Chapter 4

Databases and Systems

The computer, by providing mass storage, display, interactivity and social networking affords many advantages over more traditional means of organizing knowledge. Although books are more intimate, magnetic and optical storage achieve a higher information density and reliability. Interactive displays allow for real-time visualization and navigation of information. Digital networks provide a global system for the communication and exchange of information. While these technologies may never replace the experience of reading a book the computer can be a useful tool for research in many other ways.

The computer has led to a number of widely used information organizing technologies including digital libraries, on-line encyclopediae, databases, semantic networks and the world wide web. In addition, current research suggests several promising areas for the next generation of knowledge system, including object-oriented databases, graph-oriented databases and the semantic web. These systems for knowledge organization, and their individual benefits and drawbacks, will be explored more completely in this chapter. Finally a new system, Quanta, will be proposed that incorporates some of the features of each.

4.1. Ideal Systems

The potential for machines to become knowledge organizing tools was realized even before their construction. These images of the future provide an interesting perspective on what we expect computers should be able to do. In chapter one we examined the *memex* developed by Vannevar Bush in 1945, in which he describes a fictional system (at the time) capable of "bringing files and material on any subject to the operator's fingertips." [4-1]. In 1965, years before the first digital library catalog, J.C.R. Licklider described what would be desired of a digital system capable of large-scale knowledge organization:

> "A basic part of the over-all aim for precognitive systems is to get the user of the fund of knowledge [a knowledge system] into something more nearly like an executive's or commander's position. He will still read and think and, hopefully, have insights and make discoveries, but he will not have to do all the searching himself nor all the transforming, nor all the testing for matching or compatibility that is involved in creative use of knowledge. He will say what operations he wants performed upon what parts of the body of knowledge, he will see whether the results make sense, and then he will decide what to have done next." [4-2]

The science fiction author, Isaac Asimov, in his Foundation series describes the "Galactic Encyclopedia" as a repository capable of maintaining the knowledge and history of an entire galaxy for trillions of people over tens of thousands of years [4-4]. On November 20th, 1936, H. G. Wells read his ideas for a World Encyclopedia at the Royal Institution of Great Britain Weekly Evening Meeting:

> "This World Encyclopedia would be the mental background of every intelligent man in the world. It would be alive and growing and changing continually under revision, extension and replacement from the original thinkers in the world everywhere. Every university and research institution should be feeding it. Every fresh mind should be brought into contact with its standing editorial organization. And on the other hand its contents would be the standard source of material for the instructional side of school and college work, for the verification of facts and the testing of statements - everywhere in the world." [4-4]

The above descriptions present a vision of knowledge organization as the large scale logical manipulation of symbols. However the ultimate purpose of knowledge tools is the same as other educational tools, to encourage curiosity, enhance learning and promote understanding in the individual. Charles Van Doren discovered an encyclopedic treatise titled *L'Encyclopedie francaise,* intending to answer the question: "What is an encyclopedia?" Doren summarizes:

> "The aim of the typical American encyclopedia is to inform, *rensigner*; it intends to make known, not to make comprehensible. Even if it is true that this kind of book is a great educational tool, it is so only secondarily. Education is too important to be a mere secondary end. More than that, education involves understanding that some things are more important than others." [4-5]

A knowledge system, such as a library, encyclopediae or computer, not only embodies the facts of a culture, it should also illuminate the researcher while not inadvertently get in the way of personal progress.

4.2. Existing Systems

This thesis is intended to be a study of new technologies based on the technical and aesthetic integration of existing ones. While the story of human knowledge extends in time from speech, to writing, to libraries, to print, to the world wide web, a detailed account of this history and the social influence of these many systems is only addressed briefly here. Instead, the emphasis will be placed on how these different systems may contribute to an integrated solution.

4.2.1. Libraries

Libraries were the first human efforts to collect knowledge from different peoples and cultures in one place. In 1980, archaeologists discovered a royal palace at the ancient site of Ebla in Syria. In addition to manuscripts, the find included the remains of tablets dated around 2000 B.C which listed the contents of other tablets - in effect, perhaps the first library catalog. Libraries grew in size and expanded throughout time [4-6]. From written catalogs the card catalog was developed, along with guidelines for their use [4-7]. After the invention of the computer the MARC standard, Machine-Readable Card Catalog, was developed by Henriette Avram and colleagues at the Library of

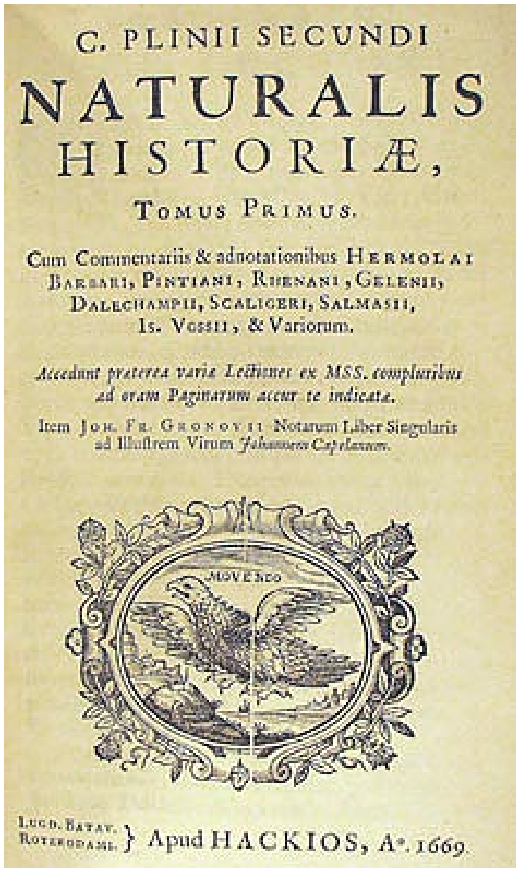
Congress to distribute card catalogs on magnetic tape and has become the international standard for digital card catalogs.[4-8].

The first experiments to provide the full text of documents on-line took place in the 1980s with the Mercury project at Carnegie Mellon and the CORE project at Cornell University. These provided the first on-line access to scientific journals along with images [4-9]. In the commercial sector, Bartleby.com started on-line publishing in 1993 and now offers hundreds of full texts of classical literature for free. In 2004 the company Google, Inc., by collaborating with libraries and publishers, announced its Library Project aimed at providing full text searching of books [4-10].

National digital library efforts are also underway, including the National Sciences Digital Library (NSDL) and the National Digital Information Infrastructure and Preservation Program (NDIIPP). Some of the challenges include technology infrastructure, publisher support, economic expectations (the public is accustomed to public libraries, and the internet, being free), the digitization of physical libraries, and the need for social collaboration. The primary benefits of digital libraries are government supported infrastructure and the potential to be a global national resource. The transition to digital libraries and their potential benefits over physical libraries is further explored in a comprehensive book by William Arm's titled *Digital Libraries* [4-9].

Current efforts by the Library of Congress are not so much about a particular technology as they are about combining many different existing technologies [4-11]. While this has its advantages, there are also significant problems with integrating systems that were not originally designed to work together. Many of these technologies are based on the Internet, which as we will see, was itself established from a fusion of several different standards.

A global, well-organized, interactive digital library is a unique yet still unrealized vision. The details of its construction, operation, interface, participants and mechanisms for collaborative contribution are still being explored and debated. Many of these issues revolve around solving technical challenges that will be discussed here.



C. PLINII SECUNDI
NATURALIS
HISTORIÆ,
TOMUS PRIMUS.
Cum Commentariis & adnotationibus HERMOLAI
BARBARI, PINTIANI, RHENANI, GELENII,
DALECHAMPII, SCALIGERI, SALMASII,
IS. VOSSII, & Variorum.

Accedunt præterea variæ Lectiones ex MSS. compluribus
ad oram Paginarum accurate indicatæ.
Item JOH. FR. GRONOVII Notarum Liber Singularis
ad Illuſtrem Virum *Johannem Capellanum.*

LUGD. BATAV.
ROTERODAMI. } Apud HACKIOS, A°. 1669.

Figure 4.1. Pliny's Naturalis Historia, an encyclopedia in thirty seven volumes of the natural world, 77 CE.

4.2.2. Encyclopediae

Sir Thomas Browne was perhaps the first to use the word encyclopediae in 1646 in his work *Pseudodoxia Epidemica* (Vulgar Errors). This was a collection of knowledge from many areas created to refute commonly presumed truths [4-12]. The goal of summarizing knowledge, however, can be traced to more ancient times. One early example is Pliny's Naturalis Historia (Figure 4.1) first compiled in 77 CE. and consisting of 37 books:

Table 4.1. Chapter organization in Pliny's Naturalis Historia, 77 CE.

