Meta
Journal des traducteurs
Translators' Journal

META

A Concept-Oriented Approach to Terminology Work on PC

Cay-Holger Stoll

Volume 34, Number 3, septembre 1989

1. Actes du Colloque Les terminologies spécialisées : Approches quantitative et logico-sémantique et 2. Actes du Colloque Terminologie et Industries de la langue

URI: https://id.erudit.org/iderudit/004296ar
DOI: https://doi.org/10.7202/004296ar

See table of contents

Publisher(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (print)
1492-1421 (digital)

Explore this journal

Cite this article

Stoll, C.-H. (1989). A Concept-Oriented Approach to Terminology Work on PC. *Meta*, *34*(3), 615–628. https://doi.org/10.7202/004296ar

A CONCEPT-ORIENTED APPROACH TO TERMINOLOGY WORK ON PC

CAY-HOLGER STOLL
EUROLUX Computers, Gonderange, Luxembourg

1. INTRODUCTION

Terminology work (at least basic management of terms) is an indispensable task for any professional translator or translation service. In the past, terminology work had to be carried out on cardfiles or other paper media. Maintenance was time-consuming and difficult, access to information cumbersome.

The introduction of computer databases of terms, first on mainframes, later on mini computers and PC's has been a big step towards flexibility and ease of maintenance. However, mainframe databases generally suffer from a number of problems, such as:

- rigid, inflexible entry structure
- access usually read-only for the non-privileged user
- poor response times
- high telecommunication and retrieval cost

On medium and small size computers, general purpose database systems are also available. The are usually relational and serve as a shell to create customized applications tailored to a given task. Although smaller computers may overcome the read-only, response time, and telecommunication problems of mainframe databases, there are problems associated with relational database software. Even though relational database systems are very powerful when manipulating well-structured data, when they are applied to terminological and other lexical data, they have several problems:

- fixed-length record structure
- pre-defined fields with pre-defined field-lengths
- inefficient use of space for variable length data

Recently, PC-based tools for translators have appeared on the market. Unfortunately, most of these systems also use a fixed record and field structure and hence are not ideal for serious terminology work.

The system presented in this article avoids the pitfialls of relational databases and database systems. We will consider some of the basic properties of the system and develop a model for a concept-oriented terminology approach on PC.

2. THE MERCURY / TERMEX (TM) DATABASE SYSTEM

The database model described below has been developed under the control of the Mercury / Termex (tm) terminological database manager.

The system runs on IBM PCs and compatibles with a minimum requirement of 384 KB main memory. A hard disk is recommended but not required for the operation of the system.

The system may be used in two different modes: RAM resident and stand-alone.

The RAM resident mode is for use of the system in combination with a word processor. The main purpose of this mode is to acess a database during translation of a document, look up a term and paste its translation into the text or to enter single entries into an existing database file.

Access to the system in RAM resident mode is done through a pop-up window from other commercial software packages.

The RAM resident operation requires the system to be loaded at the beginning of a PC session by typing TERMEX from the DOS prompt.

The stand-alone operation is meant for systematic terminological work independent from the work processor. In this mode the program acts as a closed unit and uses the full-screen for its operation.

On the upper part of the screen, the Termex (tm) main menu will be displayed. On the lower part of the screen the user enters terminological information into his database.

3. STRUCTURE OF A MERCURY / TERMEX (TM) DATABASE

Physical organization of a database

The information accessed through Termex (tm) (stand-alone or RAM-resident operation) is stored in **files** on either hard disk or diskettes.

Files used by the system are encoded in an internal format and organized as a **database.**

Generally speaking, a database is a structured collection of **data.**

The **data** is organized in **physical records** (each terminated internally by a carriage return and line feed control sequence). Each record is sub-divided into **fields** corresponding to different types of information (like definitions, synonyms, date, responsible, etc.).

Each field has a **field label** defining the type of information stored in this field.

Logical organization of a database

The logical organization of a database is based on the physical structure and is user defined. The user decides how information or data shall be divided on records, what types of information he wants to integrate into his database and how he wants to label this information.

The Termex (tm) database system is tailored to the requirements of terminological databases.

