InformationWeek :: reports

The Rise Of Semantic Databases

By David S. Read

Emerging database technology promises to automate more analysis. Here's where it could replace relational systems.

Don't buy into the idea that semantic database technologies are just for consumer-facing services such as BBC Online or the semantic Web initiatives embraced by the likes of Best Buy and Cisco. In much the same way that consumerization drives innovation in end user computing, semantic database technologies deliver benefits that businesses of all stripes should be exploiting.

At a high level, semantic databases offer five main benefits: They work with your existing relational databases. They align with Web technologies. Their underlying technology speeds integration of multiple databases. They're based on data structures that are flexible by design. And thus they can help enter-

prises tackle big data challenges.

Their biggest selling point is that data relationships are continually evolving. In the 1980s, we associated people with home and work phone numbers and a mailing address. By the '90s, we also had to represent cellphones and email addresses. And now we associate people with a broad spectrum of websites and social networking identifiers.

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.

Not coincidentally, the mantra for semantic technology is that information must be machine readable. The data definition must provide meaning and context so that a computer can understand the information rather than require a human to interpret it. This goes beyond supplying a structure for data, such as an XML schema. It requires that concepts *within* the data be given an ontological basis — 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 on purpose). OWL defines a rich set of data relationship descriptors used to create a set of definitions for business terms, data sets and attributes. This resulting set is called an ontology.

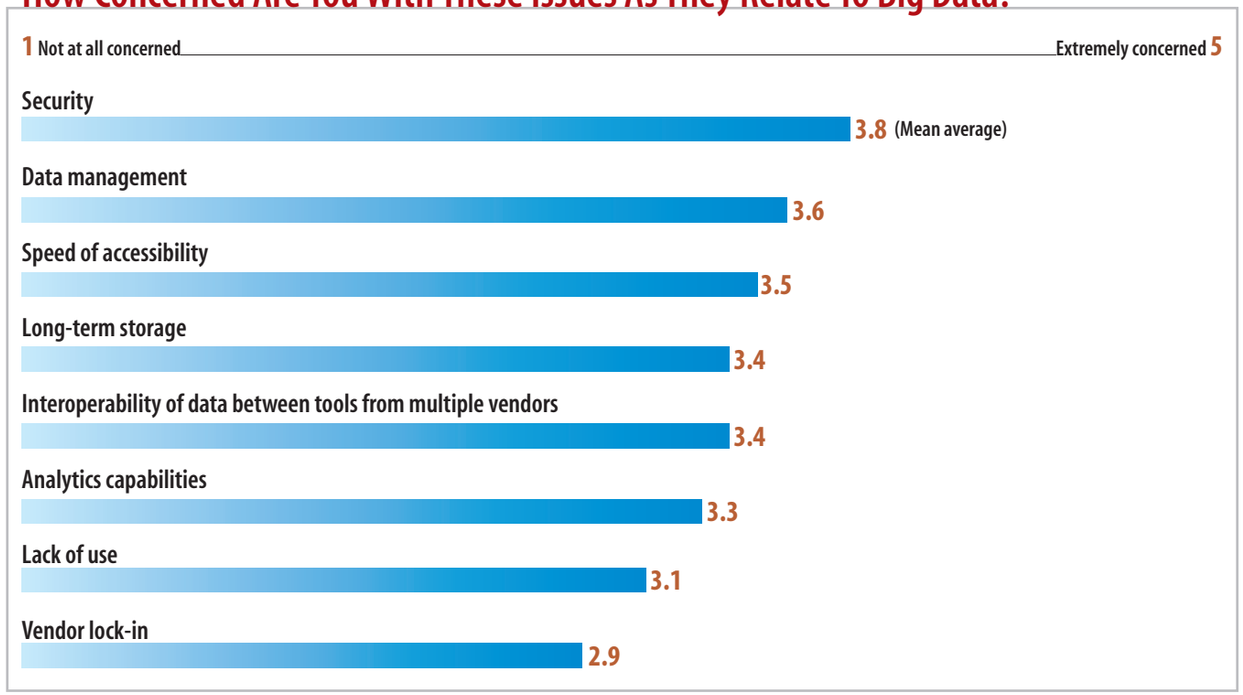
Typically, a business will create and use multiple ontologies. At a corporate level, they define operations, departments, metrics and other core business concepts. At a system level, ontologies describe the types of data being managed and the specific attributes of that data, as well as the structure and relationships inherent in the attributes. Although separate, these ontologies are related, so that system-level details relate to corporate-level concepts.

