World Wide Web: Internet and Web Information Systems, 8, 261–285, 2005

DOI: 10.1007/s11280-005-0360-8

Towards Ontology Generation from Tables

YURI A. TIJERINO ontologist@ksc.kwansei.ac.jp
Kwansei Gakuin University, Japan

DAVID W. EMBLEY embley@cs.byu.edu
DERYLE W. LONSDALE lonz@byu.edu
YIHONG DING ding@cs.byu.edu
Brigham Young University

GEORGE NAGY nagy@ecse.rpi.edu
Rensselaer Polytechnic Institute

Published online: 1 August 2005

Abstract

At the heart of today's information-explosion problems are issues involving semantics, mutual understanding, concept matching, and interoperability. Ontologies and the Semantic Web are offered as a potential solution, but creating ontologies for real-world knowledge is nontrivial. If we could automate the process, we could significantly improve our chances of making the Semantic Web a reality. While understanding natural language is difficult, tables and other structured information make it easier to interpret new items and relations. In this paper we introduce an approach to generating ontologies based on table analysis. We thus call our approach TANGO (Table ANalysis for Generating Ontologies). Based on conceptual modeling extraction techniques, TANGO attempts to (i) understand a table's structure and conceptual content; (ii) discover the constraints that hold between concepts extracted from the table; (iii) match the recognized concepts with ones from a more general specification of related concepts; and (iv) merge the resulting structure with other similar knowledge representations. TANGO is thus a formalized method of processing the format and content of tables that can serve to incrementally build a relevant reusable conceptual ontology.

Keywords: ontology, table understanding, ontology generation, semantic web

1. Introduction

The exponential increase in new knowledge that characterizes our modern age of information technology precludes depending solely on individual effort to keep up with new information. We must therefore develop new ways of "keeping up," and we must develop them quickly. The Semantic Web [3] offers a promise that we can "keep up" by allowing software agents to roam in cyberspace in our behalf, where they can gather information of interest and synergistically assist us in decision making and in negotiating for our wants and desires. This ideal, however, relies on agents being able to find and manipulate useful information, which, in turn, relies on having an abundance of ontologically annotated data.

Unfortunately, ontologically annotating information repositories is nontrivial. If we could automate the process, or at least make the process semiautomatic, we could significantly improve our chances of making the Semantic Web a reality. In this vision paper, we propose and describe a unified framework for ontology generation from tables grounded on previous work that meets this challenge.

Motivated by our belief that inference about unknown objects and relations in a known context can be automated, we describe an information gathering engine that assimilates and organizes knowledge. While understanding context in a natural language setting is difficult, structured information such as tables[1] makes it easier to interpret new items and relations. We organize the new knowledge we gain from "understanding" tables as an ontology and thus we call our information-gathering engine *TANGO* (Table ANalysis for Generating Ontologies) [45].

Our approach to ontology generation can be considered as semiautomated, applied "ontological engineering" [38]. However, instead of humans collaborating to design an ontology, we enable tables to "collaborate" to design an ontology. In a sense, this is the same because TANGO assembles information from specific instances of human-created tables.

We present the details of our vision for TANGO as follows. In Section 2 we describe the basics of our approach to automated knowledge gathering. For illustration we use the domain of geopolitical facts and relations, where relevant empirical data is widely scattered but often presented in the form of tables. Using this domain, we illustrate the specifics of our ideas in: Section 3, where we show that most semi-structured, factual data is table-equivalent; Section 4, where we show how to discover ontologies from tables; Section 5, where we show how to discover mappings between ontologies; and Section 6, where we investigate how to merge ontologies. Section 7 describes potential applications where the results of this work could make a significant impact, particularly as related to the Semantic Web. We make some concluding remarks in Section 8.

2. Ontology generation approach

Our table analysis approach to ontology generation addresses the principled creation of ontologies based on the content of canonicalized tables. TANGO operates in four steps:

1. Recognize and canonicalize table information.
2. Construct mini-ontologies:[2] from canonicalized tables.
3. Discover inter-ontology mappings.
4. Merge mini-ontologies into a growing application ontology.

We will describe these steps in the following sections. First, though, some general remarks on knowledge sources are necessary.

In support of these four steps TANGO relies on auxiliary information. This auxiliary information includes dictionaries and lexical data (including WordNet [22], natural language parsers, and data frames [14], which are similar in intent to the base knowledge for ontologies proposed in [44].

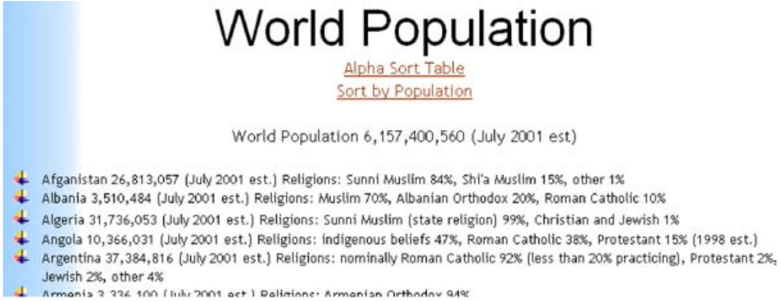
We are creating our own data frame library. In essence, each data frame in the library encapsulates the essential properties of one of the common data formats in the real world such as dates, currencies, numbers, percentages, weights, measures, and so forth. A data frame extends an abstract data type to include not only an internal data representation and applicable operations but also detailed representational and contextual information that allows a string that appears in a text document to be classified as belonging to the data frame. Data frames can be thought of as recognizers that help us associate unstructured data with common concepts. Thus, for example, a data frame for a longitude/latitude location on the earth's surface has regular expressions that recognize all forms of longitude and latitude values and regular expression recognizers for keywords such as "lon.", "lat.", "degrees north", "degrees east", and "position".

Given the data frame library and other auxiliary information mentioned above, we begin with the first step: recognize and canonicalize table information. We illustrate this step in the following section.

3. Table recognition and canonicalization

Although many consider the idea of a table to be simple, a careful study (e.g. [30]) reveals that the question "What constitutes a table?" is indeed difficult to answer. As only two of thousands of examples, does the information in Figure 1 constitute a table? What about the information in Figure 2?

We have chosen to define a table indirectly through information canonicalization. Working backwards, we first consider relations in a relational database to be tables in a canonical form. Using a standard, formal definition of a relational database table [32], we can define a canonical table as follows. A *schema* for a canonical table is a finite set $\{L_1, \ldots, L_n\}$ of label names or phrases, which are simply called *labels*. Corresponding to each label L_i, $1 \leq i \leq n$, is a set D_i, called the *domain* of L_i. Let $D = D_1 \cup \ldots \cup D_n$. A *canonical table* T with table schema S is a set of functions $T = \{t_1, \ldots, t_m\}$ from S to D with the restriction that for each function $t \in T$, $t(L_i) \in D_i$, $1 \leq i \leq n$.



World Population

Alpha Sort Table

Sort by Population

World Population 6,157,400,560 (July 2001 est)

- Afganistan 26,813,057 (July 2001 est.) Religions: Sunni Muslim 84%, Shi'a Muslim 15%, other 1%
- Albania 3,510,484 (July 2001 est.) Religions: Muslim 70%, Albanian Orthodox 20%, Roman Catholic 10%
- Algeria 31,736,053 (July 2001 est.) Religions: Sunni Muslim (state religion) 99%, Christian and Jewish 1%
- Angola 10,366,031 (July 2001 est.) Religions: indigenous beliefs 47%, Roman Catholic 38%, Protestant 15% (1998 est.)
- Argentina 37,384,816 (July 2001 est.) Religions: nominally Roman Catholic 92% (less than 20% practicing), Protestant 2%, Jewish 2%, other 4%
- Armenia 3,336,100 (July 2001 est.) Religions: Armenian Orthodox 94%

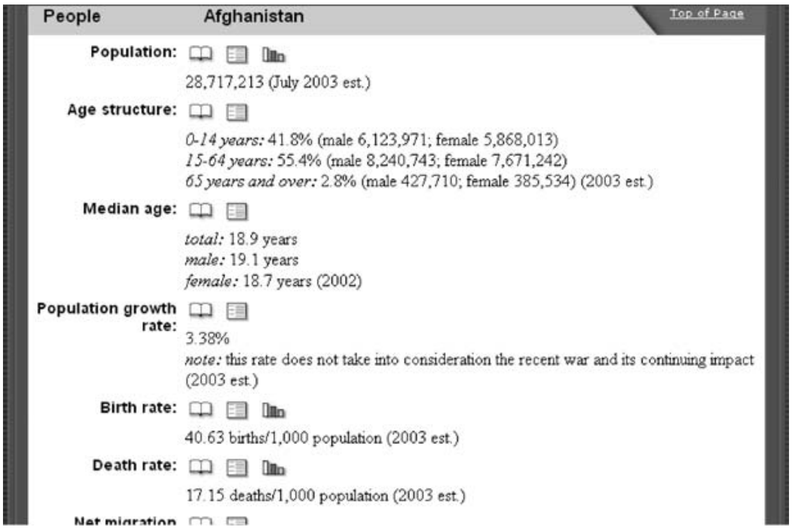
Figure 1. Partial page of world religious populations [12].

