Reasoning about Taxonomies and Articulations

David Thau[*]
Advisor: Bertram Ludäscher
University of California at Davis
1 Shields Avenue
Davis, California
dthau@ucdavis.edu

ABSTRACT

Taxonomically organized data pervade science, business and everyday life. Unfortunately, taxonomies are often underspecified, limiting their utility in contexts such as data integration, information navigation and autonomous agent communication. This work formalizes taxonomies and relationships between them as formulas in logic. This formalization concretizes notions such as *consistency* and *inconsistency* of taxonomies and articulations (inter-taxonomic relations) between them, enables the derivation of new articulations based on a given set of taxonomies and articulations and provides a framework for testing assumptions about underspecified taxonomies.

Given the typical intractability of reasoning with taxonomies and articulations, this research investigates many optimizations: from those that reduce the search space, to those that leverage parallel processing, to those investigating logics more tractable than first-order logic (*e.g.*, monadic first-order logic, propositional logic, description logics, and subsets of the RCC-5 spatial algebra). Finally, in addition to reasoning with taxonomies and articulations, this research investigates how to repair inconsistent taxonomies and articulations, how to explain inconsistencies and discovered relations, and how to merge taxonomies given articulations. Critical to this research is the development of a framework for testing logics and support for the development of taxonomies and articulations. This framework, CleanTax is already well under way and has been used to study articulations between two large-scale biological taxonomies.

1. INTRODUCTION

Humans classify, and taxonomies are one of the most natural forms of classification. Taxonomies pervade our lives, from the Dewey Decimal system of book classification, to business org-charts, to the tree of life. Taxonomies are prevalent in science and engineering, where data are often organized hierarchically. Because taxonomies are "views" of data, different taxonomies organizing the same set of data often arise; due to changing domain information or differing expert opinion. These varying taxonomies can make data integration difficult, especially when data are distributed or organized using different but related schemas. Integrating data organized by alternative taxonomies requires, in addition to the given taxonomies, a set of articulations indicating how the concepts (taxa, classes) in the different taxonomies relate to each other. A set of taxonomies and articulations between them is called an *alignment.* The process of discovering an alignment given a set of taxonomies is called the *taxonomy alignment problem.*

Taxonomies can be large, potentially necessitating a great number of articulations. For example, a recent analysis of treatments of the plant genus *Ranunculus* [23] considered 9 taxonomies, covering 654 concepts (called taxa in biology) and 704 articulations. Although the taxonomies and articulations are syntactically correct according to a standard XML schema, the semantics of the taxonomies and articulations are not well described.

This research addresses reasoning about taxonomies and articulations drawn between concepts in multiple taxonomies. Topics within this focus range from investigations into modeling taxonomies and articulations, the use of automatic reasoners to detect inconsistencies introduced by articulations, optimizations for calculating the deductive closure of a set of taxonomies and articulations, algorithms for determining fixes for inconsistent taxonomies and articulations, and mechanisms for merging taxonomies.

1.1 Motivating Scenario

Consider a biologist faced with combining species occurrence data from multiple studies. Species are organized hierarchically, and the definitions of species names change over time, so data sets using different biological taxonomies may classify species differently [17]. If the data are classified using well-known taxonomies, and an expert has indicated how concepts in the various taxonomies relate, it may be possible to automatically integrate the data. Before this is possible, the articulations between the taxonomies must be drawn.

To create the articulations, a metadata curator must consider multiple taxonomies and draw the articulations between their concepts. While creating articulations between these taxonomies, the curator will want to know (i) when a newly proposed articulation leads to an impossible situation, (ii) if so, why, and how the inconsistency may be repaired, (iii) if the proposed articulations entail other articulations, and (iv) if an unexpected articulation is entailed, how was

[*] Expected Date of Completion: September 2009.

EDBT 2008 Ph.D. Workshop '08, March 25, 2008, Nantes, France

it derived.

All of these questions depend in part on the taxonomies being compared. Unfortunately, taxonomies are frequently under-specified, described only by the subsumption relationships of the concepts. Taxonomies often carry additional unstated (but assumed) constraints, such as that concepts are composed of the disjoint union of their children. These constraints may impact the logical consistency of articulations, as well as the entailment of additional articulations. Whether or not the expected constraints hold should be checked before the taxonomy is used in a data integration task or when determining which articulations are logically sensible.

In general, an articulator will prefer articulations that minimize ambiguity, choosing unambiguous articulations such as "concept A and concept B are congruent" to ambiguous ones, such as "concept A and concept B are either congruent, or A is a subset of B." The amount of uncertainty in the relationship between two concepts will be in part determined by the additional constraints acting on the taxonomies.

The work described below lays the groundwork for the development of tools to help a metadata curator ensure that taxonomies adhere to expected constraints, and to construct sensible and unambiguous articulations between concepts.

1.2 Domain Description

Traditionally, taxonomies have been defined as a partial ordering of concepts where the ordering relation denotes some sort of "inclusion" relation [9]. The partial ordering relation is often called an "isa" relation, and represents a rule of the form: A isa B means that if instance x is an example of A, then x is also an example of B. In biological taxonomies, this translates to rules like, "if Fido is an instance of Canis lupus, then Fido is an instance of Canis." In a phylogeny, the rule might be "if Fido is an instance of the things descended from ancestor A, then it is also an instance of things descended from ancestor B." In a business org-chart, the rule might be "if employee x is a member of the software engineering department, then employee x is also a member of the engineering department."

This definition is very general. Partial orders may be strict (the ordering relation is irreflexive, asymmetric and transitive) or non-strict (the ordering relation is reflexive, transitive and antisymmetric). Some taxonomies permit multiple inheritance while others do not. In some taxonomies, child concepts partition their parents, while in others, parent concepts may contain instances not contained in their children.

In addition to being too general, the definition of a taxonomy as a partial order does not describe how taxonomies are actually defined. Taxonomies may be defined entirely as graphs, with edges representing inclusion relations and nodes representing taxonomic concepts. In other cases, such as in Formal Concept Analysis (FCA) [12], taxonomies are defined using the properties of the instances of the concepts. In yet other cases, taxonomies are defined using characteristics of the concepts, without reliance on instances. Sometimes a combination of these occurs [10].

When given two or more taxonomies organizing similar data, it is natural to wonder how the taxonomies relate. One method of comparing taxonomies is to describe the relationships between concepts in each taxonomy. The vocabulary used to describe these relationships can be very simple or quite complex, depending on the language used to express the relationships, and the taxonomies themselves.

