Dagobert Soergel
504 North Quaker Lane
Alexandria, VA 22304
Home: (703) 823-2840 Office: (301) 454-5451

Design of an Interlinked Food Description (IFD) Database

A New Food Description System

Expanded and revised from the initial Factored Food Vocabulary

Including a conceptual schema and procedures for the use of the new system

Prepared for
National Cancer Institute. Diet and Cancer Branch
Silver Spring, MD

Order # 263 SVC15296

June 3, 1988

Revised June 20, 1988

Contents

Introduction 1

Conceptual schema and thesaurus structure for an Interlinked Food Description (IFD) database 2

Interlinked Food Descriptions database Indexer interface specifications 18

Construction of the prototype 23

Introduction

This document discusses the structure of a database of food descriptions that is a further development from FDA's Factored Food Vocabulary (FFV). The new system, called Interlinked Food Description (IFD) allows for recording more information, especially full ingredient indexing. It accommodates the complex structure of foods, where one food product is often an ingredient of a more complex food product. The concept of such a system is described in Soergel et al. 1988. This document discusses the design in more detail and lays the foundation for the construction of a prototype.

The first step in the process of making a decision on implementation of the new system is a test of feasibility and usefulness through development of a prototype. Only after such a test can decisions about large-scale implementation be made. Such decisions must weigh the expected benefits against cost, including the cost of reformatting large existing databases. These databases are not restricted to food databases but include other data bases that use the FFV.

When the test is completed, intermediate solutions should also be considered, such as doing full ingredient indexing but describing each ingredient using the current FFV rather than the more elaborate new method.

Any implementation can and must maintain compatibility with the present FFV in both directions. It must be possible to convert FFV food descriptions into the IFD format, preserving all information. (However, the converted descriptions will not contain all information that can be captured in an IFD system.) It must be possible to convert an IFD description into the FFV to enable parallel operation.

The development of the prototype requires the ideas and critical input from all participants in the evolving scheme for cooperation in the area of food data. It should be done under the aegis of the Steering Committee and the Technical Committee as discussed at the January meeting in Washington.

The following remarks concern the development of the prototype. The major part of the work done will be useful for further development if the approach proves to be feasible. Again, the experience gathered through the prototype is essential for intelligent decisions about the future system.

Conceptual schema and thesaurus structure for an Interlinked Food Description (IFD) database

This section describes a conceptual schema and the structure of a thesaurus needed in a food description database which is structured as an interlinked network of food description using the entity-relationship approach as discussed in Soergel et al. 1988. It furthermore gives directions on how to develop such a thesaurus through restructuring and expanding FDA's Factored Food Vocabulary (FFV).

Figure 1. **Conceptual schema for a moderately detailed IFD database**

Use concsch.new here

Interlinked Food Descriptions database

Indexer interface specifications

Indexing for an IFD database means to weave the new food"s description into the interlinked network of existing descriptions. Often it means creating the building blocks that are needed to build the description of a new food; this is especially true when the database is still small.

The following is a semi-formal description of system actions from beginning to end of indexing a food. The indexer's responses are usually obvious; if they are not, they are spelled out.

1 2 3 4 5

PROCEDURE main

Ask for a Universal Product Code (UPC)

```
TEST UPC for null (1)
    CASE   UPC not null
        TEST whether UPC is already in the database (2)
            CASE   UPC is already in the database
                display the product name
                TEST whether description in database is complete (3)
                    CASE description complete
                        ask indexer whether s/he wants to review the description
                        TEST answer yes/no (4)
                            CASE yes
                                display description for editing
                            CASE no
                                end of transaction (no action needed)
                            ENDTEST (4)
                    CASE description incomplete
                        display description and ask for missing information
                    ENDTEST (3)
            CASE UPC not yet in database
                DO new-product
            ENDTEST (2)
    CASE UPC null
        DO new-product
    ENDTEST (1)
```

1 2 3 4 5

PROCEDURE new-product

Ask for product name. The indexer should answer as specifically as possible; at the very least s/he should enter the name of a product type (that is, a generic product included in the thesaurus).

