A model driven approach to enterprise data integration

Sreedhar Reddy

Tata Consultancy Services Ltd.
India

Extended Abstract

Enterprises have a lot of data locked up in their operational data stores, warehouses, documents and so on, from which it should be possible to extract a lot of useful knowledge that can help them plan their businesses better and serve their customers better. However the success of this depends entirely on the quality of data available, and unfortunately in most organizations there is very little confidence in the quality of their data assets. There are many reasons for this. A large enterprise has 1000s of data stores that are produced in different contexts under different assumptions to serve different purposes. There is uncertainty as to exactly what kind of data exists in each of these sources, what their semantics are and what kind of quality problems exist. There does not exist a single, unified, enterprise wide view of data. Traditionally the practice has been to build ad hoc, purpose specific solutions on a need to do basis, such as building a purpose specific warehouse. However, such point solutions proliferate over time, exacerbating the problem further, as they end up adding additional sources of inconsistency. We are working on a model driven approach to address some of these issues.

In a complex data environment, it is critical to model the semantic context of an entity in order to interpret and integrate it correctly with other entities; context is also critical to assess its quality accurately. Different instances of an entity may have different contexts associated with them, such as different process context, geographical context, temporal context and so on. A context has a set of associated rules that express the relationships and constraints that apply in that context. A context may also have a set of rules to assess quality attributes such as accuracy, consistency and so on.

We have developed a conceptual modelling framework using which semantic contexts can be modelled and mapped. We have also developed a reasoning framework that processes these models and mappings. Mapping composition and query rewriting are two such reasoning tasks. While choosing our modelling language, we had to balance the expressive power of the language with reasoning complexity. Keeping in mind the data sizes we have to deal with in a large enterprise, we chose our modelling, mapping and rule languages to ensure query answering is SQL reducible. However this restricts the expressive power significantly. It is possible to increase the expressive power and still keep the algorithm SQL reducible, but completeness is no longer guaranteed. Lack of completeness, however, does not mean that all query rewritings are incomplete; it only means that there is no guarantee that all query rewritings are complete. An open question then is: is it possible to identify which rewritings are complete and which rewritings potentially incomplete? For instance is it possible to profile the model fragments that contribute to a given rewriting and from there identify if the rewriting is potentially incomplete? The tool can then present such a potentially incomplete answer as its best guess and ask a user to refine it. Another limitation is, at present we support just one context model per data source, whereas in reality different instances can have different contexts even within the same data source. This would require reasoning to proceed in two distinct steps: first identifying the context of an entity and then applying the rules of the context. Doing this in a way that scales up to enterprise data sizes is a challenging problem.

International Conference on Management of Data
COMAD 2010, Nagpur, India, December 8--10, 2010
©Computer Society of India, 2010

Integrating quality aspects such as provenance and uncertainty management into this reasoning framework is another big challenge. While there has been plenty of research in areas such as data integration, knowledge representation, data provenance, uncertainty management, and so on, there is nothing yet that brings them all together into a unified framework. For instance Prof. Jennifer Widom's work on Trio integrates uncertainty management into query processing. But this assumes that uncertainty measures are recorded in a database along with data tuples. In reality, however, assessing uncertainty itself is a complex reasoning task, requiring knowledge of the context. Question then is: can this be integrated with the rest of the reasoning framework? When integrating data from multiple sources, quite often we end up with conflicting values and there is uncertainty about which value to choose. Can uncertainty management be integrated into the integration framework to resolve such conflicts? These are some of the open issues we are still exploring.