

Logical Support for Terminological Modeling

Stefan Schlobach^a, Ronald Cornet^b

^a Language and Inference Technology, ILLC, Universiteit van Amsterdam

^b Academic Medical Center, Universiteit van Amsterdam

Abstract: Terminological modeling, in particular in medical domains, is difficult and has enjoyed growing attention over the recent past. Logical reasoning often plays a fundamental role to support the modeling process. Nevertheless, the development of the logical tools is usually driven by computational or theoretical criteria rather than by the direct needs of a modeler. In this paper we attempt to shift this balance and discuss a number of logical reasoning services to support the modeler of a (medical) terminology to construct a logically correct, complete and concise terminology. This modeling support is logical as it is based on reasoning services with respect to formally defined semantics. Practical results of this systematic discussion are formal definitions of several new reasoning tasks.

Keywords:

Controlled Terminologies, Knowledge Representation

Introduction

Our work was motivated by the development of the DICE terminology. DICE implements frame-based definitions of diagnostic information for the unambiguous and unified classification of patients in Intensive Care medicine. This representation of DICE is then migrated to the Description Logic (henceforth DL) ALC to facilitate logical inferences. In [1] the authors describe the migration process in more detail. The resulting DL terminology (usually called a “TBox”) contains axioms, such as those shown in Figure 1 defining, e.g., hepatitis as a liver disease located in the liver (and possibly somewhere else) which is either drug- or alcohol related, or caused by inflammation or an infection. Furthermore, the involved tract is the digestive system. The TBox in Figure 1 contains a number of modeling flaws, which we will discuss in this paper. We added a few errors to illustrate some of the problems and to explain the respective reasoning services, but most were actually detected while we worked on improved versions of DICE.

This TBox T will be our running example and contains a number of modeling errors, which cannot be identified using the traditional logical reasoning services such as satisfiability. A Description Logic concept is said to be *satisfiable* if it is not necessarily interpreted as the empty set. In the recent past, a number of highly optimized reasoning systems have been developed, which can efficiently check satisfiability of a concept even with respect to very large terminologies. One of these systems is RACER [2]. However, checking for satisfiability or the other reasoning services implemented in RACER might not be sufficient to discover all crucial modeling flaws. Take as an example the concept viral hepatitis, which could be caused by the Cytomegalo virus according to our definition. This, however, contradicts the information that an infective hepatitis is caused either by the Epstein-Barr or the Hepatitis virus, or Amoeba, Spirochaetes or Toxoplasm. We will say that the implicitly defined concept of an infective hepatitis caused by the Cytomegalo virus is *locally unsatisfiable*. Unfortunately, such contradictions are not detected by RACER or other logical systems.

Local unsatisfiability is a relatively typical example: when building a terminology we try to construct concepts that correctly model our intention. If infective hepatitis cannot be caused by the Cytomegalo virus, our terminology T must be erroneous, and we look for tools that can detect such modeling errors automatically. Although local unsatisfiability is relatively straightforward property, there are currently no reasoners who detect this and other types of modeling flaws. What can we do? We will answer the question systematically, by first studying the requirements a *correct* model should fulfill, and then discussing logical reasoning services to help enforce the correctness of a terminology with respect to these requirements. In the main part of this paper we will investigate three questions: *Is my model sound? Is my model complete? Is my model concise?*

