Explanation of Terminological Reasoning
A Preliminary Report

Stefan Schlobach
Language and Inference Technology, ILLC
Universiteit van Amsterdam, NL
schlobac@science.uva.nl

Ronald Cornet
Academic Medical Center
Universiteit van Amsterdam, NL
R.Cornet@amc.uva.nl

Abstract

This paper describes our current activities to supply extended reasoning support to knowledge engineers who are building terminologies using Description Logics (DL) reasoners. The new services originate in the development of the DICE[1] terminology where the lack of appropriate debugging or explanation facilities hindered a more efficient (and possibly more concise) construction of a corresponding DL TBox. We discuss a number of alternative methods to explain incoherence of TBoxes, unsatisfiability of concepts and concept subsumption.

1 Introduction

Developing a terminology is a time-consuming and error-prone process. DICE, a terminology developed at the AMC in Amsterdam for the unambiguous and unified classification of patients in Intensive Care medicine, defines more than 2400 concepts and uses 45 relations. Let us illustrate some of the problems: in a first version of DICE a "brain" was incorrectly specified, among others, as a "central nervous-system" and "body-part" located in the head. This definition is contradictory as nervous-systems and body-parts are declared disjoint in DICE. Fortunately, current Description Logic reasoners, such as RACER [4], can detect this type of inconsistency and the knowledge engineer can identify the cause of the problem. Unfortunately, many other concepts are defined based on the erroneous definition of "brain" forcing each of them to be erroneous as well. In practice, DL reasoners provide lists of hundreds of unsatisfiable concepts for the DICE TBox and the debugging remains a jigsaw to be solved by human experts, with little additional explanation to support this process. The situation is even worse when ontologies without formal semantics are translated into DL terminologies [11, 3]. Cornet and Abu-Hanna, for example, have studied alternative translations, using e.g. rigid translations of slot-fillers in frames into existentially **and** universally quantified concepts. In some cases this will be too strong

[1]DICE stands for "Diagnoses for Intensive Care Evaluation".

an assumption and will lead to high numbers of unsatisfiable concepts, and the cause for the unsatisfiability will have to be examined separately for each individual concept. In this case, explanation of TBox unsatisfiability and incoherence is paramount. A similar approach to render the intended semantics of a terminology more precise is to replace primitive concept definitions with non-primitive ones. In this case, a number of concepts turn out to be equivalently defined or to subsume each other, and the identification of the causes of the subsumption relations becomes a valuable asset.

In [9] we introduce *pinpointing* as a method to identify the precise position of errors in a TBox by, first, calculating minimal incoherence-preserving sub-terminologies and, secondly, maximally generalising axioms. There, algorithms and preliminary practical results are described and we will, in Section 4, only briefly recall the main notions of MIPS and GITs. In the Description Logic $\mathcal{ALC}$ [10] there are two main reasons why concept subsumption might be intuitively difficult to understand, first, because of the *structural* complexity of subsumer and subsumed concept and, secondly, because of the *logical* interplay between the two concepts. We propose to explain subsumption by reducing the structural complexity and by bringing the logical interplay more to the foreground. We will describe two alternative approaches. An *Illustration* for a subsumption $\mathcal{T} \models C \sqsubseteq D$ is a single concept I which "bridges" a subsumption relation as an intermediate subsumer $\mathcal{T} \models C \sqsubseteq I \sqsubseteq D$, and which captures some of the common structure of C and D. Alternatively, a *magnification* for $\mathcal{T} \models C \sqsubseteq D$ is a pair C' and D' of concepts where $\mathcal{T} \models C \sqsubseteq C' \sqsubseteq D' \sqsubseteq D$, and where C' is a structural simplification of C and D' a structural simplification of D. In this way the relation $\mathcal{T} \models C' \sqsubseteq D'$ "zooms" into the relation $\mathcal{T} \models C \sqsubseteq D$, it *magnifies* it. Illustration and magnification are new ideas that will be discussed in more detail in Section 5.

While pinpointing has been already applied in explaining incoherence in DICE, implementation and evaluation of illustration and magnification is work in process. In this paper we describe how to define different notions of structural similarity, and how to apply them to explanation. With this preliminary report we aim at introducing new mechanisms for explanation of DL reasoning.