Chapters	Contents
I	Preface and tables of contents, lists of authorities
II	Mathematical and Physical description of the world
III-VI	Geography and Ethnography;
VII	Anthropology and human physiology
VIII-XI	Zoology
XII-XXVII	Botany
XXVIII-XXXII	Pharmacology;
XXXIII-XXXVII	Mineralogy (including applications to art, casting, painting and modeling)

Through history a large number of encyclopediae have been developed. Unlike libraries, which hold knowledge for many people, encyclopediae have the goal of summarizing knowledge to the individual reader. First published in 1768, the Encyclopedia Britannica was compiled on a new plan that included integrating alphabetic entries for both arts and science rather than keeping disciplines is separate volumes [4-13].

The first digital encyclopedia was Microsoft's Encarta from 1994. Microsoft originally approached Britannica, but was turned down because it was believed digital materials could not compete with the high quality and editorial value of printed editions. Instead, Microsoft purchased *Funk and Wagnalls* and launched Encarta. By 1996, Britannica was unable to compete with Encarta in sales. The current version of Encarta contains 65,000 articles [4-14]. Since 1996, Encyclopedia Britannica has been sold and restructured and now contains 120,000 articles in both print and digital versions [4-15].

A unique new encyclopedia is Wikipedia, self-described as "the free encyclopedia", it was founded on the free software movement [4-16]. It is an encyclopedia which allows the general public to upload content and edit articles, thus favoring public improvement over established authors. Because of its continual development Wikipedia is surprisingly capable of keeping up to date in certain areas [4-17]. Drawbacks are starting to emerge, however. First, unlike encyclopedia that rely on established authors, its content on historic events may not be as accurate. Secondly, when viewpoints differ there can often be a public battle over the contents of an article. While Wikipedia sees this as a positive feature that encourages resolution, the resulting articles do not necessarily reflect truth. Finally, while modification of content is free, the submission of content is still subject to copyright law so that published material is only as good as the authors who contribute. Finally Wikipedia administrators, selected by invitation, hold ultimate control over policy regarding content and can delete whole articles without notification.

In many ways, Wikipedia achieves H.G. Wells' goal of being "alive and growing and changing continually under revision" [4-4]. While the Wikipedia audience is large, it does not collect material from "all of the scholarly minds" of the world and therefore must be supplemented with other resources. Both printed and digital encyclopedia are useful in certain circumstances as reference materials, but it is difficult for them to compete with the history found in university libraries as in-depth research tools.

Encyclopediae differ from libraries in that they are a summary of the contents of the latter. However, the technology that may be used for a general encyclopedia is the same as that which would be used for a digital library. Both are based on current trends in internet technology, as the internet is seen as the ultimate interface for digital libraries and on-line encyclopediae. The internet, at least technically, is therefore the driving force behind future global, integrated knowledge systems.

4.2.3. The Internet

Preceded by disjoint digital networks, a key shift toward global networks took place in a 1960 paper by J.C.R. Licklider, titled a Man-*Machine Symbiosis*:

> "Man-computer symbiosis is a subclass of man-machine systems. There are many man-machine systems. At present, however, there are no man-computer symbioses...The hope is that, in not too many years, human brains and computing machines will be coupled together very tightly, and that the resulting partnership will think as no human brain has ever thought and process data in a way not approached by the information-handling machines we know today." [4-18]

Shortly after this presentation, Licklider joined the Advanced Research Projects Agency (APRA) where he developed a group to work on what was called the "Intergalactic Network". The problem was how to physically connect computers to form a single network. The solution, packet switching, allows messages to be broken up so that the partial "packets" can take alternate routes to their destination [4-19]. See Figure 4.2.

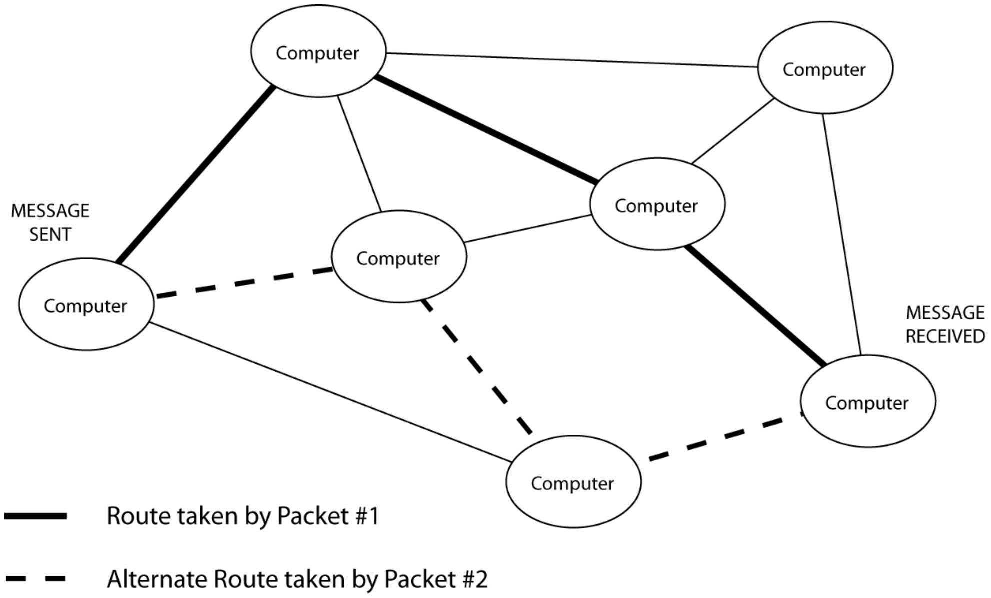


Figure 4.2. Packets take alternate routes to form a completed message

In 1969, Figure 4.3, the first four nodes connected to ARPANet included the University of California, Los Angeles, the Stanford Research Institute, the University of Utah, and the University of California Santa Barbara [4-19]. It is useful to point out that the technological advancement at this stage is primarily communicative while storage, access and resource retrieval problems associated with networked documents were not yet an issue.

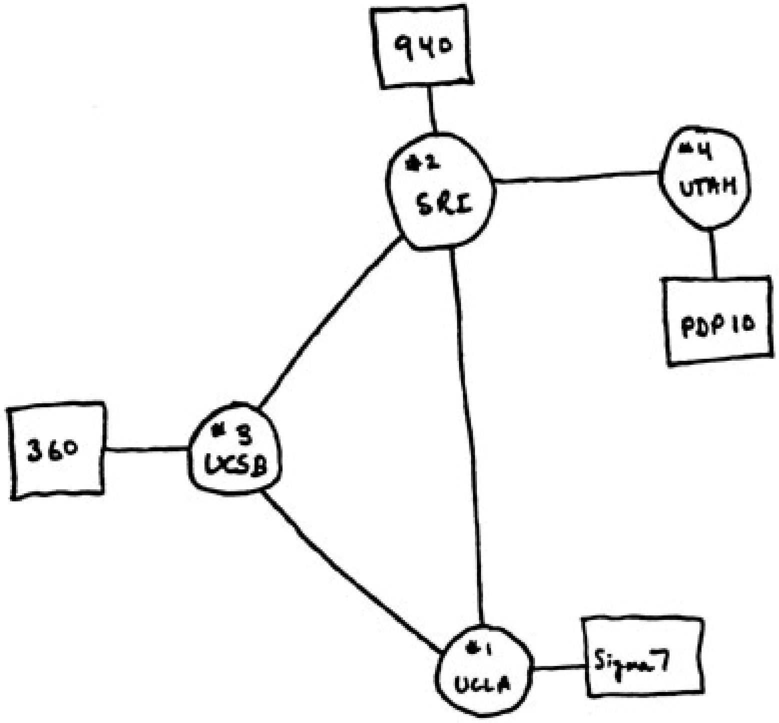


Figure 4.3. First four nodes of the ARPANet, 1969

At the Department of Advanced Research Projects (DARPA), Robert Kahn and Vint Cerf developed a *protocol* layer in 1973, called TCP/IP, that essentially allowed any two computers or networks to be connected. Other advances at this time included the development of usenets and e-mail. However, it was not until 1991 that Vannevar Bush's idea of a "thread of connecting thoughts" was realized with hypertext by Ted Nelson, Douglas Engelbart and Tim Berners-Lee [4-20]. Hypertext directly connects remote documents through links in written text.

