Methods for auditing medical terminological systems
Cornet, R.

Citation for published version (APA):
Description-Logic-based Methods for Auditing Frame-based Medical Terminological Systems

Ronald Cornet, Ameen Abu-Hanna

Published in *Artificial Intelligence in Medicine* 2005;34:201-217.
Abstract

Objective Medical terminological systems (TSs) play an increasingly important role in healthcare by supporting recording, retrieval and analysis of patient information. As the size and complexity of TSs are growing, the need arises for means to audit them, i.e. verify and maintain (logical) consistency and (semantic) correctness of their contents. This is not only important for the management of TSs but also for providing their users with confidence about the reliability of their contents. Formal methods have the potential to play an important role in the audit of TSs, although there are few empirical studies to assess the benefits of using these methods.

Methods and Material In this paper we propose a method based on description logics (DLs) for the audit of TSs. This method is based on the migration of the medical TS from a frame-based representation to a DL-based one. Our method is characterized by a process in which initially stringent assumptions are made about concept definitions. The assumptions allow the detection of concepts and relations that might comprise a source of logical inconsistency. If the assumptions hold then definitions are to be altered to eliminate the inconsistency, otherwise the assumptions are revised.

Results In order to demonstrate the utility of the approach in a real-world case study we audit a TS in the intensive care domain and discuss decisions pertaining to building DL-based representations. This case study demonstrates that certain types of inconsistencies can indeed be detected by applying the method to a medical terminological system.

Conclusion The added value of the method described in this paper is that it provides a means to evaluate the compliance to a number of common modeling principles in a formal manner. The proposed method reveals potential modeling inconsistencies, helping to audit and (if possible) improve the medical TS. In this way, it contributes to providing confidence in the contents of the terminological system.

4.1 Introduction

4.1.1 Medical terminological systems

A medical terminological system (TS) is a representation of medical concepts, relations and terms. For example, “inflammation of the membranes of the brain or spinal cord” is a concept. This concept can be described by the synonymous terms Cerebrospinal Meningitis and Meningitis. Figure 4.1 gives an example of a TS that represents concepts, relations and terms. In the remainder of this paper we focus on concept definitions, and we will represent concepts with their preferred names.

Medical TSs provide an invaluable source of structured medical knowledge, serving a wide range of purposes. Historically they have grown primarily from the need to encode causes of death used for obtaining epidemiological data [1]. However, their application has been expanding towards for example decision
4. Description-logic-based Auditing of Frame-based systems

In order to adapt to this wide range of purposes, medical TSs have grown in size and complexity [2]. They evolved from simple taxonomies to semantic networks with (informal and formal) concept definition capability. The growth of terminological systems both in number and size is demonstrated by the UMLS Metathesaurus\(^1\), which incorporates over 100 (versions of) TSs, totalling over one million concepts. From the numerous papers about medical TSs, a list of desiderata for these systems, consisting of 12 items, has been formulated [3]. Examples of such desiderata include: the need for concept orientation of the representation formalism; the ability to allow for multiple superclasses; and the ability to define concepts formally. A formal, concept-oriented approach to modeling terminological knowledge can largely contribute to fulfil a number of these desiderata. Many contemporary TSs are based on this approach. Advances in the field of knowledge representation have resulted in representation formalisms that can deal with the increased complexity of medical TSs. The most notable are frames and description logics (DLs).

A frame-based representation [4] is commonly used to express definitions of concepts, as this formalism supports an intuitive way of knowledge modeling in which a concept is represented as a frame where its (characteristic) attributes are represented as slots of that frame. However, this formalism lacks declarative semantics, which hinders automated reasoning. Automated reasoning about the (medical) knowledge is important not only during knowledge modeling but also for exploiting knowledge to support the end user during e.g. the navigation through the knowledge base. Examples of reasoning services expected from the utilization of the TS include the classification of concept definitions, the detection of semantically equivalent forms of definitions (concept redundancy), and logical consistency checking of a concept definition. To perform this automatically, a formal basis for the knowledge representation formalism is needed. The formalism should have enough expressive power to appropriately represent con-

cepts in the domain of interest but it should also support algorithms that are tractable in practice.

A seemingly attractive formalism to consider is that of description logics, which is a family of formal languages. The attractiveness stems from the fact that a DL usually corresponds to a decidable fragment of first-order logic and, at the same time, organizes concept definitions in an intuitive object-oriented-like structure. A variety of DLs have been defined, each of which is characterized by the logical constructors allowed for expressing concepts and relations (which are called roles in the DL literature). These DLs provide varying levels of expressiveness, for which highly optimized reasoning algorithms have been developed.

4.1.2 Auditing medical terminological systems

As modeling knowledge in very large knowledge bases and evaluating their contents are complicated processes, the need arises for systematic, reproducible methods to support these processes. Modeling and evaluating TSs concern various aspects, ranging from ontological decisions to the comprehensiveness of the medical contents of a TS. Ideally, a knowledge base should satisfy four requirements according to [5]: (1) it should have the necessary knowledge (completeness), (2) the knowledge should be faithful to the real world (correctness), (3) the knowledge should not be self-contradictory (consistency), and (4) the system should have efficient algorithms to perform the inferences needed for the application (competence). Auditing is the process of assessing the fulfilment of (one or more of) these requirements.

During the last decade, various techniques have been applied for auditing medical TSs. In [6], so-called “semantic methods” are applied for the detection of: ambiguity; redundancy of concept pairs; inconsistency of parent-child relationships; and lack of semantic links. In [7], a technique is presented to audit concept categorizations, based on intersections of semantic types. In [8] the use of Protégé is described for detection of inconsistencies and reduction of redundancy. In [9], methods for finding missed synonymy are described. In [10], an algorithmic methodology called “semantic refinement” is presented, which helps detection of ambiguity, non-uniform classification, classification errors, omissions, redundant classification and missed synonymy. In [11], two algorithms are applied to detect (among others) improper assignment of relationships, redundant concepts, and omission of relationships.

In these approaches the modeller eventually does a manual interpretation of the knowledge but the computational methods help focus attention on possible errors or flaws. In our approach the interpretational burden is further shifted towards the method itself by means of automated reasoning. Apparent errors are automatically detected and the modeller has to decide whether they correspond to real errors or flaws, or whether they emerge because of assumptions made which later turn out to be too stringent. Our goal is to explore the possibilities for deploying DLs for auditing the concept definitions within a TS. In particular, we use the reasoning service of satisfiability testing to support checking the consistency of a medical TS. Our audit focuses on consistency and correctness.
4. Description-logic-based Auditing of Frame-based systems

**Fig. 4.2:** An example of (partly erroneous) concept definitions in a medical TS, representing concepts as boxes, explicitly defined involved \_anatomy relations as dashed lines, and \_a relations as solid arrows.

of concept definitions, including not only hierarchical but also non-hierarchical links. This forms another difference with the majority of the related research being performed as the focus in these approaches is on hierarchical relationships.

Figure 4.2 shows an example of concept definitions in a medical TS that contains a number of suboptimal representation of concepts: Meningitis and Cerebrospinal Meningitis are defined as separate concepts, whereas these should either be defined as synonymous terms referring to one concept (as in Figure 4.1), or as two equivalent concepts. Viral Hepatitis redundantly defines involvement of the Liver, and Hepatitis B should be subordinate to Viral Hepatitis instead of being coordinate to it. Although these definitions are complete and consistent as the semantics expressed by the definitions are correct, information is missing or redundantly specified. An incorrect definition is that of Kidney Hepatitis, which should not be defined at all, as this is non-existing. This definition can be rendered inconsistent by making explicit that Kidney and Liver are disjoint concepts, and that the involved \_anatomy of a Viral Hepatitis is always a Liver.

