Terminological concept modelling and conceptual data modelling

Bodil Nistrup Madsen and Hanne Erdman Thomsen*

Department of International Language Studies and Computational Linguistics,
Copenhagen Business School,
Dalgas Have 15, DK-2000 Frederiksberg,
Copenhagen, Denmark
Email: bnm.isv@cbs.dk
Email: het.isv@cbs.dk
*Corresponding author

Abstract: Ontologies are useful for many purposes. The use of an ontology is, for example, crucial for writing consistent definitions of concepts within a specific domain. In this paper, we will argue that the principles of rigorous terminology work are useful for building consistent ontologies. In many cases, developers of IT systems encounter severe problems, because they neglect the necessity of developing a proper ontology (concept model) before they develop a conceptual data model as a basis for an IT system. In this paper, we will argue that the development of an ontology is crucial for setting up a conceptual data model, and therefore it should always be added as an initial stage to data modelling. Also we will give some examples of the mapping between ontologies and conceptual data models. Future research will reveal to what extent it will be possible to set up rules for automatic mapping of concepts of an ontology into classes and attributes of a conceptual data model.

Keywords: data modelling; concept model; terminology; UML; concept diagram; concept clarification; characteristics; properties; individual concepts; ontology construction.

Reference to this paper should be made as follows: Madsen, B.N. and Thomsen, H.E. (2009) 'Terminological concept modelling and conceptual data modelling', *Int. J. Metadata, Semantics and Ontologies*, Vol. 4, No. 4, pp.239–249.

Biographical notes: Bodil Nistrup Madsen is a Professor in Computational Linguistics at Copenhagen Business School. Her primary research falls within the areas: terminology management systems, data modelling, ontologies, knowledge structuring and knowledge representation. She is Director of the Centre for Terminology, the DANTERMcentre, which has developed the terminology and knowledge management system i-Term. She is working as a Consultant for public authorities in the development of ontologies as a basis for large IT systems and metadata taxonomies. She is chairman of SC 3, *Systems to manage terminology, knowledge and content* in ISO TC 37 *Terminology and other language resources*.

Hanne Erdman Thomsen is Associate Professor in Computational Linguistics with focus on terminology and terminology management at Copenhagen Business School, where she has been employed since 1992. Her primary research falls within formal semantics, ontologies and formalisation of terminology work, especially inheritance of characteristics. She currently teaches terminological theory and methods to students of language, language administration and computational linguistics. She is convenor of SC1/WG3 (*Principles and methods of terminology work*) in the ISO TC 37 *Terminology and other language resources*.

1 Introduction

In recent years, ontology construction has become an important part of software engineering as a basis for data modelling in order to ensure interoperability between systems and to facilitate reuse of components. There are many similarities between concept modelling in terminology work and the construction of formal ontologies, and for this reason terminologists are increasingly involved in IT development projects. The amount of research done in ontology construction in recent years shows a need for sound methods, and in this area it is our opinion that terminologists have something to offer.

Simultaneously, the ISO committees concerned with metadata registries and meta models (such as ISO/IEC JTC1/SC32 WG2) are becoming interested in the standards for terminology work developed in TC 37 *Terminology and other Language Resources*.

In both cases, the different goals of terminology work and data modelling give rise to discussions of basic concepts that underlie both fields.

The paper is organised as follows. In Section 2, we introduce some of the central characteristics of terminological concept diagrams that are a result of concept modelling in terminology work. Section 3 gives examples of our proposal for the use of UML (Unified Modelling Language) for concept diagrams. In this section we also point out some possible misunderstandings between terminologists and data modellers, due to differences in the use of UML for development of IT systems and for terminology work.

In Section 4, we compare terminological concept modelling and construction of formal ontologies. We argue that terminology work can be characterised as non-formalised ontology construction, and that the principles of rigorous terminology work are useful for concept clarification prior to data modelling.

Concept modelling and data modelling are often confused. In Section 5, we give an example of a terminological concept model that functions as the background for setting up a conceptual data model. We also point out the main differences between terminological concept modelling and conceptual data modelling.

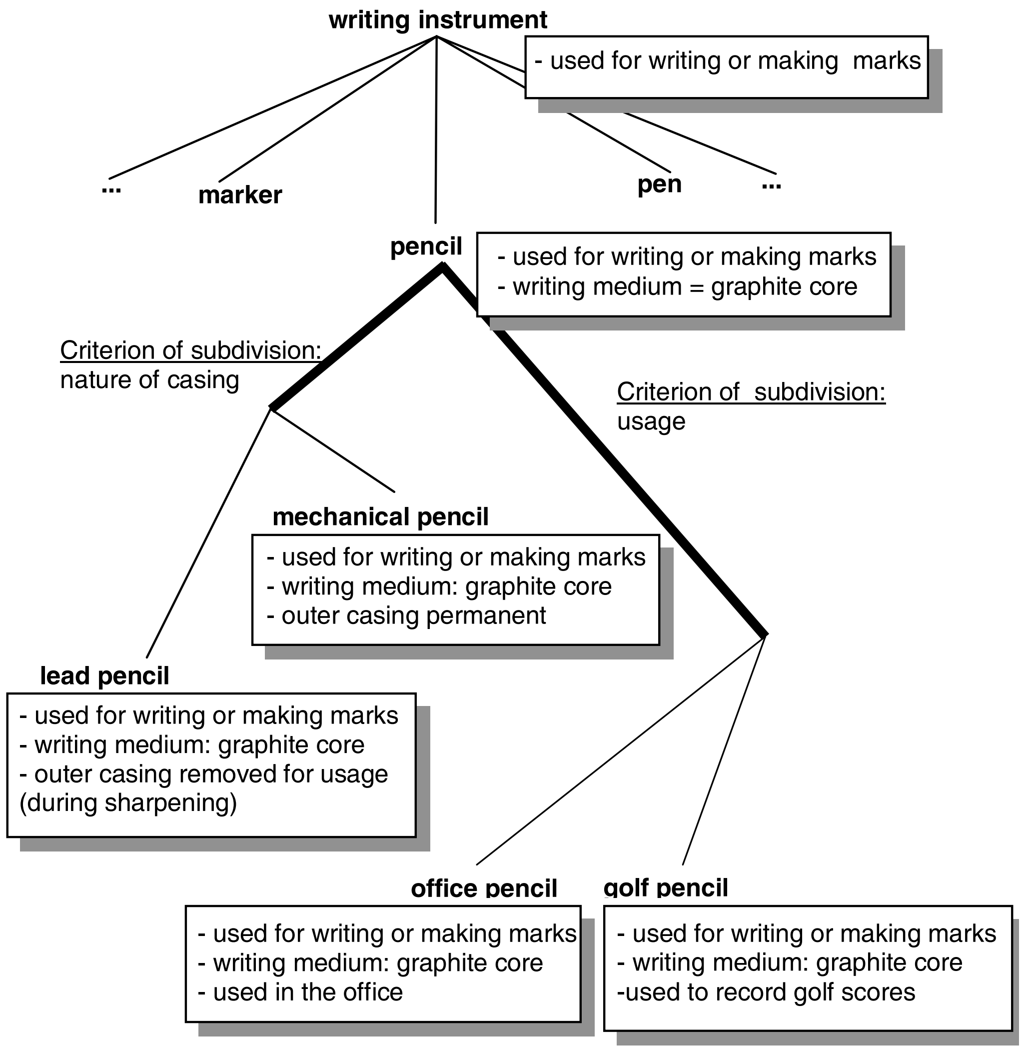
In Section 6, we give examples of the necessity of integrating concept modelling in the data modelling process, and of the benefits of adding terminology modelling in front of the conceptual data modelling.

