Next-Wave Publishing, Part 3: Revolutions in Content

By Mills Davis

The conclusion to our research into next-generation publishing looks ahead to the future, when semantics redefines the nature and value of content. We examine the potential impact semantics will have on data, computing, people, publishing, government and manufacturing.

In Part 1 of this series *(see Vol. 3, No. 15)*, we saw that next-wave technology is the result of several long cycles of innovation: computing, distributed intelligence, nanotechnology and biotechnology. We pointed out that the dominant driver of next-wave publishing technology is *distributed intelligence* rather than computerization. We outlined a framework for interpreting technology advances that distinguished five technology waves. The first three were past and present waves; the last two defined present and future waves, which we set out to explore in Part 2 and Part 3.

In Part 2 *(see Vol. 3, No. 20)*, the focus was on revolutions in process. We examined the impact that networked services and new media technologies will have on corporations and commercial publishing businesses. We explored repercussions in four areas—content strategies, media platforms, product platforms and infrastructure, and value chains—with industry examples illustrating how the broader trends play out in specific industry applications.

In Part 3, the conclusion to the series, we will turn our focus to revolutions in content. The unfolding story of semantic webs at this stage will lead to the emergence of a new kind of content—semantic-form declarative knowledge. In this stage, semantics (the meaning of something) gets encoded separately from content. Ultimately, semantics gets encoded separately from process.

In previous technology stages, we saw that digitization of content and the separation of data and process representations from software applications created new economic value in the form of new tools and product categories, new categories of output and new markets, resulting in major breakthroughs in process economics.

Semantics, it turns out, can directly encode ideas and patterns of thought—all theory, all knowledge; in fact, anything that has or can ever be thought by anyone. The arrival of semantic computing heralds the dawn of the knowledge age, in which truly new vistas open for publishing, information technology and manufacturing. The direction is toward systems that know, learn and can reason the way humans do.

In the following pages, we explore what these revolutions in content are and what they'll mean for us. We'll start by asking:

- What is semantics all about?
- What is the significance of semantics technology for:
  - Content?
  - Computing?
  - People?
  - Publishing?
  - Government?
  - Manufacturing?

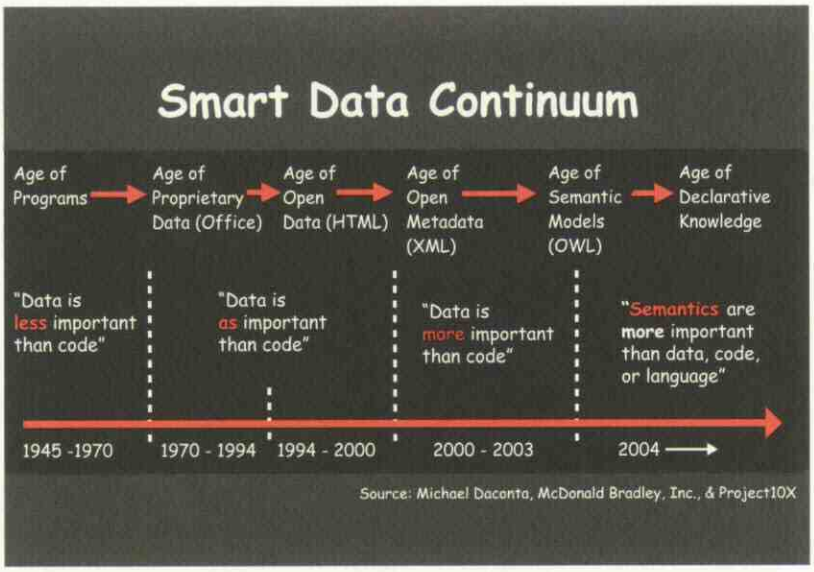
Semantics and Smart Data

What is semantics all about? Semantics is defined as the meaning of something. In a computer, what exists is what can be represented. So, digital semantics is a kind of content. But more properly, semantics is a kind of knowledge, because the meaning of anything is something that we know about that thing and represent separately from it in the computer.

When we talk about the semantics of content, we're referring to something we know about the content. Similarly, if we talk about the semantics of processes, we're referring to something that we know about a process. The trend, as we know more about something, is to represent this knowledge digitally as data, rather than as program code. Of course, it didn't start this way *(see Figure 1)*.

Evolution of data. The evolution of digital semantics started with the "age of programs." At that stage, data was simply embedded in applications and used locally. Data was less important than the application code.

The next two stages were the ages of proprietary and open (based on HTML) data exchange. Here data became separately managed and widely shared. Data became just as important as the programming code.



Source: Michael Daconta, McDonald Bradley, Inc., & Project10X

Figure 1: Smart Data Continuum. The illustration depicts the evolution of data. The basic direction is that data gets smarter, but the evolution happens in stages.

With adoption of XML for metadata, followed by RDF for the semantic web, we entered the age of open metadata exchange between systems and across networks. Metadata structures embed information and abstract ideas within recognized linguistic, graphical or symbolic forms (such as schemas). Metadata structures convey accepted patterns of meaning and well-understood relationships, which provide a needed component of "theory" that helps integrate data, content and processes.

Now, with the release of the Web Ontology Language (OWL), we are entering the age of semantic models. The key advance in this era will be that semantic models employ formal "named or coded relationships" to spell out and make explicit the "theories" left implicit within a metadata structure. Plus, behind the very plastic definition of a relationship, there now exists some underlying axiom or theory that explains and constrains how one related concept affects the other. This formality enables mechanization of reasoning, which is why semantic models improve information search, simplify integration, enable advisory services, automate custom communications and facilitate interoperability across systems and repositories. Semantically modeled metadata is now more important than the program code.

The progression does not stop with deployment of language-based ontologies. Next, a wholly new species of content emerges, and we will enter the age of declarative knowledge. At this stage, it becomes widely recognized that the only way to gain the benefits of precise meanings is to move away from natural language toward pure semantic codes and relationships. Semantics moves to center stage and becomes more important than either data or program code or, for that matter, natural language. This shift leads to new categories of products and services that open multi-billion dollar markets in publishing, IT and manufacturing industries.

Semantics for Content: Information Management, Libraries and Research

Information managers, librarians and researchers have a long history of experience with managing digital and physical collections. Library science goes back thousands of years and continues to evolve. While the environment and economics of content management are changing, the key concerns of librarians and information managers have remained pretty constant.

Key questions that these professionals are asking include:

- How can I automate the process of managing the repository?
- How do I classify, index and organize my collection?
- How can I improve search with digital technology—make it faster, more productive and less time-consuming for researchers?
- How can researchers find what is in the repository (and relevant to their questions) without having to look at it all, read it all or listen to it all?
- Can I have self-organizing repositories?
- How can I navigate across multiple repositories that have been indexed by different communities?

Information managers and librarians have been trying a broad range of knowledge representation techniques in order to resolve language ambiguities and improve the quality of content searches. To illustrate, we recently came across a "Dictionary of Search Terminology" (www.topquadrant.com) that discusses 70 categories of digital search technology. All of these are in use.

An example, from the NSF Digital Library Initiative, is the bio-science, medical and health-care repository being developed by the CANIS center at the University of Illinois. A sample question might be: For a patient with rheumatoid arthritis, what is a drug that reduces the pain but does not cause stomach bleeding? According to Dr. Bruce Schatz, finding the answer requires navigating across hundreds of millions (currently, 250 million) of concepts within millions of documents stored in multiple repositories, each of which has been indexed by a different community, but nevertheless can be accessed using the searcher's own special vocabulary.

From search to knowing

Semantics is strategic for information management, libraries and research for two important reasons. First, semantics is the key to better information search. Better search is worth millions of dollars. Second, semantics enables applications that know. This is what happens when the entire library is seen as one extended book, with all of its concepts, theories and factual matters present, unambiguously encoded and organized in a way that enables exploration of every question as well as every relevant path of reasoning. A researcher could not only navigate concept across repositories to locate sources, she or he would be able to reason and simulate directly across all knowledge contained in these repositories.

Semantics trumps linguistics

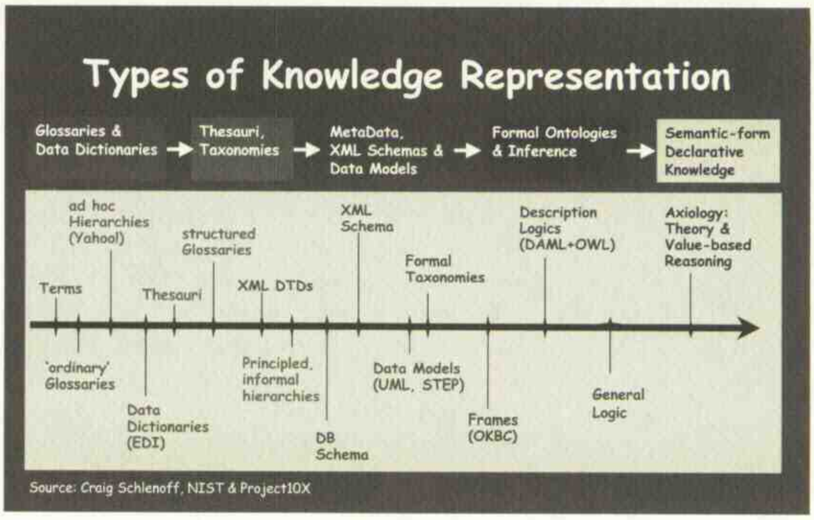
Semantic-stage applications always start with some sort of knowledge representation. An application reasons across this knowledge. The question is, what sort of knowledge and representation are we talking about?

Figure 2, "Types of Knowledge Representation," identifies 15 forms of knowledge representation arranged in five categories along an axis of increasing knowledge value. As we review the capabilities of different categories of knowledge representation, we'll also identify some limitations as well. A key theme is that language-based approaches suffer from ambiguities inherent in natural language. We contrast language-based knowledge representation with semantic-form declarative knowledge and conclude that *semantics trumps linguistics*.

Glossaries and data dictionaries. The simplest forms of knowledge representation are glossaries and data dictionaries. Lexicons collect terms used in information systems. Dictionaries and glossaries define terms in natural language, and can be used to define keyword searches.

