Dagobert Soergel
College of Library and Information Services
University of Maryland
Office: (301) 405-2037 Home: (703) 823-2840
Fax: (301) 314-9145 E-mail: ds52@umail.umd.edu (or dsoergel@...)

Software support for thesaurus construction and display

Thesauri are complex structures that must be displayed in a variety of print and online formats. Computers can be very helpful in developing and maintaining thesauri and creating a large variety of formats. This paper presents a number of desirable advanced functions; many of these are implemented in TermMaster, a thesaurus program under development.

By way of introduction, Figures 1a-d show several output formats (from the Alcohol and Other Drugs Thesaurus) that illustrate formatting capabilities. Note the three levels of detail in the hierarchical lists. In the annotated hierarchical list, note the running heads, the hierarchical context lines on left pages (right pages do not need them), the term numbers with the crossreferences (+ means that the term has narrower terms), and the use of typography. In the scope note for JD, note the bolded descriptor with preceding number; such embedded descriptors are marked in the input file and the program does the rest. In the alphabetical (KWOC) index, note that the access word **model** assembles all multi-word terms that contain either *model* or *models.*

The bulk of this paper deals with support for thesaurus development. A lot of the knowledge to be included in a new thesaurus is already available in other thesauri and dictionaries. TermMaster can maintain a database that includes multiple thesauri. A thesaurus to be included must first be transformed into one of six input formats. Each thesaurus can then be processed individually, but the real benefits are achieved by combining data from several thesauri or comparing a target thesaurus (the thesaurus currently being worked on) with one or more source thesauri.

Figures 2a-c present three thesaurus samples in a hierarchical input format (a and c) and an alphabetical input format (b).

The heart of a good thesaurus is a well-structured hierarchy; therefore, the program provides extensive support for processing hierarchies. It can read a hierarchy presented in the usual format — linear arrangement with indentions — and preserve the meaningful sequence by assigning notations, or storing user-assigned notations, or a mixture of the two. In Figure 2a, the lexicographer has assigned one- or two-letter notations to broad terms; the program takes over from there. TermMaster creates explicit records for the hierarchical relationships implied by the arrangement.

A thesaurus term is linked to other terms through a variety of relationships. TermMaster allows a large number of such relationships, making it possible, for example to distinguish in the database (not necessarily in the user version) between ST (Synonymous Term), and ET (Equivalent Term, quasi-synonym) (see, for example, under *elderly*); one might even use, in addition, SP (SPelling variant) and AB (ABbreviation). Scope notes are treated as relationships to text; thus a term can have multiple scope notes and there can be different kinds of scope notes, such as History Notes or Internal Notes (See again under *elderly*).

Some relationships are more important than others; TermMaster allows for (but does not require) specifications of three levels of relationship importance which can be used to govern the inclusion of relationships in various printouts (not illustrated). The program can easily be customized to include a user-defined set of relationships.

At input, TermMaster reduces terms to singular (unless overridden), so a term can be recognized as the same even if one thesaurus uses singular and the other plural. The program does keep the information on the term form for each thesaurus in the database.

TermMaster can display the contents of the thesaurus database in many ways. Files for printing are output as WordPerfect 5.1 documents. The user has considerable control over the content of each output and over features such as type font and size or number of columns; such features are specified in a print command file.

The chief **output formats for publication** are:

Hierarchical list. Figure 3a shows a print command file for a typical annotated hierarchical list of thesaurus AOD, Figure 4a shows the result. A hierarchical list can be indented as shown (printout type HI) or aligned at the left margin (HL.). Levels are always indicated by a superscript to the notation. The user can control many features, for example: the number of levels to be included, the appearance (large, bold) for each level, the appearance of synonyms (italic or regular), the first hierarchical level to be shown by indention, and the information to be included for each term; relationship types can be grouped (e.g., SP, AB, ST, ET together), with the external label specified at printout time (e.g., all four could be mapped to ST). Relationships can be selected by importance level. For properly marked terms included in a scope note, the program inserts the notation and replaces the term by the corresponding preferred term. Figure 3b shows a sample print command file for thesaurus s2 with different parameters for the levels; Figure 4b shows the result. The print command file for a quick hierarchical list (see Figure 1b) would simply omit the relationship specifications. An outline (Figure 1a) is a special quick hierarchical list that includes all descriptors whose notation consists entirely of letters and indention starting with level 1.

Alphabetical list (under development). Same user control over information included.

Alphabetical KWOC index. A multiword term appears under the singular access word even if the word appears in the term in the plural. A sample print command file is shown in Figure 3c, the result in Figure 4c.

Subsets. Different uses of the same overall thesaurus often require different subsets to be printed out separately. A subset is different form a separate thesaurus in that all subsets use the same basic hierarchical structure and notations stay the same across subsets. A subset may also be established to include specially important descriptors that should be marked with a special symbol. TermMaster supports the definition and use of up to 80 subsets. In Figures 1a-d, the subscript e following the notation indicates that the descriptor belongs to the subset of descriptors used to index the Alcohol Science Database (ETOH).

There are three **output file types that assist in thesaurus development**.

