Probabilistic XML in Information Integration

Ander de Keijzer

Faculty of EEMCS, University of Twente
POBox 217, 7500AE Enschede, The Netherlands
a.dekeijzer@utwente.nl

Abstract

Information integration is a difficult research problem. In an ambient environment, where devices can connect and disconnect arbitrarily, the problem only increases, because data sources may become available at any time, but can also disappear. In such an environment, information integration needs to be *unattended*, because information integration opportunities arise on ad-hoc basis.

We propose to use probabilistic XML to store the integration result and instead of resolving conflicts at integration time, just store these conflicts in the integrated information source and resolve them at query time.

1 Introduction

The problem of integration of data from heterogeneous information sources has been on the research agenda for many years. With the increasing amount of information that is available, the need for an answer to this problem increases. However, because of the huge amounts of data, manually integrating information is not a viable option. Unfortunately, due to the semantical assumptions not captured in schema or data, automatic integration solutions often make mistakes. Therefore, most of the existing integration systems are semi-automatic.

The environment we work on is an ambient intelligent environment. Many autonomous devices, with network capabilities, are equipped with applications that use a database. This database resides on the device itself, but the device should be capable of integrating and synchronizing its database with databases on other connected devices. Because all devices are autonomous and can, in principle, be mobile, connections will appear and disappear continuously.

We envision an integration system that is capable of *unattended* integration of information sources from different devices. The process needs to be unattended, because of the irregular availability of surrounding devices. It is infeasible for the user to provide information every time a device connects and integration is initiated or resumed.

The integration problem can be divided into two classes, schema integration and data integration. There already is a large body of work on schema integration, but less work on data integration. This paper addresses the data integration part of the integration problem.

Suppose, for example, two devices with address books are within connection range. The address books of these devices are depicted in Table 1. The persons mentioned in the address books have the same name, but different phone numbers. Therefore, both elements could refer to the same person. In this case, there may have been a mistake in entering one of the phone numbers, or this person may have two phone numbers. Another possibility is, that both elements refer to two different person, who have the same name.

Instead of asking the user, the system should decide itself if elements refer to the same real world object (in this case a person). In this paper we use XML as a data model. Elements referring to the same real world object are said to be *equal*; the same holds for equality of entire XML subtrees. However, in most cases this will be impossible due to the fact that not all semantics is captured in the data. Instead of resolving these conflicts, we propose to simply store the encountered uncertainty. Semantical conflicts, i.e. two elements referring to the same real world object, can be seen as uncertainty. Therefore, after integration, the information source will contain uncertain information. This uncertainty is stored as probabilistic XML.

At query time, the end user of the system is already actively involved. This provides an opportunity for the system to get feedback from the user, without really bothering the user. The feedback provided by the user can then be used to reduce the uncertainty contained

Proceedings of the VLDB2006 Ph.D. Workshop Seoul, Rep of Korea, 2006

in the information source.

Application

Municipalities in the Netherlands are obligated by law to provide their services to the public through the internet. The amount of data they have collected is huge and divided over different departments. As a result of this division, the data is also divided over different databases. However, a user accessing the services through the internet is not (always) aware of this division, therefore the data needs to be integrated. Unfortunately, the data in the different sources often conflicts on several attributes. Fully automatic integration will possibly result in errors in the data, because of the conflicts in the original sources. Manual integration is to difficult due to personal information contained in the data that needs to be checked at integration time to ensure correct integration, but even if this would be possible, it would be to labor intensive.

Probabilistic integration solves the problem of keeping wrong information and throwing away correct information, because no data will be deleted. Checking the data can be postponed to the time when the data is accessed either by the person himself through the internet, or an employee at the municipality involved in the particular case file.

2 Research Questions

For this PhD research the following questions are addressed.