### 4.1.3 Approach

Formal representation of knowledge supports the automatic discovery of logical inconsistencies based on the unambiguous semantics of statements. For example, the DL-based expression $M \equiv W \land \exists C P$ has unequivocal, agreed-upon logical semantics, facilitating automated reasoning. Although the semantics of the logical constructs are clear, any meaning of the symbols, providing the expression with a real-world interpretation, remains unspecified. If we interpret the capital letters as respectively Mother, Woman, hasChild, and Person, this statement not only has formal semantics, but also real-world semantics. However, it needs to be made explicit how Woman and Person are defined, and when a relation between people is considered as “having a child”. In the remainder of this article we will not pay attention to the latter problem, but focus on
formal semantics. In addition, one needs to keep in mind that whereas free-text definitions can be verbose and are restricted only by the limits of natural language, formalization restricts the expressiveness of definitions to the limits of expressiveness of the formalism.

Our starting point is that the TS at hand is specified or implemented in a frame-based language. This is the case in the great majority of medical TSs available today. In our approach we migrate the frame-based TS to a DL-based one. The advantage of using a DL is that DLs usually come with well-designed algorithms for reasoning. These algorithms have been implemented in reasoners that are readily available. Because the frame-based representation leaves room for various interpretations, the migration requires making the semantics of the frame-based representation explicit before specifying a TS in the DL-based representation. We have developed a method to perform this migration by posing explicit assumptions on semantics of, for example, frame slots and slot-fillers. One needs to realize that the knowledge base resulting from this migration process is for the purpose of detection of inconsistencies, and not for use in (clinical) practice.

The idea behind the method is to start with stringent assumptions about definitions in order to force the reasoning system to identify potentially inconsistently defined concepts. An example of such an assumption is stating that subclasses are mutually disjoint. The identification of inconsistency is realized by exploiting the satisfiability services of a DL reasoner. An unsatisfiable concept may indicate a too stringent assumption about a definition, but may also indicate actual errors in the frame-based definition from which the DL-based concept is derived. However, as unsatisfiability may also be caused by a correct definition that involves another unsatisfiable concept, one needs methods to locate the erroneous definition(s), as is described in [12]. Our hypothesis is that going through the migration process, performing satisfiability testing and locating errors, provides a serious contribution for maintaining the contents of real-world medical TSs. To assess this hypothesis, we have applied our method to a real-world TS of reasons for admission in intensive care (called DICE) [13]. The development of the DICE knowledge base is an ongoing effort that is being performed at the department of Medical Informatics at the Academic Medical Center in Amsterdam, The Netherlands since the late 1990s.

This paper is organized as follows. In Section 4.2 we provide preliminaries on frame-based representation, description logics, and the differences between them. We describe our method in Section 4.3 and focus on error detection in Section 4.4. Section 4.5 reports on the results of the case study, whereafter the methods and the results are discussed in Section 4.6. We conclude with lessons learned about using DLs, and about our approach for modeling and evaluation of medical TSs.
4.2 Background

4.2.1 Frame-based and description-logic-based representations

During the last two decades, various efforts have been made towards the formal representation of medical knowledge bases using DLs. On the one hand, attempts have been made to apply an existing DL-based system to medicine, e.g. NIKL [14], or LOOM [15]. On the other hand, DLs tailored to medicine have been developed, prominently GRAIL [16], and Ontylog [17]. In this paper, we describe the application of a common DL and existing reasoners, specifically for the aim of auditing a knowledge base, using a medical TS as a case study. Auditing is performed by means of semi-automatic migration from a frame-based representation to a DL-based representation. In order to be able to perform such a translation, the mismatch between what can be represented using frames and using DL must be overcome. We will first describe the essentials of frame-based representation, exemplified by DICE, and of DLs, and then look deeper into important differences between them.

Throughout this paper we use in the running text Capitalized Sans Serif typeface when referring to class frames and DL concepts (e.g. Hepatitis), whereas we use lowercase emphasized typeface when denoting frame slots and DL roles (e.g. involves_anatomy).

4.2.2 Frame-based representation, exemplified by the DICE TS

Frames [4] provide a means of describing classes and instances, with slots of frames representing either relations to other frames, or properties of the represented class or instance. Frames can represent subclasses by means of an is_a relation, which implies inheritance by the subclass of slots (and any slot-fillers) from the superclass.

As an example of a real-world medical TS, we will use the DICE system (Diagnoses for Intensive Care Evaluation) [13]. Like many other medical TSs, DICE is organized around health problems, which are defined according to their anatomy, morphology, etiology, and system (e.g. vascular system, digestive system), as shown in Figure 4.3.

The frame-based representation of DICE is restricted to a small number of constructs, which are demonstrated in Figure 4.4. Class frames are inter-related with is_a and other slot types. Slots can be defined as transitive or non-transitive (this is not represented in Figure 4.4). An example of a transitive slot is is_part_of, implying that all parts of a structure that is part of a larger structure are also parts of the larger structure. Slot-fillers can be grouped and labeled with an “XOR” or “OR” facet, in order to specify whether instances can be defined with exactly one (XOR), or more than one (OR) value from the slot-fillers. Providing no value for the refinability facet, or explicitly specifying a DEF value, means that the slot-fillers are definitional. By “definitional” we mean that the slot-filler (e.g. Meninges) is entailed by the disease concept (e.g. Meningitis). Apart from those mentioned above, no other constructs (such as slot-hierarchies, cardinality constraints, or inverse slots) exist.
4.2. Background

Fig. 4.3: Overview of the domain model of the DICE TS. Two types of reasons for admission are distinguished: health problems and operative procedures. Slots are represented in italics, and followed by their fillers. Various examples of subclasses are shown in Figure 4.4.

The DICE knowledge base is implemented using class frames only, that is, without instance frames. The rationale behind this choice is that this allows for the definition of a taxonomy without a predefined level of maximal detail. Instance frames are used to represent patient-specific information (e.g. a diagnosis of an individual patient), hence they are not part of the DICE knowledge base.

The representation formalism of DICE provides the possibility of specifying facets of slots. A facet provides additional information about the semantics of a slot. Examples of such facets are transitivity of a slot (such as the is_part_of slot), and refinability of a slot. Refinability, which is a DICE-specific slot facet,

Fig. 4.4: Examples of frame-based class definitions. The is_a slot defines direct superclasses. Slot facets “XOR” and “OR” specify whether instances can be defined with exactly one (XOR), or more than one (OR) value from the slot-fillers.
is important for making a concept more specific by creating a composition with other concepts. The refinability facet enables applications to determine whether and how a proper slot-filler may be used in a composition. Figure 4.4 shows an example of refinability in the etiology of Viral Meningitis. Upon selection of Viral Meningitis, an application can show the possible values (i.e. Virus and all its subclasses) and request to specify one or more values as the etiology of the instance of Viral Meningitis. Practically, this means that when a diagnosis of an individual patient is specified, the user is enabled to provide any known details.

