Domain-Specific Concepts and Ontological Reduction within a Data Dictionary Framework

B. Heller*, H. Herre#, K. Lippoldt*

Onto-Med Research Group
*Institute for Medical Informatics, Statistics and Epidemiology (IMISE)
#Institute for Computer Science, Department of Formal Concepts
University of Leipzig, Germany
Liebigstrasse 27, 04103 Leipzig, Germany
barbara.heller@imise.uni-leipzig.de

Abstract. This paper describes a new method for the ontologically based standardization of concepts in the medical domain. As an application of this method we developed a data dictionary which firstly focused on trial-specific context-dependent concepts and relations. The data dictionary has been provided to different medical research networks via the internet by means of the software tool *Onto-Builder*. It is based on an architecture which includes terminologies, domain-specific ontologies and the top-level categories of *GOL*[1]. According to our approach top-level concepts are used to build definitions of domain-specific concepts on a firm ground through the process of ontological reduction, which is currently under development and the main ideas of which are outlined in this paper.

Introduction

Traditional medical terminology systems, for example the International Classification of Diseases (ICD) and Systemized Nomenclature of Medicine (SNOMED), include concept bases in hierarchical structures. These concept bases are limited in their expressive power as a result of e.g. their static structure, undefined mixed partitioning criteria, a mixture of different views and imprecisely defined relationships. Taking into consideration these limitations it is clear that such terminology systems cannot be used as standards for developing medical software applications. The missing standards are one reason for singular solutions for software systems in medicine, e.g. management systems for patient data and medical administration, medical research data bases, hospital information systems, electronic health care records and clinical trial management systems. Each software system has its own data model derived mainly from the goals of the corresponding application. According to this

---

[1] General Ontological Language is a formal framework for building ontologies. GOL is being developed by the Onto-Med Research Group at the University of Leipzig [http://www.onto-med.de].

background, new requirements on the representation methods of medical terminology systems have arisen in the last decade. Engineers developing medical software applications are demanding reusability in more than one software application, the handling of multiple granularities, the management of multiple consistent views and so on to finally achieve semantic correctness in the data model. [1] [2] The realization of these criteria presupposes a solution to the real problems in medical terminology, such as the uncertainty of medical knowledge, context-dependent representation, clarification of different user views (e.g. corresponding to different branches), and the specification of relations between concepts. Therefore a deeper semantic foundation at the level of concepts is necessary. Our approach exploits the method of ontological reduction to achieve the semantic foundation of terminologies.

We have developed and implemented the software tool *Onto-Builder* which supports the internet-based construction of ontologically founded data dictionaries. Such a data dictionary is a terminological framework for domain concepts which is partly based on the top-level categories of GOL [3] [4]. The GOL (General Ontological Language) project was launched in 1998 as a collaborative research project of the Institute for Medical Informatics, Statistics and Epidemiology (IMISE) and the Institute for Computer Science (IfI) at the University of Leipzig. The project is aimed, on the one hand, at the construction of a formal framework for building and representing ontologies, and, on the other hand, at the development and implementation of domain-specific ontologies in several fields, especially in the medical domain [5]. The results of these research activities led in the establishment of the interdisciplinary research group Onto-Med (Ontologies in Medicine). The *Onto-Builder* is one of the self-developed applications of the Onto-Med research group which represents among others formal ontological aspects and its computer-based application in the biomedical domain.

Our paper is structured as follows. In section 2 we sketch how the data dictionary can be integrated into the development process of medical software applications. Following this, we introduce our methodology in section 3 and define the relevant components. Sections 4-7 provide deeper insight into our approach by describing the model of the data dictionary, introducing the relevant ontological categories and relations of GOL and discussing our idea of ontological reduction. In the last two sections we discuss the chosen method and outlook on further work in this area of ontological research.

Application Environment

Designing flexible and scalable software applications for the medical domain requires expertise in software engineering and knowledge modeling as well as experience in the medical domain and medical terminology. Especially the latter is often a problem in this area of software development because of the so-called acquisition gap, i.e. the unavailability of the necessary advanced domain knowledge. Furthermore, existing medical terminology systems like UMLS and SNOMED are not powerful enough to represent concepts in a machine-processable way. Therefore our aim is to provide a unified medical concept base which can be used by software engineers. We have developed a data dictionary which offers context-dependent definitions of concepts, and contains general concepts for medicine (e.g. `therapy`, `laboratory parameter`) in its basic configuration. For the design and implementation of new medical software applications, relevant concept definitions can be extracted and queried from the data dictionary. If no adequate definitions are available in the dictionary, the basic concepts can be expanded by means of appropriate alternative definitions.

The following figure gives an overview of the use of the data dictionary during the definition process of a data base.