- *Probabilistic model and approach*
  Conflicts between elements at integration time can be seen as uncertainty between data values, or even between entire XML subtrees. A probabilistic model based on the possible world approach will be developed to support postponing decisions about conflicts in the data. Instead of resolving these conflicts, the uncertainty about elements or subtrees will be stored.
- *Querying*
  A probabilistic XML document should be queried like a regular, non probabilistic XML document. Only the answer will be uncertain, i.e. the result will be a set of possible worlds. Therefore, the semantics of XPath will have to be redefined for querying probabilistic XML.
- *Reducing size of integration result*
  With no additional world knowledge, the size of the resulting probabilistic XML document after integration tends to be huge. We plan to use a component called "The Oracle" that *predicts* if it is imaginable that two elements refer to the same real world object. Using very simple knowledge rules, the number of possibilities in the resulting information source can be reduced enormously [KKL06], thus removing a major obstacle in the practical viability of this approach.

(a) Address book 1

name	**phone**
John	1111

(b) Address book 2

name	**phone**
John	2222

Table 1: Two example address books

- *Feedback*
  Although no human effort should be required at integration time, this is not the case for query time. When querying the document, the user is already actively using the system. Facilities for giving feedback on query results without much overhead can easily be incorporated in a user interface. Feedback can not only increase accuracy of the integrated information source, but also reduce the size of the database.
- *Usage in Integration Framework*
  The probabilistic model will have to be implemented in an integration framework capable of also supporting schema transformations. A prototype will be developed. We will show feasibility of the approach by means of some real-life experiments.

So far, we have a good understanding of the main principles and semantics of required functionality [KK04, KKA05, KKL06]. We are currently working on prototype development and performance aspects in order to prove the practical viability of the approach.

3 Probabilistic Data

In order to support *unattended* information integration, we do not make any decisions about conflicts in the data. Instead, the uncertainty is stored. For example, if the address books from Table 1 are integrated, several results are possible.

1. John_1 (value "John" from address book 1) and John_2 refer to the same real world person. One of the phone numbers is correct, either 1111 or 2222, the other one is incorrect.
2. John_1 and John_2 refer to the same real world person. Both phone numbers are correct.
3. John_1 and John_2 refer to different real world persons.

From this list, option 2 can be eliminated if we allow schema information to be used in the integration process and if only one phone number is allowed for each person. The other two options can not be resolved without using external information. This uncertainty is stored by means of probabilistic data.

We use XML to store information, because we found XML to be more expressive than the relational model [KKA05]. We introduced two new kinds of nodes 1. *probability nodes* ($\triangledown$) and 2. *possibility nodes* ($\circ$). The document root is a probability node. Child nodes of probability nodes are always possibility nodes and child nodes of possibility nodes are always regular

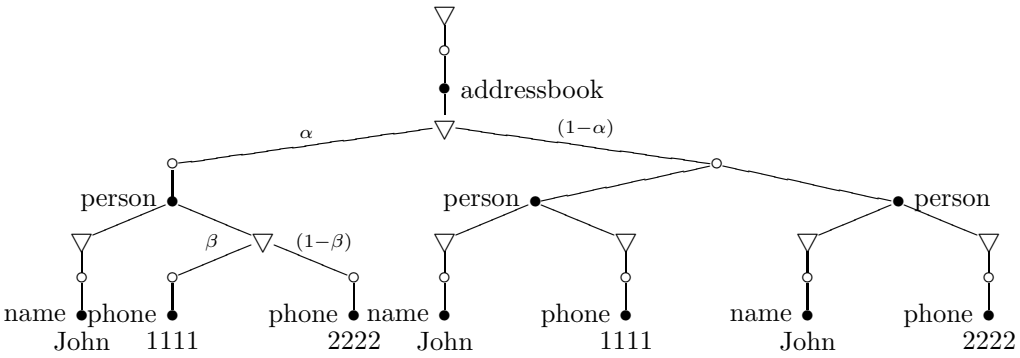

Figure 1: Integration result

XML nodes, which in turn have only probability nodes as child nodes. Therefore, on each level of an XML tree there is only one kind of nodes. Figure 1 contains the resulting XML document after integrating the address books from Table 1.

