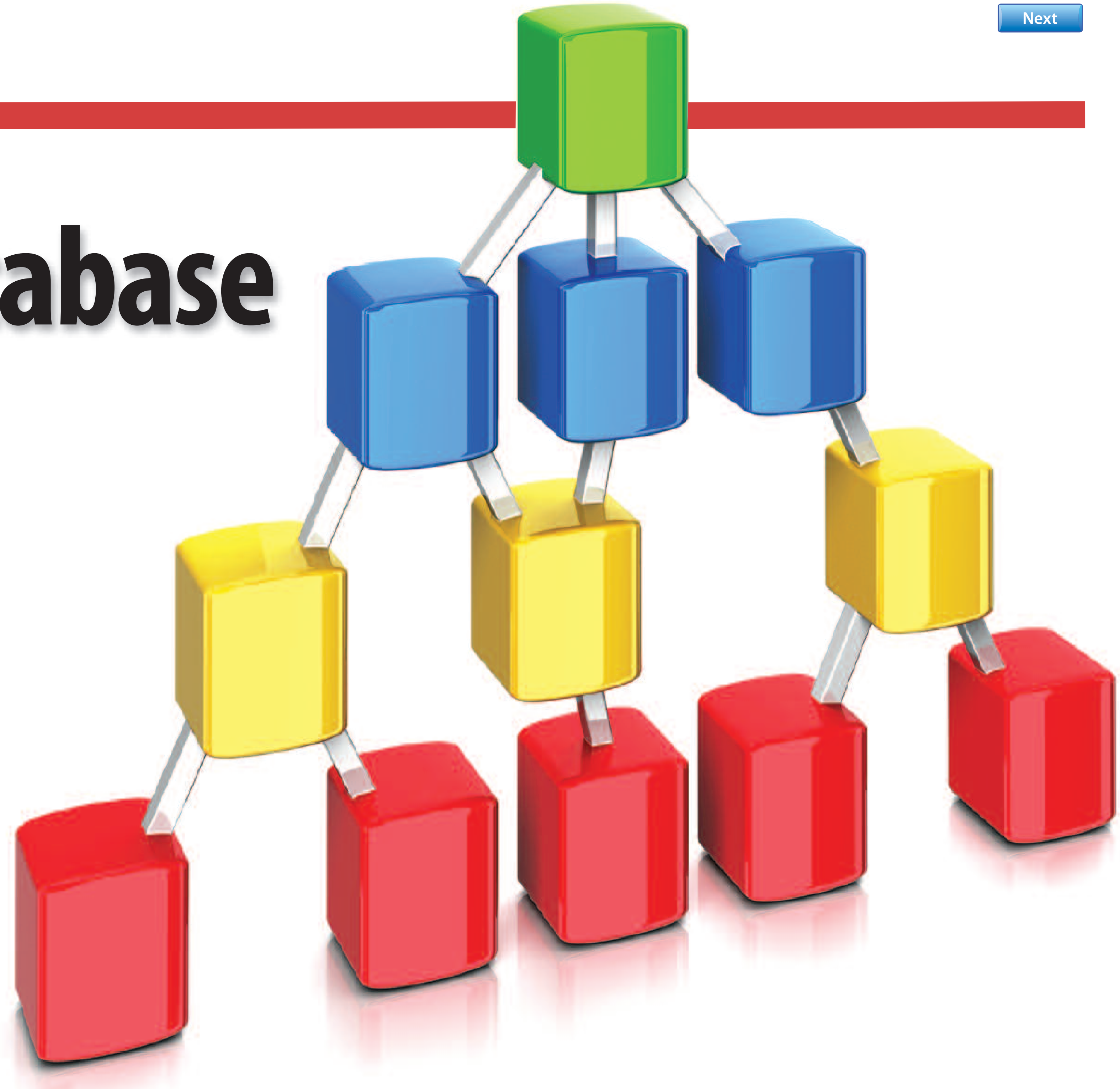
InformationWeek :: reports
reports.informationweek.com
December 2012 $99

Next

Semantic Database Technology

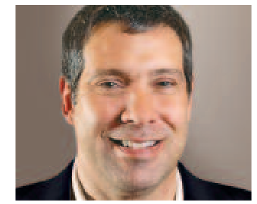


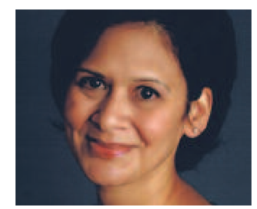
What do concepts like the semantic Web, mashups, Web 3.0 and open data have to do with your enterprise relational databases? Much like consumerization continues to drive innovation in end user computing, these technologies deliver benefits that your business should be exploiting.

By David S. Read

Report ID: R6171212

TABLE OF CONTENTS





ABOUT US

InformationWeek Reports' analysts arm business technology decision-makers with real-world perspective based on qualitative and quantitative research, business and technology assessment and planning tools, and adoption best practices gleaned from experience. To contact us, write to managing director **Art Wittmann** at *art.wittmann@ubm.com,* content director **Lorna Garey** at *lorna.garey@ubm.com* and research managing editor **Heather Vallis** at *heather.vallis@ubm.com.* Find all of our reports at *reports.informationweek.com.*

David S. Read
InformationWeek Reports

David S. Read is CTO of Blue Slate Solutions and leads the company's technology consulting team, helping determine functional requirements, platforms, products, integration approaches, architectures, QA procedures and implementation processes, primarily within healthcare and financial services. Blue Slate focuses on effective integration and construction of secure and reliable service-oriented systems, including appropriately leveraging each client's existing systems and architecture investments.

Dave provides leadership surrounding data security, including encryption, protocols, policies and processes for multiple healthcare and insurance clients. He is also involved in speaking engagements and education at forums such as the PCIAA, an IT conference for property and casualty insurers, and the Business Rules Forum. His expertise extends to significant data mining experience, and he is working on projects to leverage semantic Web concepts as a path to improve data reusability and integration approaches. Prior to joining Blue Slate, Dave had 15 years of IT leadership experience, including various positions within GE, where he was a lead architect for many diverse Web-based applications focusing on a scalable and standards-based infrastructure. The success of these applications led to his receiving GE's Internet Pioneer of the Year award. He holds a Bachelor of Science in computer information systems from the College of Saint Rose.

Michael Delaney, Blue Slate senior consulting software engineer, contributed research and interviews for this report.

EXECUTIVE SUMMARY

Semantic technology has created a disruptive opportunity for businesses to obtain more value from their data. The concepts surrounding the semantic Web, such as linked data cloud and data mashups, are powered by a set of emerging standards and products that, for now, are mainly used for consumer services. However, these technologies are equally compelling as part of an enterprise data platform behind the firewall.

At a high level, there are five main benefits of semantic technology:

> It works in tandem with your existing database investments;
> It aligns with Web technologies;
> It speeds the integration of multiple databases;
> It's based on data structures that are flexible by design; and
> It can help enterprises tackle big data challenges.

In this report, we'll introduce the key data-centric semantic technologies, compare semantic and relational data management, outline several situations where this technology should be considered, discuss products in the semantic space and highlight companies already finding value through the use of semantic technology.