1.3 Outstanding Problems

Taxonomies may be seen as simplified ontologies [20]. The primary difference between a taxonomy and an ontology is that in a taxonomy the relations between concepts are considerably more restricted. For example, the concepts in taxonomies must be related by a partial ordering relation, while this is not the case with ontologies. However, many of the problems that apply to the comparison of ontologies (often called the ontology matching, or ontology alignment, problem [28]) also apply in the realm of taxonomies:

- *Representation.* How are taxonomies and articulations represented?
- *Representation of uncertainty.* How should uncertainty in the relationship between two concepts be represented?
- *Consistency of taxonomies.* Given a formal definition of taxonomy, is a given set of concepts and relations a valid taxonomy?
- *Models.* Does a set of instances and information about how those instances are sorted into the concepts of a taxonomy describe a model that satisfies a given taxonomy?
- *Consistency of articulations.* Given a set of consistent taxonomies, and articulations between their concepts, are there any contradictory assertions?
- *Explanation of Inconsistency.* From where do the contradictions arise?
- *Repairs.* How can contradictions be repaired?
- *Inference.* Are any unstated relations implied?
- *Explanations of discovered relations.* If unstated relations are discovered, how were they derived?
- *Quality.* How can alignment quality be quantified?
- *Improvement.* How can alignment quality be improved?
- *Minimality.* Are the given relations within a taxonomy, or between articulated taxonomies, a minimal set, or can some be removed while entailing the same relations?
- *Maximal/minimal consistent/inconsistent subset.* What are the maximal consistent or minimal inconsistent subsets of concepts and relations in single taxonomies or articulated taxonomies?
- *Merge.* Is there a single canonical representation of the combination of two or more taxonomies, given relationships between them?

1.4 Current Solutions

The alignment problem has been studied in various guises: from database schema matching [24], to XML schema and document alignment [21], to ontology alignment in the Semantic Web [22, 15, 27, 29, 28]. This research spans a vast landscape of techniques and scenarios. Some basic differentiators are techniques that rely on instances [30], versus those that do not, such as PROMPT [22]. Some techniques rely on a lexical analysis of the concept names to support alignment [11], while others focus on structural elements such as the relations between concepts [14]. The techniques used will largely depend on the domain of the ontologies. Like [14], the current research focuses entirely on the structure of the relations between concepts in the taxonomies being aligned. This focus may be seen as a point of departure from which investigations into the effect of instances and lexical similarities may be investigated.

Another key differentiator between solutions to the align-

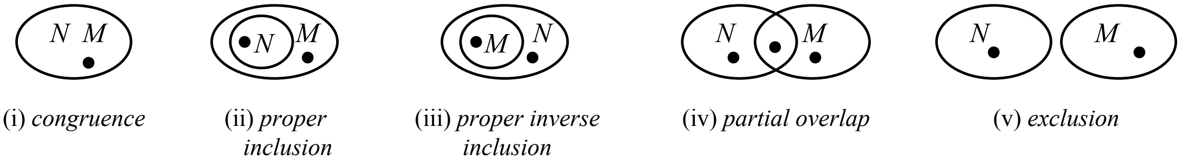

Figure 1: $\mathbb{B}_5$ – the five basic relations $N \equiv M$, $N \subsetneq M$,$N \supsetneq M$, $N \oplus M$, $N \,!\, M$ between two sets N, M

ment problem involves the languages used to express the data being aligned, and the language of the articulations. In general, an alignment problem takes two or more data structures to be aligned, $i_1 \dots i_n$, a set of previously stated articulations A, and a set of constraints on legal articulations C and returns a complete set of articulations A'. Each data structure, the previously stated articulations, the constraints, and the resulting set of articulations are stated in a language $L_{i_1} \dots L_{i_n}$, L_A, L_C, $L_{A'}$ where the languages are all compatible (there is a common language, such as first-order predicate calculus, into which statements in all languages involved may be converted).

For example, in [27] ontologies are represented in a modified description logic, and relations between concepts in the ontologies may be either subsumption, equivalence, or disjointness. In [14] on the other hand, ontologies are represented in propositional logic, and relations between concepts are modeled as implication, equivalence and disjointness. A third example [21] models relational databases and XML documents in a nested relational model and supports arbitrary *n:m* transformations between concepts in the inputs. None of the above mentioned cases deals directly with incomplete knowledge of the relations between concepts (*e.g.*, even the most basic "isa" relation indicates incomplete knowledge when it represents "equals or included in.")

Taxonomies have also been studied outside the realm of knowledge representation - primarily in the life sciences. For example, in 2005, the Taxonomic Data Working Group [2], an international not-for-profit organization that develops standards for biodiversity data, ratified the Taxonomic Concept Schema (TCS) [1]. TCS is an XML Schema which defines a syntax for describing taxonomic concepts. The TCS includes a list of terms which may be used to define the relationships between two different taxonomic concepts. This list includes set-theoretic terms, such as *is congruent to* and *excludes.* However, some of the terms are not well defined. For example, it is unclear whether *is included in* means *proper subset* ($\subsetneq$), or if it means *subset* ($\subseteq$), which includes the possibility that both sets are equal. It also includes more vague relationships such as *has synonym.* These vague relationships are needed in TCS, as it aims to provide a standard for information providers to communicate information about their data. However, unless the meaning of such relations is specified more precisely, their utility for automated reasoning will be diminished.

Beach et al. [4] introduced, and Berendsohn [7] elaborated, the notion of a *potential taxon*, which identifies a taxonomic concept by referencing the context in which the name is used; *e.g.*, *Hypnum flagellare* Dicks. sec. MÖNKEMEYER 1927. This notion is central to the MoReTaX project [8], in which potential taxa are considered sets of objects, and the relationships between them are described in precise set-theoretic terms. The five so-called *basic relations* which may hold between any two potential taxa (or, in fact, any two non-empty sets) A and B are: (i) congruence ($A \equiv B$), (ii) proper inclusion ($A \subsetneq B$), (iii) proper inverse inclusion ($A \supsetneq B$), (iv) partial overlap ($A \oplus B$), and (v) exclusion (disjointness) ($A \,!\, B$) (see Figure 1). Geoffroy and Güntsch [13] study the problem of *propagating* knowledge about such binary relationships between taxa: *e.g.*, what can we say about the relationship between potential taxa A and C, provided we only know that $A \supsetneq B$ and $B \oplus C$? Inspection of all possibilities allows one to deduce that $A \supsetneq C$ or $A \oplus C$, but none of the other three options $\equiv$, $\subsetneq$, or ! is possible between A and C. Thus, the authors study *combined relationships* (*i.e.*, disjunctions of basic relations: *e.g.*, $\{\supsetneq, \oplus\}$) and demonstrate how these may be composed to propagate taxonomic knowledge in a *potential taxon graph.* A *generalized path* in such a graph bundles all existing *simultaneous paths* between two nodes (say A and C) and can employ 'strong agreement' (conjunction of expert knowledge on simultaneous paths) or 'weak agreement' (disjunction of such paths). Rules for knowledge propagation in a taxon graph are given as `if-then` rules, embedded in the MoReTaX system; thus, computing with taxon relations is handled programmatically.