Thesauri and taxonomies. A *thesaurus* identifies synonyms, related and contrasting words, and antonyms. *Taxonomies* categorize, abstract and classify terms. Metamodels of knowledge frequently employ taxonomies as a hierarchical arrangement (tree structure) for all abstract concepts. These trees may focus at any given level to show how concepts are exclusively different. Models at the base of taxonomies are very detailed and specific. Progressing up the tree, models become progressively more abstract. To abstract is to ignore certain distinctions in order to focus selectively on certain other commonalities. So, moving up the tree, each category subsumes or includes the broadest, most general characteristics of those below.

Typically, no two groups will make the same abstract choices as they move up or down. Different users have different social roles and needs. So, they produce multiple taxonomies (or mappings of concepts). For example, if there is one catalog (such as a collection in a library), then there are just as likely a dozen or more taxonomies depending on the usages and purposes of the taxonomies' users. Still, taxonomies allow concept-based search and guided navigation as an alternative to search.



Source: Craig Schlenoff, NIST & Project10X

Figure 2: Types of Knowledge Representation. This diagram arranges 15 forms of knowledge representation in five categories along an axis of increasing knowledge value. Semantic-stage applications always start with some sort of knowledge representation. An application reasons across this knowledge.

Taxonomies have proven ROI in a number of areas. For example:

- Knowledge management—Within the organization, capturing and sharing institutional knowledge, such as best practices and corporate intelligence.
- Customer service—Reducing the time it takes customers to find answers in technical documentation, troubleshooting guides, how-to's, FAQs and best practices, organized product by product.
- E-commerce—Customers can't buy what they can't find, so businesses must support different modes of shopping (*e.g.*, classic "men's vs. women's" methods); organize materials according to marketing needs; offer technical details of different items; and merchandize related items.

Metadata, XML schemas and data models. Metadata, XML schemas and data models make up the next group. Knowledge is represented in the form of an XML DTD, XML schema, RDF, database schema or data model (*à la* UML or STEP).

Metadata here are still definitional data that provide information about or documentation of other data managed within an application or formal computer language environment.

A model describes how concepts and phenomena are similar and how they differ.

- An *object* model is a networked data-set tightly bound to the procedures that directly access or update it. Each describes its own modular world inside a system. Base classes behave like abstractions high in a taxonomy, expressing those properties that are shared by the many more-specialized systems derived from it.

- A *data* model describes a world outside the system. Several applications can share a database. In a relational DBMS, each table in the schema models an abstract concept dictating every distinction its collection of concept-instance records are prepared to save—some common, others different. Another table sets forth its one common idea and fixed list of shared or different property values. Typically, relationships are defined only by the records they connect, so each instance has no unique identity, property or justification in theory.

> *The problem we still have to solve is the ambiguity of language used in creating DTDs or schemas.*

XML is one formal language used for defining collections of metadata to be exchanged between applications over the Internet. XML has given us syntax interoperability.

Given that the XML grammar is fixed, the problem we still have to solve is the ambiguity of language used in creating the DTD or schema. XML Namespaces allow groups to register vocabularies used in tags. This is a step toward interoperability, but names alone don't supply the knowledge needed to resolve many ambiguities. Does the label (or tag) infer one or many possible semantic concepts? No one knows or can know. Another party can plausibly take your words and then describe a very different semantic definition than the one you chose. That is the fundamental and inescapable burden of using language to define semantics.

Resource Description Framework (RDF) and RDF Schema provides a model based on XML syntax to represent and transport metadata. RDF Schema is an extension of RDF that provides mechanisms for describing groups of related resources and the relationships between them. RDF integrates a variety of applications: library catalogs and worldwide directories; syndication and aggregation of news, software and content; personal collections of music, photos and events, and so forth.

RDF gives us an Internet with two-way named hyperlinks and thus a way to expose metadata. However, in schema architecture, relationships are where you put them. What you call them has few constraints. In a "two-way, named hyperlink," then, the only relationship naming concern is that the name previews what you are likely to see first, given that you take that link. If so, intersite negotiations will favor a minimal set of abstract relationship types, because one-time users are not going to spend much time learning some new subtlety roaming through yet another site. They only need enough to take or not take the branch offered. Abstract schemas present minimal constraints—*i.e.,* offer little knowledge value.

Formal ontologies and inference. Formal ontologies and inference make up the next level of knowledge representation. These include formal taxonomies, topic maps, frames, description logics and general logic.

As the semantic model becomes richer, it more completely specifies not only the formal class-subclass relationships, but also relationships between concepts, and the descriptive logic and conditional assertions that are used to perform inference.

Topic maps are a method of using XML to represent networks of concepts to be superimposed on content resources such as documents of various types, providing a means to represent, navigate and query the topic-map network itself, rather than the full text of a document collection. Topic maps support both hierarchies of concepts (topics) and relationships between them (associations).

An *ontology* is an explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. In computer systems, what "exists" is that which can be represented. Ontologies organize concepts and their interrelationships in ways that facilitate machine reasoning and inference.

The Web Ontology Language (OWL) is a semantic markup language for Web resources and ontology construction that has just been developed and approved by W3C. It is based on RDF and extends the work of DAML and OIL languages for defining ontologies. The OWL language has three levels, with progressively more expressiveness and inferencing power. OWL will be used as a semantic markup for Web pages, data sharing and Web services. The goal is to have resources, repositories and processes semantically related and interoperable through ontologies.

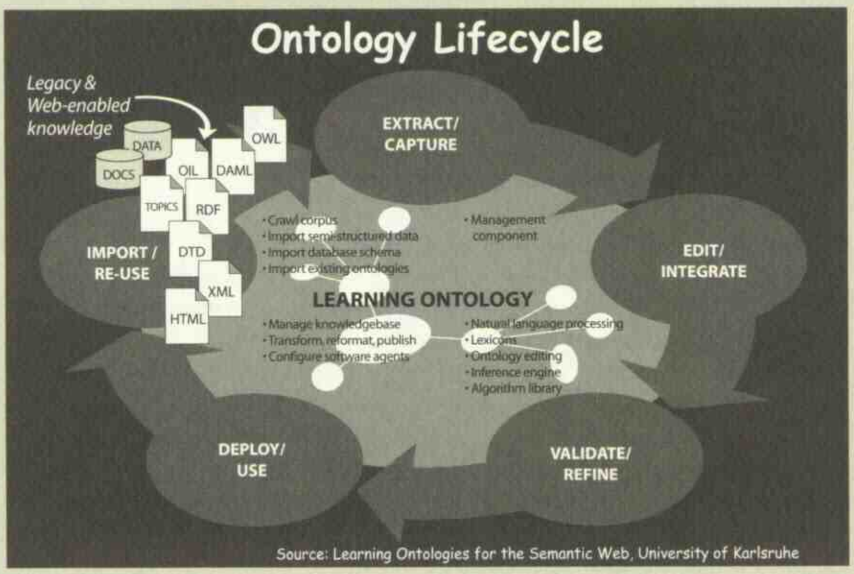
Ontology-based solutions are already having economic impact. For example, global publisher Bertelsmann owns Empolis, a semantic-technology supplier. Empolis has developed an ontology-based, self-help and customer-service application for a division of Siemens, which produces a wide array of control and monitoring devices that are sold all around the world. This application, distributed to Siemens service personnel (who are not necessarily Siemens employees) on-site for use with 65,000 users worldwide, is helping Siemens save $3 million a year.

Figure 3, "Ontology Life Cycle," presents an overview of a computer-aided process for constructing ontologies. Companies providing software for building ontologies include: CognIT, Empolis, Intelligent Views, Language and Computing, Lockheed Martin, Modulant, Network Inference, Ontoprise, Plugged In

Figure 3: Ontology Life Cycle

This diagram outlines the process for constructing ontologies. The stages include:

(1) *Import and reuse* legacy and Web-enabled knowledge sources. It is possible to "crawl" the corpus to identify and download sources in a variety of formats as well as to mine text to extract terminology from documents. Tools should allow import of legacy forms such as database schemas, product catalogs, yellow pages listings, semi-structured data sources and Web-enabled data. Also, it is possible to reuse ontologies, in whole or in part, that have already been developed.



Source: Learning Ontologies for the Semantic Web, University of Karlsruhe

(2) *Extract and capture.* This entails format conversion, analysis of source material, and creation of class structures and subject indices.

(3) *Edit and integrate.* Conceptual analysis is done to determine what an information object is "about" and to establish a knowledge-organization scheme that includes subject index, thesauri, classifications, etc. This can be approached manually, semiautomatically or automatically. Automated indexing ranges from simple natural-language processing (recognize parts of speech and words in a document) to sophisticated analyses that identify key names, words and phrases. Automated classification assigns documents to categories or classes.

(4) *Validate and refine* the ontology. Detect and correct errors and omissions; respond to changes in the environment. To validate the ontology, critique the formal semantics of what the class structure means. Description logics provide frameworks that support this sort of reasoning. Some tools employ automated description-logic engines to determine if an ontology has contradictions or gaps in the knowledge representation, or to detect when a particular concept can be classified differently, according to its description and that of other concepts. These critiques can be used to modify the ontology automatically, consolidating information contained within it.

When maintenance requires merging ontologies from diverse provenance, some tools provide human-centered capabilities for searching different ontologies for similar concepts (usually by name) and merging the concepts. Other tools perform more elaborate matching, based on common instances or patterns of related concepts.

(5) *Deploy and use* the ontology in any of several ways. An ontology provides a natural index of the concept instances described in it, and thus serves as a navigational aid for browsing contents. An ontology can drive similarity measures for case-based retrieval and reasoning. Web-ontology languages (such as DAML and OWL) have capabilities for expressing axioms and constraints on concepts that enable powerful reasoning engines to draw conclusions about instances in the ontology. Probably the fastest-growing area of development for ontology-based systems is semantic integration across various applications and content repositories. **TSR**

Software, Sandpiper Software, Semagix and Unicorn Solutions.

To date, most ontology work is driven by the assumption that knowledge search is the goal, and so the primary interest is in amassing and searching the set of all independent concepts (subject indicators) in the ontology.

A key goal of language-based knowledge representation is to eliminate the ambiguity of describing things with labels and natural language, leading to improved search and easier integration of content and processes. But, as we have seen, this goal is difficult to achieve when we use language to describe what we mean. Natural language use is inherently ambiguous. Many words have multiple meanings. There is no way to guarantee that two occurrences of the same word have the same meaning.

