



Managing Semantic Content for the Web

Amit Sheth, Clemens Bertram, David Avant, Brian Hammond, Krzysztof Kochut, and Yashodhan Warke • Voquette and the University of Georgia

By associating meaning with content, the Semantic Web facilitates search, interoperability, and the composition of complex applications.¹ A recent *Scientific American* article described the Semantic Web as “an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”² Not long ago, researchers at a Stanford University symposium predicted that this second phase of the Web would be as revolutionary as the Web itself.

As this article describes, the Semantic Content Organization and Retrieval Engine (SCORE, see www.voquette.com), which is based on research transferred from the University of Georgia’s Large Scale Distributed Information Systems, belongs to a new generation of technologies for the emerging Semantic Web. It provides facilities to define ontological components that software agents can maintain. These agents use regular expression-based rules in conjunction with various semantic techniques to extract ontology-driven metadata from structured and semistructured content. Automatic classification and information-extraction techniques augment these results and also let the system deal with unstructured text.

Because a semantic engine with main-memory-based indexing provides high-performance content, metadata, and knowledge querying, SCORE can comprehensively support semantic application development that would otherwise require lot more programming and perform much slower using information retrieval and database management systems. Such applications involve context-sensitive search, browsing, correlation, normalization, and content analysis.

How SCORE Works

SCORE supports four key capabilities that constitute the core of semantic technology:

- *Semantic organization and use of metadata.* Realizing a semantic Web solution often involves using ontologies to organize concepts and domains, as well as metadata to annotate and enrich content.^{3,4} Metadata takes two forms: syntactic and semantic. Syntactic metadata describe noncontextual information about content, such as language, bit rate, and format. Such metadata offers no insight into a document’s meaning. By contrast, semantic metadata describe domain-specific information about the content. If the content is from the finance domain, for instance, the relevant semantic metadata might be company name, ticker, industry, and executives. If it’s from the baseball domain, the relevant semantic metadata might be player, team, and league. Ontologies provide the context for semantic metadata.
- *Semantic normalization.* Normalization plays an important role in dealing with semantic heterogeneity associated with multiple data sources. One kind of metadata normalization associates the same metadata with content belonging to the same domain regardless of source and format. If an article in a NewsML feed and a PDF article posted on a Web site both discuss equity research reports, for example, the same type of metadata should be associated with them. The other kind of metadata normalization homogenizes multiple names of a single concept into one canonical name. For example, Yahoo’s founder is referred to as

“Chief Yahoo” within the company, but externally as “Yahoo Founder.” Both names refer to the same person or entity, but a keyword search made on either name will not turn up results pertaining to the other. Strong mapping techniques are key to supporting normalization.

- **Semantic search.** Current search engines cannot know whether the search term `palm` is a company (`company: palm`), a technology (`operating system: palm`), or a product (`PDA: palm`). Today’s engines do not consider the query’s context. In some cases, the search could be limited to the technology category, but this alone still does not differentiate the OS from the PDA. A more difficult query would be to find movies that Robert Redford directed, but not those in which he acted with a different director. Nonsemantic search engines cannot correctly answer such queries because the keywords `director` and `Robert Redford` could occur in documents not satisfying these criteria. Semantic annotation or metadata associated with content also enable more powerful browsing and personalization.
- **Semantic association.** Consider an application to support a financial advisor who is evaluating Intel Corporation. Semantic technology can infer that a recently released report on the semiconductor industry is of interest to that advisor because Intel is an important company in this sector. The search engine presents the report to the advisor, providing relevant information the advisor did not explicitly request. To provide this functionality, a search engine could determine, with some degree of confidence, that the report has some relevancy to the keyword `Intel`. Statistical and learning methods might produce such a result. The very different semantic solution is to know with certainty that the report is about the semiconductor sector, to which the company Intel belongs. This approach requires an ontology involving the concepts of sectors and companies and the relationship between them.

The benefits of semantic associations are best realized in applications that integrate data, metadata, and knowledge queries. To better understand the key capabilities of a comprehensive semantic technology, consider a search on `Tiger Woods`. The traditional approach is to look for pages containing `Tiger Woods`, `Tiger AND Woods`, and `Tiger OR Woods`. Engines also use information-retrieval techniques, including

- word collocations and frequencies;
- a Web site’s trustworthiness, importance, and popularity; and
- restriction of results to a particular category using directories or categorization.

