

# Business language analysis for object-oriented information systems

by D. W. McDavid

*Business language analysis grows out of a philosophy that treats business organizations as living systems. A key concern is the meaning of business information that provides adaptive survival advantage and strategic leverage. Popular object-oriented methodologies correctly recognize the need to identify business objects by analyzing the problem domain. The approach described in this paper fills in the details that are implied, but not specified by other methods. It builds a business language model that clarifies both the content and structure of the terminology actually used in the business. Simple examples of business language analyses are given. Deeper insight is offered in a discussion of lexical semantic and category theory and in the proposed notion of business language patterns.*

Information is an essential dimension of any business. In fact, "... organisations, themselves are information systems."<sup>1</sup> Communication of information within and among organizations comes in the form of conversations, commitments, contracts, and transactions. Industries and professions often communicate in a jargon that is incomprehensible to outsiders. The challenge for information systems is to facilitate organizational communication, sometimes translating one group's jargon into terms that are meaningful to others. The information systems profession will only be successful in this endeavor to the extent that it builds systems based on a fundamental appreciation for the meaning of business language.

An accurate and timely understanding of information needs is a prerequisite for effective enterprise-

wide information systems, whether object-based or procedure-data applications. Business language analysis identifies domain-specific business terms from documents and conversations. It draws on predefined patterns of generic business concepts to classify and link business terms into a semantic network. This network of terms then provides the basis for object modeling, user interface design, persistent data management design, and test case generation.

Business language analysis produces models of the information that is used and exchanged among business organizations. It follows an engineering tradition of separating analysis from design and using models to create shared understanding across teams of people working on a technical problem.

It is always important to understand the purpose and intended audience of any model or modeling activity. Various aspects of the business domain can be modeled. Requirements models treat essential, or logical, aspects of data and data processing systems. Design models explore physical aspects of information systems. There are also models of objective reality, including business process models, organization charts, charts of accounts, and plant layouts. And, finally, there are models of the *information rep-*

©Copyright 1996 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

resentation of things important to the business.<sup>2</sup> Business language analysis creates models of this latter type. It is a method for analyzing business semantics: the meaning of business information. It treats information of all types, from inventories to goals, from processes to rules and procedures, whether accessed by computers, bound up in documents, or present in human brains.

This paper discusses the limitations of a paradigm for understanding business information needs that is based purely on an engineering perspective. It explores an alternative way of thinking about business information systems using concepts from general systems theory that view the information system as the mind of a living system. We present object orientation as the most hopeful approach to realizing an architecture based on cooperating mental agents. We propose business language analysis as the conceptual framework for information system construction and show how business language analysis identifies terms in actual use in the business, and then classifies and links those terms using a set of generic business concepts. The paper recommends specific activities and work products to produce a model of business language and suggests use of the language model in various information systems development activities.

The paper presents preliminary findings from the author's experience. Business language analysis is exploratory and invites participation and feedback from users. The paper concludes with suggestions of areas where future work should proceed.

### **Architecture, engineering, and understanding**

Successful methodologies have been built around the concepts of architecture<sup>3</sup> and engineering<sup>4</sup> as applied to information systems. The resulting models have been useful, and lessons from those endeavors have helped the information systems profession make great strides in recent decades.

An engineering perspective will always be vital to the success of information systems with respect to performance requirements for hardware and network components. However, when architecture and engineering are taken to be sufficient models for what information systems are all about, a fundamental conceptual confusion results. The following quote presents the nature of this confusion:

When applied to an information system, the word architecture is a metaphor that compares the construction of a computer system to the construction of a house. . . . Enterprise or business model[s are analogous to] . . . the architect's drawings that depict the final *building* from the perspective of the owner, who will have to live with it in the daily routines of business. They correspond to the enterprise (business) model, which constitutes the design of the *business* and shows the business entities and processes and how they interact.<sup>5</sup> [Emphasis added.]

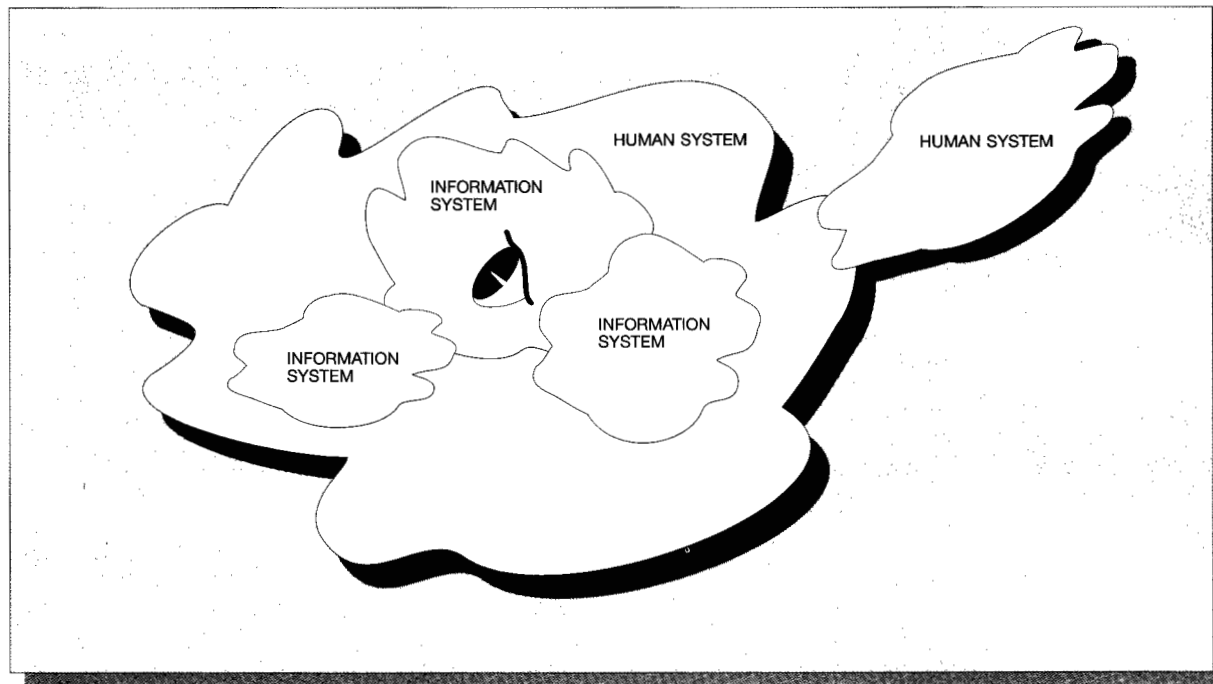
In this passage, a computer information system is compared to a building (a house or a business structure). The passage indicates that an enterprise model is both a model of an artifact (building) and the business served by that artifact. It is like equating a house with the family that will live in the house. Clearly a house and a family are not the same, and cannot be described by the same model. A model of a family, including the number of members, interests and hobbies, and ages and expected growth patterns, would be very useful input for the design of a home. In the same way, a model of a business provides very useful input for the model of its information systems. The point is, they are different models.

Understanding user requirements is widely acknowledged as a critical success factor for information systems. Many methods, even when they address business issues, do so from the perspective of a particular system development effort, looking outward (see Figure 1). The very words "requirements" and "user" reveal a perspective from inside some (proposed) information system. All we need to do is ask "Requirements for *what?*" and "Users of *what?*" to see that perspective. The answer must be "Requirements and users of some information system."

In contrast, the perspective that underpins business language analysis (Figure 2) is from outside the enterprise, looking in at the human systems and at the information systems that make up that enterprise. This perspective forces the information needs of the enterprise to be examined as a whole. It can even embrace an extended enterprise, which reaches out to incorporate external enterprises as part of a larger human system.

A human-centered approach to information management provides the underpinning for business language analysis. The perspective of business language analysis is less on *engineering* and more on *under-*

Figure 1 Outward perspective on business issues



*standing.* It recognizes that it is a losing proposition to try to engineer human communication. It aspires to understand the meaning of communication within the human activity system so as to support the evolution of the business, along with its information systems capability.

#### **An alternative view of information systems**

Business language analysis is a bottom-up approach to articulating the information needs of business enterprises. It is based on an alternative, systemic view of information systems as the minds and nervous systems of living organizations.

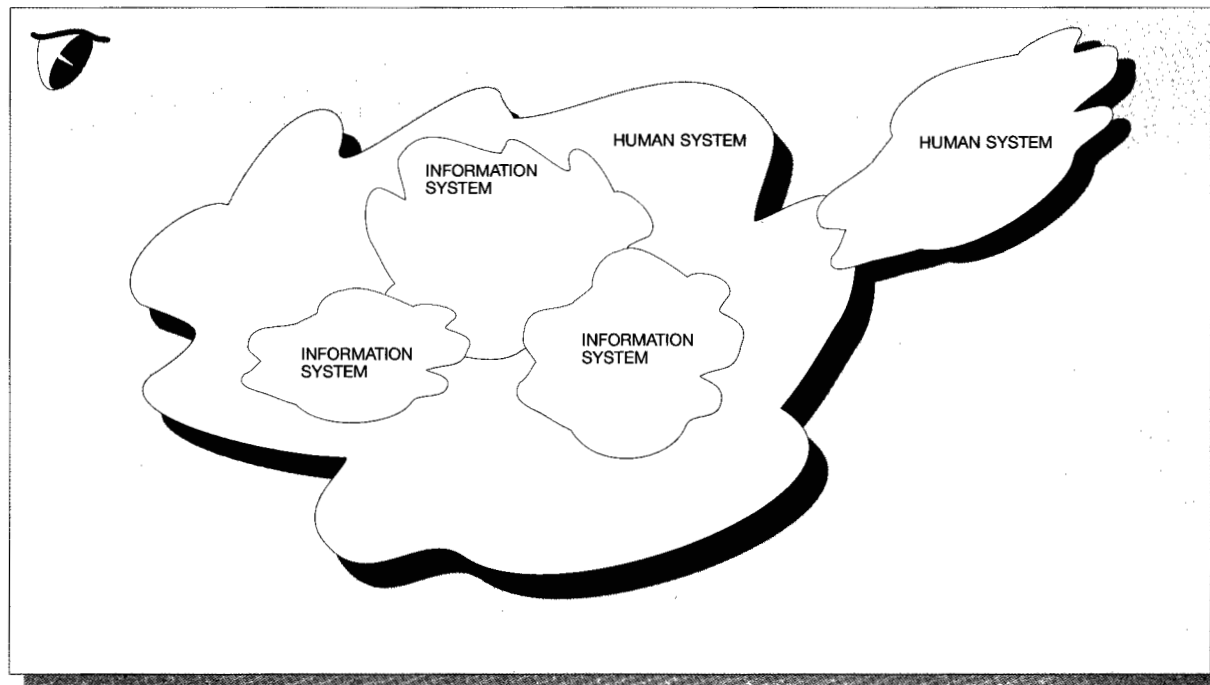
**Motivation.** There are several motivators for taking an alternative view of information systems. One motivator is the trend toward ever-increasing amounts of information being held and manipulated by automated data processing systems. Hardware has become a commodity, as have many basic software components. As more technical issues are resolved, and as information technology penetrates deeper into business enterprises, we are better able to focus attention on the use of business information as

a strategic resource in an increasingly competitive environment.