2 Description Logics

We shall not give a formal introduction to Description Logics (DL) here, but point to the first two chapters of the DL handbook [1] for an excellent overview. Briefly, DLs are set description languages with concepts (usually we use capital letters), interpreted as subsets of a domain, and roles which are binary relations, denoted by small letters. $\mathcal{ALC}$ 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 $\exists r.C$. Commonly, $C \rightarrow D$ is used as an abbreviation for $\neg C \sqcup D$.

In a terminology $\mathcal{T}$ (called TBox) the interpretations of concepts can be restricted to the *models* of $\mathcal{T}$ by *axioms* of the form $C \sqsubseteq D$ or $C \doteq D$. Based on this model-theoretic semantics, concepts can be checked for *unsatisfiability*: whether they are necessarily interpreted as the empty set. A TBox $\mathcal{T}$ is called *coherent* if no unsatisfiable concept-name occurs in $\mathcal{T}$. Other checks include *subsumption* of two concepts C and D (a subset relation of $C^{\mathcal{I}}$ and $D^{\mathcal{I}}$ w.r.t. all models $\mathcal{I}$ of $\mathcal{T}$). Subsumption between concepts C and D w.r.t. a TBox $\mathcal{T}$ will be denoted by $\mathcal{T} \models C \sqsubseteq D$.

A TBox is *unfoldable* if the left-hand side of the axioms (called the defined concepts) are atomic, and if the right-hand sides (the definitions) contain no reference to the defined concept [7]. Subsumption and satisfiability without reference to a TBox will be denoted by *concept subsumption* and *concept satisfiability* and we will write $\models C \sqsubseteq D$ and $C \neq \bot$, respectively. Most examples in this paper are using concept subsumption only, but the definitions are usually more general and for subsumption w.r.t. a TBox. Note, that subsumption and satisfiability of concepts w.r.t. unfoldable TBoxes can be reduced to concept subsumption and concept satisfiability.

3 Structural Similarity

The new reasoning services of *generalised terminologies, illustration* and *magnification*, which we are going to introduce in this paper, will require target concepts to be structurally related to the original concepts such as subsumers or definitions. Structural properties of concepts have been used in early DL systems to decide subsumption [5], but it is well known that structural subsumption is incomplete for more complex languages such as $\mathcal{ALC}$. However, to explain subsumption or incoherence logical completeness is irrelevant and we can, actually, use structural similarity of concepts to simplify terminologies or subsumption relations.

The general principal will be the same in all cases: we will explain logical relations by the same relations between simplified but structurally related concepts. For different representation languages, types of knowledge bases and application problems different types of structural relatedness can be considered.

The first proposed notion is based on sets of *qualified sub-concepts* as the structurally related concepts. Informally, the qualified sub-concepts are variants of those concepts a knowledge engineer explicitly uses in the modelling process, where the context of this use is kept intact. By context we mean the sequence of quantifiers a concept-name occurs in and the polarity, i.e. whether it is used within the scope of an even or odd number of negations[2] The definition is by induction.

$$
\begin{aligned}
&qs(C \sqcap D) = \{C', D', C' \sqcap D' \mid C' \in qs(C), D' \in qs(D)\}\\
&qs(C \sqcup D) = \{C', D', C' \sqcup D' \mid C' \in qs(C), D' \in qs(D)\}\\
&qs(\exists r.C) = \{\exists r.C' \mid C' \in qs(C)\}\\
&qs(\forall r.C) = \{\forall r.C' \mid C' \in qs(C)\}\\
&qs(\neg C) = \{\neg C' \mid C' \in qs(C)\}\\
&qs(A) = \{A\} \text{ if } A \text{ is atomic}
\end{aligned}
$$

The set of qualified sub-concepts for a concept C will be denoted by $qs(C)$. Take for example a concept C defined as $\exists r.(D \sqcup \forall s.E)$. In this case the concept $\exists r.\forall s.E$ is structurally related to C w.r.t. qualified sub-concepts, whereas concept $\exists r.\neg D$ or $\exists s.D$ are not. A concept C is then *structurally similar* to a concept D if $C \in qs(D)$.