$Liver \sqsubseteq \exists part.DigestiveS \sqcap BodyPart \sqcap \exists region.Abdomen \sqcap \forall region.Abdomen$
$LiverDisease \sqsubseteq \forall tract.DigestiveS \sqcap \exists tract.DigestiveS \sqcap MedicalDiagnosis \sqcap \exists located.Liver$
$Hepatitis \sqsubseteq \exists tract.DigestiveS \sqcap \forall tract.DigestiveS \sqcap \exists located.Liver \sqcap LiverDisease \sqcap \exists abnormality.(Drug \sqcup Inflammation \sqcup Infection \sqcup Alcohol)$
$InfectiveHepatitis = Hepatitis \sqcap \forall etiology.(EBVirus \sqcup HVirus \sqcup Amoeba \sqcup Spirochaetes \sqcup Toxoplasm) \sqcap \exists abnormality.Infection \sqcap \forall abnormality.Infection$
$ViralHepatitis = \forall etiology.(HVirus \sqcup EB-Virus \sqcup CytomegaloVirus) \sqcap InfectiveHepatitis$
$Kidney \sqsubseteq \exists part.GUSystem \sqcap \exists part.ESystem \sqcap BodyPart \sqcap \exists tregion.Abdomen \sqcap \forall region.Abdomen \sqcap \forall side.(Left \sqcup Right)$
$KidneyDisease \sqsubseteq \exists located.Kidney \sqcap \forall located.Kidney \sqcap MedicalDiagnosis$
$Nephropathy \sqsubseteq \forall tract.GUSystem \sqcap \exists tracGUSystem \sqcap \exists located.Kidney \sqcap \forall located.Kidney \sqcap MedicalDiagnosis$
$GUSystem \sqsubseteq Tractus$ $ESystem \sqsubseteq Tractus$ $DigestiveS \sqsubseteq Tractus$ $MedicalDiagnosis \sqsubseteq ReasonForAdmission$

Figure 1 An example TBox T

It is well known that the standard reasoning services of satisfiability and subsumption checking are computationally difficult problems even for very simple modeling languages [3]. To allow scalability to real-world medical terminologies most research efforts were invested to improve efficiency of logical reasoning and to investigate the trade-off between expressivity and tractability. Nowadays, a number of highly optimized reasoning systems exist, and the interest of logical modeling support can shift from making systems more expressive and efficient to further increasing the functionality with currently unknown reasoning services. With this paper we advocate a systematic investigation of the potential and requirements of logical modeling. Following this analysis we will introduce a number of new reasoning services to support logically correct modeling: local unsatisfiability, non-atomic subsumption, modularization, similarity, exhaustiveness, atomic grouping and similarity extensions. According to our formal approach we define these reasoning services formally, although it is out of the scope of this paper to go into too much detail.

Medical Modeling

Building a medical terminology is a time-consuming and error-prone process. This has a number of reasons. It requires expert knowledge and thorough understanding of the modeling methodology being used (if any). Often, modeling is a continuous effort, covering a large period of time, and involving a considerable number of people. This makes it extremely difficult to recollect previous modeling activities. The size of current medical terminologies, which typically contain up to hundreds of thousands of concepts, renders them beyond comprehension. To assist knowledge modelers in the process of terminology development, tools have been developed within each of the projects involving development of large medical terminologies. For example, in the GALEN project, the Classification Workbench has been developed [4] and SNOMED uses the Terminology Development Environment (TDE) that was developed by Apelon [5]. Also in the DICE project, a modeling environment has been developed, that allows knowledge modelers to define concepts, and to specify how concepts can be further qualified, but also supports interrelation of concepts and terms. A notable example of a generic modeling environment that is under development is Protégé [6]. What these environments have in common is that they aim to support Frame-based or DL-based modeling. Throughout this paper we will take Description Logic based reasoning as our canonical example for logical modeling. This is because Description Logics have been very popular representation mechanisms in medical terminologies, as they provide expressive languages with highly optimized tools, and facilitate object-oriented hierarchical modeling with explicit definitions. A further reason for choosing Description Logics is that this paper is the result of a systematic analysis of our own modeling experiences with the DICE terminology and its corresponding DL TBox. We will formally introduce DLs later but start with an informal discussion of some

requirements for good modeling, with some respective reasoning tasks to support this modeling.

What is a good terminology?

The purpose of this paper is to provide a systematic investigation of what logical reasoning can offer to improve a terminological modeling process. What do we expect from a correct terminology? In the following we will introduce three main criteria: *soundness*, *completeness* and *conciseness*, and informally discuss the relevant logical criteria mostly by referring to modeling flaws in our example in Figure 1.

Is my terminological model logically sound?