Increasing business complexity, competitiveness, and speed are part of the motivation. Stephan Haeckel and Richard Nolan present an analogy for today's fast-moving business climate in the notion of managing by wire. "Flying by wire" means flying an aircraft by controlling an information representation of the aircraft through the use of heads-up displays and electronic controls; the computer actually manipulates the aircraft control surfaces and powerplant controls. Successful companies are able to sense and respond to rapidly changing customer needs. "The ideal manage-by-wire implementation uses an enterprise model to represent the operations of an entire business. Based on this model, expert systems, databases, software objects, and other technical components are integrated to do the equivalent of flying by wire."<sup>6</sup>

However, the problem is actually more complex than this analogy would suggest. Managing a business involves social and personal dimensions, as well as physical forces. This complexity critically affects the

Figure 2 Inward perspective of business language analysis



challenge of building systems that support information needs. Thomas Davenport proposes an ecology of business information. He claims that the information technology community is in a mid-life crisis, brought about by failure to deliver anticipated value to its constituency. It has been dominated by the engineering design and architecture model—the technological plumbing. “Information management must begin by thinking about how people use information—not with how people use machines. . . . A human-centered approach assumes information is complex, ever-expanding and impossible to control completely. The natural world is a more apt metaphor for the information age than architecture.”<sup>7</sup>

This alternative view is also motivated by the shifting, insatiable nature of information systems requirements. Experience has demonstrated that the more application functionality provided, the more users demand. We need to get out in front of this requirements gap by anticipating user needs before they materialize. How is it possible to anticipate user needs? The only way is by understanding common patterns of behavior and semantic structure, which arise because of the true nature of organizations and the in-

formation systems that serve them. At the most basic level, we need to recognize that both businesses and information systems are indeed systems, to be understood by applying lessons from general systems theory.

**Systems thinking.** General systems theory is a branch of science that has emerged in the 20th century as a counterpoint to the successful, but sometimes limited, reductionist approach to science. The “systems paradigm is concerned with wholes and their properties.”<sup>8</sup> It is based on the recognition that a system has properties that emerge from, but transcend, the sum of its individual parts. Systems can be both hierarchical and interpenetrating. There is a hierarchy of systems from simple thermostats to the space shuttle, and from cells, to organs, to organisms, to organizations. A human being (a system) plays roles in many different social systems (families, corporations, organizations). A hospital is a component of both the health care system and the economic system.

A general systems principle states that when one system exists to serve another (System A serves System

B), the serving system must be understood in terms of the served system (System A must be understood in terms of System B). Information systems exist to serve human activity systems, and, therefore, "information systems design must stem from a model of the activity system served."<sup>9</sup>

The systems approach has had notable success in the creation of large, complex engineering artifacts. Lessons learned from this systems approach can be applied to business information systems where the problems are mechanical in nature.<sup>10</sup> More problematic in many ways are the so-called "soft" systems. "Hard" systems thinking is goal-directed, in the sense that [it] begins with the definition of the desirable goal to be achieved." The essence of hard systems is design engineering of a well-known solution to a well-understood problem, where the effort is to choose the best among several alternative approaches. By contrast, soft systems are "management problems . . . in social systems where the goals are often obscure."<sup>11</sup>

It is critical to soft systems thinking to avoid the trap of treating human systems as equivalent to more deterministic mechanical systems. It is tempting to reduce information system projects to hard system problems. In some cases this may be appropriate, if the requirements are simple, clear, and well-articulated. However, such requirements are increasingly the exception, rather than the rule in enterprise class information systems.

**Living organizations.** If we are to understand information systems in terms of the human activity systems they serve, it behooves us to examine the nature of human activity systems (organizations) more closely. Long tradition supports thinking of organizations as living entities.

One of the earliest applications of general systems theory to human activity systems is the living systems model.<sup>12</sup> This model abstracts a common set of functions and subsystems at several levels of recursion, from a single living cell up through very high levels of human organization. These recurring subsystems include material and energy subsystems (ingestor, converter, motor, storage, producer, etc.) and information processing subsystems (memory, encoder, decoder, decider, channel and net, etc.). This model can be used to discover the role or purpose that is served by a particular organization within the larger system of which it is a part (e.g., the phone company plays the role of channel and net in society), and it

can be used to understand the functions *within* the system of interest. Both the phone company and a toy manufacturer will have all the necessary information processing subsystems, in one form or another, created and maintained by information systems professionals.

The viable systems model is another view of organizations, from bee colonies to nations.<sup>13</sup> Every organization (viable system) exists within some environment and has a management function that is accomplished according to some mental model. Operating units are responsible for producing the primary results (products and services) of the organization. A function is responsible for coordinating the set of mental management models, and another uses a direct command channel to give orders to the operating elements. Another important function is responsible for looking outward into the environment as a whole and into the future. There is a function, ideally consisting of the most senior management, that mediates between the current and future needs of the organization. Each of these omnipresent subsystems gives rise to specific information requirements within any organization.

More recently, the concept of the learning organization has emerged from the tradition of systems thinking. Peter Senge provides powerful underlying systemic processes that can drive or inhibit business success.<sup>14</sup> Gareth Morgan proposes several ways of viewing organizations as living things, including organisms, cultures, political systems, and even brains. "Whereas in traditional theories of organization, attention has been devoted to the way communication links are established between different elements of an organization, the brain metaphor helps us appreciate that an organization can itself be regarded as a cognitive system, embodying a structure of thought as well as a pattern of action."<sup>15</sup>

Michael Rothschild has proposed a radical biological, information-centered view of the economy and business. "Orthodox economists still envision the economy as a predictable clockwork mechanism where historical change is irrelevant because all movement is cyclical . . . After DNA was discovered . . . [and] bolstered by stunning breakthroughs in cellular biology, molecular biology, paleontology and ecology . . . it was possible to completely rethink economics . . . as an evolving ecosystem. . . . Genetic and technologic information, despite manifest differences in the branching patterns of their evolutionary histories, are nonetheless members of the same

class of natural phenomena. Both are living, evolving information systems."<sup>16</sup>

Kevin Kelly goes even further. He surveys the fields of robotics, artificial life, natural and artificial ecologies, computer games and art, the Internet, forecasting, and cybernetics, and makes the case for a biology and ecology that includes organisms, organizations, and technology. "The realm of the born—all that is in nature—and the realm of the made—all that is humanly constructed—are becoming one. . . . The challenge is simply stated: Extend the company's internal network outward to include all those with whom the company interacts in the marketplace. Spin a grand web to include employees, suppliers, regulators, and customers; they all become part of your company's collective being. . . . The metaphor of IBM as an organism needs overhauling. IBM is an ecosystem."<sup>17</sup>

This sample of what can be found in systems literature demonstrates that there is value in considering the human activity systems of business as living, thinking systems. That view leads effortlessly to the notion of the human mind as a model for a malleable learning mechanism that can enable competitive business adaptation.

**The mind.** The notion of mind as the seat of human cognition has long been a source of debate among scientists and philosophers. Only recently, with advances in detailed understanding of the functions of the brain, are we beginning to articulate a coherent explanation of the mind and its workings. The distilled essence of this work provides direction to our thinking about the role of information systems in business.

In a survey of the current state of knowledge about the mind, David Taylor raises several interesting issues. He notes that the key functions of the human mind are perceiving, imagining, remembering, thinking, feeling, and controlling action. Contrary to popular belief, a memory is not housed in a single place in the brain, but rather is a distributed function. The most interesting dimension of the mind is its provision of the quality of consciousness. The external sharing of consciousness through communication is the force that drives the newest form of evolution, cultural evolution.<sup>18</sup> We might argue whether businesses exhibit consciousness, and it might be interesting to consider what imagining and feeling are for an organization. At a minimum, however, it is clear that organizations must sense, or perceive, changes

in their environment, and they must think about, or make, decisions that affect their course of action, based on external stimuli and corporate memory.

If memory is a distributed process in the mind, how is it (and other mental functions) accomplished? Marvin Minsky presents an architecture of very sim-

---

**The notion of mind as the seat of human cognition has long been the source of debate.**

---

ple mental agents, each of which is far from intelligent, but which work together in increasingly complex ways to form the society of mind. It is necessary that each agent, from the most primitive sensing mechanisms on up, perform its specialized work correctly. It is equally necessary that the relationships among these agents be maintained and continue to evolve in the learning process.<sup>19</sup>

Arnold Trehub proposes a possible architecture of the physical brain to account for basic human cognitive capabilities. Starting from the physiology of the neuron, with synaptic junctions among axons and dendrites, a mechanism is proposed that can perform tasks that range from parsing any arbitrary object as part of a scene, learning and recalling names for various entities, generating sequences and related inferences, and planning, executing, and learning sequences of actions that satisfy motivational needs. The components include synaptic matrices, simple input preprocessors, clock rings, size and rotation transformers, a semantic network, and various high-level executive processes, such as registers for plans and actions. This physical architecture sheds light on the kinds of primitive capability that are required by organizational information systems.<sup>20</sup>

One of the interesting aspects of the study of cognition is how much the attempt to simulate intelligence with machines has shed light on the nature of human cognition, and vice versa. Out of that convergence toward a unified theory of cognition, Allen Newell proposes the following useful definition that

can apply equally to businesses, computing devices, or human beings: "intelligence [is] . . . a description of adequacy over the joint range of two complex domains, the system's goals and the system's knowledge."<sup>21</sup> This definition highlights two general issues that must be present in any adequate account of organizational information system requirements: organizational goals and organizational knowledge.