Shortly thereafter, in 1993, the first graphical web browser Mosaic gave visual access to the growing number of *web documents*. Web documents are published in a relatively unrestricted language, the Hypertext Markup Language (HTML), which separates format and design from content. HTML is both an expressive language in the sense that, like natural language, it allows for a great deal of flexibility in content. At the same time, however, this comes as the cost of not having a great deal of programmatic power. As Tim Berners-Lee describes it:

> "When I designed HTML for the Web, I choose to avoid giving it more power than it absolutely needed - a 'principle of least power' which I have stuck to ever since." [4-20]

The primary drawback of HTML as a web document standard is that meaningful content is in natural language and is therefore difficult to process computationally. HTML was intended to be the "best way to represent hypertext", and was never designed to be machine understandable [4-20].

The natural language format of HTML, and thus of nearly all internet content, has led to the currently most popular way to locate and retrieve information, the *search engine*. While 87% of the American public on-line have used a search engine, only 66% report that it returns relevant results [4-21]. Much like a library card catalog, the search engine is designed on the principle of creating a keyword index of all words on the Internet, Figure 4.4. By specifying a set of keywords, the user is shown a list of relevant documents that match that keyword.

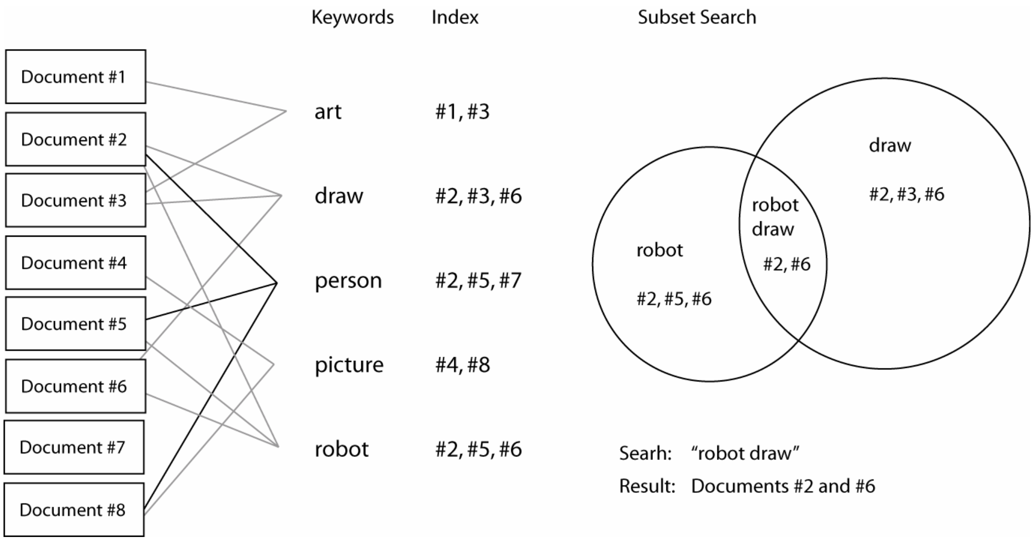


Figure 4.4. Operation of a search engine.

The first search engine, *Archie*, was created in 1990 by Alan Emtage, a student at McGill University in Montreal, to index file names. *Gopher*, created in 1991 by Mark McCahill at the University of Minnesota, was the first search engine to index text documents. Based on the first web search engine, *Wandex* created at MIT in 1993 by Matthew Gray, several commercial search engines appeared including Excite, Infoseek, Northern Light and Altavista [4-9].

The current standard, *Google*, rose in popularity in 2001. Google achieved this partly because of a unique *page ranking* algorithm that sorts results according to their importance. In this case, importance is measured by the number of external links that link to a particular site, thus establishing certain web sites as *authoritative* for a particular keyword [4-22]. The page rank algorithm used by Google is based on the same principle but includes other factors such as memory and keyword distance.[1]

While the success of Google is notable, it is important to point out the limitations of this system as well. Since documents are ranked based on external links, "authoritative" sites will appear more frequently even though they may not provide a factual or balanced view of a given keyword. For example, a Google search for the keyword "music" returns the results shown in Table 4.2.

[1] Soruce: Google's website. http://www.google.com/technology/

Table 4.2. Google search on the keyword "music".

Rank	Result
1	Yahoo! Music - Internet Radio
2	MTV.com (Music Television)
3	All Music Guide (a comprehensive guide to music recordings)
4	Apple + iPod + iTunes
5	Musicmatch Jukebox - the World's Best Music Player
6	Sony Music USA
7	CDNOW (Music Sales)
8	MP3 Music Downloads
9	Billboard.com (Popular music charts)
10.	Free Music Downloads

All results in the top ten are related to the businesses and advertising of music. Compare these results to the music taxonomy in Table 3.4. of chapter three. Notice that none of them provide an overview of what music *is*. This does not appear until the *twenty-first* search result in a page on the Essentials of Music. The problem is that an "authoritative link" denotes only one kind of semantic relationship: that of authority, in this case commercial authority. Yet we might like to explore the internet according to other semantics. The more fundamental problem is that pages cannot be searched in this way because they are ultimately in natural language rather than a semantic or computational one.

Some may claim the demonstration above is unfair because one typically uses multiple keywords when searching with Google. However, let us imagine the digital artist who wishes to find examples of robots that draw pictures of things. The term "robot" alone clearly provides too many results. Unfortunately, the keywords "robot draw" together return a great number of pictures of robots drawn by people or instructions on how to draw robots. By adding quotes around the keywords we can search for the exact phrase "robot draw", yet this never occurs in natural language so we try "robot that draws" instead. This is partially successful, but we also get a great number of generic robots that "draw" a certain amount of *electrical current* due to phrases like "X is a robot that draws 12 amps." Finally, many other types of robots are returned because the phrase "robot that draws" occurs in other contexts as well. Table 4.3 shows some of the other results that are returned.

Table 4.3. Some results from a Google search on the phrase "robot that draws"

robot that draws...	green lines of varying heights
robot that draws...	visitor's portraits
robot that draws...	on ideas from cognitive science
robot that draws...	Pictures
robot that draws...	12 amps
robot that draws...	useful traits from both parallel and serial robots
robot that draws...	the attention of the public

Notice that at least five different definitions of the word *draw* are returned. These are: 1) drawing on ideas, 2) drawing electrical current, 3) drawing attention, 4) drawing on a feature, and 5) the desired result of drawing a picture, of which only three of the seven results (42%) match the intended search. The problem is that, due to the medium of HTML, search engines are unable to distinguish the different meanings of the word draw. The way in which search engines resolves multiple keywords is by performing a mathematic intersection of the returned sets (Figure 4.5). Yet language is more rich than this. An intersection of sets does not capture the relationship between keywords or differing definitions in a single keyword.

Finally, we should observe that in the visions described at the beginning of this chapter searching is just one type of useful activity to perform on information. We would also like to compare it, navigate it according to specific criteria, visualize it in relation to other knowledge, and explore it according to chronology, by discipline or by some other concept.

Tim Berners-Lee, in acknowledge the limitations of HTML, has proposed a new system, called the Semantic Web, which may provide the semantic infrastructure needed to support a computable framework for the Internet.

4.2.4. Semantic Web

Natural language is inherently difficult for a machine to process. While research in natural language processing is improving, current trends indicate that complex statistical methods are required to resolve verb ambiguity, tense, and other features of natural language [4-23]. The solution is to express ideas in a language, or *grammar*, that is simpler and therefore easier for machines to process.

Tim Berners-Lee, the inventor and key advocate of the Semantic Web defines it as follows:

> "The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web - a web of data that can be processed directly or indirectly by machines." [4-21, p. 177]

The first step in this direction is, *metadata*, or data about data. A library card catalog is one such example as it provides information about other information. Another example is image metadata, which provides information about the content or format of a digital image.

In 1996, an internet work group of eleven developed the first specifications for XML [4-24]. XML provides a text-based tree structure for information. Consider the examples of XML tags in Table 4.4. Inherent to XML is the distinction between content and markup tags, yet unlike HTML both entities may contain any text. Due to this structure, XML is a suitable format for many kinds of metadata on the Internet as the tags provide a means to specify the semantics of any fragment of text. Notice that the use of the tag in Table 4.4 is arbitrary. They may indicate formatting style (e.g. bold), the value of a field (e.g. John's name), or other parts of speech that modify the text fragment.

Table 4.4. Different uses of XML as metadata.

Formatting metadata	<bold>This is a line of text</bold>
Record metadata	<name>John</name>
	<place>New York, NY</place>
Grammatic metadata	<short><happy>boy</happy></short>

The flexibility of XML tags allows the information to behave differently depending on how it is used. For this reason, XML has become very useful in application domains on the internet such as business transactions, record keeping, and for data exchange among web databases.