As is common for relational databases, we often display tables in two dimensions. When we display a table two dimensionally, we fix the order of the labels in the schema for each function and factor these labels to the top as column headers. Each row in the table constitutes the domain values for the corresponding labels in the column headers. Thus, for example, we can display the canonical table $\{\{(A, 1), (B, 2), (C, 3)\}, \{(A, 4), (B, 5), (C, 6)\}\}$ as follows.

A	B	C
1	2	3
4	5	6

Displayed in this form, a canonicalized table is simply called a *table*. Whether the original information should be called a "table" may be debatable. To avoid the argument, whenever there may be doubt (e.g. Figures 1 and 2), we will refer to the information as *table-equivalent data*.

When we canonicalize the table-equivalent data in Figure 1, we obtain Table 1.[3] To canonicalize the table-equivalent data in Figure 2 to obtain Table 2, we first recognize that the data is split across many web pages; each page has the same data but for a different country. Thus, each page is itself a function from the labels, which are phrases on the left composed with the sub-label phrases on the right, to domain values, which are non-label



People Afghanistan

Population:
28,717,213 (July 2003 est.)

Age structure:
0-14 years: 41.8% (male 6,123,971; female 5,868,013)
15-64 years: 55.4% (male 8,240,743; female 7,671,242)
65 years and over: 2.8% (male 427,710; female 385,534) (2003 est.)

Median age:
total: 18.9 years
male: 19.1 years
female: 18.7 years (2002)

Population growth rate:
3.38%
note: this rate does not take into consideration the recent war and its continuing impact (2003 est.)

Birth rate:
40.63 births/1,000 population (2003 est.)

Death rate:
17.15 deaths/1,000 population (2003 est.)

Net migration

Figure 2. Partial page from people in the 2003 CIA World Factbook [49].

Table 1. Partial canonicalized table for world religious populations [12].

		Religion					
Country	Population (July 2001 est.)	Albanian Orthodox	Muslim	Roman Catholic	Shi'a Muslim	Sunni Muslim	Other
Afghanistan	26,813,057				15%	84%	1%
Albania	3,510,484	20%	70%	10%			
...							

Table 2. Partial canonicalized table for people in the 2003 CIA World Factbook [49].

		Median age (2002)			
Country	Population (July 2001 est.)	Total	Male	Female	Population growth rate (2003 est.)
Afghanistan	28,717,213	18.9 years	19.1 years	18.7 years	3.38%*
Albania	3,582,205	26.5 years	24.8 years	28.1 years	1.03%
...					

*Note: this rate does not take into consideration the recent war and its continuing impact

values on the right. In addition, there are explanatory comments, which we can standardize as footnotes.

So, how can we determine whether we have table-equivalent data, and how can we turn table-like information into canonicalized tables? Since we have defined a table indirectly and by construction, we only need to answer the second question. If we can turn semi-structured information into a canonicalized table, we can declare that the semi-structured information is table-equivalent data and that the canonicalized table is a table.

There is a spectrum of cases to be considered. At the one extreme, we may already have information presented as a canonicalized table. All relational database tables, for example, are canonicalized tables, and many tables on the web appear essentially in canonicalized form. Other web tables, however, pose problems such as tables displayed piecemeal, tables spanning multiple pages, tables with no `<tabel>` tag, folded tables, tables with factored rows, tables with linked subtables, and table rows with additional linked row values, all of which we have worked with in previous research [20] related to data extraction from tables.[4] Some tables, more difficult to interpret, include features such as tables nested within table rows, folded table rows, and tables with both column and row headings. Table-equivalent data that does not have a typical two-dimensional layout is more difficult, but we have experimented with techniques to interpret them. Using ideas developed in [20], for example, we can distinguish label text versus value text from the World Factbook in

Table 3. Preliminary table generated from Figure 1.

Char string	Num string	Parenthesis data	Same char string	Combined data	Combined data	Combined data
Afghanistan	26,813,057	(July 2001 est.)	Religions:	Sunni Muslin 84%	Shi'a Muslim 15%	Other 1%
Albania	3,510,484	(July 2001 est.)	Religions:	Muslim 70%	Albanian Orthodox 20%	Roman Catholic 10%

Figure 2 by comparing the pages—the label text stays constant from page to page whereas the value text changes.

As an example of how TANGO interprets tables, we describe the process it uses to generate canonicalized Table 1 from Figure 1.

Segment Page: TANGO recognizes the different parts of the page. Simple analysis of the document HTML source indicates that there are two main regions in the document. One is the table data as indicated by `<table>` and `</table>` which encapsulate the records for each of the countries. The second part is everything else. Though often indicative of a table, we note that neither the presence nor absence of `<table>` and `</table>` tags dictates the presence nor the absence of a table. It is possible, for example, to create the same structure for Figure 1 using itemization tags such as `<li>` and `</li>`.

Identify Columns: Analysis of patterns using techniques described in [11] leads TANGO to the segmentation shown in Table 3. Recognizing that there are common patterns in records, TANGO can extract different columns for this table. For instance, TANGO can recognize that there is a character string (i.e. a country), followed by a number string (i.e. a population), followed by a mixed character and number string in parentheses, followed by the string "Religions:", followed by a comma-separated combination of character strings (i.e. a religion) and a value (i.e. a percentage).

Apply Data Frames: Using our data frames, TANGO recognizes that the strings in Column 1 are country names; thus TANGO names the first column "Country". WordNet specifies instances of country names as part of its database. In addition, there are other geopolitical databases such as http://www.gazeteer.com/, which include names of countries. It is fairly simple to create data frames that capture lexical information of this nature and then recognize it as lexical instances of ontological concepts. In other words, the name of countries are lexical in nature and therefore we use lexical sets in our data frames to recognize them as instances of concepts in our ontology. We are making a trade off between conceptual information and lexical information and it makes more sense to put lexical information in a lexicon as opposed to the ontology. Our data frame based approach allows us to do this in a simple, but powerful manner. Further TANGO recognizes percentages, which incidentally add up to 100% in each record. Even in the absence of a data frame for recognizing religion names, TANGO can detect the pattern that a string (often the same string) precedes each

Table 4. Second preliminary table generated from Table 3.

Country	Num String	Paren. data	Same char string	Albanian Ortho-dox	Muslim	Roman Catholic	Shi'a Muslim	Sunni Muslim	other
Afghanistan	26,813,057	(July 2001 est.)	Religions:				15%	84%	1%
Albania	3,510,484	(July 2001 est.)	Religions:	20%	70%	10%			
. . .									

value, which leads to the inference that the character strings should be promoted as column names. This process results in preliminary Table 4.

Apply WordNet heuristics: Table 4 can be further canonicalized by applying other techniques. In this case TANGO recognizes that the 4th column consistently shows the same item, namely "Religions:". Using WordNet TANGO recognizes that the labels for Columns 5 through 10 are hyponyms of the *Religion* concept. This leads it to promote "Religions" to a parent column for column labels that refer to religions. A similar situation occurs with the string "(July 2001 est.)" in Column 3. WordNet, however, does not recognize "(July 2001 est.)" as a hypernym for the religion strings. Therefore, TANGO leaves it as is for now.

Apply other heuristics: Next TANGO recognizes, from the table label in Figure 1, "World Population", that the table is about population. Since the string "(July 2001 est.)" also appears in the header of the table near "World Population", TANGO infers that "Population" refers to the numbers in Column 2 and that "(July 2001 est.)" is part of the label as well. Table 1 shows the result from this and the previous steps.

In addition to the process described above, it should be noted that we not only canonicalize the structure of the tables as explained, but we also use data frames to canonicalize the values. Hence for each common data item we have all values in the same units, and we can display values with the same (or different) precision, as desired. For example, we can use meters rather than feet or yards, and we can display population values in (rounded) millions if we wish.