Unfortunately, there are possibly exponentially many qualified sub-concepts, which makes enumerating unfeasible in practice. For magnification of subsumption, which will be introduced in Section 5, we will not just use structural similarities but a semantic ordering on concepts to choose an explanation. For this purpose we can include a semantic ordering on the qualified sub-concepts, i.e. we get two sets of concepts related

[2] We call these concepts the *qualified sub-concepts* of a concept because they are *qualified* (or restricted) in the same way as the original concepts w.r.t. negation and quantification.

to a concept C, the *generalised qualified sub-concepts (GQS)* (abbreviated by $gqs(C)$) and the *specialised qualified sub-concepts (SQS)* (abbreviated by $sqs(C)$). GQSs of C are those qualified sub-concepts which are more general w.r.t. the subsumption relation, SQSs those that are more special. The definition is inductively as follows:

$$gqs(C \sqcap D)=\{C', D', C' \sqcap D' \mid C' \in gqs(C), D' \in gqs(D)\}$$
$$gqs(C \sqcup D)=\{C' \sqcup D' \mid C' \in gqs(C), D' \in gqs(D)\}$$
$$gqs(\exists r.C)=\{\exists r.C' \mid C' \in gqs(C)\}$$
$$gqs(\forall r.C)=\{\forall r.C' \mid C' \in gqs(C)\}$$
$$gqs(\neg C)=\{\neg C' \mid C' \in sqs(C)\}$$

$$gqs(A)=sqs(A) = \{A\} \text{ if } A \text{ is atomic}$$

$$sqs(C \sqcap D)=\{C' \sqcap D' \mid C' \in sqs(C), D' \in sqs(D)\}$$
$$sqs(C \sqcup D)=\{C', D', C' \sqcup D' \mid C' \in sqs(C), D' \in sqs(D)\}$$
$$sqs(\exists r.C)=\{\exists r.C' \mid C' \in sqs(C)\}$$
$$sqs(\forall r.C)=\{\forall r.C' \mid C' \in sqs(C)\}$$
$$sqs(\neg C)=\{\neg C' \mid C' \in gqs(C)\}$$

Note the interplay of GQSs and SQSs in the negated case, and the dual structure of the definitions for the boolean operators. A simple consequence of this definition is that $\models C \sqsubseteq C'$ for every $C' \in gqs(C)$ and $\models D' \sqsubseteq D$ for each $D' \in sqs(D)$.

Both notions of structural similarity might be inappropriate or insufficient for certain, more advanced, reasoning tasks, such as the *suggestion of corrections*. Experience with the DICE terminology has, for example, shown that a common modelling error is to define a concept as a sub-concept rather than as related by, for instance, a *part_of* relation. In this case the use of qualified sub-concepts for structural similarity can be too restrictive. The original definition *CentralNervousSystem* $\sqcap$ *BodyPart* $\sqcap \dots$ of the concept *Brain* in the DICE terminology (given disjointness of the two concepts *CentralNervousSystem* and *BodyPart*) should be corrected to $\exists$*part_of.CentralNervousSystem* $\sqcap$ *BodyPart* $\sqcap \dots$, which is not structurally related to the original concept according to the chosen definition. If we intend to use structurally related concepts not just for detection of errors by pinpointing (as proposed in Section 4) but to suggest corrections this definition will not produce satisfying results. In this case a notion of similarity which relates concepts $\exists$*part_of.CentralNervousSystem* and *CentralNervousSystem* is required. As we believe this to be an interesting extension of our approach, we plan to investigate such an alternative notion of structural relatedness in future research. This example shows that the definition of structural similarity gives us a handle to identify particular types of errors, or to explain particular reasoning steps, but might need adaption for others.

Let us now discuss the new reasoning services for explanation in more detail.

4 Explaining Unsatisfiability and Incoherence

In this section we study ways of explaining unsatisfiability and incoherence in DL terminologies. Consider the following (incoherent) TBox $\mathcal{T}_1$, where A, B and C are primitive and $A_1, \dots, A_7$ defined concept names:

$A_1 \doteq \neg A \sqcap A_2 \sqcap A_3$	(ax_1)	$A_2 \doteq A \sqcap A_4$	(ax_2)
$A_3 \doteq A_4 \sqcap A_5$	(ax_3)	$A_4 \doteq \forall s.B \sqcap C$	(ax_4)
$A_5 \doteq \exists s.\neg B$	(ax_5)	$A_6 \doteq A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$	(ax_6)
$A_7 \doteq A_4 \sqcap \exists s.\neg B$	(ax_7)		

The set of unsatisfiable concept names is $\{A_1, A_3, A_6, A_7\}$. Although this is still of manageable size, it hides crucial information, e.g., that unsatisfiability of A_1 not only depends on the contradiction between A and A_2, but also on the unsatisfiability of A_3 because of the contradictions between A_4 and A_5. We will use this example to explain our debugging methods. In [9] we propose to simplify a terminology $\mathcal{T}$ in order to reduce the available information to the root of the incoherence. More concretely we first exclude axioms which are irrelevant to the incoherence and then provide simplified definitions highlighting the exact position of a contradiction within the axioms of this reduced TBox. We will call the former *axiom pinpointing*, the latter *concept pinpointing*.

