Acquisition of Ontological Knowledge from Canonical Documents

Raphael Malyankar
Dept. of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287, USA.
E-mail: rmm@acm.org

(Position Paper)

Abstract

This paper describes experiences with quasi-automated creation of a computational ontology for maritime information from a mixed collection of source material. Based on these experiences, hypotheses and conclusions concerning the creation of computational ontologies for engineering and other technical or scientific domains are presented. Heuristics for resolving anomalies in ontologies generated from mixed sources are also described.

1 Introduction

This paper describes our experiences with ontology acquisition in the context of maritime information. Ontological information is acquired from multiple types of sources, including standards documents, database schemas, lexicons, collections of symbology definitions, and also by inference from semi-structured documents. This is followed by a description of the computational approach to rationalization, alignment, and merging of the ontological information derived from these sources. The computational ontology thus created is intended to be used in creating a Maritime Information Markup Language (MIML) for tagging of documents in this domain. An example of the kind of application that will be enabled is a question-answering system that extracts only necessary and relevant information from marked-up text documents.

The observations and heuristics described in this paper apply to domains - here, maritime information - where ontological knowledge must be acquired from different types of source material. It appears that in some domains, the sub-ontologies thus generated are likely to different not only linguistically, but also in their topological profiles (i.e., depth and other structure). The heuristics described in this paper are designed for a computational approach to combining such sub-ontologies.

2 Sources of Ontological Knowledge

The sources used for ontological knowledge were selected from a canonical set, that is, thery are documents accepted within the domain as normative and that are widely used.

2.1 Standards Documents

The most recent normative standard for digital nautical chart content is the S-57 Standard for [International Hydrographic Organization, 1996]. The ‘object catalog’ section of this document consists of a list of chart entities, definitions, and entity attributes, which gives us a collection (sic) of domain entities that can be considered canonical as far as the scope of the standard goes. Extraction from this ‘object catalog’ was automated by using graph traversal programs that exploit links between entities and attributes in the object catalog. The automated extraction resulted in 173 classes and 186 slots. A comparison of 10% (selected at random) of the extracted information with the original source indicated error rates of 8% to 20% (for different categories of ontological knowledge - classes/types/attributes). The additional effort needed to reduce this error in the automated portion of the extraction was not undertaken, as it proved no very laborious task to make the corrections by hand (about 10 hours for a non-expert who compared the extracted ontology with the original source).

A second source was the Spatial Data Transfer Standard [FGDC, 1998]. The parts we used were the sections that list ‘included terms’ (analogous to a synonym list) and attribute definitions. Extraction from this was less satisfactory in some ways, since these sections are less rigorous than the object catalog of the S-57 standard, but, on the other hand, the synonym list covers more of the terms used in practice.

While the S-57 standard is normative, there are two deficiencies involved in using it:

1. It is limited in scope. This standard covers only objects (entities) that are used in digital nautical charts. Important concepts such are weather conditions are not mentioned at all, and other concepts such as tides are mentioned only incidentally or in an implicit manner, for example in defining entity classes and as attribute qualifiers for entities (e.g., *foreshore areas*, the part of the shore covered and uncovered by tides).
2. It uses a restricted terminology, i.e., usually only one of multiple synonymous terms. The ‘missing’ terms are sometimes used in other documents and it is necessary to establish synonym relationships to facilitate understanding.

Further semantic structure is induced from lexical clues and attribute sets. The heuristics used for this induction process currently consist of lexical clues from the linguistic similarity of entity names and entity definitions, and comparison of attribute sets to compute measures of the semantic distance between attributes. For example, there are multiple ”beacon” objects in the object catalog (”cardinal” beacon, ”danger” beacon, etc.). Lexical comparison of the object names for these several classes, and of the descriptions associated with these classes (also scraped from the abovementioned object catalog) indicated the possibility of a ‘beacon’ class as a superclass for these several classes. This is further described in Section 4.

2.2 Databases and Schemas

The primary database we have used so far is the sample Digital Nautical Chart (DNC) data files available from NIMA. It has somewhat more *semantic* structure than the aforementioned standards, consisting as it does of feature classifications organized by ‘layers’, for example, environmental features, cultural features, land cover features, etc. (‘Feature’, as used in the domain, is equivalent to ‘class’). Induction of ontological knowledge from this consisted of mapping the structure to a class hierarchy. This mapping was also done automatically from the schema for the database. It resulted in 134 classes of which 118 are feature classes, 12 are coverage classes, and 4 are geographic structure type (point, line, area, or text) classes.