Table 5. Partial canonicalized table for geography in the 2003 CIA World Factbook [49].

Country	Location description	Geographic coordinates
Afghanistan	Southern Asia, north and west of Pakistan, east of Iran	33 00 N, 65 00 E
Albania	Southeastern Europe, bordering on the Adriatic Sea and Ionian Sea, between Greece and Serbia and Montenegro	41 00 N, 20 00 E
. . .		

In discussing the remaining three steps in the subsequent sections, we assume for these examples that we have all the information from the partial tables in Tables 1 and 2, and from the partial canonicalized tables[5] in Tables 5–8.

4. Construction of mini-ontologies

We have chosen to use OSM [19] for the representation of our ontologies in TANGO because of the richness this representation affords us.[6] OSM is an expressive object-oriented model for system analysis, specification, design, implementation, and evolution [15]. The structural components of OSM include object sets[7] and relationship sets. OSM supports the abstraction of generalization and specialization because an object set can be a superset or subset of another object set. Relationship sets support n-ary relationships among objects sets, whole/part aggregations, and set/member associations. A relationship set also allows for the definition of cardinality constraints among object sets.

Figure 3 gives a graphical representation of mini-ontologies that capture the conceptual model instances for our six sample canonicalized tables in Tables 1, 2, 5–8. We refer to a table-specific ontology as a mini-ontology. It is an ontology because it captures concepts, relationships, and constraints related to the table. It is a mini-ontology because it does not expand its concepts beyond the context of the table, which is usually small compared to typical ontologies.

In the OSM notation,[8] boxes represent *object sets*—dashed if displayable (e.g. *Population* in Figure 3(b) and *Longitude* in Figure 3(e)) and not dashed if not displayable because their objects are represented by object identifiers (e.g. *Geopolitical Entity* in Figure 3(d)). With each object set we can associate a data frame to give it a rich description of its value set. We represent actual objects by labeled dots (e.g. *July 2001* in Figure 3(a)). Lines connecting object sets or object sets and objects are *relationship sets*; these lines may be hyperlines (hyperedges in hypergraphs) when they have more than two connections to object sets (e.g. the relationship set among the attributes *Country*, *Religion*, and *Percent* in Figure 3(a)). Optional or mandatory *participation constraints* respectively specify whether objects in a connected relationship may or must participate in a relationship set (an “o” on a connecting relationship-set line designates *optional* while the absence of an “o” designates *mandatory*). Thus, for example, the mini-ontology in Figure 3(e) declares that a *Place* must have a *Name* and may (but need not) have an *Elevation*. Arrowheads on lines specify *functional constraints*—for n-ary relationship sets, $n > 2$, acute versus obtuse angles on the origin of the arrows disambiguate situations where tuples of two or more tails or heads form the domain or co-domain in the function. Thus, according to Figure 3(e), a *Place* has a single *USGS Quad*, and *Geographic Coordinates* and the pair *Longitude* and *Latitude* have a one-to-one correspondence. Open triangles denote *generalization/specialization hierarchies* (ISA hierarchies, subset constraints, or inclusion dependencies), so that in Figure 3(d) *Continent*, *Country*, and *City* are all specializations of *Geopolitical Entity* and thus are each themselves geopolitical entities. We can constrain ISA hierarchies by partition ($\uplus$), union ($\cup$), or mutual exclusion ($+$) among specializations or by intersection ($\cap$) among generalizations. Filled-in triangles denote part/whole, part-of, or *aggregation hierarchies*.

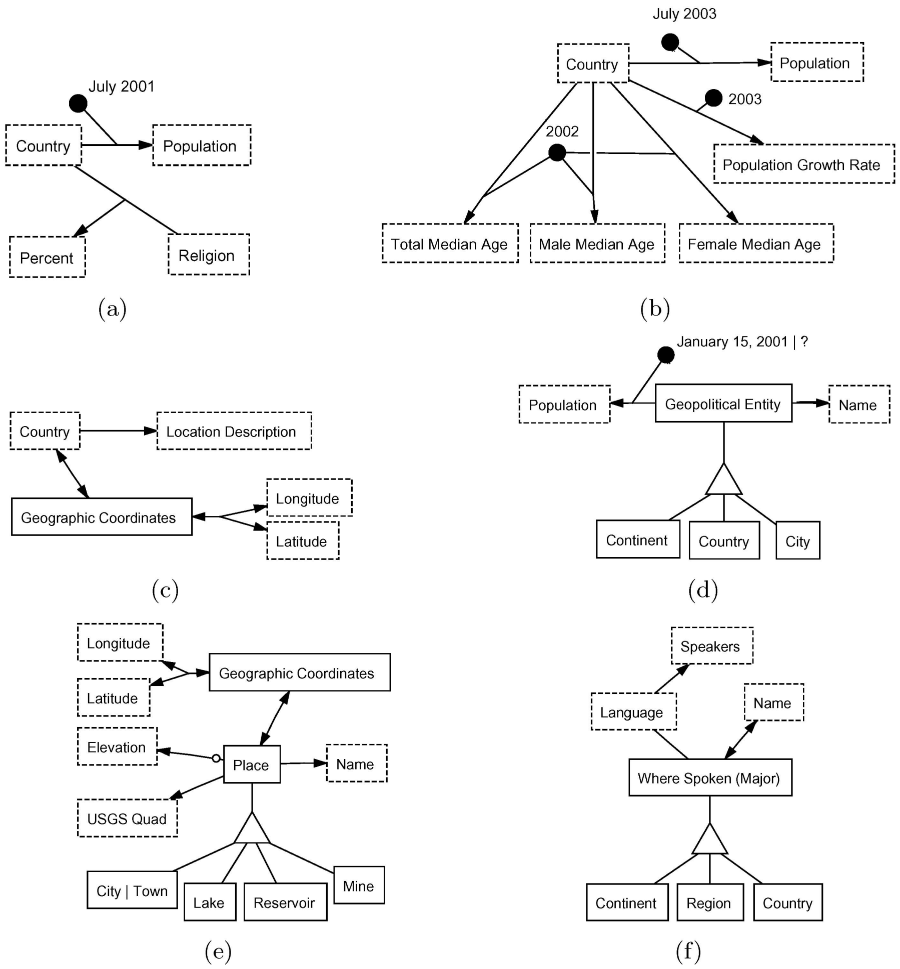


Figure 3. Mini-ontologies constructed from Tables 1, 2, 5–8.

Based on this representation for mini-ontologies, the construction of mini-ontologies from tables becomes the process of reverse engineering the tables into conceptual models (i.e., mini-ontologies). The literature describes many techniques for reverse engineering relational databases and schemas into conceptual models and entity-relationship models [9,25,35]. In [21], we introduced an interactive approach for reverse engineering, upon which we expand further in this paper.

To construct mini-ontologies from tables, TANGO must discover what concepts (object sets) are involved and how they are related (relationship sets). It must also determine the constraints that hold over the relationship sets (functional, mandatory/optional participation, aggregations) and among the object sets (generalization/specialization). It does so by mining the table values for constraints such as functional dependencies and inclusion dependencies [27,34]; by observing mandatory and optional patterns in the data; by using lexicons to find hypernyms/hyponyms and kind-of relationships among terms; and by using data frames to recognize values in labels, tables with multiple concept values in a column, and tables with columns whose values should be split into two or more concepts.

As an example, we obtain the mini-ontology in Figure 3(a) from Table 1 as follows. *Country* is a key and appears in a leftmost column, strongly suggesting that it should be the tail side of functional dependencies. *Population* depends on *Country*, but because *July 2001 est.* has been factored out as a value associated with the attribute of *Population*, *Population* also depends on *July 2001 est.*. We anticipate the need to use abbreviation dictionaries along with WordNet, to determine that *est.* is an adjective for the value *July 2001* and drop it. Thus, we obtain the functional dependency *Country*, *July 2001* → *Population* and hence the functional ternary relationship among these three as Table 1 shows. Knowledge from the data frame library recognizes that the values in the *Religion* columns are *Percent* values. The religions, which either could be object sets that hold percent values or could themselves be values in a *Religion* object set, are values since there are many of them (our current threshold is five). Given that religions are values, we therefore have a ternary relationship among *Country*, *Religion*, and *Percent*. It is possible, using constraint mining as described in [27,34], to determine that *Country* and *Religion* together functionally determine *Percent*.