A standard ontology with just one noun-phrase selected for every category is an improvement. But, what else do you (must you) offer in evidence to guarantee the unambiguous meaning for this particular noun-phrase? Ask everyone to give their form, defining each concept. Do they have text definitions? Do they offer a series of sentences that connect different word concepts to one another? Can they use those sentence structures to define relationships? How do they define the relationships so that they are rare and effective discriminants between meanings? How many other concepts use these same relationships?

Ultimately, the only way to ensure precise meanings is to move away from natural language toward pure semantic codes and relationships; that is, use unique identifiers to identify concepts. (We may draw an analogy here with the UPC [universal product code] identifier that has no significance other than that it is unique.) Do not use labels or names of things. Rather, determine meaning by the sum of all the relationships the concept has.

Semantic-form declarative knowledge. Semantic-form declarative knowledge is the next form of knowledge representation. It is based on theories of knowledge and computation that define an emerging science of representation.[1] Here, conceptualism and semantics replace nominalism and language to solve seemingly unsolvable problems of great complexity and ambiguity. For example, model and instance codes are the same for all languages. A concept (model-instance) appears only once in any semantic web; the codes identifying it also locate it instantly without search. Meaning is defined only by the unique web of relationships tying every concept to those connecting to it. Properly implemented, semantic-form declarative knowledge webs approach absolute limits on physical size, speed, and efficiency.

A central goal of declarative knowledge is the development of a standard scientific representation for "all knowledge"—one that can be shared jointly by humans and machines—a common, evolving, most-fundamental representation seeking to encompass all science and all learning. A unique, scientifically precise and minimal representation for encoding every idea known and imaginable then provides the basis for a language-independent, standard, global ontology (initially hundreds of millions of "concept models").

Machine theories and computation will model human theories and reasoning. All things can be represented in semantic form. Semantics can directly encode ideas and patterns of thought—all theory, all knowledge; in fact, anything that has been or can ever be thought by anyone.

According to Dr. Richard Ballard (rlballard@earthlink.net), the scientific foundations for semantic-form declarative knowledge are made quantitative by the following axioms:

- Knowledge is defined by an unanswered question or that set of questions defining some field.
- Knowledge becomes measurable when one can define one or more "expected" or "acceptable" answer forms.
- The number of acceptable forms defines the initial answer uncertainty, or problem size, describable in bits.
- Knowledge is anything—influential theory or observable fact—that decreases answer uncertainty.
- Knowledge is measured by the amount that its possession (theory) or receipt (information) reduces uncertainty.

Knowledge is then defined as a learned and stored system of constraints that tell us why particular answers or classes of answers should be discarded. These constraints do not assure us initially that there is just one or any particular "right answer." They only narrow or eliminate possibilities. We deal with the remainder as we choose. Knowledge does not solve problems. It only helps us to predict, anticipate and manage their consequences.

Semantic-form declarative knowledge is composed of theory and information. Theory is a "metaphysical" constraint that asserts something thought absolute (this always/never happens) or conditional (this happens when/if/while/after some condition exists). Information is a "physical reality" constraint (observably—this situation is happening or has happened).

Theory and information constraints are absolutely and fundamentally different. To have influence, theory has to be learned and known before (often long before) an event. By contrast, information can only be known during and after an event. A "new" theory is likely to be 30–50 years old; most go back 3,000 to 40,000 years, and some are more than 2 million years old. Information looses value continually as situation awareness fades in minutes, days or years, becoming past history—recorded or not.

The expectation of semantic-form knowledge representation is that ontologies will be massive, rather than small and handcrafted. They will top 100–500 million concepts within five to ten years. Ontologies will be stable, slow evolving and, subject independent, and will encompass all knowledge codes.

Professions, publishers and governments will lead ontology codification. Knowledge content is a capital asset. Knowledge ownership and practice will define an organization's value.

Semantics for Computing

The information technology (IT) community has a 60-year history with computing that has evolved from vacuum tubes to networked services. Whether we approach the matter from the standpoint of consumer electronics companies, telecommunications providers, hardware manufacturers, software publishers or enter-

---

[1] The semantic-form representation of knowledge described here is based on "Physical Theory of Knowledge and Computing," by Richard Ballard, which expands the Mathematical Theory of Communication developed by Claude Shannon to provide limit-case bit-measures for the constraints of theory and information and a quantitative definition and measure of knowledge.

According to Ballard, semantic representational form should combine: (a) the concept-relationship formalism of Porphyry (ca. third century AD) with (b) structuralist semantic definitions (Ferdinand de Saussure) and (c) mediating structures (John F Sowa/Charles Sanders Peirce Synthesis) expanded to (d) higher logical order empirically by use of n-ary information-limit relationships.

Relevant reading includes: Claude E. Shannon and Warren Weaver, The Mathematical Theory of Communications (University of Illinois Press, 1949); John Sowa, "Ontologies: Lattice of Categories," in Sowa and Dietz, Knowledge Representation: Logical, Philosophical, and Computational Foundations (Brooks/Cole, 2000); and Richard Ballard and Robert Smith, "On the Evolution of a Commercial Ontology and Coding System" (Knowledge Foundations, 2001).

prise IT departments, semantics are now essential to make IT work.

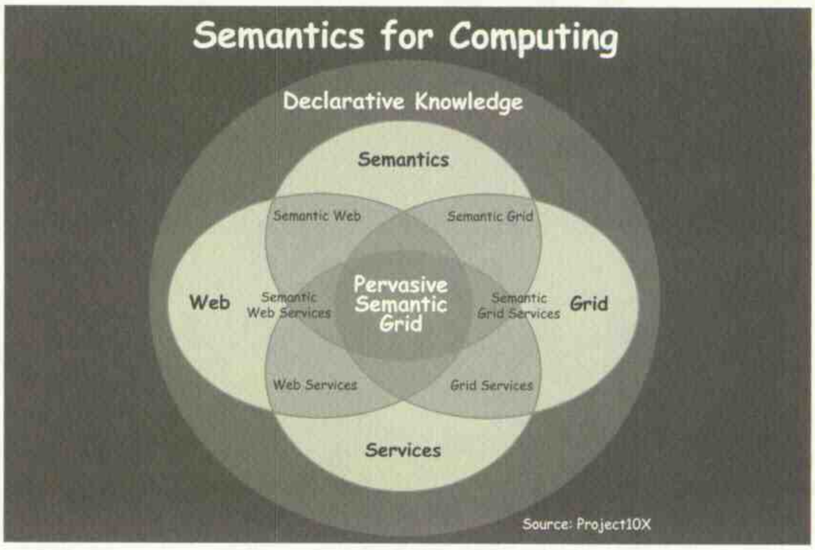
The fundamental issues for IT today are managing complexity and uncertainty. Key challenges, such as security, pervasive services, stack complexity, autonomic systems and legacy conversion, demand solutions designed for the era of distributed intelligence, not for the desktop or the client-server world. On a global scale, market dominance will be worth trillions of dollars to the group of companies that can develop and deploy the winning architecture.

We're not making this up. IBM, Sun Microsystems, Microsoft, Oracle and other major vendors in the IT space have all focused on network services and semantic technologies as the way to fundamentally improve the economics of information technology, for both solution providers and their customers. The core argument has been that the cost to integrate, manage and maintain the expanding array of (component) systems and processes has become unsustainable. They believe the "grand challenge" for the industry to be developing systems and processes that are self-declaring, self-integrating, self-optimizing, self-protecting and self-healing. They give this vision different names, such as "autonomic computing" or "the net-effect," but they concur that you must have semantics to implement this vision. It's not an option.

We need semantics to make the World Wide Web work for computers as well as people. We need shared meanings to make Web services work between applications and organizations. It's the only effective way to reduce the time and cost required to integrate our processes. We need semantics to make grid computing work for large numbers of dynamic projects and distributed resources. Nothing scales without it. Likewise, we need semantics to cope with the swarms of devices, services and resources that will be orchestrated through the pervasive service grid. There's no other way to pull it off. And finally, we need semantics if we're ever going to develop systems that can self-configure, self-integrate, self-deploy, self-optimize and self-repair. The world of autonomic computing is unreachable without semantics.

Knowledge trumps programming

When it comes to semantics for computing, what kind of semantics do we need? The semantics of processes are represented separately from the application code and the content. But is this going to be enough? In some cases, the answer is yes. However, the argument we are making is that language-based approaches to semantics may max out at some point. They may not scale to handle the diversity of communities, the size and weight of the "standards" stack, the security requirements, etc. The IT community may find that it needs to examine assumptions and take a fresh look. In the following discussion, we point out two new directions that appear promising: context computing and declarative knowledge computing. Our take is this: *Knowledge trumps programming.*



Source: Project10X

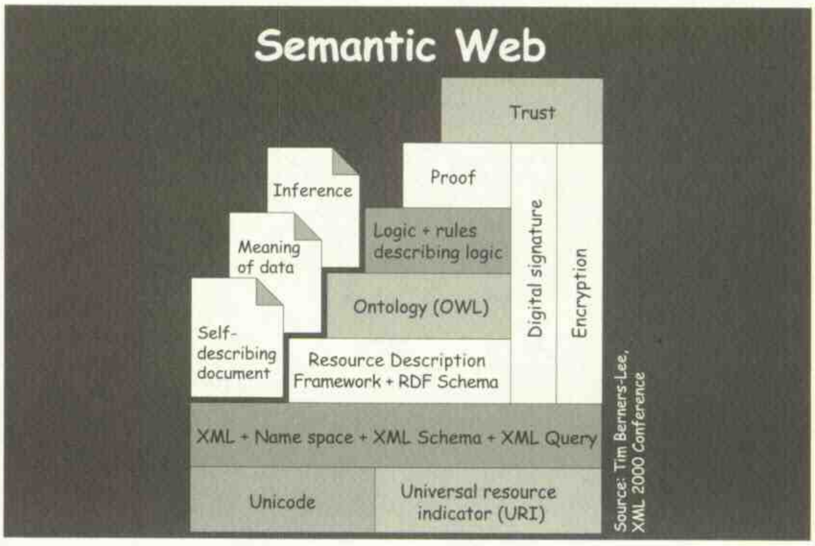
Figure 4: Semantics for Computing. This diagram depicts major lines of development for semantic technology in computing. Semantics for information processing is converging on the pervasive service grid as its new paradigm. This is the intersection of four major technology themes: semantics, the Web, grid computing and services. Surrounding and subsuming this is another emerging knowledge-technology paradigm, which we call declarative-knowledge computing.