2. PROBLEM STATEMENT

The current research focuses on the alignment problem in taxonomies. The questions described in section 1.3 are addressed by translating taxonomies and articulations between them into formulas in a variety of subsets of first-order logic (*e.g.*, monadic first-order logic, description logic, propositional logic), and then applying the machinery of logic to determine the consistency of the formulas and their deductive closure. The taxonomies being aligned are assumed to classify the same types of information (*e.g.*, books, plants, employees) and the languages used to describe a given set of taxonomies and articulations may all be translated into a common formal language. Part of the contribution of this research is to define taxonomy and articulation languages that may be translated into the targeted subsets of first-order logic.

In addition to addressing representation and reasoning questions, the work explores a variety of optimizations and presents CLEANTAX, an architecture for running large scale experiments, exploring a variety of languages and providing tools for supporting metadata curators as they build and test taxonomies and articulations.

3. INITIAL RESULTS

Initial work [31] focused on applying a subset of first-order logic (monadic first-order logic, $\mathcal{L}_{\text{MFOL}}$) to model biological taxonomies and articulations between them.

Taxonomies and Global Taxonomic Constraints. As mentioned in section 1.2, taxonomies have been defined

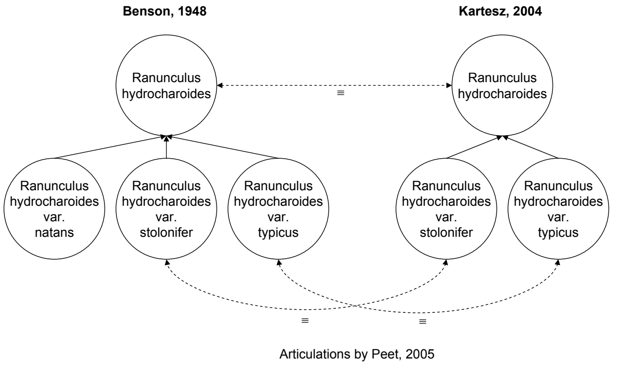
Benson, 1948
Kartesz, 2004
Ranunculus hydrocharoides
≡
Ranunculus hydrocharoides
Ranunculus hydrocharoides var. natans
Ranunculus hydrocharoides var. stolonifer
Ranunculus hydrocharoides var. typicus
Ranunculus hydrocharoides var. stolonifer
Ranunculus hydrocharoides var. typicus
≡
≡
Articulations by Peet, 2005

Figure 2: An alignment that is inconsistent under the non-emptiness, sibling-disjointness and coverage constraints. Solid lines represent isa {≡,⊊} relations. Dotted lines represent equivalence (≡) articulations provided by an expert.

as a partial ordering of concepts where the ordering relation denotes some sort of "inclusion" relation. To model this in $\mathcal{L}_{\text{MFOL}}$, we associate with every edge $N \overset{\texttt{isa}}{\dashrightarrow} M$ in a given taxonomy, a first-order formula (or logic constraint) $\forall x\colon N(x) \rightarrow M(x)$, stating that if x is in N, then x is also in M. With this logic formalization, the containment relation defined by $N \overset{\texttt{isa}}{\dashrightarrow} M$ is true if given interpretation $\mathcal{I}$, either $N^{\mathcal{I}} \subsetneq M^{\mathcal{I}}$ (proper containment) or $N^{\mathcal{I}} = M^{\mathcal{I}}$ (set equality).

However, this definition is quite weak. Taxonomies in general will have additional constraints. Some constraints may apply to a small set of concepts, while others will apply globally, across the taxonomy. We call these latter constraints *global taxonomic constraints* (GTCs). A given taxonomy may adhere to some GTCs but not others. Three common[1] GTCs are:

- **Non-Emptiness** (N): All concepts have at least one instance.
- **Sibling Disjointness** (D): Sibling concepts share no instances.
- **Coverage** (C): Parent concepts are covered by (*i.e.*, included in) the unions of their child concepts.

There are good reasons to treat GTCs as "first-class citizens." First, a person creating a taxonomy from scratch may want to enforce some of the GTCs, and not others. A taxonomy building tool should therefore provide a means to enforce the constraints while a taxonomy is being built. Second, while some representations of discovered taxonomies (*e.g.*, found online) will explicitly represent the relevant GTCs, others will simply *assume* that the GTCs hold. For example, taxonomies represented in the Taxonomic Concept Schema [1] often do not explicitly represent disjointness between sibling taxa. Instead, the D GTC is generally assumed. Even if a GTC is assumed, however, it may not be enforced in any given taxonomy or alignment. For example, the alignment shown in Figure 2 was created by an expert taxonomist, but is inconsistent under the NDC GTCs. Anyone wishing to use a downloaded taxonomy or alignment should check whether or not a set of desired GTCs holds.

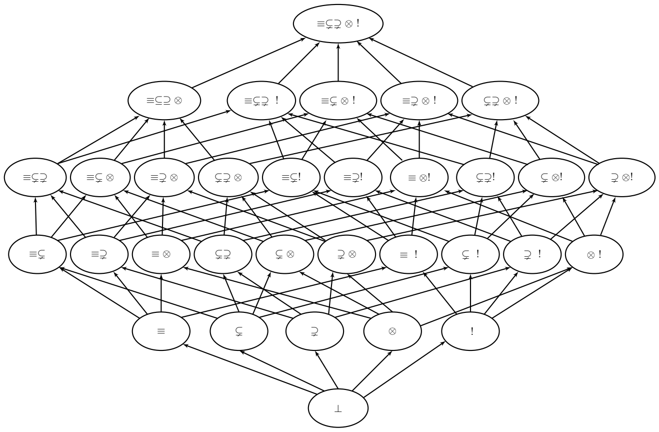

Figure 3: The $\mathbb{R}_{32}$ lattice.