Advocating a domain-independent approach corresponds to assuming that the individual concepts in the terminology are modeled by domain-experts in a knowledgeable way, and that it is the structural complexity of a terminology and the implicit relations between concepts that are difficult. **Soundness is then the fact that the (possibly very complex) structure correctly represents the modeler's intentions.** Logical criteria can help: first, a terminological model must be **logically consistent** to be logically sound. The most common criterion for logical inconsistency is *unsatisfiability* of one or more defined concepts, and we have already discussed *local unsatisfiability* as a more fine-grained non-standard reasoning task. The second logical criterion to ensure soundness of the model is that **implicit information** must be **verifiable**. This means, that both the explicit and implicit structure is correct, i.e. correspond to the modeler's intention. The hierarchy induced by the classical *subsumption relation* between concept-names makes such a structure explicit. On the other hand, there is more implicit knowledge that needs to be verifiable, such as the fact that the Kidney is the system part of some tract. As there is no explicit definition of a concept "something with a tract", this type of *non-atomic subsumption* cannot be detected by standard reasoners. To summarize, checking for soundness means to check whether

1. the concepts in our model are globally and locally consistent, and whether
2. implicit information is verifiable or not.

Is my model logically concise?

Assuming that the model is logically sound, we can address the explicit structure of the terminology itself. **Conciseness is then the fact that the represented structure is not more complex than necessary.** From a logical perspective, there are two criteria that can help to model in a concise way. First we need to ensure **structured modeling**, which is related to the *modularity* of a TBox and the modeling of similar concepts. Often terminologies, such as DICE, are built in a modular fashion, e.g., one starts with modeling the anatomy and etiology, and then uses these modules to model the reasons for admissions. If this is not the case, though, there should be a mechanism to extract modules automatically from a given terminology. In our example TBox of Figure 1 body-parts, reasons for admission and tracts are modeled independently from each other, but the corresponding modular

structure is not made explicit by any of the current logical reasoner.

As well as being modularly structured we require our terminology to model similar concepts in a similar way. *Similarity modeling* therefore tries to ensure that related definitions are modeled in a structurally similar way. Consider the definitions for liver and kidney in our terminology T. Both are body-parts located in the abdomen differing in their systemic part only. However, we know that kidneys have a left or right side and the question could be suggested to a modeler whether it makes sense to define the side of a liver. Medical terminologies are of little use if they cannot be **presented by and used by medical experts** in an understandable way. Logical modeling neither deals with graphical nor with natural language interfaces but can help to structure the information and to reduce *redundancies*. In our example TBox T infective hepatitis is defined redundantly as hepatitis where there must be an infection as abnormality. This, however, is already implied by the fact that any hepatitis has an abnormality, and that all abnormalities of an infective hepatitis must be infections. To summarize, conciseness can be supported by logical tools checking whether the concepts in our model are

3. systematically structured, and
4. easily representable to domain experts.

Is my model logically complete?

The third requirement to a logically correct model is that it is complete. **Completeness of the model means that all the relevant information is modeled.** Completeness needs to be discussed with respect to the concepts that are defined, or that should be defined. There are two obvious questions: are the formalized concepts exhaustively defined, and are all the concepts that I would like to define formalized in my model? Let us discuss **concept exhaustiveness**. Assume that an axiom $Appendix \sqsubseteq \exists part.DigestiveS \sqcap BodyPart \sqcap \exists region.Abdomen \sqcap \forall region.Abdomen$ is introduced to our terminology T. Notice that *Appendix* and *Liver* are now defined in precisely the same way, which points to an under-specification of one or both of them. The second logical support is **terminological completeness**, i.e. to ensure that every concept that should be defined is indeed in our terminology. Human experts know for example that *HVirus*, *CytomegaloVirus* and *EBVirus* belong to a particular class of concepts, namely the viruses, but there is no concept in our terminology T which groups the three together. Formal mechanisms to support terminological completeness will be discussed based on *concept grouping* and *similarity extensions* later. Summarizing, to ensure logical completeness we want to develop formal criteria to check

5. whether the concepts are exhaustively defined, and
6. whether the concept space is completely covered.

Logical Modeling Support

Following the systematic requirement analysis of the previous section we now want to make the respective reasoning processes formally precise. The tasks will be based on the formal semantics of Description Logics, and we shall briefly introduce some necessary concepts.

