Merging Taxonomies under RCC-5 Algebraic Articulations

David Thau
Dept. of Computer Science, University of California at Davis
dthau@ucdavis.edu

Shawn Bowers
Genome Center, University of California at Davis
sbowers@ucdavis.edu

Bertram Ludaescher
Dept. of Computer Science, University of California at Davis
ludaesch@ucdavis.edu

Received 21 February 2009; Accepted 16 March 2009

Taxonomies are widely used to classify information, and multiple (possibly competing) taxonomies often exist for the same domain. Given a set of correspondences between two taxonomies, it is often necessary to "merge" the taxonomies, thereby creating a unied taxonomy (e.g., that can then be used by data integration and discovery applications). We present an algorithm for merging taxonomies that have been related using articulations given as RCC-5 constraints. Two taxa N and M can be related using (disjunctions of) the ve base relations in RCC-5: $N \equiv M$; $N \subsetneq M$; $N \supsetneq$; $N \oplus M$ (partial overlap of N and M); and $N \,!\, M$ (disjointness: $N \cap M = \emptyset$). RCC-5 is increasingly being adopted by scientists to specify mappings between large biological taxonomies. We discuss the properties of the proposed merge algorithm and evaluate our approach using real-world taxonomies.

Categories and Subject Descriptors: I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods; D.2.12 [**Software Engineering**]: Interoperability; H.4.m [**Information Systems Applications**]: Miscellaneous

General Terms: Algorithms, Management, Theory

Additional Key Words and Phrases: Integration, Merging, Taxonomy, Mapping, Aligning, Automated Deduction, Reasoning, RCC-5

Copyright(c)2009 by The Korean Institute of Information Scientists and Engineers (KIISE). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Permission to post author-prepared versions of the work on author's personal web pages or on the noncommercial servers of their employer is granted without fee provided that the KIISE citation and notice of the copyright are included. Copyrights for components of this work owned by authors other than KIISE must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires an explicit prior permission and/or a fee. Request permission to republish from: JCSE Editorial Office, KIISE. FAX +82 2 521 1352 or email office@kiise.org. The Office must receive a signed hard copy of the Copyright form.

1. INTRODUCTION

Classification hierarchies, i.e., *taxonomies*, are widely used to organize various types of information [Bailey 1994; Staff 1975; Linnaeus 1758]. For example, taxonomies have been used for centuries to classify living organisms, and more recently, to species based on their evolutionary history [Doolittle 1999], proteins [Orengo et al. 1997], diseases [Côté et al. 1993], and genes [Henikoff et al. 1997], among others. It is common for similar information to be represented by multiple differing taxonomies. The differences may be due to disagreement among experts, changes in how a field is conceptualized, or because of overlaps within different fields of study. To effectively use multiple, overlapping taxonomies (e.g., for data discovery or integration), it is crucial to be able to both represent and reason over their similarities and differences.

In this paper, we focus on *merging* multiple taxonomies based on a given *alignment* [Klein 2001; Ehrig 2007; Euzenat 2004; Jung 2006], i.e., a set of *articulations* specifying the relationships among concepts (classes, taxa) in different taxonomies. Our work is motivated by taxonomies arising in biological applications, e.g., where large species or phylogenetic taxonomies have been created and the mappings (i.e., articulations) between them constructed by one or more domain experts [Koperski et al. 2000]. Unlike other approaches focussed on general ontology alignment and merging [Dou et al. 2004; Noy and Musen 2003; Kotis and Vouros 2004; Stumme and Maedche 2001b], we assume that articulations are given as RCC-5 algebra constraints [Randell et al. 1992], which are often used for expressing set-based topological relations and are increasingly used to specify articulations among species taxonomies [Kennedy et al. 2005].

The primary contribution of this paper is an algorithm for merging taxonomies in which the result of the merge operation is a new, unified taxonomy that maintains links to the original sources. Our approach has the following main advantages.

RCC-Based Articulations. Unlike the articulation relationships supported in most alignment systems [Noy and Musen 2003; Kotis et al. 2006], articulations using RCC-based relationships mirror the articulations seen in biological taxonomic alignments [Koperski et al. 2000; Franz et al. 2007]. For example, unlike in many ontology approaches, the RCC algebra supports representation of incomplete knowledge via explicit disjunctive relationships between taxa (e.g., taxon A is either disjoint from, or included in taxon B). In addition, complexity analyses of the RCC algebra have provided results showing when polynomial time reasoning is possible using the RCC relationships [Jonsson and Drakengren 1997].

Merge Results as Taxonomies. Because the result of a merge is itself a taxonomy, it is amenable to the application of known taxonomic operations. For example, from a merged taxonomy we can determine if the merge result adheres to specific taxonomic constraints, if it is logically consistent, if it contains synonyms, if it contains uncertainty that can be reduced, or if it contains redundant articulations.

Links to Original Sources. Merged taxonomies that contain links to source taxonomies may be used by applications such as data aggregators that combine observations of

species from many data sources (occurrence counts, height and weight measurements, etc.) – where each source may use a different "field guide" (species taxonomy). For example, using a merged taxonomy, it becomes possible to: determine if two data sets contain observations of the same species even when the species are described using different taxonomies; convert data sets into equivalent ones but with a different taxonomy; and discover data sets via concepts drawn from familiar, underlying taxonomies [Dou et al. 2004].

Simplified Taxonomic Views. A single merged taxonomy can also help users understand the effect of articulations between source taxonomies. Although a large set of articulations might be consistent, it still may be diffcult to understand all implications simply by considering pairwise combinations of taxa. Providing a minimal "taxonomic" view of the product of the alignment can help a user understand the impact of an alignment, and refine it as necessary.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the CLEANTAX framework, upon which our merge algorithm is based, and discusses related work, comparing our approach to those for general ontology-based alignment and merging. Section 3 introduces features of a merge algorithm that we see as important for systems utilizing biological taxonomy alignments. Section 4 describes in more detail the CLEAN-TAX framework [Thau and Ludascher 2007; Thau 2008], and presents our approach for merging taxonomies within CLEANTAX. Section 5 presents an initial implementation and experimental results of our framework for the merging of two large biological taxonomies as well as smaller examples highlighting interesting features of our merge algorithm. Finally, Section 7 summarizes the findings presented here and describes future work related to taxonomic merge operations.