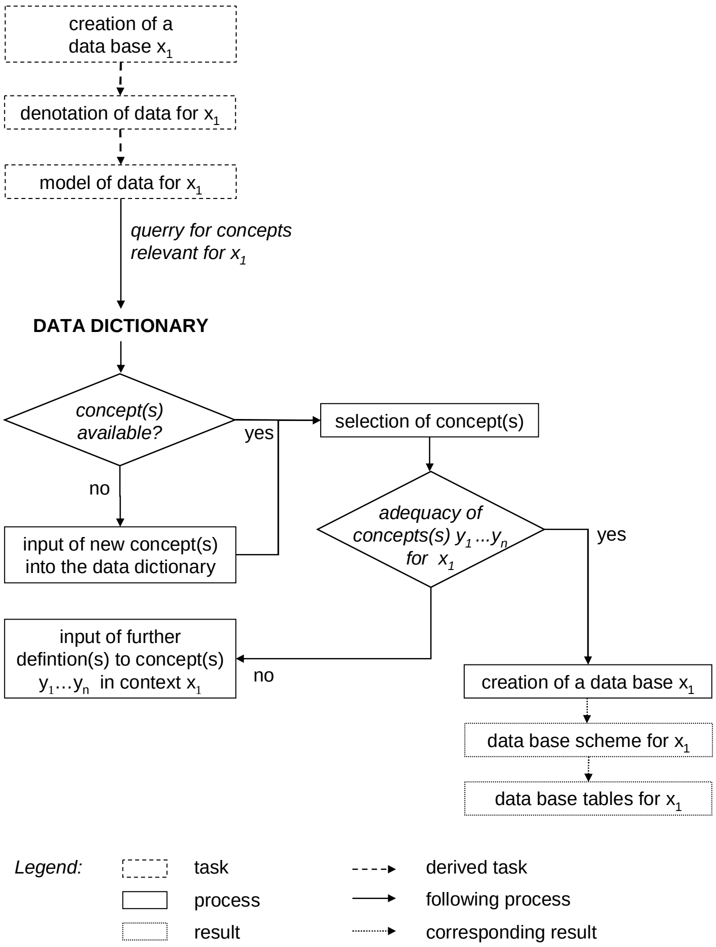

Figure 1: Use of the data dictionary for creating data bases

Definitions and Methodology

Our approach to an ontologically founded terminology is based on different interacting computer-based components, namely terminology, data dictionary, domain ontology, and top-level ontology (see also fig. 2). In the following, we briefly define these components and describe their interaction within our ontological approach:

Terminology: According to [6] a terminology is the complete stock of the concepts, their definitions and names in a concrete domain. An example of a very early medical terminology in the area of anatomy is the *Nomina Anatomica* [7].

Data Dictionary: A data dictionary is to be understood as a collection of data which are described and interpreted as concepts with context. We claim that our notion of data dictionary is applicable on the one hand to different domains such as medicine, biology or technology, and on the other hand to different application scenarios such as paper-based documents or software applications.

Domain Ontology: We use the notion of a domain ontology in accordance with Gruber [8]. A domain ontology provides formal specifications and computationally tractable standardized definitions for the terms used to represent knowledge of specific domains in ways designed to enhance communicability with other domains.

Top-Level Ontology:. A top-level ontology is concerned with the most general categories of the world, their analysis, interrelations, and axiomatic foundation. On this level of abstraction ontology investigates kinds, modes, views, and structures which apply to every area of the world.

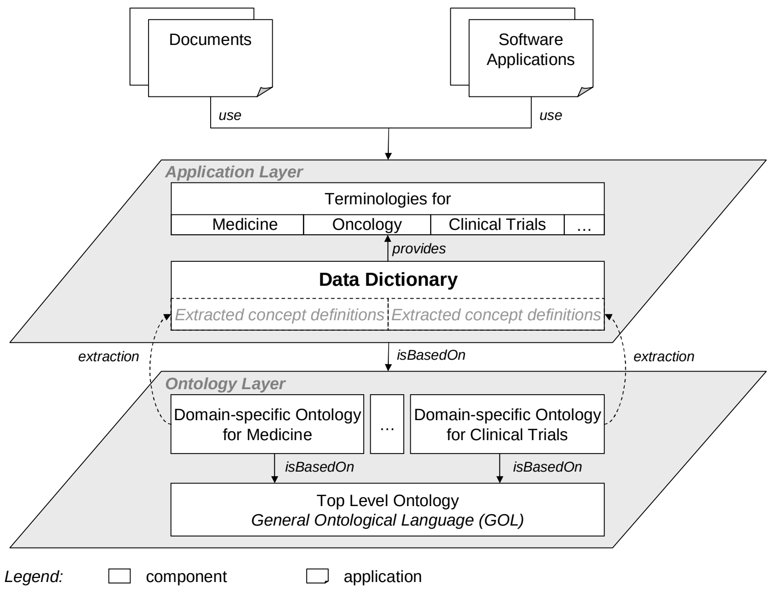

Figure 2: Two-layer model of an ontologically founded data dictionary

We assume as a basic principle of our approach that every domain-specific ontology (here in the field of clinical trials and medicine) must use as a framework some top-level ontology which describes the most general, domain-independent categories of the world. Therefore our data dictionary structure consists of two layers and is depicted in figure 2.