Subtrees of a probability node denote mutually exclusive possible subtrees. A probability is associated with each possibility node. Since possibility nodes with a common parent (probability) exclude each other, their probabilities should sum up to 1. The probability associated with possibility nodes indicate the level of confidence in the subtree. A probability value of 1 means that it is *certain* that the subtree will be part of the XML document. If all probabilities in the XML document are 1, the document is *certain* and can therefore be considered to be a normal XML document.

In our probabilistic XML model, we use the *possible world approach*. This means that any uncertainty in the data produces *possible worlds*, or views on the real world. Possibilities of an element can be seen as a possible representation of the real world object. If there are two elements in the information source and both have three possible representations, then there are nine possible worlds represented by the probabilistic XML tree. In Figure 1 three possible worlds can be distinguished. Two worlds where there is only one person "John", in the first world (with probability $\alpha \times \beta$) this person has phone number "1111" and in the second world (with probability $\alpha \times (1 - \beta)$) this person has phone number "2222". In the third world (with probability $(1-\alpha)$) there are two persons, both named "John", one of them has phone number "1111" and the other has phone number "2222".

The advantage of the possible world approach is that it provides for a natural way to reason about the data. First of all, the semantics of a query is simply the set of answers obtained from posing the query to each possible world individually. The probability of each possible answer is the probability of the corresponding possible world. An implementation deriving such a query result will, of course, use a more efficient approach.

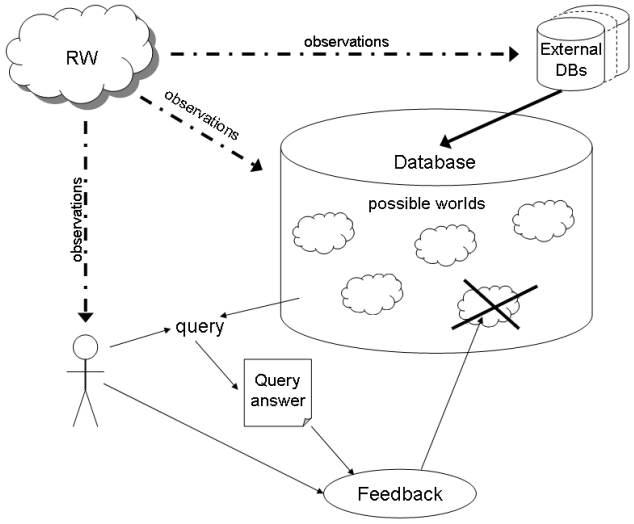

Figure 2: Information Cycle

On the whole we envision it to work like the information cycle shown in Figure 2. The database of a device is integrated or synchronized with other (external) information sources. As a result, possible worlds are created. This is caused by conflicts in the data between different information sources. At all times, the database reflects possibly conflicting observations of the real world (RW in the figure). The user of the system poses queries to the database and also *observes* the real world. He can provide feedback on query results based on his own observations of the real world. All possible worlds represented in the database that contradict the feedback can be deleted from the set of possible worlds.

4 Integration Framework

We plan to use the probabilistic model in an integration framework. An overview of this framework is shown in Figure 3. The integration framework acts as a DBMS. The user can pose queries to the

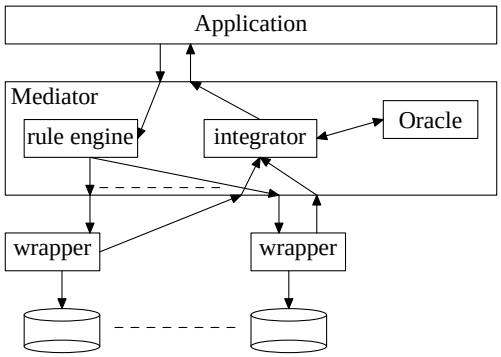

Figure 3: Integration Framework

entire DBMS, while the *rule engine* distributes the queries over autonomous underlying databases. We use a rather standard rule-based approach to deal with schema integration issues.