Introduction To Enterprise Data-Relevant Semantic Technology

Let's first state the mantra for semantic technology: Information must be machine readable. The data definition must provide meaning and context so that a computer can understand information, rather than requiring a human to interpret it. This goes beyond simply supplying a structure for the data, such as an XML schema; it requires that concepts within the data be given an ontological basis.

In other words, the data attributes have to be given definitions that humans and machines can interpret.

The semantic technology standard for describing the meaning of data is the Web Ontology Language, or OWL (yes, the "O" and "W" are reversed). OWL defines a rich set of data relationship descriptors that are used to create a structured set of definitions for business terms, data sets and attributes. The resulting set of definitions is called an ontology. OWL is a World Wide Web Consortium (W3C) recommendation, meaning that it is a published standard and available for vendors to use.

OWL is built on top of another W3C recommendation, Resource Description Framework Schema. RDFS defines basic data structural concepts allowing for grouping of data (classes) and typing of data elements (attributes). Simplistically stated, RDFS provides structure for data, while OWL adds the ability to supply detailed meaning within those structures.

Typically, a business will create and use multiple ontologies. At a corporate level, ontologies may define operations, depart-

Figure 1

Relational vs. Semantic At a Glance

	Relational Technology	Sematic Technology
Maturity	> Mature, with over 40 years of standards, product development and enhancements, and field use. > Wealth of supporting tools for database design, visualization, integration and reporting.	> Very young set of standards a first-generation product set. > Narrow ecosystem of tools. Very limited options for plug-and-play reporting.
Performance	> Table and index designs specific to type of use, such as OLTP, ad-hoc reporting and analysis. > ETL tools required to copy and restructure data into marts and warehouses.	> Data storage is standardized, streamlining performance for multiple uses. > Transformations and federation are core to the technology and defined within the data environment.
Maintenance	> New data integrations become ever more challenging with each new data set. > Challenges arise with sparseness, null values, keys and normalization.	> Data storage is largely separated from data integration, simplifying the process of adding new data structures. > Data structure anticipates missing values and requirement to add new relationships.

Data: *InformationWeek* Reports R6171212/1

ments, metrics and other core business concepts. At a system level, ontologies describe the types of data being managed and the specific attributes that make up that data, as well as the structure and relationships inherent in the attributes. Although separate, these ontologies are related so that system-level details tie back to corporate-level concepts.

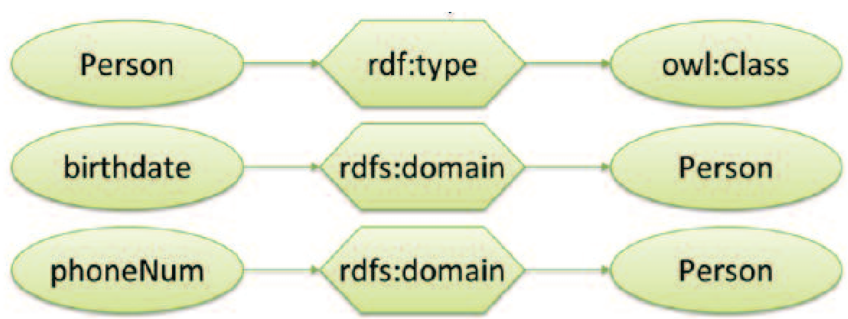
From a system perspective, the ontology is a logical representation of the data's meaning and relationships. In the same way that a data analyst would create logical models for data, the ontologist produces an ontology for the data. The ontology provides the definition of data and its relationships, but not the layout for the physical storage of data.

Figure 2 presents a graphical representation of a simple ontology defining the concept of a person and defining a person as having attributes of "birthdate" and "phoneNum." For context, a relational model depicting a similar structure is depicted in Figure 3, which shows a person table having columns of "birthdate" and "phoneNum." Note that the ontology (Figure 2) doesn't generate a table or define a storage mechanism; it simply communicates the relationships that may be found in the data. In the relational table, the columns in the table do dictate the data storage. That's a key difference.

Figure 2

Simple Ontology

Graphical depiction of an ontology defining a person class with two attributes.

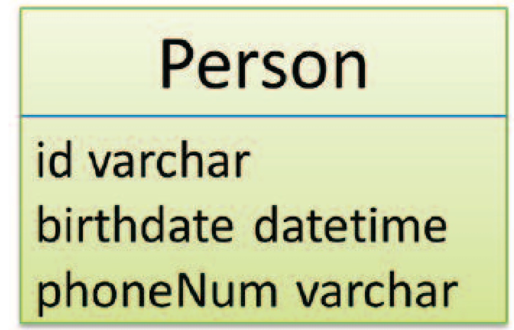


Data: *InformationWeek* Reports R6171212/2

Figure 3

Relational Definition

In a relational database, a person entity has columns.



Data: *InformationWeek* Reports R6171212/3

Core to the physical representation of data within a semantic technology stack is a concept known as "the triple." The triple is the fundamental storage structure, providing the ability to associate any piece of information with any other piece of information. It is called a triple because the principal components for representing a data relationship are the subject, the predicate and the object. These three concepts — subject, predicate, object — comprise the triple. Note, however, that there is no requirement that the data actually be stored in a triple-based structure. Data from other sources, such as relational databases, key-value data stores and spreadsheets, can be used as sources for semantic data. In fact, it is a core feature of semantic technology that you do not have to rehost data if the structure of existing databases already serves the business well.

The triple is formally defined as the Resource Description Framework. Conceptually, RDF uses uniform resource identifiers (URI) to create links within data that computer applications can use. Uniform resource locators, which we know commonly as Web addresses, such as http://google.com, are a type of URI. RDF is also a W3C recommendation and serves as an underpinning of RDFS.

What is important to note regarding the triple is that the relationships are directed. That is, the subject points to the predicate, which points to the object. There is no assumed relationship back from the object to the predicate or from the predicate back to the subject. Once a set of triples has been defined, it results in a data structure known as a directed graph.

Figure 4 depicts a set of data triples, also known as individuals, which conforms to the person definition shown in Figure 2. For comparison, a relational table is shown in Figure 5 containing the same data and matching the person table described in Figure 3.

A third W3C recommendation that is core to the use of semantic technology is SPARQL. The consensus regarding the meaning of the acronym is that it is recursive and stands for SPARQL Protocol and RDF Query Language.

SPARQL serves some of the same purposes when querying semantic data stores as SQL does with relational data sources. The key similarity is that SPARQL is used to access subsets of data, and with recent extensions, to update data as well. The major difference between them is that SPARQL is not used to create data structures in the way that SQL DDL would be used. Put another way, SPARQL is not used to create ontologies.

This code string shows a simplified SPARQL Query (namespaces and prefixes are omitted for readability) that would retrieve all of the person entries along with their birth dates and phone numbers. The SQL equivalent for working with the person table is also shown.

```
SELECT ?id ?bdate ?pnum
WHERE {
    ?id rdf:type Person ;
            birthdate ?bdate ;
            phoneNum ?pnum .
```

In SQL, this same query to select all person rows would look like this:

```
SELECT id, birthdate, phoneNum
FROM PERSON
```

Beyond these semantic platform standards there are two technologies to consider when evaluating semantic technology. These are triple stores and semantic reasoners. Each represents implementations of the standards described above.

