

Incoherence as a Basis for Measuring the Quality of Ontology Mappings

Christian Meilicke and Heiner Stuckenschmidt

Computer Science Institute
University of Mannheim, Germany
{christian,heiner}@informatik.uni-mannheim.de

Abstract. Traditionally, the quality of ontology matching is measured using precision and recall with respect to a reference mapping. These measures have at least two major drawbacks. First, a mapping with acceptable precision and recall might nevertheless suffer from internal logical problems that hinder a sensible use of the mapping. Second, in practical situations reference mappings are not available. To avoid these drawbacks we introduce quality measures that are based on the notion of mapping incoherence that can be used without a reference mapping. We argue that these measures are a reasonable complement to the well-known measures already used for mapping evaluation. In particular, we show that one of these measures provides a strict upper bound for the precision of a mapping.

1 Introduction

Assessing the quality of alignments is an important aspect of ontology matching. A number of different measures have been proposed for this purpose. According to [4] it can be distinguished between compliance measures, measures concerned with system usability, and performance measures that focus on runtime or memory requirements. A compliance measure compares a set of correspondences with a gold standard which should be the complete set of all correct correspondences. The most prominent compliance measures are precision and recall which have been adapted from information retrieval to the field of schema and ontology matching [1]. As complement to these measures we propose a family of measures based on the definition of mapping incoherence. These measures do not fall in one of the above mentioned categories, but should be categorized as formal or logic-based measures.

Compared to the widely used measures of precision and recall, the measures that will be proposed in this paper do not rely on the existence of a gold standard (also referred to as reference mapping). Contrary to this, they measure internal properties of a mapping based on the semantics of the ontologies aligned via the mapping. This makes our approach applicable in matching scenarios where we do not have a gold standard. Measuring the incoherence of a mapping is motivated by the idea that the incoherence of a mapping will hinder its sensible use even though it might contain a significant amount of correct correspondences. Although we introduce incoherence as a new dimension for quantifying mapping quality, there is a non-obvious relation to traditional measures. In particular, we show that we can use one of the suggested measures to compute a strict

upper bound for the precision of a mapping. This result is surprising at first sight and shows the significance of the overall approach.

In the following section we discuss related work on quality measures for mappings and explain how our approach extends existing work. In section 3 we recall and refine the theory of mapping incoherence as basis for the following sections. Before introducing incoherence measures, we discuss some problems caused by mapping incoherence which justify the importance of measuring incoherence (section 4). In particular, we explain the effects of incoherence with respect to the application scenario of data transformation and query processing. In section 5 we introduce four incoherence measures divided in two groups. Measures of the first group are concerned with the impact of incoherence, while measures of the second group are used to measure the effort of repairing an incoherent mapping. Finally, we show that one of these measures can be used to compute a strict upper bound for mapping precision (section 6) followed by some concluding remarks (section 7).

2 Related Work

Several suggestions have been made to extend and introduce new evaluation measures to the field of ontology matching. In [2] Ehrig and Euzenat introduce relaxed precision and recall. Their work is motivated by the idea that a correspondence of a mapping \mathcal{M} might not be totally incorrect even though it is not contained in reference mapping \mathcal{R} . Thus, it can be measured how close a correspondence is to a similar one in \mathcal{R} . Amongst others they suggest to measure the correction effort to transform such a correspondence into a correct one. We pick up this idea and suggest measuring incoherence based on the effort necessary to remove all causes of incoherence from \mathcal{M} . In [3] Euzenat introduces semantic precision and recall. These measures are based, roughly speaking, on comparing the (bounded) deductive closure of \mathcal{R} and \mathcal{M} instead of a direct comparison. Such an approach requires the use of logical reasoning where both correspondences and ontologies are considered. While Euzenat focuses on the entailment of correspondences, our approach accounts that certain combinations of correspondences result in incoherence.

Measuring mapping incoherence is closely related to measuring and repairing ontology incoherence. Thus, we adapted some of the measures defined by Qi and Hunter in [11]. Later we will show how to reduce the incoherence of a mapping \mathcal{M} to concept unsatisfiability in an ontology that results from merging the ontologies matched via \mathcal{M} . An obvious way of measuring mapping incoherence is thus based on counting the number of unsatisfiable concepts in the merged ontology. As proposed in [6], it can be distinguished between root and derived unsatisfiable concepts. Accordant to [11], we pick up this distinction and distinguish between two types of measuring the impact of incoherence based on concept unsatisfiability.

In previous work [8, 9] we have developed and tested strategies to repair incoherent ontology mappings.¹ These strategies rely on discarding individual correspondences from an incoherent mapping \mathcal{M} to finally arrive at a coherent submapping $\mathcal{M}^* \subseteq \mathcal{M}$.

¹ Notice that in previous work we misleadingly used the notion of inconsistency instead of incoherence. Precise definitions of these notions are given in [5].