4.2.3 Description logics

Description logics are rooted in the knowledge representation formalism of semantic networks [18]. Semantic networks provide a graphical representation of knowledge, but lack declarative semantics, making it hard to reason with the knowledge represented. In order to provide a logical basis for KL-ONE-like languages [19], research was directed towards representations that are based on explicit (set-theoretic) semantics. This has resulted in the family of DLs, which are subsets of first-order logic (FOL). FOL provides substantial expressive power which can be used for the representation of knowledge. However, FOL does not provide an explicit structure for concept definitions, and reasoning in FOL can be undecidable. DLs provide fragments of FOL for formal definition of concepts. These definitions can either specify only necessary conditions (which we will refer to as primitive definitions), or specify both necessary and sufficient conditions (which we will refer to as non-primitive definitions). For example, axiom (1) in Figure 4.5 states that every Mother is necessarily a Parent, whereas axiom (2) states that every Mother is necessarily and sufficiently both a Woman and a Parent. This implies that every subsume of both Woman and Parent is also a Mother.

Each DL is characterized by the constructors it allows for. Examples of concept constructors are AND (\(\cap\)), OR (\(\cup\)), NOT (\(\neg\)), SOME (\(\exists\)), ALL (\(\forall\)), AT-LEAST (\(\geq\)). Axiom (3) in Figure 4.5 shows an example of a definition in which three constructors have been used.

Examples of role constructors are transitive closure (e.g for the is_part_of role: if A is_part_of B and B is_part_of C then A is_part_of C), role inverses (e.g. is CAUSED_BY is the inverse role of causes), or role hierarchies (e.g. has_sister is a kind of has_sibling role).

The constructors that have emerged from the migration process and will be used in this paper are AND (\(\cap\)), OR (\(\cup\)), NOT (\(\neg\)), SOME (\(\exists\)), ALL (\(\forall\)), and AT-MOST (\(\leq\)). No role constructors will be used.

The formal, set-theoretic semantics of DLs provide statements with an unequivocal meaning, although these statements are restricted by the expressiveness of the underlying DL. The foremost reasoning tasks with DLs are subsumption (classification) and satisfiability testing. Subsumption testing amounts to checking whether one concept is more general than another. Satisfiability testing is checking whether a concept expression does not necessarily denote the empty concept [20]. Reasoning in DLs is based on the open world assumption,
4.2. Background

1. Mother ⊆ Parent
2. Mother ≡ Woman ∩ Parent
3. Happy Father ≡ Father ∩ (Rich ⊨ ≥ 3 has_children)

Fig. 4.5: Some example description logic axioms.

basically meaning that the truth of what is not specified is yet unknown. This is in contrast to the closed world assumption often made in database querying, in which what is unspecified is assumed to be false. The complexity of reasoning algorithms is related to the expressive power of a DL. Increased expressive power generally leads to increased computational complexity. Reasoning with very expressive logics can be intractable or even undecidable.

DL-based knowledge bases generally consist of a TBox (Terminology box) containing axioms (such as the examples in Figure 4.5), and an ABox (Assertion box) containing assertions (e.g. Mary is a Mother; Betty is a child of Mary). A TBox is called coherent if it does not contain any unsatisfiable concepts.

For an in-depth description of DL the reader is referred to the Description Logic Handbook [20].

4.2.4 Differences between frames and description logics

Frames and DLs both provide means for representing concepts, relations, and instances. There are however a number of significant differences, which need to be taken into account in the process of migration from frames to DL. We will discuss four of these differences that are relevant in the context of this paper.

**Taxonomic placement.** In contrast to frames, DL-based reasoning makes it possible to infer subsumptions, beyond those explicitly stated in the classification hierarchy. Inferred subsumption in DL is driven by non-primitive concept definitions that hold necessary and sufficient conditions (consider the Mother example from axiom (2) in Figure 4.5).

**Disjointness and covering.** In contrast to frame-based representation, DLs allow to explicitly specify that concepts are mutually exclusive (disjoint). In DL, concepts can be defined as disjoint by stating that one is subsumed by the complement of the other: Virus ⊑ ¬ Bacterium.

In a similar way, covering can be specified in DL. In the example above, it is not clear whether or not a microorganism can exist that is neither a virus nor a bacterium. The proposition that there are no other microorganisms than viruses and bacteria, can be expressed as: Microorganism ≡ Virus ⊔ Bacterium.

**Slots versus roles.** Slots in frames and any slot-fillers can be interpreted in various ways. For example, a slot has_cause with slot-filler (Virus, Bacterium) may mean that virus and bacterium together form the actual cause, or any one of them is a possible cause (possibly combined). Moreover, it is not stated whether other causes are allowed, etc.
Conversely, explicit quantification of roles in DLs leaves no room for such ambiguity in interpretation. Existential quantification ($\exists$), e.g., Disease $\cap \exists$ has cause Virus, denotes diseases that necessarily have a cause which is a virus, but other causes might exist. Universal quantification ($\forall$) is used to limit possible role-values. E.g., Disease $\cap \forall$ has cause Virus, denotes diseases of which all causes (if any) are a (subsume of) virus. Combining existential and universal quantification makes it possible to precisely define the semantics of roles and role values.

**Slot facets versus role constructors.** Semantics of slot facets in frames are often unclear and application-dependent. Examples of such facets are the refinability and the transitivity facet as described above. The semantics of role constructors in DL are explicitly defined, and taken into account by DL reasoners.

### 4.3 Migration Methods

#### 4.3.1 Migration from frame-based to DL-based representation

In order to support automated consistency checking, definitions must be formalized. In our approach, the frame-based knowledge base will be translated into a DL-based representation. Because of the differences between these representations, as discussed above, the DL-based representation will form one interpretation of the frame-based representation. This interpretation is largely based on assumptions of the semantics that cannot be formally expressed in frames. As this migration process mainly aims at detection of inconsistent concept definitions, we will focus on modeling issues that we expect to have the greatest impact on satisfiability. We will first discuss how to perform the migration process automatically, and then describe how to fine-tune the migration in order to deal with assumptions that turn out to be too stringent.

In order to support automated consistency checking, definitions must be formalized. In our approach, the frame-based knowledge base will be translated into a DL-based representation. Because of the differences between these representations, as discussed above, the DL-based representation will form one interpretation of the frame-based representation. This interpretation is largely based on assumptions of the semantics that cannot be formally expressed in frames. As this migration process mainly aims at detection of inconsistent concept definitions, we will focus on modeling issues that we expect to have the greatest impact on satisfiability. We will first discuss how to perform the migration process automatically, and then describe how to fine-tune the migration in order to deal with assumptions that turn out to be too stringent.

#### 4.3.2 Stringent assumptions for creating a DL-based representation

The process of migration from the frame-based representation to DL is based on a number of assumptions and modeling decisions. These assumptions are guided by the first aim of the migration process: semi-automatic detection of inconsistent concept definitions. In order to be able to detect as many potential inconsistencies as possible, maximally stringent definitions are assumed. These stringent definitions are aimed at restraining the open world assumption, for example by explicitly stating disjointness of siblings, and universal as well as existential quantification. Without such stringent assumptions, no inconsistencies can be detected. For example, to detect inconsistency in role values, disjointness must be made explicit, and role values must be both universally and existentially quantified. The next example should help demonstrate this: Consider the frame-based representations of classes, (a) and (b), in Figure 4.6, where Incorrect Hepatitis is, of course, a fictive class. A reasonably straightforward DL-based representation of these classes would be the ones denoted by (1)
(a) Infective Hepatitis : Disease; involves_anatomy: Liver, has_abnormality: Infection

(b) Incorrect Hepatitis : Infective Hepatitis; involves_anatomy: Kidney

1. Infective Hepatitis ⊑ Disease ⊓ ∃ involves_anatomy Liver ⊓ ∃ has_abnormality Infection
2. Incorrect Hepatitis ⊑ Infective Hepatitis ⊓ ∃ involves_anatomy Kidney
3. Kidney ⊑ ¬ Liver
4. Infective Hepatitis ⊑ Disease ⊓ ∃ involves_anatomy Liver ⊓ ∀ involves_anatomy Liver ⊓ ∃ has_abnormality Infection

Fig. 4.6: Examples of frame-based specifications and statements for possible interpretation stated in description logic.

and (2). The DL-based Incorrect Hepatitis, as defined by (2), will however be satisfiable. To infer that this concept is unsatisfiable, which is what one wants, it needs to be explicitly stated that a Kidney is not a Liver, and that Infective Hepatitis is located in the Liver and only in the Liver, as is denoted by definitions (3) and (4).