Although creation of the remaining mini-ontologies is similar, there are several interesting observations we can make.

(1) The features of Table 2 are very similar to the features in Table 1. We therefore process them in the same way, obtaining the two functional ternaries depending on *July 2003* and *2003*. This time, however, the *Median Age* subcategories should be object sets rather than values because there are fewer than five.

(2) For Figure 3(c), our data frame library helps us recognize the *Longitude* and *Latitude* values and place them pairwise in a one-to-one correspondence with *Geographic Coordinates*. Further, since both *Country* and *Geographic Coordinates* are keys, they are in a one-to-one correspondence.

(3) For Figure 3(d), WordNet not only knows about continents, countries, and cities, it also knows specific continents and some specific countries and cities. WordNet therefore helps us realize that the unnamed column in Table 6 contains three categories, and it gives us *Object* as a common hypernym for the name of the generalization. Further, recognition that *Object* is a common hypernym for thousands of terms would prompt an IDS (*Issue*/*Default*/*Suggestion*) statement [4] raising the *Issue* that the term *Object* is likely to be far too general, stating that the *Default* is to do nothing, and making a *Suggestion* that the user choose a more meaningful name. We assume that the user follows the suggestion and chooses *Geopolitical Entity* as the name.

(4) For Figure 3(e), natural language processing helps us recognize that the column in Table 7 with label *Type* contains instances that represent different concepts, namely *City*|*Town*, *Lake*, *Reservoir* and *Mine*. Since each *Place* is one of these concepts, each of which has a *Name*, we make *Place* a generalization of these concepts and then factor out *Name* from each concept and associate it with *Place*. Our data frame

Table 6. Partial canonicalized table for largest populations [48].

	Population
Asia	3,674,000,000
Africa	778,000,000
. . .	
New York City, New York	8,040,000
Los Angeles, California	3,700,000
. . .	
Mumbai, India	12,150,000
Buenos Aires, Argentina	11,960,000
. . .	
China	1,256,167,701*
India	1,017,645,163*
. . .	

*January 15, 2000

Table 7. Partial canonicalized table for US topographical maps [46].

Place	Type	Elevation∗	USGS Quad	Lat	Lon
Bonnie Lake	reservoir	unknown	Seivern	33 72 N	81 42 W
Bonnie Lake	lake	unknown	Mirror Lake	40 71 N	110 88 W
. . .					
New York	town/city	unknown	Jersey City	40 71 N	74 01 W
New York	town/city	149 meters	Leagueville	32 17 N	95 67 W
New York	mine	unknown	Heber City	40 62 N	111 49 W
. . .					

*Elevation values in this table are approximate, and often subject to a large degree of error. If in doubt, check the actual value on the map

library recognizes that *Lat* and *Lon* are *Latitude* and *Longitude* and that together they are *Geographic Coordinates*. Table 7 indicates that the *Geographic Coordinates* functionally determines *Place* and also that *Place* is unique (although *Place* does not have unique names). Further, some of the *Elevation* values are *unknown*, which lets us conclude that the *Elevation* can be optional.

(5) For Figure 3(f), we can recognize and disregard the rank (*Pos*) numbers in Table 8. Further, for Figure 3(f), we use natural language processing and WordNet to find continents, countries, and regions as concepts that are all specializations of *Where Spoken*. Further, they tell us that *Major* is not a noun and therefore not another object or concept. Constraint mining [27,34] leads to an understanding that the relationship

Table 8. Partial canonicalized table for most spoken languages [40].

Pos	Language	Speakers	Where spoken (major)
1	Mandarin	885,000,000	China, Malaysia, Taiwan
2	Spanish	332,000,000	South America, Central America, Spain
3	English	322,000,000	USA, UK, Australia, Canada, New Zealand
. . .			

from *Language* to *Speakers* is functional, that the relationship between *Language* and *Where Spoken* is many-to-many, and that the relationship between *Where Spoken* and the *Name* of each *Continent*, *Region*, and *Country* is one-to-one.

5. Discovery of inter-ontology mappings

Although data, schema and ontology integration has been explored in great depth by many in the past [6,26,42], this is still an open area of research. Our approach to discovering inter-ontology mappings is multifaceted, which means that we use all evidence at our disposal to determine how to match concepts. In using this evidence we look not only for direct matches as is common in most schema matching techniques [2,13,31,42], but also for indirect matches [4,50]. Thus, for example, we are able to split or join columns to match the single *Geographic Coordinates* column in Table 5 with the pair of columns, *Lat* and *Lon*, in Table 7; we are also able to divide the values in the *Place* column in Table 7 into several different object sets. We discuss relevant techniques in the following paragraphs.

Label matching. We have successfully experimented with machine-learned decision trees over WordNet features such as synonyms,[9] word senses, and hypernyms/hyponyms [18]. In [8] we have also successfully experimented with modified soundex matching, Levenshtein edit-distance, and longest common subsequence. These modified measures are particularly useful when name matching is obscured by shortened mnemonic names, abbreviations, and acronyms, which are sometimes found in table headers.

Value similarity. We [18] and others (e.g. [29]) have successfully used machine-learned rules to match object sets based on value characteristics such as alphanumeric features including length, alpha/numeric ratio, space/nonspace ratio and numeric features such as mean and variance. Gaussian value matching and regression matching allow us to match imprecise but highly correlated value sets such as population values and import/export estimates.

Expected values. Using constant value recognizers in data frames, we have shown that finding and matching expected values in value sets provides significant leverage in schema matching [20]. Being able to recognize values such as latitudes, longitudes, distances, dates, times, and percentages helps us match object sets. Data frame recognizers also help distinguish labels from values in tables, decompose or compose value strings for matching, and determine whether value sets are unions or subsets of other value sets [20].

Constraints. In [4] we studied constraints in the context of schema matching. These include keys in tables (as well as nonkeys), functional relationships, one-to-one correspondences, subset/superset relationships, and optional and mandatory constraints involving unknown and null values. Others have derived constraints from typed hierarchies [41] and recurrent subpatterns [47].

Structure. We [50] and others [7,13,31,37] have developed matching algorithms based on structural context. We have been able to use proximity, node importance as measured by in/out-degree, and neighbor similarity to help match object sets.

As an illustration of mappings among mini-ontologies, we next describe candidate mappings between the mini-ontologies Figure 3.

For mini-ontologies 3(a) and (b), we discover label similarities between concepts in 3(a) and (b). Indeed, the labels *Country* and *Population* in mini-ontology 3(a) match exactly the same labels in mini-ontology 3(b). Further, examination of the data value characteristics associated with those concepts in the tables results in reinforcement of the label matches. For population values, Gaussian matching and regression matching apply nicely. In addition, we discover that the data frames that match values for concepts in mini-ontology 3(a) are the same data frames that match to values of corresponding object sets in mini-ontology 3(b), including the data-frame matches recognizing both *July 2001* and *July 2003* as dates. These mappings and matchings strongly suggest that the two ternary *Population* relationship sets 3(a) from [12] and 3(b) from [49] match. They also suggest an adjustment—replace the two dates with a *Date* object set and let the two dates be objects in the object set rather than individual objects connected to the relationship set.

It is common to find this kind of strong agreement between geopolitical information sources. This is of interest to us, because when this does happen, it is common for the information to be presented in different formats as is the case here (see Figures 1 and 2). The fact that someone apparently took the trouble to reorganize the information in [12] in a structure different from its source [49] is interesting. It supports the notion that although we use tables to build ontologies, humans who build tables indirectly collaborate in ways that TANGO ontologies approximate.

In looking at other mini-ontologies in Figure 3, we discover that the label *Country* also matches labels in mini-ontologies 3(c), (d), and (f). We can perform a direct evaluation of the match for the data associated with the label *Country* in the mini-ontology 3(c) because its *Country* object set is displayable, but for 3(d) and (f) we must do something different because their *Country* object sets are non-displayable. In both cases, the evaluation involves searching for associated object sets that contain names of the non-displayable object identifiers. In both 3(d) and (f) we find *Name* associated with a generalization. Further analysis reveals that many values in the *Name* object sets match names in the *Country* object set in 3(a). Thus we conclude that *Country* in 3(a) matches with the structural aggregation through a generalization/specialization of *Name* and *Country* in Figures 3(d) and (f).