A triple store is a data management platform that manages triples, the RDF-based graphs described earlier. These platforms are optimized to support management and querying for large numbers of triples. For example, in 2011, Franz's AllegroGraph database, in collaboration with Intel, successfully stored 1 trillion RDF triples. Several vendors offer very robust triple store technology, and many that leverage existing, typically relational, data stores also allow them to be used as triple stores. Each has its place, and we will describe those in detail in the product section of this report.

A semantic reasoner provides a logic-processing engine within the semantic technology realm. This component does not have a direct parallel in other data structure and storage technologies. The reasoner acts as a forward-chaining engine and as such is more related to a business rule engine than a database management system. As with the triple stores, we will describe the reasoner's operation as we delve into products.

From a standards perspective, the triple store is concerned with RDF and SPARQL. The reasoner is most concerned with RDFS and OWL. However, as we'll see when discussing products, the line between triple stores and reasoners can be blurry.

Semantic And Relational Data Management

Most companies have made significant investments in relational database (RDB) technologies, including database management software; extract, transform and load (ETL) tools; data marts; data warehouses; and a variety of reporting systems. The question, however, is whether there's enough value offered by semantic technology to justify the investment in infrastructure, training and refactoring that a move might entail.

To understand why semantic technology should be on your radar, let's contrast the feature sets of semantic and relational database technology and consider some situations where a semantic data store may provide significant advantages over a relational database.

Technologically speaking, a relational database and the operations it supports are governed by the mathematics of relational algebra and largely deal with representing information within matrices (tables with columns and rows). From an implementation perspective, the data representation uses keys to create relationships between data.

Keys give us a lot of flexibility. Representing one-to-one, one-to-many and many-to-many relationships is easily done with primary keys and foreign keys. A one-to-one relationship can be represented by placing both pieces of data in a single table. For one-to-many relationships we can use two tables, and for many-to-many we would likely use three tables. Note that the relationship type (one-to-one and so forth) impacts the physical design of the relational database.

These common data relationships do not pose a challenge for a relational database to represent. However, changing those relationships can become more of an issue. This agility with respect to the physical data model is a key difference between semantic and relational data stores and is driven by the way the technologies relate the data as well as the relationships within the data.

The relational database's use of keys requires us to restructure data if relationships change. That is, the physical structure of the database is used to represent the relationships in the data. In contrast, the semantic technology paradigm places most of the data relationship information in the ontology (a logical construct), wholly separate from the physical storage of the data; that means that refactoring of physical structures is much less frequent in semantic data stores even as the underlying relationships change. Mathematically speaking, semantic data storage, a.k.a. the triple store, represents data as a directed graph, which has far more flexibility than a matrix representation.

As described earlier, in a semantic data store, the core structure used for data representation is the triple. The structure used to represent the data relationships is the ontology. This means that whether we have a one-to-one, one-to-many or many-to-many relationship, it will still be expressed with triples (data). If we have a one-to-one relationship today and tomorrow need to represent the relationship as one-to-many, we do not need to restructure the data. Instead, we alter the ontology (relationship) and leave the data intact (see story, p. 18, for an example).

Applied to the concepts of data marts and warehouses, this provides a path to creating multiple views of data without rehosting the data and creating multiple physical models for the data. Although this is a powerful concept, the immaturity of semantically based reporting tools means this approach still comes with challenges.

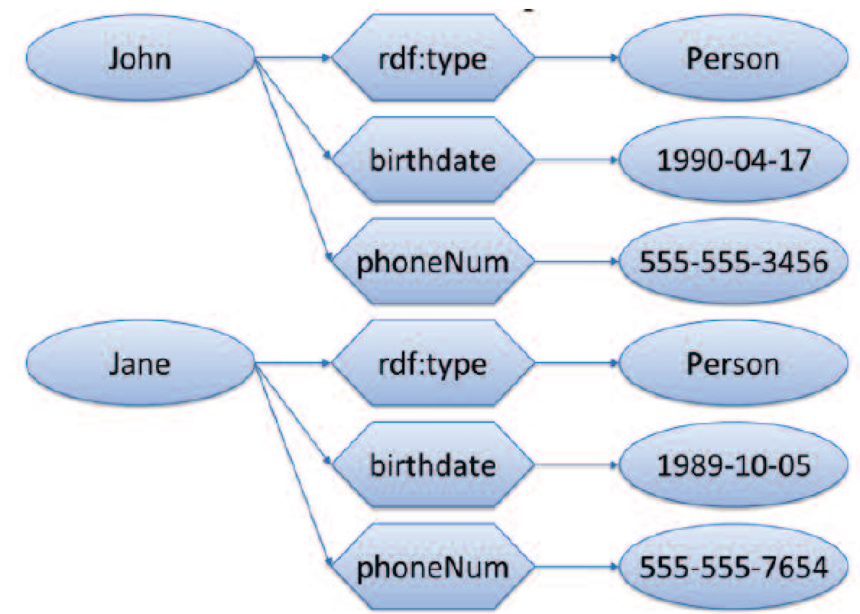
Also, the logical and physical separation of semantic data isn't a perfect solution, and if the data being stored needs to be represented with a significantly different set of relationships, it may become necessary to transform the data. However, even this transformation is often easier to do with semantic technology than with RDB-based systems through the use of the semantic reasoner.

The semantic reasoner provides another interesting comparison between semantic data stores and RDB environments. As we said earlier, there is no parallel RDB technology that mimics the reasoner. In fact, the reasoner's role is to provide the link between the ontology (data relationship definitions) and the data itself (the triples). The reasoner uses the ontology to apply the relationships to the triples. This reasoned information may be

Figure 4

Graphical Depiction of Triples

This set of Individuals conforms to the Person ontology



Data: *InformationWeek* Reports R6171212/7

used in real time, or it may be persisted and stored so that it is available without running the reasoner multiple times.

Federation of data also differs between semantic and relational technology. Often when businesses need to combine RDB data from multiple systems, they build data marts and data warehouses. The purpose of these environments is to unify the view of the data from these systems. ETL processes are often used to convert data represented in different ways into a standard representation used in the mart or warehouse.

Data federation is a core objective of semantic technology. Ontologies, through the use of a semantic reasoner, can provide on-the-fly ETL-type functionality without using a separate tool or requiring the creation of new data structures to house the mapped information. The federation of data from multiple semantic sources through the use of mapping (or chaining) ontologies is a common use case when creating mashups of semantic data on the Web.

Finally, the need to support big data continues to be a focus for business. On this front, both technologies can scale to support very high processing volume and very large storage needs. Semantic and RDB vendors routinely release metrics regarding big data that show the viability of these platforms. Both also work well in lower-volume situations and do not add significant overhead, regardless of the underlying data volume dynamics.

Figure 5

Relational Table

In a relational setup, the Person data is set up as shown.