Using these definitions, Incorrect Hepatitis will become an unsatisfiable concept. In other cases however, the assumption of universal quantification may be incorrect, leading to a concept that is unjustly considered unsatisfiable, as will be discussed in section 4.3.3

Below we will discuss ten assumptions that have been made in the migration process. These assumptions are categorized as follows. First, we treat assumptions that relate to concepts in general. Then, assumptions that are related to anatomy are discussed, as anatomical reasoning is important especially in the domain of medicine. Finally, we address assumptions regarding slot-fillers. Examples of the assumptions are shown in Table 4.1. The row number in the table refers to the number of the item addressing the respective assumption.

**Concept-related Assumptions**

1. All concepts are defined as primitive.

Concepts defined as non-primitive can be inferred to subsume other concepts, as they provide necessary and sufficient conditions. In the migration process it is not necessary to distinguish between necessary conditions on the one hand and necessary and sufficient conditions on the other hand, for detection of inconsistent definitions. This does not mean that only primitive definitions can be used once a system is migrated, but during the migration process, they play a minor role.
Tab. 4.1: Examples, taken from the DICE TS, of frame-based expressions and their assumed DL-based counterparts. The numbers refer to the assumptions in the text.

<table>
<thead>
<tr>
<th>Frame-based representation</th>
<th>Assumed DL-based equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Bacterium : Microorganism</td>
<td>Bacterium ⊑ Microorganism</td>
</tr>
<tr>
<td>2 CABG: Graft; Coronary_artery_procedure</td>
<td>CABG ⊑ Graft ⊓ Coronary_artery_procedure</td>
</tr>
<tr>
<td>3 Virus : Microorganism</td>
<td>Virus ⊑ Microorganism</td>
</tr>
<tr>
<td>Fungus : Microorganism</td>
<td>Fungus ⊑ Microorganism</td>
</tr>
<tr>
<td>... : Microorganism</td>
<td>Fungus ⊑ ¬ Virus ⊓ ¬ ...</td>
</tr>
<tr>
<td>HeartDisease: has_anatomy Heart</td>
<td>HeartDisease ⊑ ∃ has_anatomy Heart</td>
</tr>
<tr>
<td>HeartLungDisease: HeartDisease, has_anatomy Lung</td>
<td>LungDisease ⊑ HeartDisease ⊓ ∃ has_anatomy Lung</td>
</tr>
<tr>
<td>4 HeartDisease: has_anatomy Heart</td>
<td>HeartDisease ⊑ ∃ has_anatomy Heart</td>
</tr>
<tr>
<td>LungDisease: has_anatomy Lung</td>
<td>LungDisease ⊑ ∃ has_anatomy Lung</td>
</tr>
<tr>
<td>HeartLungDisease: HeartDisease, LungDisease</td>
<td>HeartLungDisease ⊑ HeartDisease</td>
</tr>
<tr>
<td>5 Heart: is_part_of Body</td>
<td>Heart_Struct ⊑ Body_Part</td>
</tr>
<tr>
<td>Heart_Entity ⊑ Heart_Struct</td>
<td></td>
</tr>
<tr>
<td>Heart_Part ⊑ Heart_Struct ⊓ ∃ is_part_of Heart_Entity</td>
<td></td>
</tr>
<tr>
<td>6 Heart : Organ</td>
<td>Heart_Structure ⊑ Organ_Structure</td>
</tr>
<tr>
<td>Kidney_Structure ⊑ Organ_Structure</td>
<td></td>
</tr>
<tr>
<td>Kidney_Structure ⊑ ¬ Heart_Structure ⊓ ¬ ... Structure</td>
<td></td>
</tr>
<tr>
<td>7 has_cause: DEF(Virus, Bacterium)</td>
<td>∃ has_cause Virus ⊓ ∃ has_cause Bacterium ⊓ ∀ has_cause (Virus ⊔ Bacterium)</td>
</tr>
<tr>
<td>has_cause: OR(Virus, Bacterium)</td>
<td>∀ has_cause (Virus ⊔ Bacterium)</td>
</tr>
<tr>
<td>has_cause: XOR(Virus, Bacterium)</td>
<td>≤ 1 has_cause ⊓ ∀ has_cause (Virus ⊔ Bacterium)</td>
</tr>
</tbody>
</table>
2. **Frames with multiple parents correspond to classes that are defined as a conjunction of multiple subsuming superclasses.**

This assumption is in accordance with the default interpretation of these frame-based descriptions.

3. **Ill-defined sibling frames are interpreted as mutually disjoint concepts.**

Many medical TSs define concepts in a domain of interest (for example diseases, operative procedures, etc.) using concepts from other domains (such as anatomy, or microorganisms). Whereas the concepts in the domain of interest are relatively well-defined (by specifying necessary conditions), concepts in the other domains are often ill-defined, i.e. subsumption is specified, but only few necessary conditions. The ill-defined concepts can hardly be distinguished due the lack of specified conditions. By stating mutual disjointness of subordinate concepts, concepts are “forced” to be different. For concepts in the domain of interest, which are well-defined, disjointness can be inferred from their definitions, hence disjointness needs not be stated explicitly. E.g.:

Viral Disease ⊑ Disease ⊓ ∃ cause Virus ⊓ ∀ cause Virus,
Bacterial Disease ⊑ Disease ⊓ ∃ cause Bacterium ⊓ ∀ cause Bacterium

These concepts are necessarily disjoint if Virus and Bacterium are disjoint.

In DICE, we have defined all concepts subsumed by Act, Abnormality, System and Etiology as mutually disjoint to each of their siblings. In Figure 4.4 for example, Virus, Bacterium and Fungus are defined as subclasses of Microorganism. In the migration process these are defined as mutually disjoint using axioms as expressed in Table 4.1.

4. **Slots for which fillers are specified override slot-fillers defined by superclasses.**

In a frame-based representation the interpretation of slot-fillers in subframes is unclear. It can either be an additional qualification (Heart-Lung-Disease: Heart-Disease, has_location Lung), a specification (Heart-Valve-Disease: Heart-Disease, has_location Heart-Valve), or a contradiction (Penguin: Bird, has_ability: Not_Flying) of the definition of the superframe. Interpretation of definitions as specifications implies that a conflicting specification will lead to an unsatisfiable concept. If a specification is intended to represent an additional qualification, the frame-based representation should represent this explicitly, by repeating the inherited qualification. In the example above, Heart-Lung-Disease should be specified with has_location (Heart, Lung).