The label *Population* in the mini-ontology in Figure 3(a) matches with *Population* in Figure 3(d). The date objects, *July 2001* and *January 15, 2001 | ?*, also match in the sense that the *Date* data frame recognizes them both. (The "?", explicitly denoting the possibility of a null, is not problematic because when we consider the concept to be a *Date*, we can simply make the connection optional.) In order to match the relationship sets in 3(a) and (d) in which *Population* appears, however, we have to recognize that *Country* in 3(d) is a specialization of *Geopolitical Entity* and that we can match the non-displayable *Country* with the displayable *Country* using *Name* associated with *Geopolitical Entity* as described above.

In examining potential concept mappings between mini-ontology 3(b) and other mini-ontologies, we encounter situations between 3(b) and (d) identical to those between 3(a) and (d). Thus, the resulting mappings are the same.

For mini-ontology 3(c), in addition to the previous matches discovered for the *Country* concept with mini-ontologies 3(a), (b), (d) and (f), there is one additional match of interest. The labels for *Geographic Coordinates*, *Longitude* and *Latitude* match with identical labels in mini-ontology 3(e). Applying our multifaceted approach we are able to confirm these matches.

Using similar analyses for mini-ontology 3(d), TANGO is able to recognize that not only do *Country* and *Population* match with concepts in other mini-ontologies as described above, but that *Continent*, *Country*, *City*, *Name*, and *Geopolitical Entity* also have potential matching concepts in other mini-ontologies. The labels of the non-displayable concepts *Continent* and *Country* match identically with the labels of non-displayable concepts in mini-ontology 3(f). The label *City* matches partially with the label *City|Town* in mini-ontology 3(e), both of which are also non-displayable concepts. The label for the displayable concept *Name* matches in Figures 3(e) and (f). Close examination, however, reveals that the data values for the data associated with the concept *Name* in these three mini-ontologies do not have a strong correlation. Nevertheless, we also note that *Name* is associated with an object set which is the parent concept of *Continent*, *Country*, and *City* and that, limited to these associations, the data has a high correlation, especially for *Continent* and *Country* between 3(d) and (f). This allows us to conclude that *Continent* in 3(d) and (f) match, that *Country* in 3(d) and (f) match, and that *City* in 3(d) and *City|Town* are a likely match. Since we have both *Continent* and *Country* matches between 3(d) and (f), which cover a large majority of the possible matches between *Geopolitical Entity* in 3(d) and *Where Spoken (Major)* in 3(f), TANGO also concludes that these two generalizations are a likely match.

Having tried all the combinations but one, TANGO attempts to discover additional mappings between mini-ontology 3(e) and (f). But it finds none.

6. Ontology merge

Once TANGO has discovered mappings between mini-ontologies or between a mini-ontology and the ontology we are building, it can begin the merging process. Sometimes the match is such that we can directly fuse two ontologies by simply merging directly corresponding nodes and edges of both. Often, however, merging induces conflicts that must be resolved.

We use three basic approaches to conflict resolution: (1) automatic adjustment based on constraint satisfaction, (2) synergistic adjustment based on IDS statements [4], and (3)

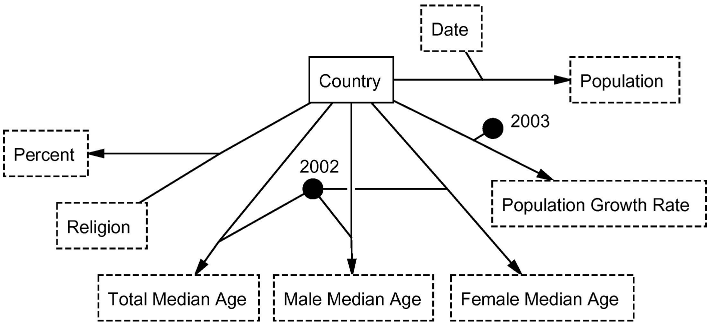


Figure 4. Growing ontology after merging the mini-ontologies in Figures 3(a) and (b).

multiple adjustments leading to multiple ontological views with mappings between them. All three of these approaches rely on being able to determine plausible merges. Then, for automatic adjustments, we can take the best among the plausible merges; for synergistic adjustments, we can raise the important issues and make suggestions, letting an ontologist make the final decisions. For multiple adjustments, can keep all plausible merges and later eliminate those discarded in synergistic evaluations and those that no longer stand up to new evidence gathered as the process continues.

To determine plausible merges based on discovered mappings, we consider constraint violations and congruency principles. Constraint violations include functional/non-functional mismatches, optional/mandatory mismatches, displayable/non-displayable mismatches, and subset/superset constraint violations. Congruency principles [10,24] attempt to ensure that all objects in an object set have the same properties; the objects in an object set are *congruent* when this principle holds and are otherwise *incongruent*. Other similar principles of formal ontology construction also apply [24], as well as related work on merging ontologies (e.g. [36]) and comparing and aligning ontologies (e.g. [5]). We illustrate this merging process by merging the mini-ontologies in Figure 3.

We look initially for mini-ontologies that exhibit the largest possible overlap (as measured by the number of inter-ontology mappings) with respect to the size of the mini-ontologies. Thereafter we select mini-ontologies that overlap the most with our growing ontology. In our example, the overlap is much the same for all mini-ontologies that do overlap. Thus, we just begin by merging the first two mini-ontologies 3(a) and (b).

1st Merge: *Country* matches *Country* and *Population* matches *Population*. Both *July 2001* and *July 2003* are date components associated with *Population*, and we merge them as a *Date* object set. Figure 4 shows the resulting initial ontology.

2nd Merge: Building on the 1st Merge, we add the mini-ontology 3(d) and obtain the emerging ontology in Figure 5. Here, we must reconcile the displayable/non-displayable *Country* object sets, but this is straightforward based on the mappings we have already discovered. Thus, we let *Name* be an inherited property for all continents, countries, and

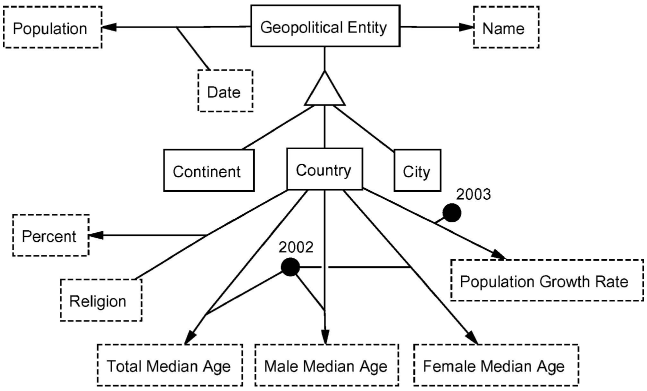


Figure 5. Growing ontology after merging the mini-ontologies in Figures 3(a), (b), and (d).

cities as Figure 5 shows. According to congruency principles, we also let *Population* be an inherited property and thus omit it from the *Country* specialization. Congruency holds in Figure 5 because the non-displayable concept *Geopolitical Entity* is a generalization of the non-displayable concepts *Continent*, *Country*, and *City*, all of which—according to mini-ontology 3(d)—mandatorily have *Population* and *Country* in mini-ontologies 3(a) and (b). Notice that in the merged ontology in Figure 5 the concept *Country* is now non-displayable; it inherits the *Name* property, which contains the names that initially were in the *Country* object sets in mini-ontologies 3(a) and (b).

3rd Merge: Continuing, we merge the mini-ontology in Figure 3(f) with the growing ontology in Figure 5 and obtain the ontology in Figure 6. Here, the mappings TANGO has already generated indicate that the objects in the object sets *Geopolitical Entity* and *Where Spoken (Major)* largely overlap and that both the *Continent* and *Country* object sets match. When merging a mini-ontology into a growing ontology, TANGO uses, as its default, the name it already has in the growing ontology (a user, of course, may change the name). Thus, the generalization in the merged ontology in Figure 6 becomes *Geopolitical Entity*. After the merge, there is insufficient evidence to maintain the mandatory participation constraints for *Population* and *Language*, and TANGO thus changes them to be optional participation constraints. There is sufficient evidence, however, to maintain the mandatory participation constraint for *Name*.

4th Merge: We next merge mini-ontology 3(c), obtaining the ontology in Figure 7. This merge is straightforward based on the already discovered mappings. The displayable *Country* object set in 3(c) becomes non-displayable, and its values become part of the *Name* object set inherited from *Geopolitical Entity*. The relationship sets attached to the displayable *Country* object set in 3(c) are instead attached to the non-displayable *Country* object set.

5th Merge: Finally, we add mini-ontology 3(e). We have already found mappings between the identically named object sets *Geographic Coordinates*, *Longitude*, and *Latitude*.