Semantic tsunami

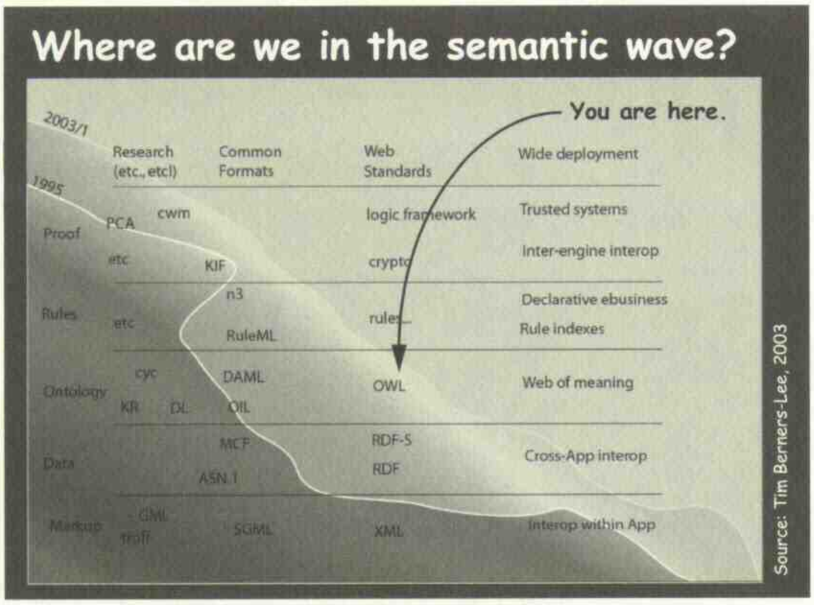
As shown in Figure 4, the major lines of semantic technology development for computing can be visualized as the intersection of four technology themes: the worldwide Web, grid computing, services and semantics, leading to the pervasive semantic grid as the emerging inner paradigm of information computing. Surrounding and subsuming this, we are suggesting, is another emerging technology paradigm: declarative-knowledge computing.

Semantic web. The *semantic web* (*see Figure 5*) envisions a transition from simple HTML linkages to machine-interpretable tagged relationships among

Figure 5: Semantic Web. This diagram depicts the arrangement of specifications (called a stack) that define the semantic web.



Source: Tim Berners-Lee, XML 2000 Conference



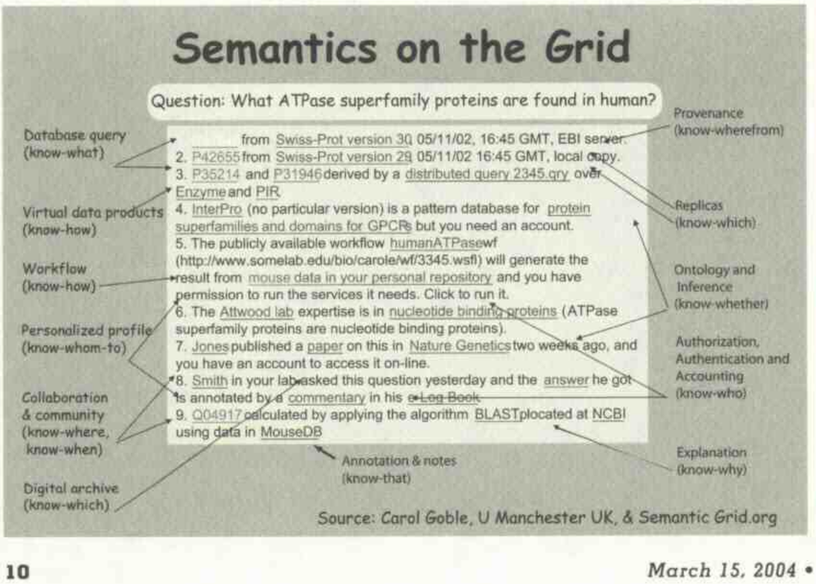
Source: Tim Berners-Lee, 2003

Figure 6: Where Are We in the Semantic Wave? This chart uses the metaphor of high watermark of an advancing wave along a beach front to describe the progression of semantic-web technologies from research to common formats, to web standards and to wide deployment.

resources. The goal: a web of globally linked, semantically related and distributed resources. A semantic web, as envisioned by Tim Berners-Lee, is a web of machine-processable data.

The semantic web is "crossing the chasm" now (*see Figure 6*). We'll see the tipping point within three years. Businesses will see it in portals. Consumers will see it in the integration of e-mail, calendar and contact lists with personal knowledge-bases (music, video, vacation, etc.).

Semantic-web services. Semantic-web services, employing shared semantics, enable disparate systems to discover a Web service and understand what it does, how it works and how to access it. Without semantics, who knows what the service provider meant?



Question: What ATPase superfamily proteins are found in human?

1. from Swiss-Prot version 30 05/11/02, 16:45 GMT, EBI server.
2. P42655 from Swiss-Prot version 29 05/11/02 16:45 GMT, local copy.
3. P35214 and P31946 derived by a distributed query 2345.qry over Enzyme and PIR.
4. InterPro (no particular version) is a pattern database for protein superfamilies and domains for GPCRs but you need an account.
5. The publicly available workflow humanATPasewf (http://www.somelab.edu/bio/carole/wf/3345.wsfl) will generate the result from mouse data in your personal repository and you have permission to run the services it needs. Click to run it.
6. The Attwood lab expertise is in nucleotide binding proteins (ATPase superfamily proteins are nucleotide binding proteins).
7. Jones published a paper on this in Nature Genetics two weeks ago, and you have an account to access it on-line.
8. Smith in your lab asked this question yesterday and the answer he got is annotated by a commentary in his e-Log Book.
9. Q04917 calculated by applying the algorithm BLASTp located at NCBI using data in MouseDB

Source: Carol Goble, U Manchester UK, & Semantic Grid.org

Figure 7: Semantics on the Grid. This diagram provides an illustration of the types of application knowledge that are required to support e-science projects across the semantic grid.

Semantic-grid services. Semantic-grid services (see www.semanticgrid.org) envision the use of shared semantics to facilitate multi-participant dynamic specification, allocation and persistent management of distributed computational resources. The original motivation for grid computing was the orchestration of distributed computing resources.

The motivation for semantics in the grid *(see Figure 7)* comes from recognition of the need to support collaborative projects and virtual organizations from diverse provenance, as well as the need to access knowledge developed and indexed by diverse groups, using different languages and methods in different regions of the world.

Pervasive computing. Pervasive, or ubiquitous, computing envisions an environment where devices that compute and communicate are everywhere, for example, the environment, clothing, everyday artifacts, sensor arrays, etc. Pervasive computing and semantic-grid services face similar issues. They are large-scale, massively peer-to-peer distributed systems. They must provide for service description, discovery and composition. They must solve issues of availability and mobility.

Pervasive service grid. The center of the diagram in Figure 4 shows that these lines of Internet development are converging on a new information-computing paradigm called the pervasive service grid. It is the manifestation of the semantic grid in the physical world through pervasive computing. This is also called ambient intelligence.

Context computing

Context computing is what will enable the pervasive service grid to function. Context might be thought of as the next level of semantic-web stack. It treats everything as nodes, spaces and relationships. These are multidimensional. The sum of these dimensions is called a context. In the semantic web, a model of a context is an ontology that integrates web, grid and mobility services, and unifies both content and procedural language forms across all layers of abstraction, right down to the bits, in a way that scales across swarms of devices and resources in a network, enabling them to combine and interact securely and efficiently.

According to Sandy Klausner, (klausner@coretalk.net), a leading proponent of context computing, the central aim is to integrate all data and processing languages and usage into a common ontology-based representation that spans all levels of the stack, all memory and all nodes of the network, and processes across all (peer) nodes as unified (symbols to bits) binary. In short, context computing attempts a grand synthesis that integrates and unifies all of the interior layers of the IT process paradigm. The advantages of context computing include scalability (which is needed to handle pervasive computing), immunity to viruses and security attacks once within context, built-in digital rights management providing a business model even for open-source components, and extreme performance across diverse network nodes.

One area where context computing is already receiving attention is national defense. According to a recent white paper by Michael Daconta (McDonald Bradley, Reston, VA) entitled "Semantic Web Foundations of Net-Centric Warfare," a key aim of context computing is the ability to model the context of content, services and transaction message spaces from multiple user perspectives. This enables synchronization of these contexts with each other, which is a key enabler for the new kinds of high-mobility, edge-powered, rapid-deployment combat operations where force superiority depends on better knowledge, high coordination and faster sense-response.

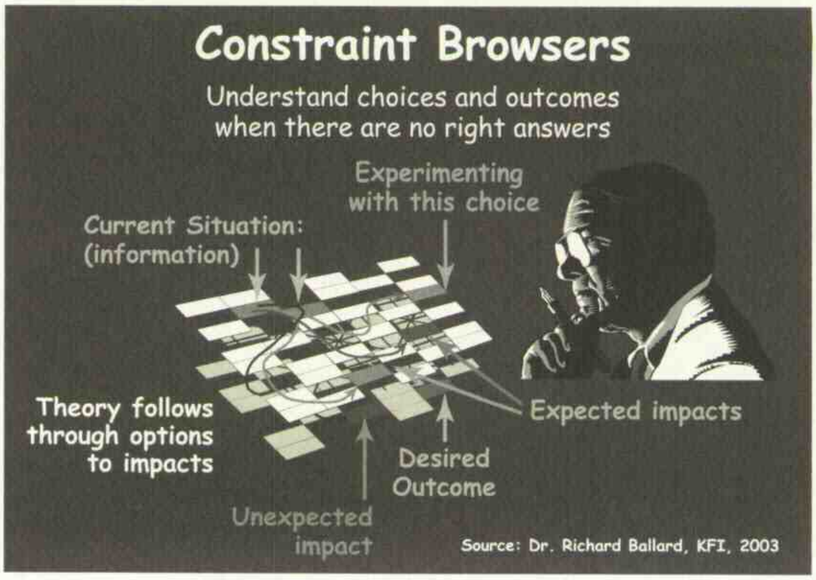
Knowledge computing

Referring back to Figure 4, the surrounding circle shows that semantic technologies are simultaneously evolving in another, profoundly different direction, which we call declarative-knowledge computing. The goal of knowledge computing is systems that know, learn and reason the way humans do—not just with logical consistency and determinism, but also with all value systems and patterns of thinking.