Clearly, such a strategy should remove a minimal set of correspondences. This approach leads to an incoherence measure that indicates the effort of repairing incoherent mappings in numbers of correspondences to be removed. Moreover, it has been emphasized that the confidence value of a correspondence plays an important role in mapping repairing. Thus, we distinguish between a cardinality based measure and a confidence based measure. Both measures quantify the effort of revising an incoherent mapping.

3 Foundations

According to Euzenat and Shvaiko [4] a correspondence can be defined as a semantic relation between ontological entities annotated with a confidence value.

Definition 1 (Correspondence and Mapping). *Given ontologies \mathcal{O}_1 and \mathcal{O}_2 , let Q be a function that defines sets of matchable elements $Q(\mathcal{O}_1)$ and $Q(\mathcal{O}_2)$. A correspondence between \mathcal{O}_1 and \mathcal{O}_2 is a 4-tuple $\langle e, e', r, n \rangle$ such that $e \in Q(\mathcal{O}_1)$ and $e' \in Q(\mathcal{O}_2)$, r is a semantic relation, and n is a confidence value from a suitable structure $\langle D, \leq \rangle$. A mapping between \mathcal{O}_1 and \mathcal{O}_2 is a set of correspondences between \mathcal{O}_1 and \mathcal{O}_2 .*

Definition 1 allows to capture a wide class of correspondences by varying what is admissible as matchable element, semantic relation, and confidence value. In this work we consider correspondences between named concepts and properties. We also restrict correspondences to match entities of the same type, i.e. both e and e' have to be concepts or both have to be properties. We also restrict r to be \equiv , \sqsubseteq or \sqsupseteq , i.e. we only focus on equivalence and subsumption correspondences. Finally, we assume that the confidence value, which can be seen as a measure of trust in the fact that the correspondence holds, is represented numerically on $D = [0.0, 1.0]$.

As argued in [8] and [9] the semantics of a mapping \mathcal{M} between ontologies \mathcal{O}_1 and \mathcal{O}_2 can be defined in the context of merging \mathcal{O}_1 and \mathcal{O}_2 via \mathcal{M} . In this section we focus on technical aspects and postpone the discussion on adequacy and implications to the next section. A merged ontology contains the axioms of \mathcal{O}_1 and \mathcal{O}_2 as well as the correspondences of \mathcal{M} translated into axioms of the merged ontology.

Definition 2 (Merged ontology). *Let \mathcal{O}_1 and \mathcal{O}_2 be ontologies (finite sets of axioms). The merged ontology $\mathcal{O}_1 \cup_{\mathcal{M}} \mathcal{O}_2$ of \mathcal{O}_1 and \mathcal{O}_2 connected by \mathcal{M} is defined as $\mathcal{O}_1 \cup_{\mathcal{M}} \mathcal{O}_2 = \mathcal{O}_1 \cup \mathcal{O}_2 \cup \{t(x) \mid x \in \mathcal{M}\}$ with t being a translation function that maps correspondences to axioms.*

Notice that in \mathcal{O}_1 and \mathcal{O}_2 a concept or property might have the same local name. To refer without ambiguity in the context of a merged ontology to an entity e which origins from \mathcal{O}_i we use prefix notation $i\#e$ in the following. There is a straightforward way to translate concept correspondences and property correspondences into DL axioms. We refer to the corresponding translation function as natural translation t_n .

Definition 3 (Natural Translation). *Given correspondence $c = \langle 1\#e, 2\#e', r, n \rangle$ between ontologies \mathcal{O}_1 and \mathcal{O}_2 , the natural translation t_n of c is defined as*

$$t_n(c) \mapsto \begin{cases} 1\#e \equiv 2\#e' & \text{if } r = \equiv \\ 1\#e \sqsubseteq 2\#e' & \text{if } r = \sqsubseteq \\ 1\#e \sqsupseteq 2\#e' & \text{if } r = \sqsupseteq \end{cases}$$

Now let us briefly recall the notion of ontology incoherence. An ontology \mathcal{O} is incoherent, iff there exist an unsatisfiable concept in \mathcal{O} . Analogous, a mapping \mathcal{M} between \mathcal{O}_1 and \mathcal{O}_2 is called incoherent due to t , if there exists an unsatisfiable concept $i\#C_{i \in \{1,2\}}$ in $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ that is satisfiable in \mathcal{O}_i . If there exists such a concept, its unsatisfiability must have (at least partially) been caused by \mathcal{M} .

Definition 4 (Incoherence of a Mapping). *Given a mapping \mathcal{M} between ontologies \mathcal{O}_1 and \mathcal{O}_2 and a translation function t . If there exists a concept $i\#C$ with $i \in \{1,2\}$ such that $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2 \models \perp \sqsupseteq i\#C$ and $\mathcal{O}_i \not\models \perp \sqsupseteq i\#C$ then \mathcal{M} is incoherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t . Otherwise \mathcal{M} is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t .*

Obviously, the incoherence of a mapping is strongly affected by our choice of t . In the following we use $t = t_n$ as translation function. Notice that it is possible to define an alternative translation function. In particular, it will turn out that the measures introduced in section 5 are independent of this choice with respect to their applicability, although their results are affected by the choice of the translation function.