Such approaches can lead the user to an official site, a home page, and fan pages for `Tiger Woods`, whatever “Tiger Woods” is.

The semantic solution starts with the realization that “Tiger Woods” is not merely two adjacent words, but actually identifies the well-known golf champion. Semantic technology can further reveal that he is both a golfer and a celebrity spokesperson. If a document involves `person: Tiger Woods` in the context of golf, the relevant metadata include tournaments and golf courses men-

The benefits of semantic associations are best realized in applications that integrate data, metadata, and knowledge queries.

tioned in or inferred from the text. If the document is about advertising, the represented companies and products are of interest.

Once the semantic search engine determines the context of information described in a document, it can explore related entities through associations. In the golf domain, these include players Tiger Woods played against, tournaments in which he participated, and the golf courses where they were held. By navigating these associations or relationships, the engine can access content about these entities.

System Architecture Overview

Ontologies play a central role in most semantic technologies. They form the basis for syntactic and semantic metadata, which can be used for annotating or tagging content. The content’s context determines which semantic metadata to extract. Automatic classification technology helps select the context by classifying documents into one or more categories and extracting or inferring semantic metadata corresponding to one or more contexts.

We have divided an ontology into two related components: the definitional component called `WorldModel` and the assertional component called

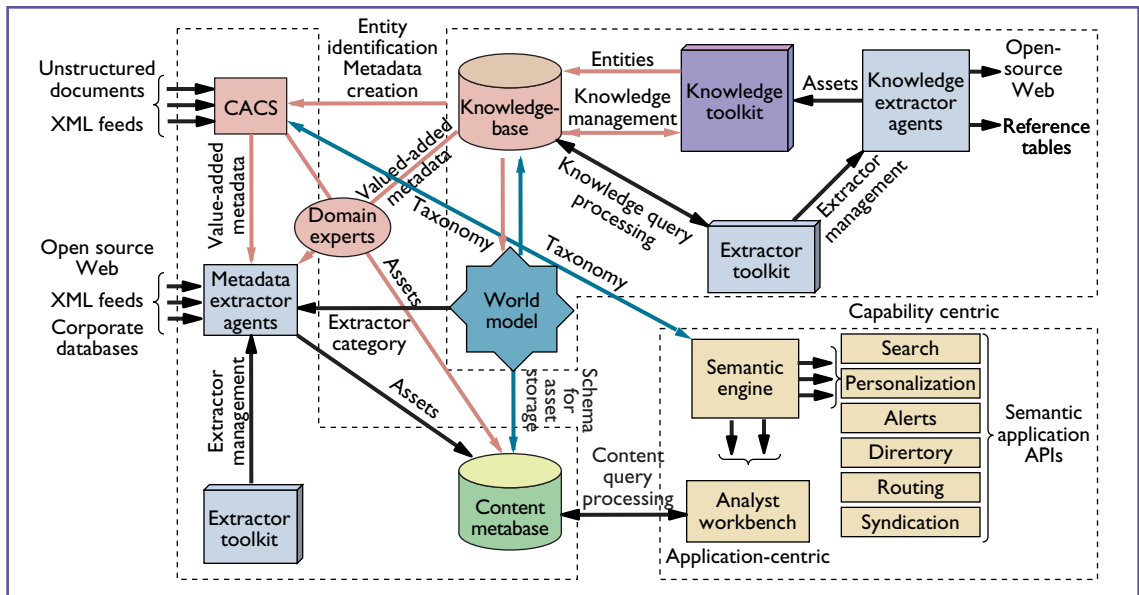


Figure 1. SCORE system architecture. The three activities bounded by dashed lines cooperate through XML-based knowledge and metadata sharing.

Knowledgebase. As with the specification of ontologies, the WorldModel and Knowledgebase definition process involves domain-specific expertise as well as an understanding of eventual application requirements. While some clustering techniques can provide initial input to semiautomate this process, it cannot generally be completely automated if high-quality results are needed.

The Knowledgebase reflects the subset of the real world for which a semantic application is created. As such, it is an important part of the solution. It lets us extract value-added semantic metadata, such as the ticker symbol “INTC” when a document only mentions the company “Intel.” Additionally, it provides the framework for semantic associations.

In addition to these two components, SCORE provides a query-processing system. A comprehensive suite of APIs uses this query-processing capability to support rapid development of semantic applications such as search, directory, personalization, syndication, and custom enterprise applications.

The operation of a SCORE technology-based system involves three independent activities as illustrated by the dashed areas in Figure 1.