Table: Person		
id	birthdate	phoneNum
John	1990-04-17	555-555-3456
Jane	1989-10-05	555-555-7654

Data: *InformationWeek* Reports R6171212/8

In light of these differences, it's clear that semantics immediately reduces risk for applications whose data structures need to change or be extended on a regular basis. Semantic technology also simplifies the federation of data across heterogeneous data sources, the types of situations where ETL tools are often called on to create data marts. Finally, the rigor of semantics leads to the creation of corporate and system ontologies that provide clarity for all aspects of application implementation, from requirements through design and on to build and test. A formal ontology language, such as OWL, benefits the company as a whole by providing a common understanding of the business and its operations.

3 Signals That Semantic Technology May Be For You

> **Heterogeneous data integration:** Semantic technology's focus on federation and interoperability makes it ideal for processing heterogeneous data. "Data integration projects are extremely painful, extremely long and often fail at great cost because they are too difficult to do," says Matthew Petrillo, president of Ontotext, the creator of sematic repos-

itory OWLIM. "Semantic is better at getting your various data stores to work together." The risk of failure is reduced by the flexibility of the environment, all the better to deal with disparate data. Petrillo goes on to describe semantic data stores as being a "schema last" paradigm, where data can be placed in the database (triple store) without knowing the structure of the data up front. You can then apply one or more schemas (ontologies) to organize the data afterward. The ontology becomes a key component of the integration, supplying a consistent vocabulary, as opposed to an entire structure.

> **Evolving data structures:** Not all businesses understand the complete picture of their operational data. As businesses demand more agility to support changing environments, markets and opportunities, they need the ability to quickly process new types of information. Semantic technology works very well under these conditions.

As a platform for working with Web-based data, the semantic technology standards were built to assume that new data would be discovered and need to be combined with existing data. The linked data cloud, connecting data from many different publicly available data sources, is an example of this concept taken to an extreme.

Within a business domain, the integration of new data is not likely to be as broad as is seen with Web-based mashups, but the concepts are the same. The fact that the technology is proven under less controlled environments bodes well for using it within the more controlled data environment of an enterprise.

> **Rule-based data interactions:** Some applications involve significant logic to filter applicable data based on other data values and relationships. For instance, a claims processing system may offer a variety of review outcome choices to a user based on the procedure codes on the claim. The review outcome choices are data, so the rules in this case are associating one set of data (procedure codes) with another set of data (review outcome choices). Rules of this nature are often coded directly in the application code or placed in a business rules engine. However, it's much more effective and maintainable to place them in the ontology, where they are used by the reasoner.

In the claim processing example, it would be very easy to understand the data relationships encoded in the ontology since the context is clear. If these data-behavior rules are placed elsewhere, they are disconnected from the data, which leads to missed, erroneous or extraneous rules. It is also much more difficult to test such rules since they are essentially housed in two places: The data and definitions are within the database, while the rules interpreting the data are housed in source code or a business rule environment.

Vendors And Products

There are a variety of commercial and open source options for working with semantic data. The key product types we'll discuss focus on ontology editing, reasoning, and data storage and retrieval. There are different approaches in each area, which we'll summarize as well.

> **Ontology editing:** The ontology serves as

the definition of data relationships. The ability to graphically represent relationships is helpful when designing an ontology. Further enhancing the value of the editor is integration with a reasoner so that inferred relationships can be reviewed without leaving the editor environment. One commercial option is Knoodl from Revelytix. Knoodl is a Web-based editor that can be provided as a hosted service or installed locally. The tool provides a graphical environment for viewing and modifying ontologies. It can also be used to create sample data, which can be used to test the data relationships being expressed in OWL. The defined ontology is easily exported for use by other semantic environments.

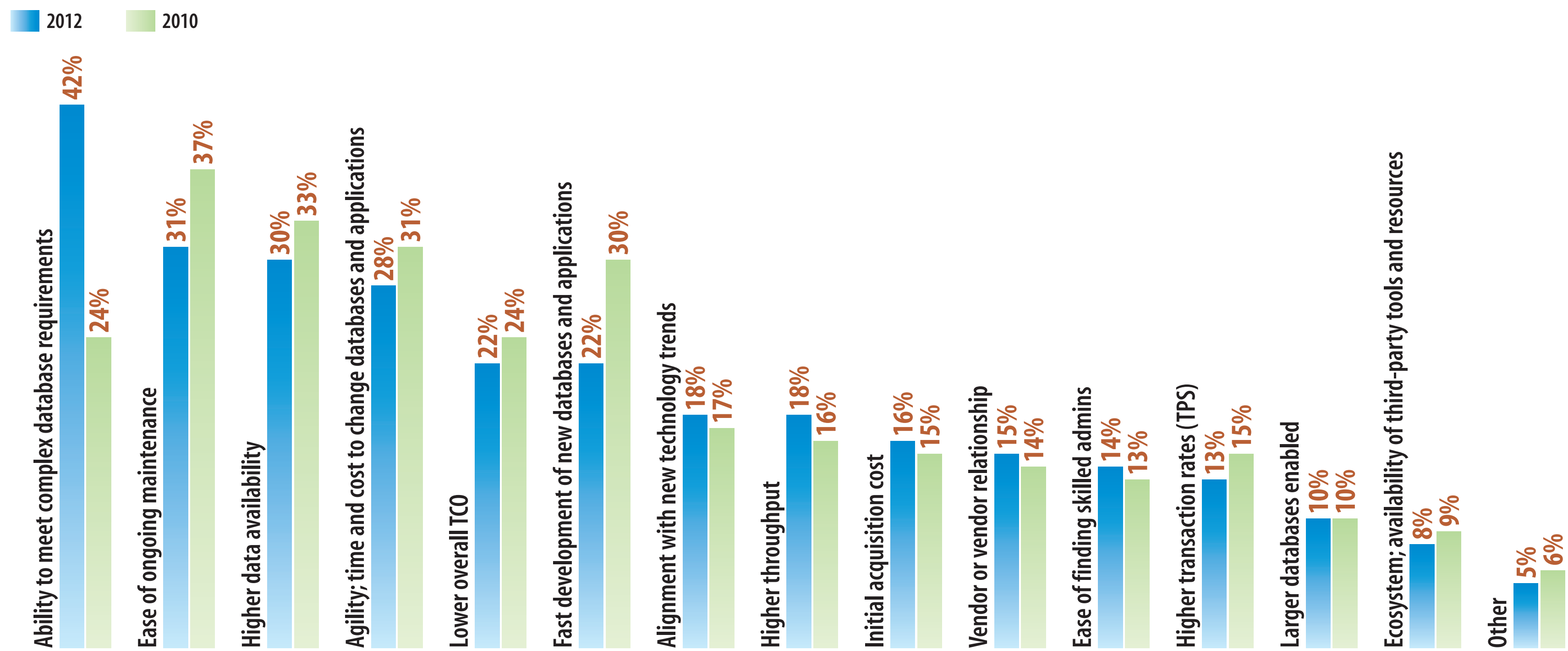
A well-known commercial ontology editor is TopQuadrant's TopBraid Composer. This is a client-side application that contains a rich feature set for working with ontologies both through direct entry of OWL syntax and through a graphical editor. The environment has many powerful features, such as reasoner integration, sample data creation and SPARQL query execution. It is a one-stop shop for building and testing an ontology.

