Procs. WISE03, pp., 242-249, Rome, Italy, December 2003

Ontology Generation from Tables[1]

Yuri A. Tijerino
Brigham Young University
yuri@cs.byu.edu

David W. Embley
Brigham Young University
embley@cs.byu.edu

Deryle W. Lonsdale
Brigham Young University
lonz@byu.edu

George Nagy
Rensselaer Polytechnic Institute
nagy@ecse.rpi.edu

Abstract

We often need to access and reorganize information available in multiple tables in diverse Web pages. To understand tables, we rely on acquired expertise, background information, and practice. Current computerized tools seldom consider the structure and content in the context of other tables with related information. This paper will address the table processing issue by developing a new framework to table understanding that applies an ontology-based conceptual modeling extraction approach to: (i) understand a table's structure and conceptual content to the extent possible; (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 for use in future situations. The result will be a formalized method of processing the format and content of tables while incrementally building a relevant reusable conceptual ontology.

1. Introduction

Motivated by our belief that inference about unknown objects and relations in a known context can be automated, we are developing an information-gathering engine to assimilate and organize knowledge. While understanding context in a natural-language setting is difficult, structured information such as tables and filled-in forms make it easier to interpret new items and relations. We organize the new knowledge we gain from "reading" tables as an ontology [1] and thus we call our information-gathering engine *TANGO* (Table ANalysis for Generating Ontologies). TANGO thus exploits tables and filled-in forms to generate a domain-specific ontology with minimal human intervention.

This paper describes our approach to develop TANGO and our current progress. The paper is organized as follows: Section 2 describes the general background to TANGO. Section 3 introduces TANGO's general framework for ontology generation. Section 4 discusses the steps to build a kernel ontology in the geopolitical domain. Section 5 illustrates how the kernel ontology is expanded through two examples. Section 6 introduces some applications for TANGO ontologies. Finally, section 7 presents some conclusions.

2. TANGO background and objectives

Our work can be considered as semi-automated, applied "ontological engineering," [2] which has as its goal "effective support of ontology development and use throughout its life cycle—design, evaluation, integration, sharing, and reuse" [3]. It builds upon previous work on of human-machine collaboration in building knowledge systems [4]. As an analogy for what we are to accomplish with TANGO, consider that instead of humans collaborating to design an ontology [5], TANGO provides an approach in which *tables* "collaborate" to design an ontology. In a sense, this is the same because information is assembled from specific instances of tables created by humans.

The information-gathering engine behind TANGO expands from an embryonic kernel rather than growing from scratch. Relevant web pages or tables are interpreted with the help of current application and tool ontologies (i.e. ontologies for tables, and ontological knowledge about a domain and about semantic integration within a domain). From each experience, new facts, relations, and interpretive techniques will be used to expand, correct, and consolidate the growing application ontology. Where necessary, human help may be invoked: one of our major goals is to find out how little human interaction is sufficient.

[1] This work is supported by the National Science Foundation under grant No. IIS-0083127.

TANGO will demonstrate the feasibility of automated knowledge gathering in the domain of geopolitical facts and relations, where relevant empirical data is widely scattered but often presented in the form of lists, tables, and forms. The geo-political application ontology will be constructed using tool ontologies that encapsulate a growing understanding of coordinate systems, geo-political subdivisions, and conventions for reading tables. The chosen domain of geography spans many important human activities: natural resources, travel, culture, commerce, and industry.

As a basis model for ontology construction, TANGO uses a formally defined conceptual modeling language that has a direct translation into predicate calculus [6]. This provides a theoretical foundation for formal property analysis. Another key element of TANGO's approach to ontology building is searching for direct and indirect schema-element matches [7] between populated database schemas (i.e. between a new document and ontologically organized, previously seen documents). TANGO also depends on (1) subject-specific lexicons and thesauri, (2) specialized data frames [8, 9] for commonly occurring fields (like latitude-longitude pairs or dates), (3) object-class congruency principles [10], (4) formally consistent tools for manipulating meta-data [11], (6) analysis tools and techniques [12, 13, 14], and (7) ontology-maintenance tools developed by others, e.g. [15].

Our earlier experiments with ontology-based information extraction have been successful on relatively narrow domains: census records [14], automobile want ads and obituaries [9], and several other domains [16]. Earlier work on related issues, including isolated tables [12], topographic maps [17], satellite images [18, 19], and geographic data processing [21, 22], has also been successful. We believe that we are now ready to integrate what we have learned into TANGO to tackle the much larger, but still bounded, domain of geographic information.