TEST whether product name is in database
CASE product name in database
DO display-frame for the product
CASE product name not in the database
analyze name given to extract information; this may give enough information to display a frame with some values filled in. If all else fails, ask for product type
DO display-frame for the product type
ENDTEST

PROCEDURE enter-new-product

Display the description of the product entered by the user for editing. Arrange the statements describing the product grouped by relationship type in the sequence given in the conceptual schema. (This sequence corresponds, by and large, to the sequence of the factors in the FFV.)

Ask the indexer whether the description is correct as is or whether it needs to be edited.

CASE description correct as is
Product is already in the database; no action necessary or link with UPC or other external product ID or with sample being indexed.
CASE description needs editing
Establish new food product ID for new description.
Highlight each line (statement) for editing.
Ask whether the statement should be
accepted as is,
deleted, or
edited
Before moving from one relationship type to the next, ask the indexer whether new statements with the present relationship type need to be added. If so present the statement pattern in a new line for the indexer to fill in.
When the last line is processed, incorporate the new product description into the database.

Notes on the procedure enter-new-product

This is where the real work takes place. The individual steps need explanation.

To assemble the description of a food product, the program starts with the statements given at the level of the food product itself and then follows successive levels of is-a relationships to pick up the statements given for higher-level food products. All statements must then be sorted in order of relationship type. A statement for a specific food product may override a statement for a more generic food product. For example, when a particular brand of whole milk is indexed, the system finds a statement for fat content - 3.35. With the broader product milk there is also a statement on fat content, giving a list of the values permissible for the various types of milk. That broader statement is not displayed.

The screen should show the hierarchy of food products from which the description of the food at hand was derived in the manner just described. For each statement displayed the food product (or the level in the hierarchy) should be given.

Statements for a food product, especially if it is a product type, may contain a range or a list of possible values. (See the example for the fat content of milk in the paper on the thesaurus.) These statements serve to remind the indexer of the aspects to be considered in indexing that type of food.

The indexer needs access to the thesaurus and to descriptions of similar foods. The program must provide for such access through pop-up windows.

When an ingredient of the product being indexed is itself a food product, the whole process starting with main must be carried out for the ingredient. The program must open a window for this purpose. Even if the ingredient is already described in the database, one must make sure that the description applies to the ingredient of the product being indexed. For example, the chocolate normally used in chocolate chip cookies contains vanillin. The chocolate used in very high quality chocolate chip cookies may contain vanilla.

When the description is completed, it must be integrated into the database. Due to the interlinked structure of the database this is not trivial. Statements in the new product description that are covered in the description of a broader product - all the way up on an is-a chain - need not be stored explicitly to the new product. When appropriate a new broad product should be introduced to economize on storage space. The description of a broader food product may need to be modified; for example, the amount of salt used (or the rank of salt on the label) in a brand of tomato soup being entered may fall outside the range for generic tomato soup recorded so far. The range in the description of generic tomato soup must be changed.

The program should allow for entering amounts of ingredients or constituents in any kind of measurement units and transform them to the standards used in the system. This may require input of package weight or portion weight. The program can deal with the solids content by referring to the mount of solids contained in each ingredient as given in the database.

Retrieval

In the indexing procedure described it is assumed that the database is accessed simply by UPC or product name. Even for indexing it may be desirable to access the database by any combination of descriptive characteristics of foods, and for using the database this will be the predominant mode.

The first step in a retrieval transaction is the display of a **frame**, that is a series of statements, one for each common relationship type, with blank values. The statements are arranged by relationship type in the sequence given in the conceptual schema.

The user then enters those values he wants to specify for the search; ranges or lists of possible values are allowed. In other words, the user describes an ad-hoc broad food product description. In this process the user needs access to the thesaurus. A user considers each relationship in turn, beginning with the first argument and moving to the second, third, etc. argument. As the user considers an argument, the top level of the hierarchy for the corresponding entity type or other pertinent information, such as the unit of measurement for a substance just selected, should be shown in a special window. The user can either enter a value or list of values (to be ORed) or pick one or more entries from the thesaurus display for entering in the query or for expansion to lower levels. If the user enters a term that is not an authorized descriptor, the system should display the authorized descriptor and ask for confirmation.