This excursion through the literature has not been provided simply for entertainment value. It is meant to lay the groundwork for thinking about business information systems in a different way. The key to this new way of thinking is recognizing the importance of concepts and meaning in the life of the organization, and accepting the validity of the study of meaning by those who would undertake to create or modify the systems that embody this meaning. The issue now becomes: how can this new approach to information systems be applied in practice? How can we take the lessons of systems, living systems, and minds, and use them productively in the service of business? Part of the answer is found in the venerable (on a software time-scale) approach of object orientation.

**Object orientation.** Object orientation has been around for a long time. It has its origins in the simulation of complex systems and is thus based on the systems thinking paradigm. It holds out the promise of addressing the software productivity gap that gives rise to the insatiable demand for increased information system functionality that we noted earlier.

The term object orientation is actually something of a misnomer. In nontechnical use, the word "object" can refer to almost anything and, in general, tends to conjure up something inert and nondescript, like a stone or a clod. If someone were to ask, "What is that object over there?" we would likely not expect to see a person, a cow, a Camaro, or a tricycle as the referent of the word "object." In general use, object is pretty boring.

Software objects are much more interesting than clods or stones. They have life. They have the potential to be the cooperating mental agents of Minsky's architecture. Perhaps a better term than object-oriented programming might be *organic* programming.

Object design can benefit from methods of analysis based on a living, organic paradigm such as we have

just explored. Business language analysis is fundamentally grounded in this paradigm.

A principle of object orientation is that there is a narrow semantic gap between domain understanding and object implementations. Analysis of business language directly supports the injection of business meaning into artifacts built using object technology.

### Business language

All object-oriented methodologies call for identifying business objects from the problem domain. They generally give a few guidelines, such as finding nouns in the requirements statement(s): "Lists of key nouns, gathered from representative documentation and/or use cases, become potential classes."<sup>22</sup> "The objects can be found as naturally occurring entities in the application domain. An object becomes typically a noun which exists in the domain."<sup>23</sup> "Begin by listing candidate object classes found in the written description of the problem. Don't be too selective; write down every class that comes to mind. Classes often correspond to nouns."<sup>24</sup> "As a first approximation one can scrutinize the requirements document, if there is one, and consider the nouns, or better yet, the noun phrases."<sup>25</sup> "As you read, consider the nouns in the written material; these words will often give you a clue about potential Objects in the system."<sup>26</sup> "Roughly speaking nouns are candidate objects, and verbs are candidate operations."<sup>27</sup>

Some methods go a step further by introducing several methods of classification and recognizing that abstraction is a process of discovery in analysis and invention in design. Object-oriented practitioners have used a number of techniques for finding and classifying objects, such as classical categorization, behavior analysis, domain analysis, use-case analysis, class responsibility collaboration cards, informal English description, and structured analysis.<sup>28</sup>

Business language analysis starts from where these other methods leave off by focusing attention on how to make the most of the rich language resources that are available within any business environment. These resources, if studied carefully, will provide guidance as to exactly what information system support needs to be provided.

**Terms and concepts.** Two basic dimensions are needed for a complete model of business meaning. They are the lexicon of terms actually in use by the business and an ontology of concepts that help sort

## LEXICAL SEMANTICS AND LANGUAGE RELATIONSHIPS

There can be many types of semantic relationships among lexical units. These include (but are by no means limited to):

### Congruence

Synonymy—The meaning of two terms is identical.

Hyponymy—The meaning of one term fully includes the meaning of another term.

Compatibility—The meaning of two terms is overlapping, but not identical.

Incompatibility—The meaning of two terms is completely disjoint.

### Opposites

Complimentaries—Two terms divide some conceptual domain into mutually exclusive compartments.

Antonyms—Two terms are on opposite ends of a gradable range.

Directional opposition—Two terms indicate opposite potential paths of a body in motion.

### Configurations

Proportional series—Sets of terms share common traits.

### Hierarchies

Branching—Hierarchies have possible multiple nodes at each level.

Taxonomies—Classifications are based on a single rule of differentiation at each level of the hierarchy.

Meronomies—Components are in assemblies.

Part/whole—Things are naturally divisible into expected parts.

Piece/whole—Things can be divided randomly into pieces.

Nonbranching—Hierarchies have only one node at each level.

The semantic relationships listed above, along with the basic definition of lexical unit, come from the field of lexical semantics.<sup>29</sup> This list is not exhaustive, but it does indicate that, within the study of business terminology, there is a much richer set of relationships than simply

homonyms, synonyms, and hierarchical relationships. In particular, it is interesting to note that taxonomy is a very specialized type of relationship, and that a well-formed taxonomy is much rarer than most analysts would generally imagine.

out the meaning of the terms that are discovered by language analysis.

A business *lexicon* is the set of actual terms used within a particular human activity system, where a term can be a word or a set of words. A term, along with its meaning, constitutes a lexical unit. See the first accompanying sidebar for a summary of some of the semantic relationships that can be used to understand one lexical unit<sup>29</sup> with respect to others in the same body of language.

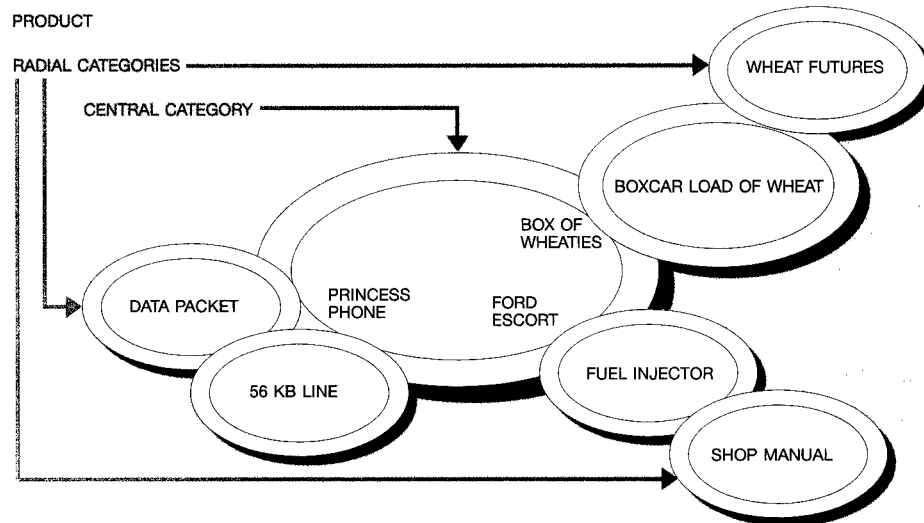
As opposed to the terms that are found in common use within any environment, including a business

environment, an *ontology* is a set of abstract concepts that define the areas of common interest within a particular community. In a philosophical sense an ontology is “a theory of what the world is, or contains.”<sup>30</sup> The scope of an ontology can be:

- Global—Concepts common to all human beings, or all members of a culture. Examples include *Roget's Thesaurus* and the Dewey decimal system.
- Business—Concepts that are common to the world of commerce and enterprise interaction generally
- Domain-specific—Concepts specific to a particular industry, profession, company, or work group



## CATEGORY THEORY AND BUSINESS CONCEPTS



Categories are basic to human cognition. George Lakoff<sup>81</sup> provides a valuable and entertaining survey of empirical and theoretical studies in the field of category theory. Cross-cultural evidence points to common mechanisms for forming categories and expanding the set of categories to accommodate more complex situations. Base-level categories (e.g., the genus level in the biological taxonomy) are the most intuitive for people to discriminate. It is easier to differentiate a cow from a fish than it is to group a cow and a whale together as mammals. Similarly, different species of whale or different varieties of pig may be difficult for the nonexpert to distinguish.

Categories have prototypical members and peripheral members. The peripheral members start to edge off into conceptual areas that eventually require the formation of new categories. Idealized conceptual models (ICMs) are patterns of concepts that define a particular category. Through various methods of extension, radial categories are formed. These radial categories share fewer and fewer of the patterns of concepts that the prototypical pattern exhibits.

As an example of a business-oriented ICM, consider the concept *product*. If we were to define the fundamental

concepts that surround the prototypical idea of product, they would probably include:

- A typical product is the output of an industrial process.
- It is consumable.
- It is composed of discrete, physical units.
- It has a producer.
- It is sold for money.
- It has a specific target set of consumers.