The schema integration problem itself will not be the focus of this PhD research, instead we will develop a probabilistic XML data model, investigate how to query this probabilistic XML and show how probabilistic XML can be used in an integration setting. This part is captured in the *data integrator* of the architecture.

The *Oracle* in the architecture is a component that can indicate if, and to what degree, two elements are referring to the same real world object. This oracle will be responsible for associating initial probabilities to uncertain elements.

5 Validation

In order to validate the research, we design and implement a prototype. First a Haskell prototype will be used to validate the research on probabilistic XML. We use Haskell, because this resembles the formal definitions of probabilistic XML and operations thereon, so we can easily check our ideas on main principles and semantics of functionality.

Next, we will design a prototype on top of MonetDB/XQuery [BGvK+06]. This prototype is mainly meant to show feasibility and performance of the algorithms. The probabilistic XML implementation will become a module in MonetDB/XQuery

6 Related Research

The amount of research on information integration is huge. [DH05] provides a nice survey. As we mentioned, we distinguish schema matching and integration from data integration. [RB01] is a good survey on schema matching techniques.

The challenges for data integration of [Lev99] have received much attention in recent years:

- Overlapping and contradictory data
- Semantic mismatches among sources
- Different naming conventions for data values

In our research, we attempt to deal with these challenges by explicitly handling the inherent uncertainties occurring in the data integration process using a probabilistic database approach. Suciu's tutorial at SIGMOD 2005 comes with an extensive bibliography on the topic of probabilistic data management [SD05]. Many results from the logic programming and artificial intelligence communities are combined in [ELLS01] which proposes a probabilistic object database. The object-oriented data model is more expressive, but also less flexible, than the XML data model. Nevertheless, many things carry over to the XML world.

A probabilistic database is not a new idea, see for example [FKL97, BGMP90, ELW01, DS96, KKA05], but in recent years attention grew considerably. Originally, work concentrated on relational databases, but in [KKA05] we argue that XML can be made to express uncertainty in a more natural way. Other probabilistic XML databases are, for example, PXML [HGS03] and ProTDB [NJ02].

Although schema and data level matching and integration can be clearly separated, schema matching techniques (see [MBR01] for a nice taxonomy) can often be used or adapted to be applicable on data level. For example, [BN05] presents a technique to search for duplicate records and to use these duplicates for schema matching. As we will see in the sequel, an important problem in data integration is how to decide whether or not two data items refer to the same real-world object. Duplicate finding techniques can be applied to (partly) solve this problem. Also in many other areas duplicate finding techniques can be found, such as data warehousing [ACG02].

A problem in using probabilistic databases for data integration is how to determine the probabilities. Many schema matching techniques suitable for data integration, however, quantify the degree of matching. For example, instance-based matchers use classification techniques [DDH01]. If two data items from different information sources referring to the same real-world object conflict on some attribute value, and one of those values is classified with less certainty that the other in the class corresponding to the attribute, then that attribute value is less likely to be correct and should receive a smaller probability. The same holds for techniques where dictionaries or thesauri are used: if a possible data value is not present in the corresponding dictionary, it should receive a smaller probability. Schema matching techniques can also be used for data conversion as [MZ98] demonstrated.

Finally, an important source of schema and data integration techniques can be drawn from the Semantic Web community. As we argued in the introduction, world knowledge is required for making decisions in the integration process. In theory, annotating the data with sufficient world knowledge may also over-

come the problem. The question remains if it is practical to demand from all information sources to be sufficiently annotated to resolve all uncertainty. Furthermore, it is an open problem how to determine beforehand when annotations suffice to resolve all uncertainty. We, therefore, approach the problem from the other end. The probabilistic data integration approach as such is independent of any world knowledge. Adding world knowledge can then be used to restrict uncertainty [KKL06].

7 Summary

The number of autonomous devices capable of communication and containing information, is still increasing. Although information integration has been an important area of research, due to the increase in available information, its importance is still growing. In this paper, we have given an overview of using probabilistic XML in data integration. We propose to use this probabilistic XML to facilitate *unattended* integration. Instead of asking the user in case of conflict, these conflicts are simply stored. At query time, when the user is already actively using the system, feedback to query results can be used to reduce the uncertainty in the database.

