ABSTRACT

The technologies which enabled the World Wide Web (HTML, HTTP, XML) have had an equally dramatic impact within corporate IT. Similarly, the Semantic Web and its underlying technologies may have their greatest impact within the enterprise. Specifically, large enterprises struggle with business information being spread across thousands of databases each of which is semantically different. This creates an information quality problem and an environment which cannot support the flexibility required by modern business.

The paper describes how semantics may be applied to enterprise data management and presents a case study in which Unicorn Solutions and Intel® worked together to capture the formal semantics of two incompatible systems within Intel’s manufacturing planning. This allowed subtle discrepancies to be surfaced improving information quality and thus operational efficiency.

Categories and Subject Descriptors

H.1 Information systems Models and Principles
E.m Data Miscellaneous

Keywords
Semantic Web, Data Semantics, Industrial Applications, Enterprise Data Management, Semantic Information Architecture, Metadata, OWL.

1. INTRODUCTION

The technologies of the World Wide Web (HTML, HTTP, XML) also brought us the corporate intranet, electronic commerce, and the foundations for Web Services – a new strategy for integrating enterprise applications.

Similarly the greatest impact of the Semantic Web is likely to be not on the public Web – where greater semantic integration is arguably a nice-to-have – but in corporate IT – where semantic incompatibility has a cost which probably reaches trillions of dollars a year.

The typical Global 1000 commercial enterprise works with many thousands of different data schemas. Each represents critical business information with different implicit semantics. As a direct result, there exists a data integration industry worth tens of billions per annum supporting the creation and the maintenance of point-to-point translations between these schemas.

However, point-to-point data translation scripts are expensive to develop, maintain, and reuse. Most importantly, though, they are not based on a semantic understanding of the data, and ultimately result in an inflexible environment and provide low-quality of business information – a problem that costs an estimated $600 billion a year of direct losses to businesses in the US alone [1].

This paper proposes an architecture for enterprise data management known as Semantic Information Architecture, inspired by the Semantic Web. The business impact is discussed and illustrated with an industrial case study completed by Unicorn and Intel.

2. THE ENTERPRISE DATA PROBLEM

2.1 Data in the Modern IT Environment

Years ago, data sat in silos attached to specific applications. As the enterprise became networked, data became available across applications, departments, subsidiaries and enterprises. However, enterprises then found that their business information resides in thousands of incompatible data schemas and cannot be moved from place to place without a translation effort. Overall, there was no common understanding which allowed the data assets to be systematically managed, integrated or cleansed.

The incompatibility is not limited to the use of different data paradigms (such as flat file, EDI, COBOL Copybooks, hierarchical and network databases, relational databases and XML)
or to the multiple different “flavors” of each technology, such as different relational databases (Oracle, DB2, SQL Server, Sybase, etc.).

The most challenging incompatibility arises from semantic differences. Each data asset is set up with its own world-view and vocabulary. This incompatibility exists even if both assets use the same technology.

For example, one database has a table called “client,” intending this to include “channel partners”, and subdivides customers into individuals and institutions; the other data asset calls the same concept a “patron,” (although not including channel partners) and subdivides patrons into individuals, corporations, government and not-for-profit. To make matters more difficult, the “patron” data includes only domestic clients, despite the fact that this is not explicitly mentioned in any documentation and the database administrator retired five years ago. In such an environment, can the VP of Sales expect to get a timely or accurate answer to the question of how many “customers” the company has per “reseller”?

This example is multiplied by hundreds of data elements located in thousands of incompatible databases and message formats – accumulated as the enterprise merged with and acquired companies, reorganized, updated business process and migrated from one generation of technology to the next.

2.2 Commercial Impact of the Data Problem
As enterprises have become more aware of the data problem, some have started to measure its impact on their bottom line.

Firstly, the fragmented data environment inevitably leads to business information quality problems causing the business to provide its executives with ill-defined inaccurate or inconsistent information on customer relationships, inventory, internal operations or financial results.

Secondly, the data problem creates an environment which all but prevents the flexibility that is critical to a modern enterprise which must respond to a constantly changing environment. IT is unable to make changes to this fragmented data environment – with its hard-coded schemas and translation scripts – in response to business events such as mergers and acquisitions or improved business processes.