Finally we conclude that terminological concept modelling is crucial for setting up a conceptual data model, and therefore it should always be added as an initial stage to data modelling.

2 Terminological concept diagrams

In Figure 1, we show an example of a terminological concept diagram. The example is adapted from Example 8 in ISO 704:2000, and shows different types of pencils. The nodes in the diagram represent concepts, the lines are relations (in this diagram only the type relation is used), and the boxes contain characteristics of some of the concepts. The bold lines indicate subdivision criteria, i.e. pencils can be divided with respect to the nature of the casing (either removable or permanent) or with respect to their usage (office or golf).

Figure 1 'Writing instruments', cf. ISO 704:2000



Characteristics of concepts and subdivision criteria are crucial in terminology work, where focus is on what differentiates concepts from each other.

At Copenhagen Business School we have proposed to represent characteristics as typed feature specifications (cf. Carpenter, 1992), see e.g. Madsen (1998) and Thomsen (1998). These feature specifications consist of a feature and a value, e.g. the tuple ‘WRITING MEDIUM: graphite core’ corresponds to the characteristic ‘writing medium = graphite core’ in Figure 1. Thus, a representation of a whole concept is a feature structure, i.e. a set of feature specifications corresponding to the unique set of characteristics, which constitutes that concept.

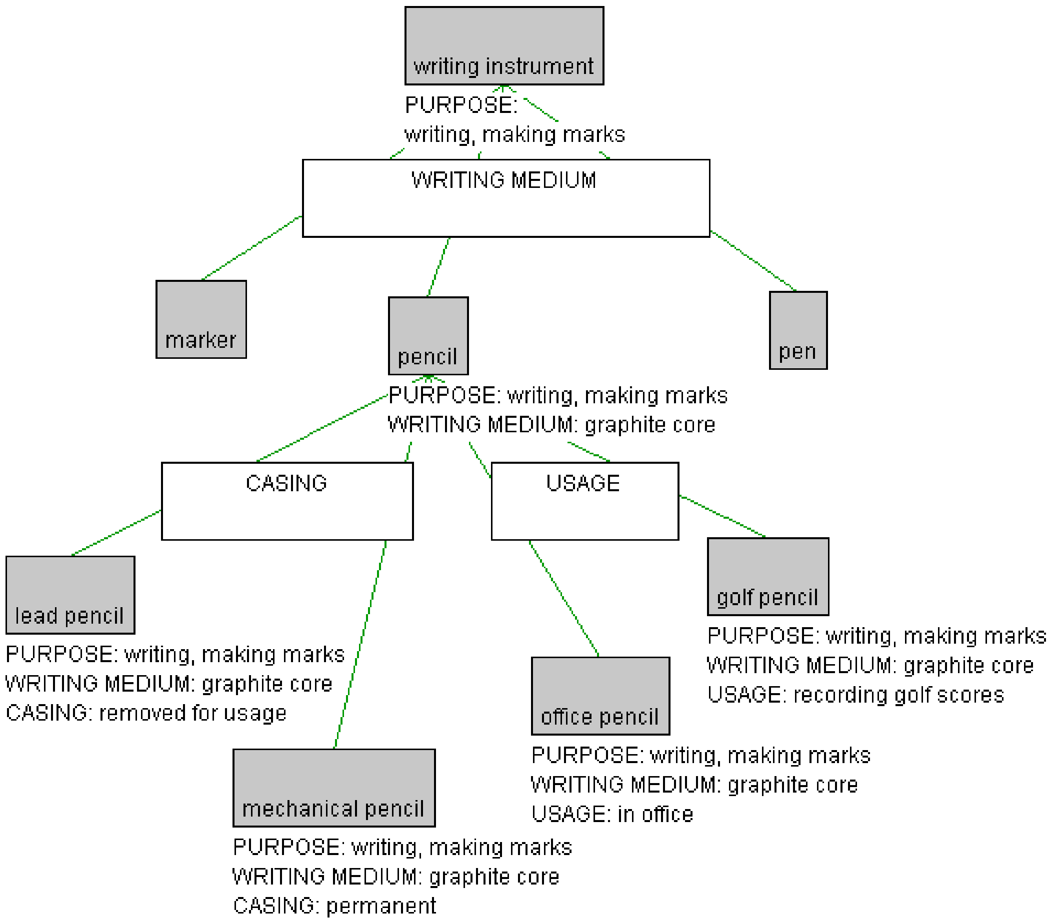
This is illustrated in Figure 2, which shows the same concept system as Figure 1 with all characteristics written as feature specifications and subdivision criteria in boxes across the relevant relations.