The *first layer* – called the application layer – contains two components: the data dictionary and the generic domain terminologies, i.e. domain-specific terminologies in the medical fields of oncology, clinical trials etc. The concept definitions of the generic domain terminologies are extracted from the identified and selected concept definitions of the data dictionary which are generic for the relevant domain. This domain generic information is taken as a basis for the definitions which are included in the component of generic domain terminologies. This means that these concept definitions are generic with respect to a confined area. The concepts of diagnosis, therapy and examination, for example, are defined generally in a terminology for medicine. In a special terminology e.g. for examination types, concrete specializations of general definitions are indicated with regard to single differentiable examination types, however.

The second component of the application layer consists of the data dictionary, which contains context-dependent concept definitions as well as references to corresponding information (e.g. radiographs, samples for medical documents) and provides the main definitions of concepts for domain-specific terminologies. The applications (here: documents and software applications) have access to the application layer from which they query relevant concept definitions and integrate them accordingly.

The *second layer* consists of two types of ontologies, namely the domain-specific ontologies (here for clinical trials, oncology and medicine) and the top-level ontology of GOL. The domain-specific ontologies describe formal specifications of concepts which are associated to a specific application. According to our approach, top-level concepts are used to build definitions of domain-specific concepts on a firm ground, and for this purpose we are developing a method of ontological reduction, the steps of which are outlined and discussed briefly in section 7. The GOL top-level ontology provides a framework with basic categories (e.g. universal/class, individual, quality, time, space, process and basic relations) which are described more precisely in section 5.

The two layers interact in the sense that the domain-specific concepts of the ontology layer are extracted from the data dictionary and are made available for the application-oriented concept descriptions which are provided for the application layer.

Application Layer

The Main Entities of the Data Dictionary

In this section we describe the model of the data dictionary and focus in particular on the following main entities: concept, denotation or term, description, context and relation. Definitions, relevant typings/classifications as well as references to the other components (Terminology, Domain Ontology, Top-Level Ontology) are included in the descriptions of these entities.

***Concept, Denotation, and Term**:* A concept is an abstract unit of meaning which is constructed over a set of common qualities [6] and which can also describe a cognitive entity (e.g., feeling, idea, thought). A denotation or term consists of one or several words and is the linguistic representation of a concept [9].

In the data dictionary model we distinguish between generic (e.g., `<disorder>`, `<process>`, `<treatment>`) and domain-specific (e.g., `<disease>`, `<symptom>`, `<medical treatment>`) concepts. A generic concept has a general meaning in different domains due to its domain-independent qualities. The concept `<treatment>`, for example, generally expresses that something or someone is handled in a certain way. A concept is generic with respect to a class D of domains if it applies to every domain which is included in D. A domain-specific concept, however, has a meaning only in a certain domain. The concept `<medical treatment>` which is only relevant in the domain of medicine is an example of this kind of concept. A domain-specific concept of the data dictionary refers to at least one ontological category which is specific for this domain and which is included in the ontology related to this domain. The examples chosen also show that it is possible to change a generic concept into a domain-specific one by adding an attribute. Rules for changing a concept type, the composition and decomposition of concepts are the topics of a forthcoming paper [9].

Description: The description of a concept contains information about its meaning with respect to its qualities, its relations to other concepts, statements about its use, etc. [9].

Our model offers the possibility of handling alternative descriptions. There are various reasons for the occurrence of alternative descriptions, e.g. different levels of granularity, static/dynamic aspects, subject area-related specifications, organization-dependent or institution-dependent differences as well as different expert opinions due to medical facts which have not yet been investigated completely. These different alternative definitions are represented with the help of contexts.

Context: With regard to the various discussions on the notion of context, e.g., in [10], we give here the following preliminary definition: A context is a coherent frame of circumstances and situations on the basis of which concepts must be understood.