From a system perspective, the ontology is a logical representation of the data's meaning and relationships. The ontology doesn't generate a table or define a storage mechanism; it simply communicates the relationships that may be found in the data. In a relational table, the columns in the table do dictate the data storage. That's a key difference. And data from

How The Technologies Compare

	RELATIONAL TECHNOLOGY	SEMANTIC TECHNOLOGY
Maturity	> More than 40 years of standards, product development/enhancements and field use. > Wealth of supporting tools for database design, visualization, integration and reporting.	> Very young set of standards and a first-generation product set. > Narrow ecosystem of tools. Very limited options for plug-and-play reporting, for example.
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 added 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.

How Concerned Are You With These Issues As They Relate To Big Data?



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

many sources, such as relational databases, key-value data stores and spreadsheets, can be used to originate semantic data. In fact, a core feature of semantic technology is that you don't have to rehost data if your existing relational database structures serve the business well.

We discuss the basics of semantic technology in much more depth, including code examples, in our full report.

Why Not Relational?

Most companies have made significant investments in relational database technologies, including management software, ETL tools, data marts, data warehouses and a variety of reporting systems. So does semantic technology justify investing in new infrastructure, training and refactoring?

To see if semantic technology should be on your radar, consider three situations in which a semantic data store can provide significant advantages.

>> You must integrate heterogeneous data: Semantic technology's focus on federation and interoperability makes it ideal for processing heterogeneous data. Matthew Petrillo, president of Ontotext, the creator of semantic repository OWLIM, describes semantic data stores as being a "schema last" paradigm, where data can be placed in the database without knowing the structure up front. You can then apply one or more schemas (ontologies) to organize the data afterward.

>> You have evolving data structures: As businesses demand more agility to support changing markets, they need to process new types of information quickly. Semantic technology works very well under those conditions. As a platform for working with Web-based data, for example, semantic technology standards were built to assume that new data would be discovered and combined with existing data.

>> You have 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 application may offer users a variety of review outcome choices based on the