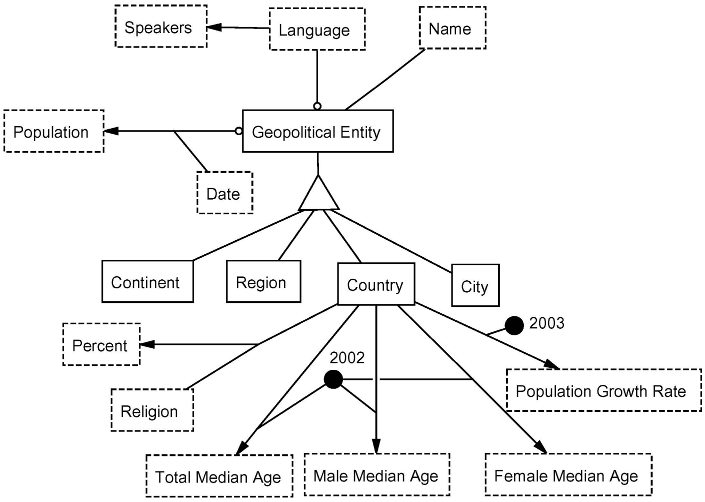


Figure 6. Growing ontology after merging the mini-ontologies in Figures 3(a), (b), (d) and (f).

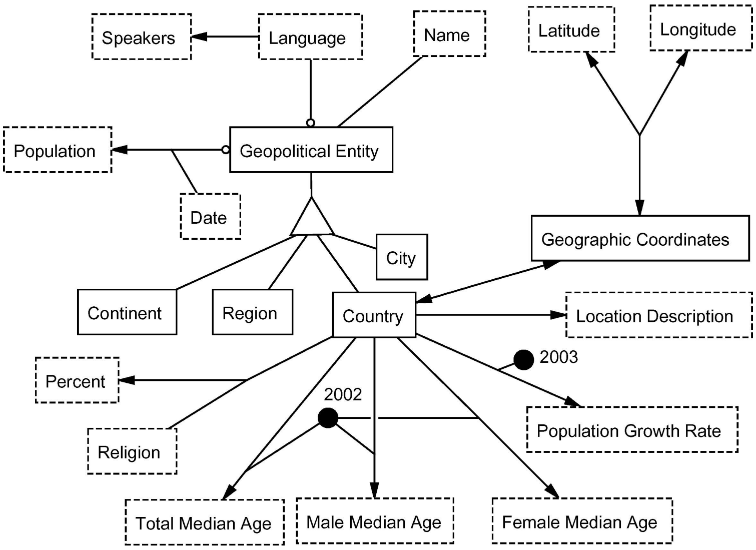


Figure 7. Growing Ontology after merging the mini-ontologies in Figures 3(a), (b), (c), (d) and (f).

This part of the merge is straightforward. The remaining part of the merge is more difficult, not so much because it is structurally complex (TANGO can handle that part), but because the evidence based on partially overlapping cities and the connection to the geographic coordinates is not likely to be strong enough for TANGO to decide on its own what the relationship between *Place* and *Geopolitical Entity* should be. It does, however, have enough evidence to be able to pose an intelligent IDS statement to a user. Observe that *Geographic Coordinates* is a property of the concept *Place* in mini-ontology 3(e), while in the growing ontology it is a property of the concept *Geopolitical Entity*. This leads TANGO to consider that perhaps *Place* and *Geopolitical Entity* are the same concept. With further exploration, TANGO can discover that although *Place* has a *Name* just like *Geopolitical Entity* does, the more specialized concepts *Elevation*, *USGS Quad*, *Lake*, *Reservoir* and *Mine* are not the kind of concepts found as specializations of *Geopolitical Entity*. The concept *City*|*Town*, however, does resemble the concept *City* in the growing ontology. Thus, the two generalization/specializations can potentially be merged. So TANGO can pose this possibility to a user. We assume in the figure that the user replies that the generalization/specializations can be merged with *Place* as a generalization of *Geopolitical Entity* and *City*|*Town* as a specialization of *Geopolitical Entity*. As for sorting out where the relationship sets should be attached, TANGO's default is to select the highest point in the generalization/specialization hierarchy. It therefore associates *Geographic Coordinates* with *Place* but makes the association optional because it now has evidence that not every place has geographic coordinates recorded for it (in particular, continents, regions, and some cities do not have geographic coordinates in our particular version of the growing ontology). The final result is the ontology in Figure 8.

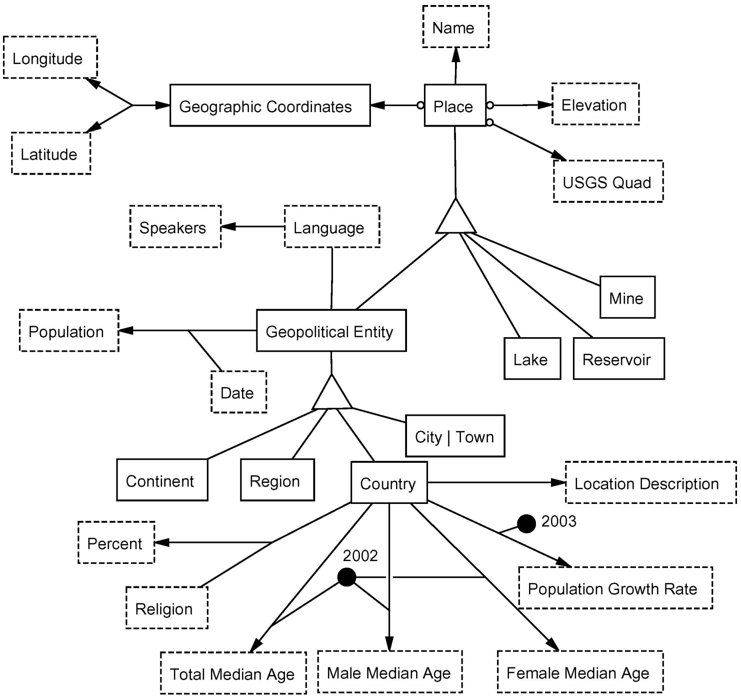


Figure 8. Growing ontology after merging all mini-ontologies.

7. Applications

Semantics is a grand challenge for the current generation of computer technology, particularly as it relates to the Semantic Web. It is the key for unlocking the door, for example, to personal agents that can roam the Semantic Web and carry out sophisticated tasks for their masters, to information exchange and negotiation in e-business, and to automated, large-scale, in-silico experiments in e-science. We do not claim that TANGO will resolve this challenge, but we do claim that it addresses related issues and that its successful realization would help us move a step closer to a resolution. As specific research in this direction, we offer the following observations about Semantic Web construction.

As the Semantic Web becomes more popular, a question of increasing importance will be how to convert some of the interesting unstructured and semi-structured, data-rich documents on the web as they now stand into Semantic Web documents. Others have recognized the importance of this conversion and are working on research and commercial efforts to address this particular issue [1,39,43]. Lixto [1] is a commercial effort in the EU to develop commercial-grade tools for the construction and use of the Semantic Web. MoA [39] is an effort sponsored by the Korean Ministry of Information to allow mapping and merging of distributed OWL ontologies and content. SEWASIE [43] is an integrated Semantic Web environment that allows advanced search capabilities for small and medium enterprises in the EU.

In [8] we proposed a way to bridge the gap between the current web and the Semantic Web by semiautomatically converting Resource Description Framework Schemas (RDFS's) and DAML+OIL ontologies into data extraction ontologies [16]. The prototype system we built does this conversion. It extracts data and then converts it to RDFS, making it accessible to Semantic Web agents. In addition, the prototype system superimposes the metadata of the extracted information over the document for direct access to data in context, as suggested in [33]. We believe that TANGO-constructed ontologies will work even better for this application.

