

Principles of a system for terminological concept modelling

Bodil Nistrup Madsen, Hanne Erdman Thomsen, Carl Vikner

Department of Computational Linguistics, Copenhagen Business School
Bernhard Bangs Allé 17B, DK-2000 Frederiksberg, Denmark
{bnm, het, cv}.id@cbs.dk

Abstract

We are working on a project called CAOS - Computer-Aided Ontology Structuring - whose aim is to develop a computer system designed to enable semi-automatic construction of concept systems, or ontologies. The system is intended to be interactive and presupposes an end-user with a terminological background (terminologist or professional translator). CAOS supports terminological concept modelling. The backbone of this concept modelling is constituted by characteristics modelled by formal feature specifications, i.e. attribute-value pairs. Our use of feature specifications is subject to a number of principles and constraints. In this paper we want to demonstrate some of these principles and to show why they are necessary in order to permit the construction of an interactive tool for building terminological ontologies. We will also show how they contribute to determine the structuring of the ontologies in CAOS and to facilitate the work of the terminologist user.

The principles

In this paper, we will discuss the following principles¹:

1. unique empty top concept
2. polyhierarchical structure
3. inheritance of feature specifications
4. function from attributes to values
5. primary feature specifications reflected by dimension specifications
6. uniqueness of dimensions
7. uniqueness of primary feature specifications
8. grouping by subdividing dimensions
9. distinction of mother and daughter
10. distinction of sisters

The Principles 1, 3, 4, and 7 are taken over from works on formal feature structures because they reflect terminological principles (e.g. Copestake 1992: 23). The principles 2, 8, 9 and 10 are our attempt at formalizing other principles from traditional terminology work, and the principles 5 and 6 are developed within the CAOS-project. The approach to the modelling of characteristics used in CAOS was proposed in Thomsen 1998, 1999 and Madsen 1998.

All the principles reflect constraints on the structure of CAOS ontologies. The observance of most of these constraints is enforced by the CAOS system during the construction of an ontology. However, the observance of principle 8 - 10 will be checked only in a final normalization cycle.

1 Unique empty top concept

The topmost concept in a CAOS ontology is always an empty concept named *top*.

The reason for this is that the user may want to introduce one or more concepts that are not related to any of the other concepts in the ontology thus far. In ontologies with an empty top concept such isolated concepts can simply be attached as (temporary) daughters of *top* instead of

being left without any contact at all with the ontology under construction.

In the following, *top* is not present in the illustrations, as we are only showing fragments of ontologies.

2 Polyhierarchical structure

A concept in CAOS may have more than one nearest superordinate concept. The reason for this structural latitude in CAOS is that terminologists are often lead to recognize that a subordinate concept may have characteristics in common with more than one superordinate concept (Madsen 1999: 35-37).

3 Inheritance of feature specifications

The principle of inheritance says that a concept automatically inherits all the feature specifications of its superordinate concepts.

This principle models the principle of traditional terminology that 'the intension of the subordinate concept includes the intension of the superordinate concept' (ISO 704: 5.4.2.2; cf. also Madsen 1999: 21). For an example see Fig. 1.

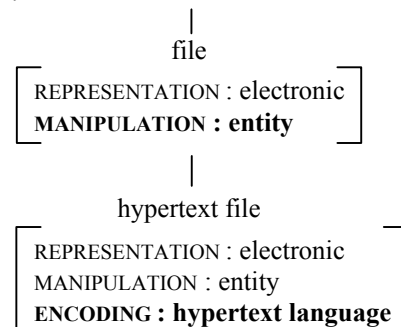


Figure 1: Inheritance of features

The principle of inheritance means that we have to distinguish two kinds of feature specifications: primary and inherited. A primary feature specification is assigned directly to a given concept, whereas an inherited feature specification is inherited from the concept's superordinate concepts. In the diagram in Fig. 1 we have written primary feature specifications in bold types. Thus, the concept *file* has one primary feature specification, [MANIPULATION :

¹ As yet the CAOS system has only been implemented partially, and some of the functionalities discussed in the following are for the time being to be considered as mere declarations of intent.

entity], whereas [REPRESENTATION : electronic] is inherited to *file* from a superordinate concept not shown in the diagram.