References

[ACG02] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *Proc. VLDB Conf., Hong Kong, China*, pages 586–597, Aug. 2002.

[BGMP90] Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. A probabilistic relational data model. In François Bancilhon, Costantino Thanos, and Dennis Tsichritzis, editors, *Advances in Database Technology - EDBT'90. International Conference on Extending Database Technology, Venice, Italy, March 26-30, 1990, Proceedings*, volume 416 of *Lecture Notes in Computer Science*, pages 60–74. Springer, 1990.

[BGvK+06] P. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner. MonetDB/XQuery - Fast and Scalable XQuery Processor Powered by a Relational Engine. In *ACM SIGMOD International Conference on Management of Data, Chicago, USA, June 26-29, 2006*, June 2006.

[BN05] A. Bilke and F. Naumann. Schema matching using duplicates. In *Proc. ICDE Conf., Tokyo, Japan*, pages 69–80, Apr. 2005.

[DDH01] A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: a machine-learning approach. In *Proc. SIGMOD Conf., Santa Barbara, USA*, pages 509–520, 2001.

[DH05] A. Doan and A. Halevy. Semantic integration research in the database community: Brief survey. *AI Magazine, Sp.Issue on Semantic Integration*, 2005.

[DS96] Debabrata Dey and Sumit Sarkar. A probabilistic relational model and algebra. *ACM Trans. Database Syst.*, 21(3):339–369, 1996.

[ELLS01] T. Eiter, J.J. Lu, T. Lukasiewicz, and V.S. Subrahmanian. Probabilistic object bases. *ACM Trans. Database Syst.*, 26(3):264–312, 2001.

[ELW01] Thomas Eiter, Thomas Lukasiewicz, and Michael Walter. A data model and algebra for probabilistic complex values. *Annals of Mathematics and Artificial Intelligence*, 33(2-4):205–252, 2001.

[FKL97] D. Florescu, D. Koller, and A. Levy. Using probabilistic information in data integration. In *Proc. VLDB Conf., Athens, Greece*, pages 216–225, Aug. 1997.

[HGS03] E. Hung, L. Getoor, and V.S. Subrahmanian. PXML: A probabilistic semistructured data model and algebra. In *Proc. ICDE Conf., Bangalore, India*, pages 467–, Mar. 2003.

[KK04] A. de Keijzer and M. van Keulen. A possible world approach to uncertain relational data. In *Proc. SIUFDB Workshop, Zaragoza, Spain*, Sept. 2004.

[KKA05] M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic xml approach to data integration. In *Proc. ICDE Conf., Tokyo, Japan*, pages 459–470, 2005.

[KKL06] A. de Keijzer, M. van Keulen, and Y. Li. Taming Data Explosion in Probabilistic Information Integration. In *Proceedings of the International Workshop on Inconsistency and Incompleteness in Databases (IIDB), March 26, 2006, Munich, Germany*, March 2006.

[Lev99] A.Y. Levy. Combining artificial intelligence and databases for data integration. In *Artificial Intelligence Today*, LNCS 1600, pages 249–268. 1999.

[MBR01] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proc. VLDB Conf., Roma, Italy*, pages 49–58, Sept. 2001.

[MZ98] T. Milo and S. Zohar. Using schema matching to simplify heterogeneous data translation. In *Proc. VLDB Conf., New York, USA*, pages 122–133, Aug. 1998.

[NJ02] A. Nierman and H.V. Jagadish. ProTDB: Probabilistic data in XML. In *Proc. VLDB Conf., Hong Kong, China*, 2002.

[RB01] E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, Dec. 2001.

[SD05] D. Suciu and N.N. Dalvi. Foundations of probabilistic answers to queries. In *Proc. SIGMOD Conf.*, page 963, 2005. Bibliographic notes to this tutorial at `http://www.cs.washington.edu/homes/suciu/tutorial-sigmod2005-bib.pdf` .