Description Logics. We will not give a formal introduction to Description Logics here, but point to the first two chapters of the DL handbook [6] for an excellent overview. Briefly, DLs are set description languages with concepts interpreted as subsets of a domain, and roles which are binary relations. ALC is a simple yet relatively expressive DL with conjunction $C \sqcap D$, disjunction $C \sqcup D$, negation $\neg C$ and universal $\forall r.C$ and existential quantification $\exists r.C$. We also include \perp and \top as the empty and universal concept for modeling reasons. An interpretation maps concept-names to subsets of a domain, roles to binary relations and extends over the operators in the obvious way.

DL Terminologies. In a terminology T (called TBox) the interpretations of concepts can be restricted to the *models* of T by *axioms* of the form $C \sqsubseteq D$ or $C = D$. Two TBoxes are *equivalent* if they have the same models. A concept-name is called *primitive* if it is not defined, i.e., occurs on the right-hand side of the axioms, only. In our TBox T in Figure 1 the concepts *Virus* and *Left* are primitive. The non-primitive concept-names, such as *Liver* or *Hepatitis*, will be called *defined* concepts. An interpretation I that only interprets the primitive concepts is called a *base interpretation*. An interpretation I' is an *extension* of I if it also interprets the defined concepts, if it has the same domain as I, and if it agrees with I on the base symbols.

Logical Soundness

Logical soundness of a terminology requires *consistency* and *verifiability of the implicitly modeled information*. Let us study the related reasoning tasks one-by-one.

Consistency. Checking whether there are models satisfying a terminology and individual concepts are well studied reasoning tasks in DL research [8]. Based on the model-theoretic semantics concepts can be checked for *unsatisfiability*: whether they are necessarily interpreted as the empty set in all models of a TBox T. T is called *coherent* if all concept-names occurring in T are satisfiable. The TBox of our introductory example is, e.g., coherent because all concepts are satisfiable. But the question of correctness of the model goes deeper: are all the parts of definitions consistently defined, or are there local inconsistencies? Standard unsatisfiability corresponds to equivalence of a concept-name with the \perp concept. This observation can be generalized. Remember, the introductory example of an infective hepatitis caused by the Cytomegalo virus. Formally, the concept $InfectiveHepatitis \sqcap \forall etiology.CytomegaloVirus$ is only satisfiable if a viral hepatitis has no etiology, i.e., the concept *CytomegaloVirus* can be replaced by \perp without changing the meaning of the TBox. Let us make the notion more precise.

Let $T[C' \rightarrow \perp]$ denote a TBox where an instance of a sub-concept C' of C has been replaced in an axiom $A \sqsubseteq C$ of T by the \perp concept. The TBox T is *locally incoherent* if there is a concept C' such that $T[C' \rightarrow \perp]$ is equivalent to T . The concept C' is then called *locally unsatisfiable*.

Verifiability of implicit information. Consistency of a terminology was the first requirement for logical correctness. Classification, to make the induced structural properties of the terminology explicit, is the other “standard” task in terminological reasoning systems. Formally, *subsumption* of two concepts C and D in a TBox T is a subset relation of $I(C)$ and $I(D)$ with respect to all models I of T , and is usually denoted by $T \models C \sqsubseteq D$. A human modeler can now check whether the implicit hierarchy corresponds to the intended one. In our example it follows, e.g., implicitly that nephropathy is a kidney disease. Hierarchical information between concept-names is not the only information that can be derived from the terminology. Remember that we can derive that a kidney is part of a tract from our example TBox T , i.e. that $T \models \text{Kidney} \sqsubseteq \exists \text{part.Tractus}$. From a formal point of view, calculating whether such a *non-atomic subsumption* relation holds corresponds to standard subsumption, and can be reduced to satisfiability. The challenge of this reasoning service is to evaluate criteria, based for example on generality, size or reasoning complexity, and to automatically choose subsumption relations that could be relevant to a human modeler.

Conciseness

A formally sound terminology is not necessarily useful as long as it is not systematically structured and representable to human experts. The perfect modeling structure differs from domain to domain, but there are some general logical criteria to analyze the modeling structure. Let us study *presentation of information* and *structured modeling* in more detail starting with the latter.