2. RELATED WORK

The merge approach described in this paper is based on the CLEANTAX framework [Thau and Ludascher 2007; Thau 2008], in which a taxonomy $T = (N, \preccurlyeq_N, \Phi)$ is denoted by a set of taxa, or names, N (i.e., a set of concepts or names[1]); a partial ordering relation on those taxa $\preccurlyeq_N$, denoting an "isa" relation; and a set of additional taxonomic constraints Φ. Each taxon is thought to be a set of instances (although the complete extent of a taxon is typically not known). Each "isa" relation between taxa translates to an implication; if A "isa" B, then all things that are in A are also in B, i.e., $A \subseteq B$, or in predicate logic,

$$\forall_x : A(x) \rightarrow B(x):$$

The additional *taxonomic constraints* Φ describe other set-based relationships between taxa, e.g., stating that two taxa are disjoint (share no instances), i.e. $\neg\exists x : A(x) \wedge B(x)$. An *articulation* between taxa in different taxonomies is also a set-based relationship. For instance, an articulation may state that two taxa are equivalent: $\forall x : A(x) \leftrightarrow B(x)$, or that one taxon is properly contained within another: $\forall x : A(x) \rightarrow B(x)) \wedge \exists y : B(y)$

[1]Our taxon names are typically quantified with an authority, so we use the terms (qualified, constrained) name and concept, synonymously.

$\wedge \neg A(y)$. Equivalence, disjointness and proper inclusion are three of the five RCC-5 relations, see Section 4.3 for details.

A *merge operation* in our approach takes as input a pair of taxonomies T_1, T_2 and a set of articulations between them A_{12}, and outputs a new taxonomy T_3. In CLEANTAX, taxonomies, articulations, and additional taxonomic constraints are represented in first-order logic and reasoning over these is performed via a first-order reasoner (e.g., [W.W. McCune 2008; Riazanov and Voronkov 2002]). This reasoning may discover inconsistencies in the articulations, or may discover additional articulations. Given a consistent alignment, a merge is then performed by combining equivalent taxa, and creating a new taxonomy based on the given and inferred articulations. The primary contributions are a characterization and algorithm for the merge in this setting, and the manner in which taxa are combined.

Taxonomies may be seen as simplified ontologies. There has been a considerable amount of work on merging ontologies. Much of this work has focused on using instances [Stumme and Maedche 2001a], or lexical information in the names and definitions of classes [Kim et al. 2005] to automatically generate articulations between concepts in separate ontologies. The ontologies are then merged together based on these articulations. The use of instances and lexical information in these systems differs from the work described here, which focuses specifically on the structure of the taxonomies being merged (i.e., the concepts, or taxa, and their relationships). Of the many tools and approaches for ontology merging, the OntoMerge [Dou et al. 2004], Chimæra [McGuinness et al. 2000b; 2000a], and iPrompt [Noy and Musen 2003] systems are most similar to CLEANTAX.

In OntoMerge, the merge of two ontologies is the union of the axioms defining the ontologies and the articulations between them. The approach employed by OntoMerge is meant to assist in the translation of data represented using terms from one ontology into data that can be represented using another ontology. In addition to data translation, OntoMerge is meant to support query answering between ontologies, so that queries stated using terms of one ontology may be rewritten into queries over other ontologies. In both of these scenarios, the merge must maintain connections to the source ontologies. Unlike CLEANTAX, which uses relations drawn from the RCC-5 algebra, articulations in OntoMerge are represented using an enriched full first-order logic (WebPDDL). Whereas the current implementation of CLEANTAX uses monadic first-order logic, which is decidable, OntoMerge uses a first-order reasoner called OntoEngine that performs forward and backward chaining to provide data transformations between ontologies. The result of merging ontologies in OntoMerge is represented as a set of first-order logic formulas, whereas in our approach we always construct a new "unified" taxonomy T having the structure defined above. This taxonomy can further be simplified in our approach, resulting in taxonomic merges that are often more intuitive and easier to understand for end users.

The Chimæra and iPrompt systems differ from OntoMerge in that their goal is primarily to create a new ontology from the source ontologies. Chimæra and iPrompt's merges often involve fusing identical terms in the source ontologies into a new term, and determining the subsumption and disjointness relations between the classes in the separate ontologies. Unlike both CLEANTAX and OntoMerge, these systems are

interactive, giving users hints about how concepts in the separate ontologies may relate. Whereas CLEANTAX restricts articulations to relations covered by the RCC-5 algebra, Chimæra and Prompt use framebased and description-logic based representation languages. Finally, unlike OntoMerge and CLEANTAX, determining the relationships among source concepts from a merged ontology is not supported by Chimæra and iPrompt (although iPrompt does maintain a separate log describing the process used in creating the merged ontology). Maintaining these source relations is critical for applying merged taxonomies, e.g., for data discovery and integration.

3. DESIDERATA

Below we describe a number of desirable features that we believe systems for creating, using, and managing taxonomy merges should have, and briefly describe how they are supported by our approach. These desiderata come primarily from the settings in which we wish to apply CLEANTAX, as well as those from more general settings such as ontology merging, as described in the previous section.

A goal of CLEANTAX is to effectively represent large biological taxonomies (such as classifying organisms via species taxonomies or their evolutionary history via phylogenetic trees), and to provide effcient reasoning services over them. In the case of species taxonomies, one or more domain experts often specifies articulations among taxonomies by hand, resulting in potentially tens of thousands of articulations between any given pair of taxonomies. This situation is compounded by the large number of taxonomies that exist, often having overlapping and competing taxon definitions. Thus, it is crucial for articulation providers to have tools that allow them to easily express articulations, and to understand their ramifications. Further, systems that manage taxonomies and articulations, or that use merged taxonomies to discover, translate, or integrate data also introduce a number of requirements.

In the following, we assume two taxonomies T_1 and T_2, and a set A_{12} of articulations between them. As described in Section 2, taxonomies and articulations in CLEANTAX are formalized as sets of first-order formulas. For the taxonomies and articulations defined above, we denote the union of their respective first-order formulas as:

$$\Phi_{12} = \Phi_{T_1} \cup \Phi_{T_2} \cup \Phi_{A_{12}} .$$

We denote the taxonomy T_3 resulting from the the merge of T_1 and T_2 as:

$$T_3 = T_1 \oplus_{A_{12}} T_2 .$$

3.1 Desiderata for Merge Results

The following desiderata focus on desirable features of the output (merge result) of a merge operation.