The distinction between primary and inherited feature specifications is reflected in the behaviour of feature specifications, for instance, as regards user intervention. Under certain conditions, the user may change, move or delete a primary feature specification, but never an inherited one.

Automatic inheritance of feature specifications means a lot of economy in the ontology building because the constructor is freed of specifying great amounts of redundant information. At the same time it contributes to assure consistency in the ontological structure: because inherited features are shown explicitly on all lower concepts, errors in the classification of concepts will be detected more easily.

4 Function from attributes to values

A concept is characterized by means of a set of feature specifications, called a feature structure. A feature structure is formally defined as a (partial) function from the set of attributes to the set of values (cf., for instance, Carpenter 1992). This means that an attribute may only be associated with one value in a feature structure.

This principle helps discovering errors and avoiding inconsistency in the structuring of an ontology. Imagine a user having read that an open-string stair is a stair where the strings leave one end of the treads visible. The user then places the concept *open-string stair* as subordinate to *stair* with the feature specification [STRING : reveals one end of treads]. Later the user finds documentation that shows that both tread ends may be visible in an open-string stair. Having forgotten about the first description she now wants to enter a new feature specification [STRING : reveals both ends of treads]. This would result in the structure in Fig. 2.

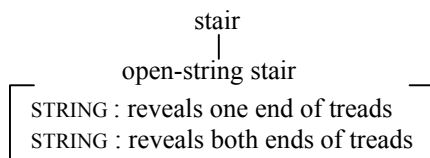


Figure 2: *Open-string stair* (illicit representation)

But this is not allowed by CAOS, and the user will have to find another solution, for instance, by changing the formulation of the value such that it covers both cases: one or both ends left visible.

5 Primary feature specifications reflected by dimension specifications

We have introduced a principle to the effect that a concept must register attributes and values of all the primary feature specifications of its daughters. This is done by creating dimension specifications on the mother concept consisting of a dimension and a list of values:

(DIMENSION : [value1, value2, ...])

For every primary feature specification on a concept, its attribute must figure as the dimension in a dimension specification on the mother concept, and its value must figure in the value list of that dimension specification.

An example is shown in Fig. 3 (adapted from Byrval: 2001).

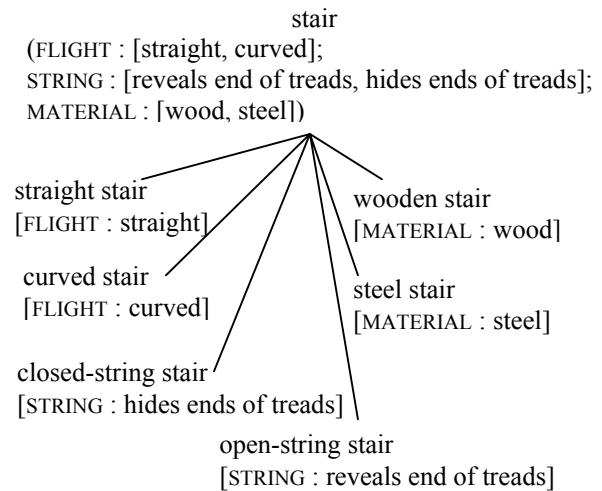


Figure 3: Dimension specifications

There is thus a close connection between primary feature specifications and dimension specifications. It is not possible to have a primary feature specification on a concept without its being reflected in a dimension specification on the mother concept, and conversely, for every value on the value list in a dimension specification there must be a corresponding primary feature specification on one of the daughter concepts.

This principle may perhaps seem to create a sort of unnecessary double dealing. We have introduced it firstly to be able to handle subdividing dimensions (principle 8) and secondly to be able to establish the following principle 6, Uniqueness of dimensions, and to have an easy way to implement it.

6 Uniqueness of dimensions

This principle says that a given dimension may only occur on one concept in an ontology. This means that primary feature specifications with the same attribute must always occur on sister concepts.

