# Information Organization

Dagobert Soergel

Berkshire Encyclopedia of Human-Computer Interaction  July 2004

We organize information – in our minds and in information systems – in order to collect and record it, retrieve it, evaluate and select it, understand it, process and analyze it, apply it, and rearrange and reuse it. We also organize things, such as parts, merchandise in a store, or clothes in a closet, using similar principles for similar purposes.

Using data on foods as an example, this article will introduce

- the entity-relationship (E-R) approach as the basis for all information organization;
- database organization: relational databases, object-oriented databases, and frames;
- templates for the internal organization of documents;
- cataloging and metadata;
- knowledge organization systems (KOS): (faceted) classification schemes, taxonomies, ontologies, and thesauri; knowledge representation.

## The Entity-Relationship Approach

Information organization depends on object characteristics (or properties), often expressed as statements: **entities** (nouns) are connected through **relationships** (verbs), for example:

pecan pie *has ingredient* (shelled pecans, 2 cups, for taste)

Figure 1 shows an E-R **conceptual schema** for foods ─ a list of **statement patterns**, each defining a type of data stored in the database.

| | | |
|---|---|---|
| Food product | *hasName* | Text |
| Food product | *hasDescription* | Text |
| Food product | *hasHomePrepTime* | Time duration |
| Food product | *isa* | Food product [Gazpacho *isa* soup] |
| Food product | *comesFromSource* | Food source [plant or animal] |
| Food product | *comesFromPart* | Anatomical part [leaf, root, skeletal meat] |
| Food product | *hasIngredient* | (Food product, Amount [number and unit], Purpose) [(Chocolate, 50 g, for taste), (BHT, 0.1 g, preservation)] |
| Food product | *underwentProcess* | (Process, Intensity, Purpose) [(broil, low heat, to brown)] |
| Food product | *containsSubstance* | (Substance, amount) [(fat, 13 g), (vitamin A, 4000 IU)] |
| Food product | *intendedFor* | Type of diet [low-fat, low-salt] |

**Figure 1. Entity-relationship (E-R) schema for a food product database**

This E-R schema is the basis for several approaches to storing and presenting data and for organizing a database for access and processing.

## Database Organization

In a **relational database**, data corresponding to one relationship type are expressed in a table (also called relation, Figure 2); data about one "object", such as a food product, are distributed over many tables. Tables are very simple data structures that can be processed simply and efficiently.

**Table for Food product *has ingredient***

| ***foodName*** | ***ingredient*** | ***no*** | ***unit*** | ***purpose*** |
|---|---|---|---|---|
| pecan pie | flaky pie crust | 1 | count | body |
| pecan pie | shelled pecans | 2 | cup | taste |
| pecan pie | eggs | 5 | count | body |
| pecan pie | white sugar | 1 | cup | taste |
| Diet Coke | carbonated water | 355 | ml | body |
| Diet Coke | aspartame | 200 | mg | taste |

**Table *for intended for***

| ***food product*** | ***diet*** |
|---|---|
| pecan pie | normal |
| Diet Coke | low cal |
| split pea soup | normal |
| unsalted butter | low salt |
| ice cream | normal |
| frozen yogurt | low cal |

**Figure 2. Tables (relations) in a relational database**

**Object-oriented databases** store all the information about an object in one **frame** which has a **slot** for every object characteristic as expressed in a relationship (Figure 3). A frame can also call procedures operating on its data, such as computing fat content by adding fat from all ingredients. Frames are complex data structures that require complex software. Frames (in databases or in the mind) use the mechanism of **hierarchical inheritance** for efficient data input and storage; for example, the frame for chocolate pecan pie simply refers to the pecan pie frame and lists only additional slots, such as

*ingredient:* (chocolate, 50 g, for taste).