Defining WorldModel and Knowledgebase is the first activity (Figure 1, top-right). Knowledge extraction agents manage the Knowledgebase by exploiting trusted knowledge sources. Different parts of the Knowledgebase can be populated from different sources. Various tools help detect ambiguities and identify synonyms. Commercial deployments of SCORE can be expedited with a pre-defined WorldModel and Knowledgebase.

Content processing comes second (Figure 1, left). This includes classifying and extracting metadata from content. The results are organized according to the WorldModel definition and stored in the Metabase. Knowledge and content sources can be heterogeneous (XML, resource description framework [RDF], static and deep Web pages, database, or documents in various formats), internal or external to the enterprise, and accessible in push (content feeds or database exports) or pull (Web site) modes.

Support for semantic applications comes last (Figure 1, bottom-right). The semantic engine processes semantic queries, but does not currently support inference mechanisms found in AI or logic-based systems. Instead, it provides limited inferencing based on the traversal of relationships in the Knowledgebase. An API for building traditional and customized applications returns results as XML to facilitate GUI creation.

Semantic Metadata Extraction of Structured and Semistructured Text

Crawling and information extraction technologies come in wide varieties (see the “Crawler and Extraction Technologies” sidebar). SCORE’s approach combines four key capabilities:

- Extracting metadata by scanning unstructured text as well as by exploiting the content structure.
- Identifying both domain-specific (semantic) and domain-independent (syntactic) metadata, including those from audio/video content

Crawler and Extraction Technologies

Traditional crawlers and screen scrapers started with the Harvest project. Today's search engines employ highly scalable crawlers and generate extensive statistical information beyond scraping text on Web pages and indexing. Wrappers and extractors allow creation of more metadata that capture syntactic, structural, and in some cases semantic metadata. The following are example toolkits that support wrapper and extractor creation:

- Andes (J. Myllymaki, "Effective Web Data Extraction with Standard XML

Technologies," WWW10, www.I0.org/cdrom/papers/I02);

- XWRAP (L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," *Proc. Int'l Conf. Data Engineering*, 2000, www.cc.gatech.edu/projects/disl/XWRAPelite),
- W4F (A. Sahuguet and F. Azavant, "Building Lightweight Wrappers for Legacy Web Data-Sources Using W4F," *Proc. Int'l Conf. Very Large Data Bases (VLDB)*, 1999, db.cis.upenn.edu/Research/w4f.html).

Relevant terms used in the literature (with some distinction in meaning), include Web mining, focused crawling and extraction, and information extraction. The SCORE white paper from Voquette (www.voquette.com/demo) provides additional information on crawler technologies in terms of crawling range, categorization, extracted and indexed features, attribute search, and so forth. Related terms to information extraction are metadata tagging and metadata extraction.

(speech-to-text data and encoded metadata in the header).

- Enhancing the extracted information using the Knowledgebase.
- Maintaining the Knowledgebase using extraction technology, avoiding the problem of static, soon-obsolete dictionaries.

Guided by the WorldModel (Figure 1, center-middle), the creation of extractor agents is ontology-driven. The WorldModel contains a hierarchy of categories or domains, each possessing a set of inheritable attributes. The idea is that documents belonging to different domains have their own distinct sets of interesting, "domain-specific" properties.

All documents, however, have a source, creation date, title, description, and other domain-independent properties. Those generic attributes are associated with a top-level category and are inherited by all other categories. An *asset* is the collection of all metadata for one piece of content.

The extractor toolkit creates extractor agents for a particular information source, such as a NewsML feed or a Web site. SCORE assigns this information source to a particular WorldModel category. Regular expression-based crawling rules then guide the agent through the source to individual pieces of content, which might be generated dynamically. Similarly, extraction rules take advantage of available structure within the content to reliably retrieve attribute values for the assigned category.

When an extractor agent runs, it applies these rules to the source text and generates assets. Only in some cases will the content source be structurally "rich" enough to provide values for all attributes. Enhancement rules define how the

extractor agent will use the Knowledgebase to populate empty attribute values, either by identifying relevant entities in the text or inferring them through relationships. Once the categorization and auto-cataloging system (CACS, Figure 1, top-left) identifies candidate entities, scoring heuristics weed out false positives. SCORE then stores the enhanced asset in the Metabase.