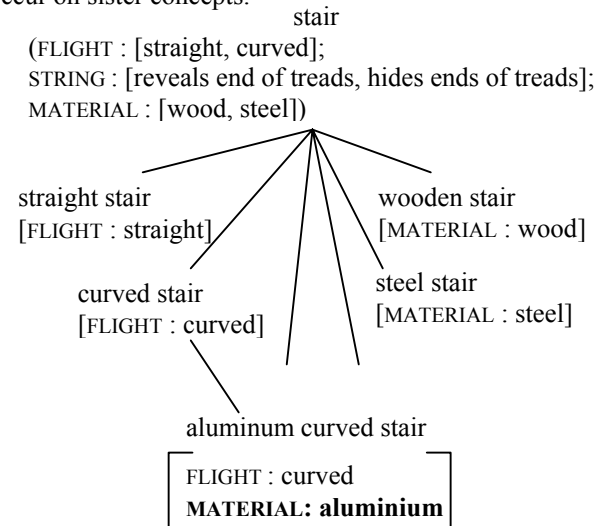
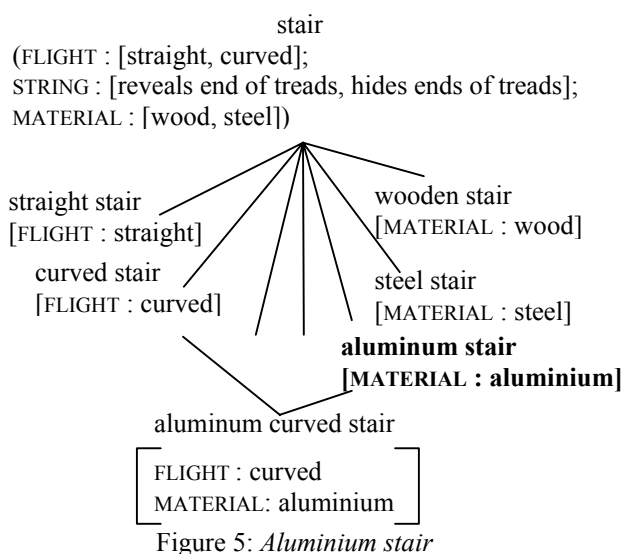


Figure 4: *Aluminum curved stair*

Uniqueness of dimensions contributes to create coherence and simplicity in the ontological structure, because concepts that are characterized by means of a certain common dimension must appear as descendants of the same superordinate concept.

To see how the principle of Uniqueness of dimensions contributes to improve the logical coherence of an ontology, consider a situation where a user wants to introduce a concept *aluminium curved stair* to the *stair*-ontology. The simplest solution might seem to be to add this new concept as a daughter of the concept *curved stair* and to ascribe to it the feature specification [MATERIAL : aluminium], as shown in Fig. 4.

This solution is not allowed by CAOS. The user will be told that MATERIAL is a dimension already found on the concept *stair*, and therefore the attribute MATERIAL can only be part of primary feature specifications on daughters of *stair*, not on granddaughters. The user can then choose to introduce a new concept *aluminium stair* as a daughter of *stair*, so that *aluminium curved stair* - in a poly-hierarchical structure - can inherit the feature specification [MATERIAL : aluminium] from this new concept. This will result in a structure like the one in Fig. 5.



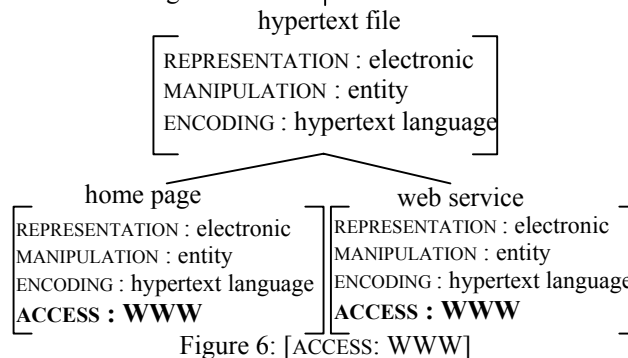
7 Uniqueness of primary feature specifications

A feature specification may only occur once in an ontology as primary. Uniqueness of dimensions (principle 6) means that a given primary feature specification can only appear on concepts that are daughters of the concept containing the relevant dimension. Uniqueness of primary feature specifications means that a given primary feature specification can only appear on one of the daughters.