4 The Importance of Mapping Coherence

One might argue, that mapping coherence is only important in a very specific application scenario like reasoning in a merged ontology. In the following we show that incoherence has a negative effect on a wide range of relevant applications. In [10] four different purposes of using ontology mappings have been distinguished. A more fine-grained distinction has been proposed in [4], but most of these scenarios can be subsumed under one of these use cases.

- *Frameworks.* Mappings are described in frameworks on an abstract level independent of an intended use.
- *Terminological Reasoning.* Mappings are used to perform reasoning across aligned ontologies.
- *Data Transformation.* Data from one ontology is transferred into the terminology of another ontology based on the knowledge encoded in a mapping.
- *Query Processing.* Queries formulated with respect to a certain ontology are translated into the terminology of a different ontology.

The *Frameworks* use case is about describing mappings on an abstract level. Since we try to argue for the applicability of our approach in a practical context, it is of minor interest and will not be discussed. It is obvious that incoherence is undesirable in the *Terminological Reasoning* case as incoherence will lead to inconsistency of the whole ontology when instances are added to unsatisfiable concepts. Inconsistency, however, disables meaningful reasoning as everything can be derived from an inconsistent ontology. It is less obvious that coherence is important for the *Data Transformation* and *Query Processing* use cases. In the following, we show that an incoherent mapping will lead to serious errors in the context of data translation and query processing.

4.1 Data Transformation

To better understand the effects of incoherence in the context of *Data Transformation* let us consider the following example. Suppose there are two companies C_1 and C_2 . Both use different ontologies, say \mathcal{O}_1 and \mathcal{O}_2 , to describe human resources and related topics. Now it happens that C_2 takes over C_1 . C_2 decides to migrate all instance data of \mathcal{O}_1 into \mathcal{O}_2 . \mathcal{O}_1 will no longer be maintained. A terminological mapping \mathcal{M} between \mathcal{O}_1 and \mathcal{O}_2 has to be created to migrate the instances of \mathcal{O}_1 to \mathcal{O}_2 in a fully automated way.

Fragments of ontologies \mathcal{O}_1 and \mathcal{O}_2 are depicted in figure 1. We refer to these fragments throughout the whole section without explicitly mentioning it in the following. *Data Transformation* can be roughly described as the following procedure.

1. For all instances a of \mathcal{O}_1 create a copy a' of this instance in \mathcal{O}_2 .
2. For all concept correspondences $\langle 1\#e, 2\#e', r, n \rangle \in \mathcal{M}$ with $r \in \{\equiv, \sqsubseteq\}$ and for all instances a with $\mathcal{O}_1 \models 1\#e(a)$ add axiom $2\#e'(a')$ to \mathcal{O}_2 .
3. For all property correspondences $\langle 1\#e, 2\#e', r, n \rangle \in \mathcal{M}$ with $r \in \{\equiv, \sqsubseteq\}$ and for all instances a, b with $\mathcal{O}_1 \models 1\#e(a, b)$ add axiom $2\#e'(a', b')$ to \mathcal{O}_2 .

In the following we refer to the ontology resulting from migrating instances from \mathcal{O}_i to \mathcal{O}_j based on mapping \mathcal{M} as $\mathcal{O}_j \cup \mathcal{M}(\mathcal{O}_i)$. At first sight, mapping coherence seems to be irrelevant with respect to this use case, because we do not copy any of the terminological axioms into \mathcal{O}_j . Consider the following correspondences to understand why this impression is deceptive.

$$\langle 1\#Person, 2\#Person, \equiv, 1.0 \rangle \quad (1)$$

$$\langle 1\#ProjectLeader, 2\#Project, \sqsubseteq, 0.6 \rangle \quad (2)$$

Let now mapping \mathcal{M} contain correspondence (1) and (2). \mathcal{M} is incoherent, due to the fact that in $\mathcal{O}_1 \cup_{\mathcal{M}} \mathcal{O}_2$ concept $1\#ProjectLeader$ becomes unsatisfiable. Concepts $2\#Project$ and $2\#Person$ are disjoint and due to \mathcal{M} concept $1\#ProjectLeader$ is subsumed by both of them, resulting in its unsatisfiability. Suppose now, there exists an instance a with $1\#ProjectLeader(a)$. Applying the migration rules results in a new instance a' with $2\#Project(a')$ and $2\#Person(a')$. Due to the disjointness of $2\#Project$ and $2\#Person$ there exists no model for $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$ and thus $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$ is an inconsistent ontology. Opposed to our first impression there seems to be a tight link between *the incoherence of \mathcal{M}* and *the inconsistency of $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$* .