Among others, it has the following special properties:

- the **record length** may vary between records contained in one and the same database;
- records may contain **different types** of information;
- the **space** occupied by one field **may vary** as well from record to record;
- **information** belonging to a terminological entry **may be split** on several records.

File Formats

For different purposes and tasks the Termex (tm) database system works with and supports different file formats. The internal format is used to work in dialogue with an existing database. The external format serves for printing, merging and exchanging terminological information.

Internal Format

The internal format differentiates between two file types:

- Foreground (or Access) Files
- Background Files

Foreground Files

Foreground Files are working glossaries for individual purposes or terminological databases containing individual information which may still be subject to modification.

The information contained in a foreground file typically is made permanent after verification by the terminological advisory board or the database administrator.

Background Files

Background Files contain the common terminological resources of all translators working in a group together. Data may be shared within a network, but not modified (read-only access).

Updating of terminological information contained in a Background file is carried out by the database administrator.

A Termex (tm) database may have a very simple structure and be used exclusively as working glossary for translation purposes. But it is also possible to develop complex terminology applications based on the system. The system's flexible structure offers a wide variety of user-defined customized approaches for terminological work.

Templates

Input may be done either arbitrarily, by by-passing all structural considerations, or following a pre-defined structure. Serious terminology work, however, will always require a good deal of conceptual considerations before terminological information may be entered into a database. Systematic terminological work is done based on templates.

A template is part of a Termex (tm) database and has in principal the same status as any other record contained in the database. As a matter of fact, a template is stored as physical record in the database file.

From the user's point-of-view, a template is an empty shell to be filled in systematically with terminological information.

A user may switch at any time between different templates. A complex terminological entry will typically consist of several templates to be filled in consecutively.

Cross-References

The terminological structure under discusion in the following section makes use of a specific feature of the Termex (tm) software, which is particularly useful for any kind of dictionary work. It is referred to as chaining operation and relates entries in **a database.**

The function of the chaining feature corresponds exactly to what is known in paper dictionaries as a cross-reference. A cross-reference between two dictionary entries consists of a field-label in the first entry pointing to a second entry in the same dictionary.

The cross-reference function is invoked by a function key and the input of the corresponding field name to be used for the cross-reference. For the chain to be successful, the field information governed by the field name in the first entry has to exist as an entry of its own in the same dictionary. If the second entry does not exist, the chaining operation will fail.

Ex.: 1. Entry:
***Buch**
{1} book **{*} Taschenbuch**

2. Entry
***Taschenbuch**
{1} paperback **{*} Buch**

Expl.: The term 'Buch' has in its translation field a field name '*****' pointing to 'Taschenbuch'. If (and only if) the second entry (Term: Taschenbuch) exists, the user may chain from 'Buch' to 'Taschenbuch' by use of the key 'F3' and the field name '*'.

On the second entry, a chain or cross-reference exists, pointing to 'Buch'. In this case, the user may chain as well from 'Taschenbuch' to 'Buch'.

NOTE: Any field name may serve as operand for a chaining operation, as long as the contents of this field point to another entry. The field name '*' is the default chaining operand. If you use the '*' for chaining be sure that the 'Automatic Chaining' in the Option Sub-Menu is set to 'n'. Otherwise, you may not obtain the desired results. See Supplement to Mercury/Termex (tm) 1.33, section 5.2, page 57.

The chaining technique will play an important role in connection with the different types of entries discussed below.

4. A CONCEPT-ORIENTED TERMINOLOGICAL DATABASE

4.1 CONCEPTUAL CONSIDERATIONS

The approach to terminological work presented on the following pages is the result of a cooperation between a number of Danish banks and the author.

From the conceptual point of view, the approach consists of three main structural elements:

- Bilingual or translation-oriented record
- Conceptual or language-independant record (b-Template)
- Linguistic or monolingual record (d-template)

The different entities contain the following information and are interconnected:

Bilingual Record

consisting of word equations, where one source or key-word is mapped to on one or several target words

cross-references to the monolingual record for zsource word and target word(s)

Conceptual Record
(b-Template)

consisting of operational and concept-oriented information
cross-references to the monolingual record/records

Monolingual Record
(d-Template)

consisting of language-specific information pertaining to one word of a given language

cross-references to the bilingual as well as the conceptual record

---

The graphic on the following page depicts the relation between the different record types.