As in the case of concepts, we similarly distinguish between generic and domain-specific contexts. A context is – roughly speaking – generic if it is associated with concepts whose descriptions include general properties/qualities (e.g., a generic context is `<process>` which includes the concept `<process course>` with among others the generic property `<process duration>`). Contrary to this, a domain-specific context includes concepts whose qualities/properties and their corresponding values specifically apply to a given domain (e.g., a domain-specific context is `<disease>` which contains the concept `<course of a disease>` with among others the domain-specific property `<course expression>` and the values `<chronic>` or `<acute>`) [9].

Relation: According to [3], relations are defined as entities which glue together the things of the world. We distinguish between three classes of relations: basic, domain-specific and terminological relations [9]. Our method handles at the present stage 12 basic relations which are briefly outlined in section 5. Examples of domain-specific relations are: <treatedBy>, <SideEffectOf>, and for terminological relations: <synonymy>, <homonymy>, <polysemy>.

The Model of the Data Dictionary

A brief overview of the basic entities and relations of the data dictionary model is given in figure 3. The syntax of the model in figure 3 follows the UML[2] syntax, whereas rectangles represent classes (here: entities), rhombus n-ary associations (here: relations) and lines represent relations between the entities.

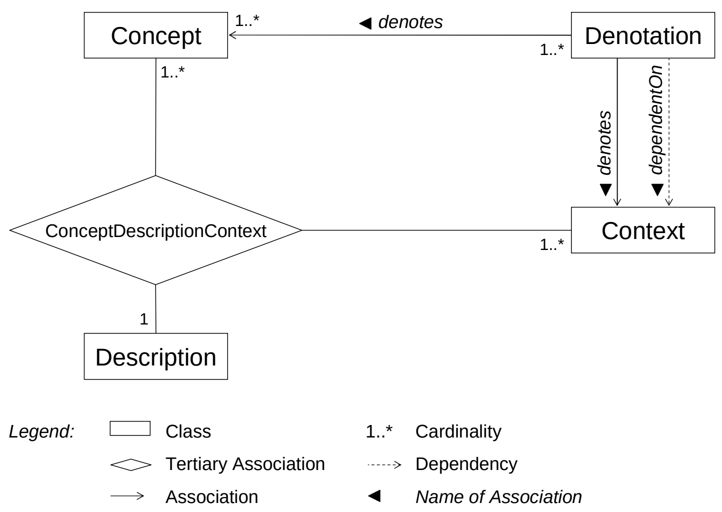

Figure 3: Data dictionary model (excerpt)

In our model, one Concept can be assigned to many Description/Context pairs [1..n] and one Context can be assigned to many Concept/Description pairs [1..n]. A Concept can be defined only by one Description in one Context. Different descriptions for a concept apply in different contexts.

The relation between Description, Concept and Context is expressed by the ternary association ConceptDescriptionContext which satisfies the abovementioned constraints. The entity Denotation describes Concepts and Contexts via the association *denotes*.

The dependency relation (here: *dependentOn*) between Denotation and Context means that the Denotation of a Concept can be dependent on the corresponding Context. If a Concept is not yet assigned to a Context, a default Denotation is given.

[2] Unified Modeling Language [11].

Ontology Layer

Domain-specific Ontology

A domain-specific ontology describes a specification of basic categories as these are instantiated through the concrete concepts and relations arising within a specific domain. For this reason, ways must be found to take into consideration different experts' views on the domain concepts and relations, as well as different goals and contextually determined foci.

Domain-specific ontologies have a low portability; they can be transferred to other applications only to a very limited degree. Methods also have to be found to raise the degree of portability of domain-specific concepts, for example by using strictly modular description methods.

The Top-Level Ontology GOL

The General Ontological Language GOL is intended to be a formal framework for building and representing ontologies. The main purpose of GOL is to provide a system of formalized and axiomatized top-level ontologies which can be used as a basis for constructing more specific ontologies. The GOL-framework consists of three components representing different levels of abstraction. Meta-GOL contains basic principles of semantic choice, a general view on categories and classes, methods of semantic transformation, and principles of meta-ontological analysis such as consistency checking. GOL on the object-level consists of a basic logic and a representation language based on a typed logic which is specified by a syntax and a semantics. The core of GOL is a library of top-level ontologies [4].

In the following sections we sketch briefly certain ontologically basic categories and relations of GOL which support the development of domain-specific ontologies. A more detailed description of the ontological categories, the basic relations and some axioms of GOL are expounded in [3] [4].

Hierarchy of GOL Categories (Excerpt)
The following figure (Figure 4) shows an excerpt of the hierarchy of categories in GOL (advanced version) [3].

Sets, Classes, and Urelements
The main distinction we draw is between *urelements* and *classes*. C*lasses* (which include sets) constitute a metamathematical superstructure above the other entities of our ontology.