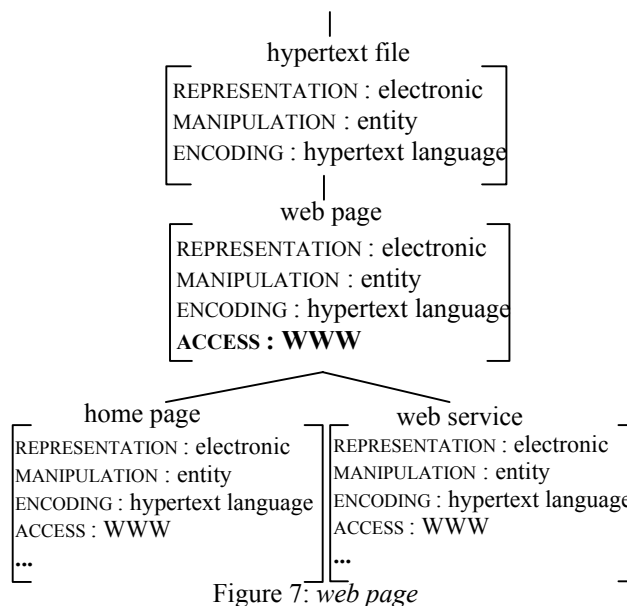
Like the previous principle (uniqueness of dimensions) this principle contributes to create coherence and simplicity in the ontological structure because closely related concepts, i.e. concepts with common characteristics, are kept closely together in the ontology in that they must be subconcepts of one common superordinate concept.

In conjunction the two uniqueness principles make it possible to a certain extent to carry out automatic placing of concepts into an ontology. If a new concept is characterized by one or more feature specifications, the system can be instructed to search the ontology for concepts with the attributes as dimensions and possibly concepts having more or less the same feature specifications, and on this basis propose a location for the new concept.

It is conceivable that a user will want to introduce the same primary feature specification on two sister concepts, as shown in Fig 6.



However, this is ruled out by principle 7, and this forces the user to reconsider the origin of the feature specification [ACCESS : WWW]. There are several possibilities. The most natural is to contemplate whether this feature specification could be placed on the mother concept, *hypertext file*, or one of its superordinate concepts. In this case, however, this solution is not compatible with the conceptual content of these concepts. The terminologist must therefore introduce a new concept such that the new concept contains the feature specification [ACCESS : WWW], and *home page* and *web service* inherit it from their new mother.



Such a concept can indeed be found in this case, because the concept *web page* may be introduced as an intermediate concept, as shown in Fig. 7.

8 Choice of subdividing dimensions

One or more of the dimensions of a concept must be chosen as subdividing dimensions. Subdividing dimensions must be chosen in such a way that each daughter concept has one and only one feature specification containing as an attribute a subdividing dimension of the mother concept. That is, there can be no overlapping subdividing dimensions.

A primary feature specification containing a subdividing dimension as an attribute is called a *delimiting feature*

specification. Thus, a concept must contain one and only one delimiting feature specification.

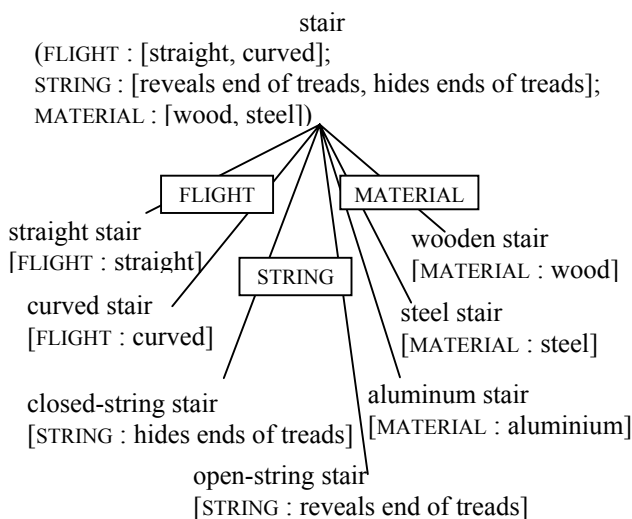


Figure 8: subdividing dimensions

In graphic illustrations we represent subdividing dimensions by means of boxes covering the relevant branches between a mother and its daughters as in Fig. 8. As can be seen from the illustration, subdividing dimensions group sister concepts according to the attributes contained in their delimiting feature specifications. They will often prove helpful to the user because they help significantly to give a clearer overview of the field.