The frame displayed is dynamically updated in response to user input. In particular, if the user enters a value in the is-a relationship statement, the system retrieves the statements for the product named and replaces blank values on the screen with values found in this description.

Any food whose description is narrower is retrieved. The user can ask for display of any kind of information about them. If the retrieval is done for the purpose of indexing, the indexer selects one of the products for use in the procedure enter-new-product.

Due to the interlinked structure of the database the retrieval process is somewhat complex. The statements that must be considered in order to determine the relevance of a food product are not assembled in one place for that food product but are found with broader food products connected through an is-a chain as well. Furthermore, statements made for ingredients (and their broader products) may also be important for determining relevance. The retrieval algorithm must consider this complexity.

The language used for developing the prototype might support such a complex retrieval process so that the programmers would not need to worry about it. Otherwise the system must be programmed to do the search. The next paragraph give a possible algorithm.

Pick the query descriptor that retrieves the smallest number of food products; do an index search with this descriptor. The food products are found are candidates.

For each candidate do the following:

> Mark the other query descriptors that are satisfied.
>
> Go up along chains formed of is-a and/or has-ingredient relationships; for each food product encountered, mark additional query descriptors that are satisfied.
>
> As soon as all query descriptors are satisfied, the candidate and all its narrower products are marked relevant.
>
> If a candidate does not satisfy all query descriptors in this way, keep the list of the query descriptors that are satisfied and examine the candidates narrower products, following each chain formed of is-a and/or has-ingredient relationships. Mark additional query descriptors encountered. When all query descriptors are marked, the currently examined product and all its narrower products are relevant.

In a refinement of this procedure one must consider that not all food product characteristics are inherited along the has-ingredient relationship.

Construction of the prototype

Thesaurus

In the long run the thesaurus will be stored as an integral part of the database as indicated in the introduction. Thesaurus creation and maintenance functions must be part of the database program. But work on the restructuring of the FFV thesaurus should start immediately so that the core of a restructured thesaurus is ready for indexing as soon as a prototype database system is completed. It should be possible to write a Prolog program to support minimal thesaurus development functions as a warm-up exercise in preparation for the prototype development. Some data should be obtained from FDA in machine-readable form.

Prototype program

A prototype may be most easily implemented using a fast version of Prolog or another language allowing for ease of data representation and ease of modification running on a powerful microcomputer, such as 386 machine or a Sun work station. The programmers doing the work must be supported by a good consultant in systems design and programming for knowledge-based systems.

Manpower requirements

Since this system charts new territory, estimates for the effort needed, especially in the area of programming, are very difficult and must, by necessity, be quite tentative.

Developing the thesaurus requires a minimum of one person-year of a vocabulary specialist with proper clerical support.

It is estimated that the development of the prototype program will require at least one person-year of a programmer with consultant support.

The project also needs a technical director who can provide leadership in both areas. If that person is full time, he or she can also do some of the work.

In both areas work can be done in parallel so that development time can be shortened by putting two people to work in each.

Brief job descriptions are given on the following page.

1. **Vocabulary specialist**

Task

Based on the Factored Food Vocabulary, develop a structured list of entity values to be used in pilot database using the entity-relationship approach to food description.

Qualifications needed

Familiarity with the Factored Food Vocabulary

Ability to structure a set of terms

Ability to collect and present information about terms

Ability to learn the CFSAN thesaurus system and work with it

2. **Systems analyst/programmer**

Task

Based on system specifications, design and implement a system for organizing a database of food description data using the entity-relationship approach and for entering indexing data. Because the food descriptions are so interrelated, indexing a new food product entails searching the database for more general or similar food products.

Qualifications needed

Ability to translate system specifications into a more specific design that capitalizes on the capabilities of the language chosen

Ability to implement the design using a high-level language or DBMS, such as Prolog, ORACLE, or INGRES