Articulations.

CleanTax uses the RCC-5 [25] topological algebra as the basis for representing articulations. This algebra describes relationships between sets, and supports the expression of incomplete knowledge when stating articulations. Furthermore, the RCC-5 algebra has been represented using first-order logic [25], propositional logic [5] and description logic [32], providing a substrate to explore and compare these different logics in this context.

The RCC-5 algebra uses the same five *basic relations* ($\mathbb{B}_5$) as MoReTaX. Given any two non-empty sets N, M exactly one of the $\mathbb{B}_5$ relations holds (*cf.* Figure 1): (i) congruence ($N \equiv M$), (ii) proper inclusion ($N \subsetneq M$), (iii) proper inverse inclusion ($N \supsetneq M$), (iv) partial overlap ($N \oplus M$), or (v) exclusion (disjointness) ($N \,!\, M$).

In general, the instances of N and M are not given, so disjunctions of $\mathbb{B}_5$ are used to describe any (partial) knowledge about the relation between N and M. The powerset $\mathbb{R}_{32} = 2^{\mathbb{B}_5}$ contains all 32 disjunctions obtainable from $\mathbb{B}_5$ relations. For example, an isa-edge $N \overset{\texttt{isa}}{\dashrightarrow} M$ captures the constraint $N \subseteq M$, *i.e.*, either N is properly contained in, or equal to M, which in turn corresponds to a disjunction $\{\equiv, \subsetneq\} \in \mathbb{R}_{32}$. The constraints in $\mathbb{R}_{32}$ form a lattice (Figure 3) with bottom element $\bot = \emptyset$, singleton relations (corresponding to $\mathbb{B}_5$ relations) in layer-1, combinations of two disjuncts in layer-2, three disjuncts in layer-3, *etc.* up to layer-5 with the (vacuously true) top element $\top = \{\equiv, \subsetneq, \supsetneq, \oplus, !\}$.

The translation of taxonomies, articulations and the N, D, and C GTCs into $\mathcal{L}_{\text{MFOL}}$ has been described in [31]. As an example, consider Figure 4, which shows two taxonomies of species in the genus *Ranunculus*, and a set of articulations provided by an expert biologist, Peet [23]. Translating this set of taxonomies, articulations, and the NDC GTCs into $\mathcal{L}_{\text{MFOL}}$ results in the formulas shown in Table 1. Once a set of taxonomies, articulations and GTCs is translated into $\mathcal{L}_{\text{MFOL}}$ an automated model finder (such as Mace4 [19]) may be applied to determine whether the alignment is satisfiable. If it is not, we call the alignment *inconsistent.*

3.1 Basic Methodology

The CleanTax methodology begins with a formalization of the given taxonomies and expert articulations in $\mathcal{L}_{\text{tax}}$ a language with an accompanying translation into monadic

[1] This is not an exclusive list.

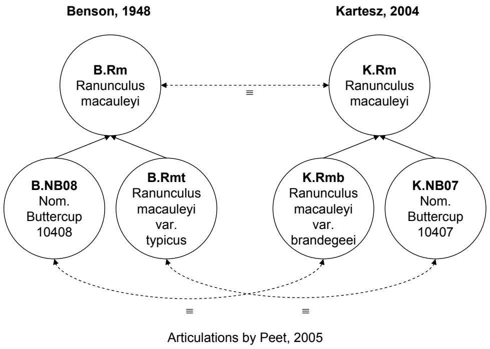

Figure 4: An expert says these two taxonomies are equivalent.

Authority	Rule
Benson, 1948	$\forall x$: $B.NB08(x) \rightarrow B.Rm(x)$ $\forall x$: $B.Rmt(x) \rightarrow B.Rm(x)$
Kartesz, 2004	$\forall x$: $K.Rmb(x) \rightarrow K.Rm(x)$ $\forall x$: $K.NB07(x) \rightarrow K.Rm(x)$
Peet, 2005	$\forall x$: $B.Rm(x) \leftrightarrow K.Rm(x)$ $\forall x$: $B.NB08(x) \leftrightarrow K.Rmb(x)$ $\forall x$: $B.Rmt(x) \leftrightarrow K.NB07(x)$
Sibling Disjointness	$\forall x$: $B.NB08(x) \rightarrow \neg B.Rmt(x)$ $\forall x$: $K.Rmb(x) \rightarrow \neg K.NB07(x)$
Coverage	$\forall x$: $B.Rm(x) \leftrightarrow B.NB08(x) \vee B.Rmt(x)$ $\forall x$: $K.Rm(x) \leftrightarrow K.Rmb(x) \vee K.NB07(x)$
Non-Emptiness	$\exists x$: $B.Rm(x)$ $\exists x$: $B.NB08(x)$ $\exists x$: $B.Rmt(x)$ $\exists x$: $K.Rm(x)$ $\exists x$: $K.Rmb(x)$ $\exists x$: $K.NB07(x)$

Table 1: $\mathcal{L}_{\text{MFOL}}$ rules for Figure 4 plus non-emptiness, sibling-disjointness and coverage constraints

first-order logic. This allows us to apply first-order logic reasoning techniques to automatically detect possible inconsistencies, and to infer missing articulations via a deductive closure.

Given a set of taxonomies, a set of articulations (constraints) between pairs of concepts, and a set of GTCs, the CleanTax **basic algorithm $\mathcal{A}^0$** proceeds as follows:

1. For each taxonomy, apply each combination of GTCs, $(\{\{GTC\} \mid \{GTC\} \in 2^{GTC}\})$ and translate the result into a set of $\mathcal{L}_{\text{MFOL}}$ formulas Φ_T. Eliminate GTC combinations that result in an inconsistent Φ_T.

2. Apply each GTC combination to the articulations, generating a set of $\mathcal{L}_{\text{MFOL}}$ formulas Φ_A. Eliminate GTC combinations that result in an inconsistent Φ_A.

3. For each GTC combination that is consistent with all the taxonomies and the articulations, check whether it is consistent with the combined taxonomies and articulations.

4. Compute the deductive closure for the combined taxonomies and articulations under each consistent GTC combination. For each pair of concepts N, M from different taxonomies and each relation $\circ \in \mathbb{R}_{32}$, determine whether $N \circ M$ holds.