TANGO will identify and quantify what can be accomplished by combining the best available ideas and tools (1) in the geo-knowledge domain of high global interest, (2) in the growing field of ontology analysis and development, and (3) in a sphere of knowledge engineering where further invention is necessary.

3. Ontology generation

Ontology generation in TANGO makes use of auxiliary knowledge sources, including an ontology-based system for (1) table understanding, (2) data extraction, and (3) data integration. Based on completed research, we offer the following specifics.

- Our ontology-based table-understanding system allows us to take a table as input and produce attribute-value pairs as output [12, 14, 22, 23].
- Our ontology-based data-extraction system allows us to take semi-structured text as input, including in particular attribute-value pairs extracted from tables [23], and produce as output a database corresponding to a given application ontology and populate it with the given semi-structured data. (We have developed resilient web wrapper-generation systems that do not break when pages change or new pages come on-line because the basis for the extraction is an ontology rather than a page grammar and its variations [9, 16, 24, 23].)
- Our ontology-based integration system produces schema-element matches between populated database schemas: direct matches when schema elements in two schemas have the same meaning, and many indirect matches when schema elements have overlapping meanings or have different encodings [7, 25, 26]. The key ideas for matching, which we explore in this integration work, are (1) value characteristics, (2) expected values based on our data-extraction techniques, (3) attribute names and their synonyms, and (4) the structure of a schema.

Our ontology-generation procedure has three steps, the first of which we do only once for any given domain:

1. An ontology engineer builds a kernel application ontology, which should be small (having only a few concepts), central (containing the most important concepts for the application), and example-rich (containing typical sample data, descriptions of common data values such as dates and times, and typical operations over this data).
2. For any given table, the system creates a mini-ontology based on its understanding of the table. This yields a schema of object and relationship sets, values for the object sets as attribute-value pairs, and tuples for the relationship sets each representing a relationship among attribute-value pairs.
3. The system attempts to integrate each new mini-ontology with the ontology it is building. Integration may raise several issues: (a) there may be alternative ways it can integrate the mini-ontology into the evolving global ontology, (b) constraints may be inconsistent, (c) adjustments to the evolving ontology may be necessary, and (d) it may need human intervention. To resolve these issues, the system can use congruency principles [10] and principles of ontology construction [1, 4, 27, 28, 29, 30]; and when we need human intervention we can use Issue/Default/Suggestion (IDS) statements as in [25] as well as tools for cleaning ontologies, e.g. [29, 31].

4. Building the kernel application ontology

Figure 1 shows a graphical representation of a proposed kernel application ontology for geopolitical entities. We briefly explain the notation and the knowledge associated with the notation. In the notation a box represents an object set—dashed if printable (e.g. Longitude in Figure 1) and not dashed if not printable (e.g. Geopolitical Entity).

Lines connecting object sets are relationship sets; these lines may be hyper-lines (hyper-edges in hyper-graphs) when they have more than two connections to object sets (e.g. the relationship set named Latitude and Longitude designate Location). Names of binary relationship sets have a labeled reading-direction arrow, which along with the names of connected object sets form its name (e.g. Location has GMT Time). Optional or mandatory participation constraints 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 ontology in Figure 1 declares that, geopolitical entities must have specified names, but need not have specified locations. Arrowheads on lines specify functional constraints—for n-ary relationship sets, n > 2, acute versus obtuse angles disambiguate situations where tuples of two or more tails or heads form the domain or co-domain in the function. Open triangles denote generalization/specialization hierarchies (ISA hierarchies, subset constraints, or inclusion dependencies), so that both Country and City, for example, are subsets of Geopolitical Entity. We can constrain ISA hierarchies by partition (⊎), union (∪), or mutual exclusion (+) among specializations or by intersection (∩) among generalizations. Thus, for example, the ontology in Figure 1 declares that countries and cities are all the (currently) known geopolitical entities, and that countries and cities are mutually exclusive (there might be some exceptions such as the city-state Singapore in which case one object represents the city and another object represents the country, both sharing the same name). Filled in triangles denote part/whole, part-of, or aggregation hierarchies (e.g., a city is part of a country).