Structured Modeling. We will present two approaches to automatically extract modules from a given terminology. In the first, a module is a set of hierarchically related *concepts*, in the second a set of *TBox axioms*, which are semantically independent from the rest of the terminology. In our terminology T there are three disjoint sub-hierarchies, the concepts subsumed by *Bodypart*, *ReasonForAdmission* and *Tractus*, respectively. Common to the elements of a module is that they are all subsumed by a single concept, and that they are not related to any concept in another module. Let us define this formally: given a TBox T , a set S of concept-names occurring in T is called a *hierarchical module* if there is a concept $C \in S$ such that $T \models C' \sqsubseteq C$ for any element $C' \in S$, and where $T \not\models D \sqsubseteq C$ for any concept-name $D \notin S$. Alternatively, we can define modules of axioms directly using the formal semantics of terminologies. Semantic modularity means that we have no means to change the interpretation of the concepts defined in the module from outside the module. Formally, a sub-TBox T' is a *semantic module* of a TBox T if the

extensions of the base interpretations for the concepts defined in T' remain the same when other axioms from T are added.

Not only should a terminology be modeled modularly, but also should similar concepts be modeled in a similar way. From a logical perspective we need a formal notion of similarity to enforce this requirement. The simplest criterion can be based on membership in the same module, possibly using subsumption as an additional criterion for *modular similarity*. An alternative criterion for similarity makes use of the semantic structure of the terminology. As a simple example replace the systemic parts in the definitions of the concepts *Liver* and *Kidney* with their immediate subsumer (the concept *Tractus*), and the first is subsumed by the second. This example suggests a simple definition of similarity. Let C and D be concepts occurring in a TBox. We say that C and D are *hierarchically similar* if $T \models C' \sqsubseteq D'$ or $T \models D' \sqsubseteq C'$, where C' and D' correspond to C and D , but where some concept-names have been replaced by their immediate super-concepts w.r.t. subsumption in T .

Presentation of information. How to present information is a question strongly linked to the structural properties of the model. A modular terminology is usually more intuitive than an unstructured, and presentation of similar concepts might help to understand the model by analogy. One further aspect is to reduce redundancies when presenting concepts to domain experts. Similarly to local unsatisfiability redundancy can be defined as a simple check whether the modeler uses a complex concept to express the most general concept \top . Formally, we define redundancy as follows. Let $T[C' \rightarrow \top]$ denote a TBox T where a single sub-concept C' of C has been replaced in a single axiom $A \sqsubseteq C$ of T by the concept C' . A concept C' is *redundant* in T if $T[C' \rightarrow \top]$ is equivalent to T . Alternatively, we can consider a notion of redundancy where a TBox is checked for equivalence with a TBox where a concept is replaced by a super-concept according to the subsumption hierarchy instead of by \top .

Logical Completeness

For logical completeness we have to ensure that all relevant information about the given concepts is modeled and, secondly, that every useful concepts is indeed specified in our terminology. The first requirement was called *concept exhaustiveness*, the second one *terminological completeness*.

Concept Exhaustiveness. The simplest mechanism to support concept exhaustiveness is *similarity-based*. Look at the definition of *Hepatitis* and *Nephropathy* in the TBox T . Both concepts are “hierarchically similar”, i.e., they are elements of a common module, and have super-concepts *LiverDisease* and *KidneyDisease* which are, again, similar. There is no information about abnormalities of a nephropathy, and a hepatitis is not restricted to be located in the liver. Both differences are important as they point to possible under-specifications. A simple formal method to support concept exhaustiveness is therefore to highlight structural differences between similar concepts within a terminology. Technically,

this is straightforward; we just need to choose a formalization of similarity and to define a structural relation between concepts.

Terminological Completeness. The modular structure of the model, whether intended or extracted, can also be used to propose other potentially useful axioms. A simple notion of *similarity extension* suggests to extend the TBox T also by a concept *PancreasDisease* if the concept *Pancreas*, formally similar to *Liver*, is added to our TBox T . Even more, we can also suggest an axiom $PancreasDisease \sqsubseteq \exists located.Pancreas \sqcap MedicalDiagnosis$ covering the minimal common properties of the medical diagnosis related to liver and kidney diseases. Similarity extension requires a formal definition of covering similar to the one in [9]. This, however, is an open problem in the presence of disjunction in our representation language ALC.