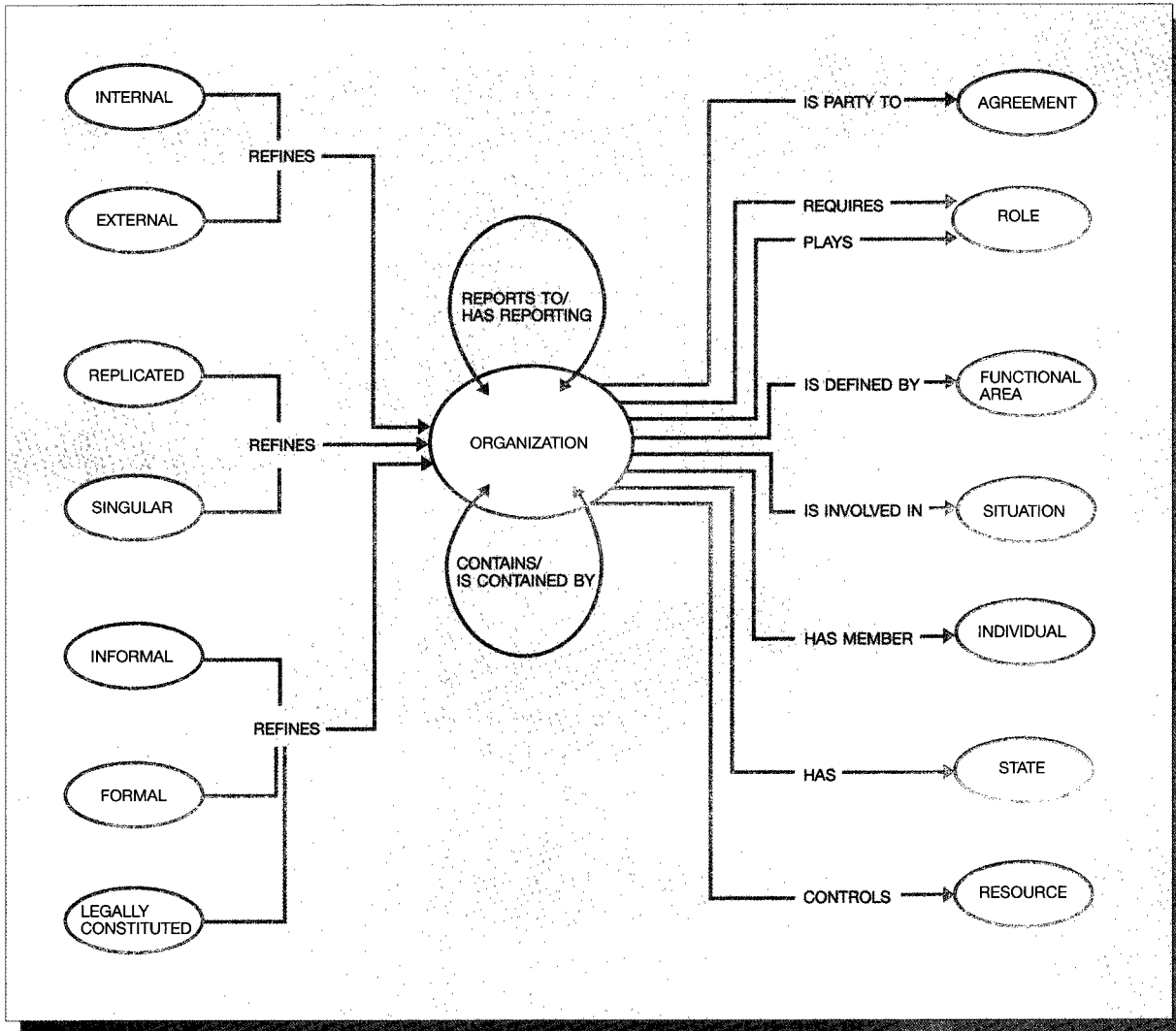
In the ICM above, the canonical idea of *product* is extended in various directions.

We could argue whether the central category that represents "product" for us is more prototypically a car or a box of cereal. There is no question, however, that the radial categories have something in common with the basic concept, while departing from it in various significant ways. Is a service a product? What about a leased 56 KB line? What about a monthly fixed-rate pricing scheme for a 56 KB communications circuit? Is documentation a product in its own right? How about information in the form of a financial derivative?

These concepts may be taken for granted and are essentially invisible to the people who harbor them. It is the task of the business language analyst to articulate this largely unspoken ontology and to ensure that the

information systems reflect the important concepts of the business users. In the second sidebar there is a discussion of how human beings form mental categories, and how this process extends into business concepts.

Figure 3 A fragment of the overall semantic network of generic business concepts



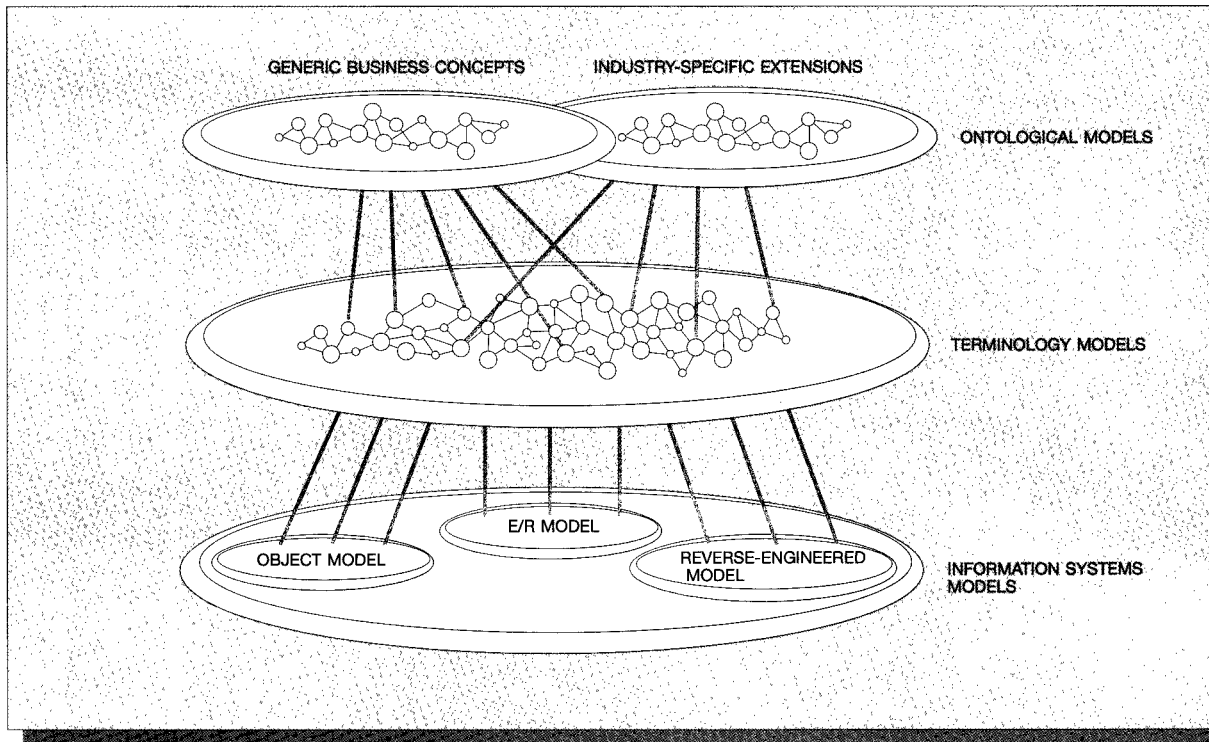
The following set of concepts arises from consideration of the kinds of things with which *any* business needs to concern itself:

- People—including both individuals and organizations
- Resources—material, energy, skills, money, and information
- Processes—events, end-to-end processes, functions, and discrete actions
- Results—the products and services that are the reason to be in business

- Locations—physical geography and logical points such as accounts and network addresses
- Time periods—the standard concept of time

This set is just one of many possible ways of dividing the conceptual space at a high level. Another published scheme divides the business world into resources, processes, and organizations.<sup>32</sup> Still another scheme has the following top-level set of divisions: party, contract or agreement, product, resource, event, location, and account.<sup>33</sup>

Figure 4 A generic concept network extended by a more specific network



There is no absolute best way to make this kind of classification. Concerns that appear at the top of one list are bound to appear elsewhere on someone else's list. For example, in our scheme, one type of resource is an information resource, and a type of information resource is a relationship, of which there are many that a business has to manage. One particular kind of relationship is a role that brings together individuals or organizations on one side and some function or set of functions on the other. Our scheme thus puts the concept of role three levels down from the top, whereas in another scheme, role might be at the very top of the conceptual taxonomy because it is such a powerful concept. Another type of complex relationship is a situation, which is an identifiable state of affairs that demands resolution. An academic discipline actually exists called situation theory, which forms the basis of its own logic system.<sup>34</sup> Clearly a case could be made that situation should be a first-order concept. The point is that information is immune to the law of gravity. The top is somewhat arbitrary.

Business concepts do not stand alone. Instead, they link together in naturally occurring patterns. These patterns appear in organizations of all kinds, across industry boundaries. Concept patterns form a semantic network<sup>35</sup> of interrelationships. Here are examples of typical concept relationships: Resources are transformed by processes that are triggered by events and invoke functions, discrete actions, and flows of material and information. People and organizations play various roles that are responsible for various functions. Processes create results, which in turn may become resources. Figure 3 is an example of one fragment of the overall semantic network of generic business concepts.

The paper does not present an exhaustive catalog of business concepts and their interrelationships, partly because of space limitations, and partly because, as noted above, there is more than one way to divide the conceptual space. More importantly, it is impossible to be exhaustive. As soon as we move into a more specific industry or enterprise environment, it becomes necessary to extend the generic concepts to account

for the information that is most important in that context. The top level of Figure 4 shows a generic concept network extended by a more specific network.

The main purpose for articulating the patterns of a business ontology is to provide a set of templates for organizing the specific terms that are encountered in the jargon of work groups and professional specialties. Through this linkage into templates, or patterns of conceptual relationships, the business terms themselves begin to form patterns of meaning and relationships that are unique to a specific business situation and community of communicators within a human activity system.

The set of terminology patterns forms a model of meaning that can be linked to various technical artifacts from the solution domain of information systems. This model provides traceability from implementation back to business meaning, and from unique domain language back to powerful generic templates. This complete set of linkages is shown in Figure 4.

**Activities and work products.** Business language analysis produces several modeling components and formal documentation to make these modeling components accessible. The work products range from documents and graphic pattern depictions to complex multidimensional semantic networks in appropriate repository technology.

Because business language is essentially a bottom-up analysis of an existing corpus of specific business terminology, the work is very detail-oriented. It starts with a large mass of language material that is provided or found in the environment. By determining definitions, applying existing patterns, and filling in new patterns of abstraction, we add detail to a higher-level framework to clarify and reduce the ambiguity of domain-specific language.

The following subsections describe the activities of business language analysis and their related work products. A small sample of language is employed from a hypothetical insurance company to illustrate some of the steps and results of a typical business language analysis.

*Gather language sources.* We can derive the patterns of language from several sources of business language. Some can be proactively developed sources: interviews, facilitated sessions, and questionnaires. The advantage of these techniques is that they involve people from the business in fostering discus-

sion, raising issues, and moving the group toward consensus. However, they make time demands on people who are already overworked, and they are limited by the memory and biases of a small group of individuals constrained by a time-box.