However, XML is not necessarily the ideal solution for a semantic internet. While the plain text format of XML makes it more readable at a glance it is less secure, results in storage inefficiencies, and necessitates additional techniques to allow for scalable queries [4-25]. Furthermore, XML does not change the document-centric nature of the web. Rather, it provides a means to simplify natural language by collecting *metadata* on existing internet documents that may be later processed by machines. Even so, researchers are examining ways to query large XML documents using relational databases [4-26].

Another candidate for this migration is the Resource Description Framework (RDF), which specifies metadata as subject-verb-object triples. This and other specifications are developed and maintained by the World Wide Web Consortium [4-27].

While there are many arguments for *metadata*, the use of metadata in practice presents some difficulties in building a truly semantic internet [4-28]. Tim Berners-Lee, one of the key advocates of metadata, suggests that web page authors themselves construct metadata as they build their websites [4-20]. Consider the example provide in the W3C RDF Primer:

```
ex:index.html   dc:creator             exstaff:85740 .
ex:index.html   exterms:creation-date   "August 16, 1999" .
ex:index.html   dc:language              "en" .
```

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:exterms="http://www.example.org/terms/">

  <rdf:Description rdf:about="http://www.example.org/index.html">
      <exterms:creation-date>August 16, 1999</exterms:creation-date>
      <dc:language>en</dc:language>
      <dc:creator rdf:resource="http://www.example.org/staffid/85740">
            Jane</dc>
</rdf:Description>
```

It is difficult to see how this simplifies creation of a web document, when it would be much easier to simply say:

> Created by Jane, staff member, on August 16th, 1999.
>
> Written in English.

In the design of semantic systems we should be working *toward* natural language, not away from it. We do not think at the level of metadata but at the level of language and meaning. Other issues with metadata include the fact that it increases storage requirements beyond the original data, requires curation to remain up-to-date, and is only as reliable as the translation made from the original document. Progress toward a semantic web based on metadata is moving forward, but slowly [4-29].

Performance is another challenge to metadata adoption. Consider the web pages of one million people, and their background information, held on one million different servers. While each computer holds the metadata of one person, it would be difficult to answer a natural question such as: List the first hundred people whose names begin with the letter 'S'? Distributed relational databases solve this by maintaining sophisticated indices in a central location. Metadata resides, without global indexing, with each original document. Thus, without database technology it is impossible to retrieve such queries efficiently. A solution to this problem will be presented in chapter eight (Distributed Knowledge Systems).

Perhaps an entirely new approach is needed as current solutions are all based conceptually on HTML, a natural language document format. Even RDF refers to its resources in plain text, which is extremely inefficient both in storage and computation. We must go back to the grammatical level, before the HTML document, to the protocol and data layer and acknowledge that *the best systems to operate on semantic knowledge will operate directly on semantic structures, not textual facsimiles, and store those structures in a genuine low-level data format.* Like the pioneers of early database systems, we should not be afraid to write entirely novel database architectures and protocols from the ground up in low-level languages. The above discussion reveals that there are compelling reason to do so, in order that we may hold English-like grammatic structures with the same reliability, scalability and performance as current relational database systems.

4.2.5. Databases

The database is a step removed from the traditional written document. Databases, unlike digital documents, can be more easily manipulated by machine. Three early database models were the *hierarchical model*, the *network model* and the *relational model*. The hierarchical model was developed into IMS, a database system by IBM used to manage the Bill of Materials for the Saturn V moon rocket. The network model, by Bachman , would be used to initiate database Codasyl - precursor to the programming language COBOL [4-30]. The relational model, introduced in 1970 by E.F. Codd, included a way to separate the mechanism of data storage from its content [4-31]. For this reason, and others, the relational model has become the industry standard for database systems.

All early databases included a *schema*, or a metadata description of the information the database contains. The schema provides the structure for the actual *instances* of data. The motivation for a schema, however, is considerably different from the metadata described in the previous section. First, the schema is usually fixed prior to data entry and forms a template for new items. Second, data in a database is constructed from the schema, whereas metadata is derived from existing data.

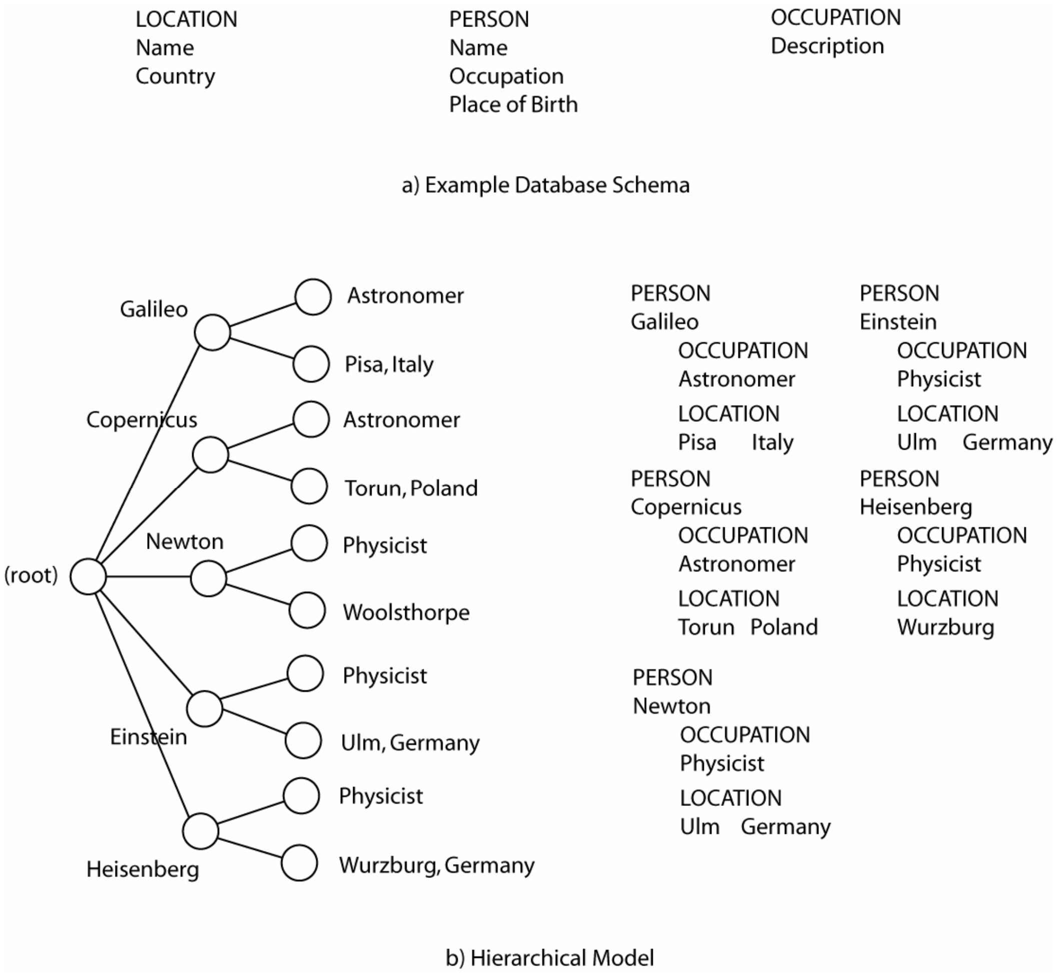
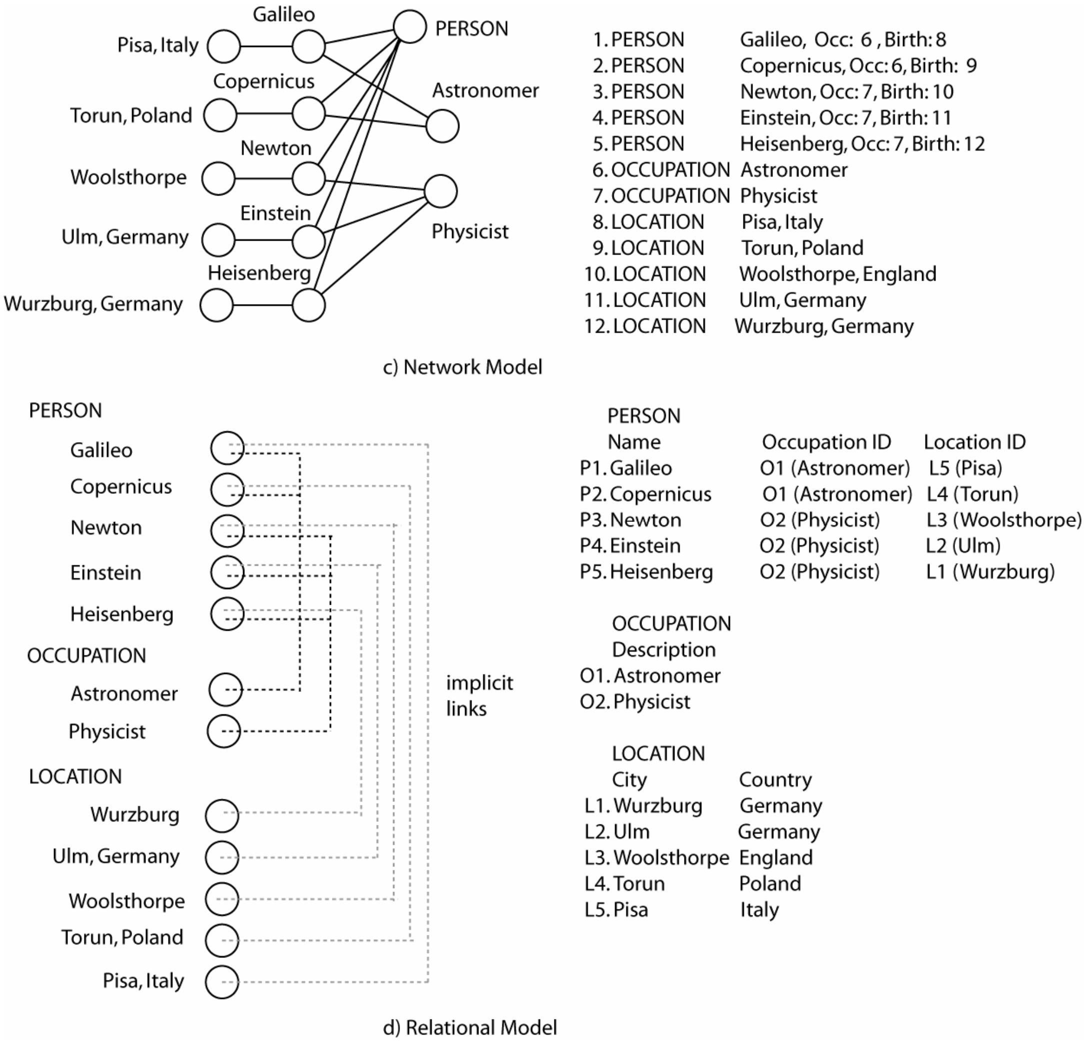


Figure 4.5. Hierarchical database model. a) Sample database schema and b) specific examples represented used the model.

One of the first database models was the hierarchical model. In this system, the structure is limited so that any instance can have only one parent. A sample schema and an example of a hierarchical database are shown in Figure 4.5. Due to the structure, the occupation field must be duplicated for each person that shares that occupation.