(D1) Conservative. The result of a merge should preserve all consequences of the union of the source taxonomies and articulations. Formally, if $\Phi_{12} \models \varphi$, then $T_3 \models \varphi$: When this is true, we can say the merge result is *conservative*: what was true before is still true–consequences are preserved. For example, the merge of the alignment in Figure 1(a) shown in Figure 1(b) violates this desiderata because the disjointness between taxa 2 and 3 is not maintained. One ramification of this desiderata is that

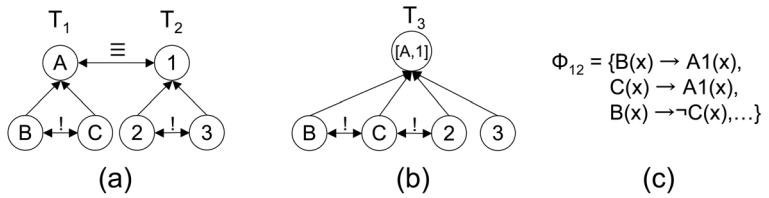

Figure 1. Given the Alignment in (a), the Merge in (b) Violates All the Described Desiderata, Except for D5 (Closure). The Merge in (c) Shows a Violation of D5.

it places restrictions on merge operations that attempt to simplify the representation of the merge result (i.e., it should still be possible to obtain all consequences of the alignment via the simplified version of the result).

(D2) Sound. The result of a merge should not introduce consequences that do not follow from the alignment. We consider two different notions of soundness: soundness and soundness under renaming. In *soundness*, all inferences that follow from the merge result should also be true of the alignment: if $T_3 \models \varphi$ then $\Phi_{12} \models \varphi$. Soundness is violated if the merge result includes new taxa that did not appear in either of the source taxonomies; e.g., if the merge result includes new taxa representing the fusion of equivalent source taxa. On the other hand, *soundness under renaming* is not violated by taxa that have been introduced during the merge if these taxa are equivalent to taxa in the original taxonomies. For example, if the relation $N \supsetneq M'$ (i.e., N is a proper superset of M') is in the merge, where N is a taxon in one taxonomy and M' is a taxon created during the merge, strict soundness will always be violated (because M' is not mentioned in either T_1, T_2, or A_{12}). However, if $M \equiv M'$ where M is in one of the taxonomies, and $N \supsetneq M$ is a relation in the original alignment, then soundness under renaming is not violated. Figure 1(b) violates both soundness and soundness under renaming because it introduces disjointness between taxa C and 2 and this disjointness does not follow from the alignment in Figure 1(a).

(D3) Source Maintaining. Many of the use cases for a taxonomic merge operator require a connection between the merged taxonomy and the source taxonomies. This type of connection is required, e.g., to translate data sets from one taxonomy to another. It is also required to query one taxonomy using terms from a second. In both of these cases, without the connection between the merged taxonomy and the sources, there would be no way to determine how the terms in the source taxonomies relate to those in the merged taxonomy. Approaches such as OntoMerge [Dou et al. 2004] contain these types of connections because the merged ontologies are precisely the formulas derived from the source ontologies and articulations. Alternatively, in approaches such as iPrompt [Noy and Musen 2003], these connections are maintained in a more indirect way, e.g., by recording the decisions made during the creation of the merge result, or in a separate mapping file. However, the connections between source taxonomies and the merge result that are maintained using iPrompt's provenance-based mechanism are diffcult to exploit in data translation tasks.

To help leverage the applicability of a source-maintaining merge result, we introduce

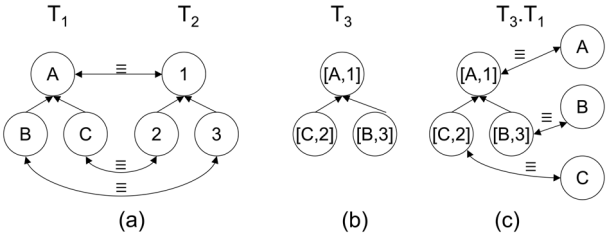

Figure 2. Projecting Taxonomy 1 from the Merge.

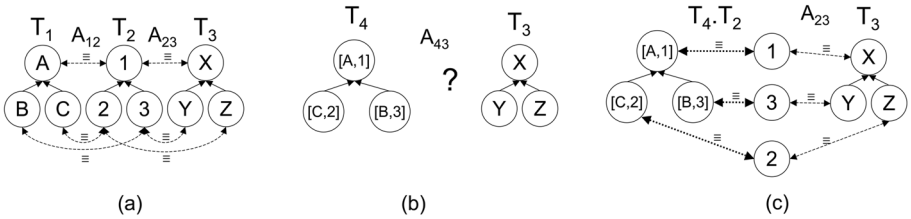

Figure 3. Using the Projection.

the "source projection" of a merged taxonomy. Given a merged taxonomy T_3 derived from taxonomies T_1 and T_2 and articulations A_{12}, the source projection (or simply projection) of the merge result provides linkages to the source taxonomies. For example, Figure 2(b) shows a merge of the alignment in Figure 2(a). The projection of T_1 from T_3 in Figure 2(c), denoted $T_3.T_1$, shows how the taxa in T_3 relate to the taxa in T_1. Note that projection does not recreate the entire source taxonomy. It simply provides linkages from the merge to its sources. Figure 3 shows how the projection might be used in a merge. Figure 3(a) shows three taxonomies and two sets of articulations. After merging T_1 and T_2, the resulting taxonomy might look like T_4 in Figure 3(b). When the merge of T_4 is attempted, the articulations in A_{23} cannot apply because of the renamed nodes in T_4, and there is no known set of articulations between T_4 and T_3. To resolve this mismatch between the taxa in T_4 and those referenced in A_{23}, T_2 is projected from T_4 in Figure 3(c), and this projection provides connection points for the A_{23} articulations. More concisely,

$$T_4 = ((T_1 \oplus_{A12} T_2).T_2) \oplus_{A_{23}} T_3$$

The merge in Figure 1(b) violates the source maintaining desiderata because there is no connection between taxon [A,1] in the merge and either taxon A or 1. In other words, $T_3.T_1$ cannot be calculated.

3.2 Desiderata for Merge Operations

The following desiderata focus on desirable properties of the merge operation itself.

(D4) Closed. The result of a merge operator should be output as a taxonomy. If the output of the merge operation is itself a taxonomy, all of the operations that apply to taxonomies may also be automatically applied to the merge result. These operations include checking the merge result for consistency, displaying the result visually,

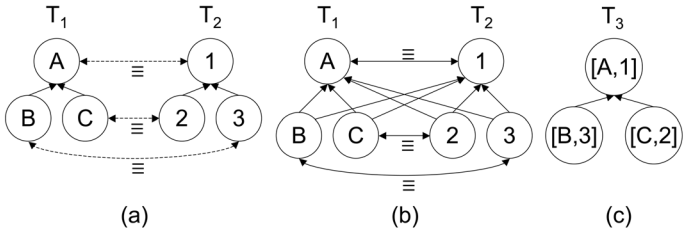

Figure 4. Merging with and Without Fusing Equivalent Taxa.

determining the minimal set of axioms to describe the merge result, and potentially merging the result with additional taxonomies. The set of logic axioms in Figure 1(c), though it may represent a merge result that satisfies all other desiderata is not a taxonomy according to our definition of a taxonomy; it has neither a specified set of taxon names N, nor a specified partial order $\preccurlyeq_N$.