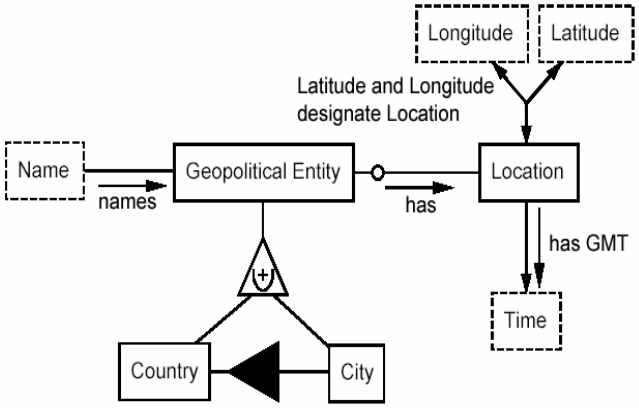


Figure 1. Initial geopolitical ontology

Each object set in an application ontology has an associated a data frame.[2] We provide seed values for our initial, kernel application ontology. For example, we initialize a lexicon with a few entries for Country such as United States, Germany, Hungary, Japan, Brazil, and another lexicon with a few entries for City such as New York, Philadelphia, Los Angeles, Chicago, Salt Lake City, Berlin, Frankfurt, Budapest, Tokyo, Yokohama, Sao Paulo. We also provide regular expressions for infinite value sets. For Time, for example, we let ([*1 – 9*]|*10* |*11* |*12*) : [*0 – 5*] \ *d*(\s * (*a*|*p*) \ .? \ *s* * *m* \ .?)?, which denotes strings such as *2:00 pm* and *11:49 a.m.*, be part of the recognizer, which uses Pearl-like syntax. Finally, we add appropriate procedural knowledge that may be useful. Examples include distances between locations based on latitude and longitude, the duration between two times, or the number of time zones between two geopolitical entities.

5. Creating the mini-ontologies and integrating them with the kernel ontology

Having exemplified Step 1, production of a kernel ontology, we now give two examples to illustrate Steps 2 and 3. Besides illustrating these steps, we also illustrate the types of input tables we intend to consider in our research. Note (1) that the examples range from full tables directly available on the web to partial tables hidden behind forms on the web, (2) that they range from electronic tables to scanned table images, and (3) that their diversity ranges from simple tables to semi-structured tables with auxiliary information.

- www.gazetteer.de/home.htm on 17 September 2002 (Figure 2). Given this table, we create the mini-ontology in Figure 3(a) and then integrate this ontology into the ontology we are constructing (initially the ontology in Figure 1). The result is the ontology in Figure 3(b). This is the heart of our research, and there are a host of problems to resolve. Briefly, we reach the ontology in Figure 3(b) by reasoning as follows.

[2] Using regular expressions and lexicons, a data frame for a concept *C* recognizes self-describing constant values of *C* and keywords that signal the presence *C* objects or *C* values. Data frames also include transformations between internal and external representations and computational knowledge as multi-sorted algebras over the concepts within the knowledge domain. See [8].

Understand Table: Table "understanding" means to associate the attributes with values and obtain atomic attribute-value pairs. This is straightforward for the table in Figure 2.

The World Gazetteer
current population figures for cities, towns and places of all countries

Metropolitan areas with more than one million inhabitants - sorted by size (descending order)
Population in [1000] for 2002

Agglomeration	Population	Continent	Country
Tôkyô	31 036.9	Asia	Japan
New York-Philadelphia	29 936.9	The Americas	United States of America
México	20 965.4	The Americas	Mexico
Seoul	19 844.5	Asia	Korea (South)
São Paulo	18 505.1	The Americas	Brazil
Ôsaka-Kôbe-Kyôto	17 592.4	Asia	Japan
Jakarta	17 369.2	Asia	Indonesia
Dilli	16 713.2	Asia	India
Mumbai	16 687.8	Asia	India
Los Angeles	16 615.6	The Americas	United States of America
al-Qahira	15 546.1	Africa	Egypt
Kolkata	13 821.6	Asia	India
Manila	13 503.2	Asia	Philippines
Buenos Aires	12 916.9	The Americas	Argentina
Moskva	12 100.1	Europe	Russia
Shanghai	11 900.0	Asia	China
Rhein-Ruhr	11 297.8	Europe	Germany
Paris	11 293.2	Europe	France
Rio de Janeiro	11 246.6	The Americas	Brazil
London	11 230.5	Europe	United Kingdom
Auckland	1 154.6	Australia/Oceania	New Zealand
Phnum Pénh	1 133.8	Asia	Cambodia
São Luís	1 133.8	The Americas	Brazil
Torreón	1 132.2	The Americas	Mexico

Figure 2: Partial City Population Table.