In Figure 2, all inherited characteristics are included. Typed feature theory accounts for the inheritance of characteristics, and therefore the formalism can be used to check concept systems for consistency. We exploit this feature in CAOS, our system for Computer-Assisted Ontology Structuring (see e.g. Madsen et al., 2005).

The representation of characteristics and subdivision criteria shown in Figure 2 has not been adopted by terminologists in general, but we include it here because it clarifies the relation between characteristics and subdivision criteria: For any given concept the feature of a non-inherited characteristic may be a subdivision criterion for the superordinate concept. This clarification makes it much easier to identify subdivision criteria and differentiating characteristics in practical terminology work, and in fact it has proven much easier for students to understand subdivision criteria when using this notation.

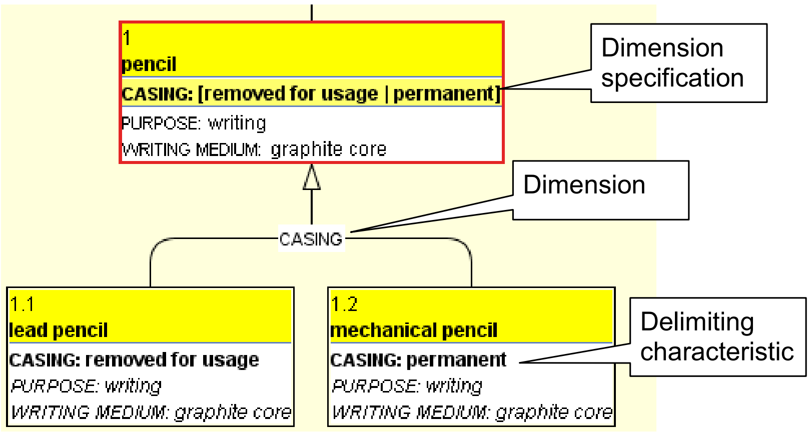
Typed feature theory does not account for subdivision criteria, therefore, in CAOS, we have added what we call dimensions and dimension specifications, cf. Madsen et al. (2005) and Figure 3, and formalised the relation between these and the corresponding characteristics of subordinate concepts.

Figure 2 ‘Writing instruments’ with feature specifications (see online version for colours)



Note: This figure is made with i-Model, the terminological concept modeling tool integrated with the terminology management system i-Term.

Figure 3 ‘Pencils’ in the CAOS system (see online version for colours)



3 UML for concept diagrams

3.1 UML representation

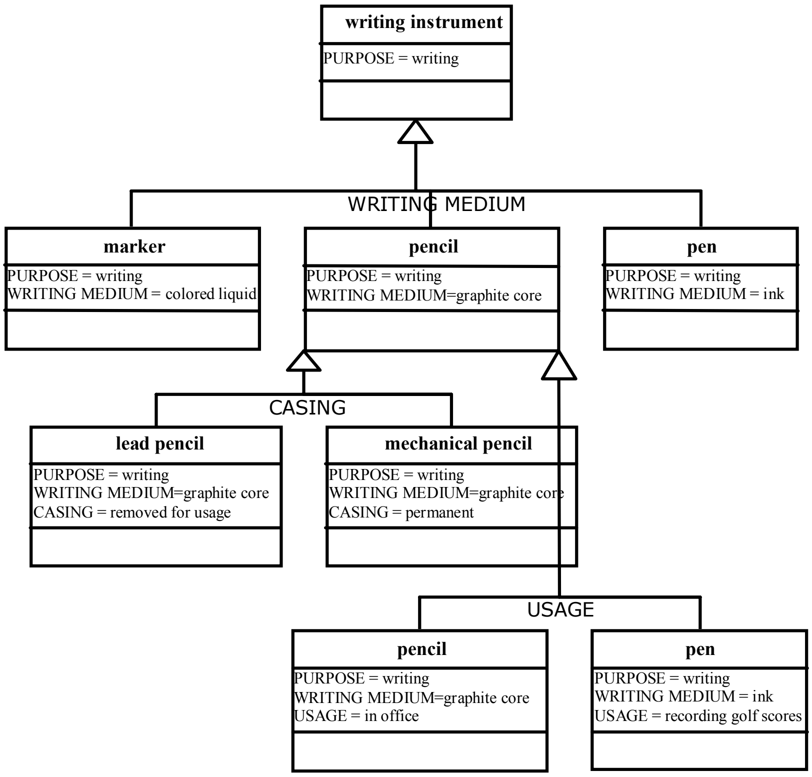
The use of UML (Unified Modelling Language) for conceptual data modelling is widespread, and in many cases it is being used also for terminological concept modelling.

If the concept diagram in Figure 2 is to be represented using UML class diagrams, we would get the diagram in Figure 4, where concepts are represented by UML class symbols, type relations as UML generalisations, and characteristics are represented as UML attribute-value pairs, where the values are default values. Subdivision criteria are represented as UML discriminators, although formally speaking there is no connection between UML attributes and UML discriminators.

3.2 Concepts and objects

In terminology, the term (or designation) *concept* is used to refer to the meaning of a term. A definition can be used to represent the concept, and the concept is reflected by an object or by a set of objects. All of these observations correspond with the general view in linguistics dating back to Saussure (1916), according to whom a linguistic sign (French: *signe linguistique*) consists of two parts, the *concept* [or, as Saussure prefers to call it, *signified* (French: *signifié*)] and the *sound-image* [character string or sounds uttered, Saussure calls this the *signifier* (French: *signifiant*)].

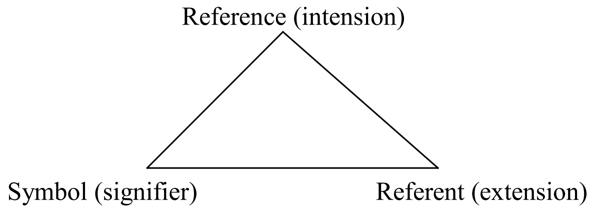
Figure 4 ‘Writing instruments’ in UML notation



Note: This figure is made with a trial copy of SmartDraw 2008.

