Mapping Ontologies into Cyc

Stephen L. Reed and Douglas B. Lenat

Cycorp, Inc.
3721 Executive Center Dr.
Austin, Texas, 78704
reed@cyc.com, lenat@cyc.com

Abstract

The advent of Web services, and the Semantic Web described by domain ontologies, highlight the bottleneck to their growth: ontology mapping, merging, and integration. In this paper we present the process by which over the last 15 years several ontologies of varying complexity have been mapped or integrated with Cyc, a large commonsense knowledge base. These include SENSUS, FIPS 10-4, several large (300k-term) pharmaceutical thesauri, large portions of WordNet, MeSH/Snomed/UMLS, and the CIA World Factbook. This has to date required trained ontologists talking with subject matter experts. To break that bottleneck – to enable subject matter experts to directly map/merge/integrate their ontologies – we have been developing interactive clarification-dialog-based tools.

Introduction

The advent of Web services and the Semantic Web described by domain ontologies highlight the bottleneck to their growth: ontology mapping, merging, and integration. In this paper we present the process by which over the last 15 years several ontologies of varying complexity have been mapped or integrated with Cyc, a large commonsense knowledge base. These include SENSUS, FIPS 10-4, several large (300k-term) pharmaceutical thesauri, large portions of WordNet, MeSH/Snomed/UMLS, and the CIA World Factbook. This has to date required trained ontologists talking with subject matter experts. To break that bottleneck – to enable subject matter experts to directly map/merge/integrate their ontologies – we have been developing interactive clarification-dialog-based tools.

Cyc

The Cyc knowledge base (KB) has an ontology of over 100k atomic terms axiomatized by a set of over 1M hand-crafted assertions stated in n^{th}-order predicate calculus employing over 10k predicates which are themselves first class terms in the KB. The KB spans human consensus reality, i.e., commonsense knowledge. Nonatomic terms such as (GovernmentFn Sudan) obviate the need for families of terms such as TheGovernmentOfSudan; this is recursive, hence there are effectively a countably infinite number of such terms already referable to in Cyc's KB. The Cyc inference engine includes general theorem provers but maintains efficiency by relying on a suite of over 500 heuristic level modules: special purpose data structures and algorithms that support the most heavily used predicates and make narrow but common types of reasoning run very quickly. E.g., answering a question like "Is George Bush tangible?" can and should be done through graph-walking or even better cached graph lookup, not through a 9-step proof. The KB is divided into locally-consistent contexts called micro-theories (mt's), each of which contains both content – a body of assertions – and assumptions shared by all those assertions. The contexts are first-class terms in the KB; they appear in assertions, they are organized into an ontology, and so on. An mt's assumptions are themselves Cyc assertions about time, space, granularity, topic, etc.

The Cyc KB is language-independent, except insofar as part of it *is* an English lexicon. E.g., three different Cyc terms represent the English word "bear" and the language-independent concepts of carrying, of bears, etc., and assertions involving the predicate "denotations" relate the word to those meanings. That lexicon today covers 15k root words and 20k proper names.

Much to-do has been made – by everyone from Aristotle to Sowa – about what the upper ontology (in which the most abstract and general sorts of concepts are defined) should be. But we have found empirically that most of the "action" – the minute-by-minute *work* of ontology mapping – is performed primarily at the middle and lower levels of the ontology, where the defining vocabulary for a domain is located and where it is being "hooked in" to the existing ontology. If one didn't have a large existing ontology, if one only had an upper level ontology, then of course the "hooking-in" action would have to occur at the upper level.

The variant of predicate calculus that we use to represent the assertions is called CycL; we shall present evidence suggesting that its level of expressiveness is both necessary and sufficient for mapping/merging/integrating ontologies. In particular, the expressiveness is necessary to handle the cases where the terms don't correspond exactly 1-to-1

across ontologies, but rather require a full-fledged axiom to express the mapping.