Discover Constraints: (1) By looking at the data, we can obtain the functional dependencies (FDs) with reasonable, but not absolute, certainty. Since *Agglomeration* is a key—and particularly, a left-most-column key—we have *Agglomeration* → *Population, Continent, Country*. We have overwhelming evidence that *Population* → *Agglomeration*, *Continent*, *Country*. We also have overwhelming evidence that *Country* → *Continent*, plus overwhelming counter-evidence that *Continent* ↛ *Population*, *Country*, *Agglomeration*, and that *Country* ↛ *Population*, *Agglomeration*. Figure 3(a) shows the mini-ontology for the table after having determined the FDs and after having removed those that are redundant. (2) The data shows (nearly 100%) that the relations over (*Continent*, *Country*) and (*Country*, *Agglomeration*), are irreflexive, asymmetric, and transitive.

Match: (1) *Country* matches *Country*. (2) We parse the strings under *Agglomeration* and, using techniques in [7], discover that they are cities. Moreover, using techniques in [23], we discover that some are city groups when we recognize, for example, both *New York* and *Philadelphia* in *New York-Philadelphia*. This leads us to believe that *Agglomeration* is a group of one or more hyphen-separated cities. (3) The value characteristics of *Agglomeration*, *City*, *Continent*, and *Country* all correspond to the expected characteristics for *Name* of *Geopolitical Entity*. *Population*, however, does not.

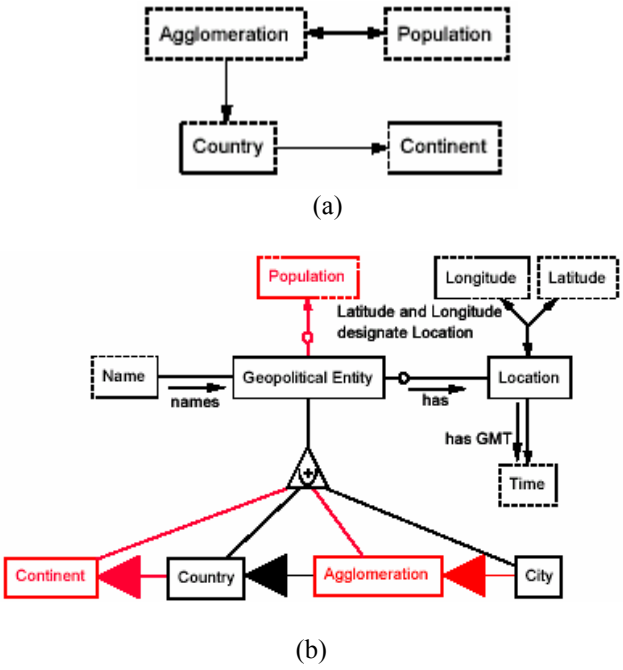


Figure 3: Mini-Ontology Constructed from the Table in Figure 2 (top), and Updated Ontology after Integrating Figure 2 into Figure 1 (bottom). (The red elements are new.)

Merge: (1) Based on ISA for *Country* and *City* in Figure 1, plus importantly that the names satisfy the name constraints for *Name* of *Geopolitical Entity*, we are led to believe that *Continent* and *Agglomeration* should be added as specializations of *Geopolitical Entity*. (2) Since the FDs are consistent with the typical 1-*n* relationships of aggregation, the names satisfy the name constraints for *Name* of *Geopolitical Entity*, and the relations are irre.exive, asymmetric, and transitive, we are led to believe that *City isPartOf Agglomeration isPartOf Country isPartOf Continent*. (3) We do not include *Population* since it satisfies neither the name constraints nor the 1-*n* constraints. (4) Because of the *isPartOf* constraints and the relationship of both *Agglomeration* and *City* with *Population*, we are led to the conclusion that *Population* should be an attribute of all the specializations under *Geopolitical Entity*. We thus relate *Population* directly to *Geopolitical Entity*. Its functional constraints, however, are in question. We observe that the counter-evidence *Continent* → *Population* and *Country* → *Population* suggests that *Geopolitical Entity* →

Population, and we observe that we have sometimes split *Agglomeration* into cities with no population value and have a few counter examples for *Population* → *Agglomeration*, *Continent*, *Country*. These observations raise too many questions, so we let the user resolve the problems (the resolutions should be synergistic, based on ontological principles and tool support [25, 29, 31]). We assume that this resolution yields the optional FD from *Geopolitical Entity* to *Population* in Figure 3(b).