In declarative knowledge, all theory and information is present and expressed as pure, extra-linguistic semantic crystals. There are no labels. Meanings are structural, the sum of relationships with other concept instances. Declarative forms require no "execution." All paths, from any and all inputs to any and all outputs, are always already present. Reasoning, then, is not restricted to executing hierarchies of deterministic procedures. There is no search. Answers to questions are paths that are defined and constrained by theory and information (facts, situation awareness). It's ironic to think that the most practical solution to problems of process complexity and language ambiguity may be to abandon both in favor of declarative knowledge.

One way to think about declarative computing is to imagine a computer program (a coded algorithm), plus all of its possible inputs and all of its possible outputs, arranged so that all of its possible execution paths are already taken and recorded in the form of a knowledge web.

A constraint browser is a type of tool that a decision-maker might use to reason across a semantic-form declarative knowledge base *(see Figure 8)*. For example, there might arise a question about Sarbanes-Oxley compliance, and if a decision-maker had established that for one division the company, if one knew the capabilities of people—what they were doing, the flow of work, the functions of systems used, the separation of controls and the monitoring and testing being applied—then an answer about the extent of compliance, including material weakness of controls, could be given. Further, if the decision-maker had established that when these informational constraints were combined with legal and regulatory requirements, standards, industry benchmarks and current case law, then one could determine the nature and extent of risks, possible consequences and trade-offs that the company was facing.



Source: Dr. Richard Ballard, KFI, 2003

Figure 8: Constraint Browsers. This diagram illustrates a kind of tool that a decision-maker might use to reason with semantic-form declarative knowledge.

In this example, input about the current situation is simply an informational constraint. If you know it, then theory follows through the options to the various impacts, and you know the output(s). If you want to know "how" you got to that output, just follow the path in the forward direction. Conversely, if you want to know "why" you ended up with that result, then you can follow the path in the reverse direction.

Also, you may have a bundle of different "trade-off" paths that end up at a particular expected or unexpected result. Then you can explore "what ifs" by experimenting with different choices and examining the effect of changing different information and constraints.

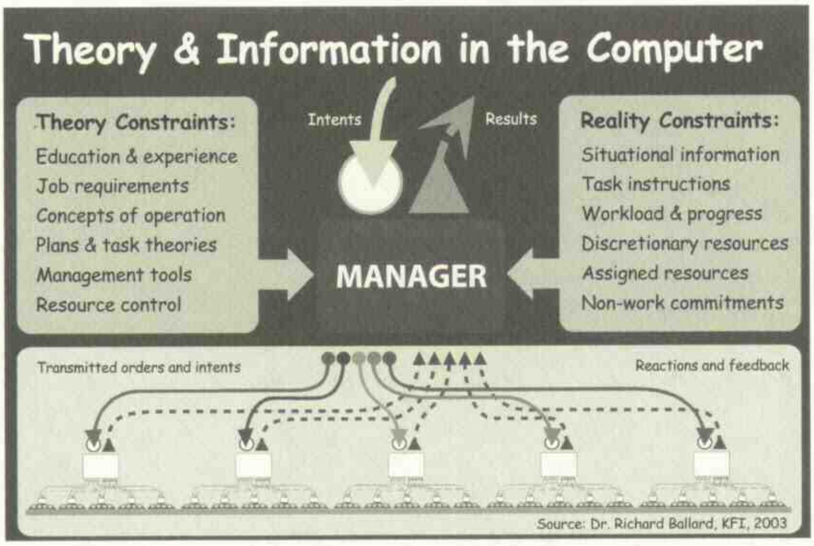
Using declarative semantic-decision tools, users walk down all branches and examine the consequences of making a decision one way and not another. Only after all of the potential decision impacts are known do users actually pick a particular branch. That is the classic "what if" methodology. As a rule of thumb, the criteria or theory that they say influenced them most in making the trade-off will end up being heuristic (in the user's head, but not formally in the knowledge asset). They have figured out some plausible advantage in choosing one thing.

Declarative-knowledge tools can have self-awareness of limitations of knowledge (*e.g*,. by identifying gaps in otherwise-valid reasoning patterns), and can take steps to learn. For example, having established an acceptable path of reasoning about Sarbanes-Oxley compliance, this "template" might be applied to another division. Doing so might reveal gaps in information. The decision maker might learn that personnel profiles were incomplete, or that the separation of controls was unknown. The tool can reason to determine the seriousness and consequences of incomplete information, and indicate what determination could be made if missing knowledge or facts were provided. Also, it can take directed action to seek and fill in missing information.

So we can see that declarative-knowledge computing is really quite different from simply executing a program, because an algorithm reasons in only one direction to a logically consistent output. Language-based, programmed solutions tend to be looking for some way to divide and conquer a problem. They assume that they need to model the sponsor's tasks and organization, and they assume that they need to model the sponsor's desired answer form. They assume that logic and algorithms are the only possible rational knowledge. And, they assume that the same inputs must always return the same answers and never learn or change.

Humans don't reason this way. They face complex decisions leading to trade-offs with varied consequences. They reason, not only with logical consistency, but using other value systems. Judging guilt or innocence, for example, is a much deeper decision to make than determining logical truth or falsity. Semantic-form knowledge bases make trade-offs in lives, dollars, morality, job loss, education, base closings, laws broken—that's complexity. It requires reasoning in many directions, using multiple value systems, the way humans do.

Perhaps, back when the price of memory was high and Von Neumann computers were all we had, then algorithms seemed like the smart way to go because that approach required less investment in memory. The price of procedural computing, however, is that the program only executes in one direction; it doesn't remember and it never learns.



Source: Dr. Richard Ballard, KFI, 2003

Figure 9: Theory and Information in a Computer. This diagram depicts a scenario in which management systems provide both theory and information for conducting the work of the enterprise. More than 80 percent of management decision-making relies on theory, with the rest being facts and situation awareness.

Semantics for People

Semantics is key to changing the economics of labor, including the cost of education, personnel acquisition, productivity and labor rates. There are several current strategies for improving workforce productivity and managing labor costs:

- *Mechanization* seeks to substitute capital investment in machines for labor.
- *Outsourcing* seeks to exploit differentials in labor rates and other costs among different geographies and business entities.
- *Labor transitions* (*e.g.*, from professional to paraprofessionals in law and medicine) seek to substitute less-skilled workers for higher-cost workers in certain tasks.
- *Service automation* seeks to displace labor or maximize productivity.
- *Self-service* seeks to offload labor costs to the customer or supplier.
- *Information technology* seeks to improve labor productivity through digitization, automation, integration and optimization of information-based tasks and activities.
- *Education, training and distance learning* seek to transfer knowledge efficiently from sources to empower new generations (of labor).

Knowledge technologies (*i.e.*, semantics embedded in tools, processes and infrastructure) will accelerate and dramatically intensify the impact of all of these approaches for dealing with labor costs. Knowledge technologies will promote an unprecedented degree of career mobility and enhanced productivity at all levels of the job market. Given a professionally adept machine backup, early-career specialty training will be substantially shorter, but adaptive mid-career training will be constant. This is good news. However, labor transitions will impact professions, management and technical ranks—categories that previously have been less impacted than agriculture, manufacturing and service industries. Sustainable careers for highly educated, specialized professions will shift toward new knowledge discovery and marketable knowledge-asset creation.

Knowledge tools

In any job, declarative knowledge becomes the basis for new categories of research, analysis, planning, design, diagnosis and decision-management tools. Today's information systems focus on bringing information to the job, that is, situation awareness. Next-wave knowledge systems will deliver all of the theory and information needed to perform the job or task *(see Figure 9)*.

The realm of theory has never been fully present in information technology. For example:

- Rules engines in software applications and databases are trivial.
- Semantics (meanings of things) has been hard-coded into IT technology, so it can never learn.
- AI, to date, has remained machine-like in reasoning, not delivering a foundation for engineering practical knowledge solutions in adaptive human settings.

Rather, the knowledge required to do a job is something an employee has to bring with him or her (via previous education and experience) or learn (on the job or by formal training). This education is expensive to acquire. Also, when people leave, the knowledge is rapidly lost to the organization. Similarly, to automate work on the farm, in the factory or in the office, the knowledge required to accomplish the task is laboriously hard-coded into mechanical parts, circuitry and software algorithms. Improvements in capability require repeated investments in next-generation solutions.

Visualize the role of declarative knowledge in discovery, simulation, diagnosis and decision-making. Having theory-in-the-computer enables a legal researcher to both retrieve case law that is relevant to the brief *and* see its reasoning applied to the case at hand. Also, imagine a paraprofessional with knowledge-based tools that enable less educated personnel to perform diagnoses and other key functions of professionals, in legally defensible ways.

In engineering, theory-and-information-in-a-computer leads to a new kind of design-build process in manufacturing, architecture and engineering. Here, semantic-form declarative-knowledge tools accelerate the design cycle, especially for complex engineered products such as cars and airplanes.

The question might be: What are all designs that have specified properties of performance, noise and safety characteristics? The declarative-knowledge web includes all science, engineering, manufacturing, standards and regulations, as well as history. Design tools embody knowledge and theory for all possible designs in a solution space, enabling "what if" simulations that reason from desired results and attributes backwards.

From a semantic-form model, design flows directly to a manufacturing process that proceeds from virtual to actual. Designs can be automatically rendered as drawings, described as specifications, presented as briefings, planned and scheduled as a work breakdown structure and bill of materials, outsourced and subcontracted through a multi-tier supplier network, submitted for regulatory approval, and so on.

Simulation is another low-hanging fruit of declarative knowledge. Theory-in-a-computer immediately calls for some way to test it. Simulation is the preeminent way to test. Semantics and the abundance of theory concerning every physical, rational and social process will make knowledge-based simulation a central subject of every argument on plans, policies, strategies, new law, economics, social values, etc. Proponents and skeptics alike will test macro and micro models against past history and their suitability to predict the future will become the basis for debate and vivid interactive demonstrations.