5. **Parent frames and slots for which fillers have been specified are conjuncted.**

As explained in assumption 2 above, multiple parents are interpreted as a conjunction of subsuming classes. Slot-fillers that are specified for the parents, are all assumed to hold. If, for example, one parent specifies has_location heart, and another parent specifies has_location lung, this is interpreted as has_location (heart, lung). See assumption 8 below, on the interpretation of such multi-valued slot-fillers.
Anatomy-related Assumptions

Anatomy plays an essential role in medical TSs, as the vast majority of symptoms and diseases are associated with particular anatomical areas. The Foundational Model of Anatomy (FMA) [21] aims at becoming a reference ontology for anatomical knowledge. It is a frame-based ontology, containing about 70,000 distinct anatomical concepts. The most prominent relations between these concepts, apart from the is-a relations, are partitive relations. Whereas partitive relations play an important role in medical knowledge bases, they may pose great demand on expressiveness of DLs. This can be overcome by the use of structure-entity-part (SEP) triplets, as suggested in [22]. Motivation for using SEP triplets is to reduce computational complexity by avoiding the use of transitive roles and role chaining. This comes at the cost of having to define every anatomical component three times (as an entity, a part, and a structure). The SEP representation turned out to be very useful for the aim of detecting inconsistencies, as we will describe below.

6. The Anatomy taxonomy consists of anatomical structures, which can be represented using SEP triplets.

Anatomical reasoning is an important but difficult task. The use of stringent definitions for location is hindered by the transitivity of anatomical localization. A concept that is defined as located in some part of an anatomical component X, is implied to be located in component X as well. This can be dealt with by using role hierarchies, defining \textit{is\_part\_of} role as a subrole of \textit{has\_location}, and both roles being defined as transitive. This is, however, only useful for the existential quantifications, but is not feasible for universal quantifications, as is explained below.

Figure 4.7 shows various possibilities of dealing with part-whole reasoning. Using definitions (c) and (d), \texttt{HeartValveDisease} can correctly be inferred to be a \texttt{HeartDisease} due to the use of transitive roles and role hierarchies as defined in (a). However, using stringent definitions, as in (e) and (f), \texttt{Str\_HeartValveDisease} (which is explicitly defined as a \texttt{Str\_HeartDisease}) will, unjustly, become unsatisfiable. This can be overcome by changing the role-values as shown in (g) and (h). This role-value is an anonymous form of the \texttt{Heart\_Structure} concept in the SEP triplets, where \texttt{Heart} is the \texttt{Heart\_Entity} and \(\exists \text{ is\_part\_of} \texttt{Heart}\) is the \texttt{Heart\_Part}. We have therefore chosen to use SEP-triplets, in which case there is no need to define role hierarchies and transitive roles. The assumption is made that all concepts in the anatomy taxonomy are “structure” elements, for which corresponding “entity” and “part” concepts are defined.

7. The Anatomy taxonomy is treated as a partition, i.e. parts are considered to be disjoint.

The use of SEP triplets, as motivated above, has another major advantage. If the anatomical model is considered to imply a partition, it can (and for migration purposes should) be explicitly stated that every anatomical component is part of exactly one other component. We discussed above
4.3. Migration Methods

(a) is_part_of ⊑ has_location; is_part_of transitive, has_location transitive
(b) Heart Valve ⊑ ∃ is_part_of Heart
(c) Heart Disease ≡ Disease ⊓ ∃ has_location Heart
(d) Heart Valve Disease ⊑ Disease ⊓ ∃ has_location Heart Valve
(e) Str_Heart Disease ≡ Disease ⊓ ∃ has_location Heart ⊓ ∀ has_location Heart
(f) Str_Heart Valve Disease ⊑ Str_Heart Disease ⊓ ∃ has_location Heart Valve
   ⊓ ∀ location Heart Valve
(g) SEP_Heart Disease ≡ Disease ⊓ ∃ has_location (Heart ⊓ ∃ is_part_of Heart) ⊓ ∀ has_location (Heart ⊓ ∃ is_part_of Heart)
(h) SEP_Heart Valve Disease ⊑ SEP_Heart Disease ⊓ ∃ has_location (Heart Valve ⊓ ∃ is_part_of Heart Valve) ⊓ ∀ location (Heart Valve ⊓ ∃ is_part_of Heart Valve)

Fig. 4.7: Various ways of dealing with part-whole reasoning.

that definition (h) in Figure 4.7 results in a satisfiable concept. However, should we replace Heart Valve with, say, Kidney, this concept would still be satisfiable, as Kidney may be explicitly stated to be disjoint from Heart, but not from (∃ is_part_of Heart). SEP triplets can solve this issue by defining Heart Structure disjoint from Kidney Structure. This assumption is based on the fact that DICE has a relatively simple anatomy model. In contrast to this, FMA provides various views on partitioning, making it impossible to apply this assumption without introducing different kinds of parthood relations, as described in [23].

Slot-filler-related Assumptions

As discussed earlier, slot-fillers have no unambiguous interpretation. In the case of DICE, also the refinability facet of slots, described in Section 4.2.2, needed to be taken into account.

8. Slot-fillers with “DEF” refinability facet are interpreted as conjunctions of existential quantifications of the role values and universal quantification of the disjunction of the role values. Slot-fillers that are specified as part of the definition of a class frame (i.e. the refinability facet has the value “DEF”, like Hepatitis, by definition, has_location Liver) are interpreted as role values that necessarily all exist. Hence it is interpreted as a conjunction of existential quantifications. Moreover, the values are interpreted as the only allowed role values, which is represented as a universal quantification of the disjunctions.

9. Slot-fillers with “OR” refinability facet are interpreted as universal quantification of disjunction of the role values.
As the “OR” value for the refinability facet specifies possible slot-fillers, no existential quantification is assumed. However, the specified values are interpreted to be the only allowable values, hence they are interpreted as universal quantification of disjunction of the role values.

10. **Slot-fillers with facet “XOR” are interpreted as universally quantified and number restricted roles:** ≤ 1.
An “XOR” value for the refinability facet is a stricter alternative of the “OR” value, as at most one value is allowed. Hence, the interpretation is the same as the “OR” value, with an additional number restriction (at most 1) on the allowed role values.

### 4.3.3 Revising assumptions in light of contradiction

As the assumption of universal quantification may turn out to be too stringent in many cases, the frame-based representation has been extended with a special-purpose slot-facet used during the migration. This facet has been added to the slots to explicitly specify whether a slot should be considered to represent universal quantification or not. This facet can be set during the migration process to override the default assumption. Removing universal quantification on role values prevents other existentially qualified role values (from subsumed or subsuming concepts) to lead to inconsistencies. Disabling universal role quantification is only done for cases where the semantic correctness of the initially conflicting definitions has been verified. For example, **HeartDisease** is defined as referring to diseases that involve the heart. **Cardiac Pulmonary Oedema** is a **Heart Disease**, but it also involves the **Lungs**. Hence, universal quantification is disabled on the **has_anatomy** role of **HeartDisease**.

The mutual disjointness of sibling concepts may also turn out to be a too stringent assumption, especially in cases where more than one siblings are justly subsuming a concept. As these cases might actually indicate the need for a more drastic remodeling, no measures have been taken to change this assumption. In practice, only a few cases with this problem were detected.

### 4.4 Methods for Detecting and Correcting Errors using DL-based Reasoning

The migration process as described in the previous section results in a DL-based representation of a knowledge base. Subsequently, a DL reasoner such as FaCT\(^2\)[24] or RACER\(^3\)[25] can be used to classify this knowledge base and find any unsatisfiable concepts. We will first discuss which kind of modeling errors can be detected using a DL reasoner, and will then look into strategies to find the actual erroneous definition, and to correct them. We conclude this section with more advanced means of error detection, which are currently under development.