The best-known open source ontology editor is Protégé. This editor was created at Stanford University and is supported by a variety of academic, government and corporate entities. It provides a good editing plat-

Figure 6

Factors Influencing Choice of Operational Database

What are the top factors that influence your choice of operational database?



Note: Three responses allowed
Base: 760 respondents in November 2011 and 755 in August 2010
Data: *InformationWeek* State of Database Technology Survey of business technology professionals

R3950112/9

FAST FACT

31%

of respondents to our 2012 State of Database Technology Survey say ease of ongoing maintenance is a top factor influencing their choice of operational database.

form, including integration with visualization libraries to allow for graphical review of an ontology.

The resulting ontology from any of these editors is then used by a reasoner, discussed below. The serialization of the ontology can be done in several ways and are equivalent. Common ways to represent ontologies are RDF-XML, N3 and Turtle. These are simply formats for storing ontologies and do not change the actual definition of the ontology. Most editors and reasoners can process any of these formats.

> **Triple stores:** Storage of semantic data can be handled in several ways. There are pure-play triple stores, products that map between relational databases and ontologies, and frameworks that create triple stores using denormalized relational tables.

In the realm of pure-play triple stores, three well-known options are Oracle's Spatial and Graph, Ontotext's OWLIM and Franz's AllegroGraph. The features of these environments are similar in that all offer an optimized environment for natively storing triples. Most triple stores either provide built-in reasoning or integrate with a third-party reasoner. Reasoning level and speed are areas that a business should investigate when evaluating triple stores. We discuss reasoners below.

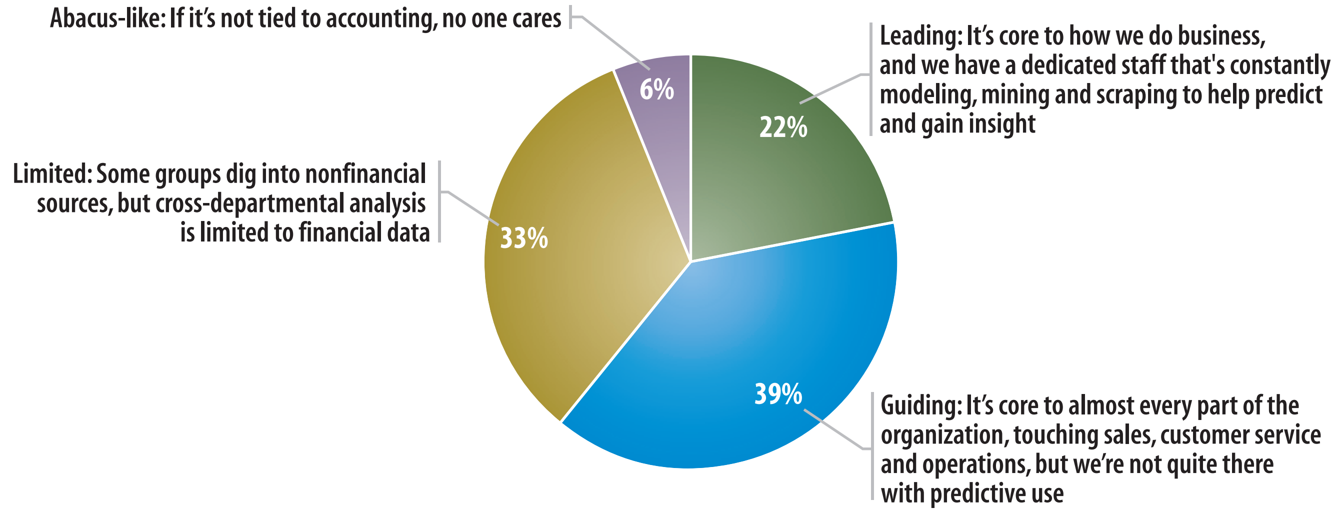
When evaluating triple stores, you are likely to encounter vendors discussing the fact that the triples in their products are actually represented using quads or quints. The reason for adding data to the triple is that often there is relevant metadata, which is useful to associate with each triple. Common uses for such metadata is provenance, used to understand how trusted the information should be, and keys, used to speed query performance.

Mapping between relational databases and

Figure 7

Approach to Data Analysis

How would you classify your organization's approach to data analysis?



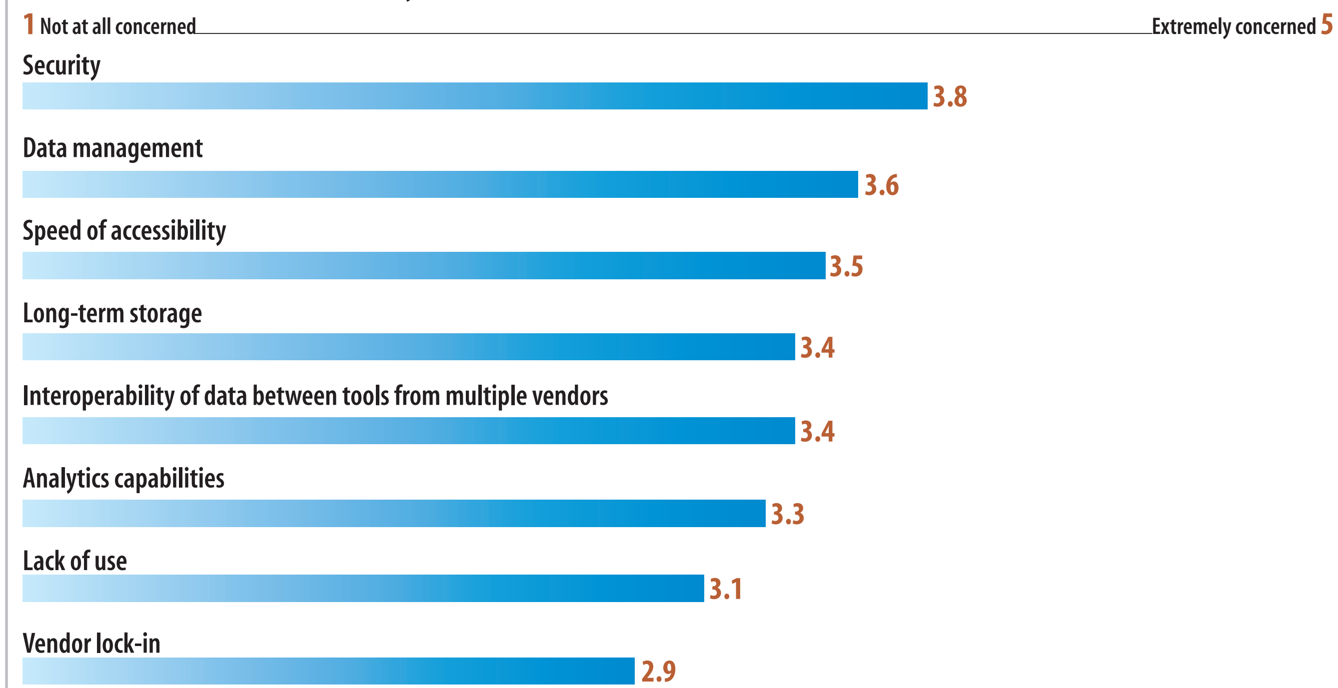
R5561012/3

Data: *InformationWeek* 2013 Big Data Survey of 257 business technology professionals at organizations with 50 or more employees, September 2012

Figure 8

Concern With Big Data-Related Issues

How concerned are you with the following issues as they relate to big data? Please use a scale of 1 to 5, where 1 is "not at all concerned" and 5 is "extremely concerned."