Contrary to this, mappings can be constructed, where such a direct link cannot be detected. Let \mathcal{M} , for example, contain correspondences (3) and (4). \mathcal{M} is incoherent due to the unsatisfiability of $2\#ProductLine$ in the merged ontology.

$$\langle 1\#Deadline, 2\#TimedEvent, \sqsubseteq, 0.9 \rangle \quad (3)$$

$$\langle 1\#ProjectDeadline, 2\#ProductLine, \sqsupseteq, 0.7 \rangle \quad (4)$$

Now we have to acknowledge that both $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1)$ and $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)$ do not become inconsistent. But what happens if we first transfer all instances x of \mathcal{O}_2 to $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)$ and then again transfer the x' instances of $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)$ to $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$?

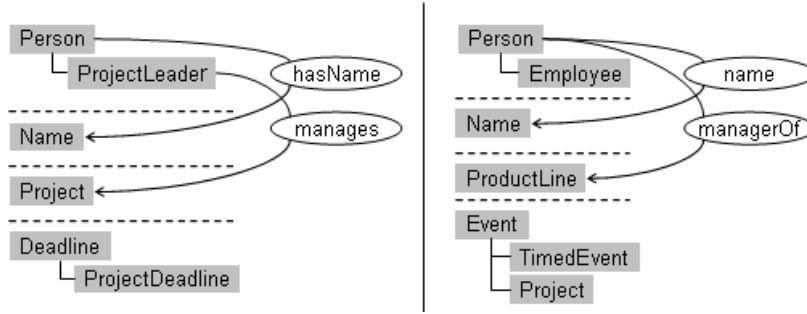


Fig. 1. Fragments of ontologies \mathcal{O}_1 (on the left) and \mathcal{O}_2 (on the right). A square represents a concept, an ellipse a property, subsumption is represented by indentation. Domain and range of a property are restricted to be the concepts connected by the accordant arrow. Dashed horizontal lines represent disjointness between concepts.

Given some instance a with $\mathcal{O}_2 \models 2\#ProductLine(a)$ after the first step we have the counterpart of a , namely a' , with $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2) \models 1\#ProjectDeadline(a')$ by applying correspondence (4) and can derive $\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2) \models 1\#Deadline(a')$. After the second step we have $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2)) \models 2\#TimedEvent(a'')$ by applying correspondence (3). We expect that adding axiom $a = a''$ does not affect the consistency of $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$. But $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$ now implies $2\#ProductLine(a)$ as well as $2\#Event(a)$. Since $2\#ProductLine$ and $2\#Event$ are defined to be disjoint, there exists no model for $\mathcal{O}_2 \cup \mathcal{M}(\mathcal{O}_1 \cup \mathcal{M}(\mathcal{O}_2))$. Again, we find a strong link between mapping incoherence and inconsistency after instance migration.

4.2 Query Processing

In the following we revisit a variant of the example given above to better understand the use case of *Query Processing*. Again, company C_2 takes over C_1 . But this time both \mathcal{O}_1 and \mathcal{O}_2 are maintained. Instead of migrating all instances from \mathcal{O}_1 to \mathcal{O}_2 queries are rewritten at runtime to enable information integration between \mathcal{O}_1 and \mathcal{O}_2 . A terminological mapping is the key for information integration. It is used for processing queries and generating result sets which contain data from both ontologies. As we are concerned with theoretical issues, we argue on an abstract level instead of discussing e.g. characteristics of a SPARQL implementation. A query language for DL based knowledge bases should at least support instance retrieval for complex concept descriptions. Depending on the concrete query language there might be a complex set of rewriting rules. At least, it must contain variants of the following two rules.

- R_1 : Let $i\#C$ and $i\#D$ be concept descriptions in the language of \mathcal{O}_i . If $\mathcal{O}_i \models i\#C \equiv i\#D$, then query q can be transformed into an equivalent query by replacing all occurrences of $i\#C$ by $i\#D$.
- R_2 : Let $i\#C$ and $j\#D$ be concept descriptions in the language of \mathcal{O}_i , respectively \mathcal{O}_j . If there exists a correspondence $\langle i\#C, j\#D, \equiv, n \rangle \in \mathcal{M}$, then query q can be transformed into an equivalent query by replacing all occurrences of $i\#C$ by $j\#D$.

