Data models for an integrated thesaurus database

Dagobert Soergel

College of Library and Information Services
University of Maryland

ds52@umail.umd.edu www.clis.umd.edu/faculty/soergel/

Paper presented at the Research Seminar on Compatibility and Integration of Order Systems, Warsaw, Poland, September 13-15, 1995.
Published in Compatibility and Integration of Order Systems: Research Seminar. Proceedings of the TIP/ISKO Meeting. Issued by International Society for Knowledge Organization; Polish Library Association; Society for Professional Information. Warsaw: Wydaw. SBP; 1996. p. 47-57.

Abstract

This paper presents two data models for storing multiple thesauri in a single integrated database to be used as an aid to searchers in multi-database searching, for the construction of conversion tables between thesauri, and as a tool for constructing and maintaining individual thesauri. The paper first describes the nature of thesaurus data and a relational data structure for such data, which is flexible and — through its use of term numbers in recording relationships — economical in storage. It then describes two data models for structuring an integrated thesaurus database. In both models, general data on terms and relationships are stored once, with indication of one or more sources, resulting in storage economy. The term-based model stores all relationships as relationships between terms. This is flexible but redundant: If the same concept relationship is expressed through different terms in different thesauri, it is stored multiple times in the integrated database. The concept-based model identifies concepts by concept numbers and uses these concept numbers to record concept relationships, thus bringing together all occurrences of the same concept relationship regardless of the terms used to express the related concepts. This results in more compact storage but is less flexible.

Introduction

This paper presents two data models for storing multiple thesauri in a single integrated database. Such a database can serve the following purposes:

- Aid searchers in finding the appropriate descriptors for a search, particularly a search in multiple data bases. The searcher could start from any term, or from general concepts, or from a known descriptor in another data base, to find the appropriate descriptors and/or free-text terms for the data base at hand.

- Produce indexing and searching conversion tables between index languages of different data bases and support semi-automatic indexing conversion and conversion of query formulations from one data base to another.

- Print subsets of the data base with subset-specific selection of preferred terms and descriptors. This can be used for producing any thesaurus that was used as a source in generating the data base, for improving and maintaining any such thesaurus, and for producing new specialized thesauri.

Structure of thesaurus data

A thesaurus deals with terms and concepts and the relationships among and between these entities. A term is a linguistic entity, a character string with meaning in a given language. If the same character string has two meanings, we have two terms (homonyms); most thesauri use parenthetical qualifiers to make each string unique. The same character string occurring in two different languages represents two different terms, even if the meaning is the same.

A thesaurus captures a great many relationships among terms, between terms and concepts, and among concepts. **Term-term relationships** include A *has morphological variant* B (such as *job* and *jobs*), A *has spelling variant* B (such as *labor* and *labour*), and A *has synonymous term (ST)* B. More precisely, *has morphological variant* relates character strings that are derivative from the same stem. *Has spelling variant* relates stems and partitions the set of all stems into mutually exclusive groups. Each such group constitutes a normalized term; a preferred spelling variant can be selected to represent the term. (To keep matters simple, we mostly sidestep the spelling variant problem in this paper.) *Has synonymous term* relates normalized terms (preferred spelling variants) and partitions the set of normalized terms into mutually exclusive groups. A concept can be operationally defined as such a group of normalized terms. A preferred term can be selected from each group to uniquely designate the concept. All preferred terms may be used as descriptors, or descriptors may be further selected from the preferred terms. These considerations give rise to a status hierarchy among all terms (character strings).

The primary **term-concept relationship** is Term A *designates* Concept B; this relationship is implied by *has synonymous term* relationships, unless a thesaurus identifies concepts independently, for example through class numbers or notations. **Concept-concept relationships** include A *has broader term* B, A *has narrower term* B, A *has related term* B.

This simplified picture presents clear-cut distinctions, but reality is not that simple. Normalized terms often represent shades of meaning so that it is hard to tell whether two terms are synonyms or whether they represent closely related but different concepts. If the two concepts that are so closely related that to distinguish between them would not be useful for retrieval, some thesauri use the relationship *equivalent term* (ET) at least in their internal database (in the user version they may map ET to ST). The ET relation can be seen as partitioning the set of concepts into mutually exclusive groups. Each group corresponds to a newly formed ISAR (Information Storage And Retrieval) concept which is broader than any concept in the group. A preferred term can then be selected for each ISAR concept. The *equivalent term relationship* is at the borderline between term-term relationships and concept-concept relationships. For the term-based model (discussed below) this does not present a problem, but for the concept-based model one must decide whether to treat ET as a term-term relationship or a concept-concept relationship. This problem is particularly thorny for an integrated thesaurus database since terms that are equivalent from the

point of view of one constituent thesaurus may need to be distinguished for retrieval purposes in another thesaurus.