Axiom pinpointing Axiom pinpointing identifies debugging-relevant axioms, where an axiom is *relevant* if a contradictory TBox becomes coherent once the axiom is removed or if, at least, a particular, previously unsatisfiable concept turns satisfiable. Unsatisfiability-preserving sub-TBoxes of a TBox $\mathcal{T}$ and an unsatisfiable concept A are subsets of $\mathcal{T}$ in which A is unsatisfiable. In general there are several of these sub-TBoxes and we select the minimal ones, i.e., those containing only axioms that are necessary to preserve unsatisfiability.

Definition 4.1 Let A be a concept which is unsatisfiable in a TBox $\mathcal{T}$. A set $\mathcal{T}' \subseteq \mathcal{T}$ is a *minimal unsatisfiability-preserving sub-TBox (MUPS)* of $\mathcal{T}$ if A is unsatisfiable in $\mathcal{T}'$, and A is satisfiable in every sub-TBox $\mathcal{T}'' \subset \mathcal{T}'$.

MUPS for our TBox $\mathcal{T}_1$ and A_1 are $\{\{ax_1, ax_2\}, \{ax_1, ax_3, ax_4, ax_5\}\}$, for $\mathcal{T}_1$ and A_6 $\{\{ax_1, ax_2, ax_4, ax_6\}, \{ax_1, ax_3, ax_4, ax_5, ax_6\}\}$. MUPS are useful for relating unsatisfiability to sets of axioms but are also used to calculate MIPS.

MIPS are minimal subsets of an original TBox preserving unsatisfiability of at least one atomic concept.

Definition 4.2 Let $\mathcal{T}$ be an incoherent TBox. A TBox $\mathcal{T}' \subseteq \mathcal{T}$ is a *minimal incoherence-preserving sub-TBox (MIPS) of* $\mathcal{T}$ if $\mathcal{T}'$ is incoherent, and every sub-TBox $\mathcal{T}'' \subset \mathcal{T}'$ is coherent.

For $\mathcal{T}_1$ we get three MIPS $\{ax_1, ax_2\}, \{ax_3, ax_4, ax_5\}, \{ax_4, ax_7\}$. It can easily be checked that each of the three incoherent TBoxes is indeed a MIPS as taking away a single axiom renders each of the three coherent. The first one signifies, for example, that the first two axioms are already contradictory without reference to any other axiom, which suggests a modelling error already in these two axioms.

Concept Pinpointing Incoherence of a TBox can be regarded as an over-specification of one or more concepts in the relevant definitions. *Generalised terminologies* are terminologies where some of the definitions have been generalised.[3] Furthermore, we require generalised definitions to be structurally related to the original axioms and we use qualified sub-concepts of Section 3 for this purpose. Formally, a concept C' is called a *syntactic generalisation* of a concept C if $C' \in qs(C)$ and $\models C \sqsubseteq C'$ (independent of $\mathcal{T}$). Now, generalised incoherence-preserving terminologies (GITs) are TBoxes where the defining concepts of the axioms are maximally generalised without losing incoherence.

Definition 4.3 Let $\mathcal{T} = \{C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n\}$ be an incoherent TBox. An incoherent TBox $\mathcal{T}' = \{C_1 \sqsubseteq D'_1, \ldots, C_n \sqsubseteq D'_n\}$ is a *generalised incoherence-preserving terminology (GIT)* of $\mathcal{T}$ iff

- each D'_i is a syntactic generalisation of D_i $(1 \leq i \leq n)$, and
- every TBox $\mathcal{T}'' = \{C_1 \sqsubseteq D''_1, \ldots, C_n \sqsubseteq D''_n\}$ with a syntactic generalisation D''_i of D_i where $\mathcal{T} \models D'_i \sqsubseteq D''_i$ and $\mathcal{T} \models D''_i \not\sqsubseteq D'_i$ (for some i) is coherent.