As part of making TANGO-generated ontologies compliant with the Semantic Web, we need to be able to convert an OSM ontology into public ontologies such as RDFS, DAML+OIL and OWL. As an example, Figure 9 shows a partial listing of an OWL ontology for the mini-ontology in Figure 3(a). It is not hard to convert an OSM ontology into an OWL ontology. Each object set in the OSM ontology in Figure 3(a) maps to an *owl:Class* object in the OWL ontology in Figure 9. Each binary relationship set in the OSM ontology maps to an *owl:ObjectProperty* with a domain and range. We cannot, however, directly transform relationship sets with higher arities, such as the ternary relationship between *Country*, *Religion*, and *Percent* in Figure 3(a). To overcome this limitation without loss of generality, we create an artificial object set to represent the ternary relationship set and then decompose the ternary relationship set into three binary relationship sets. Figure 9 shows the necessary artificial new class *CRP*. Then, we create a binary relation specification between *CRP* and each of the three OWL classes, *Country*, *Religion*, and *Percent* in Figure 9. Figure 9 shows the relation specification between *CRP* and *Percent* in the new binary relationship set called *atPercent*. Inside the *ObjectProperty* of *atPercent*, the *rdf:type* indicates that this is a functional property, and the *rdfs:domain* and *rdfs:range*