Note. The term-term relationships and the equivalence relationship are equivalence relations in the mathematical sense (they are both reflexive and transitive) and thus partition the set in which they hold into mutually exclusive groups.

A complex thesaurus may further differentiate relationship types (for example, distinguish between genus-species and whole-part hierarchical relationships) and include other types of relationships. In addition, the selection of preferred terms and, from them, descriptors, gives rise to **instructions** which can be combined with the term-term and concept-concept relationships:

SEE refers from a non-preferred term to a preferred lead-in term

SF is the reciprocal

USE refers from a non-preferred term or from a preferred lead-in term to a descriptor

UF is the reciprocal.

Thus, an individual thesaurus at a given time is a complex, highly interrelated structure. Adding the time dependency of terms and concepts, their status and their relationships, increases complexity. Integrating several thesauri into one data base while maintaining their individual identities, increases complexity still further. The data structure of an integrated thesaurus database must be able to handle this complexity efficiently.

Data structures for thesaurus databases

Some computer systems for thesaurus construction and maintenance use a record for every term with the information about the term, such as synonyms, broader, narrower, and related terms, stored in — usually repeating — data fields in the record (Figure 1a). Information is stored in large packages, and to access or change any piece of information we must get into the appropriate package. Even for an individual thesaurus such a structure is inflexible. For an integrated thesaurus data base it is unwieldy. For example, comparing two records for the same term from two different thesauri requires cumbersome processing of the two records.

The relational approach to data base organization leads to a more elegant and efficient structure (Figure 1b). Information is stored in individual pieces that can be arranged in different ways. For example, *employment* RT *labor relations* is a piece of information that is stored by itself. Combining two thesauri stored in this format can be accomplished simply by putting all the pieces of information into one data base and eliminating duplicates. This structure has an additional advantage: Relationship types are not defined as fields in a record (and thus fixed in the database structure), but they are simply data values in a relationship record; thus new relationship types can be introduced with ease.