As with the S-57 standard, this database and schema covers only chart entities, and the terminology is even more restricted (and to some extent, more opaque) than the S-57 standard, due to the use of abbreviated names for entities and attributes, and the lack of textual definitions.

2.3 Lexicons and Symbology Definitions

A separate effort used Protege [Grosso *et al.*, 1999] and a standard collection of symbology definitions from NOAA’s Chart No. 1 [National Oceanic and Atmospheric Administration, 1997] to create an ontology of navigation aids, hazards, and other entities. Chart No. 1 is a collection of symbology for nautical charts accompanied by brief definitions of what the symbol stands for. It is organized semantically (in that related symbols are in the same section or subsection). This was supplemented with a widely popular publication on navigation and seamanship (*Chapman Piloting* [Maloney, 1999]) and an online dictionary of chart terms (discovered and used by the creator, a student unfamiliar with nautical terms). Ontology creation based on these documents consisted of manual entry of information using Protege, due to the lack of electronic versions of the symbology definitions. Approximately 500 classes and 100 slots resulted from this effort, which was carried out by non-experts using the publications mentioned. (The paucity of slots is due to the nature of the documents, which contain little mention of details corresponding to symbols).

2.4 Semi-Structured Normative Material

The *United States Coast Pilot* is a 9-volume series containing information that is important to navigators of US coastal waters (including the Great Lakes) but which cannot be included in a nautical chart. Each volume consists of 200 to 300 pages or more of two-column text in 10-point type. Included are photographs, diagrams, and small maps. The flow of text follows the coastline geographically, e.g., from north to south. This is a ‘lightly structured’ document, with each volume containing a preliminary chapter containing navigation regulations (which includes a compendium of rules and regulations, specifications of environmentally protected zones, restricted areas, etc.), followed by chapters dealing with successive sectors of the coast. Each chapter is further divided into sections (still in geographical order); each section is further divided into sub-sections and paragraphs describing special hazards, recognizable landmarks, facilities, etc. The internal structure of subsections and paragraphs provides taxonomical hints, indicating, for example, which leaf entities are categorizable as sub-classes of weather conditions, as well as providing a small amount of additional taxonomical information that extends taxonomies derived from other classes (e.g., tide races as a form of navigational hazard). The *Coast Pilot* is normative (in the sense of using well-understood terms) and comprehensive. A version marked up with XML would have proved invaluable for ontology learning, but there is no such version available at this time.

2.5 Other Sources

Online content proved a useful and irreplaceable source of some information, especially attributes relating to weather data. Entry of this part was entirely manual. Other sources to be used include the Ports list and Light list, for information on port facilities and navigation aids respectively.

3 Alignment, Merging, and Rationalization

We have discovered that though there is a certain amount of duplication between the above sources, they are largely independent and produce different parts of the taxonomy for the maritime information domain as a whole, and sometimes different taxonomical structures for some parts of the domain. The need to merge and align the ontologies generated from the sources mentioned naturally arises, along with the need to reconcile conflicts between different ontologies. This section describes the major issues arising in combining different ontologies, and the techniques adopted to resolve them. In addition, we are using some of these heuristics to rationalize individual ontologies by detecting anomalies in their structure.

3.1 Alignment and Merging

There are at least two distinct taxonomic hierarchies in our source material: (i) a classification into point, area, or line features, and (ii) a different, natural, semantic hierarchy (natural in the sense that it is the categorization that a human tends to create). Item (i) is attributable to the original purpose of the standards document that produced such a taxonomy — it was intended for geographical information systems and therefore its point of view is that of a computer graphics system instead of a knowledge-based system. Alignment of the ‘sub-ontologies’ consists of assembling a jigsaw puzzle in the sense of [Noy and Musen, 2000].