(D5) Associative and Commutative. Given a sequence of (e.g., binary) merge operations, the order in which the operations are executed should not matter: $(T_1 \oplus_{A_{12}} T_2) \oplus_{A_{23}} T_3 = T_1 \oplus_{A_{12}} (T_2 \oplus_{A_{23}} T_3)$. Besides being more intuitive for users, this desiderata is also important for optimization within systems for managing taxonomies. For example, if T_2 and T_3 have been merged in the past, and the result is easily retrievable, it would be beneficial to be able to use that cached result when determining $(T_1 \oplus_{A_{12}} T_2) \oplus_{A_{23}} T_3$. The merge result in Figure 1(b) loses associativity in a merge like $(T_1 \oplus_{A_{12}} T_2) \oplus_{A_{23}} T_3$ because taxon 1 in T_2 no longer exists in the merge result; it is replaced by taxon [A,1]. This replacement of taxon names means the articulations in A_{23} involving taxon 1 from T_2 will not apply to the merged taxonomy resulting from $T_1 \oplus_{A_{12}} T_2$, and so will not be reflected in the subsequent merge with T_3. Similarly, given two taxonomies, the order in which they are provided in a merge operation should not matter, i.e., commutativity should also hold: $T_1 \oplus_{A_{12}} T_2 = T_2 \oplus_{A_{21}} T_1$.

Finally, it is also desirable for a taxonomy merged with itself to result in the original taxonomy, i.e., idempotence should also hold: $T_1 \oplus_{A_{11}} T_1 = T_1$.

(D6) Minimal. A taxonomy free of redundant information is often easier to use and understand. For example, in the alignment in Figure 4(a) could be merged as in Figure 4(b), however this merge contains a great deal of redundant information. Combining equivalent nodes, as in Figure 4(c) eliminates the redundant information and creates a merge that is easier to understand.

(D7) Scalable. As described above, merge operations should be able to scale-up to large taxonomies, containing many articulations, while preferrably providing reasonable responsetime, e.g., for articulation providers so they can quickly see merge results, for systems managing taxonomies, and for systems performing taxonomy-based data discovery, translation, and integration services.

In the following section, we describe the CLEANTAX framework in more detail, and present our CLEANTAX merge algorithm which satisfies each of the above desiderata.

4. TAXONOMY MERGING IN CLEANTAX

This section begins with a description of the representations used to describe taxonomies and articulations in the CLEANTAX framework [Thau and Ludascher 2007; Thau 2008]. It then describes the taxonomic merge operation used within CLEANTAX.

4.1 Taxa

A *taxon* (plural, *taxa*) represents a name, concept, or class. Each taxon in CLEANTAX is represented as a tuple (S, C), where S represents a unique identifier for the source taxonomy in which the taxon appears, and C represents the name of the taxon. We often refer to taxon names as unary objects. Following the example of XML elements and their namespaces, these unary taxon names may be constructed by prepending the taxon name with the unique name of the source taxonomy.

4.2 Taxonomies

Taxonomies have traditionally been defined as a partial ordering of taxa where the ordering relation denotes "inclusion" [Brachman 1983]. We start with that definition here, and then show how it needs some embellishment.

ISA-Hierarchies. Let $a, b, \ldots \in N$ be a set of taxa and $\preccurlyeq_N$ a *partial order* on N (*i.e.*, $\preccurlyeq$ is reflexive, transitive, and antisymmetric). We interpret $b \preccurlyeq a$ as $\forall x.b(x) \rightarrow a(x)$ or equivalently as $b \subseteq a$ and call $H = (N, \preccurlyeq_N)$ an *isa-hierarchy*. Formally, we can view a hierarchy $H = (N, \preccurlyeq_N)$ as a set of first-order (FO) logic formulas:

$$\Phi_H = \{\forall x.b(x) \rightarrow a(x) \mid a, b \in N \text{ and } b \preccurlyeq_N a\}.$$

Note that the *signature* of H consists only of unary predicate symbols (the taxon names in N), *i.e.*, $\sigma_H = N$. As shown, H is formalized in monadic first-order logic (MFO).

Note that this definition allows *multiple inheritance*. For example, consider $N = \{a, b, c, d\}$ and $\preccurlyeq_N = \{ba, ca, db, dc\}$.[2] This is a well-formed hierarchy in the above sense, where, e.g., instances of d are instances of both b and c.

Covering Relation. Taxonomies are often specified using the transitive reduction of the partial order $\preccurlyeq_N$, rather than the partial order itself. For example, rather than giving the complete partial order $\preccurlyeq_N = \{aa, bb, cc, dd, ba, ca, db, dc, da\}$, in CLEANTAX the transitive reduction of the relation is given $\{ba, ca, db, dc\}$.

Strictly speaking, this latter set is the covering relation

$$\prec_N = \{ba, ca, db, dc\}$$

of $\preccurlyeq_N$. We write $x \prec y$ and say that x is *covered* by y, if $x \prec y$ and there is no other $z \in N$ with $x \prec z \prec y$. Since N is finite, for $x \prec y$ there is a finite covering path $x = x_0 \prec x_1 \prec \ldots \prec x_n = y$. Thus the partial order $\preccurlyeq_N$ determines, and is determined by the covering relation $\prec_N$.

[2]Strictly speaking this is not $\preccurlyeq_N$ but its covering; see below.

Taxonomies. As described above, a *taxonomy* $T = (N, \preccurlyeq_N, \Phi)$ consists of a set of *names* N, a partial order (isa-hierarchy) $\preccurlyeq_N$, and a set of constraints Φ over N. The latter contains for each c $\prec$ p in $\prec_N$ a formula $\forall x.c(x) \rightarrow p(x)$. Note that axiomatizing $\prec$ instead of $\preccurlyeq_N$ in this way is sufficient, since logical implication $P \rightarrow Q$ is reflexive, transitive and antisymetric. Φ may contain other constraints of T as well. Typical constraints that might be in Φ include:

- *non-emptiness*: $c \neq \varnothing$ (for some or all $c \in N$)
- *sibling-disjointness*: if $c_1 \prec p$ and $c_2 \prec p$ then $c_1 \cap c_2 = \varnothing$
- *parent coverage*: $p \subseteq c_1 \cup ... \cup c_n$ (where all $c_i \prec p$)

When any of these constraints is applied to every applicable taxon in a given taxonomy, we call the constraint a *globally applied taxonomic constraint* (GTC). These constraints are often implicitly assumed in the context within which a taxonomy is presented, rather than being explicitly stated in the definition of the taxonomy. One of the primary benefits of the CleanTax system is the ability to explore the effects these constraints may have on reasoning and merging across multiple taxonomies.