'C-Record' stands for Conceptual record.
'Ml-...Mn-Record' stands for Monolingual records (Synonyms) directly related with one Conceptual record.

'Bl-...Bn-Record' stands for Bilingual records, each directly related with one singular 'M1-...Mn-Record'.

'C-Record' and 'M1-...Mn-Record' form together a sub-set of what will be defined below as 'Terminological Entry'. 'B-Records' are independent entities mainly used for translation purposes.

The two-directional arrows on the graphic represent cross-references (or chaining operations) established on the different input templates.

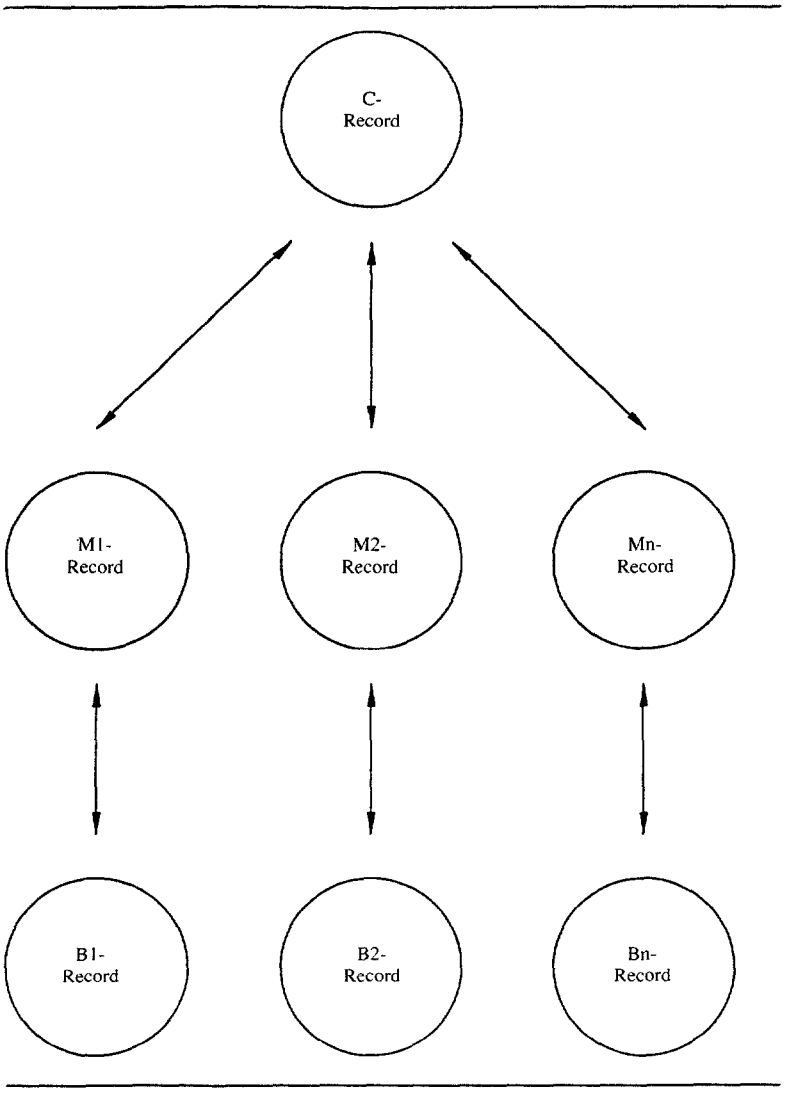

TABLEAU — TEXTE DE STOLL

4.2 DEFINITIONS

Before we start the actual description of the templates used with the database, we will define some of the basic concepts used for the current approach.

We have referred above to the different kinds of entities contained in our database as records and we have defined a record as an identifiable physical part of a database in general.

In some cases a physical record will be identical with the corresponding logical entity called 'Entry'. In other cases, an entry might consist of two or more physical records.