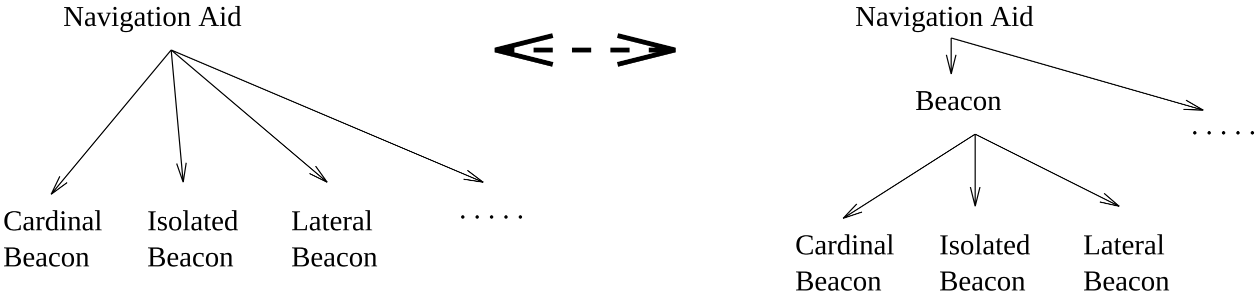

Figure 1: Merging Similar Classes

3.2 Resolution of Structure Mismatches

Another issue is structure mismatch, leading to what can be called the reification question — should a concept distinguishing two entities be made manifest through distinct values for a slot, or should the distinction be manifest as a type within the class (thus giving distinct sub-classes). We have discovered that automated extraction from an object catalog or schema tends to produce shallow, bushy, class hierarchies (i.e., it prefers translating distinctions into a range for an attribute slot), while manual creation tends to create deeper and less bushy type hierarchies. It appears that choosing between the two may be merely a question of convenience of utilization, but investigations into this issue continue. (This difference may be a characteristic of the source of ontological knowledge — databases vs. other source material.) The immediate issue raised by this is that ontology merging or assembly will need to resolve questions of whether to sub-class a class from one partial ontology, or de-sub-class a corresponding collection of classes in the other, and how to detect this problem, i.e., identify which slot can be used as a sub-class type.

3.3 Rationalization

The term ‘rationalization’ is used here to mean removal of anomalies within a single ontology, such as slots with different names but playing the same role, multiple indistinguishable (or almost indistinguishable) sibling classes that are not specializations of their own distinguished abstract class, etc. Some such situations are justified and necessary, but where ontologies are generated automatically, it appears that numerous such anomalies may creep in.

4 Computational Approach

A computational method for solving the problems described earlier has been designed and partially implemented. The approach to combining the ontologies and resolving conflicts is reinforcement-based in that multiple heuristics are applied to detect candidates for merging, renaming and other operations. Instead of making suggestions to a user based on triggering single rules, the set of recommendations obtained by applying all applicable heuristics is presented to the user (as a list of positive or negative recommendations for possible actions); the user is expected to decide based on the evidence presented and considerations that may not be known to the computational recommender. The current set of heuristics, and the recommendations indicated by them, is described below:

Rule 1: Classes whose names are linguistically synonymous are suggested as candidates for merging. Distance between classes is measured in terms of the use of synonyms within class names. For example, two different ontologies contain ‘Bridge’ classes (the same word is used in each). Further, cognate terms are discovered by looking for meaningful synonyms within the class name. Figure 1 shows an instance of such cognate names (the different kinds of beacons). A merger recommendation is issued when this rule is triggered.

Rule 2: Class pairs which have a high proportion of slot names that are linguistically synonymous, and sufficiently low differences in the rest of their slots, are nominated as candidates for merger or alignment. As for Rule 1, distance between slot names is measured in terms of the appearance of synonyms.

Comparison of two classes C_1 and C_2 with slot sets SL_1 and SL_2 respectively, returns a 3-tuple (C, D_{12}, D_{21}), where C is a numeric value representing the degree of commonality of the slot sets and D_{12} and D_{21} are numeric values representing the respective difference sets between the individual slot sets SL_1 and SL_2 and the union set $SL_1 \cup SL_2$ of all the slots for either class. For example, D_{12} can be computed as the number of slots of C_1 that are not synonyms of slots of C_2. This computation is similar to that described by Chalupsky [2000], but uses individual elements instead of an all-round measure computed by combining the 3 numeric values.

Rule 2 recommends merger/alignment if $C > 0$ and $D_{12}, D_{21} < \epsilon$, where ϵ is chosen to minimize spurious positive recommendations.

Rule 3: Conceptual relatedness for class pairs is computed by comparing the class names using a lexicon of ‘included terms’, derived from the SDTS [FGDC, 1998]. This means that hypernym/hyponym relationships between terms within class names are included, in contrast to Rule 1, which uses synonyms. The reason is that the ‘included terms’ are expected to be likely to result in alignment operations instead of merger operations.