4.3 Articulations

CleanTax uses the RCC-5 [Randell et al. 1992] topological algebra as the basis for representing articulations. This algebra describes relationships between sets, and supports the expression of incomplete knowledge when stating articulations.

The RCC-5 algebra uses the same five *basic relations* ($\mathbb{B}_5$) as several biological taxonomic alignments and taxonomic reasoning systems [Berendsohn 2003; Koperski et al. 2000; Franz et al. 2007]. Given any two non-empty sets N and M, exactly one of the $\mathbb{B}_5$ relations holds (*cf.* Figure 5) between them: (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" relation N isa M captures the constraint $N \subsetneq M$, 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 with bottom element $\perp = \emptyset$, 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 $\top = \{\equiv, \subsetneq, \supsetneq, \oplus, !\}$.

For any pair of taxa, N, M, many of the relations in R32 may hold. For example, if $N \equiv M$ is true, then so is $N\{\equiv, \subsetneq\}M$. However, there is a single distinguished relation in the $\mathbb{R}_{32}$ lattice that implies *all* the relations that hold between any two taxa; the

(i) *congruence* (ii) *proper inclusion* (iii) *proper inverse inclusion* (iv) *partial overlap* (v) *exclusion*

Figure 5. $\mathbb{B}_5$ – the five basic relations $N \equiv M$, $N \subsetneq M$, $N \supsetneq M$, $N \oplus M$, $N ! M$ between two sets N, M.

meet of the $\mathbb{R}_{32}$ sublattice of true relations for those two taxa. We call this the *maximally informative relation* mir. In general, we will only discuss the mir relation between two taxa.

Articulations are converted into logic formulas (Φ_A) in a straightforward way (e.g., see the rules given in [Thau and Ludascher 2007]). Most of the relations in $\mathbb{R}_{32}$ are bidirectional. However $\subsetneq$ and $\supsetneq$ are not. We assume here that the directional intent of $\subsetneq$ and $\supsetneq$ are maintained when the order of subscripts of A are reversed. In other words, if A_{12} contains the articulation $N_1 \subsetneq N_2$, then A_{21} contains the articulation $N_2 \supsetneq N_1$. Similarly, if A_{12} contains the articulation $N_1\{\equiv, \subsetneq, \oplus\}N_2$ then A_{21} contains the articulation $N_2\{\equiv, \supsetneq, \oplus\}N_1$. In this way, a set of articulations can be inversed, e.g., allowing commutative merge operations.

4.4 Taxonomies versus Ontologies

Taxonomies as defined here differ from standard description-logic ontologies. Specifically, the taxa in our taxonomies have no description logic style concept definitions like those often found in ontologies. Furthermore, relations between taxa in our taxonomies are restricted to set-theoretic relations, whereas roles between concepts in ontologies are considerably more flexible. The benefit of these restrictions is the promise of greater computational tractability. While reasoning in ontologies that conform to traditional description logics, such as those underlying OWL-DL, have an NEXP-Time complexity when answering satisfiability questions, [Renz and Nebel 1999] has shown that reasoning with all relations in $\mathbb{R}_{32}$ is an NP-complete problem, and reasoning with several subsets of the $\mathbb{R}_{32}$ relations can be performed in polynomial time. Additionally, these set-theoretic relations are convenient for specifying articulations among taxonomies; and are more expressive than the *isa, equivalence*, and *disjoint* constraints used commonly in ontology-based merging approaches.

4.5 Merging Taxonomies

The merge algorithm begins by using a reasoner to calculate the deductive closure of the union of the logic axioms describing the source taxonomies and the articulations.

$$\Phi_{12} = \Phi_{T_1} \cup \Phi_{T_2} \cup \Phi_{\Lambda 12}$$

This type of merge is much like that described in the OntoMerge system [Dou et al. 2004], whose merge result is represented by the set of logic statements rather than as a new taxonomy (i.e., violating the closure requirement of Section 3).

In CLEANTAX, we construct a taxonomic merge by coercing Φ_{12} into the signature for a taxonomy $T = (N, \preccurlyeq_N, \Phi)$. This step consists of determining the taxa involved in the merged taxonomy, deriving the transitive reduction of the partial order describing the relationships between those nodes, and deriving the additional taxonomic constraints.

We determine N, $\preccurlyeq_N$, and Φ initially as follows. N is simply the set of taxa that appear in the initial taxonomies. The transitive reduction $\preccurlyeq$ is determined by constructing a graph of the taxa in N where each taxon is a node, and there is a directed edge between any two taxa N_1 and N_2 when the $\mathbb{R}_{32}$ relation $\subsetneq$ or $\{\equiv, \subsetneq\}$ can be deduced from the deductive closure. Once this graph has been constructed, the

transitive reduction may be determined using a standard transitive reduction algorithm [Aho et al. 1972]. Finally, Φ is simply the union $\Phi_{T_1} \cup \Phi_{T_2} \cup \Phi_{\Lambda 12}$.

Once this initial taxonomy is formed, the final merge is created by merging taxa found to be equivalent, due to provided or inferred articulations.

We define an equivalence relation on N such that:

$$a \sim b \text{ if } \Phi \models \forall x.a(x) \leftrightarrow b(x),$$

where the equivalence class of $a \in N$ is $[a] = \{x \in N \mid x \sim a\}$. We say that taxonomy T has *synonyms* if for some $a, b \in N$ with $a \neq b$ we have that $a \sim b$; otherwise T is called *synonym-free*. Using this definition we can construct a unique, synonym-free version of the initial merge result. We call this simplified version a *quotient taxonomy* $T_{/\sim}$ such that:

$$N_{/\sim} = \{[a] \mid a \in N\},$$
$$\preccurlyeq_{/\sim} = \{([a]; [b]) \mid [a] \preccurlyeq [b] \text{ if } a \preccurlyeq b\},$$
$$\Phi_{/\sim} = \{[\varphi] \mid \varphi \in \Phi\}.$$

Here for every FO formula φ, we define its quotient $[\varphi]$ to be the formula where each atom a(x) has been replaced by the atom [a](x).