Finally, IT remains unnecessarily inefficient so long as it lacks a strategic approach to data management. In the meantime, developers deal with the frustrating and costly challenge of administering databases (some of which are redundant), mapping each database multiple times, and writing and manually maintaining translation scripts.

If data is to have long-term strategic value to the enterprise, enterprises must explicitly catalog their data assets and must elevate data into information by explicitly capturing the meaning and context of the data. This process of data semantics is a major step on the way to a Semantic Web for the enterprise.

3. SEMANTIC INFORMATION ARCHITECTURE
The proposed Semantic Information Architecture [2] combines best practices from the Semantic Web community and from the data management and metadata professions (e.g. [3]) who are tasked with bringing overall order to the data environment in the enterprise.

It has three main elements.
- Metadata
- Ontology
- Data Semantics

3.1 Metadata
Before data assets can be understood, they must be cataloged. Metadata – or data about the data – should include the asset’s schema as well as information about an asset’s location, usage, origin, relationship to other assets, rules associated with it, and assignment of ownership and responsibility. Some of this metadata may be scanned automatically from assets such as relational databases or from existing sources of metadata. Metadata Repository products are available for this purpose [4].

3.2 Ontology
The core of the Semantic Information Architecture is an ontology. The ontology should capture the desired business world-view. In an enterprise where there are thousands of physical ways of representing information, the ontology should capture the preferred world-view used by the business people. This can then act as a neutral model to which each physical data asset is mapped.

It is hoped that the OWL standard [5] will provide a framework for authoring such enterprise ontologies. There is also a strong need for standard ontologies for each vertical industry which will give companies an important short cut to creating their own enterprise ontology.

Many modern enterprises have data models and some have an official enterprise data model. Although data models capture less business depth than an ontology, these should not be dismissed but instead they may be leveraged and extended into an enterprise ontology.

3.3 Data Semantics
Semantics captures the formal meaning of data. It is achieved by mapping (or rationalizing) a data asset’s schema to the ontology.
In principle, any database or message format with a schema can be mapped, including relational databases, XML, older hierarchical and network databases and COBOL Copybooks. Data that is structured without a schema (e.g. EDI messages and flat files) can be parsed into XML documents with an appropriate schema and then mapped.

Software can aid the mapping process using type information, foreign keys, and even name similarities to suggest matches and by providing an efficient graphical mapping environment. However, mapping will never be totally automatic; only a database administrator or other expert will know how to interpret data accurately.

Semantic mapping can create immediate savings. Having mapped an asset once to an Ontology, its relationship to all other assets may be inferred. Every asset is therefore mapped only once, in contrast to the current situation in which every data asset is mapped many times, often using inappropriate and non-scaleable tools such as MS Word or Excel.

This cost and time savings is easy to quantify – mapping a hundred data formats in all combinations of point-to-point mapping requires nearly five thousand mappings; mapping them to a central Ontology (and then inferring the five thousand point-to-point relationships automatically) takes merely one hundred computer-aided mappings. This savings is compounded by the ability to maintain and update just one hundred transformations based on a one set of rules captured in the ontology.

Data Semantics is currently the subject of increasing interest amongst forward looking data practitioners (see [6]).

4. CASE STUDY: THE PROBLEM

As a leader in semiconductor manufacturing and technology, Intel has a complex factory network including 12 fabrication facilities and 12 assembly and test facilities worldwide. Intel has declared a mission to be a worldwide, 100 percent e-Corporation. As such, more than 60 percent of Intel's materials transactions and 85 percent of customer orders are processed electronically.

Intel has a clear business objective of a timely and accurate production flow throughout the manufacturing network. However the multiple applications which support the planning function have deep semantic differences – for example the work week is cut off at different times in different geographies and systems and materials are categorized using different taxonomies and product codes. This means that discrepancies between the systems are not immediately transparent and unchecked could cause costly missed customer shipments or conversely inventory excesses.