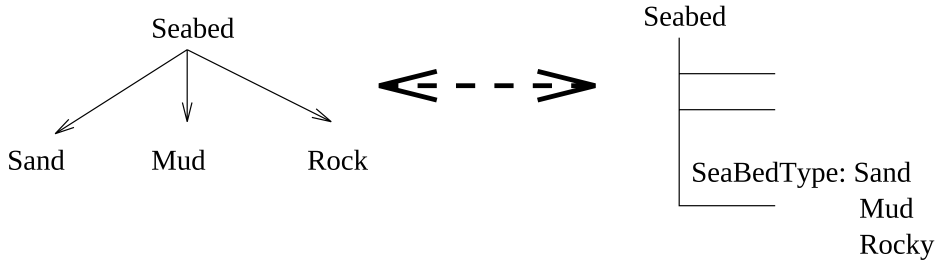

Figure 2: Structure mismatches

Similarity comparison in our heuristics is keyword based, in that it assumes (supported by human observation of the class and slot names) that names are of the general form $\{$*QualifyingTerm KeyTerm*$\}$ (or *AdjectivalPhrase Noun*). Greater importance is given to the *KeyTerm* in computing semantic closeness, since the *QualifyingTerm* portion generally appears to define a sub-type of an abstract class denoted by *KeyTerm*. A consequent limitation is that special requirements on the internal structure of class and slot names must be imposed, and further, the heuristics produce spurious results in several cases.

Partial synonyms (complex names with synonymous key terms) are recommended as candidates for abstraction or merger, e.g., by merging their superclasses.

Rule 4: Concept similarity for class pairs is computed by comparing the names of their slots, using the same lexicon as before. The resultant recommendation suggests mergers of classes.

Rule 5: Sibling classes without unique slots, i.e., those that have only inherited slots, are examined. The implied solution is to merge the two into their parent class or introduce an intermediate class and add a *type* or equivalent slot to the immediate super class. (But see rules 8 and 9 for possible reasons not to accept the recommendations generated by this rule.)

Rule 6 : Subsumption relationships are detected by comparison of slot names as in Rule 2, but the implication and conditions respectively that must be satisfied by C, D_{12}, and D_{21} are now: C_1 is a subclass of C_2 if $C > 0$, $D_{12} > 0$, $D_{21} = 0$.

Rule 7: This heuristic is intended to detect structure mismatches of the type vs. subclass category described earlier. Figure 2 shows an instance of such a mismatch, arising from capturing the same information from different sources. Siblings X_a, X_b, ... of class X are compared to allowed value ranges for slot S of class Y; if the allowed values for slot S match (that is, are linguistically close to) the names of siblings X_a, X_b, ..., a structure mismatch is indicated. This rule is applicable when values are categorical variables. This rule detects the commonality between different classes, each corresponding to a sea floor characteristic type (sand, pebbles, etc.), combining them into a single feature with the sea-floor type as a slot.

Two further rules are being implemented; these operate not on the ontologies themselves, but on the knowledge base, methods used for accessing it, and its contents:

Rule 8: Determine how often the instances of a class are retrieved in isolation. If there are many requests for entities of a specific class, there may be implementation reasons for retaining the class as a unique class. This rule, of course, can be effectuated only after a study of actual use of the ontology.

Rule 9: Determine the population of instances for each concrete class, and compare with those for its siblings or merger candidates. If the population size is large, or if there is significant skew in the population of merger candidates, there may be implementation reasons (e.g., if instances are ultimately retrieved from a database) for retaining distinct classes. As with Rule 8, this heuristic can be investigated only after populating the underlying knowledge store (database, frames, etc.).

Rules 8 and 9 are expected to produce contra-indications when triggered, i.e., recommend *against* mergers or alignment.

Instead of applying rules individually and effecting their suggestions as detected, we use them to detect problems and suggest changes; the changes actually effectuated are expected to be those suggested by multiple rules, i.e., those supported by multiple forms of evidence.

5 Implementation

All but one of the ontologies extracted are currently in the format used by the Protégé tool. However, implementation of the rules above is currently 'off-line' as far as Protégé is concerned, that is, it is being done by a separate program that uses a translation of the ontologies into a different format. This was adopted due to the necessity of including the ontologies in a Web server back-end program for extraneous reasons (the question answering site mentioned earlier). Currently individual rules are applied to pairs of ontologies and suggestions (and contra-indications) printed for separate evaluation by a human user. Work on incorporating these rules into a Protégé plugin will commence shortly.

6 Related Work

Noy and Musen [1999; 2000] describe an algorithm and tool for merging ontologies in Protégé. Chalupsky [2000] describes OntoMorph, a tool for translating symbolic knowledge from one KR formalism to another, and describes ontology alignment in [Chalupsky *et al.*, 1997]. Hovy [1998] describes a procedure for ontology alignment and heuristics for suggestions, including pattern matching on strings, hierarchy matching and data/form heuristics .