We briefly describe how the above merge algorithm satisfies the desiderata of Section 3. First, based on the deductive closure, the results produced by the merge operation are conservative and sound under renaming. Namely, all consequences of the union of the source taxonomies and articulations are preserved, and no new information has been added to the merge result that could not be derived from the original taxonomies and articulations, where each taxon in $N_{/\sim}$ is equivalent to at least one source taxon. Merge results are also source maintaining. In a quotient taxonomy, each taxon $[a] = \{x_1, x_2, ...\}$ for $a \in N_{/\sim}$ implicitly carries its linkages to corresponding source taxa, where the source projection operation simply selects the desired source taxa of $[a]$. For instance, for the [A,1] taxon in Figure 4(c), $N = [T_1.\text{A}, T_2.1]$ such that the source projection $T_3.T_1$ is $\{(T_3.\text{A1}, T_1.\text{A}), (T_3.\text{B3}, T_1.\text{B}), (T_3.\text{C2}, T_1.\text{C})\}$. This projection can then be either rendered into a set of first-order axioms, or can be used to rewrite a set of articulations. In the former case, each pair in the projection (m, n) would add an axiom $\forall x.m(x) \leftrightarrow n(x)$ to Φ. In this case we can define $(T_1 \oplus_{A12} T_2).T_2 = (N_{T_3} \cup N_{T_2}, \preccurlyeq_{T_3}, \Phi_{T_3} \cup \Phi_{T_3.T_2})$. In the latter case, each taxon in the set of articulations matching the second element of a pair in the projection will be replaced with the name of the first element of that pair.

Furthermore, the merge operation itself is closed since it results in a taxonomy T as defined above. The merge is also commutative since it is possible to invert a set of articulations, and similarly associative under source projection. For quotient taxonomies, the merge operation is idempotent. That is, given two identical quotient taxonomies the same quotient taxonomy is returned.[3] Quotient taxonomies can be considered minimal views being synonym-free and consisting of the transitive reduction. And finally, as we describe further in the following section, the merge operation can scale-up to large taxonomies, in part due to CLEANTAX's use of RCC constraints.

5. EXPERIMENTS AND DISCUSSION

We have implemented the merge algorithm above within the CLEANTAX system and have tested our approach using a data set of nine aligned taxonomies for the plant genus *Ranunculus* [Peet 2005]. The experiments described below used the two largest taxonomies, one covering 218 taxa and the other covering 142 taxa, and 206 articulations between them created by a domain expert. Each taxonomy is three levels deep covering the genus, species, and variety biological ranks.

The first step in creating the merge is translating the taxonomies and articulations into monadic first-order logic and determining all the relationships implied by the resulting axioms. As described in earlier work [Thau 2008], this step is currently expensive, taking approximately 8 hours (using Prover9 [W.W. McCune 2008]) to determine the relations between each pair of taxa in the two described taxonomies. We expect future optimizations to reduce this time significantly (e.g., see [Thau and Ludascher 2007; Thau 2008]). Once these calculations have been made, the merge is computed very quickly. The limiting factor of the algorithm is the calculation of the transitive reduction, for which we used the tred filter that comes with the graphviz software package[4]. **Tred** uses a depth-first search algorithm of complexity $O(V * E)$ [Aho et al. 1972; Ioannidis and Ramakrishnan 1988]. In the current context, V is the number of taxa and E is the number of articulations describing inclusion (either $N \subsetneq M$, or $N\{\equiv, \subsetneq\}M$) maximally informed relations (mir). The other steps of the algorithm are $O(E)$ where E is the number of mir articulations. On average (after 5 runs with little variance between them) merging the two taxonomies described above took 84 milliseconds, 62% of which was spent determining the transitive reduction.

A primary advantage of the CLEANTAX framework is the ability to apply a variety of taxonomic constraints when reasoning and merging across taxonomies. To get a better sense for the impact of these taxonomic constraints, we divided the taxonomies into sub-taxonomies, each involving a species in one taxonomy and all the below-genus taxa connected to it in both taxonomies. Of the 81 sub-taxonomies thus created, 75 were consistent under all three of the global taxonomic constraints: non-emptiness, sibling-disjointness, and parent coverage. Calculating the merge for these smaller sub-taxonomies, which contained on average 8 taxa each, took on average 18 milliseconds, 99% of which was spent determining the transitive reduction.

Figure 6 shows the impact of the constraints on the merge of one of these subtaxonomies. The two sub-taxonomies for the species *Ranunculus hispidis* and their articulations are shown in Figure 6(a). When no additional assumptions are made, the merge results in Figure 6(b). It is important to recognize that in Figure 6(b), the lack of an edge between two taxa represents the situation where either a transitive edge has been removed in the transitive reduction or nothing is known about the relationship between the taxa. Thus, in Figure 6(b) the relationship between taxa [C] and [E] is completely unknown. Applying the non-emptiness constraint to all the taxa in the taxonomies results in the additional knowledge that taxa [C] and [E] are not disjoint.

Figure 6(c) represents the merge when the sibling-disjointness, coverage and nonemptiness constraints are assumed. In this merge, the taxon labeled [E] becomes

[3]Note that it is also straightforward to convert source taxonomies into corresponding quotient taxonomies.

[4]http://www.research.att.com/sw/tools/graphviz/

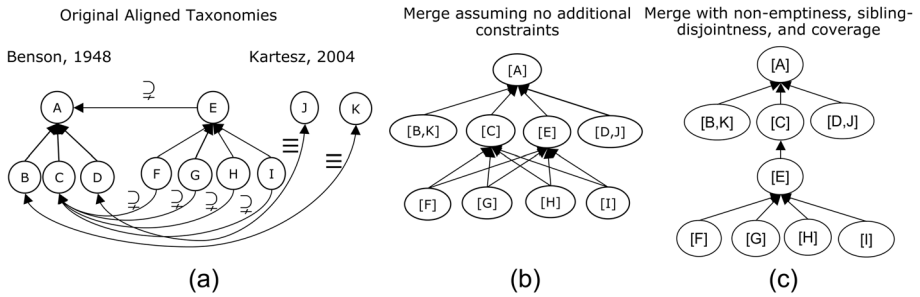

Figure 6. Merging *Ranunculus Hispidus* under Different Assumptions. For Clarity, the Disjointness Relations between Taxa in (c) are not Shown. See Text for Further Detail.