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY dlbeck "http://www.deg.byu.edu/ontologies/dlbeck#" >
    ... ]>
<rdf:RDF xmlns = "http://www.deg.byu.edu/ontologies/dlbeck#"
    ... >
<owl:Class rdf:ID="Country">
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#hasReligionAtPercent" />
          <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction> </rdfs:subClassOf>
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#hasReligionAtPercent" />
          <owl:allValuesFrom rdf:resource="#CRP" />
    </owl:Restriction> </rdfs:subClassOf>
    ...
</owl:Class>
<owl:Class rdf:ID="Religion">
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#CountryAtPercent" />
          <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction> </rdfs:subClassOf>
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#CountryAtPercent" />
          <owl:allValuesFrom rdf:resource="#CRP" />
    </owl:Restriction> </rdfs:subClassOf>
    ...
</owl:Class>
<owl:Class rdf:ID="Percent">
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#CountryHasReligion" />
          <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction> </rdfs:subClassOf>
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#CountryHasReligion" />
          <owl:allValuesFrom rdf:resource="#CRP" />
    </owl:Restriction> </rdfs:subClassOf>
    ...
</owl:Class>
<owl:Class rdf:ID="CRP">
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#atPercent" />
          <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction> </rdfs:subClassOf>
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#atPercent" />
          <owl:allValuesFrom rdf:resource="#Percent" />
    </owl:Restriction> </rdfs:subClassOf>
    <rdfs:subClassOf> <owl:Restriction>
          <owl:onProperty rdf:resource="#hasReligion" />
          <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction> </rdfs:subClassOf>
    ...
</owl:Class>
<owl:ObjectProperty rdf:ID="atPercent">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
    <rdfs:domain rdf:resource="#CRP" />
    <rdfs:range rdf:resource="#Percent" />
</owl:DatatypeProperty> <owl:ObjectProperty rdf:ID="CountryHasReligion">
    <owl:inverseOf rdf:resource="#atPercent" />
</owl:ObjectProperty>
...
</rdf:RDF>
```

Figure 9. Partial OWL listing for the mini-ontology in Figure 3(a).

indicate the direction of the functional property. The *owl:inverseOf* property shows that the *ObjectProperty* for *CountryHasReligion* is an inverse of the *ObjectProperty atPercent*. As for constraints Figure 9 indicates that the OWL ontology can directly represent OSM's min-max participation constraints using the tags *owl:minCardinality*, *owl:maxCardinality*, or *owl:cardinality*. Thus, for example, the *owl:cardinality* in the class *CRP* for the relation association in the *atPercent* property is exactly 1; whereas the relation associations in the *hasReligion* property and the *Country* property (not shown in Figure 9) have a minimum cardinality of 1 and no maximum cardinality. Although not shown in this example, it is also straightforward to transform OSM's generalization/specialization to an OWL ontology. To accomplish this, we need to map the object sets onto OWL classes and then specialize using the *rdfs:subClassOf* property to link the parent and child object sets.

Given that we can convert TANGO-generated ontologies to Semantic Web ontologies, we are now able to annotate web pages associated with those ontologies with ontologies that software agents can use to "understand" the tables in those web pages. Thus, we are able to realize, at least partially, the goal of semi-automatically converting HTML pages into Semantic Web pages.

8. Concluding remarks

We have presented our vision for TANGO—a way to generate ontologies from tables. Our generation procedure has four steps.

1. Recognize and canonicalize table information. Based on the notion of table-equivalent data, we use heuristics and resources such as data frames and WordNet to convert semi-structured data to canonicalized table information.
2. Construct mini-ontologies from canonicalized tables. Each table represents a small part of a larger ontology. Given a canonicalized table, we exploit the data and relationships in the table to construct a conceptual model. We represent conceptual model instances in OSM, which gives us a convenient and powerful way to represent ontological concepts (as object sets) and ontological associations (as relationship sets) and a way to represent ontological constraints (functional dependencies, cardinality relationships, optional/mandatory requirements, and generalization/specialization).
3. Discover inter-ontology mappings. Based on previous work on schema mapping (both our own and the work of others), we discover semantic mappings among mini-ontologies and also between mini-ontologies and larger application ontologies. The approach is multifaceted and thus depends on exploiting multiple auxiliary resources and multiple self-contained clues about the data and metadata in a populated ontology.
4. Merge mini-ontologies into a growing application ontology. We automatically find plausible ontology merges. When conflicts arise, we use alternative approaches to resolve the conflicts: adjust based on constraint satisfaction, synergistically use interactive IDS (Issue/Default/Suggestion) statements, and support multiple versions and allow delayed resolution.

As further motivation for TANGO, we have discussed its application to the Semantic Web. We showed how to convert OSM ontologies into Semantic Web ontologies. This, together with table understanding, provides an immediate way to generate annotated pages that Semantic Web agents can understand and use.

Acknowledgments

This work was supported in part by NSF grant IIS-0083127

Notes

1. Tables have a particular spatial layout of material [47] that carries significant meaning. Lamke [28] describes tables as "organizational resources to enable meaningful relations to be recovered from bare thematic items in the absence of grammatical constructions," and argues that there is always "an implied grammar" and a recoverable textual sentence or paragraph for every table."
2. Our ontologies fit the standard definition for an ontology as Tom Gruber describes them in [23] "An ontology is a formal, explicit specification of a shared conceptualization." Our ontologies are however small and thus the name mini-ontology.
3. In this and other tables, "missing" values are null values, which we assume are elements of every domain.
4. Although we have spent much effort in understanding tables, our approach described in [20] was mainly to understand tables based on an intermediate schema. In this work, we go one step beyond by transforming all the tables into a common canonical form. We have also had some experience working with this more general table transformation problem—indeed, we have been invited to write a survey of table-processing work [17]. The current state of research, however, still does not offer a general solution to the problem we seek to solve.
5. These canonicalized tables are subparts of actual tables found on the web—subparts in the same sense that the table-equivalent data in Table 2 is a subpart of the table in Figure 2 (i.e. we have omitted some of the information). A reference for each original table from which we drew the information appears in the bibliography. We chose the subset presented here for the purpose of illustration.
6. For a complete description of OSM formal semantics, the reader should consult the appendix on [19].
7. Object sets are in essence what others refer to as concepts in the ontology literature, thus they are used interchangeably in this paper and have the same meaning.
8. The particular notation we use to represent ontologies is not significant, but the concepts it represents are significant. We choose it because: (1) it is fully formal in terms of first-order predicate calculus [19], (2) it covers the typical ontological properties of interest—ISA hierarchies, part/whole hierarchies, relationships, and concepts including lexical appearance, representation, and computational manipulation, and (3) it has specialized tools for ontology creation and manipulation, ontological table understanding [20], ontological data extraction, and ontological data integration [50].
9. Surprisingly, neither direct word match nor synonym match mattered in our machine-learned decision-tree rule for matching labels. Instead, the number of common hypernym roots and the distances to common hypernyms dominated the rule. Of course, identical words and synonyms have common hypernym roots at a minimal distance from the words, which mitigates our surprise.

References

[1] R. Baumgartner, S. Flesca, and G. Gottlob, "Visual web information extraction with Lixto," in *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01).* Rome, Italy, 2001, pp. 119–128.

[2] S. Bergamaschi, S. Castano, and M. Vincini, "Semantic integration of semistructured and structured data sources," *SIGMOD Record* 28(1), 1999, 54–59.

[3] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Scientific American* 36(25) 2001.

[4] J. Biskup and D. Embley, "Extracting information from heterogeneous information sources using ontologically specified target views," *Information Systems* 28(3), 2003, 169–212.

[5] A. Burgun and O. Bodenreider, "Comparing terms, concepts, and semantic classes in WordNet and the Unified Medical Language System," in *WordNet and Other Lexical Resources: Applications, Extensions, and Customizations; An NAACL-01 (North American Association for Computational Linguistics) Workshop*. Pittsburgh, Pennsylvania, 2001, pp. 77–82.

[6] A. Cali, D. Calvanese, G. D. Giacomo, and M. Lenzerini, "On the expressive power of data integration systems," in *Proceedings of 21st International Conference on Conceptual Modeling (ER2002)*. Tampere, Finland, 2002, pp. 338–350.

[7] S. Castano, V. D. Antonellis, M. Fugini, and B. Pernici, "Conceptual Schema Analysis: Techniques and Applications," *ACM Transactions on Database Systems* 23(3), 1998, 286–333.

[8] T. Chartrand, "Ontology-based extraction of RDF data from the world wide web". Master's thesis, Brigham Young University, Provo, Utah 2003.

[9] R. Chiang, T. Barron, and V. Storey, "Reverse engineering of relational databases: Extraction of an eer model from a relational database," *Data & Knowledge Engineering* 12(2), 1994, 107–142.

[10] S. Clyde, D. Embley, and S. Woodfield, "Improving the quality of systems and domain analysis through object class congruency," in *Proceedings of the International IEEE Symposium on Engineering of Computer Based Systems (ECBS'96)*, Friedrichshafen, Germany, 1996, pp. 44–51.

[11] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards automatic data extraction from large web sites," in *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*. Rome, Italy, 2001, pp. 109–118.

[12] dlbeck.com, 2003, "dlbeck.com," www.dlbeck.com/population.htm.

[13] A. Doan, P. Domingos, and A. Halevy, "Reconciling schemas of disparate data sources: A machine-learning approach," in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data (SIGMOD 2001)*. Santa Barbara, California, 2001, pp. 509–520.

[14] D. Embley, "Programming with data frames for everyday data items," in *Proceedings of the 1980 National Computer Conference*. Anaheim, California, 1980, pp. 301–305.

[15] D. Embley, *Object Database Development: Concepts and Principles*, Addison-Wesley: Reading, Massachusetts, 1998.

[16] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y.-K. Ng, and R. Smith, "Conceptual-model-based data extraction from multiple-record web pages," *Data & Knowledge Engineering* 31(3), 1999, 227–251.

[17] D. Embley, D. Hurst, D. Lopresti, and G. Nagy,"Table processing Paradigms: A research survey," *International Journal on Document Analysis and Recognition*, 2004a. (Submitted).

[18] D. Embley, D. Jackman, and L. Xu, "Multifaceted exploitation of metadata for attribute match discovery in information integration," in *Proceedings of the International Workshop on Information Integration on the Web (WIIW'01)*. Rio de Janeiro, Brazil, 2001, pp. 110–117.

[19] D. Embley, B. Kurtz, and S. Woodfield, *Object-Oriented Systems Analysis: A Model-Driven Approach*, Prentice Hall: Englewood Cliffs, New Jersey, 1992.

[20] D. Embley, C. Tao, and S. Liddle, "Automating the extraction of data from tables with unknown structure," *Data & Knowledge Engineering*. (to appear) currently at http://www.deg.byu.edu/papers/dke2003etl.pdf, 2004b.

[21] D. Embley and M. Xu, "Relational database reverse engineering: A model-centric, transformational, interactive approach formalized in model theory," in *DEXA'97 Workshop Proceedings*, Toulouse, France, 1997, pp. 372–377.

[22] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press: Cambridge, Massachussets, 1998.

[23] T. R. Gruber, "Towards principles for the design of ontologies used for knowledge sharing," in N. Guarino and R. Poli (eds.), *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Deventer, The Netherlands, 1993.

[24] N. Guarino, "Formal ontologies and information systems," in N. Guarino (ed.), *Proceedings of the First International Conference on Formal Ontology in Information Systems (FOIS98)*. Trento, Italy, 1998, pp. 3–15.

[25] J.-L. Hainaut, "Database reverse engineering: Models, techniques and strategies," *Proc. of the 10th International Conference on Entity-Relationship Approach (ER'91)*. San Mateo, California, USA, 1991, pp. 643–670.

[26] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: The state of the art," *The Knowledge Engineering Review* 18(1), 2003, 1–31.

[27] M. Kantola, H. Mannila, K.-J. Räihä, and H. Siirtola, "Discovering functional and inclusion dependencies in relational databases," International Journal of Intelligent Systems 7, 1992, 591–607.

[28] J. Lemke, "Multiplying meaning: Visual and verbal semiotics in scientific text," in J. Martin and R. Veel (eds.), *Reading Science: Critical and Functional Perspectives on Discourses of Science*. Routledge, 1998, pp. 87–113.

[29] W.-S. Li and C. Clifton, "Semantic integration in heterogeneous databases using neural networks". in *Proceedings of the 20th Very Large Data Base Conference*. Santiago, Chile, 1994.

[30] D. Lopresti and G. Nagy, "A tabular survey of table processing," in A. Chhabra and D. Dori (eds.), *Graphics Recognition—Recent Advances*, Lecture Notes in Computer Science, LNCS 1941. Springer Verlag, 2000, pp. 93–120.

[31] J. Madhavan, P. Bernstein, and E. Rahm, "Generic schema matching with cupid," in *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB'01)*. Rome, Italy, 2001, pp. 49–58.

[32] D. Maier, *The Theory of Relational Databases*, Computer Science Press, Inc: Rockville, Maryland, 1983.

[33] D. Maier and L. Delcambre, "Superimposed information for the internet," in S. Cluet and T. Milo (eds.), *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*. Philadelphia, Pennsylvania, 1999.

[34] F. D. Marchi, S. Lopes, J.-M. Petit, and F. Toumani, "Analysis of existing databases and the logical level: The DBA companion project," *SIGMOD Record* 32(1), 2003, 47–52.

[35] V. Markowitz and J. A. Makowsky, "Identifying extended entity-relationship object structures in relational schemas," *IEEE Transactions on Software Engineering* 16(8), 1990, 777–790.

[36] D. McGuinness, R. Fikes, J. Rice, and S. Wilde, "An environment for merging and testing large ontologies," in *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning*. Breckenridge, Colorado, 2000, pp. 483–493.

[37] T. Milo and S. Zohar, "Using schema matching to simplify heterogeneous data translation," in *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB-98)*, 1998, pp. 122–133.

[38] R. Mizoguchi and M. Ikeda, "Towards ontology engineering," in *proceedings of the Joint 1997 Pacific Asian Conference on Expert Systems / Singapore International Conference on Intelligent Systems*. Singapore, 1997, pp. 259–266.

[39] MoA, 2004, "MoA—An OWL ontology merging and alignment tool," http://mknows.etri.re.kr/moa/index.html.

[40] MostSpokenLanguages, "The 30 most spoken languages of the world," www.krysstal.com/spoken.html, 2003.

[41] S. Nestorov, S. Abiteboul, and R. Motwani, "Extracting schema from semistructured data," in *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data (SIGMOD'98)*, Seattle, Washington, 1998, pp. 295–306.

[42] E. Rahm and P. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal* 10, 2001, 334–350.

[43] M. Schoop, A. Becks, C. Quix, T. Burwick, C. Engels, and M. Jarke, "Enhancing decision and negotiation support in enterprise networks through semantic web technologies," in *XML Technologien fur das Semantic Web—XSW 2002, Proceedings zum Workshop*, 2002, pp. 161–167.

[44] P. Spyns, R. Meersman, and M. Jarrar, "Data modeling versus ontology engineering," *SIGMOD Record* 31(4), 2002, 12–17.

[45] Y. Tijerino, D. Embley, D. Lonsdale, and G. Nagy, "Ontology generation from tables," in *Proceedings of the 4th International Conference on Web Information Systems Engineering*. Rome, Italy, 2003, 242–249.

[46] TopoZone2002: 2002, ‘TopoZone,’ www.topozone.com.

[47] K. Wang and H. Liu, “Schema discovery for semistructured data,” in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*. Newport Beach, California, 1997, pp. 271–274.

[48] WorldAtlas2003, ‘WorldAtlas.Com,’ 2003, www.worldatlas.com/geoquiz/thelist.htm.

[49] WorldFactbook2003, “The World Factbook—2003”, 2003. www.cia.gov/cia/publications/factbook.

[50] L. Xu and D. Embley, “Using domain ontologies to discover direct and indirect matches for schema elements,” in *Proceedings of the Workshop on Semantic Integration (WSI’03)*. Sanibel Island, Florida, 2003, pp. 105–110.