SCORE stores entities and their associated relationships in the Knowledgebase (Figure 1, center-top), classifying them according to a hierarchical entity class tree. A given entity can belong to multiple entity classes. For example, one branch of the class tree contains *person* with subclasses *politician*, *artist*, and *sportsPerson*; *sportsPerson* divides further into *coach*, *athlete*, and so on. The entity Jesse Ventura belongs to the entity classes *politician* and *athlete*. Clearly, he plays two roles.

The Knowledgebase also defines relationships between entity classes that the entities belonging to those classes (or their subclasses) can participate in. *Jesse Ventura holdsOfficeOf Governor* is an instance of the relationship *politician holdsOfficeOf politicalOffice*, and *David Letterman interviewed Jesse Ventura* is an instance of *person interviews person*.

Entities can possess synonyms that support variations of their name (Toys 'R Us versus Toys-R-Us) or even nicknames (Jesse Ventura versus The Body). The enterprise can define a custom Knowledgebase structure, including the types of entity classes and relationships, and tailor it to the application's requirements.

In addition to providing reliable entity detection, enhancement rules might also use CACS to automatically classify a document. In such cases, an

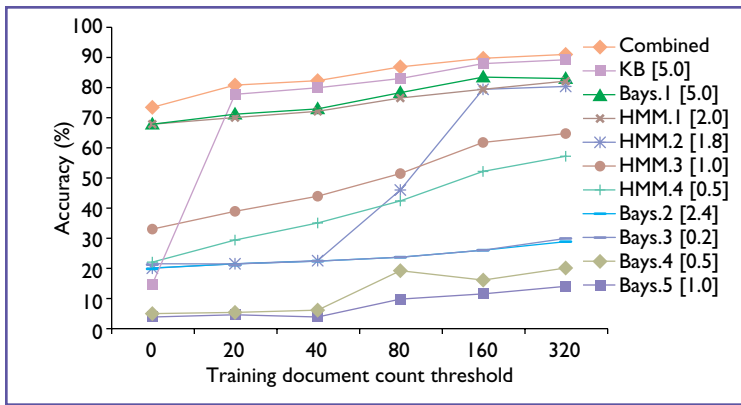


Figure 2. Classification results for the Reuters-21578 text categorization test collection. Values in brackets are the weights for the various classifiers.

Table 1. Category count.

Threshold	Category Count
0+	115
20+	44
40+	31
80+	18
160+	10
320+	7

asset’s source text feeds into CACS, which returns a document category. For example, a user might want to classify *finance* documents into the topics *analysis*, *IPO*, *earnings*, *market commentary*, and *mergers*. Instead of introducing a number of subcategories for the *finance* WorldModel category, the user could add an attribute “topic” to distinguish these types of documents; this attribute would then get its value from CACS’s classification result.

Automatic Classification by Classifier Committee

A wide variety of developers in both the academic and corporate realms have researched, designed, and implemented numerous classification methods.⁵⁻⁸ Debate rages over which method works best. Rather than take a “one size fits all” approach, we decided to combine disparate methods into a *classifier committee*.⁵ A classifier committee combines techniques such that the overall accuracy exceeds each classifier’s accuracy. This approach works best when the individual classifiers use disparate techniques. It also lets system designers integrate new techniques into the combined result easily.

Classifier committees use various methods, including probabilistic (Bayesian), learning (Hidden Markov Models), and knowledge-based techniques. Both entity recognition and use of domain phrases fall under the last category. The former uses entities and entity classes found in the text to derive the classification result, while the latter relies on either handcrafted or automatically discovered phrases that are significant for the various categories; for instance *birdie*, *putting*, *tee* for golf or *loss per share* for earnings. Because the various classifiers seldom have the same accuracy, this technique attaches a weight to each classifier, which then scales the results before combining them. A variant of the weight linear combination approach developed by Leah Larkey and Bruce Croft determines these weights at training time.⁵

Figure 2 shows results based on the Reuters-21578 text categorization test collection. In this test, we used a different threshold for the minimum number of documents per category in the training sets under the LEWISSPLIT (a well-known partitioning of the Reuters document set into training and test sets), which produced the category counts at various thresholds (Table 1). For each threshold, the categories meeting the minimum number of training documents served to train the classifiers, with tests then conducted on the corresponding test sets. The classification committee consistently outperformed the individual classifiers.