Ontology analysis and merging in *Chimæra* is described in [McGuiness *et al.*, 2000]. Syntactic analysis of class and slot names, taxonomic resolution, and semantic evaluation (for example, slot/value type checking and domain-range mismatches) are also discussed.

All the current methods for ontology alignment and merging generally use linguistic methods of determining similarity for class and slot names, as is done in some of the heuristics described in Section 4 in this paper. Our approach appears to differ from those described in the form and utilization of the results of comparisons, and apparently also in the use of multi-criterion indicators/contra-indicators for suggesting operations as compared to computing a single score. Further, an additional heuristic is used for concept (class) linking, by comparing similarities between the member slots of classes. Structure mismatches are also mentioned by Chalupsky. Access convenience and instance population-based heuristics (rules 8 and 9) have not been discussed in descriptions of ontology merging and alignment.

7 Conclusion

The source material described here constitutes in a sense a canon for the domain of maritime information, in that the collection is (except for the items in Section 2.5) normative and comprehensive for the domain of maritime information. Based on our observations while deriving ontological knowledge from it, the following positions and hypotheses are put forward, admittedly on the basis of a single experience:

- No single source (standard, schema, etc.), will suffice for a reasonably complete computational ontology. This fairly tame conclusion has been remarked by other groups, and leads to the next:

- No single *type* of source will suffice for learning a computational ontology; i.e., it will be necessary to include multiple *kinds* (structured, semi-structured, lexicon-like, etc.) of sources; further, after the possibilities of 'organized' or standardized sources have been exhausted, it will be necessary to fill in the gaps with inductions from unstructured or 'free-form' content; this means that no single means of ontology learning will suffice for a reasonably complete ontology.

- Ontological information extracted from different sources will be in qualitatively different structural forms; therefore, an attempt at combining these different sub-ontologies into an overall whole will need to resolve these structural differences before any other form of merging can be usefully applied.

- The above will hold even for a domain that has experienced significant organization and standardization efforts.

A computational approach for resolving anomalies in ontological knowledge that exhibits the characteristics mentioned above was also presented, and investigations into its use and applicability are ongoing.

Acknowledgments

The efforts of Helen Wu and Koi-Sang "Leo" Leong in entering ontological information and scraping ontological information from on-line sources are gratefully acknowledged. This work was partially supported by the National Science Foundation under grant EIA-9983267, NOAA, and the U.S. Coast Guard. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of these agencies.

References

[Chalupsky *et al.*, 1997] H. Chalupsky, E. Hovy, and T. Russ. Progress on an automatic ontology alignment methodology, 1997. ksl-web.stanford.edu/onto-std/hovy/index.htm.

[Chalupsky, 2000] H. Chalupsky. Ontomorph: a translation system for symbolic knowledge. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000), San Francisco, CA.* Morgan Kaufman, 2000.

[FGDC, 1998] FGDC. Spatial data transfer standard. Federal Geographic Data Committee, U. S. Geological Survey. Proposed standard, 1998.

[Grosso *et al.*, 1999] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. H. Gennari, S. W. Tu, and M. A. Musen. Knowledge modeling at the millennium (the design and evolution of Protege-2000). Technical report, Stanford University, Institute for Medical informatics, Stanford, CA, 1999. Technical Report SMI-1999-0801.

[Hovy, 1998] E.H. Hovy. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC). Granada, Spain*, 1998.

[International Hydrographic Organization, 1996] International Hydrographic Organization. IHO transfer standards for digital hydrographic data, edition 3.0, 1996.

[Maloney, 1999] Elbert S. Maloney. *Chapman Piloting: Seamanship and Boat Handling*. Hearst Marine Books, New York, 63rd edition, 1999.

[McGuiness *et al.*, 2000] D. McGuiness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado,* April 2000. Tech. report KSL-00-16, Knowledge Systems Laboratory, Stanford University.

[National Oceanic and Atmospheric Administration, 1997] National Oceanic and Atmospheric Administration. Chart no. 1: Nautical chart symbols, abbreviations, and terms, 1997.

[Noy and Musen, 1999] N. F. Noy and M. Musen. SMART: Automated support for ontology merging and alignment. In *Twelth Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Canada*, 1999.

[Noy and Musen, 2000] N. F. Noy and M. A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. Technical report, Stanford University, Institute for Medical informatics, Stanford, CA, 2000. Technical Report SMI-2000-0831.