Note, that the set of GITs of a TBox $\mathcal{T}$ is equivalent to the union of GITs for the MIPS of $\mathcal{T}$ [9]. We could also have defined GITs using the generalised qualified sub-concepts of Section 3. However, such a definition would not be equivalent to the one we introduced in [9], which we also decided to present here to emphasise the different character of structural simplification and semantic ordering. Of all possible GITs we conjecture that the simplest ones, e.g., those with *syntactically minimal* generalisations, are most likely to be useful for the identification of errors. In our experiments we use two alternative formalisations, the first with respect to minimal size of axioms, the second with respect to the number of concept names occurring in the GIT. Three minimal sized GITs exist for our example TBox $\mathcal{T}_1$, where we only show the non-trivial axioms:

$$\{\{A_1 \sqsubseteq \neg A \sqcap A_2, A_2 \sqsubseteq A\}, \{A_3 \sqsubseteq A_4 \sqcap A_5, A_4 \sqsubseteq \forall s.B, A_5 \sqsubseteq \exists.\neg B\},$$
$$\{A_7 \sqsubseteq A_4 \sqcap \exists s.\neg B, A_4 \sqsubseteq \forall s.B\}\}.$$

Algorithms for axiom and concept pinpointing were presented in [9]. Given that the set $\{ax_4\}$ is a subset of two of the three MIPS the axiom $\{ax_4\}$ is most likely to cause incoherence of $\mathcal{T}_1$ (we call such a set a core of arity two). Furthermore, we will have to take care of a contradiction on A in (ax_1) and (ax_2). Practical experience with DICE has shown that the cause for incoherence could be the erroneous use of concepts as well as their actual definition. Note that our methods ignore cases like (ax_6) where the second disjunct remains unsatisfiable even though now every concept-name in the corrected TBox is satisfiable. Although such a "local unsatisfiability" might indicate an erroneous specification there is currently no way of identifying such problems.

5 Explaining Subsumption

Not only can DL reasoner efficiently check for consistency, but also for subsumption, i.e. whether a concept, e.g. $\exists child.\top \sqcap \forall child.Doctor$ is a subclass of another concept $\exists child.(Doctor \sqcup Rich)$. Unfortunately, reasoners do not explain why this is the case, and it is not always easy to see why subsumption holds.

[3] To simplify matters we generalise the right-hand side of axioms only as we are currently only considering unfoldable TBoxes.

5.1 Explaining Subsumption by Illustration

Our first approach is simple: a suitable illustration for this subsumption relation seems to be that $\exists child.\top \sqcap \forall child.Doctor$ is more specific than $\exists child.Doctor$ which, again, is more specific than $\exists child.(Doctor \sqcup Rich)$. Not only is this concept $\exists child.Doctor$ strongly linked by the common vocabulary to both the subsumer and the subsumed concept it is also the simplest illustration one can give.

Consider the following concept subsumption $\models C_{ex} \sqsubseteq D_{ex}$, taken from [2], which the authors introduce to give an example for their "explanation as proof-fragment" strategy, where $C_{ex} := \exists child.\top \sqcap \forall child.\neg((\exists child.\neg Doctor) \sqcup (\exists child.Lawyer))$ is subsumed by $D_{ex} := \exists child.\forall child\ (Rich \sqcup Doctor)$. Instead of providing a concise and simplified extract of a formal proof as an explanation as done in [2] we suggest an alternative, more static approach, which we call *explaining by illustration.* Imagine the above information is given in natural language:

> *Suppose somebody has children, and each child has neither a child which is not a doctor nor a child which is a Lawyer. Then, this person must have a child, every child of which is either rich or a doctor.*

A natural language explanation for this statement could be:

> *The person described above has a child every child of which is a doctor.*

The intermediate statement *a person with a child, every child of which is a doctor* can be considered an *illustration* of $\mathcal{T} \models C_{ex} \sqsubseteq D_{ex}$. It can be formalised as $I_{ex} := \exists child.\forall child.Doctor$, and it has the particular property that it subsumes C_{ex} and is subsumed by D_{ex}. Moreover, it is constructed from vocabulary which occurs both in C_{ex} and D_{ex}, e.g., the information that the person's grandchildren might be rich is irrelevant, as no information about the grand-children's finances is provided in C_{ex}. Finally, the illustration is of minimal size, as this increases the likelihood of the explanation being understandable.