Suppose we query for the name of all project leaders, formally speaking we are interested in the instances of $\exists 1\#hasName^{-1}.1\#ProjectLeader$. To receive instances of both \mathcal{O}_1 and \mathcal{O}_2 we have to rewrite the query for \mathcal{O}_2 . Now let \mathcal{M} contain correspondences (5), (6), and (7).

$$\langle 1\#hasName, 2\#name, \equiv, 0.9 \rangle \quad (5)$$

$$\langle 1\#Project, 2\#Project, \equiv, 1.0 \rangle \quad (6)$$

$$\langle 1\#manages, 2\#managerOf, \equiv, 0.7 \rangle \quad (7)$$

Suppose that \mathcal{O}_1 contains axiom $1\#ProjectLeader \equiv \exists 1\#manages.1\#Project$. We exploit this axiom by applying R_1 . Now for every concept and property name that occurs in $\exists 1\#hasName^{-1}.1\#ProjectLeader$, there exists a direct counterpart in \mathcal{O}_2 specified in \mathcal{M} . By applying R_2 we thus finally end with a concept description in the language of \mathcal{O}_2 .

$$\exists 1\#hasName^{-1}.1\#ProjectLeader \quad (8)$$

$$\xleftrightarrow{R_1} \exists 1\#hasName^{-1}.\exists 1\#manages.1\#Project \quad (9)$$

$$\xleftrightarrow{R_2} \exists 2\#name^{-1}.\exists 2\#managerOf.2\#Project \quad (10)$$

What happens if we process the query based on this concept description to \mathcal{O}_2 ? As result we receive the empty set. The range of $2\#managerOf$ is concept $2\#ProductLine$, and $2\#ProductLine$ is defined to be disjoint with $2\#Project$. Thus, for logical reasons there exists no instance of concept description (10) in \mathcal{O}_2 .

This problem is obviously caused by the incorrectness of correspondence (7). But the incorrectness of (7) does not only affect the query under discussion. It also causes mapping \mathcal{M} to become incoherent, because in the merged ontology $\mathcal{O}_1 \cup_{\mathcal{M}} \mathcal{O}_2$ concept $1\#ProjectLeader$ becomes unsatisfiable due to its equivalence with concept description $\exists 1\#manages.1\#Project$. This time we find a strong link between *mapping incoherency* and the *incorrectness of a query result* due to processing the mapping.

5 Measuring Incoherence

The definition of mapping incoherence given above is a boolean criterion that only distinguishes between coherent and incoherent mappings. Contrary to this, an incoherence measure should satisfy $m(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) > m(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}')$ if \mathcal{M} has a *higher degree of incoherence* than \mathcal{M}' . At the moment we might have an intuitive understanding of different degrees of incoherence, but a precise definition has to be given in the following subsections. Up to now, we define an incoherence measure to satisfy the following constraint.

Definition 5 (Incoherence Measure). *Let \mathcal{M} be a mapping between ontologies \mathcal{O}_1 and \mathcal{O}_2 and let t be an translation function. An incoherence measure m^t maps $\mathcal{O}_1, \mathcal{O}_2$, and \mathcal{M} to a value in $[0, 1]$ such that $m^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = 0$ iff \mathcal{M} is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t .*

In the following we distinguish between effect-based and revision-based measures. The former are concerned with the negative impact of mapping incoherence. The latter measure the effort necessary to revise a mapping by removing incoherences. Both approaches make it possible to extend the boolean property of incoherence to a continuous measure of its degree.

5.1 Measuring the Impact of Incoherence

The first measure to be introduced is based on the idea of counting unsatisfiable concepts. It is an adaption of an ontology incoherence measure introduced in [11]. Before we proceed, we need to agree on some abbreviations and naming conventions.

Definition 6. *Let \mathcal{O} be an ontology. Then $CO(\mathcal{O})$ refers to the set of named concepts in \mathcal{O} and $US(\mathcal{O}) = \{C \in CO(\mathcal{O}) \mid \mathcal{O} \models C \sqsubseteq \perp\}$ refers to the set of unsatisfiable concepts in \mathcal{O} .*

Contrary to measuring incoherences in ontologies, we have to distinguish between two types of concept unsatisfiability in the merged ontology: There are unsatisfiable concepts in $\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2$ which have already been unsatisfiable in \mathcal{O}_1 , respectively \mathcal{O}_2 , while there are unsatisfiable concepts which have been satisfiable in \mathcal{O}_1 , respectively \mathcal{O}_2 . We are interested in the latter concepts. In particular, we compare the number of these concepts with the number of all named concepts satisfiable in \mathcal{O}_1 or \mathcal{O}_2 .

Definition 7 (Unsatisfiability Measure). *Let \mathcal{M} be a mapping between ontologies \mathcal{O}_1 and \mathcal{O}_2 , and let t be a translation function. Unsatisfiability measure m_{sat}^t is defined by*

$$m_{sat}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{|US(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (US(\mathcal{O}_1) \cup US(\mathcal{O}_2))|}{|CO(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (US(\mathcal{O}_1) \cup US(\mathcal{O}_2))|}$$

This measure can be criticised for the following reason. Suppose again, we have an incoherent mapping \mathcal{M} for ontologies \mathcal{O}_1 and \mathcal{O}_2 depicted in figure 1. Suppose that due to \mathcal{M} concept $1\#Person$ becomes unsatisfiable in the merged ontology. As a consequence concept $1\#ProjectLeader$ becomes unsatisfiable, too. By applying definition 7 we thus measure both direct impact (unsatisfiability of $1\#Person$) and indirect impact (unsatisfiability of $1\#ProjectLeader$) of \mathcal{M} , even though we might only be interested in the direct impact. The distinction between root and derived unsatisfiability, as introduced in [6], solves this problem. A precise definition requires us to recall the notion of a MUPS, defined in [12], which is minimal unsatisfiability preserving sub-TBox.