Metadata Extraction from Unstructured Text

There is a strong tie between classification and metadata extraction. The best way to show this connection is via a simple example:

- *Example 1: Category = Finance.* Many venture capitalists have decided that now is a good time to invest in the technology sector. John Smith recently invested in Voquette in anticipation of a large return on a modest investment.
- *Example 2: Category = Baseball.* Coming into the World Series, John Smith has had an incredible number of homeruns and is heralded as the “Babe Ruth of BeanTown.”

In each example, a section of text, *John Smith*, matches an entity. The ambiguity arises because there might be more than one *John Smith* in the Knowledgebase, one a businessman and another a baseball player. Deciding which to choose in each case is

Table 2. Performance data, based on a dual Pentium III 766-MHz processor, 2-Gbyte RAM, running RedHat Linux with Apache as Web server and MySQL as a lightweight database.

Parameter	Specification
Queries per server per hour	>1,980,000
Query response time (light load)	1 to 10 ms
Query response time (64 concurrent users)	65 ms
Incremental index update frequency	1 minute (near real-time)
Population/update rate in a Knowledgebase with 1 million entities/relationships	>10,000 entities/relationships per hour

called resolving metadata extraction ambiguities.

SCORE uses two methods of resolving ambiguities: classification and Knowledgebase. Each method has complementary strengths and weaknesses.

The classification-based method associates sets of entities and entity classes with document categories, either by human-generated rules or from training documents. Once a tool using classification-based methods classifies an incoming document, it chooses the appropriate set for the domain. It then intersects this set with the set of matched entities, thereby reducing or eliminating ambiguities. This method works well, but depends on correct classification. Conversely, these sets can be used to classify documents as well.

SCORE's Knowledgebase method of resolving ambiguities leverages the fact that entities in a document are often related (*John Smith lives in Boston*). They might also belong to the same entity class (*John Smith and Babe Ruth are both athletes*). To combine these methods, SCORE attaches a weight to each entity at each position in the text at which it is matched. SCORE also considers additional textual features, such as exact case matches, surrounding words, and position within the sentence.

Semantic Search Engine

Using extracted and enhanced metadata stored in the WorldModel will yield high-quality search results because they provide the basis for contextual search. Attribute search produces highly precise results. Here, the user specifies the category and one or more attribute values, for instance `Golf::player=Tiger`.

SCORE's semantic engine (SSE) creates a main-memory index of the metadata and Knowledgebase components. It retains attribute information, supports phrases and exclusion, and typically performs queries in fewer than 10 ms. Storing the index in main memory exhausts the server's capacity limits sooner than in a database-driven environment. SCORE is not primarily designed for

full-text indexing of a large body of documents, but the system can be configured to also index all or a part of document text. User feedback of the deployed systems confirms that searching against only the documents' syntactic and semantic metadata still produces better search results.

The SSE provides an API that serves as the basis for all semantic applications. All query results return as XML, allowing for easy creation of browsing, search, or more customized applications.

Performance and Scalability

Given the ever-increasing volume of available content, it is important that a semantic technology's components and processes—including Knowledgebase creation and maintenance, classification, and metadata extraction—are as scalable and automated as possible. Table 2 shows the performance, scalability, and robustness characteristics of the current version of SCORE.

The main memory index holds metadata of about 4.5 million documents per server in the configuration shown in Table 2. If users need to store more data, they can use a server with more memory or distribute the index over a number of servers.

Minimizing human involvement by automating most of the work is a key factor in scaling up extraction and maintenance of the Knowledgebase. Three full-time extractor writers can write and maintain a few hundred Web-based extractors, assuming that the extracted sources change no more than a few times per year. If content is made available as XML or RDF, this number increases manifold, as the data's final presentation would change but not the underlying data format.

Extractors can run concurrently because the execution environment (Java Virtual Machines) supports a distributed agent infrastructure.

Figures 3 and 4 (next page) show two Semantic Web applications that demonstrate SCORE's core capabilities. See www.voquette.com for more application examples.



Figure 3. The Semantic Console illustrates SCORE's semantic search and knowledge-browsing capabilities. The left panel presents search results from the Metabase, while the right panel displays related entities. Links to other associated entities enable Knowledgebase navigation and refresh the Metabase results upon request. This interaction mode is called blended semantic browsing and querying.¹⁰

Conclusion

SCORE has shown that it is possible to build a comprehensive solution for research- and analysis-oriented semantic applications that deal with a broad variety of content sources. It draws from and enhances many techniques from work in information retrieval, AI, database management, and knowledge representation, and represents a milestone in information system's move from syntax to semantics.¹² SCORE facilitates the creation and maintenance of large knowledgebases that provide a foundation for its semantic capabilities. Enterprise applications enabled by SCORE and other semantic technologies are already being deployed. At the same time, many advanced and exciting capabilities of the Semantic Web are in research and prototyping phases.^{1,11}

Acknowledgments

Research leading to SCORE was performed in the Large Scale Distributed Information Systems Lab at the University of Georgia. This technology was licensed to Taalee, which further developed and patented the technology.¹⁰ Voquette later acquired Taalee and now provides SCORE-based products.