Urelements
Urelements are entities of type 0 which are not classes. Urelements form an ultimate layer of entities lacking set-theoretical structure in their composition. Neither the membership relation nor the subclass relation can reveal the internal structure of urelements. We shall assume the existence of three main categories of urelements,

namely *individuals*, *universals*, and *entities of space and time*. An *individual* is a single thing which is in space and time. A (*primitive) universal* is an entity that can be instantiated by a number of different individuals. We distinguish several classes of universals: immanent universals, concepts and textual types. We assume that the immanent universals exist in the individuals (*in re*) but not independently of them. On the other hand, humans as cognitive subjects conceive of (immanent) universals by means of concepts that are in their mind. For this reason, every relevant top-level ontology has to include the class of concepts. The symbolic-linguistic representation of concepts is based on textual types which exhibit another kind of universal. We want to emphasize that also higher-order (non-primitive) universals are needed for the classification of domain concepts (e.g. in the biomedical domain), and that a universal cannot be captured by its extension. For these reasons the underlying representation language of GOL contains (intensional) categorial types and (extensional) class types of arbitrary finite order. Here, in developing GOL, the Onto-Med team draws on its experiences in analyzing UML and other modeling languages [12].

Alongside urelements there is the class of *formal relations*. We assume that formal relations are classes of certain types.

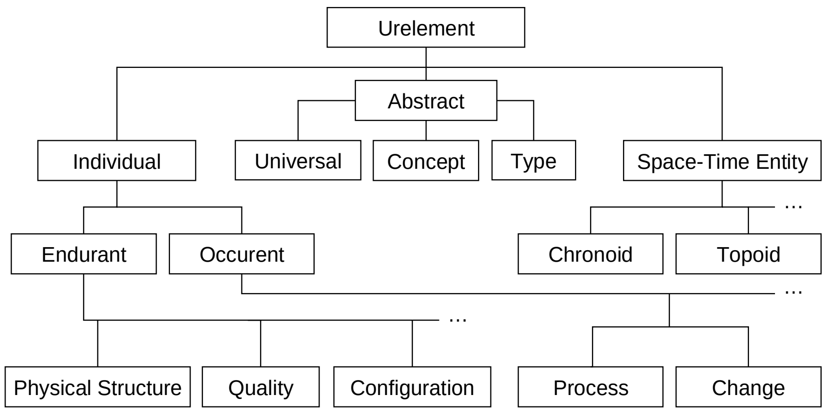

Figure 4: Hierarchy of categories in GOL (excerpt)

Space and Time

In the top-level ontology of GOL, *chronoids* and *topoids* represent kinds of urelements. *Chronoids* can be understood as temporal intervals, and *topoids* as spatial regions with a certain mereotopological structure.

Chronoids are not defined as sets of points, but as entities *sui generis*. Every chronoid has boundaries, which are called *time-boundaries* and which depend on chronoids, i.e. time-boundaries have no independent existence. We assume that temporal entities are related by certain formal relations, in particular the *part-of relation between chronoids*, the relation of *being a time-boundary of a chronoid*, and the relation of *coincidence between two time-boundaries.*

Endurants and Processes

Individuals are entities which are in space and time, and can be classified with respect to their relation to space and time.

An *endurant* or a continuant is an individual which is in time, but of which it makes no sense to say that it has temporal parts or phases. Thus, endurants can be considered as being wholly present at every time-boundary at which they exist.

Processes, on the other hand, have temporal parts and thus cannot be present at a time-boundary. For processes, time *belongs to them* because they *happen in time* and the time of a process is built into it. A process *p* is not the aggregate of its boundaries; hence, the boundaries of a process are different from the entities which are sometimes called *stages* of a process.

Substances, Physical Structures, and Objects

Physical structures are individuals which satisfy the following conditions: they are endurants, they are bearers of properties, they cannot be *carried by* other individuals, and they have a spatial extension.

The expressions *x carries y* and *x is carried by y* are technical terms which we define by means of an ontologically basic relation, the *inherence relation* which connects properties to substances. Inherence is a relation between individuals, which implies that inhering properties are themselves individuals. We call such individual properties *qualities*. Examples of objects are `an individual patient, a microorganism, the heart` (each considered at a time-boundary).

We assume that the spatial location occupied by a substance is a *topoid* which is a 3-dimensional space region. A *physical object* is a physical structure with unity, and a *closed physical object* is a is a physical structure whose unity is defined by the strong connectedness of its parts. Objects may have (physical) boundaries; these are dependent entities which are divided into *surfaces*, *lines* and *points*.