Definition 8 (MUPS). *Let \mathcal{O} be an ontology, let $\mathcal{T} \subseteq \mathcal{O}$ be the TBox of \mathcal{O} , and let $C \in US(\mathcal{O})$. A set $\mathcal{T}' \subseteq \mathcal{T}$ is a minimal unsatisfiability preserving sub-TBox (MUPS) in \mathcal{T} for C if C is unsatisfiable in \mathcal{T}' and C is satisfiable in every $\mathcal{T}'' \subset \mathcal{T}'$. The set of all MUPS with respect to C is referred to as $mups(\mathcal{O}, C)$.*

A MUPS for a concept C can be seen as a minimal explanation of its unsatisfiability. Whenever there exists another unsatisfiable concept D such that the minimal explanation of C 's unsatisfiability also explains the unsatisfiability of D then C is referred to as derived unsatisfiable concept, because one reason for C 's unsatisfiability is the unsatisfiability of D .

Definition 9 (Derived and Root Unsatisfiability). Let \mathcal{O} be an ontology and let $C \in US(\mathcal{O})$. C is a derived unsatisfiable concept if there exists $D \neq C \in US(\mathcal{O})$ such that there exist $M \in \text{mups}(\mathcal{O}, C)$ and $M' \in \text{mups}(\mathcal{O}, D)$ with $M \supseteq M'$. Otherwise C is a root unsatisfiable concept. The set of derived unsatisfiable concepts of \mathcal{O} is referred to as $US_D(\mathcal{O})$ and the set of root unsatisfiable concepts is referred to as $US_R(\mathcal{O})$.

Similar to the unsatisfiability measure we can now introduce the root unsatisfiability measure by considering only root unsatisfiable concepts instead of all unsatisfiable concepts in the merged ontology.

Definition 10 (Root Unsatisfiability Measure). Let \mathcal{M} be a mapping between ontologies \mathcal{O}_1 and \mathcal{O}_2 , and let t be a translation function. Root unsatisfiability measure m_{rsat}^t is defined by

$$m_{rsat}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{|US_R(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (US(\mathcal{O}_1) \cup US(\mathcal{O}_2))|}{|CO(\mathcal{O}_1 \cup_{\mathcal{M}^t} \mathcal{O}_2) \setminus (US(\mathcal{O}_1) \cup US(\mathcal{O}_2))|}$$

Obviously, we have $m_{sat}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) \geq m_{rsat}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M})$ for each mapping \mathcal{M} between two ontologies \mathcal{O}_1 and \mathcal{O}_2 . As argued above, the m_{rsat}^t measure has to be preferred. Nevertheless, a non trivial algorithm is required to compute the set of root unsatisfiable concepts as described in [6] which makes the application of this measure more expensive from a computational point of view.

5.2 Measuring the Effort of Mapping Revision

The second type of measure is concerned with the effort of revising incoherent mappings. We use the term revision to describe the process of removing correspondences from an incoherent mapping until a coherent submapping has been found. If a revision is conducted by a domain expert we would, for example, be able to measure the time necessary. Since we want our measure to be computable without any human intervention, we thus have to think of an automated strategy to revise a mapping. Such a strategy should obviously remove a minimum number of correspondences, because we would like to keep as much information in the mapping as possible. The following measure is based on this idea and compares the number of correspondences that would be removed by such a strategy with the number of all correspondences in the mapping.

Definition 11 (Maximum Cardinality Measure). Let \mathcal{M} be a mapping between ontologies \mathcal{O}_1 and \mathcal{O}_2 , and let t be a translation function. Maximum cardinality measure m_{card}^t is defined by

$$m_{card}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{|\mathcal{M} \setminus \mathcal{M}'|}{|\mathcal{M}|}$$

where $\mathcal{M}' \subseteq \mathcal{M}$ is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t and there exists no $\mathcal{M}'' \subseteq \mathcal{M}$ with $|\mathcal{M}''| > |\mathcal{M}'|$ such that \mathcal{M}'' is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t .