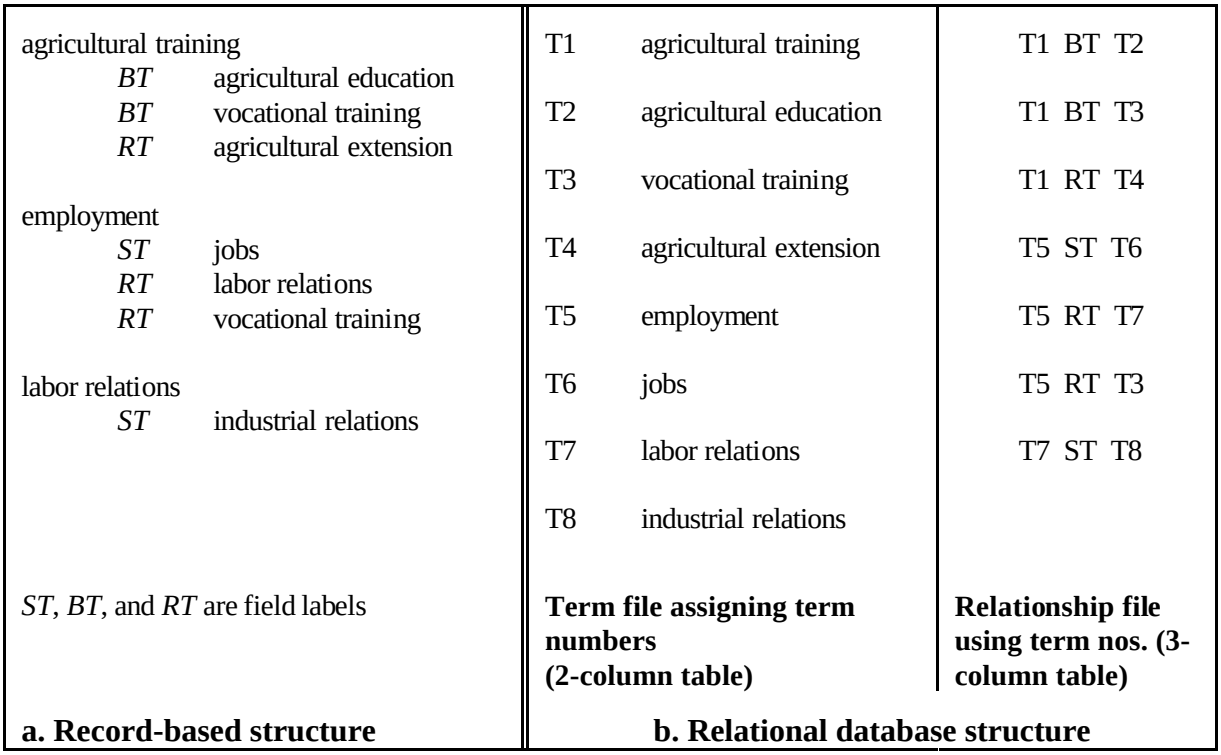

Figure 1. Data structures for a single thesaurus

Furthermore, some thesaurus databases are fashioned after the structure of a printed thesaurus and use the full term string wherever a term is referred to. This introduces considerable redundancy: The same lengthy terms appear over and over again in cross references. In an integrated thesaurus database, the redundancy becomes even more severe; terms, concepts, and relationships are repeated. Efficiency of storage can be achieved by assigning each term a four-byte number and using these term numbers in all relationship pairs. (For clarity, term numbers will be represented in this paper as T1, T2, T3 etc., relationship numbers as R1, R2, R3, etc., and concept numbers as C1, C2, C3, etc. In a real system, these would be plain 4-byte numbers whose meaning is determined by the file and the data field in a record.) Figure 1 compares two data structures for a single thesaurus.

Data models for an integrated thesaurus database

A sample set of data to be stored in an integrated thesaurus database is given in Figure 2. It consists of terms and relationships structured according to the relational data structure, with terms spelled out (rather than represented by term numbers). The data in the data set have been collected from several thesauri, each identified by a three-letter abbreviation. The data from all thesauri were combined and the resulting pool sorted alphabetically by main term, relationship type, and cross term.

Main term	Rel. Type	Cross term	Thesaurus
agricultural training	MT		UNE
agricultural training	MT		MAC
agricultural training	ST	farmer training	MAC
agricultural training	TR	formation agricole	MAC
agricultural training	BT	agricultural education	UNE
agricultural training	BT	vocational training	UNE
agricultural training	BT	vocational training	MAC
agricultural training	RT	agricultural education	MAC
agricultural training	RT	agricultural extension	UNE
agricultural training	RT	experimental farm	MAC
employment	MT		UNE
employment	ST	jobs	UNE
employment	MT		MAC
employment	MT		ERI
employment	RT	industrial relations	ERI
employment	RT	labor relations	UNE
employment	RT	labor relations	MAC
employment	RT	vocational training	UNE
job	MT		KAS
job	RT	employee relations	KAS
labor relations	MT		UNE
labor relations	ST	industrial relations	UNE
work	MT		DRI
work	MT		ZID
work	MT		DAS
work	ST	employment	DAS
work	RT	industrial relations	DRI
work	RT	labor relations	ZID
work	RT	labor relations	DAS

Figure 2. Data from several thesauri combined

We will present two data models to handle these data, a **term-based data model** and a **concept-based data model**. **Both data models use the same method for organizing data about terms as illustrated in Figure 3**. The **term file** contains one record for each term stem. Thus if the term occurs in the singular in one thesaurus and in the plural in another, the integrated thesaurus database has only one record. The term record also gives the language of each term. (Remember that the same character string used in different languages represents different terms.) The **term source file** records for each term the thesauri in which it occurs; if a term occurs in three thesauri, it has three records in the term source file. Each record also gives the suffix that must be combined with the term stem to arrive at the exact form of the term used in the specific thesaurus. For example, UNE uses the form *jobs* while KAS uses the form *job*. In most cases, the stem is also the singular, but not always (for example, stem *hypothes*, singular suffix *is*, plural suffix *es*. This model of term data takes care of the morphological variation most important for thesauri, singular/plural variation. Other types of morphological variation could be accommodated through elaboration of the model.

The term file contains general information, the term source file contains information specific to each thesaurus. This distinction, which recurs in other files discussed below, is very important for compact storage of data. A source is any thesaurus in the integrated database, be it an independent thesaurus that was used in creating the integrated database or a new thesaurus created from the integrated database.

The two data models use different methods for organizing data about relationships.