Demonstrations of knowledge-based simulation will extend to historical, professional and archetypical personalities. We'll probably see this first as entertainment, then as models of great teachers in action, of enlightened prophets and practitioners—or, alas, of individuals trapped in narrow and ignorant worldviews. Ultimately, the most attractive of these may become images of the kind of person or expert or teacher or parent others might become if they could apprentice themselves to the training and use of particular knowledge assets.

To summarize, knowledge tools have broad applications. There are as many domains for knowledge-enabled labor tools and systems as there are:

- Industry sectors and segments—government, manufacturing, services, energy, publishing, etc.
- Job categories—by role and responsibilities within an organization.
- Functions—such as decision-making, research, design, planning, analysis, marketing, sales, support.
- Disciplines—including management, projects, engineering, accounting, finance, software development, medicine, law, scholarship, etc.
- Hobbies and interests—gardening, home improvement, entertainment, games.

Semantics for Publishers

In the knowledge age, the concept of publishing needs redefinition. Part 2 of this series discussed the content cycle through which publishers create, acquire, manage, package, deliver and make content public for use.

FIGURE 14: CHARACTERISTICS OF NEXT-WAVE INFORMATION AND KNOWLEDGE TECHNOLOGIES

Summary of some of the general characteristics of next-wave information and knowledge technologies.

	Information	Knowledge
New Capabilities	Solve or better manage existing IT problems, such as search and integration of content and processes.	New product categories, different capabilities tap new value sources; know, learn, communicate; do things we couldn't do or afford to do before.
Operating System	Context computing creates pervasive service grid.	Knowledge operating system maintains and reasons across massive semantic webs of theory and information.
Data management	Ontology-based smart data with massive associations exceeds capabilities of RDBMS.	Semantic-form knowledge stacks avoid exponential complexity growth associated with relational DBMS technology.
Software	Thinner applications as more "smarts" goes into the data.	Thin applications for declarative knowledge computing; semantic form tools; knowledge engines embedded in intelligent systems.
Hardware	New virtual machines optimized for context processing in a massively peer-to-peer mobile world.	Long-lived, non-Von Neumann computing architectures, optimized for *n*-ary memory traversal and mobility.
Systems	Autonomic systems that can self-declare, self-configure, self-integrate with other systems, self-optimize, self-protect and self heal.	Autonomous systems that know, reason like people and can learn.
Ecology	Improved IT life-cycle economics benefits existing participants: reduced effort, cost and time to develop, deploy, operate, service and maintain, and upgrade solutions.	Software value chain with new players including content providers, government, and third- and fourth-party developers. Self-evolving products transform life-cycle economics of IT, publishing, manufacturing and other industries.

To this we must now add the life cycle for declarative knowledge, which is about knowing, learning and communicating. Together, these open huge new market opportunities for the publishing industry.

Business-information services and professional publishers have a long history of working with information and reference sources in digital form. They've needed to solve problems of corpus building, maintenance, classification and indexing (including multiple indices), print and digital delivery, currency and relevance to customer need, ease of use and integration with their customers' processes and usage context.

Having experience with different approaches, publishers and business-information services recognize that they need semantics for their content and processes. They've built taxonomies to facilitate access. They recognize the need for markup and metadata to enable better machine processing and searching, as well as content multi-use and multi-channel delivery. Also, some have recognized that ontologies can extend the effectiveness of user interfaces. As they gain experience with Web services for internal-process integration, as well as for customer-facing services, they recognize that process semantics play an important role.

Historically, publishers with knowledge-rich content assets enjoyed the greatest success in domains where content was well structured, or organized so as to be reasonably well understood by the using community. Here, the limitations of language-based approaches to semantics were not overly burdensome, because the target audience could supply the knowledge needed to use the service effectively. But this limited the opportunity to add value, since the customer was supplying the smarts. Semantic-form declarative knowledge provides a way for publishers to escape that limit and move from an information service to a knowledge platform for their products. This leads to product families that change the rules of the game by taking significant time and cost out of their customer's or client's process through knowledge tools.

For publishers, declarative knowledge creates a new class of business opportunity that applies across many categories of business-information service as well as professional and scholarly publishing. For that matter, it applies to many categories of consumer, hobby and entertainment publishing just as well.

Moving from searching to knowing adds a new level of value. The direction is from computer-aided access to information, to semantics-enabled navigation of concepts, to declarative knowledge enabled reasoning across knowledge assets. At each stage, semantics increases asset value.

Using semantics in product strategy

The basic strategy is to convert legacy assets from a publishing division or business-information service to create a new type of product that combines all relevant theory and fact into a reasoning tool. For example, when asked a question, the new tool answers the question and shows tradeoffs and reasoning. It doesn't just return a list of sources. It isn't a book, but it might be a DVD with a book about it. Complex theory and details (probably organized as tables) and other reference knowledge, which would just not be practical to publish in a 2D format, become practical and valuable as a semantic-form encoded knowledge tool.

Putting both theory and information into the computer creates a wholly new experience for the customer. It's like the difference between reading a book about playing a game of chess and having an expert advisor to help you strategize and play the game better than ever before. That's why the future of reference services will be embedded software rather than static reading material.

Combining relevant theory with information changes the rules of the content marketplace. It opens a new competitive vector. The new category of asset is a tool rather than just a publication. The tool commands critical reference knowledge. The new product becomes an active (not passive) asset that is self-evolving and self-learning, and that increases in value as it is used.

For publishers with vision, next-wave competition will be based not just on the completeness and timeliness of information, the quality of its organization and the ease of access to sources, but also on the performance of the knowledge asset at conducting specific tasks and functions performed by those using the service. Success will depend on the quality of the results that customers achieve from applying the knowledge-based tool to activities such as research, analysis, planning, simulation and testing and evaluating alternatives, consequences and trade-offs.

Declarative knowledge becomes the cornerstone of knowledge-age publishing strategies and the basis for sustainable brand dominance. This will be true in areas of popular culture, fashion and entertainment; in news and information segments; and in professional, scholarly and business segments. Knowledge dominance will be the key determinant of who "owns" which media space.

The content providers that are in the best position to win are likely to be publishers that already have a strong base of (knowledge-rich) content as well as strong, established relationships with the specific (consumer or other) micro market. But this cannot be taken for granted, since competing interests could tap a broad range of sources to develop competing knowledge assets. The prize for the publisher is to "own" the meeting place for those that want to learn and do (whether they be hobbyists or business professionals), with those that want to market to that interest (advertisers, etc.), and those that have something to say or communicate or teach to this audience.

Owning the forum, and with it strong, highly valued life-cycle relationships with end customers, is what enables the publisher to collect (multi-channel) subscription revenues, advertiser-based revenues (including co-marketing and co-selling revenues, when the publisher's business model includes e-commerce), and ancillary service revenues.

Moving first to establish a knowledge-age market is important because it creates a barrier to entry to other publishers. The first into a new market spends the least and gains the greatest share. The second to arrive must spend twice as much to gain half the share of market. And so on.

To summarize, the opportunity for commercial publishers is to amass and organize a dominating reference source. The marshaling and authentication of any body of theory is a capital expense conveying ownership and creating substantial barriers to competition. Dominating the theory positions in economically significant markets creates the frameworks for structuring all tasks and information use. In declarative semantic-form, theories will remain relevant for tens to hundreds (potentially thousands) of years, independent of facts and language changes that make them appear different. New market opportunities build firmly on existing customer relationships, content assets and subject-matter expertise.

Publishing models facilitate knowledge capture

A key gating factor for knowledge-age publishing markets is the cost of converting legacy content to declarative semantic-form knowledge. The economics of knowledge acquisition have matured over the past 30 years. In that time, important lessons have been learned. They include:

- Hand-building ontologies and rational architectures is too expensive.
- Experts validate system particulars, but these are self-assembled on a productive industrial scale requiring investment and ongoing production and maintenance support.
- The sub-language hypothesis for ontology construction (*i.e.*, that a semantically well-formed context of language exists and is widely understood in some domain) doesn't work out in practice. Rapid change intermixes specialized terminology in short order.
- Language-based ontologies have inherent ambiguities.
- Capturing the declarative knowledge from legacy reference-content sources is economically feasible.
- The cost of converting from semantics to language forms is tractable and has bounded economics.
- The cost of attempting to capture semantics using language forms is unbounded.
- Developing language-based content and then converting it to semantic form adds about 5–7 percent to the cost of the original content product.

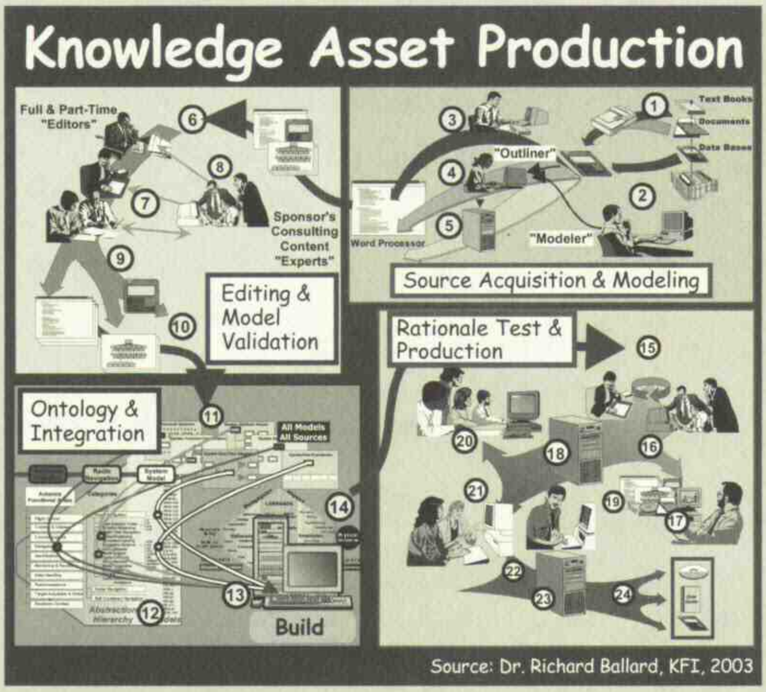
Figure 10: Producing Knowledge Assets

This diagram depicts 24 steps in the process of building knowledge assets from legacy sources. This process is divided into four phases:

- Source acquisition and modeling.
- Editing and model validation.
- Ontology and integration.
- Rationale test and production.