“Found” sources, in contrast, are documents and other materials produced by the business for its own use. They range from public pronouncements to proprietary items, and from formal to *ad hoc* documents. Examples include: requirements documents, business plans, product specifications, catalogs, training materials, regulatory filings, methods and procedures, process models, forms, charts of accounts, business plans, organization charts, business engineering and total quality models, contracts, and mission or vision statements. Often existing business documents prove to be the best sources of raw material for models because, in many cases, the material is not raw at all; it is already quite refined. Some existing information sources are well on their way to being models, worked over by many business minds in an attempt to reach consensus.

The example in Figure 5 is a single document fragment from an insurance policy, scanned and transformed via optical character recognition software from a paper copy of an insurance policy form. It is a section of the policy informing the policyholder of certain conditions of the contract related to designating and changing beneficiaries.

A work product from the gathering activity is an annotated bibliography of source documentation, along with notes and lists of participants for structured interviews and facilitated sessions.

*Extract business terms.* The next step after obtaining the sources of language is to identify the business terms they contain. Recognition of a business term becomes a matter of intuitive feeling for business language analysts. The search through the files and documents produces a list of terms. A fragment of such a list is shown in Table 1.

**Table 1** Partial list of terms from search

beneficiary	file	Automatic Mode of Settlement
request	Home Office	payment
form	survive	make a payment
needs	right to be paid	decide
take effect	paid	proof
designate	prior class	proof of identity
change	right to be paid	identity

Figure 5 Single document fragment from insurance policy

You may designate or change a beneficiary. Your request must be in writing and in a form that meets our needs. It will take effect only when we file it at our Home Office; this will be after you send the contract to us to be endorsed, if we ask you to do so. Then any previous beneficiary's interest will end as of the date of the request. It will end then even if the Insured is not living when we file the request. Any beneficiary's interest is subject to the rights of any assignee of whom we know.

When a beneficiary is designated, any relationship shown is to the Insured, unless otherwise stated. To show priority, we may use numbered classes, so that the class with first priority is called class 1, the class with next priority is called class 2, and so on. When we use numbered classes, these statements apply to beneficiaries unless the form states otherwise:

1. One who survives the Insured will have the right to be paid only if no one in a prior class survives the Insured.
2. One who has the right to be paid will be the only one paid if no one else in the same class survives the Insured.
3. Two or more in the same class who have the right to be paid will be paid in equal shares.
4. If none survives the Insured, we will pay in one sum to the Insured's estate.

Before we make a payment, we have the right to decide what proof we need of the identity, age or any other fact about any persons designated as beneficiaries. If beneficiaries are not designated by name and we make payment(s) based on that proof, we will not have to make the payment(s) again.

As we extract terms from the original source document, we can reduce the file by replacing found terms with surrogates, such as "\*\*\*". What we end up with looks like the skeletal remains shown in Figure 6.

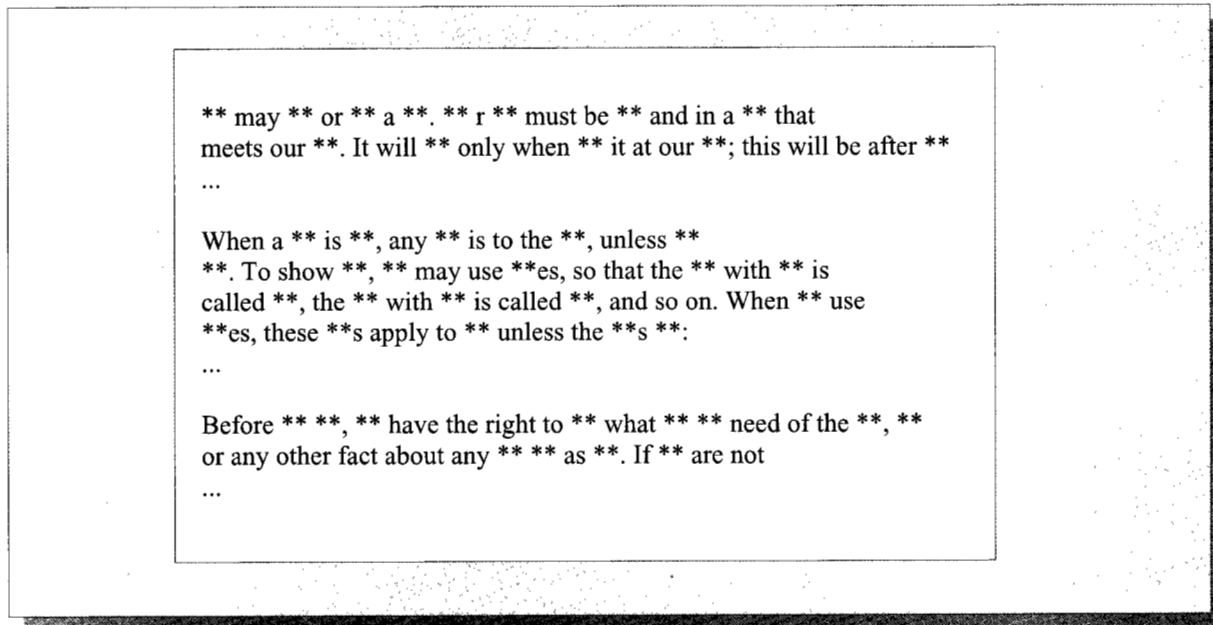
At this stage it is still possible to identify terms that may have been previously missed. In the stripped down text of Figure 6, we can identify at least two interesting terms that had not yet found their way into our list: "apply to" and "terms."

*Build glossary.* After alphabetizing and removing duplicate terms from the list, we can create a glossary with definitions. While building the glossary, it is particularly important to involve business experts—

those who actually know how terms are used and can identify and differentiate among different uses of the same word. Often glossaries that provide the raw material for the language analysis already exist in source documents. The following is a sample of glossary entries:

- **Beneficiary**—A person or other entity designated to receive benefits from an insurance policy upon the death of the insured
- **Proceeds**—The total amount paid out of an insurance policy upon termination of the agreement
- **Assignee**—A person or other party to whom benefits from an insurance policy are contractually assigned

Figure 6 Skeletal form of insurance policy fragment



- Interest—The type and quantity of benefits from a policy that are allocated to a particular party, as in “beneficiary’s interest”

*Classify terms.* Classification of terms begins to determine the basic shape of the information requirements that will need to be met by information systems. Areas of key importance will exhibit long lists of terms. This is a business-oriented demonstration of the Whorfian principle that the language shapes the thinking of its users. The concepts that are provided by a generic business ontology form the basis of this classification, but they will most likely need to be extended by concepts that are relevant and possibly unique to the particular business domain. An analyst can take a first cut at classifying terms, but business experts need to validate this work. Table 2 lists a set of terms extracted from the sample document previously described, classified by a very generic business ontology.

*Link terms.* Linkage among business terms sets up the meaning structures that help to build business object models (class hierarchies, object composition, variables, collaborations among objects). A set of relationships can be articulated for business terms, including linkage of terms to business concepts, link-

age of terms to each other via semantic relationships, and linkage of terms to sources in which they were found.

The next three figures provide an indication of how semantic linkage evolves as we think about a set of terms from a business source. They suggest the types of questions to be asked about each term that will allow us to understand the patterns of meaning in the business.

Figure 7 is a generic conceptual pattern. It says there is such a thing as an external role that we may expect to find. Any external role is likely to be either a source or a recipient, may be formal or informal, is played by an individual or organization, is involved in situations, and generates events.

In Figure 8, we have filled the slot in the center of the pattern with one of the terms that we found in our analysis of the document fragment. This directs attention to a set of questions, based on the fact that we have classified “beneficiary” as an “external role.” These questions cause us to go back to our term list to see whether we can find terms to fill the refinement, subtype, individual or organization, situation,

**Table 2** List of terms from sample document

<p><b>ROLES</b>  assignee  beneficiary  class 1 beneficiary  class 2 beneficiary  previous beneficiary  estate  Insured's estate  Insured</p> <p><b>ORGANIZATIONS</b>  Home Office</p> <p><b>INDIVIDUALS</b>  persons</p> <p><b>PHYSICAL RESOURCES</b>  form  statement</p> <p><b>MONETARY RESOURCES</b>  proceeds  payment  one sum  equal shares</p> <p><b>TIME PERIODS</b>  date of the request</p> <p><b>EVENTS</b>  death  Insured's death  request</p>	<p><b>AGREEMENTS</b>  contract  terms  interest  beneficiary's interest  right  right to be paid  subject to  priority  first priority  next priority</p> <p><b>INFORMATION RESOURCES</b>  class  numbered class  class 1  class 2  prior class  same class  relationship  apply to  identity  name  age  facts  proof  proof of identity  proof of age  proof of facts  request  needs  form  statement</p>	<p><b>PROCESSES</b>  make a payment  Automatic Mode of Settlement  change beneficiary  designate beneficiary</p> <p><b>ACTIONS</b>  send  show  state  ask  decide  endorse  file  know  owe  pay  take effect  end</p> <p><b>STATES</b>  based on proof  designated  living  not living  survive  paid  shown  stated  in writing  meets our needs</p>
--	--	---

and event slots that are indicated by the question marks in the figure.

Figure 9 shows the slots in the template filled in. Among the terms, there are clear-cut subtypes of beneficiaries called "class 1 beneficiary" and "class 2 beneficiary," and a refinement, "previous." The beneficiary role can be played by a person or by an estate. A term "request" may fill the event slot in this pattern, but we are going to go out on a limb and suggest that maybe it is a "claim request." We have also invented a term "death claim" to represent a situation that a beneficiary would be involved in. These suggestions made by the analyst will need to be validated by the business user, and may lead us to additional terminology that we have not discovered in the document.

Ideally, every term would be diagrammed to create semantic patterns like the one in Figure 9. Realis-

tically it is most important to create these diagrams for certain key terms that provide high leverage for understanding the domain of interest.