Because of the characteristics making up the *intension* of a concept, the concept (or signified) makes it possible for speakers to use the sound-image (or signifier) to refer to *objects* in the extra linguistic world – the *extension* of the concept. This relationship is referred to as *the semantic triangle*, c.f. Ogden and Richards (1923), see Figure 5.

Figure 5 The semantic triangle



In the field of terminology, we analyse those linguistic signs that have a specific meaning within a specialised field, i.e. Language for Special Purposes or LSP. In this context, the *signifier* is called *designation*, the *signified* is called *concept* and *object* is *object*. There is no separate term to designate the concept of 'linguistic sign within a language for special purposes', but the term *term* is sometimes used in this sense, although it is mostly used to designate the concept of 'linguistic designation' (as opposed to a non-linguistic symbol), cf. the definition in ISO 1087-1:2000.

In terminology work, the objects in the extension of a concept are not represented in concept systems. If for some special reason they were to be rendered in a terminological UML diagram, the most appropriate representation would be as instances. This means that the distinction between classes and instances in UML to some extent corresponds to the distinction between intension and extension in terminology.

3.3 Characteristics and properties

As mentioned above, the characteristics making up the intension of a concept makes it possible to identify the set of objects that make up the extension of the concept. The identification of the objects in the extension is based on their *properties*, and whether these are consistent with the characteristics of the concept. Note that a given object may have properties that do not match any characteristic of the concept, e.g. in the case of pencils a given office lead pencil may have its outer casing painted yellow, it may be 15 cm long and have the characters 'HB2' stamped on the side. It could be argued that in fact the concept 'office lead pencil' has, e.g., the characteristic of having a length, but this holds of all the subordinates of the concept 'physical object', and hence the characteristic 'having a length' does not serve to differentiate 'office lead pencil' from other concepts. For this reason it is not interesting in a terminological analysis of writing instruments and should be excluded.

Properties of specific objects are not recorded in a terminology collection, and likewise they are not represented in concept diagrams. The focus of attention is concepts and their mutually differentiating characteristics. If properties were to be represented, they could be rendered as attribute-value pairs of instances.

3.4 A possible misunderstanding

When terminologists and data modellers discuss similarities of terminology work and data modelling, examples such as the one illustrated in Figures 1–3 are often used.

Describing the concepts in Figure 2, a terminologist may say that 'lead pencil' and 'mechanical pencil' are instances of 'pencil'. This may lead a data modeller to think that 'lead pencil' and 'mechanical pencil' should be modelled as instances in a UML diagram, and that the model reflects a database of different types of pencils. Continuing along these lines, the data modeller would assume that 'lead pencil' and 'mechanical pencil' are objects, which according to terminology theory (ISO 704:2000, p.3) have properties, which are then abstracted into characteristics. Hence the data modeller might conclude that 'removed for usage' and 'permanent' are properties, while 'CASING' is a characteristic.

The terminologist does not realise that the word *instance* has a special meaning in data modelling, and the data modeller has not entirely captured the linguistic distinction between intension and extension.

3.5 Individual concepts

In terminology, concepts are subdivided into *general* concepts and *individual concepts* depending on the number of objects in their extension. As individual concepts have only one object in their extension, it is tempting to represent individual concepts as UML instances, and in fact this has been proposed in SBVR (2007, p.420). However, since individual concepts are intensional rather than extensional, this would be erroneous. Individual concepts are concepts which happen to have only one object in their extension, and therefore they should be represented as UML classes like all other concepts. This representation will also allow modelling the fact that one object may be in the extension of two different individual concepts.

3.6 Overview: UML-representation of terminological items

Table 1 summarises how we propose to use UML symbols to represent terminological items in concept diagrams.

Table 1 UML representations of terminological items

Terminology item	*UML representation*
Concept (general/individual)	Class
Characteristic	Pair: Attribute – default value
Subdivision criterion	Discriminator
Object	Instance
Property	Attribute-value pair (of an instance)

It should be noted that the UML discriminator does not have any relation to the attribute of an *attribute-default value pair* that would reflect the relation between subdivision criterion and characteristic in terminology.

Table 1 suggests a straightforward correspondence between terminological concept systems and conceptual data models. There are, however, differences in the use of UML for IT-development and for terminology work. These will be addressed in the following section

4 Terminological concept modelling vs. ontology construction

A terminological concept model for a given domain is a shared and consensual representation of concepts in the domain and the relations among them. This is very close to the widely accepted definition of ontology given by Borst (1997, p.12) (based on Gruber, 1993): "An ontology is a formal specification of a shared conceptualisation". The most salient difference between a terminological concept model and an ontology according to this definition is the degree of formalisation. Therefore, in this section, we will look at some similarities and differences between terminology work and ontology construction.

A valued feature of formal ontologies is their *application independence* (see e.g. Meersman, 2000). This feature evidently also holds for terminological concept models, as will be illustrated by the example in the next section (Figures 6 and 7).

Pisanelli et al. (2002) claim that ontologies are useful for building better and more interoperable information systems because of a number of features. Many of these features also apply to rigorous terminology work adhering to the principles outlined in ISO 704:2000 and other sources. In the following text, we will discuss each of the features mentioned by Pisanelli et al. (2002).

Semantic explicitness: Terminology work does not make use of any formal logic, and hence, in this sense, semantic explicitness is not a feature of terminology work.

An explicit taxonomy: When a complete terminological analysis of a subject area is carried out, concept diagrams including taxonomic concept systems and definitions will be one of the results, see Wüster (1985, p.19). It is true that in many existing terminological resources concept diagrams and taxonomies are not included, while others lack rigorous definitions. In most cases, this is due to lack of financial resources for the rigorous development of terminology collections.

Explicit linkage to concepts and relations from generic theories: Definitions of some of the most general concepts within a specific subject field will contain concepts that are too general to be defined within that subject field, and usually there is no 'general upper concept system' or formal 'generic theory' to link to.

Absence of polysemy within a given formal context: One of the purposes of terminology work is to ensure unambiguous communication within a given subject field, and in fact monosemy of terms is an explicit goal, cf. Wüster (1985, p.79). In descriptive terminology work, one often finds that a given term is used with varying meanings, in which case the terminologist should give advice as to which one of these meanings (intensions or concepts) the term in question should be used for. For other concepts, the terminological resource should carry the information that sometimes that term is erroneously used in that sense, together with information about a better choice of term (the preferred term).