I



1.	PERSON	Galileo, Occ: 6 , Birth: 8
2.	PERSON	Copernicus, Occ: 6, Birth: 9
3.	PERSON	Newton, Occ: 7, Birth: 10
4.	PERSON	Einstein, Occ: 7, Birth: 11
5.	PERSON	Heisenberg, Occ: 7, Birth: 12
6.	OCCUPATION	Astronomer
7.	OCCUPATION	Physicist
8.	LOCATION	Pisa, Italy
9.	LOCATION	Torun, Poland
10.	LOCATION	Woolsthorpe, England
11.	LOCATION	Ulm, Germany
12.	LOCATION	Wurzburg, Germany

c) Network Model

PERSON

	Name	Occupation ID	Location ID
P1.	Galileo	O1 (Astronomer)	L5 (Pisa)
P2.	Copernicus	O1 (Astronomer)	L4 (Torun)
P3.	Newton	O2 (Physicist)	L3 (Woolsthorpe)
P4.	Einstein	O2 (Physicist)	L2 (Ulm)
P5.	Heisenberg	O2 (Physicist)	L1 (Wurzburg)

OCCUPATION

	Description
O1.	Astronomer
O2.	Physicist

LOCATION

	City	Country
L1.	Wurzburg	Germany
L2.	Ulm	Germany
L3.	Woolsthorpe	England
L4.	Torun	Poland
L5.	Pisa	Italy

d) Relational Model

Figure 4.6. Network and Relational Database models

The network model consists of a set of directed edges that link concepts, where the head of the link defines a class and the tail defines an instance. For example, in Figure 4.6, a link with Person at the head and Galileo at the tail specifies that Galileo is an instance of a person. As we will see, network databases are very similar in structure to semantic networks.

A relational database consists of tables with columns referred to as *fields* and rows referred to as *records*. One important benefit of the table structure is that the links between fields and instances are implicit. Thus a table of people, shown in Figure 4.6, with fields for name, place of birth and occupation, automatically implies that each person has these fields. This allows a user to more easily interface with the database as relationships do not need to be explicitly specified each time. However relational databases have the drawback that, unlike the network model, a single record cannot be arbitrarily extended with new information. In the example above, to add more information to Galileo would require a new column to be added to the entire table for People. While affording easier interaction, the tabular structure makes it inefficient to extend the semantics of the database.

In general, databases are useful for storing, searching and modifying *records*. However, they have not been adapted to general knowledge systems primarily because it is difficult for them to represent complex ideas. Consider this example, taken from Asimov's Biographical Encyclopedia of Science & Technology:

> "Vannevar Bush, the son of a minister, was educated in the Boston area, doing his undergraduate work at Tufts University and obtaining his doctorate at MIT and Harvard University in 1916. He taught at Tufts for a few years, but in 1919 accepted a professorial position at MIT. In 1926, Bush and his colleagues constructed a machine capable of solving differential equations." [4-32]

To express this in a relational database would be difficult since it would require new tables for people, geography, types of degrees, occupations, social relationships, machines, and mathematical entities (for differential equations). Relational databases also have difficulty in representing complex grammatic structures since the schema is fixed [4-33]. The most common solution is simply to place the above text in a new field named "Description", but this suffers from the same problem as the typical web article in that both must be translated from natural language.

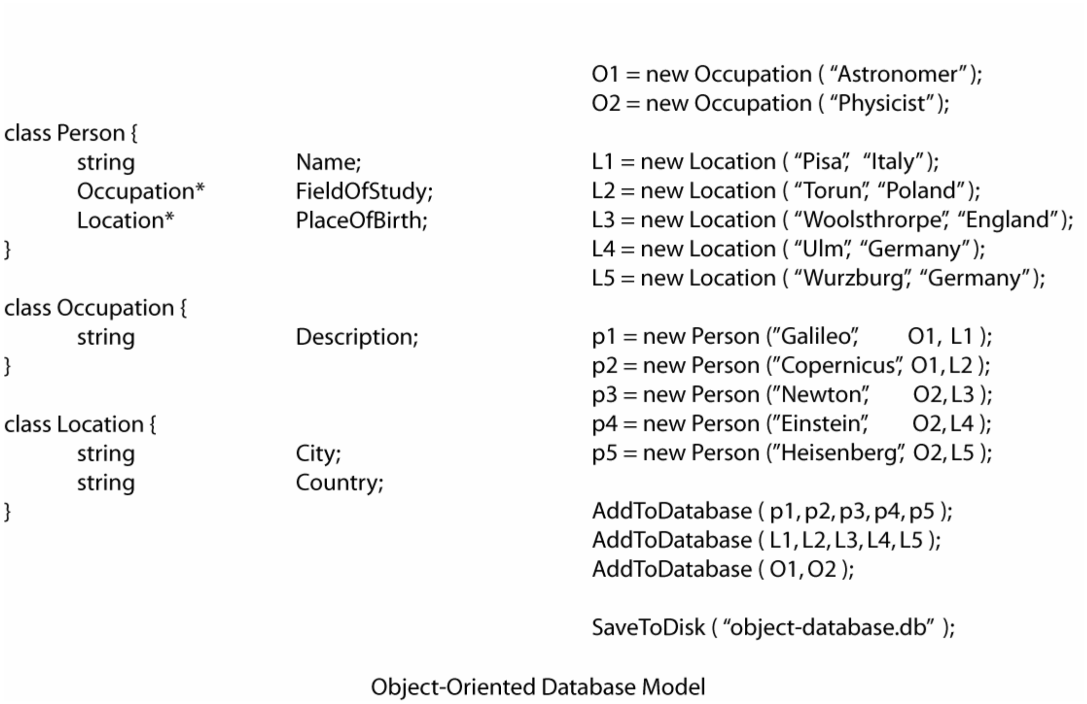
Relational databases are more *computable* than written documents, because records have a predefined semantics, but less *expressive* overall. Thus retrieving a single record by name or address, for example, is easy but representing complex information about that person is difficult. Despite this, relational databases have become the industry standard. This is due to the significant efforts placed on building reliable, distributed, lockable, multi-user systems that operate on millions of records. In the early 1980s, relational database research received significant government funding, only to be overwhelmed by the internet revolution in the early 1990s [4-34]. Now, realizing some of their limitations, new systems are being designed. The object-oriented database (OODB) and the object-relational database derive their improved flexibility from programming languages [4-35].

4.2.6. Object-Oriented Databases

Object-Oriented Database rose out of the realization that the schema (Figure 4.7) of a relational databases are very much like *classes* in programming languages. For example, to write a program that computes a person's age from their birth year, one might first develop a class for Person with data variables for the month and year of birth. One then creates as many *instances* of the Person class as are needed. In other words, both programming classes and database schema are used to generate instances of data:

> "The object-oriented database (OODB) paradigm is the combination of object-oriented programming language (OOPL) systems and persistent systems. The power of the OODB comes from the seamless treatment of both persistent data, as found in databases, and transient data, as found in executing programs." [4-36]

A persistent system is simply a software program capable of transferring memory structures to disk, and back to memory again. Thus, an Object-Oriented Database is essentially a computer program, with all its necessary classes, objects and variables, along with a means of saving existing data in memory to disk. Figure 4.8 shows an Object-Oriented Database model for the example used in the previous section.