Our further work will be mainly concerned with the logical, i.e. terminological structure of the data. To avoid confusion, we will define the main concepts used in the sections below as follows:

Entry

> Total of information pertaining to one item in a list of items; in our context we will use Entry to cover both the Terminological and the Bilingual Entry defined below.

Template

> Pre-defined and structured form to be used for input of terminological or translation-relevant information; in our approach we use three different types of Templates: c-Templates, m-Templates and a Template for the Bilingual Entry. Each type of template has as many derivatives as languages used in the database, i.e., the c-Template is subdivided into Danish, German, English, French and Spanish c-Templates.

Terminological Entry

> Structured, language-independent compilation of information pertaining to a concept; in our approach, a Terminological Entry consists of two structural components (m-Template and c-Template) and of four or more physical records; the full Terminological Entry representing all information available for one concept may be identified through the serial number (Field = num) assigned to its parts.

Terminological Unit

> Keyword or keyphrase or idiomatic formulation referring to one concept; the Terminological Unit is usually used to look up information in the database. We will refer to the Terminological Unit as **'Term'** in the following sections; a Terminological Unit or Term may appear as part of a c-Template, m-Template or Bilingual Entry.

c-Template

> Part of a Terminological Entry, formulated in one of the languages used in the database; it contains information common to all existing derivations within one language; all c-Templates belonging to one concept represent the concept-oriented part of one Terminological Entry.

m-Template

> Language-dependent part of a c-Template containing language specific information for one single derivative of a concept in a given language

NOTE:

In the graphic representation above, one c-Template (referred to as C-Record) points to one ore more m-Template (referred to as M-Record), but each m-Template has only one corresponding c-Template.

Bilingual Entry

Represents a word equation between one Term in a given source language and one or more translations of this Term into a given target language; strictly speaking the Bilingual Entry is not part of the Terminological Entry, although it contains as one of its parts the number of a corresponding Terminological Entry; the Bilingual Entry functions as translation-oriented part of the database. Each Bilingual Entry is connected with one corresponding c-Template in the source language and one or more corresponding c-Templates in the target language;

NOTE:

The link between Bilingual Entry and m-Template (M- and B-Entry in the graphic above) is only reversible for the relation between Term in the source language and Term functioning as entry point to the m-Template; the link between translations of a given Term and their corresponding m-Templates has not been taken into account.

Cross-reference

Pointer(s) from one Entry to one or more Entries within the same database; from the theoretical point of view, a cross-reference is not an integral part of either a Terminological or a Bilingual Entry. Cross-references are operational aids to connect related entities of information.

Datafield

Labelled part of an Entry, reserved for information pertaining to one class or category of information within one language; one Entry is typically sub-divided into several Datafields; in the sections below, we will use the abbreviated form 'Field' when referring to Datafields.

Fieldname

Each Field is preceded by an identification label (in curly braces) identifying the class or category of information contained in the Field.

Summary

Our terminological database consists of **Entries**. Each entry is sub-structured into classes or categories of information called **Fields**. Each Field is preceded by a corresponding **Fieldname**.

We differentiate between **Terminological Entries** and **Bilingual Entries**.

A terminological Entry is represented by all records containing one and the same serial number within one database. It consists typically of two **c-Templates** and two or more (in case of synonyms) **m-Templates**. From this fact follows immediately that our current **approach is bilingual** as far as one database file is concerned.

NOTE:

The Bilingual structure chosen in our present context is not a theoretical limitation imposed by the system's architecture. The user is free to choose as many languages as he needs for his work.

The Bilingual Entries carrying the same serial number as the corresponding parts of a Terminological Entry are not an integral part of this Terminological Entry.

A Bilingual Entry consists of one **Term and** its **translations** into another language.

Entries are connected by **Cross-references**, which do not form an integral part of an Entry.

The unifying or combining element to identify a Terminological Entry as a whole is its **serial number**. The serial number field is part of each record in the database.

4.3 TEMPLATES

Based on the above definitions we may now turn to the description of the Templates and Fields used in the database. As far as the Fields are concerned, we will give only a summary overview.

4.3.1 Fieldnames used in Templates

The fieldnames are listed as they appear in the actual database records (enclosed with curly braces). The language reference (like 'da' for 'Danish' or 'en' for 'English') has been replaced by the abbreviation '<In>; if more than one language is concerned '<In1>' is used in addition.