Relationships in the term-based data model. All relationships are stored explicitly as they occur in the sources. Terms are referred to by their numbers. If the same relationship (same terms, same relationship type) occurs in several thesauri, it is stored only once while preserving the information about the individual thesauri in which the relationship appears. This give rise to the structure shown in Figure 4. The **Term relationship file** stores each relationship once, identified by a relationship number. The **term relationship source file** stores for each relationship the thesauri in which it occurs.

In the term-based model, some of the stored relationships are term-term relationships (*has synonymous term* or ST for short), while others are concept-concept relationships. The concepts are expressed through the preferred term used in the individual thesaurus in which the relationship appears. Thus in the term-based model term-term relationships and concept-concept relationships take the same form. Concepts are not represented explicitly. Consequently, concept-term relationships are not made explicit but implied

A database structured according to the term-based model can be established very easily: Simply pool the data from various thesauri, each structured according to the relational structure described above, as follows: Keep a running list of terms, sorted alphabetically, with term numbers (term file, Figure 3a); as a new thesaurus is added, check all terms against the list and add new terms. Record the thesaurus as a source for any term, existing or new (term source file, Figure 3b). In relationships, replace the terms by term numbers. Keep a running list of

relationships, each identified by a relationship number, sorted by main term, relationship type, and cross term (term relationship file, Figure 4a). As a new thesaurus is added, check all relationships against the list and add new relationships. Record the thesaurus as a source for any relationship, existing or new (term relationship source file, Figure 4b)

The term-based model allows for great flexibility: It can capture relationships between various forms of the same term (spelling variants, abbreviations) and degrees of synonymity (*synonymous term, equivalent term*). It allows for one thesaurus having A ST B while another has A RT B; the administrator of the integrated thesaurus database need not make a decision between the two. The price for this flexibility is a lot of storage space — one and the same conceptual relationship is stored as often as it has linguistic expressions in the constituent thesauri — and less efficient processing — to collect all the occurrences of a conceptual relationship the program must trace all the terms for each of the concepts involved. To trace all terms for a concept, the program must start from one term, find all its synonyms, find all their synonyms, etc. This process is very laborious and error-prone: A single erroneous *synonymous term* relationship can lead to undesirable results. A program using this model should let the user specify that only relationships from a select list of thesauri should be used in the tracing process.

Relationships in the concept-based model. In the concept-based model, concepts are identified explicitly thorough concept numbers. Accordingly term-concept relationships (or concept-term relationships) are given explicitly in a **concept-term file** (Figure 5), which links each (disambiguated) term with exactly one concept. Thus the concept-term file has one record for each term and for each concept as many records as there are terms designating the concept. Term-term relationships (*has spelling variant, has synonymous term*) on the other hand are represented implicitly: All terms linked to the same concept are synonyms or spelling variants. Concept-concept relationships (BT, NT, RT) are established explicitly between concepts in the **concept relationship file** (Figure 6a), with sources for each relationship indicated in the **concept relationship source file** (Figure 6b).

Establishing an integrated thesaurus database according to the concept-based model requires considerable effort: All terms for a concept must be brought together. While the ST relationships from the constituent thesauri are very helpful in this process, thesaurus 1 may contain term A and thesaurus 2 the synonymous term B without a relationship A ST B recorded anywhere; discovering this relationship, which is required for the proper construction of the concept-term file, takes intellectual effort

The concept-based data model is efficient for storage and processing, since each conceptual relationship, while it may expressed using various terms in the constituent thesauri, is stored only once in the integrated database. However, it is also inflexible: Decisions on term-concept relationships must be made once and for all and are then binding on all thesauri in the database; thus we cannot have A ST B in one constituent thesaurus and A RT B in another. There is also no distinction between spelling variants, and synonyms.

Term no.	Term stem	Language
T1	agricultural training	EN
T2	farmer training	EN
T3	formation agricole	FR
T4	agricultural education	EN
T5	vocational training	EN
T6	agricultural extension	EN
T7	experimental farm	EN
T8	employment	EN
T9	job	EN
T10	industrial relation	EN
T11	labor relation	EN
T12	employee relation	EN
T13	work	EN

One record for each term, each term identified by a term number.

a. Term file

Term no.	Thesaurus (source)	Suffix	Term Type
T1	UNE		DE
T1	MAC		DE
T2	MAC		NP
T3	MAC		DE
T4	UNE		DE
T4	MAC		DE
T5	UNE		DE
T5	MAC		DE
T6	UNE		DE
T7	MAC		DE
T8	UNE		DE
T8	MAC		DE
T8	ERI		DE
T9	UNE	s	NP
T9	KAS		DE
T10	UNE	s	NP
T10	ERI	s	DE
T10	DRI	s	DE
T11	UNE	s	DE
T11	MAC	s	DE
T11	ZID	s	DE
T11	DAS	s	DE
T12	KAS	s	DE
T13	DRI		DE
T13	ZID		DE
T13	DAS		DE

Multiple records for each term as needed, linked to term file through term number.

b. Term source file

Figure 3. Both data models: Term file and term source file.