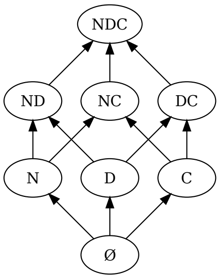

Figure 5: The GTC lattice.

Step 4 computes a deductive closure from the consistent combinations of input taxonomies, GTC combinations, and articulations, yielding all logically implied relations between concepts from different taxonomies. This step employs the FOL theorem prover Prover9 [18]; consistency checks are performed with the companion tool Mace4 [19].

3.2 Optimizations

While reasoning in $\mathcal{L}_{\text{tax}}$ is decidable [3], it is still computationally hard (NEXP-complete). The following optimizations reduce the number of logical tests necessary to calculate the deductive closure of a set of taxonomies and articulations.

GTC Lattice Optimization The powerset of the three GTCs, N, D, and C, gives rise to a lattice of eight GTC combinations (Figure 5) which can be exploited to avoid unnecessary work. Adding new formulas to a set already shown to be inconsistent will never result in a consistent set of formulas ($\mathcal{L}_{\text{MFOL}}$ is monotonic). Therefore, once a given GTC is shown to create an inconsistency for a taxonomy, no parent nodes of that GTC in the GTC lattice need to be investigated.

$\mathbb{R}_{32}$ **Lattice Optimizations.** For any pair of concepts N, M many of the relations in $\mathbb{R}_{32}$ may hold, *i.e.*, evaluate to true. For example, if $N \equiv M$, then for any disjunction $\circ \in \mathbb{R}_{32}$ containing $\equiv$, $N \circ M$ also holds. However, there is a single distinguished true relation that implies *all* other true relations in $\mathbb{R}_{32}$, *i.e.*, the *meet* of the sublattice of true relations. We call this the *maximally informative relation* (`mir`). Thus, the goal of the optimizations is to find `mir` as quickly as possible. Then all relations "above" `mir` are known to evaluate to true, and all remaining ones to false, without further checking.

Algorithm $\mathcal{A}^{\uparrow}_{\texttt{mir}}$ proceeds bottom up, starting at layer-1 in $\mathbb{R}_{32}$ and stopping as soon as a true relation is found. In the best case, one of the layer-1 relations evaluates to true (there are at most 5 proofs for layer-1); in the worst case only the single layer-5 relation evaluates to true, *i.e.*, we know nothing about the relationship between N and M. The latter will result in 30 tests in the lattice: we skip tests with $\bot$ (always false) and $\top$ (always true) relations. [2]

Algorithm $\mathcal{A}^{\downarrow}_{\texttt{mir}}$ first tests all five relations in layer-4 of the

[2] It may be the case that, due to incomplete information, none of the five layer-1 nodes evaluates to true. For example, although the disjunction $p \ \vee \neg p$ is certainly true, we may not know whether p is true or $\neg p$ is true. Similarly, it is often the case that none of the layer-1 nodes evaluates to true for a given articulation, while a node higher in the $\mathbb{R}_{32}$ lattice does. See Table 2 to get a sense for how frequently this occurs.

Relation	# Discovered	Relation	# Discovered
$\{\subsetneq\}$	137	$\{\supsetneq,\oplus\}$	5
$\{\supsetneq\}$	90	$\{\supsetneq,\oplus,!\}$	10
$\{\equiv,\subsetneq\}$	28	$\{\equiv,\subsetneq,\supsetneq,\oplus\}$	3
$\{\equiv,\supsetneq\}$	138	$\{\subsetneq,\supsetneq,\oplus,!\}$	-1

Table 2: New `mir` relations found under the No GTC condition.

$\mathbb{R}_{32}$ lattice (each layer-4 relation is the complement of a $\mathbb{B}_5$ relation in layer-1): *e.g.*, $\{\equiv, \subsetneq, \supsetneq, !\}$ in layer-4 is equivalent to *not*-$\oplus$; if true we know that N does *not* partially overlap M. Next `mir` is determined from those results.

Let $\mathbb{T}_4$ be the layer-4 relations that are true for some pair N, M. One can show that $\texttt{mir} = \bigcap_{R \in \mathbb{T}_4} R$. For example, if the layer-4 relations $\{\equiv, \subsetneq, \supsetneq, !\}$ and $\{\equiv, \subsetneq, \oplus, !\}$ hold for a given N, M, we can derive a `mir` of $\{\equiv, \subsetneq, !\}$. Thus by testing exactly 5 out of the 30 non-trivial $\mathbb{R}_{32}$ relations and combining those results, $\mathcal{A}^{\downarrow}_{\texttt{mir}}$ avoids all other 25 tests.

3.3 Experimental Results

The CleanTax approach was applied to a real-world data set consisting of two biological taxonomies and expert articulations between them. The goals of the experiments were to determine whether the system could discover inconsistencies in the data set, and if it could discover new articulations not given by the expert. The role that various GTCs might play in the system's ability to find inconsistencies and new articulations was also investigated. Finally, the efficacy of the optimizations was examined.

The test data set described two taxonomies of the plant genus *Ranunculus*, Benson, 1948 [6] and Kartesz, 2004 [16], and a set of articulations between these by Peet [23]. The taxonomies comprised 360 taxa (218 for Benson and 142 for Kartesz), and 218 articulations. The latter were of the following types: $\equiv$ (112), $\subsetneq$ (15), $\supsetneq$ (63), $\oplus$ (4), ! (12), $\{\subsetneq,\supsetneq,\oplus,!\}$ (12). The taxa in the chosen taxonomies covered the Linnean taxonomic ranks of *genus*, *species*, and *variety*.

Consistency. The two taxonomies and their articulations were only consistent under the non-emptiness GTC (N) and also without any GTCs. The coverage GTC (C) alone introduced inconsistencies, so any GTC combination involving C was also inconsistent. The sibling-disjointness GTC (D) introduced so many new formulas that neither Mace4 nor Prover9 could process the input with the given resources. Without GTCs there were 428 logic formulas, while adding the sibling-disjointness GTC D yielded a total of 18,104 formulas, most of the form $N(x) \rightarrow \neg\, M(x)$. This demonstrates that while reasoning in $\mathcal{L}_{\text{tax}}$ may be decidable, it is not necessarily tractable.