*Load semantic database.* It is easy to see from the small sample outlined above that analyzing business language leads to a complex, multidimensional network of terms, concepts, and meaning. Every way in which we try to portray this network on two-dimensional paper seems somehow inadequate. In the original text, terms can be easily overlooked. A simple list of terms is just a start. A glossary is more helpful but suffers from the circularity of definitions and the restriction of considering only one term at a time. Graphic linkages in accordance with predefined patterns help give more of a sense of the overall language and appeal to the visually oriented. They, however, are laborious to create and, in a large vocabulary, become overwhelming by their sheer numbers.

Figure 7 A generic conceptual pattern

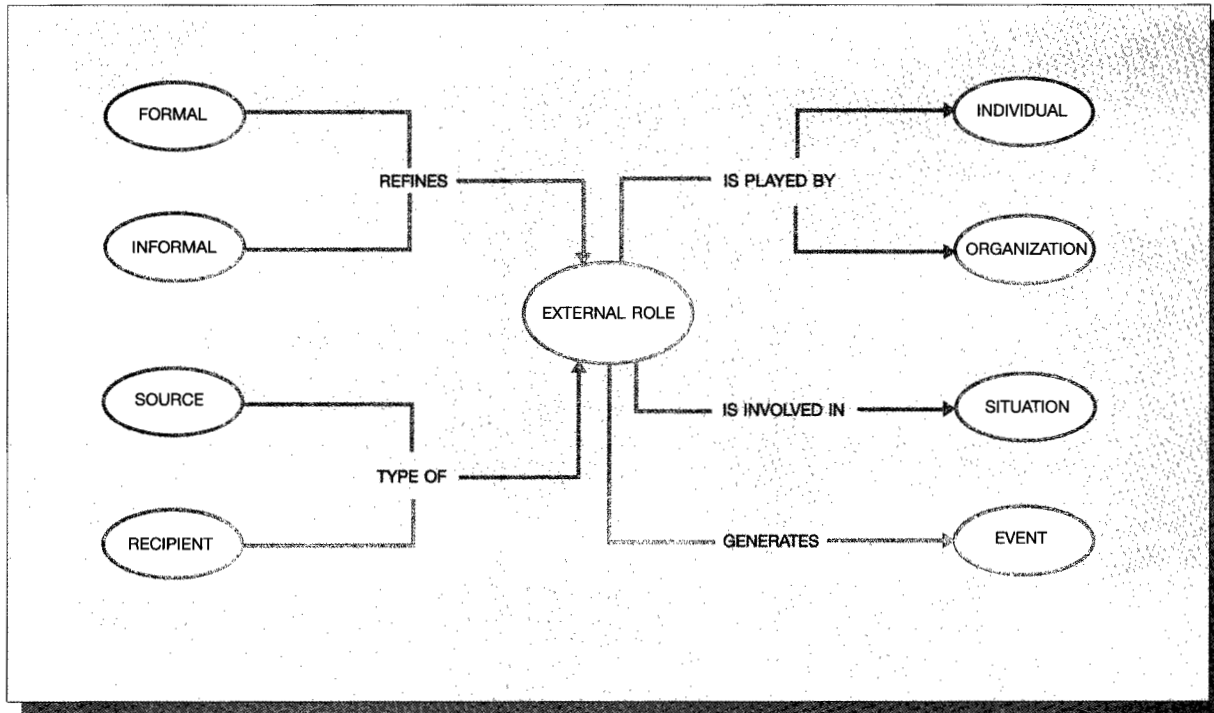


Figure 8 Center of pattern filled from document analysis

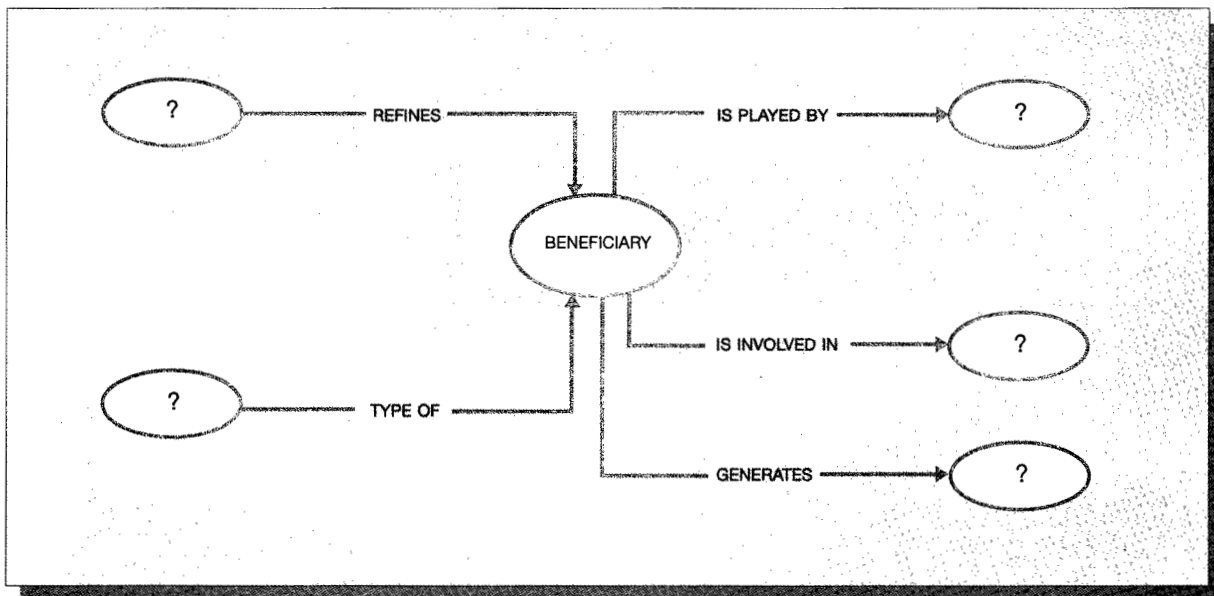
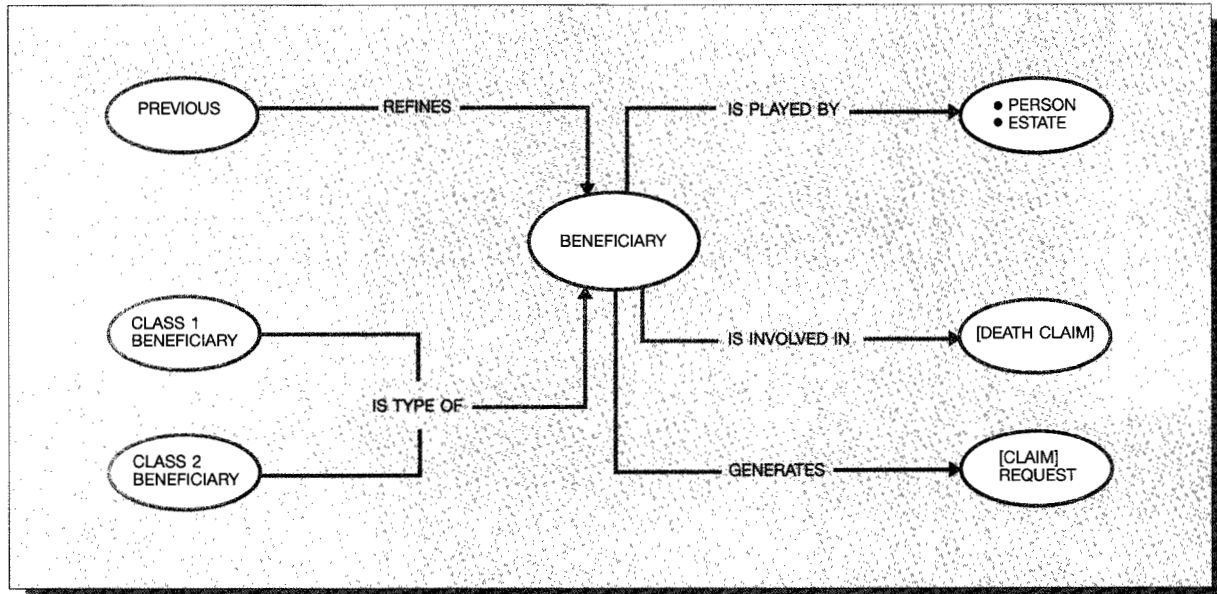




Figure 9 An ideal semantic pattern



A highly linked database can overcome most of these paper-oriented limitations by representing the terms, definitions, sources, linkage to concepts, and linkage to one another. Many products or technologies can support this requirement, including object-oriented databases, hypertext, or proprietary flat-file access methods. Also, a class of database management system specializes in capturing and maintaining multidimensional semantic networks.<sup>36,37</sup> Once in the database format, a multidimensional browsing tool mirrors the multidimensional data structure, so that all links from a specific term can be followed and displayed at the same time.

A repository of business terms, business concepts, definitions, sources, interterm linkages, concept-to-term linkages, and linkages between terms and design artifacts (object classes, database tables, etc.) can all be maintained dynamically as the models evolve. It is important to establish a data administration function to make sure that updates, backups, and data consistency matters are attended to.

*Overall documentation of results.* Throughout the process of creation and maintenance of the business language model, there are periodic points where it is useful to report results. A number of documents can serve this reporting requirement. Issues lists are working documents for the team that is performing

the business language analysis. A team member should be assigned to each issue so that there is responsibility for its resolution. A findings document is a simple listing of conclusions and implications that have emerged during the course of the analysis. Descriptive papers embed parts of the model in explanatory text.

**Roles, skills, responsibilities.** Broadly speaking, the people who do business language analysis are business modelers. Data modeling is a good background, as are other disciplines that involve classification, such as biology and library science. An academic background in linguistics, semantics, or systems theory would be ideal as preparation. Experience in building information systems, particularly object-oriented systems, provides the background to appreciate the benefits offered by business language analysis.

A modeler may work alone with documents and other language sources from the domain to produce a model. It is much more effective, however, if the analysis is done with a small team that searches sources for terms, writes definitions, classifies terms, writes documentation, and maintains the repository. It is essential that the team or individual modeler work with domain experts from the business to validate the definitions, relationships, and conclusions

that are developed in the course of the business language analysis.