Bilingual Entry

Term Level

.<In>: source

refers to a term in a given souce language

Datafield Level

{subject}

subject field information

{ser}

Serial number identifying the record as part of one specific terminological entry

{*}<In>:

Chain to monolingual concept or m-Template of term

{q}

Quality or reliability of translated term

{1}

First translation of term

{*1} <In1>d:

Chain to monolingual concept or m-Template of translated term.

For further translations, the quality field, term translation field and chain-field may be repeated by increasing the term identification number on the field name by increments of 1. ex. {2} and {*2} for the second translation, etc...

c-Template

Term Level

.<In>b

refers to the term in a given source language as part of the c-Template; in addition to the language identification, the letter 'b' serves as identification criteria of this term to be a member of the c-Template part of a terminological entry.

Datafield Level

{subject}

subject field information

{ser}

Serial number identifying the record as part of one specific terminological entry

{<In>inp}yy/mm/dd
Input date

{<In>upd}yy/mm/dd
Modification date

{<In>resp}bbb/iii
Responsible institute and person

<In>TERMS: {*}<in>d: {1*}<In>d:
Cross-reference to synonyms

{<In>com}
Comment on terminological entry

{<In>def}
Definition

m-Template

Term Level

.<In>d

referes to the term in a given source language as part of the m-Template; in addition to the language identification, the letter 'd' serves as an identification criteria of this term to be a member of the m-Template part of a terminological entry.

Datafield Level

{subject}

subject field information

{ser}

Serial number identifying the record as part of one specific terminological entry

{<In>reeg}

Information about region

{*}

Chain to bilingual entry

{<In>gram}

Grammatical note

{*<In>b}<In>b:

Chain to C-Template of concept

{<In>alt}

Alternative forms (e.g., abbrev.).

{<In>ref}

Reference

{<In>cont}

Contextual information

{<In>note}

Note about use

4.3.2. Template Layout

Bilingual Template

*. <In> source
{subject} . {ser} . {*} <In>d:
{q} .{1}.
{*1} <1n1d:

c-Template

*.<In>b
{subject} . {ser} . {<In> inp} . {<In> upd} .
{<In> resp} bbb/iii
<LN> TERMS: {*} <1n>d: {*1} <1n>d:
{<1n> com} .
{<1n> def}.
{<1n exp}.

m-Template

*. <In>d
{subject}. {ser} . {<1n> reg} . {*} <1n>:
{<1n gram} . {*<1n>b} <1n>b:
{<1n alt .
{<1n ref} .
{<1n cont} .
{<1n note} .

4.3.3 Cross-references

A total of six cross-reference fields are foreseen in the above templates. For a cross-reference to work it is absolutely required to add the corresponding target entry into the database.

If, however, the target of a cross-reference is missing, the chaining operation will fail and the user may establish the cross-reference subsequently. As a consequence, the creation of new entries does not require all templates to be filled in immediately.

It is also possible to convert a foreground file to a write-protected background file, even if most or all entries are still incomplete. Missing information may be integrated later on through a merge run (see below).

If an entry and its corresponding cross-reference entry are created, the cross-reference field should be filled in correctly and with care. A spelling error in a cross-reference field causes the chaining operation to fail.

Bilingual Entry

{*}<In>d:

Cross reference to the corresponding m-Template of the term. Since the default fieldname for chaining operations is used, the user may omit the entry of the fieldname to execute the chain.

{*1}<In1>d:

Cross-reference to the corresponding m-Template of the term translation. Input of fieldname required.

c-Template

{*}<In>d:

Cross-reference to corresponding m-Template from the first linguistic representation of a concept in a given language

NOTE:

More cross-reference fields ({*2)} <In>d: ,...{*n} <In>d:) may be added if required.

m-Template

{*} <In>:

Cross-reference from m-Template to corresponding term on bilingual entry. Input of fieldname not required.

{*<In>b} <In>b:

Cross-reference from m-Template to corresponding c-Template

5. INPUT CONVENTIONS

The term part of a template does not form an integral part of the template itself. Accordingly, input rules differ for the term and the datafield part of a database record.