Modularity of contexts: Terminology work is usually carried out within a limited subject field, and may even be done covering the terminology for just one organisation or company, and hence contexts are modular.

Some minimal axiomatisation to detail the difference among sibling concepts: As explained in Sections 3 and 4, an important goal in setting up concept systems is to find the characteristics that differentiate coordinate (or sibling) concepts. These differentiating characteristics are not expressed as formal axioms, but they are formulated explicitly in natural language in intensional definitions, cf. ISO 704:2000 and Arntz and Picht (1989, pp.64–65), or in the form of feature specifications as illustrated in Figure 3. It can be added that in terminology it is quite common to encounter cases with multiple subdivision criteria, something which is only seldom accounted for in formal ontologies.

A good naming policy: Within the field of terminology, there are, in fact, rules for term formation, cf. ISO 704:2000 and Sager (1990, p.61 ff.). One rule says that terms should reflect the concept system, e.g. the term 'office pencil' in Figures 1–3 is constructed by adding to the term representing the superordinate concept a term representing the differentiating characteristic. In a concrete normative project where the end goal is a data model, it may be decided to use special rules to form names of data elements that represent concepts in the database.

Rich documentation: In terminology work, it is good practice to document everything and include references to each piece of information stored in the collection, see Arntz and Picht (1989, p.277).

Thus, in our opinion, terminology work carried out in accordance with the classical terminological principles described in ISO 704:2000, and in the terminological literature, has most of the features that are commonly ascribed to ontologies. The only ones not found in terminology work are those that can be characterised as logically formal:

- Semantic explicitness.
- Explicit linkage to concepts and relations from generic theories.

On this basis we conclude that terminology work in general can be characterised as non-formalised ontology construction, and as such it is obvious that terminology work can contribute to software engineering prior to the data modelling stage.

5 Terminological concept modelling vs. conceptual data modelling

Figures 6 and 7 show two parts of a concept diagram. This diagram clarifies the concepts used in a specific series of conferences that has been used as a case in Matthiassen (2000), viz. conferences organised by the International Federation for Information Processing (IFIP). The case describes a conceptual data model, but only gives a description in running text of the concepts behind the model. However, in order to understand the concepts behind the model and in this way to obtain a better background for creating the conceptual data model, the data base designer should set up a concept diagram. It should be noted that some of the concepts used in the above-mentioned case may have definitions that do not apply generally to all conferences. For example a sub session corresponds to one activity, which takes place at a specific time slot, i.e. one presentation or panel discussion.

The concept relations used in Figures 6 and 7 are: type relations (e.g. between 'participant' and the two subordinate concepts 'listener' and 'contributor'), part-whole relations (e.g. between 'academic activity' and 'session') and associative relations (e.g. 'submits' between 'author' and 'paper'). In Figures 6 and 7, some associative relations, that connect the concepts in the two parts of the concept system, are missing, e.g. the associative relation 'gives' between 'presenter with paper' and 'paper presentation'. In the diagrams, we have introduced only delimiting characteristics, i.e. neither inherited nor supplementary characteristics.

Figure 8 shows our version of a conceptual data model for the conference, described in Matthiassen (2000). In Figure 8, association classes are introduced in many-to-many relationships (e.g. 'reviews' between 'Reviewer' and 'Paper'), if there will be specific information related to the association.

Figure 6 Concept diagram for a conference system – 'Persons' (see online version for colours)