- www.topozone.com/.ndresults.asp?place=Bonnie+Lake&state.ps=0&... on 6 May 2003 (Figure 4). This site uses a form: we entered "Bonnie Lake" to obtain the upper table in Figure 4. We can in addition use the site's form to look for places other than Bonnie Lake, such as New York as the lower table in Figure 4 shows. We reason, using the same *Understand-Discover-Match-Merge* steps as before. The result is in Figure 5.

Place	County	State	Type	Elevation*	USGS Quad	Lat	Lon
Bonnie Lake	Matanuska-Susit	Alaska	lake	unknown	Anchorage D-4	61.814°N	148.303°W
Bonnie Lake	Fresno	California	lake	9531 feet	Kaiser Peak	37.298°N	119.194°W
Bonnie Lake	Mono	California	lake	unknown	Tower Peak	38.189°N	119.576°W
Bonnie Lake	Jackson	Iowa	lake	unknown	Green Island	42.193°N	90.365°W
Bonnie Lake	Crow Wing	Minnesota	lake	1204 feet	Pelican Lake	46.553°N	94.126°W
Bonnie Lake	Lake	Minnesota	lake	1396 feet	Kekekabic Lake	48.086°N	91.217°W
Bonnie Lake	Klamath	Oregon	lake	6186 feet	Willamette Pass	43.549°N	122.106°W
Bonnie Lake	Aiken	South Carolina	reservoir	unknown	Seivern	33.723°N	81.420°W
Bonnie Lake	Duchesne	Utah	lake	unknown	Mirror Lake	40.711°N	110.876°W
Bonnie Lake	King	Washington	lake	unknown	Big Snow Mountain	47.566°N	121.271°W
Bonnie Lake	Spokane	Washington	lake	1790 feet	Chapman Lake	47.273°N	117.568°W

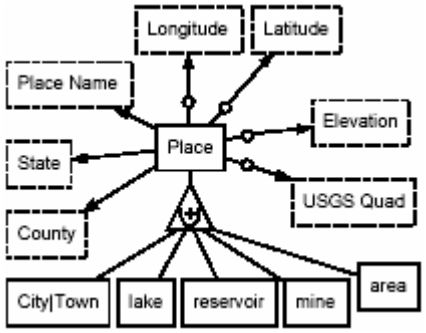
* Elevation values in this table are approximate, and often subject to a large degree of error. If in doubt, check the actual value on the map.

Place	County	State	Type	Elevation*	USGS Quad	Lat	Lon
New York	Santa Rosa	Florida	town/city	231 feet	Chumuckla	30.838°N	87.201°W
New York	Wayne	Iowa	town/city	1657 feet	Corydon	40.852°N	93.260°W
New York	Ballard	Kentucky	town/city	460 feet	Blandville	36.989°N	88.953°W
New York	Caldwell	Missouri	town/city	800 feet	Hamilton East	39.685°N	93.927°W
New York	Shelby	Missouri	area	unknown	Unknown	unknown	unknown
New York	Cibola	New Mexico	town/city	6033 feet	Cubero	35.059°N	107.527°W
New York	New York	New York	town/city	unknown	Jersey City	40.714°N	74.006°W
New York	Henderson	Texas	town/city	488 feet	Leagueville	32.168°N	95.669°W
New York	Summit	Utah	mine	unknown	Heber City	40.615°N	111.489°W

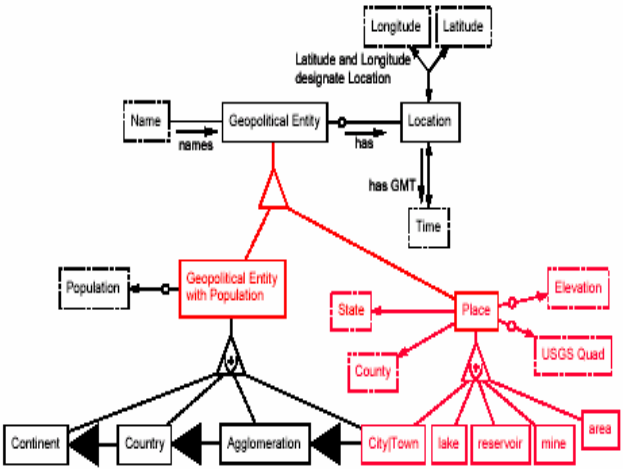
* Elevation values in this table are approximate, and often subject to a large degree of error. If in doubt, check the actual value on the map.