Background

Ontology mapping is related to the well researched areas of taxonomy and thesaurus merging. Doerr [2001], describes semantic differences among thesauri that affect the mapping process. Wiederhold [1994], describes ontology composition, and domain ontology differences. Generalizing these findings and adding other semantic issues peculiar to Cyc, we obtain the following list of differences between a source ontology and the Cyc ontology (in order of complexity):

- ♦ Terminological Differences
  - Different names for the same concept
  - Related but different concepts
  - More specialized or general versions of the same concept
  - Attributes *vs*. functions *vs.* predicates representation
- ♦ Simple Structural Differences
  - Two ontologies are similar yet disjoint
  - One ontology is a subset of the other
  - One ontology is a reorganization of the other
- ♦ Complex Structural Differences
  - E.g., having action predicates *vs.* reified events
- ♦ Fundamentally different representations
  - E.g., Bayesian probabilistic *vs.* truth-logic

Term Mapping Experience

Term mapping (to Cyc's ontology) constitutes the majority of our ontology mapping experience. The source ontology typically contains a finite set of terms having well defined attributes. The source term taxonomy is usually used in the mapping, but that is not always the case.

Term Mapping Meta Ontology

Cyc has three general-purpose term mapping predicates:

`(synonymousExternalConcept TERM SOURCE STRING)` means that the Cyc concept `TERM` is synonymous with the concept named by `STRING` in the external data source `SOURCE`. This is the simplest sort of 1-to-1 term mapping.

`(overlappingExternalConcept TERM SOURCE STRING)` means that the Cyc concept `TERM` overlaps semantically with the concept named by `STRING` in the external data source `SOURCE`. Either the overlap is *almost* complete, or else the relationship is unclear, or else this next predicate should be used instead of `overlappingExternalConcept`:

`(extConceptOverlapsColAndReln COL RELN SOURCE STRING)` means that the external structured data source `SOURCE` variously uses the term named by `STRING` as a value that semantically maps to the Cyc Collection `COL`, and as a slot that maps to the Cyc `BinaryPredicate RELN`.

Cyc has one general-purpose code mapping function:

`(MeaningInSystemFn INFOSOURCE STRING)`, applied to a character string or code `STRING` in some external information system `INFOSOURCE`, returns whatever concept is meant by that string or code in that system.

E.g., the value of

```
(MeaningInSystemFn
        WordNet-1995Version
       "N221566")
```

is the concept (or WordNet synset) represented by the synonyms (rampart|bulwark|wall), meaning "an embankment built around a space for defensive purposes" in the WordNet ontology.

We now give a few examples of our experience using this machinery to map various ontologies to Cyc's.

FIPS 10-4

The Federal Information Processing Standards (FIPS) are a set of standards that describe document processing codes. We mapped FIPS codes that designate non-US countries and their principal administrative subdivisions into Cyc, creating new terms for geographical entities not already in Cyc. This is the simplest form of ontology mapping, in which a single relationship is mapped, and in which the missing terms in the reference Cyc ontology are easily identified and created. A single microtheory (context) contains the mapping assertions. Should it be required to subsequently map a new version of the FIPS country and city codes, then a new microtheory will be created for that mapping.

For the below mapping assertion, we created the term `Waikohu-CountyNewZealand` and associated the code "NZ86" with Waikohu, the New Zealand county. Furthermore, the new term was asserted to be a geopolitical subdivision of New Zealand, as that fact was given by the source ontology.

```
(synonymousExternalConcept Waikohu-
CountyNewZealand FIPS10-4Information1995 "NZ86")

(geopoliticalSubdivision NewZealand Waikohu-
CountyNewZealand)
```

We mapped 3,418 terms from the FIPS 10-4 ontology into Cyc.

MeSH

Medical Subject Headings (MeSH) were mapped into Cyc from the expanded 1997 version of the National Library of Medicine thesaurus of Medical Subject Headings. We

created not only new terms for concepts new to Cyc but created new relationships and linked the mapped source terms into a semantically richer portion of the Cyc KB.

For example:

```
(overlappingExternalConcept
  InferiorMesentericVein MeSH-Information1997
  "Mesenteric Veins | A7.231.908.670.385")

(isa InferiorMesentericVein
  UniqueAnatomicalPartType)

(genls InferiorMesentericVein Vein)

(genls InferiorMesentericVein
  DirectedCustomaryPath)
```

The last assertion for the mapped term "Mesenteric Veins | A7.231.908.670.385", states that the Inferior Mesenteric Vein is a subclass of `DirectedCustomaryPath`, which enables Cyc to reason about the vein as a mathematical directed graph system. We mapped 252 terms from MeSH to Cyc.