Discovered Relations. Table 2 shows the counts of `mir` values of all newly discovered relationships under the No GTC condition. The -1 value in the $\{\subsetneq,\supsetneq,\oplus,!\}$ relation reflects the movement of one of the $\{\subsetneq,\supsetneq,\oplus,!\}$ relations stated by Peet to a $\{\supsetneq,\oplus,!\}$ relation. In other words, the framework not only inferred new articulations, but also found more specific versions of given articulations.

The only difference found between the No GTC condition and the Non-Emptiness condition was that two additional $\{\equiv,\subsetneq,\supsetneq,\oplus\}$ relations were discovered when the Non-Emptiness GTC was applied.

	$\mathcal{A}^0$	$\mathcal{A}^{\uparrow}_{\texttt{mir}}$	$\mathcal{A}^{\downarrow}_{\texttt{mir}}$
Judgments	928,680	912,779	154,780
Time (mins)	2,810.65	2,761.26	477.59
Logical steps (millions)	2,634	2,589	442

Table 3: Impact of optimizations on deductive closure under the non-emptiness (N) GTC.

	$\varnothing$	N	D	C	ND	NC	DC	NDC
layer-1 `mir`	245	245	393	259	451	259	417	475
layer-2 `mir`	75	75	61	85	3	85	64	6
layer-3 `mir`	17	17	17	43	47	43	34	38
layer-4 `mir`	20	22	20	4	2	4	0	0
layer-5 `mir`	192	190	58	158	46	158	34	30

Table 4: New `mir` relationships for each GTC in the 75 sub-taxonomies that are consistent under the NDC-GTC.

Optimizations. Table 3 shows the impact of the two $\mathbb{R}_{32}$ lattice optimizations. The number of judgments in the table represents how many assertions of the form "concept relation concept" were tested in the process of calculating the deductive closure for the taxonomies and articulations. The number of logical steps provides a rough measure of the total work performed by the reasoners.

Clearly, calculating the deductive closure of two taxonomies and a set of articulations under even a single GTC can involve a great number of logical tests. There is only a slight improvement of the bottom-up algorithm $\mathcal{A}^{\uparrow}_{\texttt{mir}}$ over the base algorithm $\mathcal{A}^0$. However, $\mathcal{A}^{\downarrow}_{\texttt{mir}}$ reduces the number of tests by 84% and is processed almost 6 times as quickly as the unoptimized $\mathcal{A}^0$.

3.4 Modularization via Connected Subgraphs

The improvement of $\mathcal{A}^{\uparrow}_{\texttt{mir}}$ is small because the N-GTC engenders very little deductive power, so in general the relation between any two given concepts is unknown, resulting in the worst-case scenario for $\mathcal{A}^{\uparrow}_{\texttt{mir}}$: all 30 relations must be checked.

To investigate the impact of the optimizations in a scenario with less uncertainty, the taxonomies and articulations were divided into a set of 81 connected subgraphs, each representing a species in one taxonomy and all the related taxonomic concepts below the genus level in the other taxonomy. These sub-taxonomies were then tested for consistency under each GTC combination.

Of the 81 sub-taxonomies, 6 were inconsistent under some combination of GTCs. In the 75 sub-taxonomies that were consistent under all GTC combinations, between 357 and 519 new, informative, `mir` relations were discovered, depending on the GTC combination (the top node $\top = \{\equiv, \subsetneq, \supsetneq, \oplus, !\}$ is not considered informative). For example, Figure 6 shows the inference of a new inclusion relation between *Ranunculus arizonicus var. chihuahua* according to Benson 1948, and *Ranunculus arizonicus* according to Kartesz, 2004.

This inference was determined using Prover9 after applying the NDC GTC combination to the taxonomies and given articulations and generating a set of $\mathcal{L}_{\text{MFOL}}$ formulas from the result. Table 4 demonstrates that as more constraints are placed on the taxonomies (*e.g.*, from No-GTCs, to C, to NC, to NDC-GTCs) the specificity of discovered relationships increases. For example, under the No-GTC con-

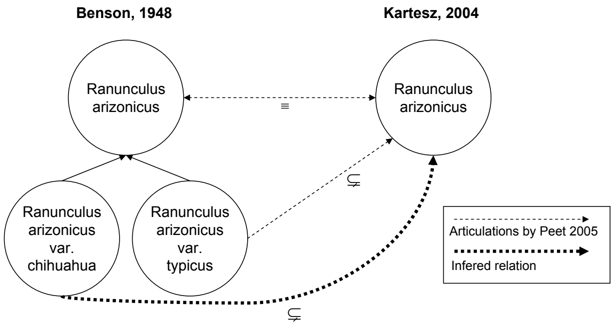
Benson, 1948
Kartesz, 2004
Ranunculus arizonicus
Ranunculus arizonicus
≡
Ranunculus arizonicus var. chihuahua
Ranunculus arizonicus var. typicus
⊊
⊊
Articulations by Peet 2005
Infered relation

Figure 6: Applying the NDC GTCs to these taxonomies and articulations facilitates the inference of a new inclusion relation between *Ranunculus arizonicus var. chihuahua* according to Benson 1948, and *Ranunculus arizonicus* according to Kartesz, 2004.

	$\mathcal{A}^0$	$\mathcal{A}^{\uparrow}_{\texttt{mir}}$	$\mathcal{A}^{\downarrow}_{\texttt{mir}}$
Judgments	17,019	2194	2745
Time (secs)	1763.18	279.52	292.83
Logical steps (thousands)	2,484	384	394

Table 5: Impact of optimizations on deductive closure under the NDC GTC for 75 sub-taxonomies.

dition, 357 new, informative, `mir` relations were found and 112 of those contained some uncertainty (were disjunctions). In comparison, under the NDC-GTC condition, 519 new, informative, relations were discovered, only 44 of which contained uncertainty. Furthermore, the number of completely uninformative relations {≡,⊊,⊋,⊕,!} dropped from 192 in the No-GTC condition to 30 in the NDC condition.

Table 5 demonstrates that the $\mathcal{A}^{\uparrow}_{\texttt{mir}}$optimization improves relative to the $\mathcal{A}^{\downarrow}_{\texttt{mir}}$optimization under the NDC GTC combination, which engenders the inference of many specific relations.

3.5 Alternative Logics

Initial tests were performed using FOL reasoners and a very expressive language for representing taxonomies and articulations ($\mathcal{L}_{\text{MFOL}}$). The trade off for this expressivity is poor performance. Although $\mathcal{L}_{\text{MFOL}}$ is decidable, its complexity is still NEXP-complete. We have begun an investigation of smaller logics, notably Description Logics, propositional logic, and subsets of the RCC calculus. Table 6 describes the trade-offs of various languages in this context.

