Dagobert Soergel

College of Library and Information Services

University of Maryland

College Park, MD 20742

Organizing Information

Organizing information is at the heart of information science and is important in many other areas as well. In bibliographic and similar information systems it involves classification as well as the description of documents or other entities; in database management it is known as data modeling; in artificial intelligence, as knowledge representation for expert systems, natural language understanding, and other purposes; in psychology, as the structure of memory and cognition; in linguistics, as syntax and semantics and structure of discourse; in technical writing, as the structure of a composition; in biology it is used on two levels: in the classification of organisms and in the study of information transferred through genes. In all scholarly and scientific fields, organizing information is important for establishing frameworks for thought used in research and teaching. It assists in the formation of useful concepts and it serves to clarify terminology to assist both authors and readers. Many of these topics are coming together in the emerging discipline of cognitive science. Finally, philosophy of knowledge is concerned with the clarification of many of these issues.

Curricula in information science should have a high-powered truly graduate level core course in organizing information or knowledge representation. Such a course should not be tied to any specific field of application but deal with general principles while drawing on specific applications for purposes of illustration.

Approaches to organizing information

A fundamental approach to organizing information is the analysis of a body of information into statements that connect entities through relationships. The semantic structure of an application area is captured through a conceptual schema of statement patterns which are composed of a relationship type and entity types, for example

document <authored-by > person

person <works-at > organization

organization <located-at > geographical-unit

organization <has-jurisdiction-over> geographical-unit

substance <is-toxic-for> (organism, organ, strength)

To each entity type belongs a set of entity values, called its domain. These entity values may be arranged in an elaborate hierarchical classification, such as a classification of geographical units, of organisms, or of concepts. Such classifications are very important in organizing information.

A database can be seen as a collection of statements formed according to a conceptual schema. There are several methods which can be used individually or in combination to represent these statements and organize their storage. One can store each statement directly, as in a predicate logic representation or in a semantic net. One can show relationships through the physical arrangement of entity identifiers in the store, as in a hierarchical database. One can pull together all statements with the same relationship type and store them as pairs or tuples in a relation or in

a owner-set structure. One can also pull together several statements about the same entity and store them as a record; for example,
the record for an organization might contain a data field

<has-jurisdiction-over> geographical-unit

A record format collects various aspects of an entity about which information is to be stored. Frames and scripts are a generalized and more powerful version of records; their slots (data fields) have default values and attached procedures which can be used to find a value. Case grammar and facet analysis are closely related notions.

Information about permissible conclusions can be captured in if-then rules (also known as production rules). In some expert systems production rules are used to represent all types of information including simple facts.

A scheme for organizing information is also a scheme for pinpointing missing information or for suggesting means to obtain this missing information. This is particularly true for frames.

Problem-driven design

The design of a scheme for organizing information should be driven by the problems to be solved by the users (Calvin Mooers). Put differently, information should be organized in such a way that it is easy to find information which is helpful in the solution of a given problem. This is not the same as a merely user-driven design. Problem-oriented design is a process of joint problem analysis and problem solving in which the user and the information

professional/systems analyst each make their contribution. This process starts with the preparation of an inventory of problems to be solved, decisions to be made, tasks to be performed. Each of these problems/decisions/tasks gives rise to one or more statements of information needed or search requests to be expected. These statements suggest the entity types and relationship types as well as specific entity values needed to express them. The designer then constructs a coherent scheme of entity types and relationship types and, for each entity type that requires it, a well-structured list of entity values (for example, a well-structured list of concepts). This scheme then serves as a framework for collecting and organizing information. It communicates the information needs of users to the operator of the information system (indexers, coders, data analysts).

The "user" of a database (also called knowledge base) may also be an expert system (e.g., a system for medical diagnosis or a system for knowledge extraction from text) which needs data from a database for its reasoning process. The scheme for organizing the information in the database should then be driven by the requirements of this expert system.

The rationale for the problem- or request-oriented approach is summarized in the following section (adapted from Section 13.7 of the book) with reference to index language construction and indexing.

Two opposing principles for building an index language can be found in the literature:

1. Follow the vocabulary of the user; omit terms not contained explicitly in the user's vocabulary, even if they are necessary for logical coherence.

2. Follow the vocabulary of the entity creator (e.g., document author) so as not to distort his meaning. Include terms from the text or title of documents, food names given by manufactures, self-descriptions of persons or organizations, etc.; omit terms not appearing explicitly in such sources, even if they are necessary for logical coherence.