```
class Person {
        string          Name;
        Occupation*     FieldOfStudy;
        Location*       PlaceOfBirth;
}

class Occupation {
        string          Description;
}

class Location {
        string          City;
        string          Country;
}
```

```
O1 = new Occupation ( “Astronomer” );
O2 = new Occupation ( “Physicist” );

L1 = new Location ( “Pisa”,  “Italy” );
L2 = new Location ( “Torun”, “Poland” );
L3 = new Location ( “Woolsthrorpe”, “England” );
L4 = new Location ( “Ulm”, “Germany” );
L5 = new Location ( “Wurzburg”, “Germany” );

p1 = new Person (”Galileo”,        O1,  L1 );
p2 = new Person (”Copernicus”, O1, L2 );
p3 = new Person (”Newton”,        O2, L3 );
p4 = new Person (”Einstein”,       O2, L4 );
p5 = new Person (”Heisenberg”, O2, L5 );

AddToDatabase ( p1, p2, p3, p4, p5 );
AddToDatabase ( L1, L2, L3, L4, L5 );
AddToDatabase ( O1, O2 );

SaveToDisk ( “object-database.db” );
```

Object-Oriented Database Model

Figure 4.7. Example of a simple Object-Oriented Database in the C++ programming language for the example from the previous section.

The benefit of the Object-Oriented Database is that very complex operations can be performed on the data. However, the drawback is that the user must essentially be a programmer. In addition, any search and query capabilities must be explicitly programmed. This has led to a hybrid solution called the Object-Relational Database in which the search and query benefits of relational databases are combined with the power of Object-Oriented Databases [4-37]. Processing of data, however, must still be done by explicit programming in a low-level language or using the query capabilities of a relational database.

Similar to the problem encountered with tables in relational databases, one consequence of relying on compiled programming structures is that they cannot be easily modified. To extend the concept of a person to include location of birth requires reprogramming the class. This results in version control issues when one attempts to import data saved prior to this revision. As with the relational model, this necessitate reformatting all prior data to the new structure.

Object-Oriented Database models and their variants are being increasingly used in industry applications where computational problems must be solved. This is because they provide a way for the programmer to write complex programs and also have access to the data needed to solve these computationally intense problems. In order to be useful as semantic systems, however, they must be linked with one of the other more knowledge specific solutions presented here.

4.2.7. Semantic Networks

A semantic network is not a database system per se. Conceptually, it is identical to the concept map described by Joseph Novak in chapter two. Formally, a semantic network is a directed graph in which the vertices are concepts and the edges represent relationships between these concepts. A simple semantic network is shown in Figure 4.8.

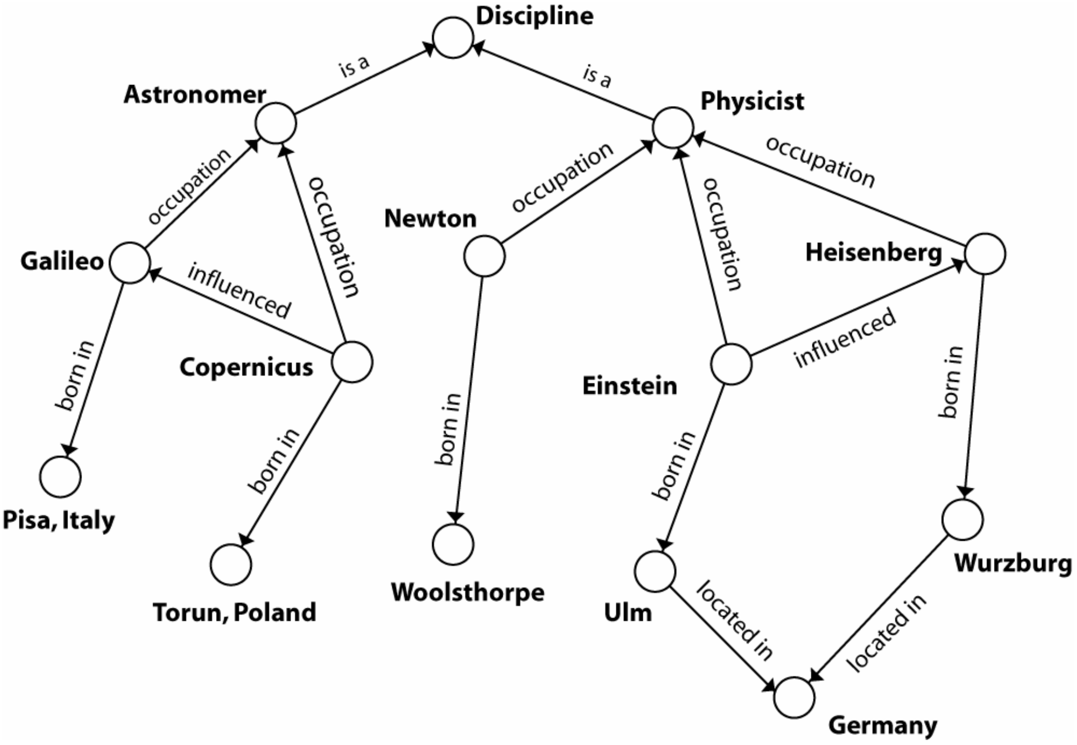


Figure 4.8. Semantic Network

Semantic networks originally developed out of linguistics, where they were first used to construct diagrams of grammatical dependence [4-38], and to perform machine translation [4-39]. A more detailed history of semantic networks is available from John Sowa. [2] One of the early researchers in this area, Sowa applied semantic networks to create *conceptual graphs* that would allow machines to process and understand English-like structures [4-40].

In general, the use of conceptual graphs has historically been applied to the areas of expert systems and artificial intelligence, where the goal is for the machine to "reason" about knowledge structures. The functional part of the system that accomplishes this an *inference engine,* a set of algorithms capable of following and generating rules to determine new relationships in semantic network.

[2] Website: http://www.jfsowa.com/pubs/semne.htm.

MYCIN, a five year project at Stanford to create a system that can diagnosing patients is one such example [4-41]. The following is a sample dialog with MYCIN:

> Q: WHAT IS THE INFECTION?
>
> A: Primary Bacteremia
>
> Q: PLEASE GIVE THE DATE AND APPROX. TIME WHEN
> SIGNS OR SYMPTOMS FIRST APPEARED.
>
> A: May 5, 1975
>
> Q: FROM WHAT SITE WAS THE SPECIMEN TAKEN?
>
> A: Blood
>
> GIVE: Gentamicin
>
> DOSE: 119 MG (1.7 MG/KG) Q8H Intra-venously for 10 days

MYCIN achieves a success rate of 65%, better than most non-bacterial physicians, but worst than a real expert (success rate of 80%) [4-41]. The above dialog is representative of the kinds of questions one might wish to ask of an expert knowledge system, as outlined in chapter one. Expert systems like MYCIN, developed in the 1980s, allow for machine reasoning in a particular, limited domain. Like a database, an expert system stores information. But where the database holds simple information in large quantities for the purpose of managing it, an expert system holds small amounts of complex knowledge for the purpose of reasoning with it.