| | |
|---|---|
| *foodName*: | shelled pecans |
| *fromSource:* | pecan tree |
| *fromPart:* | seed |
| *process:* | shelling |

| | |
|---|---|
| *foodName*: | eggs |
| *fromSource:* | chicken |
| *fromPart:* | egg (part of animal) |

| | |
|---|---|
| *foodName*: | pecan pie |
| *description:* | A custard pie, loaded with pecans |
| *ingredient:* | (flaky pie crust, 1 count, for body) |
| *ingredient:* | (shelled pecans, 2 cup, for taste) |
| *ingredient:* | (eggs, 5 count, for body) |
| *ingredient:* | (white sugar, 1 cup, for taste) |
| *contains:* | (fat, 118 gram) |
| | Call procedure *computeFatContent* |
| *forDiet*: | normal |

**Figure 3: Sample frames with slots and slot fillers**

# The Internal Organization of Documents. Templates

A recipe is a simple document describing a food product, structured into a standard outline or **document template** (a frame applied to documents) with **slots** based on relationships (Figure 4). A template can be encoded using XML (eXtensible Markup Language) tags. Each tag is defined in an XML schema (not shown) and identifies a type of information. (The ability to define tailor-made tags for each application gives XML its power.) Each piece of information has a beginning tag and a corresponding end tag. Once the information is encoded using XML, it can be used for many purposes: to display a recipe in print or on the World Wide Web, produce a cookbook with table of contents and an index, find all recipes that use certain ingredients, compose the ingredient label for a food (ingredients in order of predominance), compute the nutrient values for a serving (using a nutrient value table for basic foods). As this example shows, organization of data in databases and structuring text in documents are alike. In Figure 4, *ingredients* are given in a database-oriented mode (each element tagged separately), *processingSteps* in a text-oriented mode. (just the <*text*> tag; for database-oriented tagging, steps would be broken down into separately tagged *processes*, with data, such as *temperature* and *duration* tagged separately.) These data can then be formatted for text output.

```
<foodProduct>
    <foodName> pecan pie </foodName>
    <unitsMade> <number> 8 </number> <unit> serving </unit> </unitsMade>
    <timeToMake> <number> 1.5 </number> <unit> hour </unit> </timeToMake>
    <description> A custard pie, loaded with pecans.</description>
    <ingredients>
        <foodProduct> flaky pie crust </foodProduct> <number> 1 </number> <unit> count </unit>
        <foodProduct> shelled  pecans </foodProduct> <number> 2 </number> <unit> cup </unit>
        <foodProduct> eggs </foodProduct> <number> 5 </number> <unit> count </unit>
        . . .
    </ingredients>
    <processingSteps>
        <step> 1 </step> <text> Prebake crust . Place pecans on baking sheet and bake </text>
        <step> 2 </step> <text> Start the filling </text>
        <step> 3 </step> <text> Beat the eggs.  Beat in the sugar, salt, and butter </text>
        . . .
    </processingSteps>
</foodProduct>
```

**Figure 4. Recipe following a standard outline (template), encoded with XML**

## Cataloging and Metadata

The recipe/food database or the catalog of a Web store organizes the actual data from which users' questions can be answered. A library catalog organizes data about books, which in turn contain the data to answer questions; the library catalog stores *data about data* or metadata, as do Web search engines and catalogs of educational materials. Metadata are stored and processed just like any other kind of data; whether a data item should be called metadata or just data is often a matter of perspective. The Resource Description Framework (**RDF**) has been designed to encode metadata but can be used to encode any data represented in the E-R approach.

| | |
|---|---|
| • title | • format |
| • creator | • identifier |
| • subject | • source |
| • description | • language |
| • publisher | • relation |
| • contributor | • coverage |
| • date | • rights |
| • type | |

**Figure 5. The Dublin Core (dc) for the description of document-like objects**

There are many standards defining metadata elements for different kinds of objects, for example the Dublin Core ( Figure 5). These are often encoded in XML, for example

<*dc:title*> How to cook everything </*dc:title*>
<*dc:creator*> Mark Bittman </*dc:creator*>
<*dc:subject*> cookbook </*dc:subject*>
<*dc:publisher*> Macmillan </*dc:publisher*>

(Not all records use all dc elements.)
(The pecan pie example is based on a recipe in this cookbook, which also inspired the food type classification)