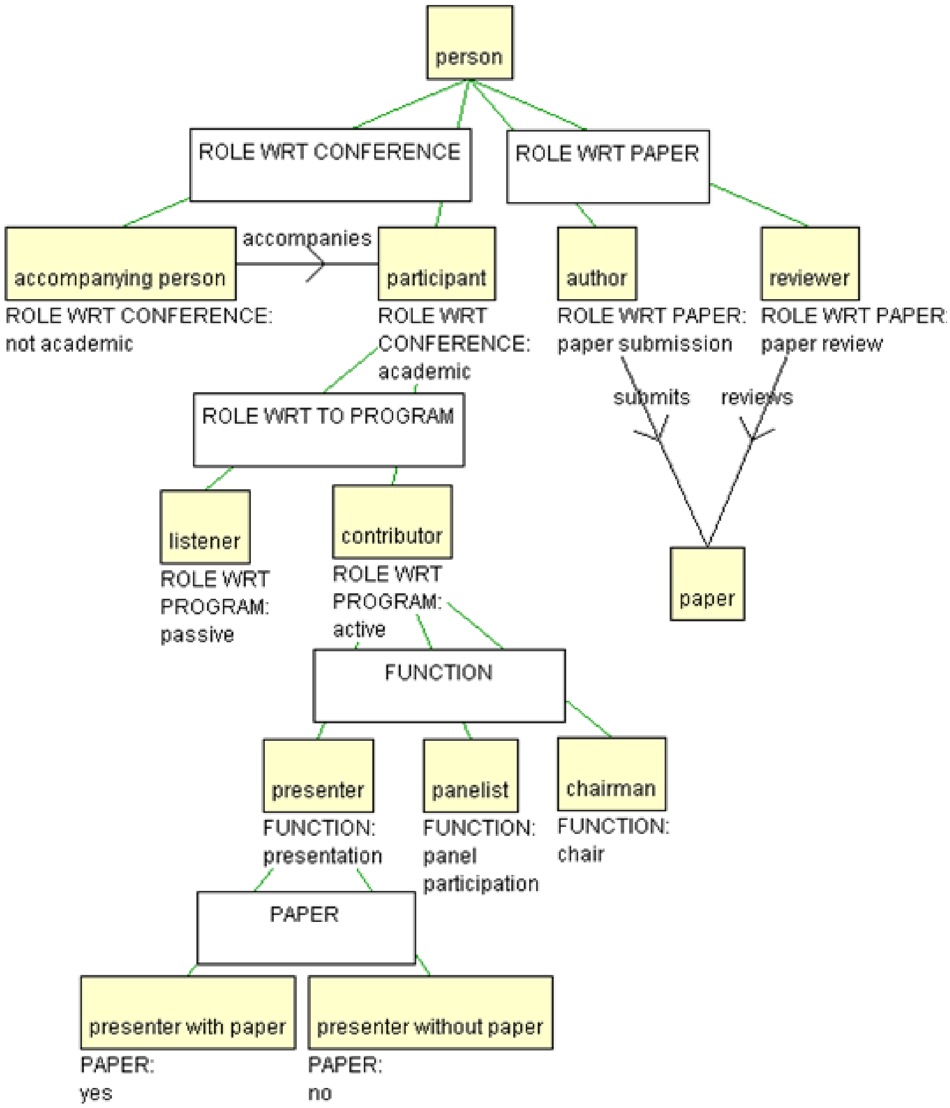
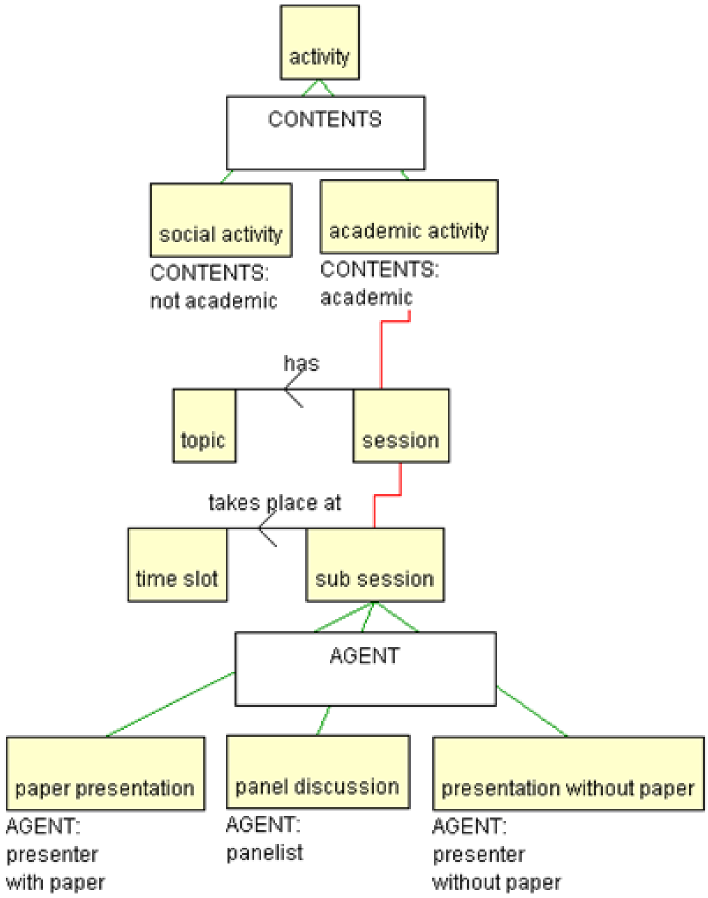


Figure 7 Concept diagram for a conference system – 'Activities' (see online version for colours)



In this model, one may recognise that most of the concepts from Figures 6 and 7, have been 'mapped' into classes of the data model. See for example 'person', 'accompanying person' and 'participant'. However, sometimes concepts in a concept diagram are not found in the corresponding conceptual data model. For example, the top concepts from concept diagrams will often not give rise to classes in a conceptual data model. This is the case for the concept 'activity'. Also some concepts correspond to attributes, e.g. the concept 'time slot' which is related to 'sub session'.

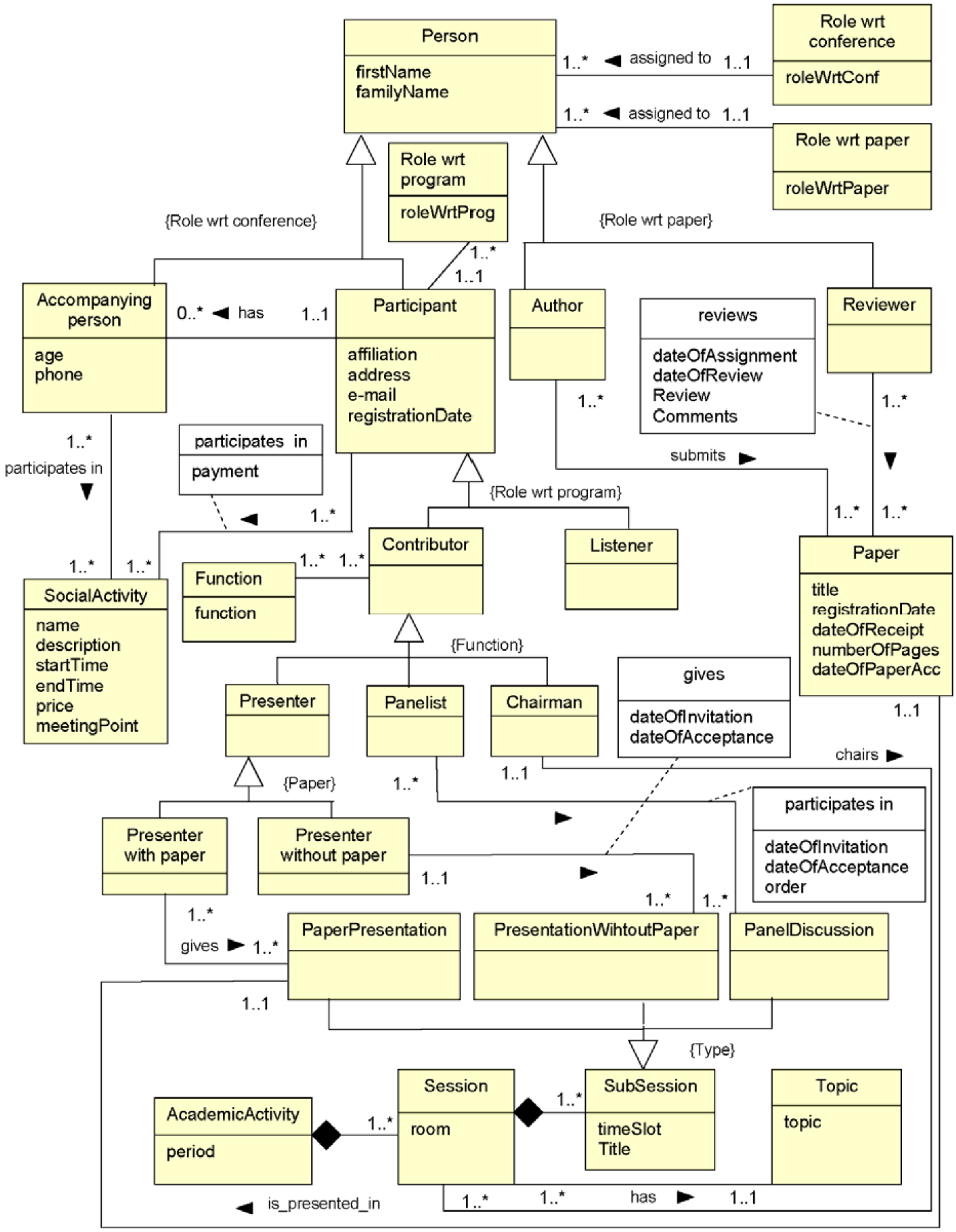
Conceptual data models are sometimes presented with attributes, sometimes without. In the latter case, attributes will be introduced when the logical data model is constructed. In this stage, primary keys, foreign keys and data types are also added. It may be argued that the attributes in the conceptual data model in Figure 8 should be found as concepts in the concept diagrams in Figures 6 and 7, and as classes in the conceptual data model, and that they should be transformed into attributes at the time of the creation of the logical data model.