References

1. D. Fensel et al., *Spinning the Semantic Web*, MIT Press, Cambridge, Mass., 2002.
2. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic

Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities," *Scientific Am.*, vol. 284, no. 5, May 2001, pp 28-37.

3. M. Carrara and N. Guarino, *Formal Ontology and Conceptual Analysis: A Structured Bibliography, Tech. Report*, Mar. 1999; www.ladseb.pd.cnr.it/infor/ontology/Papers/Ontobiblio/TOC.html.
4. A. Sheth and W. Klas, eds., *Multimedia Data Management: Using Metadata to Integrated and Apply Digital Data*, McGraw Hill, New York, 1998.
5. L.S. Larkey and W. Croft, "Combining Classifiers in Text Categorization," *Proc. 19th ACM Int'l Conf. Research and Development in Information Retrieval*, ACM Press, New York, 1996, pp. 289-297.
6. R. Liere and P. Tadepelli, "Active Learning with Committees for Text Categorization," *Proc. 14th Conf. Am. Assoc. Artificial Intelligence*, AAAI Press, Menlo Park, Calif., 1997, pp. 591-596.
7. Y.H. Li and A.K. Jain, "Classification of Text Documents," *The Computer Journal*, vol. 41, no. 8, 1998, pp. 537-546.
8. F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, vol. 34, no. 1, 2002, pp. 1-47.
9. P. Constantopoulos and M. Doerr, *The Semantic Index System: A Brief Presentation*, Inst. Computer Science Foundation for Research and Technology-Hellas, Heraklion, Crete, 1994.
10. A. Sheth, D. Avant, and C. Bertram, "System and Method for Creating Semantic Web and Its Applications in Browsing, Searching, Profiling, Personalization and Advertisement," U.S. Patent #6,311,194, 30 Oct. 2001.
11. D. Fensel and M. Musen, "The Semantic Web: A Brain for Humankind," *IEEE Intelligent Systems*, vol. 16, no. 2, Mar./Apr. 2001, pp. 24-25.
12. A. Sheth, "Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics," *Interoperating Geographic Information Systems*, M.F. Goodchild et al., eds, Kluwer Academic Publishers, Dordrecht, Netherlands, 1998, pp. 5-30.

Amit Sheth is professor of computer science at the University of Georgia, where he also directs the Large Scale Distributed Information Systems (LSDIS) Lab. He founded Taalee in 1999 and managed it as its chairman and CEO, and has also been CTO of Voquette, which acquired Taalee in 2001. His research interests include semantic Web and semantic interoperability, rich media content management, workflow and collaboration systems, and semantic applications in financial, national security, and healthcare. He received his BE from B.I.T.S., Pilani, India, and MS and PhD from Ohio State University.

Clemens Bertram is the director of engineering at Voquette. As research assistant with Amit Sheth at the University of Georgia, Clemens implemented the first prototype of a video data server called "VideoAnywhere," which later evolved into SCORE's patented metadata extraction technology. After a short interlude at Video Networks, he joined Taalee as cofounder and lead engineer of SCORE. His technical interests are in the areas of metadata, Java, and XML technologies.