Qualities and Properties

Qualities (or individual properties) are endurants; in contrast to physical structures, they are entities which can exist only within another entity (in the same way in which, for example, an individual form, color, role or weight exists only in a certain body). Examples of individual properties (qualities) are *this* color, *this* weight, *this* temperature, *this* blood pressure, this thought. According to our present ontology, all individual properties have in common that they are dependent on physical structures where the dependency relation is realized by inherence.

Situoids, Situations, and Configurations

Situations present the most complex comprehensible endurants of the world and they have the highest degree of independence among endurants. Our notion of situation is based on the situation theory of Barwise and Perry [13] and advances their theory by analyzing and describing the ontological structure of situations.

There is a category of processes whose boundaries are situations and which satisfy certain principles of coherence and continuity. We call these entities *situoids*; they are the most complex integrated wholes of the world, and they have the highest degree of independence. Situoids may be considered as the ontological foundation of contexts.

Relations

We can distinguish the following basic ontological relations of GOL in table 1, which are needed to glue together the entities introduced above. A more detailed description of the relations is given in [3] [4].

Table 1: Basic relations in GOL

Basic Relation	**Denotation(s)**	**Brief Description**
Membership	$x \in y$	set *y* contains *x* as an element
Part-of	*part(x, y)* *tpart(x, y)* *spart(x, y)* *cpart(x, y)* *part-eq(x, y)* *tpart-eq(x, y)* *spart-eq(x, y)* *cpart-eq(x, y)*	*x* is a part of *y* *x* is a temporal part of *y* *x* is a spatial part of *y* *x* is a constituent-part of *y (y contains x)* the reflexive version of *part* the reflexive version of *tpart* the reflexive version of *spart* the reflexive version of *cpart*
Inherence	*i(x, y)*	moment *x* inheres in substance *y*
Relativized Part-of	*part(x, y, u)*	*u* is a universal and *x* is a part of *y* relative to *u*
Is-a	*is-a(x,y)*	x *is-a* $y =_{df} \forall u\, (u :: x \rightarrow (u :: y))$
Instantiation	$x :: u$ $x : y$ $x ::_i y$	individual *x* instantiates universal *u* list x instantiates relation y higher order instantiation, $i \geq 1$
Participation	*partic(x, y)*	*x* participates in process *y*, where *x* is a substance, an abstract substance or a substance process
Framing	*chr(x, y)* *chr(x)* *top(x, y)* *top(x)*	situoid *x* is framed by chronoid *y* denotes the chronoid framing *x* situoid *x* is framed by topoid *y* denotes the topoid framing *x*
Location and Extension Space	*occ(x, y)* *exsp(x, y)*	substance *x* occupies topoid *y* substance *x* has extension space *y*
Association	*ass(x, y)*	situoid *x* is associated with universal *y*
Ontical Connectedness	*ontic(x, y)*	*x* and *y* are ontically connected
Denotation	*den(x, y)*	symbol *x* denotes entity *y*

In table 1 the symbols x and y are entities. The concretization of the entities x and y depends on the type of the basic relation, e.g. tpart(x, y) means that x and y are *processes*. An exact specification of the admissible types of arguments of the basic relations in table 1 is presented in [4].

Ontological Reductions

An *ontological reduction* of an expression E is a definition of E by another expression F which is considered as *ontologically founded on a top-level ontology*. An expression is considered as ontologically founded on the top-level ontology GOL [14] if it is built up from atomic formulas whose meaning is inherited from the categories included in GOL. Ontological reductions exhibit a special case of semantic transformation. A semantic translation of a knowledge base K into a knowledge base M is a semantics-preserving function tr from the specification language $SL(K)$ underlying K into the specification language $SL(M)$ underlying M. Semantic translations can be used to compare the expressive power of ontologies and constitute an approach to the integration problem for knowledge bases. Semantic translations can be used as a formal framework for schema matching, which is a basic problem in many database application domains, compare [14]. An outline of this theory which is being elaborated by the Onto-Med group is presented in [15].

We sketch the main ideas concerning the notion of an ontological reduction based on a top-level ontology *GOL*. A definition D of a concept C for example is – usually – given as a natural language expression $E(C_1,..,C_n)$ which includes concepts $C_1,...,C_n$. The concepts $C_1,..,C_n$ are in turn defined by other expressions based on additional concepts. In order to avoid this infinite regress we select a certain number of concepts $D_1,.., D_k$ – which arise from E – as primitive. An embedding of $\{D_1,...,D_k\}$ into *GOL* is a function tr which associates to every concept D_i a category $tr(D_i) = F_i$ of *GOL* which subsumes D_i, i.e. every instance of D_i is an instance of $tr(D_i)$. The problem, then, is to find a logical expression E_1 based on $\{F_1,...,F_k\}$ which is equivalent to the initial expression E; such an expression is called an ontological reduction based on *GOL*. It may be expected that – in general – the system *GOL* is too weak to provide such equivalent expressions. For this reasons *GOL* has to be extended to the suitable system GOL_1 by adding further categories. GOL_1 should satisfy certain conditions of naturalness, minimality (the principle of Occam’s razor), and modularity. The problem of ontological reduction includes four tasks:

1. construction of a set of primitive concepts (initialization problem)
2. construction of an ontological embedding into *GOL* (embedding problem)
3. construction of an extension GOL_1 of *GOL* (extension problem)
4. finding an equivalent expression (definability problem).