Rel no	Main term no	Rel type	Cross term no
R1	T1	ST	T2
R2	T1	TR	T3
R3	T1	BT	T4
R4	T1	BT	T5
R5	T1	RT	T4
R6	T1	RT	T6
R7	T1	RT	T7
R8	T8	ST	T9
R9	T8	RT	T10
R10	T8	RT	T11
R11	T8	RT	T5
R12	T9	RT	T12
R13	T11	ST	T10
R14	T13	ST	T8
R15	T13	RT	T10
R16	T13	RT	T11

One record for each relationship, each relationship identified by a relationship number.

a. Term relationship file

Rel no	Thesaurus
R1	MAC
R2	MAC
R3	UNE
R4	UNE
R4	MAC
R5	MAC
R6	UNE
R7	MAC
R8	UNE
R9	ERI
R10	UNE
R10	MAC
R11	UNE
R12	KAS
R13	UNE
R14	DAS
R15	DRI
R16	ZID
R16	DAS

Multiple records for each relationship as needed, linked to the relationship file through the relationship number.

b. Term relationship source file

Figure 4. Term-based model: Term relationship file and relationship source file.

Concept no	Term no	Term (for the convenience of the reader, not part of the file)
C1	T1	agricultural training
C1	T2	farmer training
C1	T3	formation agricole
C2	T4	agricultural education
C3	T5	vocational training
C4	T6	agricultural extension
C5	T7	experimental farm
C6	T8	employment
C6	T9	job
C6	T13	work
C7	T10	industrial relation
C7	T11	labor relation
C7	T12	employee relation

Figure 5. Concept-based model: Concept-term file.

Rel no	Main concept no	Rel type	Cross concept no
R1	C1	BT	C2
R2	C1	BT	C3
R3	C1	RT	C2
R4	C1	RT	C5
R5	C2	RT	C4
R6	C6	RT	C3
R7	C6	RT	C7

a. Concept relationship file

Rel no	Thesaurus
R1	UNE
R2	UNE
R2	MAC
R3	MAC
R4	MAC
R5	UNE
R6	UNE
R7	UNE
R7	MAC
R7	KAS
R7	ERI
R7	DRI
R7	ZID
R7	DAS

b. Concept relationship source file

Figure 6. Concept-based model: Concept relationship file and conc. rel. source file.

Some of the limitations of the concept-based model can be overcome at the cost of added complexity. A separate file connecting spelling variants to the normalized term would maintain the distinction between spelling variant and synonym. Equivalent terms, which are lumped together with synonyms in the simple model presented, can be kept distinct as follows: Include separate concept number for each of the equivalent concepts in a group and introduce a new concept which is above all the concepts in the group. Only the broad concept is selected as descriptor. As an example, consider the group of equivalent terms (with their concept numbers) *Disease* (C35), *Illness* (C45), *Sickness* (C67), and *Ailment* (C73). Each would be treated as a distinct concepts. A new concept subsuming all of them would be introduced and called, perhaps, *Disease (broadly defined)* (C87). The concept relationship file would contain

C35 BT C87
C45 BT C87

etc. Most thesauri would select just C87 as descriptor, others might need the increased specificity of the narrower concepts, giving good scope notes that would help the indexers decide on the right descriptor in each case. On could introduce a more precise relationship BT-EQ; the rules for extracting an individual thesaurus from the integrated database might stipulate that BT-EQ should be converted to ST or ET.

Concluding remarks

This paper described just the basic elements of each model. A real system needs to include a lot more data, for example the date when a descriptor was introduced and when it was discontinued.

A prototype software package for maintaining an integrated thesaurus database is described in Soergel 1994. This package uses the term-based model. It allows for a wide variety of outputs exploiting all the information in the database.

Bibliography

Soergel, Dagobert 1995. **Software support for thesaurus construction and display**. Advances in Classification Research. v. 5 (Proceedings of the 5th ASIS SIG/CR Classification Research Workshop, Oct. 1994). Medford, NJ: Learned Information; 1995. p. 157-184.