Satisfiability problems stated in a subset of the $\mathbb{R}_{32}$, $\mathbb{R}_5^{28}$ can be solved in polynomial time [26]. $\mathbb{R}_5^{28}$, however, does not permit empty concepts, coverage, or any relationship that includes {⊊ ,⊋} unless it also includes ⊕. Translations of the RCC-5 into propositional logic ($\mathbb{R}_{32}$-PL) permit all $\mathbb{R}_{32}$ relations, and given the optimizations of propositional logic reasoners, may be quite fast. However, the Coverage GTC requires the use of disjunctions, which are outside of $\mathbb{R}_{32}$. Adding these disjunctions ($\mathbb{R}_{32}$-PL + union) may add complexity. The standard Description Logic used by the Semantic Web community - $\mathcal{SHOIN}$(D), is more expressive than propositional logic, and though it's more complex than $\mathbb{R}_{32}$+ union, it may be faster than $\mathcal{L}_{\text{tax}}$. However, representing disjunctive sentences in $\mathcal{SHOIN}$(D) is not straightforward, so it may not be possible to represent the full $\mathbb{R}_{32}$ in $\mathcal{SHOIN}$(D) directly. Finally, $\mathcal{L}_{\text{tax}}$ is the most expressive language, supporting all the GTCs, but probably the most computationally complex.

Logic	Empty Concepts	Coverage	Full R32	Complexity
$\mathbb{R}_5^{28}$	No	No	No	Polynomial
$\mathbb{R}_{32}$-PL	No	No	Yes	NP-Complete
$\mathbb{R}_{32}$-PL+union	No	Yes	Yes	NP-Complete (?)
$\mathcal{SHOIN}$(D)	Yes	Yes	No (?)	NEXP-Complete
$\mathcal{L}_{\text{tax}}$	Yes	Yes	Yes	NEXP-Complete

Table 6: Features of various languages in increasing order of complexity.

4. SUMMARY AND FUTURE WORK

Taxonomies can be considered simplified ontologies, and their relative simplicity lends them increased tractability. Initial work on reasoning about taxonomies and articulations has resulted in the following findings: (i) the $\mathcal{L}_{\text{tax}}$ language can represent ambiguous relations, and this is important for modeling real-world situations (ii) the infrastructure is able to detect inconsistencies and infer new relationships, (iii) the specificity of the newly discovered relationships increases with the number of constraints applied, (iv) the optimizations were effective in reducing the number of "proof obligations" and therefore the time necessary to complete the deductive closure, and (v) the efficacy of the optimizations depended on the average specificity of the inferred relations.

Future work will progress in four directions: theory, operations, implementation, and application.

Theory. The optimizations described above lack formal proofs. In addition, several theorems have been proposed, but not formalized or proven. Among these are:

- Given a closed set of articulations and taxonomies, the GTCs applied to generate the articulations may be derived.
- Two or more of the basic RCC-5 relations can never be entailed by any given set of consistent taxonomies and articulations.
- The efficiencies achieved by using $\mathbb{R}_5^{28}$ over $\mathbb{R}_{32}$ apply in the current context.

Operations. The CleanTax methodology does not yet support the following operations: giving guidance to articulators on how to fix inconsistent taxonomies and articulations, explaining inconsistencies and discovered relations, and merging taxonomies given articulations.

In addition, further optimizations will be necessary, both in serial applications of the framework, and for distribution of reasoning tasks across multiple nodes of a cluster.

Implementation. An object-oriented Python-based architecture is well under way. As already described, the architecture serves two communities: researchers interested in studying reasoning about taxonomies and articulations, and metadata curators who would like to create taxonomies and articulations. This architecture needs to support a variety of reporting and visualization tasks.

The nature of the reasoning task lends itself to the decomposition of the Python-architecture into a scientific workflow, which would support componentized data preprocessing steps, task parallelization, and pluggable reasoners. Moving in this direction would help create a more easily modified

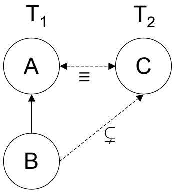

Figure 7: An alignment involving a monotypic taxon (A).

architecture.

Applications. Progress has been made in analyzing and comparing Linnaen biological taxonomies. To further demonstrate the utility of the framework, other domains will be tested. Among these may be phylogenetic data, ebXML ontologies, bizTalk taxonomies, and large web-based taxonomies such as DMOZ and Yahoo.

5. CONCLUSION

The research described here has shown the efficacy of representing taxonomies and relationships between them using at least one subset of first-order logic. Basing the relationship language on the RCC-5 topological algebra has proven useful in representing incomplete knowledge, and its application in the context of taxonomic alignment represents a new contribution. The optimizations applied to reasoning with the $\mathbb{R}_{32}$ lattice are also a new contribution, and will apply not only to $\mathcal{L}_{\text{tax}}$ but to other languages that may be translated into subsets of FOL. Future work promises to add further theoretical rigor to this approach, as well as create an application useful for those interested in studying taxonomies and articulations, and for metadata curators who need a tool to assist them in constructing and analyzing taxonomies and articulations.

6. WORKSHOP FEEDBACK

Discussion subsequent to the presentation of this work focussed on two issues: (i) how biologists have received the work, especially commentary on the consistency of articulations, and (ii) questions about the computational complexity of the approach described here.

After CleanTAX discovered inconsistencies in the Ranunculus data set, Bob Peet, the creator of the data set was contacted, and some of the inconsistencies were discussed. Some inconsistencies arose from different geographic scopes covered by the two taxonomies examined. The problem described in Figure 2 arises because there are no *Ranunculus hydrocharoides var. natans* in the geographic scope covered by Kartesz. There are several potential solutions to this problem: the articulation between the *Ranunculus hydrocharoides* taxa could be changed from ≡ to $\supsetneq$, one of the constraints (probably coverage) could be dropped, or an additional empty *Ranunculus hydrocharoides var. natans* taxon could be added to Kartesz's taxonomy. Peet argued that the articulator should be the one to decide, based on best practices and recommendations by the system. Other inconsistencies arose from problems in the translation of the data set into the representation described here, or misapplication of constraints. For example, there are cases of "monotypic" taxa, which are taxa having only one immediately subordinate taxon. For example, the genus *Ginkgo* has only one species, *Ginkgo biloba*. In such cases, the coverage constraint introduces an axiom which creates an equivalence between a monotypic taxon and its child. Consider, for example, the situation in Figure 7. In T_1, B "isa" A, which translates to $\forall x : B(x) \rightarrow A(x)$. The coverage GTC states that a parent taxon is defined as the union of its children, leading to an additional axiom: $\forall x : A(x) \rightarrow B(x)$. The combination of these results in $\forall x : A(x) \leftrightarrow B(x)$. This result, however, conflicts with Peet's $B \subsetneq C$ articulation; if $A \equiv B$ and $A \equiv C$ then $B \equiv C$. There are two solutions to this problem. First, perhaps the coverage constraint should not be applied to monotypic taxa. Second, the articulation between B and C could be changed to $\{\equiv, \subsetneq\}$.