Term Level

Conventions:

- a term on a bilingual entry is preceded by the corresponding language identifier, followed by a colon.

Ex.: en: chair

*de: Kreditinstitut

the correct language identifier and the colon have to be typed in by the user.

- a term on a **c-Template** is preceded by the corresponding language identifier, followed by the letter 'c' and a colon.

  Ex.: ***enb: chair**

  ***deb: Kreditinstitut**

  a term on a **c-Template** is preceded by the corresponding identifier, followed by the letter 'm' and a colon.

  Ex.: *end: chair

  *deb: Kreditinstitut

Homonyms

◆ Homonyms require an additional identification within one language on the term level. The identification might be a number in square brackets, following the term for both occurrences. The identifier has to appear on the term parts of all corresponding templates, as well as in **cross-references**.

Ex.: *de: Bank [1]
*deb: Bank [1]
*ded: Bank [1]

*de: Bank [2]
*deb: Bank [2]
*ded: Bank [2]

Datafield Level

The datafield part of the Termex (tm) window is divided into field labels and corresponding fields. After the term has been entered and the user has confirmed his input with ENTER, the cursor will move into the datafield part. The template may now be filled in.

Synonyms

By default, the c-Template provides space for the cross-reference to m-Template of the term plus one synonym ('{*1} dad:'). If more synonyms are to be entered, a new line and additional field may be added.

Additional Translations

On the Bilingual entry, fields are available for one translation to the term. To add more translations, the last two lines of the entry may be repeated.

6. ORGANIZATIONAL CONSIDERATIONS

To set up and manage a terminological database in a group requires a high level of cooperation and teamwork. In the Termex (tm) system, individual transactions are carried out in a user-specific foreground database. The integration of user-specific terminology into the common background database is achieved in intervals through a merge procedure.

Two major problems are:
- reduction of redundancy and duplicate work to a minimum,
- evaluation and integration of terminology into a common database.

The answer to both problems is a full-time database administrator taking care of all coordination, evaluation and integration tasks with respect to the common terminological material.

But even if several groups work more or less independently, of each other, certain steps may be taken to reduce the above mentioned problems. Under all circumstances, however, the actual integration of foreground terminology has to be carried out by one person in regular intervals.

From the operational point of view, we have to differentiate between the following cases:
- Set up of new entries
- Completion of existing entries
- Correction of existing entries.

Set-up of New Entries

Set-up and integration of new entries into the common database is fairly straightforward and requires two steps:
- Verification of new terminology by the database administrator
- Merge of new terminology with existing database.

As a rule, new entries should be as complete as possible; i.e. c-and m-Templates should exist as well as the corresponding bilingual entries.

If two entries have been drawn up for the same term by different users, the group or database administrator has to decide, which term shall be integrated into the common database. The system will only generate a message during merge that the term 'abc' exists twice.

Completion of existing entries

Since the creation of full terminological entries may not be possible due to time considerations or lack of information, some basic rules for updating are necessary.

In general, the update information to an existing entry should be drawn up by the person who entered the entry in the first place. Otherwise, the evaluation process will become more and more complicated and time-consuming.

Adding a new record to an existing entry (like an additional m-Template and corresponding bilingual entries) does not require any further intervention user on the database, apart from the merge operation.

Correction of Existing Entries

The correction of existing records as part of entries follows more or less the same procedure as above. Depending on the type of the correction operation, however, the evaluation process might be more difficult. The correction of a spelling error will definitely pose less of a problem than the reformulation of a definition.

7. CONCLUSION

Based on an adequate system architecture, a concept-oriented multilingual terminological database may be created, used, and maintained on a PC. Protection of validated data and use of the database in a group should be supported by the system as well.

A terminological database system must be based on flexible record and field structures. The individual requirements of terminological work in different environments and the variable size of terminological data do not match with "hard-wired" and inflexible systems.

The above approach represents only one of an indefinite number of possible structures. The Mercury/Termex (tm) system may serve as a simple translation aid and dictionary access tool (a great number of commercial dictionaries are already available with the system), and may also be used to build up complex lexical database structures.