Edit print with sources. At present, interaction with TermMaster is batch-oriented. The program can produce a plain ASCII file which can be edited with any word processor and re-input; for many changes, this process is actually more convenient than online editing. To assist the lexicographer, the program also produces a nicely formatted annotated hierarchical list, called the reference print. The print command file is shown in Figure 5a; AOD is the target thesaurus to be worked on, s1 and s2 are two sources that are to be consulted for more information. The two resulting files are shown in Figures 6a1 and 6a2. Starting from a target term, the program collects all its synonyms using relationships of the ST-group (ST, ET, SP, AB) from any of the specified sources (in the example s1 and s2). For example, starting from *adolescent* it finds *pubescent* (ET relationship in AOD), which leads to *teenager* (ET relationship in s1) which leads to *youth (young person)* (ST relationship in s2). The program than assembles — under the original target term - *adolescent* all non-ST

relationships from all the sources starting from any of the synonyms. It replaces the term referred to by the term preferred in the target (if available). Thus the program brings together all relationships that are conceptually the same but expressed in different terms as long as these terms are shown as synonymous in the sources used.

An edit or reference print gives relationships with their specific symbols (no mapping) in an order determined at the time the program is compiled. The reference print gives for all main terms and synonyms the sources in which they occur. Relationships new to the target are flagged with (+); if the crossreferenced term does not occur in the target, it has no notation. In the future, the program will list all the sources for a relationship.

TermMaster provides further support for the development of the relationship network: Individual words in a multi-word term are isolated as tentative semantic factors, and thus broader terms, subject, of course, to editing. At the users option, these relationships can be shown in an edit print. A more sophisticated algorithm assigning tentative semantic factors through inheritance from broader terms is under development.

Comparison print. The purpose of a comparison print is to identify terms that occur in any of a number of sources but are missing from the target thesaurus being worked on — a check on completeness. More generally, the purpose is to see how the terms from the source thesauri are treated in the target thesaurus. The print command file is shown in Figure 5b, the resulting files in Figures 6b1 and 6b2. The long version (Figure 6b2) lists all terms from the target thesaurus that have at least one word in common with the missing term; this facilitates identifying a synonym in the target thesaurus or, if none is found, finding a good place for the missing term in the target thesaurus. For example, AOD does not contain the s2 term *persons by type of residence*, but under the component word *residence* the lexicographer sees the corresponding AOD term *status by type of neighborhood of residence*. Figure 6b3 gives a page from a real comparison print; it makes it easy to find the AOD term corresponding to the ESTES term *alcohol treatment facility*.

Assembled hierarchy (no example shown). A rough draft hierarchy pieced together from binary hierarchical relationships from specified source thesauri.

In the example, the lexicographer edited file 6a2 with the aid of the reference hierarchy 6a1 and the comparison print 6b1 and 6b2. The resulting hierarchy with additional terms and relationships is shown in Figure 7. For example, *adolescent* was changed to *teenager*, following the lead of the two sources. *Teenage mother* was added. A better scope note for *elderly* was found in a source.

Lastly, the program can produce **output files for use with an information retrieval package.** In particular, the program can construct, from a hierarchy with crossreferences, an **expanded hierarchy for inclusive (hierarchically expanded) searching**. The regular hierarchy shows each descriptor at one place in the linear hierarchy, with one notation, and expresses additional relationships through crossreferences. In the expanded hierarchy (Figure 8a), a descriptor appears under each of its broader terms and thus has multiple notations (as in the Medical Subject Headings "Tree"). For example, *runaway youths* appears as TA8.6.2, TZ2.2.2, and TZ4.2.6.2. TermMaster's expansion preserves the structure of the hierarchy and the original notations, inserting additional descriptor listings at the appropriate place and creating the additional notations. If a whole branch of the hierarchy is repeated in a second location, it maintains its original structure. The expanded hierarchy is useful for implementing inclusive searching in a brut-force approach: Use the alphabetical listing shown in Figure 8b to enrich bibliographic records through adding all the notations of a descriptor, thereby making the record retrievable from any broader descriptor in any system that provides truncation. Figures 8c and give an example with a more complex structure.

A last feature of TermMaster that should be mentioned is its system of status codes. Every term and every relationship is marked for each thesaurus as to whether it is fully accepted, should be reviewed, or has been rejected or deleted. Thus rejection/deletion decisions are fully documented and available when a term or relationship comes up for consideration again, for example when examining a newly published thesaurus for new terms and relationships.

TermMaster has two **modes of interaction**: batch (implemented) and online (under development). **Batch files** are convenient for entering and editing large linear hierarchies, which can then be input into the database. As described above, editing is accomplished through producing an output file for editing and re-input. The program assigns a special status code to all pieces of information in such a file. If a piece of information is still present when the edited file is re-input, the status code is changed back to valid; otherwise, the status code remains and the piece of information is considered deleted from the thesaurus being worked on. (The information stays in the database for check by a senior editor, if desired, and for later reference.)

The planned **online interaction** is term-based. A term has a screen displaying all the information about that term (relationships, scope notes) in the database, with source and status indication. (This it would show relationships previously deleted). After full-screen editing, the changes are made in the database. Online editing of a "live" linear hierarchy is desirable but requires a very high effort for implementation, particularly if one wants to make available word processor functionality.

There are, of course, many functions and detailed specifications of TermMaster that were not mentioned in this short paper, which emphasized special features not widely available on personal computers (TermMaster runs under MS-DOS; 486 and fast hard disk recommended).