5. APPROACH

The two-week case study was structured as follows:

- Scoping the business domain and business problem with the Company
- Capturing the desired business view in an ontology
- Mapping two underlying relational data sources to the ontology
- Using these semantic mappings to automatically infer transformations in ANSI SQL to compare between the incompatible databases
- Executing the SQL query to compare between the semantically-different sources and produce an accurate list of discrepancies.

5.1 The Ontology

The ontology was authored using the Unicorn system v2.0 using an ontology formalism that was based as closely as possible on the expected structures of OWL – the standard for ontologies being developed by the W3C. The ontology was authored based on interviews with Intel personnel and included concepts related to the work week, manufacturing facilities and the flow of goods between facilities.

The ontology further included conversion rules – the ability to capture expressive arithmetical relationships between different properties – a type of rule which is not currently supported by the OWL draft – but which was considered critical in deriving data transformations.

5.2 Semantic Mapping

The Unicorn system was also used to support the mapping of each relational database schema to the ontology according to the following rules

- Relational table $\rightarrow$ ontology class
- Column in relational table $\rightarrow$ property (or property path) of class to which relational table was mapped
- Foreign key in relational table $\rightarrow$ property (or property path) of class to which relational table was mapped – with property having a target class which is mapped to the table to which the foreign key points.

Despite this elegant set of rules, many real-world relational databases are poorly designed and have data relating to two distinct classes mixed in the same table or more than one property value concatenated in the same column. Therefore conditional mapping was also used.
5.3 Transformation generation
Given two data sources mapped to the same ontology, it is possible to infer a data transformation between them. For every element in the target, the algorithm looks for an element in the source which is mapped to the same ontological concept as the target. Failing this, the algorithm looks for an element in the source which is mapped to a different ontological concept related thereto by a conversion rule.

The resulting transformation was converted into an ANSI SQL query so that it could be executed directly on the data sources using a standard relational database management system (RDBMS).

6. RESULTS
At the end of the two week project, the SQL queries were run and correctly identified all discrepancies between the two systems – discrepancies which were otherwise obscured by deeply different semantics.

While Intel’s systems could have been compared in the traditional way using point-to-point hand coded transformations, this solution would not have provided insight into the underlying business issues. It also would be awkward to maintain and would not scale to a broader data quality initiative in manufacturing. The following chart summarizes the benefit of using a semantic approach via a central ontology to integrate data sources, over the traditional point-to-point approach.

Table 1. Benefits of a semantic vs. traditional approach

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Traditional Point-to-Point Integration</th>
<th>Via Central Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>$N^2$ mappings</td>
<td>$N$ mappings</td>
</tr>
<tr>
<td>Code Reusability</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Code Maintenance</td>
<td>Hard (must understand each SQL query)</td>
<td>Easy (must understand the Ontology)</td>
</tr>
<tr>
<td>SQL Code Quality</td>
<td>Depends on developers</td>
<td>Good model = high quality</td>
</tr>
</tbody>
</table>

7. CONCLUSIONS & FUTURE RESEARCH
Semantic incompatibility is one of the most fundamental and costly problems in enterprise information technology. The Semantic Web has the potential to address this problem by allowing multiple physical data schemas to be mapped to one ontology which reflects the desired business world-view. These mappings can then be used to infer translation scripts between the data assets and to manage and cleanse the data environment.

In addition to the existing work on the OWL standard, widespread commercial applications of the Semantic Web are critically dependent on standards and technology for mapping commercial data storage and data message formats (COBOL, hierarchical, relational, XML etc.) to an OWL ontology. The use case above demonstrated the value of such mapping with the example of relational databases.

If this obstacle can be overcome – bringing physical corporate data into the realm of semantics – the Semantic Web can then be applied fully to enterprise IT – bringing in a new era in which enterprise data can be found and understood dynamically without the need for prior knowledge of its specific schema.

8. ACKNOWLEDGMENTS
The case study was executed by Idan Zu-Aretz at Intel together with Itamar Gus and Eliezer Israel under the supervision of Victor Ofstein and Rannen Meir at Unicorn and with the support of many colleagues. The two companies were introduced by Uri Arazi and Dany Star at Intel Capital. This paper relies on the earlier write up by Joram Borenstein [7].

9. REFERENCES