Figure 4: Table of Bonnie Lakes (above) and New York (below).

Understand Table: This is straightforward, except that we have at our disposal a huge table behind the form, made up of many small tables, one for each *Place* in the hidden database.



(a)



(b)

Figure 5: Mini-Ontology Constructed from the Tables in Figure 4 (top), and after processing the Tables (bottom). (The red elements are new.)

Discover Constraints: (1) First we have to observe that *Place* is not a key; yet it also contains the name we entered in our search. We conclude that the places are all different. Hence, we give each row a tuple identifier, which makes it a member of a non-lexical object set, which we call *Place*. In addition, we make a lexical object set *Place Name*, which contains the lexical name in the table under *Place—Bonnie Lake* in upper table and *New York* in the lower table in Figure 4. (2) We obtain the FDs by looking at the data and the optionals from the *unknown* values in the table. Figure 5(a) shows these constraints. (3) Next we observe that *Type* includes *City*, which we already have in our growing ontology (Figure 3(b)). With some more investigation into other tables using cities we know about such as *New York* and *Philadelphia*, we

eventually conclude that *Type* values, like cities, are each a specialization of *Place*.

Match: (1) *Longitude* and *Latitude* in Figure 5(a) match with *Longitude* and *Latitude* in our growing ontology in Figure 3(b). (2) The newly created lexical object set, *Place Name*, matches *Name* of *Geopolitical Entity*. (3) *town/city* matches *City*.

Merge: (1) Given that *City* is a specialization of *Geopolitical Entity*, and that each *Place* has a name and location (*Longitude* and *Latitude*), we conclude that *Place* is either a *Geopolitical Entity* or a specialization of a *Geopolitical Entity*. Further, since its specializations do not include *Continent*, *Country*, or *Agglomeration*, we rule out *Place* as being equivalent to *Geopolitical Entity* and conclude that *Place* must be a specialization. (2) Since we have no evidence about populations for places, by congruency [10], there must be a missing specialization object set of *Geopolitical Entity*, which we call *Geopolitical Entity with Population*. (3) We note that *City*|*Town* is in both Geopolitical *Entity with Population* and *Place*. Thus, we cannot have mutual exclusion between the two object sets and thus also no partition. We could have a union constraint, but as mentioned there are many, many more types; thus, we do not place a union constraint in the diamond under Geopolitical Entity.

These examples only illustrate the kind of thing TANGO will be able to do. When completed, TANGO should be capable of taking any readable tables in the geopolitical domain, understand them, discover constraints in them, match them with the growing ontology, and merge them such that the knowledge contained in them expands the growing ontology. Although we plan to illustrate our work only in the geopolitical domain, we intend to create TANGO so that, given a reasonable kernel ontology in any domain, it can grow ontologies for that domain.

6. TANGO applications

The focus of our TANGO project is semi-automated ontology creation, a worthy goal in and of itself. Having constructed an ontology of the type we are proposing, however, also puts us in a position to resolve many interesting and challenging problems. Examples follow:

Multiple-Source Query Processing: We can use the ontology as an integrated global schema against which we can pose queries over multiple sources [23]. Examples: What towns are within 30 miles of Bonnie Lake in Duchesne County, Utah?

Extraction Ontologies: We can use the ontology as a guide for constructing wrappers to extract geopolitical information from as-yet-unseen, semi-structured or even unstructured web pages [9].

Extraction-Ontology Generation: As [24] points out, our methodology [9] creates resilient wrappers—wrappers that do not "break" or need to be rewritten or regenerated when the wrapper encounters a changed page or a newly developed page in the same application domain. Resiliency depends on approaching the problem ontologically. Manual creation costs of ontology-based wrappers, however, are high (although the costs are mitigated by amortizing over resiliency-enabled reuse). In an effort to reduce the cost of creating extraction ontologies, we have experimented with the possibility of generating them automatically given a general global ontology and a general data-frame library. Our implementation using the Mikrokosmos ontology [32] shows that this is possible, but that it works even better when the ontology is richer in relationship structure and more tightly integrated with the data-frame library.