## Knowledge Organization Systems (KOS)

For the benefit of the user, a cookbook or a grocery store arranges like foods together, just as a library arranges books on one subject together and like subjects close to each other. Such arrangement requires a classification (or taxonomy), such as Figure 6, column 1, for foods, or the Dewey Decimal Classification for all subjects. To describe foods by their characteristics, we need, for each characteristic or facet, a classification of the possible values (the possible fillers for a given frame slot); examples of facets, each with a partial classification of values, as shown in Figure 6.

| Food type | Food source | Plant/animal part | Process | Substance |
|---|---|---|---|---|
| **side dishes**<br>. appetizers<br>. soups<br>. salads<br>**vegetable**<br>**grain/starch dishes**<br>. pasta<br>. grains<br>. breads<br>. pizza<br>**fish, poultry, meat**<br>. fish<br>. poultry<br>. meat<br>**sweet baked dishes**<br>. pies, tarts, pastries<br>. cookies, brownies, and cakes | **plant food source**<br>. Juglandaceae<br>. . Juglans (walnut)<br>. . Carya (Hickory)<br>. . . C. illinoensis<br>. (pecan)<br>. compositae<br>. . Cichorium<br>. . . C. intybus<br>. . . C. endivia<br>**animal food source**<br>. vertebrates<br>. . fish<br>. . bird<br>. . mammal<br>. . . Bovidae<br>. . . . Bos (cattle) | **plant part**<br>. below ground<br>. . root<br>. . tuber<br>. above ground<br>. . stem<br>. . leaves<br>. . fruit (anat. part)<br>. . . seed<br>**animal part**<br>. skeletal meat<br>. organ meat<br>. . liver<br>. egg fruit (anat. part) | **mechanical process**<br>. shelling<br>. peeling<br>. slicing<br>. grating<br>. crushing<br>**cooking process**<br>. c. with dry heat<br>. . baking<br>. . broiling<br>. c. w. microwave<br>. c. w. moist heat<br>. . boiling<br>. . steaming<br>. c. with fat or oil<br>**freezing** | **food substance**<br>. bulk nutrient<br>. . carbohydrate<br>. . . sugar<br>. . . starch<br>. . . fiber<br>. . . . soluble f.<br>. . protein<br>. . fat<br>. trace nutrient<br>. . vitamin<br>. . mineral<br>**non-food substance**<br>. preservative<br>. . BHT<br>. package glue |

**Figure 6. Faceted classification for the food domain. Excerpts**

A **classification** is a structure that organizes concepts into a meaningful hierarchy, possibly in a scheme of **facets**. The classification of living things is a **taxonomy.** (The term taxonomy is increasingly used for any type of classification.) A classification is now often called an **ontology**, particularly if it gives richer concept relationships.

A classification deals with concepts, but we need terms (words or phrases) to talk about concepts. However, the relationships between language and concepts are complex. A concept can be expressed by several terms, such as Belgian endive, French endive, witloof, chicory, and chicon, which all refer to the same vegetable; these terms are in a **synonym relationship** with each other. Conversely, a term may refer to several concepts, such as chicory, which refers (1) to a vegetable and (2) to a coffee substitute made from the root of the same plant; such a term has the property of being a **homonym** (in information retrieval, a character string with multiple meanings). A **thesaurus** is a structure that (1) manages the complexity of terminology by grouping terms that are synonymous to each other and disambiguating homonyms by creating a unique term for each meaning and (2) provides conceptual relationships, ideally through an embedded classification/ontology. A thesaurus often selects from a group of synonyms the term, such as Belgian endive, to be used as **descriptor** for indexing and searching in a given information system; having one descriptor for each concept saves the searcher from having to enter several terms for searching. The descriptors so selected form a **controlled vocabulary** (authority list, index language). Figure 7 shows a typical thesaurus entry.

| Belgian endive | Symbols used |
|---|---|
| DF Vegetable consisting of the leaves of Chicorium intybus, growing in a small, cylindrical head.<br>COmbination: vegetable : Cichorium intybus : leaves<br>UF chicon<br>chiccory (vegetable) [spelling variant]<br>chicory (vegetable)<br>French endive<br>witloof<br>BT head vegetable<br>salad vegetable<br>RT chicory (coffee) | DF Definition<br>UF Used For<br>USE<br>BT Broader Term<br>NT Narrower Term<br>RT Related Term |

**Figure 7. A typical thesaurus entry**

Rich conceptual relationships can be shown graphically in **concept maps**, which are used particularly in education to aid understanding; they represent **semantic networks**, which a user or a computer can traverse along links from one concept to the next (a process called spreading activation). Conceptual and terminological relationships can be encoded for computer storage using the Topic Map standard or RDF, both implemented in XML.