A developed theory of ontological reductions based on top-level ontologies is in preparation and will be expounded in [16].

Example

To illustrate some aspects of the ontological reduction method we consider the following short example. We focus on the first reduction step of selecting a set of primitive domain-specific concepts. Therefore, we will give a preliminary definition of primitive domain concepts.

Definition: A set of concepts C is called primitive concept base for a class DOM of domains (of the same granularity) iff every concept d $\in$ C is generic with respect to all domains in DOM and if there does not exist a concept $d \in C$ which is derivable from the set of concepts $C - \{d\}$ on the same granularity level.

A fully developed top-level ontology has to take into consideration levels of reality which include the problems related to the notion of granularity. The most important philosophical approach to this area of research - with respect to information system sciences is presented in [17] [18].

Application:
`Tissue` in the medical sense is to be seen as contained in a primitive domain-specific concept base because its meaning and interpretation is the same in different medical domains (e.g. pathology, endocrinology). The domain-specific concept `tissue` can be interpreted as a "part of an organism consisting of an aggregate of cells having a similar structure and function"[3]. Normally the concept `tissue` can be partly derived from the more granular concept `cell`. In our approach the derivation of concepts is limited to concepts of the same level of granularity and therefore the concept `tissue` is not derivable from the concept `cell`. In contrast to `tissue` the concept `fatty tissue` should not be considered as a primitive concept. It has the same meaning in different contexts but can be derived directly from the concept `tissue` and the concept `fatty` on the same granularity level. Further examples for primitive domain-specific concepts are `body, cell, organ, tumour, disease, therapy`.

To give an example for the main ideas of the ontological reduction sketched above we consider the

concept *C*	`organ system`
and its	
definition *D*	`A group of organs, vessels, glands, other tissues, and/or pathways which work together to perform a body function within a multicellular organism.`

As a first step we analyze the natural language definition *D* with regard to the concepts and relations it includes. These concepts and relations must be classified in primitive and derived concepts and relations. In the given definition the following concepts should be included, among others, in a primitive domain-specific concept base: `organ, vessel, gland, tissue, organism`. For further analysis let us consider the primitive concept `tissue` and focus on its structural aspects. The concept `tissue` has to be classified within the top-level ontology GOL as a substance. This assignment is part of the ontological embedding of the base of primitive concepts into the hierarchy of categories of *GOL*, i.e. `(tissue is-a substance)`.

Further steps of the ontological reduction have to take into consideration suitable extensions of *GOL* to finally achieve formal expressions (in the framework of GOL) which are semantically equivalent to the concepts included in the primitive concept base C.

---

3 [http://www.hyperdictionary.com/]

Results and Discussion

With regard to the construction of a standardized terminology for software applications in medicine we have developed a methodology for an ontologically founded data dictionary. The methodology is based on two layers – the application layer and the ontology layer. The application components and theories at the two layers have been developed in parallel since 1999. One result of our work on the ontology layer is the development of the top-level ontology of GOL with approximately 50 basic categories and 12 basic relations. In the area of the domain-ontology we have started with the definition of domain-specific concepts which are partly based on top-level categories.

Concerning the application layer we have constructed a data dictionary for clinical trials which contains context-dependent concept descriptions. This data-dictionary has been implemented as the web-based software tool *Onto-Builder* [19]. This tool has been provided via the internet to several research networks with approximately 500 medical experts. Against this background, the handling of different expert views is indispensable within the *Onto-Builder*. This requirement is fulfilled with the availability of contexts in the data dictionary model which handle different expert views, granularity issues as well as special aspects of clinical trials. The present version of the data dictionary includes approximately 13 contexts, 1000 domain-specific concepts and 2500 concept descriptions.

Using the data dictionary, a higher level of harmonization of concepts and concept descriptions in different clinical trial protocols has been achieved. This has been possible due to the availability of a terminological concept base which has led in turn to an improved quality assurance in the clinical trial context.

Conclusion and Future Work

The evaluation of the application and theory components has shown that the underlying models of the data dictionary and the top-level ontology of GOL can be adapted to other domains and to other ontologies (e.g. DOLCE) [20].