It is not the role of business language analysts to dictate language, but rather to understand all the ways in which terms are being used and their implications for system requirements.

Business language analysis may be tied to a particular project in a constrained time period. However, it is more valuable if it becomes institutionalized as a permanent business function. At a certain point, the number of new terms being discovered will diminish because the effort is achieving completeness of coverage. This point provides the opportunity to evaluate the completeness and adequacy of the enterprise-wide information system.

The language of the business will continue to evolve. With a highly tuned sensing mechanism, the information systems organization can stay abreast of the evolution of the business, as reflected in the evolution of its language.

### **Information systems use of business language analysis**

The models of terminology produced by business language analysis have a central role in many key activities throughout the information systems development environment. This role starts with an understanding of the nature of the information systems development process itself. Popular life-cycle descriptions (waterfall, spiral, etc.) give the impression that building enterprise-class information systems is a downhill effort, somehow aided by gravity or some other natural force. If business and business environments were static, this might be the case. In fact, just the opposite is true: it is an uphill struggle against the forces of entropy.

A method that is grounded in the specific information access and communication needs of the human activity system and that is also informed by powerful concepts from general systems theory, linguistics, and cognitive science provides some hope of winning this struggle. This paper recognizes information systems as serving systems for living organizations. This point of view challenges the application-oriented and project-oriented approach to information systems and affects many information systems functions.

**Planning.** We can start with the planning function. The living systems viewpoint lays the groundwork

for a new perspective on planning the evolution of the nervous system of the organization. The planning function will understand that it is not enough to provide disjoint inventory management, order processing, billing, and accounting systems, with decision support systems coming along as an afterthought. All of these must be present and coordinated. There must be visibility into information that is generated beyond the boundary of the enterprise itself. The planning process should search for areas of isolated or missing information capability and budget for work to address them. It can use existing language models to determine the completeness of coverage and may commission additional language modeling to fill in the gaps.

**Architecture.** Architecture now also takes on a different meaning. If the goal is to support the overall communication capability of the business as a living system, then architecture begins to look like putting an infrastructure in place that will support the needs of the organization for normative and long-term decision-making, not just the standard operational and control functions. By means of understanding the interactions among individuals and organizations performing roles within the business, application architects can gain a better, more effective understanding of the business functions to be supported. An architecture based on cooperating agents will change the whole notion of business applications. A detailed understanding of the language and concepts of the enterprise gives strong guidance to the types of software agents that need to be put in place to realize its adaptive goals.

**Project management.** Business language analysis must be managed as carefully as any other analysis effort, through continuing dialog between language analysts and project management. It is often attractive to follow threads of language into areas that can expand the scope of projects in an uncontrolled manner. At the same time, language analysis can lead to an expansion of scope that is appropriate and may have been overlooked without this analysis. A language model provides a means to reach agreement, in terms familiar to the user, on which functions, roles, and resources will be in or out of the scope of a particular project.

Business language analysis provides strong support for the process of team building. The focus on business language prepares for many other development activities by unfreezing the "techno-speak" that many team members may use. The process of analyzing

business language helps to mediate among many different communities: executive-to-line management, various functional organizations, supervisors-to-users, IS personnel-to-non-IS personnel, and even company-to-company, in the case of cooperative or consortium efforts. It also lays a foundation for training in new procedures and system support.

**Design.** Business language analysis does not replace system design. Business language analysis is a *discovery* activity that helps to understand information needs within the human activity system. Design is a *creative* activity that uses language models as input. "The simplistic approach is to say that object-oriented development is a process requiring no transformations, beginning with the construction of an object model and progressing seamlessly into object-oriented code. . . . While superficially appealing, this approach is seriously flawed. It should be clear to anyone that models of the world are completely different from models of software. The world does not consist of objects sending each other messages, and we would be seriously mesmerised by object jargon to believe that it does."<sup>38</sup>

A domain object model is a design of the business objects within an implementation. Business objects are generally distinguished from purely technical objects such as graphical user interface (GUI) frameworks and data broker middleware. Several techniques guide designers in making the transition from modeled language to an object perspective:

- The generic concept framework provides a first-cut set of classes, subclasses, and collaborations that can be assumed to exist in some form in almost any domain.
- Terms that map to the same high-level ontological concept should be considered for possible subclassing or parameterization of a generic class.
- Terms that are made up of a basic term and modifiers can be considered for hierarchy or variable constructs, depending on whether the modifiers indicate strong typing or state changes.
- Predominance of terms in one concept or another reveals or confirms the nature of the overall system to be built. Predominance of resource-oriented terms reveals an inventory type of system, whereas predominance of role and process terms indicates a work-flow orientation.

The language model can also contribute to database design. Database design issues are very similar to the sets of concerns that lead to an effective domain ob-

ject model. In particular, if an object-oriented database management system is to be used, these sets of concerns are highly convergent. If relational technology is to be used, the techniques noted above will need to be mapped onto an entity, attribute, and foreign key paradigm.

Business language analysis has a special relationship with use case modeling. Use cases have long been associated with the idea of a concept catalog or glossary. This listing of terms and concepts from the business domain has been an important communication mechanism for project teams and the user community. Business language analysis elaborates on the notion of a concept catalog, turning it into a model in its own right. In return, the process of building use cases of how the system will be used in the new environment becomes a rich source of terms and concepts that may not have been previously discovered in documents. Part of the reason for this is that these new terms may reflect a future scenario that has not otherwise been well documented by the business.

Clearly, other types of models beyond language models are required for effective systems design. For example, event and traffic metrics feed the physical design of systems.

**Development.** Given that information systems development is an ongoing effort to create a systemic capability, it is accomplished in stages, but always with an eye on the overarching needs of the organization as a whole. Language analysis benefits each increment that is delivered, based on the ability to reuse term and concept patterns as powerful abstractions. Reuse results in stronger, previously tested code, and a shorter development life cycle. The benefit to the development process as a whole is traceability—establishment and maintenance of linkage between project artifacts and their business sources of justification.

A business language model provides key support for the development of user interfaces. Terms from the natural business language can be brought to the surface of the interface, where they provide a feeling of familiarity for system users. Underlying code can be wrapped in alternative terminology for different communities of users and can evolve as the language of the business evolves.

The information system to support an enterprise is never complete. The application perspective regards this phenomenon as a problem that gives rise to a

separate maintenance function. In contrast, the living systems viewpoint recognizes this phenomenon as natural evolution. Developers in the next incremental project have an established nucleus of de-

---

**Language models can reveal  
the variable importance of the  
same concept from one domain  
to another.**

---

veloped and understood business terminology and meaning from which to expand seamlessly into new areas of the business.

**Testing.** A number of issues involve testing software, including functionality, usability, accuracy, consistency, and efficiency. A language model provides guidance to testing experts in the construction of meaningful test cases that assure the provision of needed functionality.

**Documentation.** Writers of both users' manuals and on-line help can benefit from organized language that represents the roles, processes, events, and resources of interest to the users of information systems. Context-sensitive help can be driven directly from the terminology that is modeled and understood via business language analysis. The key advantage is that documentation and help is in the user's language.

### **Conclusions to date**

Business language analysis has been applied by the author to a number of situations in a variety of industries and organizations. This experience has led to some interesting lessons and conclusions.

A common experience is that in any business domain we are likely to encounter a predominance of certain categories of information. These dominant categories lead to the addition of more specific concepts to the ontology in order to differentiate sets of terms that would otherwise form a long list under a single concept. One such concept expansion was the result of an analysis of the budget office at a state university. The budget office dealt almost exclusively with

information in various forms. The office collected information from other departments, performed various types of analysis, and created a variety of reports and information products for use by departments throughout the university. This situation required a major expansion of the concepts related to information resources.

Information resource concepts were also added during an analysis of language within the project management function in a consortium. The emphasis on coordinating work among several participating companies and a large number of suppliers, and managing projects that spanned multiple years, presented a severe information management challenge.

Based on the two examples presented above, the concept of information resources was expanded to include the following concepts: identifiers, motivations (including values, opinions, purposes, conditions), proposals, decisions, rules (prescriptive, proscriptive, allowances, entitlements), descriptions, templates (including specifications, forms, models, checklists), characteristics, measurements (quantitative, qualitative, comparative), category sets, commitments, goals, history, relationships (including roles, situations, agreements [contractual and informal]), forecasts, and plans. This set of concepts still may not be totally exhaustive. However, once new concepts have been established within one domain, they become available in any subsequent domain where they might apply.

Another expansion and validation of the generic business concept structure came from an analysis of a customer relationship management project at a natural gas utility company. It was not surprising that this domain forced an expansion of the concepts related to energy resources. What was surprising was the need to greatly expand the granularity of concepts related to time periods. Over 1200 terms were found in the analysis, and close to 20 percent had to do with discrete points and ranges of time. Time itself was already included in the generic set, but this experience validated its value as a significant ontological concept.

Sometimes there are important business concepts that exist in a domain but do not have explicit terminology that maps cleanly to them. An example of this insight is found in the insurance industry. The analysis of insurance has led to the conclusion that the essence of the business is the management of situations. Actuarial analysis is largely about recognizing

ing distinct situation types in which businesses and individuals can find themselves and determining the likelihood of various outcomes resulting from types of situations. Even though insurance people recognize this truism, and find it is a useful way to think about their business, there is a surprising lack of specific terminology that relates to client situations.

Language models can reveal the variable importance of the same concept from one domain to another. Models from two different companies indicate the importance of contractual agreements in the insurance business. In fact, many insurance terms are classified under both the concept of agreement and the concept of product, because a policy, which is a contract, is actually the basic product of the insurance industry. Without contracts, there is no business. This situation is contrasted with a model done for a cellular telephone company, where there are very few terms that refer to contracts. In this case, contracts are rather casual pieces of paper that are signed upon commencement of service, and as many as 40 percent never make it from the retail distributor back to the appropriate corporate file. Service, billing, and collections proceed unimpeded, so that contracts are truly not a major issue. These diverging models provide a strong indication of the types of objects needed by the respective industries.