Suppose there are incoherent mappings \mathcal{M}_1 and \mathcal{M}_2 with $|\mathcal{M}_1| = |\mathcal{M}_2| = 10$. Further suppose, according to the naming convention in definition 11, we have $|\mathcal{M}'_1| = 8$

and $|\mathcal{M}'_2| = 7$. Thus, we have $m_{card}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}_1) = 0.2$ and $m_{card}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}_2) = 0.3$. But now suppose that all of the correspondences in \mathcal{M}_1 have the same confidence value, say 1. Contrary to this, \mathcal{M}_2 differs with respect to the confidence value of its correspondences. In particular, it turns out that all of the three correspondences that have been removed have a very low confidence value compared to the remaining correspondences. The following definition introduces the maximum trust measure which is similar to the maximum cardinality measure but accounts for differences in the confidence distribution.

Definition 12 (Maximum Trust Measure). *Let \mathcal{M} be a mapping between ontologies \mathcal{O}_1 and \mathcal{O}_2 , and let t be a translation function. Further, let $conf : \mathcal{M} \rightarrow [0, 1]$ be a function that maps a correspondence on its confidence value. Maximum trust measure m_{trust}^t is defined by*

$$m_{trust}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M}) = \frac{\sum_{c \in \mathcal{M} \setminus \mathcal{M}'} conf(c)}{\sum_{c \in \mathcal{M}} conf(c)}$$

where $\mathcal{M}' \subseteq \mathcal{M}$ is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t and there exists no $\mathcal{M}'' \subseteq \mathcal{M}$ with $\sum_{c \in \mathcal{M}''} conf(c) > \sum_{c \in \mathcal{M}'} conf(c)$ such that \mathcal{M}'' is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t .

This measure is derived from the algorithm already described in [9] which can be used to compute \mathcal{M}' for a specific type of mappings. Namely, one-to-one mappings that contain only correspondences expressing equivalences between concepts. We also used it in the context of mapping extraction [8]. Its application is motivated by the idea that $\mathcal{M} \setminus \mathcal{M}'$ mainly contains incorrect correspondences given an appropriate confidence distribution. We adapted this idea to introduce the m_{trust}^t measure as confidence weighted complement to the m_{card}^t measure.

Notice that computing both of these measures requires to solve computational hard problems. On the one hand only full-fledged reasoning guarantees completeness in detecting unsatisfiability. On the other hand the underlying problem is the optimization problem of finding a hitting set $H \subseteq \mathcal{M}$ of minimal cardinality (respectively minimal confidence total) over the set all of minimal incoherent subsets of \mathcal{M} . This problem is known to be NP-complete [7]. Nevertheless, first experiments indicate that both measures can be computed in acceptable time for small and medium sized ontologies.

6 Truth and Coherence

In the following we are concerned with an important interrelation between the classical compliance measure of precision and the maximum cardinality measure of incoherence. Philosophically speaking, we are interested in how far an incoherent mapping can truly express semantic relations between ontological entities. Accordant to [1], the precision of a mapping can be defined as follows.

Definition 13 (Precision). *Given a mapping \mathcal{M} between ontologies \mathcal{O}_1 and \mathcal{O}_2 , let \mathcal{R} be a reference mapping between \mathcal{O}_1 and \mathcal{O}_2 . The precision of \mathcal{M} with respect to \mathcal{R} is defined as $precision(\mathcal{M}, \mathcal{R}) = |\mathcal{M} \cap \mathcal{R}| / |\mathcal{M}|$.*

In section 4 we argued that an incoherent mapping \mathcal{M} causes different kinds of problems when applied in a realistic scenario. More precisely, it is the incorrectness of a correspondence that causes both the problem as well as the incoherence. Even though incorrect correspondences not necessarily result in incoherence, we can be sure that an incoherent mapping contains at least one incorrect correspondence. Thus, a well-modeled reference mapping \mathcal{R} will be coherent due to each translation function t compatible with the mapping semantics accepted by the person who created \mathcal{R} . The following proposition corresponds to this consideration.

Proposition 1 (Incoherence and Precision). *Let \mathcal{R} be a reference mapping between \mathcal{O}_1 and \mathcal{O}_2 . Further let mapping \mathcal{M} be incoherent and let \mathcal{R} be coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to translation function t . Then we have $precision(\mathcal{M}, \mathcal{R}) < 1$.*

Proof. Given the coherence of \mathcal{R} , it can be concluded that every subset of \mathcal{R} is coherent, too. Since \mathcal{M} is incoherent, it is thus no subset of \mathcal{R} , i.e. $\mathcal{M} \setminus \mathcal{R} \neq \emptyset$. We conclude that $\mathcal{M} \cap \mathcal{R} \subset \mathcal{M}$. It follows directly $precision(\mathcal{M}, \mathcal{R}) < 1$.

Notice that automatically generated mappings normally do not have a precision of 1. Thus, the application of proposition 1 is only of limited benefit. Nevertheless, it can be generalized in a non trivial way by exploiting the definition of the maximum cardinality measure (definition 11). This generalization allows us to compute a non trivial upper bound for mapping precision without any knowledge of \mathcal{R} .