Note: Mean average ratings R5561012/12
Data: *InformationWeek* 2013 Big Data Survey of 257 business technology professionals at organizations with 50 or more employees, September 2012

semantic environments has the advantage of keeping a company's database assets intact while allowing them to be integrated with semantic technology. This capability is exemplified by Revelytix's Spyder and OpenLink's Virtuoso Server products.

Spyder sits on top of a relational database and exposes the data via a SPARQL endpoint, allowing it to be used with semantic technology and, more importantly, allowing it to be federated with other semantic data sources. The approach does not rehost the relational data, thus providing an effective way to continue leveraging existing RDB investments while exploiting semantic technology where it is most beneficial to the business.

Virtuoso Server will expose a relational database as a semantic data store, offering the ability to leverage an existing database infrastructure as described above, and will integrate directly with pure-play triple stores.

The open source frameworks Sesame and Jena provide basic RDF data storage using file systems, relational databases and even other triple store products. To represent triples, they create a proprietary denormalized database schema, which they use to store and retrieve the triples. Beyond providing the functionality of a triple store, they offer Java APIs that can be used to integrate Java-based applications with semantic data. These tools are effective for proof-of-concept and lightweight semantic technology implementations.

> **Reasoners:** A reasoner is a rule engine that is built to understand the constructs of RDFS and OWL. Different reasoners support different levels of OWL. The reason for not supporting the entire OWL standard comes down to a feature versus performance decision.

Many useful features of OWL are commonly used and do not create significant runtime performance risks when reasoning over a large set of triples. Other OWL constructs, however, are less frequently used and do add performance risks. In an effort to balance their feature sets and performance capabilities, some vendors allow the user to select the reasoning level to allow IT to make the trade-off between features and performance.

Triple stores may include their own reasoners but often will allow external reasoners to be leveraged. Well-known reasoners include Pellet, Fact++, OWLIM and HermiT. Generally, for more complete OWL support, a commercial system must be purchased. For example, OWLIM offers an open source, lightweight reasoner but requires a commercial license for its full OWL reasoner.

Semantics In Use

Since the initial semantic standards began emerging from the W3C in 2004, companies and governments have begun to leverage the capabilities offered by these platforms. Here are a few examples of entities leveraging semantic technology.

The U.S. and U.K. governments have made the publication of public data a priority. In both cases, they have standardized on semantic technology as the way to make data available. Each has invested in infrastructure and ontology development to promote the use of and simplify the access to their data. This effort has been made for publicly available data, such as that provided by Data.gov, as well as for internal government data federation. For instance, the U.S. Department of Defense has made significant investments in the use of semantic technology.

The British Broadcasting Company began leveraging Ontotext's OWLIM semantic technology in support of its publicly facing sites, including its 2010 World Cup and 2012 London Olympics websites. The flexibility of the data model, as well as the inferencing capability of the reasoner, drove them to develop a semantic system. Thanks to the openness of the BBC engineers who designed and implemented the semantically based systems, their valuable insights and experiences are available to any business beginning the process of understanding the value of semantic technology.

LMI, founded in 1961, is focused on solving complex management and operational challenges for the U.S. federal government. The use of semantic technology began with a small Semantic Wiki project and has expanded from there. Focusing initially on making unstructured information easily accessible, program manager Gus Creedon and systems engineer John Shepherd talked about their use of OWLIM's triple store. They used OWL to create ontologies (controlled vocabularies) that let users search through the text. "Almost like a guided search," says Creedon. "We've used Ontotext and our know-how to create a UI that sits on top of OWLIM."

He went on to describe the benefit to the users of LMI's GovReg website (govreg.lmi.org). Users who search the site see the ontology that was created, which allows them to browse for documents. "The whole point was to allow people to shop for text in the same way that they shop for shoes on the Internet," says Creedon. The site gets about 2,000 hits per month from a variety of government groups. "It's a pretty good demonstration of how you can use an ontology to tag things up," he says.

Asked about the current state of semantic technology acceptance and understanding, Creedon and Shepherd suggested that semantics is new and as a result there are only a few "industrial-strength" technologies out there. Basically, vendors have yet to catch up with government procurement requirements. Further, they say that the "industrial-strength" side is so new that people just aren't yet aware of the technology. "When we go and talk to prospective customers and you're sitting across the table from an IT guy, he'll say, 'Is this written in MySQL, Oracle or DB2 or what?' " says Creedon. "I have to say, 'No. It's written in an RDF triple store.' The look I get is deer in the headlights. So the industry hasn't been able to educate IT professionals. The word hasn't gotten out yet."

There Is No Free Lunch

A few cautions before jumping into this technology. First, there is significant upfront work required to properly use semantic technologies. Creating effective business and system ontologies, understanding how to work with triple stores, and vetting products will all take time.

Second, don't adopt semantic technology just for the sake of using a new tech. Pick a project where your current stack does not seem optimum and where some of the significant challenges appear to be addressable by semantics. If there is no true advantage offered by using semantic technology, then at this point adoption will simply slow down your development processes, frustrate your developers and add unnecessary costs to your systems.

Third, don't skimp on training. Your team will need to invest significant effort to understand RDFS and OWL, the building blocks for your ontologies. Reasoner-based design is different from relational and object-oriented design. Your team might think that semantic technology looks similar to technologies they currently use, but that assumption will lead them far astray. Investing in training and expert mentoring will be well worth the cost when compared with the frustration and rework resulting from misunderstanding semantic paradigms.

Conclusion

Semantic technology creates a data management platform that is significantly different from that of relational databases. Each technology has its sweet spot, and neither is the right answer for all data management needs. The robust ecosystem surrounding relational databases will continue to provide powerful capabilities for managing corporate data. Semantic technology offers a compelling environment for more dynamic data representation and federation needs. Leveraging both RDB and semantic technologies in ways that play to their strengths will provide a real benefit to businesses looking to optimize the value of their data management operations.

Even if you don't see an immediate need, it's worth getting educated on semantics, because data needs and relationships will continue to evolve. In the 1980s we associated people with home and work phone numbers and a mailing address. By the 1990s we also had to represent cell phones and email addresses. In the 21st century we associate people with a broad spectrum of websites and social networking identifiers.

Creating effective business and system ontologies, understanding how to work with triple stores and vetting products will all take time.

And connecting these identifiers isn't the end of the task. Our systems must understand the correct context for using this information. In other words, there needs to be machine-readable meaning associated with data. Leveraging an ontology with a collection of triples, creating a directed graph, is a very effective way to represent all of these resulting interconnections and definitions.