---

\(^2\) [http://www.cs.manchester.ac.uk/~horrocks/FaCT/](http://www.cs.manchester.ac.uk/~horrocks/FaCT/) (accessed February 17, 2006)

\(^3\) [http://www.sts.tu-harburg.de/~r.f.moeller/racer/](http://www.sts.tu-harburg.de/~r.f.moeller/racer/) (accessed February 17, 2006)
4.4. Description-Logic-based Reasoning to Detect and Correct Errors

4.4.1 Detectable and undetectable modeling errors

A number of modeling errors were shown in Figure 4.2. Unfortunately, not all of these errors can be easily detected using a DL reasoner, or can only be detected under certain conditions. We will now discuss these errors, which were described in Section 4.1.2 in more detail. Meningitis and Cerebrospinal Meningitis are duplicate definitions of what is actually one concept. Such duplicate definitions are only detected by a DL reasoner if both definitions are non-primitive. In this case, the reasoner will conclude that two concepts defined according to the same necessary and sufficient conditions are the same. However, if either one or both of them is defined as primitive, this redundancy will not be detected.

Hepatitis B should be subordinate to Viral Hepatitis instead of being coordinate to it. This could be inferred by a DL reasoner, but only when Viral Hepatitis is defined as a non-primitive concept. In that case, necessary and sufficient conditions are specified, and the reasoner can infer that Hepatitis B is subsumed by Viral Hepatitis. However, reasoners generally do not provide an overview of inferred subsumptions, although applications exist that can do this, based on the results of classification by a reasoner. OilEd is an example of such an application.

Viral Hepatitis redundantly defines involvement of the Liver. Although redundant modeling can be considered a modeling imperfection, it is not detected by DL reasoners, as it does not result in any classification change or unsatisfiable concept. Kidney Hepatitis represents a concept that should not exist. However, it could also refer to an incorrectly modeled concept, for which the involvement of Kidney was accidently and incorrectly specified. This type of modeling errors can actually be detected by DL reasoners.

The examples from Figure 4.2 show that not all modeling errors can be detected by DL reasoners. Hence, satisfiability is not a proof of correct definition, but rather, unsatisfiability is an indication of some modeling error. Modeling errors however are not the only cause, but the first possible cause of unsatisfiability of a concept.

The second cause of unsatisfiability originates from the migration process. If a class frame is modeled correctly, but incorrect (i.e. too stringent) assumptions are posed on the class frame or any of its superclasses during the migration process, the resulting concept can become unsatisfiable, as was discussed in Section 4.3.3.

The third possible cause for unsatisfiability originates from the DL-based reasoning process. All subsumees of an unsatisfiable concept are unsatisfiable as well. In the same manner any concept will become unsatisfiable that is defined according to an existentially quantified role that has an unsatisfiable concept as a role value. Figure 4.8 presents an example, where unsatisfiability of one concept (ErroneousHepatitis) leads to unsatisfiability of two other concepts. Classification of this example with a DL reasoner will result in three unsatisfiable concepts, leaving it to the user to determine the origin of the unsatisfiability. Hence, before one can correct modeling errors or change migration assumptions, one needs to

---

4 http://oiled.man.ac.uk/ (accessed February 17, 2006)
ErroneousHepatitis $\sqsubseteq$ Hepatitis $\sqcap$ $\exists$ has_location Kidney  
ViralHepatitis $\sqsubseteq$ ErroneousHepatitis $\sqcap$ $\exists$ has_ etiology Virus  
HepaticSyndrome $\sqsubseteq$ Disease $\sqcap$ $\exists$ involves ViralHepatitis

Fig. 4.8: Examples of propagation of unsatisfiability. The unsatisfiability of the concept ErroneousHepatitis leads to unsatisfiability of the other concepts.

An expedient approach to determine which unsatisfiable concepts lead to other unsatisfiable concepts is to start with those concepts that are used as role-values for other concepts (such as ViralHepatitis in the example in Figure 4.8). More specifically, knowledge bases can often be “modularized” into independent knowledge bases [26]. For example, DICE can be considered as a knowledge base that includes four mutually independent knowledge modules, i.e. “Abnormality”, “Act”, “Anatomy and System” and “Etiology”. This independence is demonstrated in Figure 4.3 by the fact that all slots for frames refer to frames within the same knowledge module (e.g. the slot-fillers in “Anatomy and System” only allow frames from this module). Hence it is expedient to first address unsatisfiable concepts in the various knowledge modules. When these unsatisfabilities are corrected, unsatisfiable subsumees of “Reason for Admission” are more likely to be defined incorrectly by themselves.

A drawback of DL classifiers is that stated taxonomies “collapse” for subsumees of an unsatisfiable concept, i.e. all subsumees of an unsatisfiable concept are unsatisfiable as well. This makes it hard to detect which unsatisfiable concept is the most generic in the stated taxonomy. In the example from Figure 4.8 both ErroneousHepatitis and ViralHepatitis are unsatisfiable, the stated subsumption is lost as they are rendered equivalent. Therefore, apart from the inferred taxonomy, the stated taxonomy is useful for finding the top-most unsatisfiable concepts. It is noteworthy that OilEd retains the stated taxonomy, facilitating the process of finding the top-most unsatisfiable concepts. This is shown in Figure 4.9. auto-immun_haemolitische_anaemie_1046 is unsatisfiable because it is subsumed by haemolitisch_anaemie_1045, which is also unsatisfiable.

For each unsatisfiable concept that neither is subsumed by an unsatisfiable concept nor has an unsatisfiable existential role value, the underlying cause of unsatisfiability needs to be determined. Examples of such causes are: being subsumed by mutually disjoint concepts, or having role values that contradict those of a subsumer. Once the cause of the unsatisfiability is detected, it can be determined whether it stems from assumptions in the migration process, or from an actually incorrect or inconsistent definition.
4.4.3 Advanced error detection

As mentioned above, DL classifiers do not provide any information other than the sheer fact that a set of concepts is unsatisfiable. Methods for more advanced support for error tracking are being developed. First, this research focuses on concepts that are “involved” in definitions of many unsatisfiable concepts. Second, investigations are ongoing on how to pinpoint the parts of a concept definition that render a concept unsatisfiable. For example, consider the definitions (2) and (3) for concepts A and B in Figure 4.10. Concept B is unsatisfiable due to the universal quantification for R in the definition of A and the existential quantification for R in the definition of B (for example: all children are boys, some child is a girl). In this case, unsatisfiability of B can be explained by the simplifications in (4) and (5). As this currently has to be determined by manual review of the concept definitions, methods for automatic pinpointing the location of errors are being developed [12].

4.5 Case Study Results

4.5.1 Application of the methods to DICE

We have applied the method described above to DICE, in order to gain insight into the feasibility of this approach. The DICE knowledge base consists of about 2500 concept frames, with over 3000 defined slot-fillers (for other slots than “is a”). We used the reasoner RACER to process the DL-based representation.
of the knowledge base, in order to check the coherence of the TBox.

For each unsatisfiable concept, the first author (RC) determined whether the unsatisfiability stemmed from reference to another unsatisfiable concept, from an incorrect assumption during the migration process or from a modeling error. We dealt with incorrect assumptions by utilizing the facet to overrule default interpretation of role quantification (described in Section 4.3.3). If an actual modeling error was detected, the frame-based representation was corrected. After changing the frame-based representation, the migration process was repeated, and a new DL-based representation emerged iteratively.

Below we will make a distinction between unsatisfiability introduced by the migration method, and unsatisfiability caused by modeling errors. The analysis presented here is specific for the DICE knowledge base, and results may be significantly different for other TSs. It does however provide insight in the possibilities and merits of using our method.