Proposition 2 (Upper Bound for Precision). *Let \mathcal{M} be a mapping and \mathcal{R} be a reference mapping between \mathcal{O}_1 and \mathcal{O}_2 . Further let \mathcal{R} be coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to translation function t . Then we have $precision(\mathcal{M}, \mathcal{R}) \leq 1 - m_{card}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M})$.*

Proof. Accordant to definition 11 let $\mathcal{M}' \subseteq \mathcal{M}$ be the coherent subset of \mathcal{M} with maximum cardinality. Further let be $\mathcal{M}^* = \mathcal{M} \cap \mathcal{R}$, i.e. \mathcal{M}^* consist of all correct correspondences in \mathcal{M} . Since \mathcal{M}^* is a subset of \mathcal{R} and \mathcal{R} is coherent with respect to \mathcal{O}_1 and \mathcal{O}_2 due to t , we conclude that \mathcal{M}^* is also coherent. It follows that $|\mathcal{M}^*| \leq |\mathcal{M}'|$, because otherwise \mathcal{M}' would not be the coherent submapping of maximum cardinality contrary to definition 11. In summary, the following inequation holds.

$$precision(\mathcal{M}, \mathcal{R}) = \frac{|\mathcal{M} \cap \mathcal{R}|}{|\mathcal{M}|} = \frac{|\mathcal{M}^*|}{|\mathcal{M}|} < \frac{|\mathcal{M}'|}{|\mathcal{M}|} = 1 - \frac{|\mathcal{M}' \setminus \mathcal{M}|}{|\mathcal{M}|} = m_{card}^t(\mathcal{O}_1, \mathcal{O}_2, \mathcal{M})$$

Proposition 2 reveals an important interrelation between coherence and precision. The counterpart of mapping precision is the measure of recall. At first glimpse it seems that recall and coherence describe independent properties of a mapping. Nevertheless, there exists a non trivial relation between recall and coherence that allows to derive comparative statements about the relative recall of two overlapping mappings in some cases. Although this interrelation is not as significant as the the one expressed in proposition 2, further theoretical considerations are required.

The utility of proposition 2 in the evaluation process essentially depends on the distance between the upper bound and the actual value of mapping precision. In particular, measuring low values for the m_{card}^t measure will lead to a poor differentiation with respect to the precision that has to be expected. In initial experiments, not included in this paper due to lack of space, first results indicate that the upper bound for mapping precision strongly varies and can be used to filter out highly imprecise mappings.

7 Conclusion

In this paper, we have discussed the notion of incoherence of ontology mappings and its role in the assessment of automatically created mappings. There are two main conclusions of this work. First, we conclude that incoherence is an important aspect of mapping quality as incoherent mappings have undesirable effects on most relevant application scenarios as we have demonstrated in section 4. Second, appropriate measures of incoherence can help to assess the quality of a mapping even if no reference mapping is available and thus precision and recall cannot be determined. In particular, the measure of incoherence provided in definition 11 provides a strict upper bound for the precision of a mapping and can therefore be used as a guideline for estimating the performance of matching systems. In future work experimental studies will show in how far the proposed measures can be effectively applied in the evaluation process.²

Acknowledgement The work has been partially supported by the German Science Foundation (DFG) in the Emmy Noether Programme under contract STU 266/3-1.

References

1. Hong-Hai Do, Sergey Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In *Proc. of the GI-Workshop Web and Databases*, Erfurt, Germany, 2002.
2. Marc Ehrig and Jerome Euzenat. Relaxed precision and recall for ontology matching. In *Proc. of the K-Cap 2005 Workshop on Integrating Ontology*, Banff, Canada, 2005.
3. Jerome Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007.
4. Jerome Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.
5. Peter Haase and Guilin Qi. An analysis of approaches to resolving inconsistencies in DL-based ontologies. In *Proc. of the International Workshop on Ontology Dynamics*, Innsbruck, Austria, 2007.
6. Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James Hendler. Debugging unsatisfiable classes in OWL ontologies. *Journal of Web Semantics*, 2005.
7. Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum, 1972.
8. Christian Meilicke and Heiner Stuckenschmidt. Analyzing mapping extraction approaches. In *Proc. of the ISWC 2007 Workshop on Ontology Matching*, Busan, Korea, 2007.
9. Christian Meilicke and Heiner Stuckenschmidt. Applying logical constraints to ontology matching. In *Proc. of the 30th German Conference on Artificial Intelligence*, Osnabrück, Germany, 2007.
10. Natasha Noy and Heiner Stuckenschmidt. Ontology alignment: An annotated bibliography. In *Semantic Interoperability and Integration*, Dagstuhl, Germany, 2005.
11. Guilin Qi and Anthony Hunter. Measuring incoherence in description logic-based ontologies. In *Proc. of the 6th International Semantic Web Conference*, 2007.
12. Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *Proc. of 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.

² A first beta-version of our system supporting the measures m_{sat}^t , m_{card}^t , and m_{trust}^t based on a slightly modified natural translation can be obtained on request.