressing inconsistency as uncertainty, we basically add all PDFs for an attribute. Thereby we have to weight the individual PDFs of the data providers, in the most simple case with the reciprocal of the number of data providers. In our case, however, we can refine the weighting using the trust values, so that PDFs from trustworthy data providers gain a higher weight than those from lesser trusted ones. This meets the supposable expectation of users that information of trustworthy data providers is more likely to be true.

In more detail, the approach consists of the following steps:

1. Applications or users may want to specify minimum requirements for certainty, consistency and trust for the data used for processing the query. Three additional selections are performed before the actual query is processed which result in a subset of the original data set, that fulfills the quality

constraints. Note that the selection of sufficiently trusted data providers has to be done before evaluating the consistency constraint, otherwise, untrusted providers would be able to force the removal of attributes from the subset by providing incorrect representations of the attribute, thus decreasing the consistency.

2. The three quality aspects are combined based on the uncertainty model. When the providers $1, \ldots, n$ provide values for the position of an object o, the resulting position is

$$o.p(x) = \frac{1}{n} \sum_{i=1}^{n} b(i)(o^i.p(x)) \ .$$

Inconsistency is incorporated by averaging the representations, trust by weighting them. Note that $\int_{-\infty}^{\infty} o.p(x)\,dx$ may be smaller than 1 (cf. Sect. 4.1). In some cases, applications may require a different fusion algorithm, so we provide a way for the application to specify the algorithm to use.

3. The calculation of the quality of the query's result is still an open issue, but using some extension of an entropy based approach seems to be promising. It is not necessary to use an additional selection here, the application itself can decide whether the quality of the result is sufficient or not and discard the result in the latter case.

An additional benefit of this approach is that the combined data quality model is closely related to models typically used in the literature, which allows us to define the semantics of our operators based on well understood concepts.

6.1 Revisiting the Example Scenario

In this section, we describe how the query in Sect. 5.1 is processed using our processing model. We use the notation $[o_1, \ldots, o_n]_p$ for a result list generated with probability p.

For the first example, we trust each data provide fully, i.e., $b(1) = b(2) = 1$ and we do not use restrictions on certainty, consistency and trust. Thus, fusing the data of the two providers results in $o_1 = o_1^1$ and o_2 with

$$o_2.p(x) = \begin{cases} 0.25, & 1 \leq x < 2 \\ 0.75, & 2 \leq x \leq 3 \\ 0, & \text{otherwise} \end{cases} \ .$$

Fig. 6 shows the intermediate results after each operator of the query and the final result. σ does not modify o_2, because its position lies completely inside the requested area, o_1, however, becomes o_1' with

$$o_1'.p(x) = \begin{cases} 2, & 1 \leq x \leq 1.5 \\ 0, & \text{otherwise} \end{cases} \ .$$

o_1' is closer to 0 than o_2 with a probability of $\frac{15}{16}$ according to (1). The probability of o_2 being the final result of the query is a little bit higher than 0.5 as expected in Sect. 5.1.

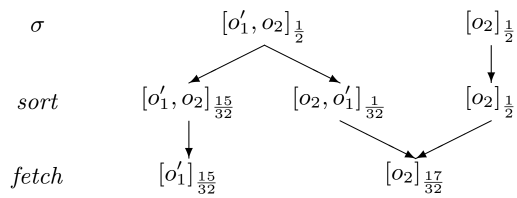

Fig. 6. Processing the query $(b(1) = b(2) = 1)$

For the second example, shown in Fig. 7, we use the trust values $b(1) = 0.5$ and $b(2) = 1$. This results in the following situation after fusing the data:

$$o_1.p(x) = \begin{cases} 0.5, \; 0.5 \leq x \leq 1.5 \\ 0, \quad \text{otherwise} \end{cases} \quad o_2.p(x) = \begin{cases} 0.25, \; 1 \leq x < 2 \\ 0.5, \quad 2 \leq x \leq 3 \\ 0, \quad\; \text{otherwise} \end{cases} .$$

Because we do not fully trust provider 1, $o_1.P$ is NULL with probability 0.5 and $o_2.P$ with probability 0.25. o_1 fulfills the selection predicate with a probability of 0.25, o_2 with a probability of 0.75, so the selection also modifies $o_2.p$:

$$o_1'.p(x) = \begin{cases} 2, \; 1 \leq x \leq 1.5 \\ 0, \; \text{otherwise} \end{cases} \quad o_2'.p(x) = \begin{cases} \frac{1}{3}, \; 1 \leq x < 2 \\ \frac{2}{3}, \; 2 \leq x \leq 3 \\ 0, \; \text{otherwise} \end{cases} .$$

Equation (1) yields $\frac{11}{12}$ for the probability of o_1' being closer to 0 than o_2'.