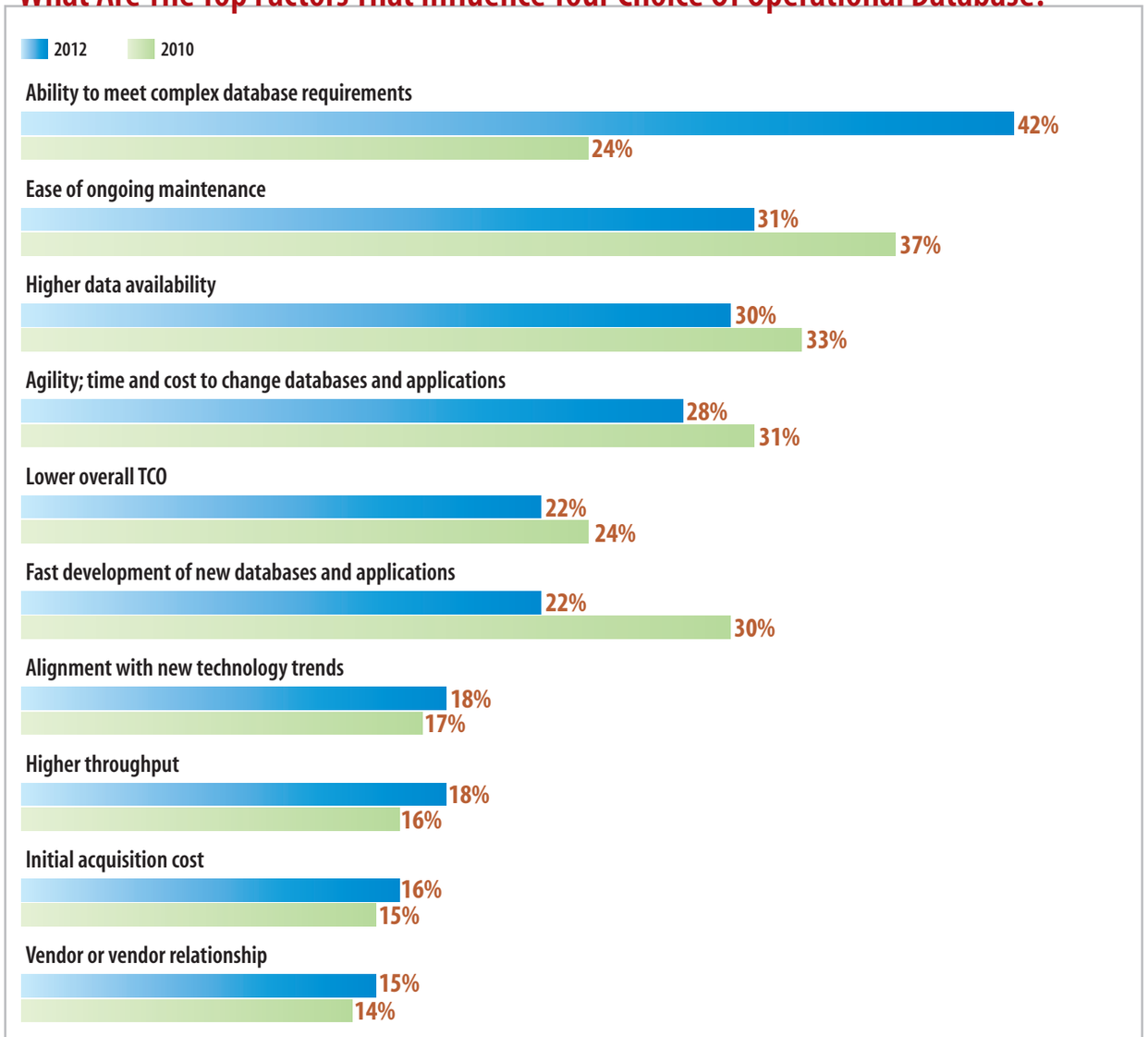
procedure codes on the claim. The review outcome choices are data, so the rules in this case are associating one set of data (procedure codes) with another set of data (review outcome choices).

Rules of this nature are often baked into application code or placed in a business rules engine. However, it's much more effective to place them in the ontology. In our claim processing example, it would be easy to understand the data relationships encoded in the ontology, since the context is clear. If these data-behavior rules are placed elsewhere, they're disconnected from the data, which often leads to errors or missed opportunities. It's also much more difficult to test such rules, since they're essentially housed in two places: The data and definitions are within the database, while the rules interpreting the data are in source code or a business rule environment.