Definition 5.1 Let two concepts C and D and a TBox $\mathcal{T}$, such that $\mathcal{T} \models C \sqsubseteq D$. Furthermore, let $size(C)$ denote the total number of atomic symbols in a concept C, and $\mathcal{N}(C)$ be the set of different concept-names occurring in C. An *illustration* for $\mathcal{T} \models C \sqsubseteq D$ is a concept I such that $\mathcal{T} \models C \sqsubseteq I$ and $\mathcal{T} \models I \sqsubseteq D$, and where $I \in qs(C) \cap qs(D)$. We say that

- I is a *minimal-sized illustration* iff there is no illustration J of $\mathcal{T} \models C \sqsubseteq D$ s.t. $size(J) < size(I)$. Alternatively
- I is a *minimal-vocabulary illustration* iff there is no illustration J of $\mathcal{T} \models C \sqsubseteq D$ s.t. $\mathcal{N}(J) \subset \mathcal{N}(I)$.

Formally, such an illustration is an interpolant with minimal vocabulary or size. In [8] we introduce *optimal interpolants* with a minimal number of concept names, and tableau based algorithms for concept subsumption in $\mathcal{ALC}$. These algorithms can be used directly to provide minimal-vocabulary illustrations. To explain more complex subsumption relations a single illustration might not be sufficient, and iterative explanation is needed. Here subsumption between concepts and their illustrations could be explained as well, either by new illustrations, by traditional explanation of proofs or by *explanation by magnifying.*

5.2 Explaining Subsumption by Magnifying

Let us look again at our previous example for subsumption where the relation $\models C_{ex} \sqsubseteq D_{ex}$ was explained by an illustration $I_{ex} := \exists \mathit{child}.\forall \mathit{child}.\mathit{Doctor}$. Remember that C_{ex} was $\exists \mathit{child}.\top \sqcap \forall \mathit{child}.\neg((\exists \mathit{child}.\neg \mathit{Doctor}) \sqcup (\exists \mathit{child}.\mathit{Lawyer}))$ and the new subsumption $\models C_{ex} \sqsubseteq I_{ex}$ might need further explanation. Now, the idea is to look, again, at the syntactic structure of subsumer I_{ex} and subsumed concept C_{ex}, and to find subsumption-preserving generalisations (or specialisations) of C_{ex} (of I_{ex} respectively). As we are now looking for generalisations and specialisations of subsumer and subsumed concept we propose to use generalised (and specialised) qualified subconcepts to define structural similarity. A *magnification* is now a pair of concepts where one subsumes the other, and which a structurally similar, but more simple versions of the original concepts.

Definition 5.2 Let $\mathcal{T}$ be a TBox. A pair of concepts C' and D' *magnifies* a subsumption relation $\mathcal{T} \models C \sqsubseteq D$ iff

- $\mathcal{T} \models C \sqsubseteq C' \sqsubseteq D' \sqsubseteq D$, if
- $C' \in gqs(C)$ and $D' \in sqs(D)$, and if
- there are no $C'' \in gqs(C)$ and $D'' \in sqs(D)$ such that $\mathcal{T} \models C' \sqsubset C''$ and $\mathcal{T} \models C'' \sqsubseteq D'$, or $\mathcal{T} \models D'' \sqsubset D'$ and $\mathcal{T} \models C' \sqsubseteq D''$.

As before we will look for magnifying concepts of small size and logical complexity. The concepts $C'_{ex} := \exists \mathit{child}.\top \sqcap \forall \mathit{child}.\neg(\exists \mathit{child}.\neg \mathit{Doctor})$ and $I'_{ex} := \exists \mathit{child}.\forall \mathit{child}.\mathit{Doctor}$ magnify then, for example, our previous subsumption $\models C_{ex} \sqsubseteq I_{ex}$. Intuitively, what happens is the following: The subsumption $\models C_{ex} \sqsubseteq C'_{ex}$ is a structural subsumption, i.e. can be understood as a non-logical simplification of concept C_{ex}:

$$\begin{aligned}\models\ & \exists \mathit{child}.\top \sqcap \forall \mathit{child}.\neg((\exists \mathit{child}.\neg \mathit{Doctor}) \sqcup (\exists \mathit{child}.\mathit{Lawyer})) \\ & \sqsubseteq \exists \mathit{child}.\top \sqcap \forall \mathit{child}.\neg(\exists \mathit{child}.\neg \mathit{Doctor}).\end{aligned}$$

The concept I_{ex} is already structurally minimal so that we have chosen $I'_{ex} \equiv I_{ex}$. What remains is a structurally non-reducible subsumption relation $\models C'_{ex} \sqsubseteq I'_{ex}$, which now represents the logical interplay in magnified form:

$$\models \exists \mathit{child}.\top \sqcap \forall \mathit{child}.\neg(\exists \mathit{child}.\neg \mathit{Doctor}) \sqsubseteq \exists \mathit{child}.\forall \mathit{child}.\mathit{Doctor}.$$

Note that our definition of structural similarity does not provide any mechanisms to deal with negation and that explanation of reasoning in application domains with complex negations might be insufficient. In the DICE terminology negation is always either atomic or represented in disjointness statements. For this reason we have not yet investigated this problem any further.