4.5.2 Unsatisfiable concepts caused by the migration method

The stringent assumptions put on the frame-based representation resulted in two major types of assumption errors: errors caused by incorrect assumption of disjointness, and errors caused by incorrect assumptions on quantification. Disjointness errors were found in the descendants of etiology. For example, the assumption was made that the sibling concepts addictive drug and analgesic are disjoint, which is false as an analgesic may or may not have the property of being addictive. The disjointness rendered Morphine and Opioids as unsatisfiable, as it is (correctly) defined as a descendant of both. This unsatisfiability could be overcome by removing the assumption of disjointness. We have currently chosen not to perform serious remodeling toward reducing multiple inheritance as much as possible (as will be discussed in Section 4.6). This may however be considered in further evaluation efforts.

A large number of unsatisfiable health problems were found, which could be explained by the stringent assumptions posed on the quantification of roles. Universal role quantification was frequently falsely assumed, mainly in regard to generic concepts, such as for example lung disease for which the location was (falsely) assumed to be restricted to lungs. This led to unsatisfiability of all diseases that were correctly defined as a lung disease, but that also involved a location different from lungs, such as Cardiac Pulmonary Oedema. In these cases, the frame-based representation was altered by use of the slot facet to mark slot-fillers as “not universal”.

4.5.3 Unsatisfiable concepts due to incorrect definitions

Various types of modeling errors were found in the process of migration. We categorize these errors under one of the following categories: missing or incorrect slot-fillers, incorrect slot-facets, incorrect taxonomic placement.
Tab. 4.2: Various types of changes made and their number of occurrence for the DICE knowledge base. The DICE knowledge base contains about 2500 concepts, 2500 is_a slot-fillers, and over 3000 defined slot-fillers (for other slots than “is_a”).

<table>
<thead>
<tr>
<th>Type of change</th>
<th>Occurrence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration assumption changes:</td>
<td>135</td>
</tr>
<tr>
<td>Slot-filler changes (for missing or incorrect slot-fillers):</td>
<td>84</td>
</tr>
<tr>
<td>Slot-facet changes (for incorrect slot-facets):</td>
<td>24</td>
</tr>
<tr>
<td>Classification changes (for incorrect taxonomic placement):</td>
<td>13</td>
</tr>
</tbody>
</table>

**Missing or incorrect Slot-fillers**

84 concepts were found with missing or incorrect slot-fillers. These can be categorized as follows:

*overriding instead of additional slot-fillers* The first typical situation encountered was the need to create additional slot-fillers at subframes. Recall that assumption (4) in Section 4.3.2 states that slot-fillers override any fillers defined by superclasses. Hence it is required to repeat these fillers for subframes that have additional fillers, otherwise the specified filler will be interpreted as an overriding value, potentially leading to inconsistent concepts. For example, in the frame Diabetes Insipidus the slot-filler endocrine system overrides the inherited metabolic system, instead of being an additional system. Hence, diabetes insipidus should also involve the metabolic system, stating system: (endocrine system, metabolic system). Whereas this need to repeat fillers for subframes might seem to introduce overhead, the advantage of this solution is that it makes explicit the fact that additional fillers are mentioned. Especially as frames generally allow for overriding slot-fillers, repetition ensures that distinction can be made explicit between overriding and additional slot-fillers.

*too limited specification of subsumers* Another typical case was related to slots with an “OR” facet, indicating slot-fillers that imply a selection to choose from. Frequently, the list of specified slot-fillers was missing some fillers that were defined for some of the subclasses. In such cases, these slot-fillers were added to the slot of the superclass.

*incorrect or redundant slot-fillers* Apart from missing slot-fillers, there were a number of cases where slot-fillers were incorrectly or redundantly defined. The redundant definitions were not detected by the reasoner, but emerged when scrutinizing concept definitions. An example of a redundant definition is Extrudal Haematoma, which was defined as a Intracranial haemorrhage, with abnormality haematoma. Specification of the abnormality is however redundant, as the abnormality haematoma pertains to the defini-
tion of Intracranial haemorrhage. Currently, we are devising methods for automated retrieval of redundant definitions.

Incorrect Slot-facets

24 specifications were found where the value of the refinability facet (“DEF”, “OR” or “XOR”) was not (correctly) specified (see Table 4.1). This means that specifications were unjustly considered part of a definition, whereas they are actually part of possible refinement. Although this is rather specific to representation of the DICE knowledge base, it is worth further study, as in generic frame representation there is no explicit distinction between definitional and refinable parts of a frame.

Incorrect taxonomic placement

As shown in Table 4.2, 13 concepts were found that were placed wrongly in the taxonomy. These misplacements can be categorized as follows:

coordination instead of subordination There were cases of concepts that were placed as siblings where one of the concepts should have been subordinate to the other (i.e. be its child). For example, self-poisoning was defined as a sibling of intoxication, whereas self-poisoning should be subsumed by intoxication.

subordination instead of slot-filler Another example of misclassification is illustrated by a concept that was defined as both a health problem and an abnormality, which are disjoint. Instead of being subsumed by abnormality, it should have been related to abnormality by a slot-filler. The anatomy taxonomy also revealed 6 concepts that were defined as being subsumed by a concept instead of having a part-of relation to that concept. For example, brain was defined as subsumed by nervous system, instead of being part of the nervous system. This type of error has been found in other systems as well, and DL reasoning provides a powerful means for detecting it [15].

subordination instead of superordination One part of the hierarchy was defined incorrectly by switching subsumers and subsumees. This involved the concept laryngo tracheo bronchitis which was defined as the subsumer of laryngitis, tracheitis and bronchitis, whereas it should be defined as a subsumee of these three concepts.

4.6 Discussion

We have described a method for detecting errors in a frame-based TS by means of making a DL-based interpretation of it. We have applied this method to a medical TS, DICE, and found a number of errors in this TS. One can not assume however, that a knowledge base that results from the migration process
holds no erroneous concept definitions if no unsatisfiable concepts are found. Concepts may be semantically incorrectly defined, though logically satisfiable. For example, if *Viral Meningitis* would be simply defined as *Hepatitis* (instead of *Meningitis*) *is caused by* a *Virus*, this could result in a satisfiable concept, although it is obviously incorrect. Moreover, application-specific slots or facets, of which the semantics are unclear or non-definitional, cannot be represented using DL. This means that these elements (such as the facets to support post-coordination that allows for the creation of new concepts based on combining existing ones) are lost in the process of migration. Therefore, parts of the functionality provided in the original frame-based representation will have to be realized outside of the DL-based environment. Although this seems to be a drawback at first, it may well turn out to be advantageous as it leads to better understanding of the various aims for which knowledge modeling is being performed.

As some contemporary medical TSs are claimed to be based on DLs, it makes sense to assess the applicability of our methods to these systems. To our knowledge the expressiveness of the DLs used in these systems is very limited, as they only allow for existential quantification, not for universal quantification. This simplifies reasoning, but limits the possibilities of consistency checking. Actually, due to the limited expressiveness, these systems’ representation very much resembles a frame-based representation, apparently making the methods that we describe in this paper also applicable to such “DL-based” systems, which can be migrated to a more expressive DL such as the one we use. Future research is needed to prove this.

### 4.6.1 Balance between effort and effect

The current migration process is still labor intensive, as it requires numerous adaptations to the default interpretation of frame-based descriptions. The changes made are quantified in Table 4.2. This table shows that about 50% of the changes involve the default assumptions made in the migration process. Although this can be regarded as a large overhead, it also contributes to the process of making semantics more explicit. This indicates that this migration process is beneficial to the frame-based representation, especially when semantics are made explicit in the frame-based representation. For example, the facet that specifies whether slot-fillers present a restrictive set of values (comparable with universal quantification in DL) or just a (non-restrictive) set of most common values provides added value to modeling.