A different kind of lesson from experience with business language analysis is the positive reaction that it evokes in business people. Significant insights always seem to arise, and great appreciation is expressed for this fresh view of language. There is gratitude that information systems professionals are willing to spend time to appreciate the unique meaning that infuses the language of the business. There are also surprises for the domain experts at times. A model prepared for an internal IBM group highlighted terminology from a mission statement that everyone had agreed to change but that was still present in source documents. Many in the group were shocked when the model highlighted language that had become invisible to the participants in the business.

Information systems professionals who have been exposed to this approach are almost unanimous in their positive reaction. The most common reaction is "If only we had followed this approach on my last project! It would have saved untold misunderstanding and rework." They recognize that a detailed understanding of language avoids a number of common problems with information system development. These problems include the cost of reworking inad-

equated requirements, the loss of credibility when delivered systems do not match the needs of the business, the risk that projects will be so focused on the data processing "plumbing" that human communication and information needs will not be served, and the risk that analysts will drift off into a haze of abstractions that are too loosely coupled with the needs of the business. The ultimate risk is that the form and operations of the business will be forced to conform to the resulting information system, instead of the other way around.

### Outlook

Several areas need further refinement and expansion of the usefulness of business language analysis.

We have talked a lot about generic and industry-specific concept patterns. These concept patterns form a metalanguage of business concerns, which are proven to help understand specific bodies of language. Coupled with robust repository technology, this ever-expanding semantic network of concepts and terminology can form a rich index for an asset base of software components. This index helps to address the issue of visibility of design and code artifacts from earlier projects where it is often difficult to determine what an object does, and where local terminology is not embodied in objects whose genesis is elsewhere.

The subject of patterns is currently a very intense topic among object-oriented developers. Design patterns have been the subject of Internet discussions and a growing published literature. These patterns were originally limited to technical design issues, such as structure, behavior, and creation of software objects.<sup>39</sup> Such patterns are in contrast to the types of patterns that emerge from business language analysis, which are patterns of meaning. Recently there is an indication of possible areas of cross-pollination with work such as Peter Coad's business object patterns<sup>40</sup> and Ward Cunningham's CHECKS pattern language that validates domain-specific input.<sup>41</sup>

Another growing area of software development is the field of groupware and workflow software. This field was pioneered by individuals for whom computers and cognition were quite compatible.<sup>42</sup> The field has expanded and become increasingly commercial, although it still has a long way to go to realize the full-blown "mirror worlds" potential.<sup>43</sup> As software comes to draw increasingly on repositories of structured business language, sophisticated group-

ware applications will provide more transparency and appeal to users across the enterprise.

The software crisis is still with us. Demand is increasing, and backlogs are seemingly hopeless. Object technology provides part of the promised solution. As Thomas Love envisioned, "These new environments for assembling powerful components will still require lots of creative programmers to build new and better components and make them available to the market. Programmers will become software component providers; users will construct the final applications and systems based upon the available repertoire of components."<sup>44</sup> If, in addition, these same sophisticated users have access to rich repositories of structured business meaning, software and language can begin to come together in intuitive and seamless support of business evolution.

### Acknowledgments

I would like to acknowledge the invaluable support and contributions made by the following colleagues, without whom this paper would be much less than it is: Rock Angier, Carl Ballard, Nancy Boyd-Schimmelman, Ken Briskey, Dave Britton, Jane Conkey, Robert Coyne, Robert Griffith, Robert Gross, Ralph Hodgson, Steve Johnson, Chris King, Mark Langman, Jim Nerney, Steve Marcus, Rich Newbold, Jack Ring, Zach Shoher, Mike Straka, David Taylor, Anne Wheeler, Lynn Wheeler, and Kathy Yglesias.

### Cited references

1. R. Stamper, "Social Norms in Requirements Analysis—An Outline of MEASUR," in *Requirements Engineering: Social and Technical Issues*, Editor, M. Jirotko, Academic Press, London (1994), p. 107.
2. D. W. McDavid, "Business and Systems Planning: Building a New Alliance," *Database Programming and Design* 5, No. 10, 29–39 (October 1992), p. 30.
3. J. A. Zachman, "A Framework for Information Systems Architecture," *IBM Systems Journal* 26, No. 3, 276–292 (1987).
4. J. Martin, *Strategic Information Planning Methodologies*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1989).
5. J. F. Sowa and J. A. Zachman, "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal* 31, No. 3, 590–615 (1992).
6. S. H. Haeckel and R. L. Nolan, "Managing by Wire," *Harvard Business Review* 71, No. 5, 122–132 (September–October 1993), p. 122.
7. T. H. Davenport, "Saving IT's Soul: Human-Centered Information Management," *Harvard Business Review* 72, No. 2, 119–131 (March–April 1994), p. 119.
8. P. Checkland, *Systems Thinking, Systems Practice*, John Wiley & Sons, Inc., New York (1981), p. 13.
9. *Ibid.*, p. 209.
10. E. Reichtin, *Systems Architecting: Creating and Building Complex Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1991).
11. P. Checkland, *op. cit.*, p. 149.
12. J. G. Miller, *Living Systems*, McGraw-Hill, New York (1978).
13. B. Clemson, *Cybernetics, a New Management Tool*, Abacus Press, Turnbridge Wells, Kent, England (1984).
14. P. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday, New York (1990).
15. G. Morgan, *Images of Organization*, Sage Publications, Newbury Park, CA (1986), p. 356.
16. M. Rothschild, *Bionomics: Economy as Ecosystem*, Holt, Rinehart, and Winston, Inc., New York (1990), pp. 335, 350.
17. K. Kelly, *Out of Control: The Rise of Neo-Biological Civilization*, Addison-Wesley Publishing Co., Reading, MA (1994), pp. 1, 188, 194.
18. D. Taylor, *Mind*, Simon & Schuster, New York (1982).
19. M. Minsky, *The Society of Mind*, Simon & Schuster, New York (1986).
20. A. Trehub, *The Cognitive Brain*, MIT Press, Cambridge, MA (1991).
21. A. Newell, *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA (1990), p. 92.
22. M. Lorenz, *Object-Oriented Software Development: A Practical Guide*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1993), p. 41.
23. I. Jacobson et al., *Object-Oriented Software Engineering: A Use Case Driven Approach*, ACM Press, New York (1992).
24. J. Rumbaugh et al., *Object-Oriented Modeling and Design*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1991), p. 153.
25. D. De Champeaux, D. Lea, and P. Faure, *Object-Oriented System Development*, Addison-Wesley Publishing Co., Reading, MA (1993), p. 34.
26. P. Coad and E. Yourdon, *Object-Oriented Analysis*, Yourdon Press, Englewood Cliffs, NJ (1990), p. 61.
27. I. Graham, *Migrating to Object Technology*, Addison-Wesley Publishing Co., Wokingham, England (1995), p. 301.
28. G. Booch, *Object-Oriented Analysis and Design with Applications*, 2nd ed., Benjamin/Cummings Publishing Co., Redwood City, CA (1994), p. 150.
29. D. A. Cruse, *Lexical Semantics*, Cambridge University Press, New York (1986), p. 112.
30. P. Checkland, *op. cit.*, p. 115.
31. G. Lakoff, *Women, Fire, and Dangerous Things*, University of Chicago Press, Chicago, IL (1987).
32. D. Taylor, *Business Engineering with Object Technology*, Wiley, New York (1995).
33. G. Simson, *Data Modeling Essentials: Analysis, Design, and Innovation*, Van Nostrand Reinhold, New York (1994).
34. K. Devlin, *Logic and Information*, Cambridge University Press, New York (1991).
35. A. Barr and E. A. Feigenbaum, *The Handbook of Artificial Intelligence*, Vol. 1, William Kaufmann, Los Altos, CA (1981), pp. 180–189.
36. R. Griffith, "Conventions for Semantic-Network Databases," *Second Meeting in Science and Technology in Informatics*, Rio de Janeiro, Brazil (November 1989).
37. R. Griffith, "Semantic-Network Databases for VLSI Design," *British National Conference on Databases*, Number 7, Edinburgh, Scotland (July 1989).
38. S. Cook and J. Daniels, *Designing Object Systems: Object-Oriented Modeling with Syntropy*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1994), p. 6.
39. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Publishing Co., Reading, MA (1995).

40. P. Coad et al., *Object Models: Strategies, Patterns, and Applications*, Prentice Hall, Englewood Cliffs, NJ (1995).
41. J. O. Coplien and D. C. Schmidt, Editors, *Pattern Languages of Program Design*, Addison-Wesley Publishing Co., Reading, MA (1995), pp. 145-155.
42. T. Winograd and F. Flores, *Understanding Computers and Cognition*, Addison-Wesley Publishing Co., Reading, MA (1986).
43. D. Gelertner, *Mirror Worlds*, Oxford University Press, New York (1991).
44. T. Love, *Object Lessons: Lessons Learned in Object-Oriented Development Projects*, SIGS Books, New York (1993).

*Accepted for publication January 8, 1996.*

**Douglas W. McDavid** IBM Object Technology Consulting, 6300 Diagonal Highway, Boulder, Colorado 80301 (electronic mail: [mcdavid@netcom.com](mailto:mcdavid@netcom.com)). Mr. McDavid is a senior consultant with IBM's Object Technology Practice, specializing in business semantic modeling in support of object-oriented systems development. For more than 25 years, he has worked on and managed projects in information management, information systems, and application development, with in-depth knowledge of the telecommunications, insurance, brokerage, banking, and information services consulting industries. Among his accomplishments, he introduced reverse engineering of legacy information systems into a large property and casualty insurance carrier, a premier discount brokerage firm, and a specialized personal banking institution. Prior to moving into data processing, Mr. McDavid had an extensive career in the library field, managing reference and circulation departments, a branch library, and a central library. He has been an innovator in the use of modeling to align business and information technology strategies. He has worked with various technologies, including object-oriented languages, semantic network databases, and simulation software to improve corporate executives' ability to "manage by wire." Mr. McDavid is a member and director of Data Administration Management Association International. He received his B.A. with honors in sociology from the University of California, Santa Cruz and his M.A. in library science from San Jose State University.

Reprint Order No. G321-5598.