## Outlook

Information organization is important for people to find and understand information. It is also important for computer programs to process information to make decisions or give recommendations, for example in medical expert systems and electronic commerce (ecommerce) and semantic Web applications (where information organization is called "knowledge representation"). These applications require well-thought-out conceptual structures which must be developed by beginning from scratch or by refining existing knowledge organization systems (KOS). The most serious challenge is ensuring the interoperability of KOS and metadata schemes worldwide so that different systems can talk to each other.

Dagobert Soergel

*See also* Expert Systems; Information Retrieval; Markup Language

**Further Reading**


Bailey, K. D. (1994). *Typologies and taxonomies: An introduction to classification techniques*. Thousand Oaks, CA: Sage Publications.

Dodds, D. (2001). *Professional XML metadata*. Hoboken, NJ: Wrox.

Jonassen, D. H., Beissner, K., & Yacci, M. (1993). *Structural knowledge: Techniques for representing, conveying and acquiring structural knowledge*. Hillsdale, NJ: Lawrence Erlbaum.

Lancaster, F. W. (1972). *Vocabulary control for information retrieval*. Washington, DC: Information Resources Press.

Lynch, P., & Horton, S. (2002). *Web style guide: Basic design principles for creating Web sites* (2nd ed.). New Haven, CT: Yale University Press.

Milstead, J., & Feldman, S. (1999). Metadata: Cataloging by any other name . . . Metadata projects and standards. *Online, 23*(1), 24<N>40. Retrieved January 22, 2004, from www.infotoday.com/online/OL1999/milstead1.html

Mondeca topic organizer. Retrieved January 22, 2004, from http://www.mondeca.com/

Ray, E. (2003). *Learning XML* (2nd ed.). Sebastopol, CA: O'Reilly

Rob, P., & Coronel, C. (2004). *Database systems: Design, implementation, and management* (6th ed.). Boston: Course Technology.

Rosenfeld, L., & Morville, P. (2002). *Information architecture for the World Wide Web: Designing large-scale web sites* (2nd ed.). Sebastopol, CA: O'Reilly.

Skemp, R. R. (1987). *The psychology of learning mathematics*. Hillsdale, NJ: Lawrence Erlbaum.

Soergel, D. (1974). *Indexing languages and thesauri: Construction and maintenance*.

New York: Wiley.

Soergel, D. (2000). ASIST SIG/CR Classification Workshop 2000: Classification for user support and learning: Report. *Knowledge Organization, 27*(3), 165<N>172.

Soergel, D. (2003). Thesauri and ontologies in digital libraries. Retrieved January 22, 2004, from http://www.clis.umd.edu/faculty/soergel/SoergelDLThesTut.html

Sowa, J. F. (2000). *Knowledge representation: Logical, philosophical and computational foundations*. Pacific Grove, CA: Brooks/Cole.

Staab, S., & Studer, R. (Eds.). (2004). *Handbook on ontologies in information systems*. Heidelberg, Germany: Springer.

Taylor, A. G. (2003). *The organization of information* (2nd ed.). Westport, CT: Libraries Unlimited.

Vickery, B. C. (1960). *Faceted classification: A guide to construction and use of special schemes*. London: Aslib.

Vickery, B. C. (2000). *Classification and indexing in science*. Burlington, MA: Butterworth-Heinemann.

Zapthink. (2002). Key XML specifications and standards**.** Retrieved January 22, 2004, from http://www.oasis-open.org/committees/download.php/173/xml%20standards.pdf XXX