We will end this introduction of magnification by discussing a relatively naive algorithm to calculate a single small magnifying subsumption: for two concepts C and D in a subsumption relation $\mathcal{T} \models C \sqsubseteq D$ we calculate first a *maximal generalisation* C' of C w.r.t. D, and secondly a *minimal specialisation* D' of D w.r.t. the new C'. The resulting relation $\mathcal{T} \models C' \sqsubseteq D'$ is a magnifying subsumption for C and D. Formally, a *maximal generalisation* of a concept C w.r.t. a concept D is a concept C' with the following three properties: $\mathcal{T} \models C \sqsubseteq C' \sqsubseteq D$, $C' \in gqs(C)$ and there is no concept

$C'' \in gqs(C)$ s.t. $\mathcal{T} \models C' \sqsubset C''$ and $\mathcal{T} \models C'' \sqsubseteq D$. The definition of *minimal specialisation* is dual using specialised qualified sub-concepts $sqs(D')$ instead. The semantic ordering on the concepts now allows to calculate maximal generalisations and minimal specialisations with relatively little effort. Moreover, as the ordering on concepts is bound to the relaxation of conjunctions (or disjunctions in the negated case) we also have an ordering of the GQSs according to their size. Although this method might not calculate magnifications of minimal size or vocabulary it is probably a reasonable compromise between computational feasibility and conceptual demands. An implementation is forthcoming.

6 Conclusion

With this paper we hope to encourage an exchange of ideas within the DL community to extend the efforts to equip the high-quality reasoning systems with equally powerful and reliable explanation facilities. This can only improve the acceptance of logical reasoning and formal reasoning in real world applications.

The results we report in this paper are of three different types: MIPS and MUPS have been introduced in [9] and there are algorithms which we implemented and evaluated on the DICE terminology. The ideas for explanation by illustrations, however, are lesser developed. In [8] we have presented algorithms for optimal interpolation and a reliable and efficient implementation is under way. Explanation by magnifying, finally, is a new idea: the definitions are still subject to discussion and the algorithms are rather naive and sketchy. Nevertheless, we believe that all three approaches will provide useful in the full-scale explanation facility we are currently working on.

As this paper describes work in progress the list of open problems and further research is obviously long: First and foremost we need to fully implement the methods described in this paper to allow for practical evaluation on more complex problems. Moreover, in the DICE terminology we have to explain subsumption with respect to TBoxes. As the axioms are unfoldable our approach here is to incorporate axiomatic information into the definition of structural similarity.

Up to now we have not discussed explanation of satisfiability or non-subsumption. As McGuinness pointed out in [6] an explanation should return a model or counter-model but more work needs to be done to identify the most suitable ones. Finally, a transformation of the DL syntax into some human-readable format will be needed to increase the acceptance of logical reasoning and its tools.

References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.

[2] A. Borgida, E. Franconi, I. Horrocks, D. McGuinness, and P. Patel-Schneider. Explaining $\mathcal{ALC}$ subsumption. In *DL-99*, pages 37–40, 1999.

[3] R. Cornet and A. Abu-Hanna. 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.

[4] V. Haarslev and R. Möller. RACER system description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *IJCAR 2001*, number 2083 in LNAI, 2001.

[5] H.Levesque and R.Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[6] Deborah McGuinness. *Explaining Reasoning in Description Logics*. PhD thesis, Department of Computer Science, Rutgers University, 1996.

[7] B. Nebel. Terminological reasoning is inherently intractable. *AI*, 43:235–249, 1990.

[8] S. Schlobach. Optimal interpolation, 2003. submitted.

[9] S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI*, 2003. Accepted for publication.

[10] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[11] S. Schulz and U. Hahn. Medical knowledge reengineering - converting major portions of the UMLS into a terminological knowledge base. *International Journal of Medical Informatics*, 64:207–221, 2001.