At the present stage our data dictionary is merely a concept base for clinical trials and not yet fully based on domain ontologies. The reasons for this lie on the one hand in the extraction of domain-specific concept descriptions from the ontological layer which has not yet been realized completely. On the other hand it is connected to the problem of the ontological reduction of natural-language concept definitions via a semi-formal definition to formal propositions based on the built-in top-level ontology and its extensions. In our methodology we have already developed and partly integrated the first attempts at solving the ontological reduction problem.

Our future work consists in:

- the expansion of the theoretical framework with additional basic categories, e.g. situations, views and qualities
- the enlargement of the ontological reduction method according to functional aspects

- the elaboration of a theory of contexts and its evaluation in the area of clinical trials
- the incremental refinement of domain-specific concept descriptions with top-level categories
- the development of criteria for the specification of domain-specific concept types
- the explicit representation of semi-formal descriptions of domain-specific concepts
- the adaptation of the data dictionary to accommodate clinical trials in additional medical research networks.

Acknowledgement

We want to thank our medical and biometrics experts in the *Competence Network Malignant Lymphomas* (Grant No.: 01GI9995/0) and the *Coordination Centers for Clinical Trials*, *Cologne* and *Leipzig* for their fruitful discussions in the field of clinical trials and medicine. Many thanks are due to the members and Ph.D. students in the Onto-Med Research Group for implementing software modules for the *Onto-Builder* terminology management system and for the numerous discussions in building a top-level ontology. Last but not least we thank Evan Mellander for his efforts in the editorial work on this paper.

References

[1] Cimino JJ. Desiderata for Controlled Medical Vocabularies in the Twenty-First Century. *Meth Inform Med* 1998; 37(4-5):394-403.

[2] Rector AL. Clinical Terminology: Why Is it so Hard? *Meth Inform Med* 1999; 38(4-5):239-252.

[3] Heller B, and Herre H. Ontological Categories in GOL. *Axiomathes* 2004; 14:71-90.

[4] Heller B, and Herre H. Formal Ontology and Principles of GOL. Leipzig: Research Group Onto-Med, University of Leipzig; 2003. Report No. 1.

[5] Heller B, and Herre H. Research Proposal. Leipzig: Research Group Onto-Med, University of Leipzig; 2003. Report No. 2.

[6] Deutsches Institut für Normung e.V. DIN 2342 Teil 1: Begriffe der Terminologielehre. Berlin: Deutsches Institut für Normung e.V.; 10/1992.

[7] International Anatomical Nomenclature Committee. *Nomina Anatomica*. São Paulo; 1997.

[8] Gruber TR. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies* 1995; 43(5/6):907-928.

[9] Heller B, Herre H, Lippoldt K, and Loeffler M. Terminology Management for Clinical Trials (submitted).

[10] Bouquet P, Ghidini C, Giunchiglia F, and Blanzieri E. Theories and uses of context in knowledge representation and reasoning. *Journal of Pragmatics* 2003; 35:455-484.

[11] Booch G, Jacobson I, and Rumbaugh J. *The Unified Modeling Language User Guide*. Amsterdam: Addison-Wesley; 1999.

[12] Guizzardi G, Herre H, and Wagner G. On the General Ontological Foundation of Conceptual Modelling. In: Spaccapeitra S, March S, Kambayashi Y, eds. 21st International Conference on Conceptual Modelling (ER-2002); 2002 Oct 7 - 11; Tampere: Berlin: Springer; 2002. p. 65-78.

[13] Barwise J, and Perry J. *Situations and Attitudes*. Cambridge, MA, USA: Bradvord Books, MIT Press; 1983.

[14] Rahm E, and Bernstein PA. A survey of approaches to automatic schema matching. *The VLDB Journal* 2001; 10:334-350.

[15] Heller B, Herre H, and Loebe F. Semantic Transformation of Ontologies. forthcoming.

[16] Heller B, Herre H, and Loebe F. Ontological Reductions Based on Top-Level Ontologies. forthcoming.

[17] Poli R. The basic Problem of the Theory of Levels of Reality. *Axiomathes* 2002; 12:261-283.

[18] Poli R. Ontological methodology. *Int. J. Human-Computer Studies* 2002; 56:639-644.

[19] Heller B, Kuehn K, and Lippoldt K. Onto-Builder - A Tool for Building Data Dictionaries. Leipzig: Research Group Onto-Med, University of Leipzig; 2003. Report No. 3.

[20] Masolo C, Borgo S, Gangemi A, Guarino N, Oltramari A, and Schneider L. Wonderweb Deliverable D17. Preliminary Report, Version 2.0. Padova [Italy]: ISTC-CNR; 2002.