Obviously the attributes in the conceptual data model do not define the classes, but they indicate which kind of information will be related to instances of the classes. For example the attributes of 'SocialActivity' do not define this class, whereas the characteristic 'CONTENTS: not academic' of the concept 'social activity' defines this concept together with its relation to 'activity'. In the concept diagram, the three concepts 'paper presentation', 'panel discussion' and 'presentation without paper' are defined by means of the super ordinate concept 'sub session' and the feature specifications 'AGENT: presenter with paper', 'AGENT: panelist' and 'AGENT: presenter without paper'. In the conceptual data model, the corresponding three classes do not even have any attributes. When implementing the data model, one may choose not to include a table corresponding to the class 'SubSession', and to add the attributes 'timeSlot' and 'title' to the three subclasses. Clearly these attributes do not contribute to a differentiation between the three classes.

It should be noted that not only concepts but also subdivision criteria in the concept diagram are mapped into classes in the conceptual data model, cf. for example 'Role wrt. conference' and 'Role wrt. paper'. These classes will form the basis for tables comprising pick lists in the resulting database, and the concepts grouped by these subdivision criteria will be possible values in the pick lists. For example the database for the conference will comprise a table 'RoleWrtConference' with the attribute 'roleWrtConf' and the two values 'Accompanying person' and 'Participant'.

Of course there are many ways of setting up a conceptual data model, and sometimes the conceptual data model may be very similar to the concept diagram, i.e. it will comprise a high degree of one-to-one mapping between concepts and classes. However, the conceptual data model will not supply semantic information about the concepts behind the classes, i.e. it will not comprise characteristics in the form of feature specifications.

Figure 8 Conceptual data model for the IFIP conferences (see online version for colours)



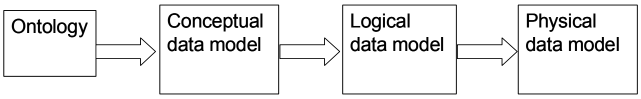
6 Terminological concept modelling as a first step in data modelling

In section 4, we concluded that terminology work can contribute to software engineering prior to the modelling stage, and in Section 5 we mentioned that some of the concepts used in the conference system case have definitions that do not apply generally to all conferences. For example a sub-session corresponds to one activity, which takes place at a specific time slot, i.e. one presentation or panel discussion. This illustrates that even though the database designer may have constructed other conference systems, he needs to be acquainted with the terminology used in the environment where this new system is going to be used. We find that the terminological concept modelling methods we have illustrated are very useful for this purpose.

In this section, we will elaborate on how terminological concept modelling may be integrated in the data modelling process in practice.

Figure 9 shows the order of the models that are needed in the development of an IT system.

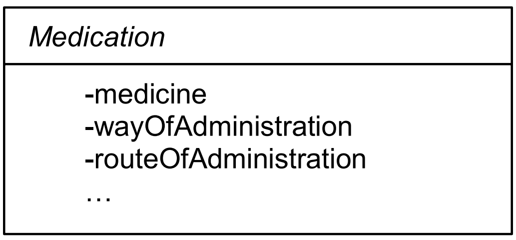
Figure 9 Order of the models related to the development of IT systems



In many cases, developers of IT systems encounter severe problems, because they neglect the necessity of developing a proper ontology (concept model) before they develop a conceptual data model for an IT system. Here we will give some examples from the development of Electronic Health Record (EHR) systems.

In Denmark, many software companies develop EHR systems that are tailored to the individual needs of hospitals all over the country. In order to obtain interoperability between all these proprietary systems and information provided to the national health statistical registries, a general conceptual data model for documentation of clinical information in EHR systems, GEPJ, was developed by the National Board of Health. Figure 10 shows the class diagram 'Medication' of GEPJ.[1] Version 1.0 was published in 2001 and version 2.2 in 2005. This means that work on the conceptual data model was initiated long before the concept clarification work was carried out within The Danish Council for Health Terminology. This work was initiated in 2004 and now nine domain groups have developed ontologies and concept definitions of concepts that are relevant to EHR systems. The domain group for medication finished its work in 2005, i.e. about the same time as the latest version of the GEPJ data model was published.

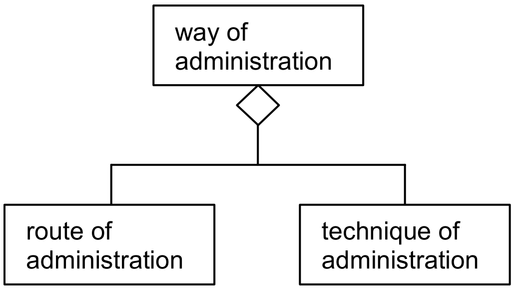
Figure 10 The class 'Medication' from the diagram 'Medication' of GEPJ version 2.2 2005



The concept clarification work of the domain groups revealed some discrepancies in the understanding of the concepts as compared to GEPJ, which may cause problems in the communication between health professionals and developers of EHR systems. Figure 11 shows a part of the ontology for *medication*.