Data Integration: Automating data integration tends to work best when when rich auxiliary knowledge sources provide a basis for analyzing sources from multiple points of view, including dictionaries of synonyms and hypernyms, value characteristics, expected values, and structure [7]. Indeed, we can achieve over 90% precision and recall both for direct as well as many indirect matches between data sources [26]. We intend to endow TANGO ontologies with the characteristics it needs to assist in data integration.

Semantic Web Creation and Superimposed-Information Generation: 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. In [33] we proposed to show how to bridge the gap between the current web and the semantic web by semi-automatically converting Resource Description Framework Schemas (RDFS's) and DAML-OIL ontologies into data-extraction ontologies [9]. Extracted data will then be converted to RDFS, making it accessible to semantic-web agents and, in addition, will superimpose the meta-data of the extracted information over the document for direct access to data in context, as suggested in [34]. We believe that the TANGO-created ontologies will work even better for this application.

Agent Interoperability: We have begun a project in which we wish to experiment with scalable ontology-based matching for agent communication [35]. Rather than relying on a specified, shared ontology, a common communication language, and a specified message format to achieve interoperability, we intend to use an independent global ontology to encode and decode messages exchanged among agents. TANGO can help us create the independent knowledge we need for an application of interest.

Document Image Analysis: The proposed techniques can eliminate some common shortcomings of current table-reading and forms-processing software [12].

7. Conclusions

This paper introduces work in progress on ontology generation through web tables. It describes TANGO as a system that captures several techniques and will embody results from our research in various fields. Once completed TANGO will be able to run across the full spectrum of human intervention—from fully automatic, where it will do its best even when encountering ambiguous and contradictory information, to fully user driven, where it will do nothing more than build ontologies as directed by its users. Between these extremes, we will allow for synergistic Issue/Default/Suggestion (IDS) usage [25], where TANGO will do all it can to resolve difficulties, but will point out issues it encounters, state what its default action will be, and suggest possible alternatives a user may choose instead. We will also instrument TANGO with a monitoring system that will log both system and user actions.

8. References

[1] M.A. Bunge. *Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World*. Reidel, Boston, 1977.

[2] R. Mizoguchi and M. Ikeda. "Towards Ontology Engineering," *Proc. of The Joint 1997 Pacific Asian Conference on Expert systems / Singapore International Conference on Intelligent Systems*, Singapore, 1997, pp. 259-266.

[3] M. Gruninger and J. Lee. "Ontology applications and design," *Communications of the ACM*, February 2002, 45(2), pp. 39–41.

[4] Y. A. Tijerino, M. Ikeda, T. Kitahashi and R. Mizoguchi. "A methodology for building expert systems based on task ontology and reuse of knowledge—Underlying philosophy of task analysis interview system Multis—" *Journal of the Japanese Society for Artificial Intelligence*. July 1993, 8(4), pp. 476-487.

[5] C.W. Holsapple and K.D. Joshi. "A collaborative approach to ontology design," *Communications of the ACM*, February 2002, 45(2), pp. 42–47.

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

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

[8] D.W. Embley. "Programming with data frames for everyday data items." *Proc. of the 1980 National Computer Conference*, Anaheim, California, May 1980, pp. 301–305.

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

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

[11] S.W. Liddle, D.W. Embley, and S.N. Wood.eld. "An active, object-oriented, modelequivalent programming language." In M.P. Papazoglou, S. Spaccapietra, and Z. Tari, editors, *Advances in Object-Oriented Data Modeling*, MIT Press, Cambridge, Massachusetts, 2000, pp. 333–361.

[12] D. Lopresti and G. Nagy. Automated table processing: An (opinionated) survey. In *Proceedings of the Third IAPR Workshop on Graphics Recognition*, Jaipur, India, September 1999, pp. 109–134.

[13] D. Lopresti and G. Nagy. "Issues in ground-truthing graphic documents." In D. Blostein and Y-B. Kwon, editors, *Graphics Recognition—Algorithms and Applications*, Lecture Notes in Computer Science, LNCS 2390, Springer Verlag, 2002, pp. 46–66. (Selected papers from the Fourth International Workshop on Graphics Recognition, GREC 2001).

[14] K. Tubbs and D.W. Embley. "Recognizing records from the extracted cells of microfilm tables." *Proc. of the Symposium on Document Engineering (DocEng'02)*, McLean, Virginia, November 2002, pp. 149–156.