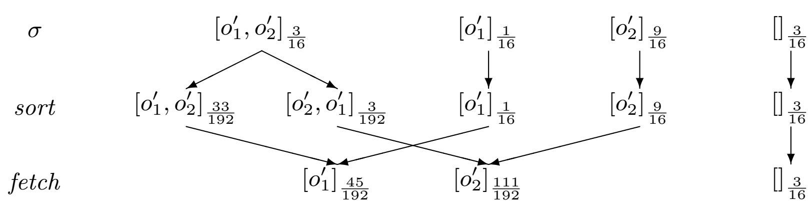

Fig. 7. Processing the query $(b(1) = 0.5, b(2) = 1)$

7 Conclusions and Future Work

In this paper we presented an abstract processing model for the quality of context data. We explained the three quality aspects uncertainty, inconsistency, and trust we use in the Nexus project and showed how the possible worlds model can be applied in this scenario.

In the future we plan to integrate more advanced models for the different quality aspects and multidimensional coordinates. The scheme for computing the quality of the overall result is also subject to ongoing research.

Acknowledgments. The work described in this paper was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center (SFB) 627.

References

1. Lange, R., Cipriani, N., Geiger, L., Großmann, M., Weinschrott, H., Brodt, A., Wieland, M., Rizou, S., Rothermel, K.: Making the world wide space happen: New challenges for the Nexus context platform. In: PerCom Workshops, IEEE Computer Society (2009) (to appear).
2. Pfoser, D., Jensen, C.S.: Capturing the uncertainty of moving-object representations. In Güting, R.H., Papadias, D., Lochovsky, F.H., eds.: SSD. Volume 1651 of Lecture Notes in Computer Science., Springer (1999) 111–132
3. Lange, R., Weinschrott, H., Geiger, L., Blessing, A., Dürr, F., Rothermel, K., Schütze, H.: On a generic inaccuracy model for position information. In: Proceedings of the First International Workshop on Quality of Context (QuaCon 2009). Lecture Notes in Computer Science, Springer (2009)
4. Cheng, R., Prabhakar, S., Kalashnikov, D.V.: Querying imprecise data in moving object environments. In Dayal, U., Ramamritham, K., Vijayaraman, T.M., eds.: ICDE, IEEE Computer Society (2003) 723–725
5. de Almeida, V.T., Güting, R.H.: Supporting uncertainty in moving objects in network databases. In Shahabi, C., Boucelma, O., eds.: GIS, ACM (2005) 31–40
6. Wolfson, O., Sistla, A.P., Chamberlain, S., Yesha, Y.: Updating and querying databases that track mobile units. Distributed and Parallel Databases **7**(3) (1999) 257–387
7. Shannon, C.E., Weaver, W.: The mathematical theory of communication. fourth edn. The University of Illinois Press (1969)
8. Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Evaluating probabilistic queries over imprecise data. In Halevy, A.Y., Ives, Z.G., Doan, A., eds.: SIGMOD Conference, ACM (2003) 551–562
9. Barbará, D., Garcia-Molina, H., Porter, D.: The management of probabilistic data. IEEE Trans. Knowl. Data Eng. **4**(5) (1992) 487–502
10. Agrawal, P., Benjelloun, O., Sarma, A.D., Hayworth, C., Nabar, S.U., Sugihara, T., Widom, J.: Trio: A system for data, uncertainty, and lineage. In Dayal, U., Whang, K.Y., Lomet, D.B., Alonso, G., Lohman, G.M., Kersten, M.L., Cha, S.K., Kim, Y.K., eds.: VLDB, ACM (2006) 1151–1154
11. Benjelloun, O., Sarma, A.D., Halevy, A.Y., Theobald, M., Widom, J.: Databases with uncertainty and lineage. VLDB J. **17**(2) (2008) 243–264
12. Cheng, R., Chen, J., Xie, X.: Cleaning uncertain data with quality guarantees. PVLDB **1**(1) (2008) 722–735
13. Yu, H., Vahdat, A.: Design and evaluation of a conit-based continuous consistency model for replicated services. ACM Trans. Comput. Syst. **20**(3) (2002) 239–282
14. Xu, B., Wolfson, O., Chamberlain, S.: Spatially distributed databases on sensors. In: GIS '00: Proceedings of the 8th ACM international symposium on Advances in geographic information systems, New York, NY, USA, ACM Press (2000) 153–160
15. Gutscher, A., Heesen, J., Siemoneit, O.: Possibilities and limitations of modeling trust and reputation. In Möller, M., Roth-Berghofer, T., Neuser, W., eds.: WSPI. Volume 332 of CEUR Workshop Proceedings., CEUR-WS.org (2008)
16. Jøsang, A.: Artificial reasoning with subjective logic. In: Proceedings of the Second Australian Workshop on Commonsense Reasoning. (1997)