No Free Lunch

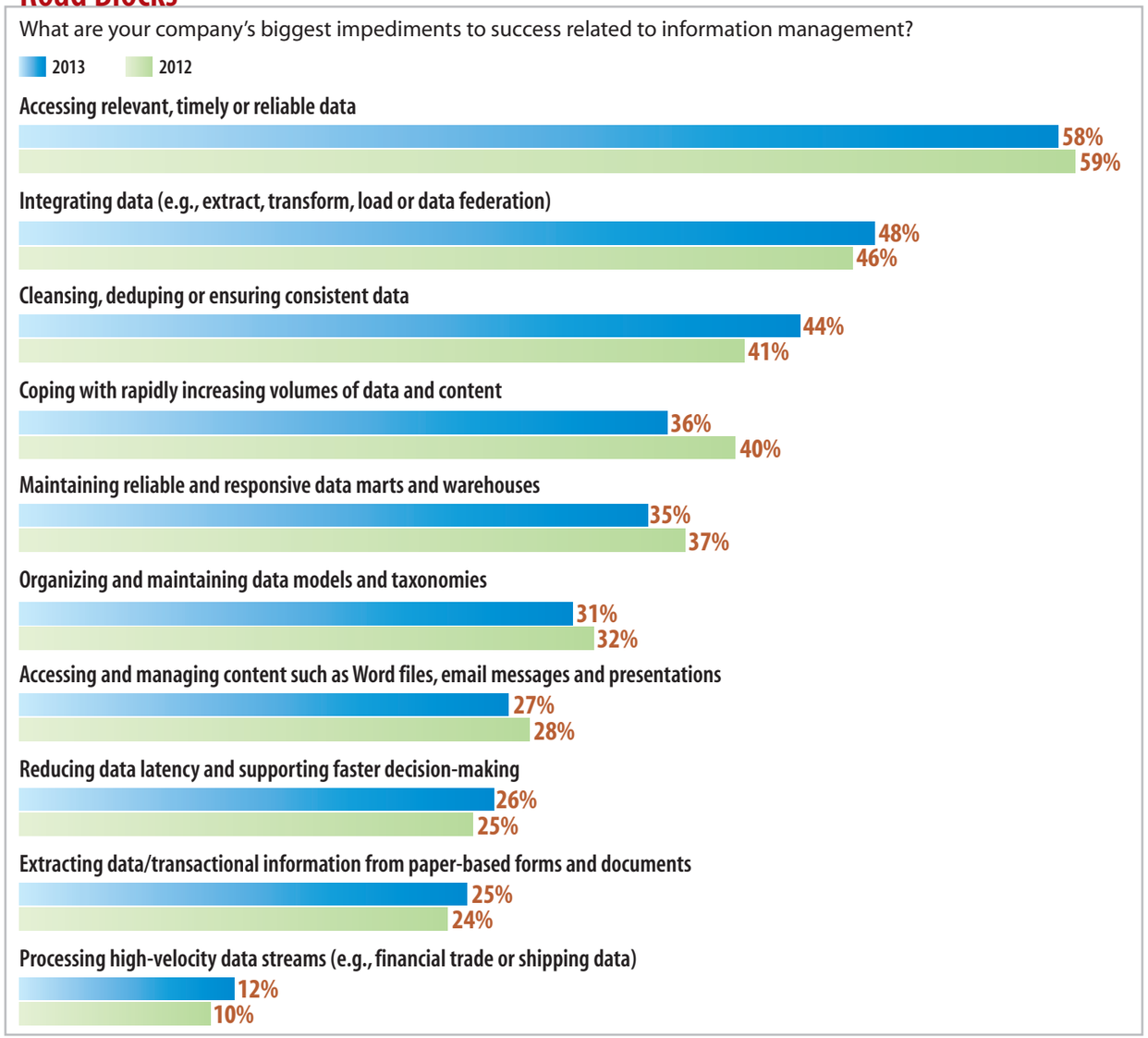
Don't adopt semantics across the board just for the sake of using a hot technology. There's significant upfront work in creating effective business and system ontologies, understanding how to work with semantic-specific technologies such as triple stores

What Are The Top Factors That Influence Your Choice Of Operational Database?



Data: *InformationWeek* State of Database Technology Survey of 760 business technology professionals in November 2011 and 755 in August 2010

Road Blocks



Data: *InformationWeek* Business Intelligence and Information Management Survey of 517 business technology professionals in October 2012 and 431 in October 2011 involved with information management technologies

and vetting still-emerging products. To start, pick a project where your current stack is showing strains that semantics could address. Otherwise, adoption will slow down and frustrate your developers and add unnecessary costs.

Don't skimp on training. Semantic design is different from relational and object-oriented design, even if your team thinks it looks similar.

Because semantic technology is significantly different from relational, it's not the answer for all data management needs. The robust relational database ecosystem will continue to support much of your corporate data. Semantic technology, on the other hand, offers a compelling environment for more dynamic data representation and federation. Leverage both in ways that play to their strengths.

Get familiar with the semantic technology landscape and gain experience with these platforms. Doing so will allow for a smooth transition and provide near-term benefits, while avoiding the need to catch up with competitors in the future.

David S. Read is CTO of management consultancy Blue Slate Solutions. Write to us at iwletters@ubm.com.