[14] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. "User-driven ontology evolution management." *Proc. of the 13th European Conference on Knowledge Engineering and Management (EKAW-2002)*, Sigüenza, Spain, October 2002. Springer Verlag.

[16] Homepage for BYU data extraction research group. URL: http://osm7.cs.byu.edu/deg/index.html.

[17] L. Li, G. Nagy, A. Samal, S. Seth, and Y. Xu. "Integrated text and line-art extraction from a topographic map." *International Journal of Document Analysis and Recognition*, June 2000, 2(4), pp. 177–185.

[18] D.W. Embley and G. Nagy. "On the integration of lexical and spatial data in a unified high-level model." *Proc. of the International Symposium on Database Systems for Advanced Applications*, Seoul, Korea, April 1989, pp. 329–336

[19] G. Nagy, M. Mukherjee, and D.W. Embley. “Making do with .nite numerical precision in spatial data structures.” In *Fourth International Symposium on Spatial Data Handling*, Zürich, Switzerland, July 1990, pp. 55–65.

[20] G. Nagy and S. Wagle. “Geographic data processing.” *ACM Computing Surveys*, June 1979, 11(2), pp. 139–181.

[21] G. Nagy. “Geometry and geographic information systems.” In C. Gorini, editor, *Geometry at Work*, Notes Number 53, The Mathematical Association of America, 2000, pp. 88–104.

[22] D. Lopresti and G. Nagy. “A tabular survey of table processing.” In A.K. Chhabra and D. Dori, editors, *Graphics Recognition—Recent Advances*, Lecture Notes in Computer Science, LNCS 1941, Springer Verlag, 2000, pp. 93–120.

[23] D.W. Embley, C. Tao, and S.W. Liddle. “Automatically extracting ontologically specified data from HTML tables with unknown structure.” In *Proceedings of the 21st International Conference on Conceptual Modeling (ER’02)*, Tampere, Finland, October 2002, pp. 322–327.

[24] A.H.F. Laender, B.A. Ribeiro-Neto, A.S. da Silva, and J.S. Teixeira. “A brief survey of web data extraction tools.” *SIGMOD Record*, June 2002, 31(2), pp. 84–93.

[25] J. Biskup and D.W. Embley. “Extracting information from heterogeneous information sources using ontologically specified target views.” *Information Systems*, 2003, 28(3), pp. 169-212.

[26] L. Xu and D.W. Embley. Discovering direct and indirect matches for schema elements. In*Proc. the 8th International Conference on Database Systems for Advanced Applications (DASFAA'03)* 2002.

[27] N. Guarino. Some ontological principles for designing upper level lexical resources. In *Proc. of the First International Conference on Language Resources and Evaluation*, Granada, Spain, May 1998.

[28] Y. Wand, V.C. Storey, and R. Weber. “An ontological analysis of the relationship construct in conceptual modeling. “*ACM Transactions on Database Systems*, December 1999, 24(4), pp. 494–528.

[29] C.A. Welty and N. Guarino. “Supporting ontological analysis of taxonomic relationships.” *Data & Knowledge Engineering*, 2001, 39(1), pp. 51–74.

[30] J. Evermann and Y. Wand. “Towards ontologically based semantics for UML constructs.” In *Proc. of the 20th International Conference on Conceptual Modeling (ER2001)*, Yokohama, Japan, November 2001, pp. 513–526.

[31] N. Guarino and C. Welty. Evaluating ontological decisions with OntoClean. *Communications of the ACM*, February 2002, 45(2), pp. 61–65.

[32] S. Beale and S. Nirenburg. “Breaking down the barriers: The mikrokosmos generator.” In *Proc. of the 4th Natural Language Processing Pacific Rim Symposium (NLPRS’97)*, Phuket, Thailand, 1997, pp. 141–148.

[33] T. Chartrand. “Ontology-based extraction of RDF data from the world wide web.” Technical report, Brigham Young University, Provo, Utah, 2002. (thesis proposal, currently at www.deg.byu.edu/proposals/index.html).

[34] D. Maier and L. Delcambre. “Superimposed information for the Internet.” In S. Cluet and T. Milo, editors, *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB’99)*, Philadelphia, Pennsylvania, June 1999.

[35] M. Al-Muhammed. “Dynamic matchmaking between messages and services in multiagent systems.” Technical report, Brigham Young University, Provo, Utah, 2002. (thesis proposal, currently at ww.deg.byu.edu/proposals/index.html).