As product vendors and the broader ecosystem of reporting and data management tools continue to develop and refine their semantic offerings, the value of leveraging this data technology will continue to overtake RDB in more and more use cases. Becoming familiar with the semantic technology landscape and gaining experience with these platforms now will allow a smooth transition and provide near-term benefits, while avoiding a need to catch up with competitors in the future.

An Example Of Restructuring A Data Relationship

By way of comparing the agility offered by semantic data technology when dealing with a new data relationship, here is an example of a common type of relationship change. We'll explore how such a change is handled in RDB and semantic technologies.

We begin with a simple relationship. We have a person entity that has a one-to-one relationship with a birth date and a phone number.

The ontology represented in Figure 9- creates a class of person and associates birth dates and phone numbers with that class. Further, it states that there is only one birth date and one phone number per person entity. Figure 10 shows some triples representing data that is aligned with the ontology.

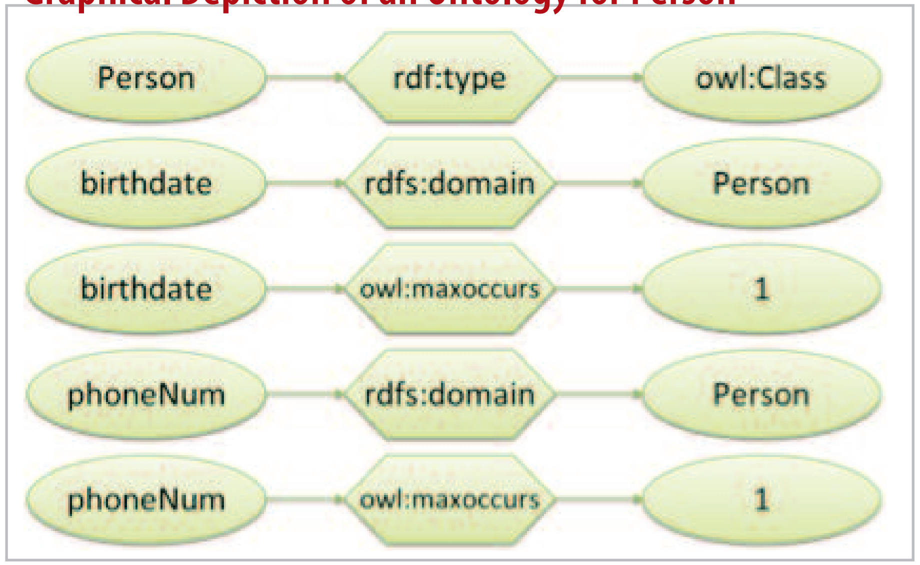
The table depicted in Figure 11 provides a similar definition for person and also houses the sample data.

Say we're now asked to change the relationship between a person and phone numbers because marketing needs to support multiple phone numbers per person. We need to change the one-to-one relationship to a one-to-many relationship.

In the semantic technology approach, we

Figure 9

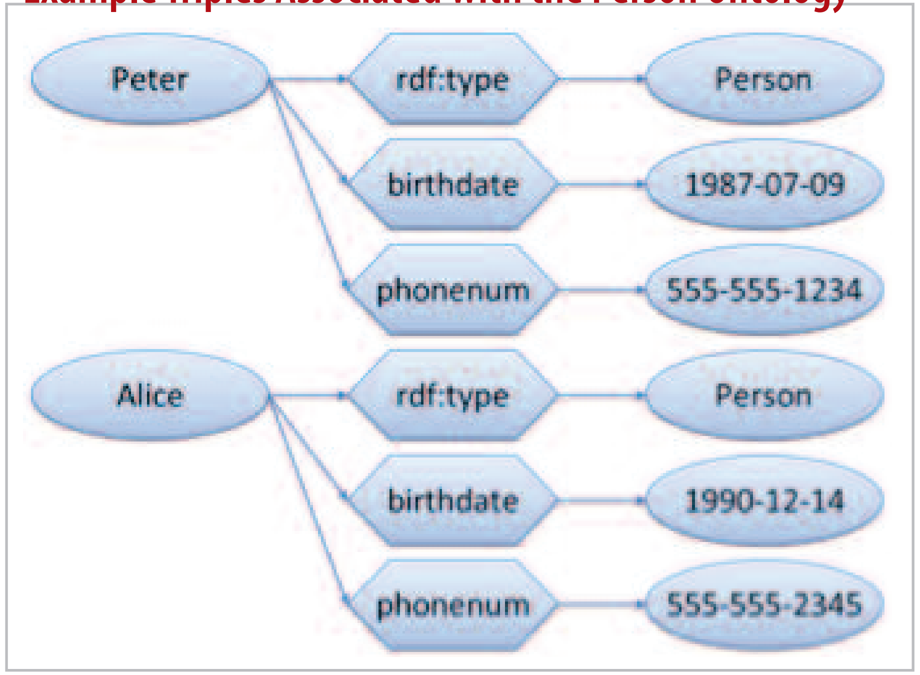
Graphical Depiction of an Ontology for Person



Data: *InformationWeek* Reports R6171212/9

Figure 10

Example Triples Associated with the Person Ontology



Data: *InformationWeek* Reports R6171212/10

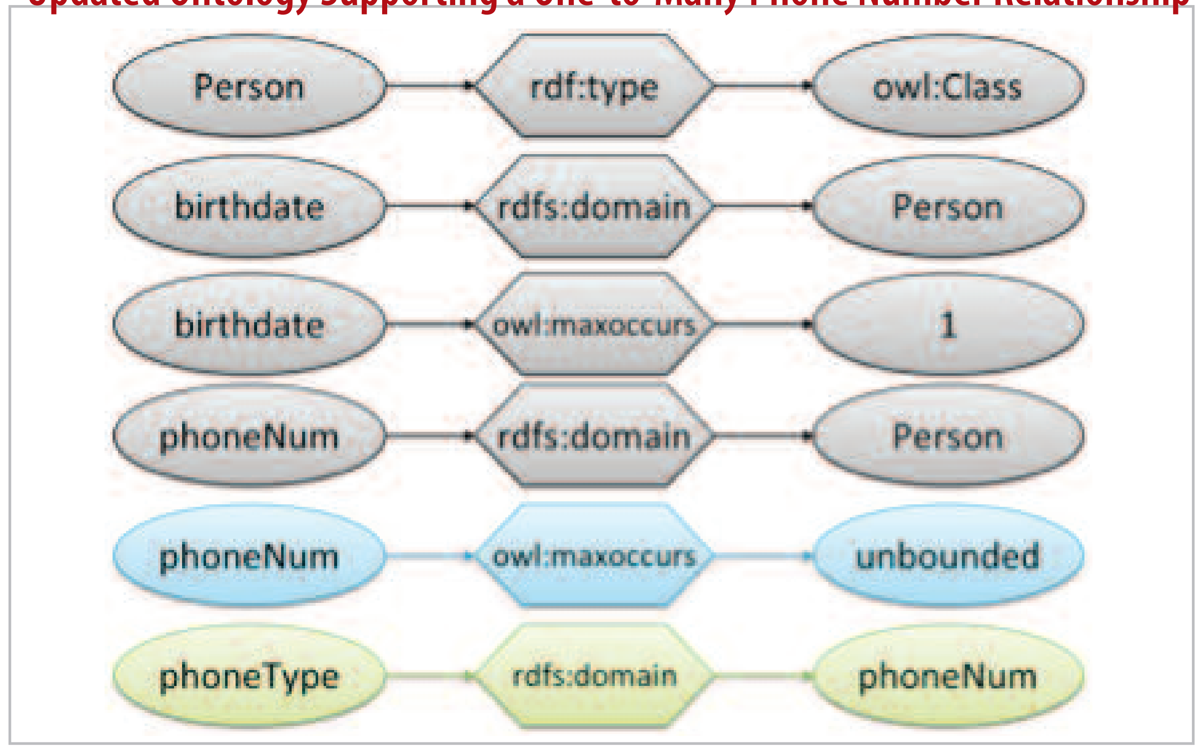
need to change the ontology to reflect this new relationship. Figure 12 shows the new ontology. Items in gray were unchanged. The "maxoccurs" definition was changed for phone number to be unbounded, meaning that multiple numbers may be applied. We have also added a phone type attribute to the phone number.