From this diagram it is seen that the concept 'way of administration' specifies how the medication should be given. The concept 'way of administration' has two subconcepts: 'route of administration' and 'technique of administration'. The route of administration specifies that the medication should be given, e.g., by the buccal route (through the inner cheek or gum) or the sublingual route (under the tongue). The technique of administration specifies that the medication should be given, e.g., by injection.

Figure 11 Part of the ontology for *medication*



In the GEPJ class 'Medication' we find two attributes: 'Way of administration', corresponding to the superordinate concept, and 'Route of administration' corresponding to one of the subconcepts of this superordinate concept, viz. 'route of administration'. This does not make sense, and may give rise to confusion. The two attributes in the class 'Medication' should be 'Route of administration' and 'Technique of administration'. The concept 'way of administration' should not be mapped into an attribute in the GEPJ class.

There are other examples of discrepancies between the GEPJ data model and the ontologies and concept definitions that are the result of the concept clarification carried out by the domain groups within The Danish Council for Health Terminology and hence the concept clarification may form the basis for a sound revision of the GEPJ data model.

Typically concepts of an ontology will be mapped into either classes or attributes in the conceptual data model. However, in some cases concepts will not be mapped to classes or attributes in the conceptual data model. Future research will reveal to what extent it will be possible to set up rules for automatic mapping of concepts of an ontology into classes and attributes of a conceptual data model.

On the basis of the above example one may formulate one rule for the mapping between concepts of an ontology and classes and attributes of a conceptual data model: Two coordinate concepts may be mapped into two attributes of a class, but it would not be correct to map one of the coordinate concepts and their superordinate concept into two attributes of the class.

7 Concluding remarks

In terminology, we look for the characteristics that distinguish concepts from each other (corresponding to the properties that distinguish classes of objects from each other). In a database, you would be interested in the properties that distinguish the individual objects from each other, while in terminology we have no interest in those – the only interest we have in objects are the properties that they share with other objects, i.e. those properties which make it possible to classify them.

If you start out with a terminological concept clarification [before making (conceptual) data models], some of the things that are modelled as concepts in concept diagrams may end up as (UML) classes, attributes or values in the data model – and some central concepts may not be relevant at all. In the case of a terminology database, important concepts in terminology such as 'extension' and 'object' are not at all relevant for the data model.

The above identification of differences between terminological concept modelling and conceptual data modelling does not mean that we propose to separate the two completely. On the contrary, we think that terminology work can contribute to data modelling in many ways.

In data modelling, the following three modelling stages are recognised, cf. CEN CWA 15045 (2004):

- Conceptual data modelling
- Logical data modelling
- Physical data modelling

As argued above, we find that terminological concept modelling is quite different from conceptual data modelling, and therefore terminological concept modelling should be added as an initial stage in order to arrive at a common understanding of the terminology used in the domain.

References

Arntz, R. and Picht, H. (1989) *Einführung in die Terminologiearbeit*, Georg Olms Verlag, Hildesheim.

Borst, W.N. (1997) *Construction of Engineering Ontologies*, Centre for Telematica and Information Technology, University of Tweente, Enschede, Holland.

Carpenter, B. (1992) *The Logic of Typed Feature Structures*, Cambridge University Press, Cambridge, Mass.

CEN CWA 15045 (2004) *Multilingual Catalogue Strategies for eCommerce and eBusiness*, European Committee For Standardization, Brussels.

Gruber, T.R. (1993) 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, Vol. 5, No. 2, pp.199–220.

ISO 1087-1:2000 *Terminology work - Vocabulary - Part 1: Theory and Application*, International Standards Organisation, Geneva.

ISO 704:2000 *Terminology Work – Principles and Methods*, International Standards Organisation, Geneva.

Madsen, B.N. (1998) 'Typed feature structures for terminology work – Part I', *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, pp.339–348.

Madsen, B.N., Thomsen, H.E. and Vikner, C. (2005) 'Multidimensionality in terminological concept modelling', in Madsen, B.N. and Thomsen, H.E. (Eds.): *Terminology and Content Development*, Litera, Copenhagen, pp.161–174.

Matthiassen, L., Munk-Madsen, A., Nielsen, P.A. and Stage, J. (2000) *Objektorienteret analyse og design*, Forlaget Marko, Aalborg.

Meersman, R.A. (2000) 'The use of Lexicons and other computer-linguistic tools in semantics design and cooperation of database systems', in Zhang, Y. (Ed.): *CODAS Conference Proceedings*, Springer Verlag, Berlin.

Ogden, C.K. and Richards, I.A. (1923) *The Meaning of Meaning*, Routlegde & Kegan Paul, London.

OMG (2003) *Unified Modeling Language Specification Version 1.5*, Adopted Formal Specification of the Object Management Group, Inc.

Pisanelli, D.M., Gangemi, A. and Steve, G. (2002) 'Ontologies and information systems: the marriage of the century?', *Proceedings of the LYEE Workshop*, Paris.

Sager (1990) *A practical Course in Terminology Processing*, John Benjamins Publishing Company, Amsterdam.

Saussure, Ferdinand de (1916) *Cours de Linguistique Générale*, Payot, Paris; English translation: *A course in General Linguistics*, Philosophical Library, New York.

SBVR (2007) *Semantics of Business Vocabulary and Business Rules (FTF Working Specification - Ballot 9)*, OMG – Object Management Group.

Thomsen, H.E. (1998) 'Typed feature structures for terminology work - Part II', *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, pp.349–359.

Wüster, E. (1985) *Einführung in die allgemeine Terminologielehre und terminologische Lexikographie*, LSP Centre, Copenhagen School of Economics, Copenhagen.

Note

1 It should be noted that we use English glosses of terminology used in Danish healthcare.