The key roles in this process include the following:

- Project manager, product designer.
- Outliners, transformers, production personnel.
- Knowledge editors.
- Subject specialists.



Source: Dr. Richard Ballard, KFI, 2003

The process of knowledge-asset production introduces several categories of new tools, for example:

- Learning agents extract meaning from any and all forms of content, and encode it in semantic form.
- Acquisition tools convert content to semantics.
- Creation tools express and manipulate ideas in semantic form.
- Building and editing tools amass and integrate knowledge.
- Knowledge-worker tools enable individuals and teams to work with ideas in semantic form and to integrate knowledge assets together.
- Semantic browsers (or "knowing" tools) view knowledge and related content, following reasoning paths or answering questions.
- Knowledge engines form part of knowledge-based computing and application processes.
- Communicating and teaching tools translate from semantic form to language, picture, simulation and other content forms.

- Developing content directly in semantic form and then expressing it in multiple language forms across multiple media can save 50–60 percent of content life-cycle costs.[2]

For the next few years, the process of creating semantic-form knowledge assets depends on human modelers and editors to carry knowledge across the semantic gap from linguistic ambiguity into precise and validated semantics. *(See Figure 10 for an overview of this process.)* The good news is that publishers already have the business and staff models to transform existing content assets into products. Further, commercially marketed knowledge assets from different sources can be integrated into small to massive layered stacks *(see Figure 11)*. However, at the point where science and commercial work products are created originally and delivered preferentially in the more valuable and sharable semantic codes, this need for some human agent in the loop disappears.

What about knowledge computing across the semantic web? For humans, knowledge computing probably does not take place across the Internet directly, but rather through massive semantic webs of knowledge that are local to them, but regularly updated via networks. For humans, the speed of thought doesn't wait on speed-of-light delays or interminable transmission jumps, but it does depend on navigating an *n*-ary reasoning path. Machines, by contrast, operate over longer times without organic short-term loss of attention.

Lost in translation

In the 1980s, the Interagency Language Roundtable (ILR), made up of 18 agencies of the U.S. Federal Government, developed a framework for testing and evaluating human proficiency with language. The framework for language proficiency addresses speaking, listening, reading and writing at five levels of complexity, *i.e.*, none, elementary, limited working proficiency, general professional proficiency and advanced professional proficiency. The framework was then expanded to measure performance at translation. The ILR framework treats translation as a composite of skills that includes reading in the source language, writing in the destination language and mak-

---

[2] In more than 50 knowledge-engineering projects for publishers and government agencies conducted by Knowledge Foundations, Inc. (KFI), the researchers found that:

(1) The cost for capturing a source completely into semantic form averaged $5,000–$7,000 in direct labor, compared with an initial investment in the $80,000–$100,000 range to research, edit and produce the source document in digital form.

(2) Reuse of knowledge assets in a field accounted for 70–80 percent of the knowledge base of subsequent projects, once the first "definitional" project had been completed.

(3) Savings of 50–60 percent over the content-media life cycle are attainable for projects involving natural-language generation of dialogs, documents, graphics and instructional materials from semantic-form knowledge.

ing congruity judgments. The framework distinguishes among professional, transitional and pre-professional levels of performance and introduces four terms associated with bringing texts across languages: translation, rendition, code-matching and glossing.

What is significant in the ILR framework is the importance it assigns to semantics in attaining progressive levels of proficiency with a language, as well as levels of performance with translation. To some extent, this echoes common sense. If you want something translated well, then choose someone who knows the subject matter, not just the language. And ideally, find someone who knows the culture of the target audience as well and how to communicate with it.

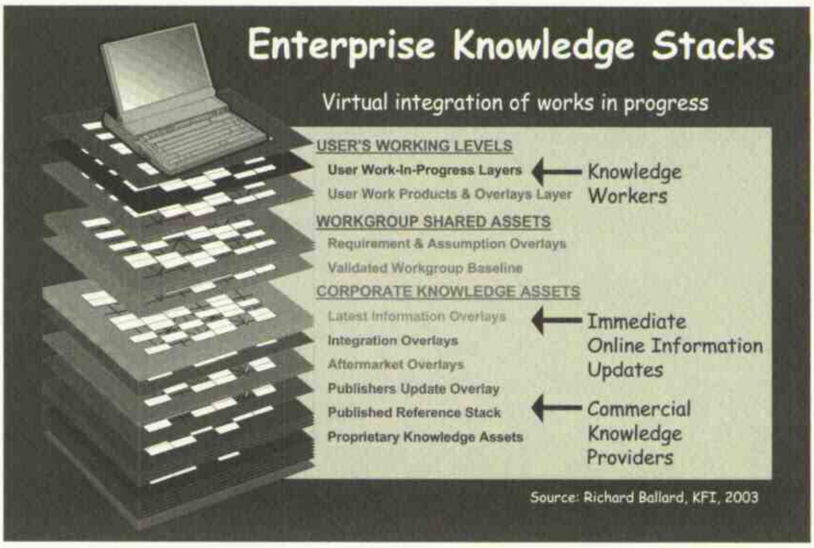
The realm of semantics contains hundreds of millions of unique ideas and concept instances, while language consists of only a few hundred thousand words, at most. As shown in Figure 12, the path from language to language is always ambiguous. The path from linguistics to semantics is always approximate, partial, unbounded and economically open-ended. You cannot get there from language using language as the encoding for meanings. On the other hand, the path from semantics to linguistics (from semantic-form meaning to text, pictures or sound) is straightforward and achievable.

Language to semantics is an "inverse problem." If you start from semantics, where every idea has a unique coded identifier in an ontology of millions of ideas, you can look backward from that ontology with far more certainty. For example, you might discover that there were 43 common uses for that sound and 2,489 rare uses by groups numbering 100 or fewer. Then, if all possible associations are known, moving in the forward direction (from semantics to linguistics) is achievable.

In the coming era, it is not unlikely that the machines accompanying us (such as cell phones and PDAs) may play a major role in mediating our conversations, asking and answering questions, noting significant agreements and differences in our planned objectives, and raising awareness of the outstanding issues we may seek to resolve.

Ideas become products, and prospects become business relationships, through a process that entails a cycle of communications. Semantic-to-linguistic translation promises extraordinary improvements in disambiguation. In the not too distant future, we should accept that the machine's semantic web-based language skills may be better than our own. Who then should write our technical literature? If we do not want the costs of continual factual edits, then probably the machine should.

Semantics-based natural-language generation will play a major role in all stages of the life cycle of customer relationships, product design and manufacturing, supply-chain relationships, legal and regulatory matters, and health care.

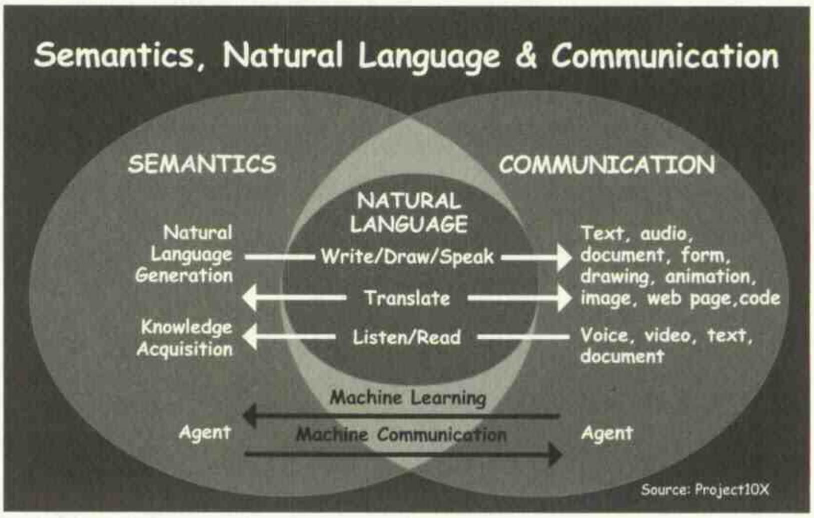


Source: Richard Ballard, KFI, 2003

Figure 11: Enterprise Knowledge Stacks. This drawing shows a work environment wherein a collection of layers has been gathered, organized, integrated and worked upon by any number of contributing users. A session overlay creates this stack by dynamically linking together the working layers. Linkages are virtual. In this example, user work-in-progress layers may be updated, while proprietary product layers are treated as "read only" within such stacks. Combined with encryption and digital rights management, knowledge stacks provide an enabling infrastructure for knowledge commerce.

Semantics for Government

A "Semantic Technologies for E-Government" conference was held at the White House Conference Center in September 2003. Among the many agencies represented by more than 130 attendees were the Army, Census Bureau, CIA, DIA, DOE, EPA, GSA, IRS, Navy, NARA, NASA, NSA, NSF, SSA, USDA and the U.S. Patent Office. A number of attendees were from nonprofit organizations such as Aerospace.org and Mitre. Major government contractors were also well represented, including BBN, CSC, Lockheed Martin and SAIC.



Source: Project10X

Figure 12: Semantics, Natural Language and Communication. This diagram depicts the relative sizes of the intersecting realms of semantics, natural language and communications. It summarizes pathways of knowledge acquisition, natural-language generation, translation, machine learning and machine-to-machine communications.

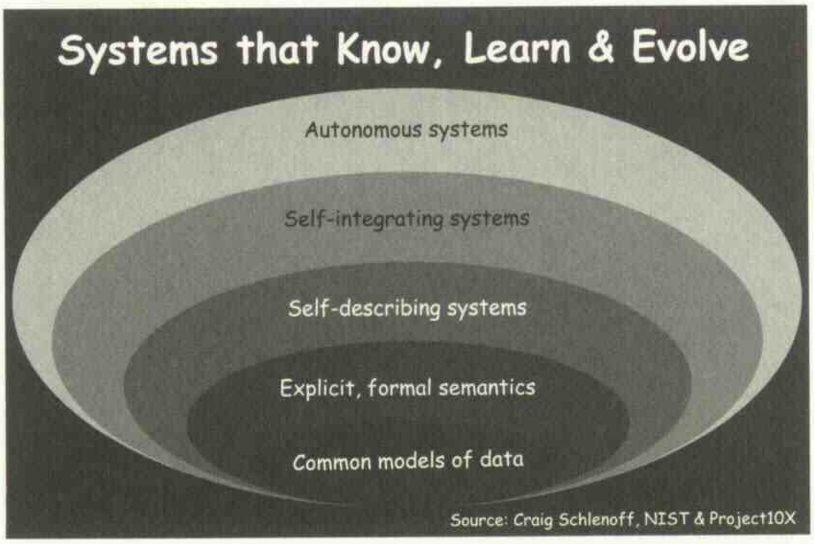
Q&A (question and answer) systems were a common concern of many of these agencies. That is, how could they put together systems that a policy researcher, program manager, intelligence analyst, executive, congressional staffer or constituent could use to integrate knowledge with public and classified information resources to rapidly explore and answer complex questions?