The second question to arise in the workshop was the computation complexity of CleanTAX. The computational complexity of the system is largely determined by the complexity of the reasoner used to determine the consistency of the logical axioms and to make new inferences. The work described here used PROVER9, a full first-order reasoner. In general, reasoning with full first-order logic (answering satisfiability questions) has an NEXP-complete complexity. This complexity overshadows the entire system. Ignoring the complexity of the reasoner, the unoptimized CleanTAX algorithm described in Section 3.1 is $O(n * m)$ where n and m are the number of taxa in the taxonomies. As described in Section 4, future work includes the exploration of a polynomial-time algorithm for reasoning in CleanTAX when relationships between taxa are restricted to those in $\mathbb{R}_5^{28}$. Should we obtain the expected results, the CleanTAX algorithm will run in polynomial time in these situations, as a function of the number of taxa in the input taxonomies.

7. REFERENCES

[1] The taxonomic concept schema. `http://tdwg.napier.ac.uk/`, February 2008.

[2] The taxonomic data working group. `http://www.tdwg.org/`, February 2008.

[3] L. Bachmair, H. Ganzinger, and U. Waldmann. Set constraints are the monadic class. In *Logic in Computer Science*, pages 75–83, 1993.

[4] J. H. Beach, S. Pramanik, and J. H. Beaman. Hierarchic taxonomic databases. In R. Fortuner, editor, *Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, Computer Vision*, chapter 15, pages 241–256. Johns Hopkins University Press, Baltimore, 1993.

[5] B. Bennett. Spatial Reasoning with Propositional Logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *KR'94: Principles of Knowledge Representation and Reasoning*, pages 51–62. Morgan Kaufmann, San Francisco, California, 1994.

[6] L. D. Benson. A treatise on the north american ranunculi. *American Midland Naturalist*, 40:1–261, 1948.

[7] W. G. Berendsohn. The concept of "potential taxa" in databases. *Taxon*, 44:207–212, 1995.

[8] W. G. Berendsohn. *MoReTax – Handling Factual Information Linked to Taxonomic Concepts in Biology.* Number 39 in Schriftenreihe für Vegetationskunde. Bundesamt für Naturschutz, 2003.

[9] R. Brachman. What is-a is and isn't: An analysis of taxonomic links in semantic networks. *IEEE*

Computer, 16:30–36, 1983.

[10] P. D. Cantino and K. de Queiroz. Phylocode: a phylogenetic code of biological nomenclature. 2006.

[11] H. H. Do and E. Rahm. Coma - a system for flexible combination of schema matching approaches. In *VLDB*, pages 610–621. Morgan Kaufmann, 2002.

[12] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Heidelberg, 1999.

[13] M. Geoffroy and A. Güntsch. Assembling and navigating the potential taxon graph. [8], pages 71–82.

[14] F. Giunchiglia, M. Yatskevich, and P. Shvaiko. Semantic matching: Algorithms and implementation. *J. Data Semantics*, 9:1–38, 2007.

[15] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *Knowl. Eng. Rev.*, 18(1):1–31, 2003.

[16] J. T. Kartesz. Synthesis of north american flora. BONAP, North Carolina Botanical Garden, 2004.

[17] J. Kennedy, R. Kukla, and T. Paterson. Scientific names are ambiguous as identifiers for biological taxa: Their context and definition are required for accurate data integration. In *2nd Intl. Workshop on Data Integration in the Life Sciences (DILS)*, LNCS 3615, pages 80–95, July 2005.

[18] W. McCune. *Prover9 Manual*. Argonne National Laboratory.

[19] W. McCune. Mace4 reference manual and guide. Technical Report ANL/MCS-TM-264, Argonne National Laboratory, August 2003.

[20] D. L. McGuinness. Ontologies come of age. In D. Fensel, J. A. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web*, pages 171–194. MIT Press, 2003.

[21] R. J. Miller, M. A. Hernández, L. M. Haas, L.-L. Yan, C. T. H. Ho, R. Fagin, and L. Popa. The clio project: Managing heterogeneity. *SIGMOD Record*, 30(1):78–83, 2001.

[22] N. F. Noy and M. A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.

[23] R. K. Peet. Ranunculus data set. June 2005.

[24] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.

[25] D. A. Randell, Z. Cui, and A. Cohn. A Spatial Logic Based on Regions and Connection. In B. Nebel, C. Rich, and W. Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.

[26] J. Renz and B. Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108(1-2):69–123, 1999.

[27] L. Serafini and A. Tamilin. Drago: Distributed reasoning architecture for the semantic web. In A. Gómez-Pérez and J. Euzenat, editors, *ESWC*, volume 3532 of *Lecture Notes in Computer Science*, pages 361–376. Springer, 2005.

[28] P. Shvaido and J. Euzenat. *Ontology Matching*. Springer, Heidelberg, 2007.

[29] H. Stuckenschmidt, L. Serafini, and H. Wache. Reasoning about ontology mappings. Technical report, ITC-IRST, Trento, 2005.

[30] G. Stumme and A. Maedche. FCA-MERGE: Bottom-Up Merging of Ontologies. In *Proc. of the 17th International Joint Conference on Artificial*, pages 225–234, 2001.

[31] D. Thau and B. Ludascher. Reasoning about taxonomies in first-order logic. *Ecological Informatics*, 2(3):195–209, Oct 2007.

[32] M. Wessel. On spatial reasoning with description logics-position paper. In I. Horrocks and S. Tessaris, editors, *Proceedings of the International Workshop in Description Logics*, pages 156–163, Touluse, France, April 2002. CEUR Workshop Proceedings.