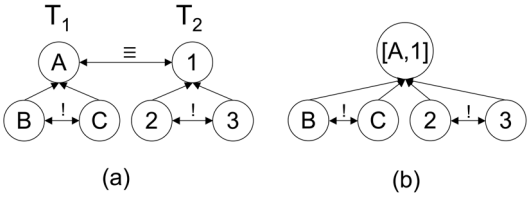

Figure 7. Constraints Placed on Taxonomies before the Merge May not Apply to the Result of the Merge.

a child of [C]. For clarity, the many disjointness relations between taxa in Figure 6(c) are not shown: the taxa [F], [G], [H], and [I] are mutually disjoint, the taxa [B,K], [C], and [D,J] are mutually disjoint, and each child of [E] is disjoint from [B,K] and [D,J].

It is important to note that when GTCs are applied to the taxonomies being merged, they are not automatically applied to the result of the merge. For example, in Figure 7, although the two taxonomies shown in (a) both exhibit the sibling disjointness constraint, the resulting merge in (b) does not; nothing is known about the relationship between taxa C and 2, for instance. Applying the sibling-disjointness constraint to the merged result would be adding additional information, violating the soundness desideratum. If the articulation provider expects taxa C and 2 to be disjoint, this articulation must be added to the alignment in Figure 7(a).

6. COMPARISON TO RELATED SYSTEMS

Fundamental differences between the OntoMerge, iPrompt and Chimæra approaches and that of CleanTax complicate comparisons between the systems. For example, OntoMerge does not have an explicit merge phase to compare with the CleanTax merge. iPrompt and Chimæra are interactive systems in which users merge ontologies by iteratively creating articulations and performing the merge, whereas the CleanTax merge assumes a set of articulations has been provided and performs the merge in one step. In all cases, the languages used for representing articulations differ, and the types of reasoning applied differ.

Table I details some of these differences between the systems.

The differences in how and when the systems apply reasoning during the merge operation impacts the result of their merge operation. Whereas CleanTax performs a

Table I. Comparing CLEANTAX to iPrompt, OntoMerge, Chimæra and iPrompt.

	CLEANTAX	OntoMerge	Chimæra	iPrompt
When Merge Happens	After articulations and reasoning	No real merge	During articulation	During articulation
Disjunctive Relation Support	Yes	No	No	No
Types of Reasoning Supported	Monadic FOL RCC	Forward and backward chaining	Extended FOL	Description logic
Result of Merge	Taxonomy	Knowledge base	Taxonomy	Ontology
Support for Roles and Union	No	Yes	Yes	Yes

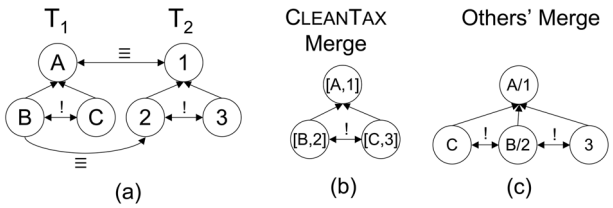

Figure 8. Comparing Merges for the Taxonomies in (a) under the Parent-Coverage Constraint. CLEANTAX Correctly Merges Taxa C and 3 (b) while the Others Do Not (c).

deductive closure before the merge occurs, neither iPrompt nor Chimæra appear to do so. And while OntoMerge supports an extended full first-order logic, its reasoning over that logic is quite restricted. An effect of these differences in reasoning may be seen in the results of the merge in Figure 8. When the child-disjointness, parent coverage and non-emptiness GTCs are in place, CLEANTAX merges taxa C and 3. Chimæra, iPrompt and OntoMerge each leave 3 and C distinct.

7. CONCLUSION

We have presented a formal approach for merging taxonomies within the CLEANTAX system. This work is motivated by current problems in managing, integrating, and exploiting large biological classifications including species taxonomies. As such, we have also identified a number of requirements related to merging taxonomies, and have described how our proposed merge approach satisfies them. We have also presented initial experimental results of an implementation of our merge approach, using a number of real-world species taxonomies and articulations created by a domain expert. Our findings suggest that the merge approach is well suited for handling large taxonomies and complex sets of articulations.

REFERENCES

AHO, A. V., M. R. GAREY, AND J. D. ULLMAN. 1972. The transitive reduction of a directed graph. *SIAM Journal on Computing 1*, 2:131–137.

BAILEY, K. D. 1994. *Typologies and Taxonomies: An Introduction to Classification Techniques.* Sage Publications, Inc.

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

BRACHMAN, R. 1983. What is-a is and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer* 16:30–36.

CÔTÉ, R., D. ROTHWELL, AND L. BROCHU, Eds. 1993. *SNOMED international: the systematized nomenclature of human and veterinary medicine*, 3rd ed. College of American Pathologists, Northfield, Ill.

DOOLITTLE, W. F. 1999. Phylogenetic classification and the universal tree. *Science 284*, 5423: 2124–2128.

DOU, D., D. MCDERMOTT, AND P. QI. 2004. Ontology translation on the semantic web. In *International Conference on Ontologies, Databases and Applications*.

EHRIG, M. 2007. *Ontology Alignment: Bridging the Semantic Gap*. Semantic Web and Beyond Computing for Human Experience, vol. 4. Springer.

EUZENAT, J. 2004. State of the art on ontology alignment. http://www.starlab.vub.ac.be/publications/kweb-223.pdf.

FRANZ, N. M., R. K. PEET, AND A. S. WEAKLEY. 2007. On the use of taxonomic concepts in support of biodiversity research and taxonomy. In *The New Taxonomy, Systematics Association Special Volume Series 74*, Q. D. Wheeler, Ed. Taylor and Francis, Boca Raton, FL., 61–84.

HENIKOFF, S., E. A. GREENE, S. PIETROKOVSKI, P. BORK, T. K. ATTWOOD, AND L. HOOD. 1997. Gene families: The taxonomy of protein paralogs and chimeras. *Science 278*, 5338:609–614.

IOANNIDIS, Y. E. AND R. RAMAKRISHNAN. 1988. An efficient transitive closure algorithm. In *Proceedings of the 14th International Conference Very Large Databases*. Los Angeles, California, 382–394.

JONSSON, P. AND T. DRAKENGREN. 1997. A complete classification of tractability in RCC-5. *Journal of Artificial Intelligence Research 6*, 211–221.

JUNG, J. J. 2006. Taxonomy alignment for interoperability between heterogeneous digital libraries. In *Digital Libraries: Achievements, Challenges and Opportunities* (2006-11-20), S. Sugimoto, J. Hunter, A. Rauber, and A. Morishima, Eds. Lecture Notes in Computer Science, vol. 4312/2006. Springer, Berlin/Heidelberg, 274–282.