Just as feature specifications are the formal modelling of the terminologist's characteristics, subdividing dimensions serve to model the terminologist's subdivision criteria (cf. ISO 704: 5.4.2.1). It is important to carefully choose subdividing dimensions that reflect essential characteristics because they are endowed with a definitional value. In terminology the corresponding delimiting characteristic should be used in the *differentia specifica* of the definition of the concept in question. That is, the definition should conform to the following schema:

(DaughterConcept is) a MotherConcept
that has TheCharacteristicCorrespondingTo[f:v]

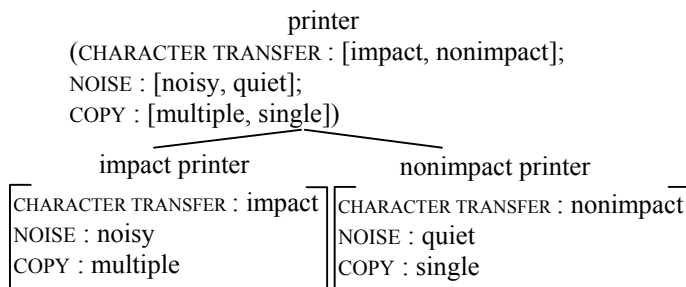


Figure 9: Three dimensions

Consider the fragment of an ontology for printers in Fig. 9. Here the attribute CHARACTER TRANSFER means 'way of transferring characters from printer to paper', and the value *impact* means 'striking the paper'. Formally all three dimensions of *printer* qualify for the role of subdividing dimension, but the terminologist will choose CHARACTER TRANSFER because she will realize

that this is the essential one in that the characteristics associated with this dimension determine the other characteristics: the characteristics of being noisy or quiet and the capacity for single or multiple copies are consequences of the kind of character transfer employed. This choice of subdividing dimension entails that *impact printer* will be associated with a definition like the following: *a printer that transfers characters from printer to paper by striking the paper*

9 Distinction of mother and daughter and 10 Distinction of sisters

A concept must be distinguished from each of its nearest superordinate concepts by at least one feature specification, and it must also be distinguished from each of its sister concepts by at least one feature specification. If there is no evidence that two concepts are distinct, they should be unified into one concept.

Conclusion

We hope to have shown that these principles help to permit the construction of an interactive tool for building terminological ontologies, on the one hand, and contribute to improve the simplicity and coherence of the resulting ontology.

References

- Byrval, Stina. (2001). Projekt opgave for Kuris Sprogservice ApS. Project report at the Department of Computational Linguistics, Copenhagen Business School.
- Copestake, Ann. (1992). The Representation of Lexical Semantic Information. University of Sussex at Brighton, Cognitive Science Research Papers, CSRP 280.
- ISO 704 (2000). Terminology work — Principles and methods. Genève, ISO.
- Madsen, Bodil Nistrup. (1998). Typed Feature Structures for Terminology Work - Part I. In: LSP - Identity and Interface - Research, Knowledge and Society. Proceedings of the 11th European Symposium on Language for Special Purposes. Copenhagen, August 1997, Copenhagen Business School: 339-348.
- Madsen, Bodil Nistrup. (1999). Terminologi 1. Principper og metoder. Copenhagen, Gads Forlag.
- Shelly, Gary B., Thomas J. Cashman, Gloria A. Waggoner, William C. Waggoner. (1993). Essential Computer Concepts. Danvers, Massachusetts, Boyd & Fraser.
- Thomsen, Hanne Erdman. (1998). Typed Feature Structures for Terminology Work - Part II. In: LSP - Identity and Interface - Research, Knowledge and Society. Proceedings of the 11th European Symposium on Language for Special Purposes. Copenhagen, August 1997, Copenhagen Business School, 349-359.
- Thomsen, Hanne Erdman. (1999). 'Typed Feature Specifications for establishing Terminological Equivalence Relations'. In: World Knowledge and Natural Language Analysis. Copenhagen Studies of Language, vol.23, Copenhagen: Samfundslitteratur: 39-55.