Semantics are the only practical way to build a Q&A system. The capabilities can be surprising. Here, for example, we envision capabilities that could become available to an enterprise or federal agency transitioning from current information-systems technology to a Q&A system based on semantic-form knowledge technology.

Stage 1. Initial capital investment builds the ontology, converts and bulk loads the key reference knowledge (hundreds or thousands of documents), validates and links the core knowledge assets, and deploys basic knowledge tools such as a constraint browser. Knowledge assets and tools enable professionals to research questions, alternatives and trade offs. The replacement for searching is the creation of ontological hierarchies filled with abstract models defined semantically by the relationships and associations that are explicit within the ontology. The expectation is that critical issues and questions, which used to take months to answer, can be researched and evaluated in hours to reach the point of decisive recommendation.

Stage 2. The next stage of development builds new tools for knowledge and information acquisition and machine learning. This includes self-learning capabilities for (1) updating situation awareness, (2) incorporating advances in theory, and (3) expanding the range of policy research and decision-making that the knowledge stack is capable of addressing. Knowledge is an active asset. The value that this system provides continues to amplify as the knowledge base grows and the system gains more experience—which it does as people use it. The system has features that enable it to learn and improve its reasoning and communications skills.

Figure 13: Systems That Know, Learn and Evolve. This diagram depicts stages in the evolution of intelligent products and processes.



Source: Craig Schlenoff, NIST & Project10X

Stage 3. The next wave of development would focus on communications capabilities. These include semantics-based natural-language generation. One development is the capability to produce good-quality briefing books and presentations and other communications directly from the system. Another is the system's capability to teach what it knows. The system can compose lesson plans, conduct sessions, answer questions and customize materials to the needs and preferences of the learner.

Stage 4. A further round of development gives the system the capability to speak, listen and write so that the system is capable of communicating effectively in multiple languages. Language proficiency can reach level 3 (requiring both subject matter and cultural semantics) in the framework developed by the Interagency Language Roundtable. Translating from one language into another can reach a practitioner's level of performance—transitional to professional, and much more than a rendition. With the acquisition of language skills, any of the system's knowledge tools can carry on a conversation with humans in any language of their choosing. Similarly, the system is capable of assimilating information written in different source languages.

Semantics for Manufacturing

Manufacturing paradigms have changed. Manufacturers used to focus on regional market dominance, vertical process integration and strategies to contain local labor costs. Now, the model is to design where the knowledge is, manufacture where labor and other factors are most economical and compete in global markets. Studies and research over the past decade have focused on how to build advanced integrated manufacturing technologies and processes. A key theme is the role of knowledge-based technologies in "smart" products and processes.

According to Craig Schlenoff of NIST (National Institute of Standards and Technology), the evolution toward smart products and processes starts with common models of data, then advances to explicit, formal semantics (dealing with the relationships rather than just the terminology), to self-describing systems, and eventually to self-integrating systems. As shown in Figure 13, the journey doesn't stop there. The goal is to create autonomic and autonomous systems that know, learn and can reason as people do and can self-evolve.

Currently, the aims of advanced manufacturing studies are to develop methodologies and approaches to machine learning and rational theory construction in every area well practiced by humans. Originally, this goal was targeted to 2010–2015, dates set by DoD and NASA for large-scale introduction of autonomous aircraft and intelligent robotic planetary and giant moon explorers. There, the distances and time delays require systems to both explore and solve their own problems during the mission.

Intelligence in aerospace. Some of our best illustrations of what intelligent machines can do are taken from aircraft designs. These applications are well aware that humans can behave most ignorantly as operators, so the machine itself has to trade off what it is commanded to do against the other imperatives it has for accomplishing its longer-term mission. This was precisely Hal's dilemma in *2001: A Space Odyssey.* (As explained in the sequel, Hal had balanced his secret instructions from a higher authority against the mission threat posed by Dave and the rest of the crew in conspiring to turn him off.)

Better examples are found in the way flight and mission computers in airplanes take a pilot's steering commands as "suggestions" rather than overrides. They have to keep such commands from tearing the wings off or steering a perfectly fine airplane into the ground, so they make trade offs between safety and radically unwise control or emergency actions. For example, if an aircraft is inverted and close to the deck, then a pilot-ejection command would kill the pilot by blasting him into the ground. The plane may automatically do a snap-role-sacrificing its wings to point the ejection skyward. These things sound far-fetched, but they are indeed in the avionics instructions of fighter jets.

Similarly, aircraft know their own condition far better than any pilot, and they can report it directly to maintenance crews even before landing. Though the pilot may too tired or stressed to go on to sortie again after landing, the plane may be quite capable of continuing operations. This is why carrier aircraft are switching from large flight sorties to smaller groups operating in "pit stop" fashion, cycling through fueling, arming and reconfiguring for the next mission. The limit to carrier productivity is the surge-sortie rate, and the limits are defined by exhaustion of the deckboard launch and handling crews. Pit-stop sequences conserve launch crews, because maintenance crews know what each plane needs long before it returns from its last mission, and because most of the planes can be serviced on deck. Currently, aircraft life cycles and refits are tied strictly to flight hours, but airplanes that had only three computers before now have close to 50. So the subsystems can monitor their own operating condition and expected lifetimes, and can carry this information and history from one platform installation to the next.

Strategies for managing aircraft life cycles vary considerably based upon availability of platform and subsystem replacements. Sometimes, management favors phasing out whole model lines at about the same time to guarantee comparable efficiency for all operating units and to limit maintenance training to just one generation of machines. Where replacements are uncertain, the strategy is to keep a few always working by scavenging parts from the ever-present "hangar queens" that always have some undiagnosable malady.

As we enter the knowledge age, one expectation is that industry will begin moving away from unique, rapidly changing hardware aggregations toward longer-lived platforms enduring for tens to hundreds of years. New hardware "limit machines" will be engineered to be flexible hosts for virtually any function within the everyday environmental limits associated with a wide range of locations and situations of regular and extraordinary use. There are many examples of such platforms in aviation (DC-3, B-52, C-130, etc.), but the knowledge age will see this strategy applied across a range of industries and product categories. These long-lived hosts will tend to remain in continuous production for extended periods so as to continually supply and evolve capabilities operating near the physical limits to material and ultimate system performance for type. These will minimize scarce resource usage and fully close the materials-recycling loops, providing an equalizing foundation for all of the global civilization.

Dawn of the Knowledge Age

In this article, we have focused on the semantic wave as a revolution in content with major economic consequences.

We started by explaining that digital semantics are all about representing more and more of the things we can know about something as a new kind of data that can be processed with computers. The trend is toward representing this knowledge as data.

We examined the role of semantics in content management, libraries and research We saw that knowledge representation plays a vital role in the present and future of content search, and it is worth millions of dollars in productivity. We reviewed the capabilities and limitations of forms knowledge representation. We learned that it is possible to overcome limitations of language-based approaches to knowledge representation through semantic-form knowledge.

Next, we investigated semantics for computing. We found that process semantics are more than strategic; they are absolutely essential for the success of all major lines of information-technology development, starting with the semantic web. All of the major consumer electronics companies, telecommunications companies and IT companies are weighing in, and the market stakes, which are already vast, are rising. Yet here, too, we learned that language-based semantics and object-oriented procedural models of computing may not be enough to win the day. We identified context computing as a key focus for a new, unified processing paradigm. And we talked about declarative-knowledge computing as, potentially, an ultimate solution to the problem of process complexity and language ambiguity.

We discussed semantic-form knowledge as a driver for labor productivity. We explored new categories of knowledge tools for research, design, planning, analysis, simulation, and decision-making that have the potential to revolutionize professions, management and most knowledge-worker job categories. Taking a longer view, education is destined to be transformed as well.

We saw that publishers and professional groups stand to win big in the knowledge age. They own the reference assets it will take to jump-start new markets. They have the organizational and editorial disciplines needed to build knowledge assets. And they have the established customer relationships needed to dominate new markets.

We briefly touched on semantics across government to sketch the type of knowledge-based capabilities that might be brought to bear on policy-making, defense, intelligence, program management, regulation and public information.

Lastly, we examined semantics for manufacturing and discovered the mainstream role that knowledge technologies will play in advanced manufacturing processes and fundamentally new categories of intelligent products and services. For manufacturers, the market stakes are high, because the economic impact of competitiveness and new markets will be played on a global scale.

The bottom line is that we are at the dawn of the knowledge age. In this report, we've only been able to sketch broad outlines of a major transition coming for the world economy that will occupy several decades *(see Figure 14)*. Knowledge technologies based on science and engineering will power economic expansions measured in the trillions of dollars world-wide. The economic driving force, as we pointed out in Part 1, is a hundred-fold shift in the economics of knowledge (as contrasted with information). The impacts cut deeper and have a much wider scope than previous waves. First, knowledge technologies impact the life-cycle costs of labor and education. Second, knowledge technologies directly affect the global competitiveness of entire industries, especially IT and manufacturing. Third, knowledge technologies open major new markets for "smart" products, services and processes that tap new sources of value. And fourth, knowledge technologies establish the horizons and means for a level of global planning and coordination that is unprecedented in human history. **TSR**

About the authors

Mills Davis is managing director of Project10X. A researcher, visionary and industry consultant for more than two decades, Davis focuses on improving profitability through process innovation, technology investment and strategic partnerships. Previously, he directed the Digital Roadmaps Project and the HiFi Color Project. He can be reached by e-mail at