Hence, the impact on improving the model is considerable. Additionally, as these methods audit the model, they help in providing confidence in the contents of the model. Work is ongoing to significantly reduce the effort to perform such an assessment. Methods are being developed for pinpointing the modeling errors, and for explanation of unsatisfiability. When such methods can be applied, the impact of the process is expected to easily outweigh the effort it takes.

It should be clear that there has not been a comparison to a “gold standard”
of errors in the knowledge base. The absolute numbers of changes made are small relative to the total number of links in the knowledge base. There were 13 classification changes, in over 2500 \textit{is\_a} links, and a total of 108 slot-filler and slot-facet changes in over 3000 non-\textit{is\_a} links. Rather than being the sole auditing method, this migration method is complimentary to other methods, such as the ones described in Section 4.1.2.

4.6.2 Observations from the case study

During the process of error detection and resolving, a number of issues came to light that require further investigation. We only have made changes needed to resolve inconsistencies in the original knowledge base. However, by studying the definitions involved, some cases were found for which a more rigorous redefinition would be justified. Also more attention should be paid to the computational properties associated with the resulting TBox. Below, we will discuss four modeling issues: entry terminology concepts, patterns, the difference between definitions and “templates”, and the properties of the TBox.

**Entry Terminology.** As mentioned earlier, a frame-based representation requires classes to be explicitly defined as subclasses of all superclasses involved. As DLs make inference possible on superclasses, it can be argued whether a better way of modeling would be to define concepts based on their actual properties, with as little as possible explicit subsumption of non-primitively defined concepts, as this subsumption can be inferred. For example, \textit{hepatitis} could be defined as a “disease, being located in the liver” instead of as a \textit{liver disease}, because the latter can be inferred from the definition of hepatitis. This observation indicates that it may be sensible to create a knowledge base that consists of concepts that are defined “ex genus et differentiae”, and adhere to the “jointly exhaustive and pairwise disjoint” (JEPD) property, which is considered important [27]. So-called entry terminology concepts (such as \textit{liver disease}), which can generally be non-primitively defined, can be defined on top of a maximally monohierarchical knowledge base, providing convenient concepts for users, and resulting in a polyhierarchical knowledge base. This approach has also been advocated in the GALEN project [28], and appreciated for increasing the comprehensiveness of the resulting model [26][29].

**Patterns.** Other concepts were found that indicated non-uniform modeling rather than incorrect definition of concepts. For example, both a “\textit{is\_part\_of}” relation and the concepts \textit{body part} and \textit{organ part} are present in the knowledge base. This makes it possible to define a concept by means of either “\textit{is\_a organ part}” or “\textit{is\_part\_of organ}”. Whereas being an \textit{organ part} can be inferred when a concept is defined as part of a (specific) organ, this need not be represented explicitly while modeling a knowledge base. Development of guidelines or modeling patterns can encourage consistent modeling.

**Definitions versus Templates.** Although we have treated frame-based descriptions as purely definitional, actually “a Frame is a collection of questions to be asked about a hypothetical situation; it specifies issues to be raised and methods to be used in dealing with them.” [4]. Hence, frame-based descriptions
of classes provide a mix of definitions and templates for questions. Although it can be argued whether or not this information should be included in a DL-based representation, it helps to determine whether templates are consistent with definitions. It must be clear that the knowledge base generated from the frame-based representation is for the purpose of detection of inconsistencies, not necessarily for use in (clinical) practice. However, the improved frame-based knowledge base can be such a practically usable knowledge base.

**TBox properties.** The language that was used for the DL-based representation is $\mathcal{ALCQ}$, which consists of the constructors $(\cap)$, $(\cup)$, $(\neg)$, $(\exists)$, $(\forall)$, $(\geq)$, and $(\leq)$. As we have represented anatomy using SEP triplets, no role hierarchies or transitive roles were required, keeping the language relatively simple. As the frame-based representation did not contain any axioms other than frame-definitions, and no cycles, the migration resulted in an unfoldable TBox. This means that all definitions are simple (defining only atomic concepts), unique (only one definition for each atomic concept exists), and acyclic (meaning the definition of a concept has no reference to the definiendum, either directly or indirectly). Reasoning on this type of TBox generally has a lower complexity than reasoning on arbitrary TBoxes with cycles and general concept inclusion axioms [20].

### 4.7 Conclusion

We have devised a method for the semi-automated migration from a frame-based representation to a DL-based representation and demonstrated how it helps focusing on definitional imperfections of a medical TS. The rationale behind the development of this method is to provide a formal basis for representation in order to facilitate automated reasoning, which would not be readily feasible in a frame-based representation. Automated reasoning is generally much cheaper than manual inspection and redefinition of a medical TS and provides an objective source for creating confidence in the correctness of its contents.

The migration into a DL-based representation relies on posing assumptions on the semantics of the frame-based definitions. In our approach one initially starts with introducing stringent assumptions about concept definitions in the DL-based representation. This is meant to allow the reasoner to perform inference that can indicate potentially incorrect or incomplete definitions. The utility of the approach has been demonstrated in a real world case-study revealing modeling errors in an intensive care TS.

Admittedly, the DL-representation would include a large number of too stringent assumptions. These assumptions are mainly concerned with the universal quantification and disjointness. However, the approach provides an automated reasoning tool to identify areas for focusing human attention. Still, a weakness of our approach is that there is no support for tracing or explaining DL-based unsatisfiability. As a consequence, pinpointing and resolving conflicts in definitions is a time-consuming task. Working on explanation facilities comprises important further work that we are planning to address.
With the advent of DL-based medical terminological systems, the method we have provided here is also useful for more interactive support during the maintenance of such systems. Whereas we have now used them to assess the quality of a TS independently from the maintenance process, research is ongoing to determine how these methods can be integrated into the maintenance process in an interactive manner.

Inconsistency is not the only modeling error that is likely to occur. Concepts can inadvertently be duplicately defined, i.e. using different concept names that should be represented as synonymous terms for a single concept (concept redundancy). Another modeling error that should be detected is redundant definition of concepts, i.e. a definition of a concept that contains conditions that are also inherited from a concept’s ancestor. Research is ongoing to explore the possibilities of using equivalence to detect such errors. Another issue that is being studied is the detection of “underdefinition” [30]. This research aims at improving TSs by encouraging definition of as many necessary conditions for concepts as possible.

Generally, when dealing with errors, there must be a gold standard or a guideline describing the meaning of correct modeling. Although these issues are being studied, e.g. [31], we are unaware of their application in real practice. Further research will address the issues of the underlying assumptions and/or standards that distinguish correct modeling from erroneous modeling. The added value of the method described in this paper is that it not only provides some modeling guidelines, but also a means to evaluate the compliance to these guidelines in a formal manner. This method reveals potential modeling inconsistencies, helping to audit and (if possible) improve the medical TS. In this way, it contributes to providing confidence in the contents of the terminological system.

Acknowledgement

A preliminary version of this paper has been presented at AIME03 [32].

The authors would like to express their gratitude to the reviewers of the original version of the manuscript for their thorough reviews. Their comments and suggestions have helped improving the clarity of this paper.

This work has been partially funded by the Netherlands Organization for Scientific Research (NWO) program “Information & Communication Technology in Healthcare” (ICZ) for the project entitled “Terminology and Semantics: Making semantics explicit”, number 014-18-014.

Bibliography