Each of these principles has merit; but the exclusive use of one or the other fails to solve the problems of communication between users and authors. An index language must support optimal service to the user by providing the foundation for indexing and retrieval operations. This task requires more than following the user's or the author's vocabulary. There is no such thing as the "user"; there are many users, and their viewpoints often contradict each other. There is no such thing as "the author" either; there are many authors, and they often use different terminology. Authors and users often have different purposes: The use a user makes of an entity is often quite different from what the author thought the entity would be useful for. The indexer serves as the user's agent by indicating possible uses of each incoming entity. The indexer must analyze the entity at hand and then make a sound relevance judgment that is as useful as (or perhaps even more useful than) the user's own relevance judgment would be. At her best the indexer does "scientific prethinking". By analyzing entities as the user's agent, the indexer saves the user time. Ideally, the indexer evaluates each entity critically, something the user may not be able to do for lack of time or lack of expertise or both.

In order that the indexer can fulfill this demanding role, he must have a clear picture of the problems or tasks of the user and the information or entities needed to solve these problems. If there were only very few users, they could communicate their interests directly to "their" indexer. However, normally there are many users, most of whom the indexer does not know. Hence the mental frameworks of many users must be combined into one logical coherent structure that can

be understood and internalized by the indexer. Developing such a framework requires careful analysis of needs and critical examination of the conceptual structure of the subject field at hand. The index language thus constructed serves as a communication device from the users to the indexers; it provides the framework that allows for a meeting of minds to take place.

The index language, once constructed based on the analysis of the needs of all users, also serves as a communication device from the information system to the individual user. It gives the user a mental framework, a knowledge map, a guide through the collection of information or entities available in the information system. (In a library where materials are arranged in meaningful order or in a grocery store the user literally has a map of where to find what.) If the structure of such a knowledge map can be made congenial to the user's own mental framework, so much the better. But the user's framework may be less suitable, less powerful for organizing the subject matter at hand than an index language/classification constructed through careful consideration of the foundations of the subject. The index language then becomes a powerful agent for education, enriching the user's mind; the conceptual framework developed for the external information system improves the user's own internal information system. This takes on particular significance with an information system for children or students, since young minds are apt to absorb the organizing principles used in such a system and use them to build their own view of the world. Hence, an index language should use structural principles derived from modern classification theory - such as the principle of facet analysis - and a semantic organization based on the newest insights and paradigms of the subject fields covered.

To conclude, the maker of an index language and thesaurus is confronted with the challenge of clarifying the muddled terminological and conceptual systems of a field (or several fields combined) and detecting its underlying logical structure, thus laying a foundation for successful communication.

Testing and evaluation

How do we know that a problem- or request-oriented organizing scheme works better than any other? Clearly, schemes for organizing information must be evaluated so that one can choose between them or at least have an idea of the level of performance of the scheme being used. Unfortunately, such evaluation is extremely hard to do. An improperly conducted test is worse than no test at all since one can easily be swayed by test results into believing what one knows not to be true. The Cranfield experiment on index languages for bibliographic retrieval is a classic case in point. This experiment was flawed in the compilation of the sample collection and the sample queries, in the process of judging relevance, in the construction of the index languages, and in the procedures used for indexing and searching. Any one of these problems alone would render the results highly suspect, their cumulative effect makes them meaningless. The results - that a free vocabulary performs better than a controlled vocabulary - were an artifact of the experimental method, yet the Cranfield experimenters accepted them against their better judgment, and many people quote them to this day to prove that the work put into carefully constructed classifications is all for naught. There has been a plethora of studies in the same vein - many using the Cranfield collection, queries, and relevance judgments - and with the same results. In contrast, the authors of the much more carefully designed study done at Case Western Reserve University acknowledged the complexity of the problem; the study contributed much to our insight but refrained from sweeping statements about the results. There has been one study of request-oriented indexing and one well-designed study of full-text retrieval, and the results are dramatically different.

Another important point that is often forgotten when drawing conclusions from retrieval tests has been succinctly stated by Fairthorne: "To test is not to evaluate". Even a properly conducted test can give only raw performance data; these raw data must then be evaluated with respect to

specific user requirements derived from user problems. A scheme for organizing information must always be judged in a specific situation.

If most retrieval tests cannot be counted on, what methods are available to evaluate schemes for organizing information with respect to expected retrieval performance? The best method is to gain insight into the functioning of information systems. Such insight can come from a microanalysis of successes and failures in retrieval tests (as done, for example, in Lancaster's work on MEDLARS), but above all it requires careful reasoning about the role of the organizing scheme in the process of indexing, retrieval, and inference. This article, and the book on which it is based, are meant as a contribution to this end.