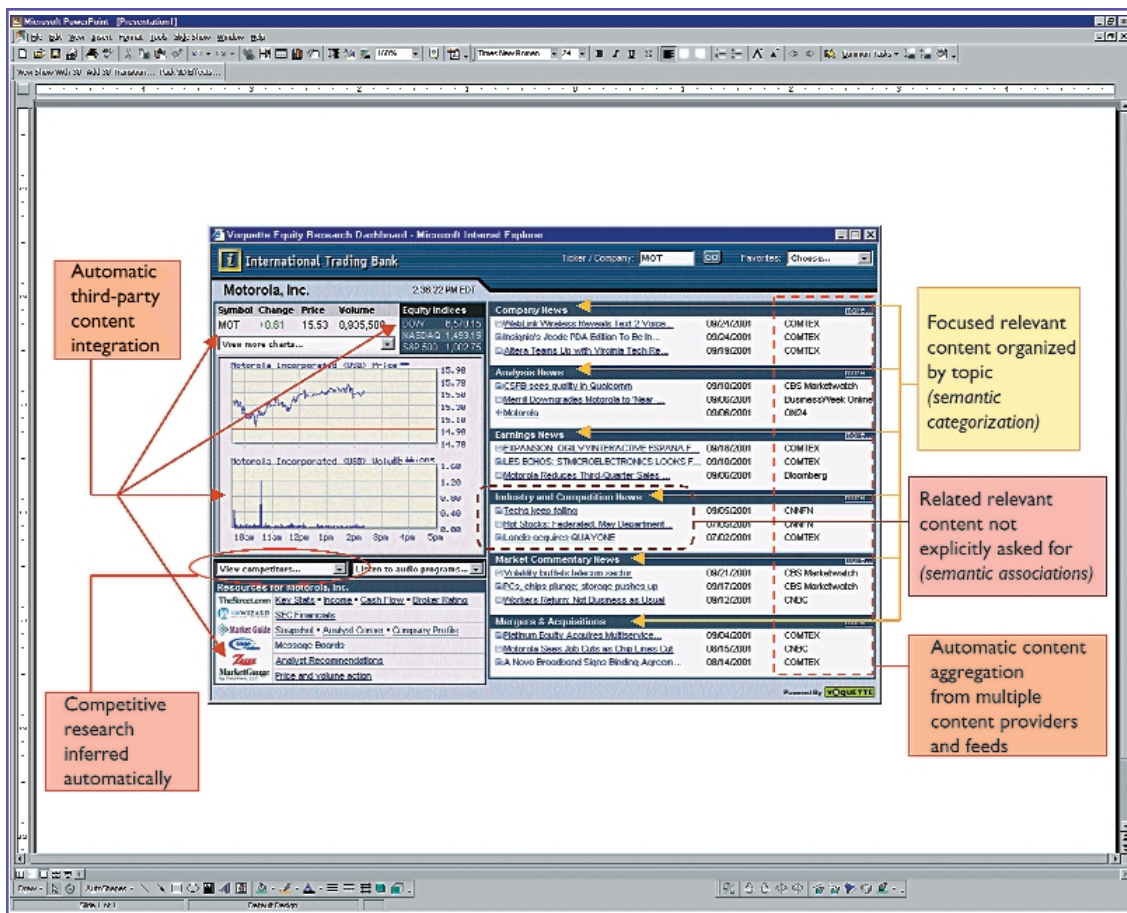


Figure 4. The Analyst Workbench is targeted at users such as financial advisors, who profit from a tool that provides a complete picture of their research area. Its key features include focused relevant content organized by topic (semantic categorization), related relevant content not explicitly asked for (semantic associations), automatic content aggregation of heterogeneous formats and media from multiple content providers and feeds, automatic third-party content integration, and links to research about automatically inferred competitors.

David Avant is a senior engineer at Voquette, responsible for the development of back-end knowledge creation tools. He, along with Amit Sheth and Clemens Bertram, authored a U.S. patent on a system for semantic metadata extraction and search. He is currently completing a master's thesis on the topic of knowledge discovery at the University of Georgia.

Brian Hammond has been instrumental in the design, implementation and deployment of semantic search engine, semantic directory, automatic classification, and metadata-extraction system at Voquette.

Krzysztof J. Kochut is a professor in the computer science at the University of Georgia. He received his PhD in computer science from Louisiana State University. His research interests include distributed systems (especially workflow systems), database systems, and bioinformatics.

Yashodhan Warke is the director of product development and marketing at Voquette. He has an MBA and MS in com-

puter science. His main interests are in the broad areas of knowledge and content management, information retrieval and analysis, along with the underlying technologies that support work in these areas.

Readers can contact the authors at {amits, clemensb, davida, brianh, kochut, yashw}@voquette.com.

Would you like to write for Spotlight?

Spotlight focuses on emerging technologies, or new aspects of existing technologies that will provide the software platforms for Internet applications. Spotlight articles will describe technologies from the perspective of a developer of advanced web-based applications. Previous articles have covered RDF, XML, and Bluetooth.

All candidates, please contact department editor Frank Maurer at maurer@cpsc.ucalgary.ca