There are several possible reasons semantic networks have not yet been widely adopted to large-scale database systems. First, research in semantic networks may be directed more toward the active agents of artificial intelligence than the (relatively) static knowledge of databases. Second, the basic definition of a semantic network is not as structured as a relational database, making it difficult to formalize the entities that will be stored. Finally, computation on a semantic network has traditionally been inefficient with large datasets since it must traverse the entire network to respond to a query. In Figure 4.8, listing all people requires traversing every node in the network to see if it is "a person".

Despite the limitations, new research is promising. Semantic networks are closely related to the network database models of Bachman. The work of Gyssens *et al.* uses patterns to give structure to a semantic network, thus allowing for the construction of graph-oriented databases [4-42]. Levene *et al.* introduce the hypernode model which links graph-based models to set-based models, such as the relational database [4-43]. Cardelli is using semi-structured data to help eliminate the fixed structure of the schema [4-44]. Finally, work by Levene and others is being done to prove the expressive power of various types of databases to make explicit their benefits and limitations [4-45].

4.3. Summary and Features of Knowledge Systems

We have observed a stead increase in the complexity of knowledge systems throughout history. Physical libraries represent an increase in scale and number of physical documents in natural language while encyclopedia represent summaries of that knowledge. The computer and the world wide web have extended the physical document into the digital medium. Beyond this, software systems have continued to extend the grammatical flexibility of machines while the database represents the first transformation of written materials into a more computable form. Newer systems such as object-oriented databases and semantic networks continue to push our attempts at making computational grammar increasingly more sophisticated.

In retrospect, the goal would appear to be to allow machines to better understand human language. In essence, we are relearning how to speak (transmit), remember (store), and think (compute) in this digital medium.

A summary of the systems described in this chapter and their relationships is shown in Figure 4.9. Dates of the earliest example of each are shown. It is interesting to note the progression from physical to analog to digital, and the increase in grammatic flexibility in modern systems.

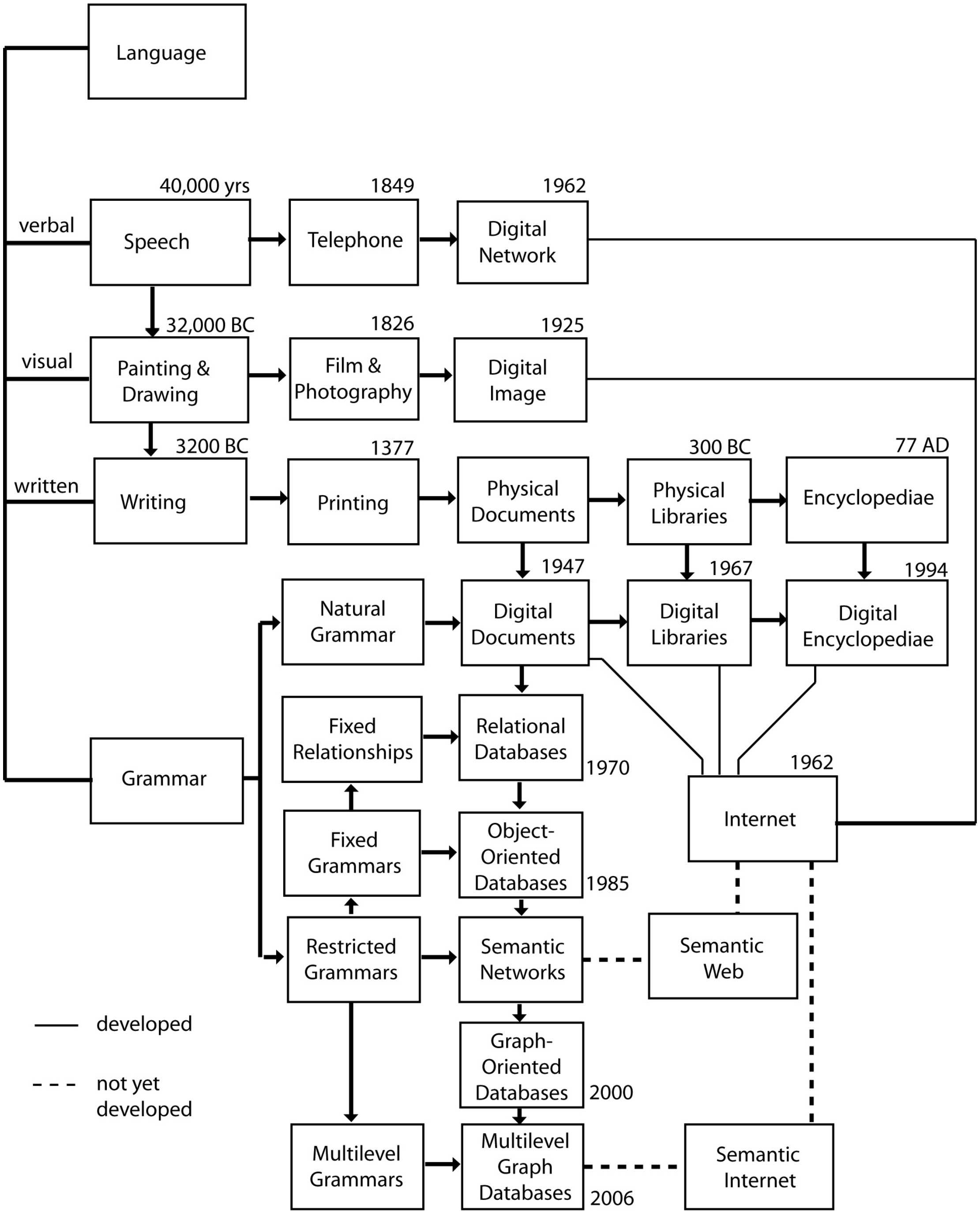


Figure 4.9. Summary of language, communication and knowledge systems throughout history.

Having examined both ideal and existing systems, we are ready to formulate a concise list of desirable features. These include not only the features of a database, but the features a general knowledge reference resource.

Table 4.5. Desirable features of a knowledge system.

Systemic Features	
8. Multi-user	Allow many users to access and update content
9. Expressive	Represent complex knowledge
10. Efficient	Efficient response to navigation and queries
11. Scalable	Able to support large amounts of data
12. Queryable	Able to answer specific, complex questions
13. Filterable	Able to be filtered like a traditional database
14. Searchable	Able to be searched like the Internet
15. Multimedia	Able to store multimedia (images, sound, etc.)
16. Reliable	Robust system design
17. Extensible	Permit additional features and visualizations
User-Interface Features	
18. Clear	Be understandable, even aesthetic, to the user
19. Adaptable	Provide interfaces for the novice and expert
20. Navigation	Allow users to easily navigate and explore ideas
21. Comparison	Allow users to construct comparisons
22. Zoomable	Allow users to look at concepts at different scales
23. Visual	Allow users to visualize specific relationships
24. Navigation	Allow user to smoothly navigate concepts
Socially-Driven Features	
1. Comprehensive	Contain large amounts of real knowledge
2. Consistent	Express things in similar ways.
3. Accurate	Contain accurate, factual knowledge
4. Interdisciplinary	Store knowledge across multiple disciplines
5. Free	Provided for the common good (ideally)
6. Ubiquitous	Accessible from any machine
7. Summative	Provides summaries where needed
8. Organized	Easy to locate or navigate to a particular item

The features in Table 4.5 are divided into the categories of 1) systematic, 2) user interface and 3) socially-driven features. The first, systematic, deals with the features of a database as a storage system. These include the usual factors of scalability, distribution, and reliability, but also address the need for expressive representations and the inclusion of multimedia content. Common operations such as searching, filtering and querying should be a part of any knowledge database.