KENNEDY, J., R. KUKLA, AND T. PATERSON. 2005. Scientific names are ambiguous as identifiers for biological taxa: Their context and definition are required for accurate data integration. In *Second International Workshop on Data Integration in the Life Sciences (DILS)*. LNCS 3615. 80–95.

KIM, J., M. JANG, Y.-G. HA, J.-C. SOHN, AND S.-J. LEE. 2005. MoA: OWL ontology merging and alignment tool for the semantic web. In *Proceedings of the International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE* (2005-07-13), M. Ali and F. Esposito, Eds. Lecture Notes in Computer Science, vol. 3533. Springer, 722–731.

KLEIN, M. 2001. Combining and relating ontologies: an analysis of problems and solutions. In *Workshop on Ontologies and Information Sharing, IJCAI-2001*, A. Gomez-Perez, M. Gruninger, H. Stuckenschmidt, and M. Uschold, Eds. Seattle, USA.

KOPERSKI, M., M. SAUER, W. BRAUN, AND S. GRADSTEIN, 2000. *Referenzliste der Moose Deutschlands*. Vol. 34. Schriftenreihe Vegetationsk.

KOTIS, K. AND G. A. VOUROS. 2004. The HCONE approach to ontology merging. In *Proceedings of the First European Semantic Web Symposium* (2004-09-16), C. Bussler, J. Davies, D. Fensel, and R. Studer, Eds. Lecture Notes in Computer Science, vol. 3053. Springer, 137–151.

KOTIS, K., G. A. VOUROS, AND K. STERGIOU. 2006. Towards automatic merging of domain ontologies: The HCONE-merge approach. *J. Web Sem. 4*, 1:60–79.

LINNAEUS, C. 1758. *Systema Naturae*. Laurentii Salvii, Stockholm.

MCGUINNESS, D. L., R. FIKES, J. RICE, AND S. WILDER. 2000a. The Chimaera ontology environment. In *Proceedings of the 17th National Conference on Artificial Intelligence* (2002-

01-03). AAAI Press/The MIT Press, 1123–1124.

MCGUINNESS, D. L., R. FIKES, J. RICE, AND S. WILDER. 2000b. An environment for merging and testing large ontologies. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04) of the Seventh International Conference on Principles of Knowledge*. Breckenridge, Colorado.

NOY, N. F. AND M. A. MUSEN. 2003. The PROMPT suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies 59*, 6:983–1024.

ORENGO, C., A. MICHIE, S. JONES, D. JONES, M. SWINDELLS, AND J. THORNTON. 1997. CATH - a hierarchic classification of protein domain structures. *Structure 5*, 8 (August), 1093–1108.

PEET, R. K. 2005. Ranunculus dataset.

RANDELL, D. A., Z. CUI, AND A. COHN. 1992. A spatial logic based on regions and connection. In *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, B. Nebel, C. Rich, and W. Swartout, Eds. Morgan Kaufmann, San Mateo, California, 165–176.

RENZ, J. AND B. NEBEL. 1999. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artificial Intelligence 108*, 1-2:69–123.

RIAZANOV, A. AND A. VORONKOV. 2002. The design and implementation of VAMPIRE. *AI Communications 15*, 2-3:91–110.

STAFF, S. S. 1975. *Soil taxonomy. A basic system of soil classification for making and interpreting soil surveys*. Number 436 in Soil Conservation Service Agricultural Handbook. United States Department of Agriculture.

STUMME, G. AND A. MAEDCHE. 2001a. FCA-MERGE: Bottom-Up Merging of Ontologies. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*. 225–234.

STUMME, G. AND A. MAEDCHE. 2001b. Ontology merging for federated ontologies on the semantic web. In *Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001)*. 413–418.

THAU, D. 2008. Reasoning about taxonomies and articulations. In *Ph.D. '08: Proceedings of the 2008 EDBT Ph.D. workshop*. ACM, New York, NY, USA, 11–19.

THAU, D. AND B. LUDASCHER. 2007. Reasoning about taxonomies in first-order logic. *Ecological Informatics 2*, 3:195–209.

W.W. McCune. 2008. Prover 9: http://www.cs.unm.edu/ mccune/prover9/.

David Thau is a Ph.D. student at the Department of Computer Science at the University of California, Davis, USA. He holds M.Sc. degrees in Electrical Engineering and Computer Science, and in Experimental Psychology from the University of Michigan at Ann Arbor, and a B.A. degree in Cognitive Science from the University of California at Los Angeles. His research interests include ontology alignment, data integration, annotation, scientific data management, and biodiversity informatics.

Shawn Bowers is a computer scientist at the UC Davis Genome Center, working closely with domain scientists in ecology, bioinformatics, and other disciplines. He is a member of the Data and Knowledge Systems Lab where he conducts research in conceptual data modeling, data integration, and scientific workflows. He is an active member of the Kepler Scientific Workflow project, where he has contributed to the design and development of Kepler extensions for managing complex scientific data, capturing and exploring data provenance, and ontology-based approaches for organizing and discovering workflow components. Shawn holds a Ph.D. and a M.Sc. in Computer Science from the OGI School of Science and Engineering, and a B.Sc. in Computer and Information Science from the University of Oregon. Prior to joining the UC Davis, he was a Postdoctoral Researcher at the San Diego Supercomputer Center.

Bertram Ludäscher is an Associate Professor in the Department of Computer Science at UC Davis and a faculty member of the UC Davis Genome Center. His current research areas include modeling, design, and optimization of scientific workflows and databases, data and workflow provenance, and knowledge representation and reasoning for scientific workflows. Until his move to Davis he was a member of the NIH-funded Biomedical Informatics Research Network Coordination Center (BIRN-CC) at UC San Diego and a co-PI of the NSF/ITR Geoscience Network (GEON). He is currently actively involved in several large-scale, collaborative scientific data and workflow management projects, including the NSF/ITR Science Environment for Ecological Knowledge (SEEK), the DOE Scientific Data Management Center (SciDAC/SDM), and two NSF projects on Cyberinfrastructure for Environmental Observatories (CEOP/COMET and CEOP/REAP). Dr. Ludaescher is also one of the co-initiators of the cross-project Kepler collaboration and PI of the new Kepler/CORE project. He received his MS in Computer Science from the University of Karlsruhe in 1992 and his PhD from the University of Freiburg in 1998 (Germany). From 1998 to 2004 Dr. Ludaescher worked as a research scientist at the San Diego Supercomputer Center, UCSD.