The second logical way to support completeness was concept grouping. Remember that the concepts *HVirus*, *EBVirus* and *CytomegaloVirus* belong to the class of viruses, but there is no concept in our terminology T grouping the three together. A simple method to define such *grouped concepts* is to study co-occurrences of concepts in atomic disjunctions. This would imply that one should introduce a super-concept *Virus* for *HVirus*, *CytomegaloVirus* and *EBVirus*, and a super-concept *MicroOrganism* for *EBVirus*, *HVirus*, *Amoeba*, *Spirochaetes* and *Toxoplasm*. If one corrects the definition to include the Cytomegalo virus to the etiology of infective hepatitis all viruses are now also microorganisms, which suggests a new axiom $MicroOrganism \sqsubseteq (Virus \sqcup Amoeba \sqcup Spirochaetes \sqcup Toxoplasm)$ for the terminology T in Figure 1.

Conclusion

The difficulty of clinical modeling, and the respective requirements for terminological modeling from a medical perspective have been discussed extensively in papers such as [10] and [11]. Our paper begins where their work ends, as we assume that our formalization language has been chosen to make it most suitable for the task at hand, i.e. that it allows, in principle, to build a good medical terminology. Some logical requirements described in this paper, such as redundancy or the verifiability of implicit information, coincide with Cimino's and Rector's criteria. Nevertheless, we argue more abstractly as we study formal, domain-independent criteria, in order to define concrete reasoning tasks within the framework of the representation languages we use to model DICE.

We are currently developing algorithms for the new reasoning services, in order to integrate them into the RICE modeling tool. RICE supports terminological modeling by visualizing the standard Description Logic reasoning facilities of RACER, but is currently extended to perform new reasoning tasks, such as to explain formal reasoning to non-expert users [12] and [13]. We are currently developing algorithms for the reasoning services introduced in this paper. This will allow a practical

evaluation of both requirements and the proposed solutions on different medical terminologies.

Acknowledgements

This work is supported by the Netherlands' Organization for Scientific Research NOW by grant 220-80-001.

References

- [1] Cornet R, Abu-Hanna A. Evaluation of a frame-based ontology. A formalization-oriented approach. In Proceedings of MIE2002, Studies in Health Technology & Information, volume 90, pages 488-93, 2002.
- [2] Haarslev V, Moller V. RACER system description. In Gore R, Leitsch A, and Nipkow T, editors, IJCAR 2001, number 2083 in LNAI, 2001.
- [3] Nebel B. Terminological Reasoning is inherently intractable. AI, 43:235-249, 1990.
- [4] http://www.opengalen.org/technology/further_tut.html
- [5] http://www.apelon.com/news/press_062298.htm.
- [6] <http://protege.stanford.edu>.
- [7] Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P, editors. The Description Logics Handbook. Cambridge University Press, 2003.
- [8] Schmidt-Schauss M, Smolka G. Attributive concept descriptions with complements. AI 48:1-26, 1991.
- [9] Hacid M-S, Leger A, Rey C, Toumani F. Computing Concept Covers: a Preliminary Report. In Proceedings of the Description Logics Workshop 2002.
- [10] Rector A.L. Clinical Terminology: Why Is it so Hard? Meth Inform Med 38:239-52, 1999.
- [11] Cimino JJ. Desiderata for controlled medical vocabularies in the twenty-first century. Meth Inform Med 37 (4-5): 394-403, 1998.
- [12] Schlobach S, Cornet R. Explanation of terminological reasoning; a preliminary report. In Proceedings of the Description Logics Workshop, 2003.
- [13] Schlobach S, Cornet R. Non-standard reasoning services for the debugging of description logic terminologies. In Proceedings of IJCAI, 2003.

Address for correspondence

Stefan Schlobach
Language and Inference Technology, ILLC
Universiteit van Amsterdam
Nieuwe Achtergracht 166
1018 WV Amsterdam,
The Netherlands
schlobac@science.uva.nl