The user interface for a knowledge resource should not be limited to only text. In this respect it should clearly convey semantic relationships through flexible navigation and the ability to freely zoom to differ scales of information. Interfaces should be available for various types of activities suitable to both the novice and expert researcher. Finally, the interface should simplify comparisons and views of knowledge from multiple perspectives.

Socially, an ideal knowledge system should be comprehensive, consistent, accurate, and well organized. In addition to coveying concepts from many disciplines it should be accessible from any machine and ideally free for all to use. Due partly to social factors, but also to technical ones, systems that meet all the above requirements do not yet exist.

4.5. QUANTA: An Interdisciplinary Knowledge Database

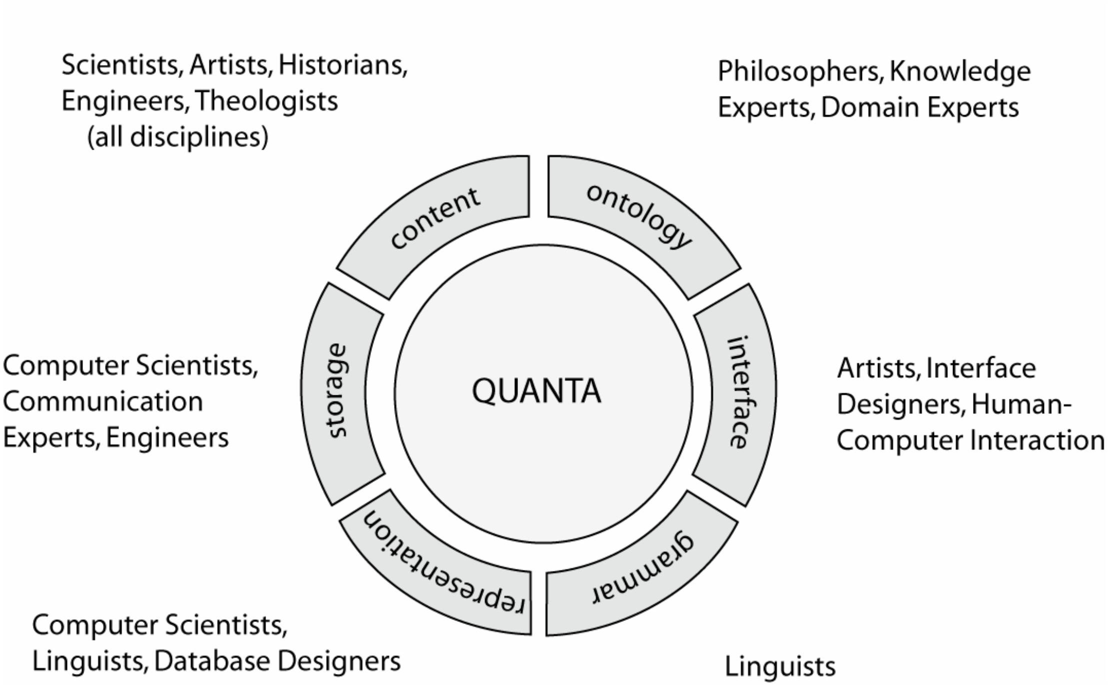

Scientists, Artists, Historians, Engineers, Theologists (all disciplines)
Philosophers, Knowledge Experts, Domain Experts
content
ontology
storage
QUANTA
interface
Computer Scientists, Communication Experts, Engineers
Artists, Interface Designers, Human-Computer Interaction
representation
grammar
Computer Scientists, Linguists, Database Designers
Linguists

Figure 4.10. Quanta: A knowledge framework supporting interdisciplinary collaboration.

Quanta is a prototype for an interdisciplinary knowledge database capable of large-scale, distributed, English-like representation of human knowledge. As such, design considerations for Quanta include an attempt to incorporate methodologies from different disciplines into a single framework. While any field may contribute to content, certain disciplines will necessarily contribute to the various components of the system as shown in Figure 4.10.

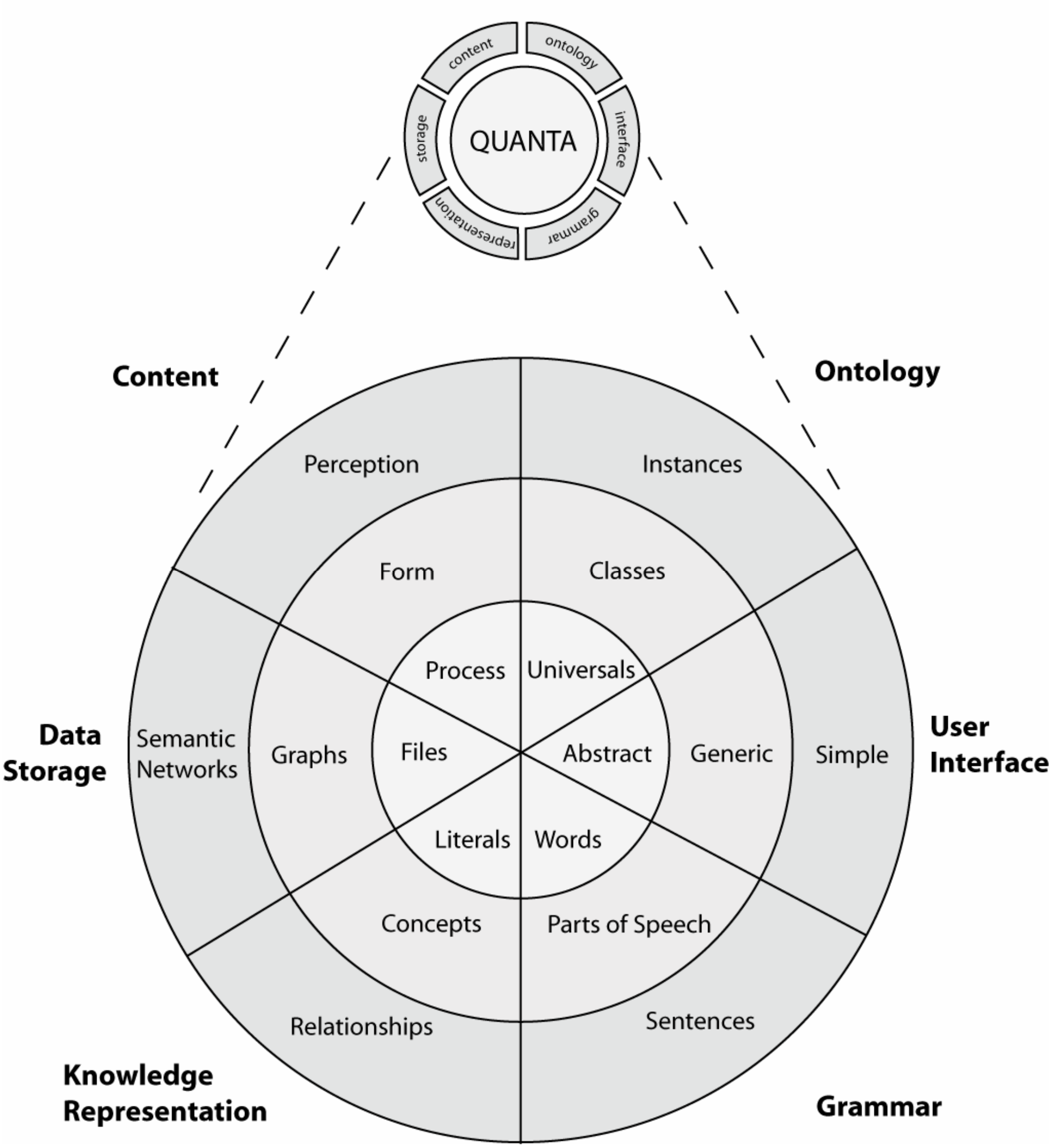


Figure 4.11. Quanta: Internal structures and layers

As a knowledge system, Quanta achieves its flexibility by implementing several layers in each problem domain. This represents the internal structure of Quanta. The layer domains include the *content layers*, the *ontology layers*, the *user interface* layers, the *grammar layers*, the *knowledge representation* layers and the *data storage layers*. as shown in Figure 4.11 It is interesting to observe that the organization of each layer roughly corresponds to Plato's Divided Line, with images and perceptions (*eikasia*) on the outer ring and universal knowledge on the inner ring (*noesis*).

Quanta is based on existing systems described in this chapter but extends them in the following novel ways:

- Novel user interfaces to aid in interaction with graph-oriented databases
- Construction of a top-level ontology suitable to interdisciplinary knowledge
- Construction of a hypergraph structure that allow a semantic network to hold large numbers of complex records.
- Some solutions to the efficiency problems of large semantic networks
- Distributed solutions for semantic networks allowing them to be applied to knowledge organization of the Internet.

More importantly, the overall design of Quanta encourages that these issues be solved simultaneously so that the resulting system is simple to use, efficient, powerful and flexible. Balance in design was a critical driving force throughout the development of the prototype.

A summary of the various layers of Quanta is presented here, while the purpose and function of each will be developed in the remaining chapters.

Table 4.6. Quanta: Description of functional layers

Quanta: System Overview	
Data Storage Layers	
Layer 1. File System	Physical disk storage
Layer 2. Nodes	Storage of the conceptual nodes
Layer 3. Graph	Storage of the semantic graph
Representation Layers	
Layer 1. Literals	Representation of literal objects (images, text)
Layer 2. Concepts	Representation of concepts and ideas
Layer 3. Relations	Representation of relationships between concepts
Grammar Layers	
Layer 1. Words	Grammar as data and symbol
Layer 2. Speech	Grammar as nouns, verbs and parts of speech
Layer 3. Sentences	Grammar as sentences (linguistic structure)
Ontology Layers	
Layer 1. Universals	Top level ontology of universal structures
Layer 2. Classes	Middle level ontology of conceptual classes
Layer 3. Instances	Bottom level ontology of specific instances
Content Layers	
Layer 1. Process	Fundamental processes (verbs)
Layer 2. Form	Fundamental existent objects (nouns)
Layer 3. Perception	Every day objects and things
User Interface Layers	
Layer 1. Abstract	Interface to change fundamental concepts
Layer 2. Generic	Interface to change common ideas & principles
Layer 3. Specific	Interface to change individual objects (e.g. people)