The updated triples, now supporting multiple phone numbers, are shown in Figure 13. The key point to note is that the existing data was not changed. There was no need to refactor the data in support of the new relationship. We simply added the new relationships, such as which phone numbers were for home, office and cell. The existing person and phone number relationships stay as they were.

The process for relational databases is different. In order to support the need for the one-to-many relationship, we will need to create a new table that houses the phone type and phone number data. A foreign key is then used to associate the phone numbers with the correct person. Finally, we need to remove the phone number column from the original person table. The updated database is depicted in Figure 14. The physical changes and moving of data are required since data relationships are represented in the physical structure of a relational database.

Figure 12

Updated Ontology Supporting a One-to-Many Phone Number Relationship



Data: *InformationWeek* Reports R6171212/12

Figure 11

Person Table with Birthdate and Phone Number Columns

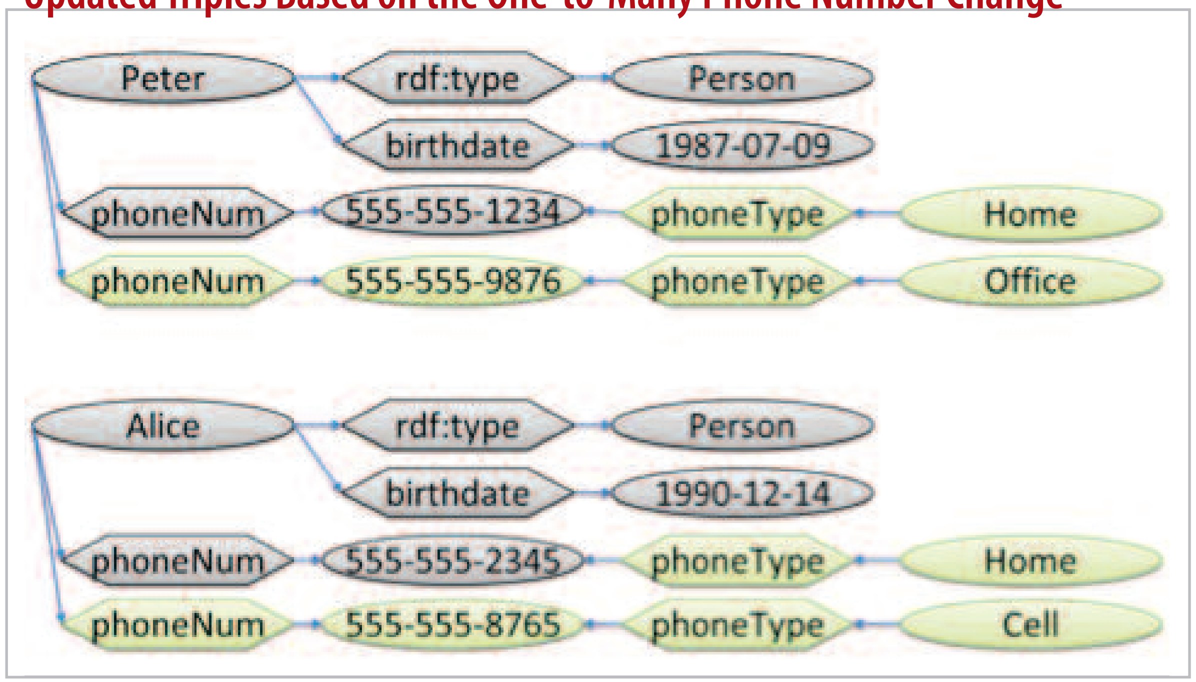
Table: Person		
name	birthdate	phoneNum
Peter	1987-07-09	555-555-1234
Alice	1990-12-14	555-555-2345

Data: *InformationWeek* Reports R6171212/11

These types of changes, which can be much more involved, are common occurrences for applications whose requirements are evolving or systems within businesses that continually seek to integrate with partners, vendors and customers. The reduced effort and lowered risk when integrating data using semantic technology provides a compelling reason to begin exploring these platforms.

Figure 13

Updated Triples Based on the One-to-Many Phone Number Change



Data: *InformationWeek* Reports R6171212/13

Figure 14

Updated RDB Schema Supporting a One-to-Many Phone Number Relationship

Table: Person

name	birthdate
Peter	1987-07-09
Alice	1990-12-14

Table: Phone

name	phoneType	phoneNum
Peter	Home	555-555-1234
Peter	Office	555-555-9876
Alice	Home	555-555-2345
Alice	Cell	555-555-8765

Data: *InformationWeek* Reports R6171212/14

MORE LIKE THIS

Want More Like This?

InformationWeek creates more than 150 reports like this each year, and they're all free to registered users. We'll help you sort through vendor claims, justify IT projects and implement new systems by providing analysis and advice from IT professionals. Right now on our site you'll find:

Research: 2012 Big Data and Analytics Staffing Survey: It's a good time to be a big data and analytics expert: demand is high and supply is low. Eighteen percent of big data-focused companies in our survey want to increase staff in this area by more than 30% in the next two years, but 53% expect it will be hard to find people with the required skills. Find out how companies in diverse industries are coping with the shortage.

IT Pro Impact: In-Memory Analytics and Databases: It's taken six years or so of effort, but vendors have finally aligned the technology and the economics to bring capabilities once limited to telcos and trading floors to companies looking to maximize the value of their big data. The falling prices of DRAM make the business case that much stronger and mean that DBMS architects can, finally, dare to think beyond disk. We'll discuss the state of the market and connect platform choices with real-world deployment scenarios.

Strategy: NoSQL Everywhere? Not so Fast: Enterprise big data workloads are incredibly diverse and require a smart mix of conventional SQL for bedrock, low-latency structured data tiers and NoSQL for unstructured workloads that can be served using commodity hardware and software. We'll discuss how to arrive at the right mix using a "three V" approach that considers data velocity, volume and variety.

PLUS: Find signature reports, such as the *InformationWeek* Salary Survey, *InformationWeek 500* and the annual State of Security report; full issues; and much more.