SENSUS

SENSUS is a large, 70,000 term ontology derived from WordNet and the Pennman Upper Model. We mapped 201 SENSUS terms that matched existing Cyc terms. For example:

```
(synonymousExternalConcept
  DisjointSetOrCollection
  SENSUS-Information1997 "DISJUNCTIVE-SET")
```

Open Directory

The Open Directory web topic directory is a very large taxonomy of over 400,000 classes and 3,000,000 instances which is available in RDF. In a commercially sponsored effort, we linked over 10,000 class terms to Cyc, using a workflow application to guide our knowledge workers through the process of linking an Open Directory topic to existing Cyc terms, and creating new Cyc terms when required. Our experience with Open Directory highlights the drawbacks of a semiautomatic mapping to a source ontology that is undergoing constant enhancement and refactoring. The Open Directory element IDs were not preserved when their editors refactored a category. Thus we faced an ever growing mapping maintenance burden that ultimately proved insurmountable. We have removed the mappings from Cyc until the customer need arises, and our ability allows fully automatic maintenance of the links.

WordNet

WordNet has become the standard lexical knowledge base with over 130,000 English words and phrases organized into taxonomies by parts of speech. The words are grouped into synonym sets (synsets) and assigned an ID. Like Open Directory, the synset ids are changed when new versions of the ontology are released, but a backwards compatibility utility program is provided to map synsets between versions. We mapped over 12,000 Cyc terms to WordNet version 1.6 and continue to support WordNet mapping via a graphical tool built into Cyc. Below are examples of mapping a WordNet noun synset, adverb synset, and verb synset:

```
(synonymousExternalConcept
  ShoppingMallBuilding WordNet-1997Version
  "N03144979")

(synonymousExternalConcept West-Generally
  WordNet-1997Version "R00318751")

(synonymousExternalConcept
  (TransportViaFn RoadVehicle)
  WordNet-1997Version "V01317106")
```

In the above assertion, the functional expression `(TransportViaFn RoadVehicle)` means the collection of transportation events in which a (more or less conventional) road vehicle is the transportation device. Cyc's functional term notation is frequently used when mapping source ontologies so that new concepts can be formed by composing existing concepts, rather than creating a new reified term for each one.

Simple Ontology Integration

We mapped Cyc relationships to the numerous but simple geographical, economic and governmental relationships found in the CIA World Factbook 1997 ontology. Then we imported over 5,900 facts into Cyc using these relationships. Here are three examples from what Cyc was told in 1998 – and still knows – about the country of Ireland:

```
(altitudeOfHighestPointIs Ireland-TheNation
  (Meter 1041))

(budgetExpenditureFractionOfGDP
  Ireland-TheNation
  GovernmentMilitaryOrganization
  (YearFn 1997)
  (Percent 1.3))

(lengthOfPathTypeInRegion Ireland-TheNation
  GasPipeline
  (Kilometer 225))
```

Single Namespace in Cyc

Cyc knowledge entry and ontology mapping methodology requires that terms exist within a single universal namespace. We distinguish between like-named words via a convention of term name suffixes as shown in the above example for `Ireland-TheNation` as contrasted with `Ireland-TheIsland`. To reiterate: the terms in Cyc

correspond to meaningful real-world *concepts;* they map to English words on a many-to-many basis, *not* 1-to-1.

DAML Ontology Import / Export

The DARPA Agent Markup Language (DAML) can be imported and exported from Cyc. We have created KB subsets in DAML format for various users and posted them at the http://www.daml.org web site. When importing DAML classes and instances, we expect only the meta classes and meta properties to initially map to Cyc, as DAMLs taxonomic properties are very similar to Cyc `isa` and `genl` subsumption predicates. When we imported the NAICS (North American Industry Classification System) and UNSPSC (Universal Standard Products and Services Classification), we used Cyc's phrase parser on the term names in an attempt to partially automate the term mapping. We found that a specialized workflow tool (at least as sophisticated as the Open Directory linking tool mentioned above) was highly cost-effective to develop and employ, to speed up the integration of the more than 10k terms of these two product & services ontologies with Cyc's.

XML schema mapping

Mapping ordinary XML schema to Cyc is naturally more difficult than mapping a DAML ontology, because the XML schema generally does not include taxonomic information for classes and properties. We expect, however, that tools that use semantic clues parsed from XML tag names will partially remedy this, and prove useful in both DAML and ordinary XML ontology mapping.

UML Ontology Integration

We are beginning to map the core elements of the Unified Modeling Language (UML) into Cyc. This enables Cyc to better understand computer system interfaces and their behavior when modeled with UML. Our work to date on this task suggests that UML may prove to be sufficiently expressive to serve as the reference Cyc ontology for system models, i.e. Cyc's vocabulary for system modeling will derive from UML.

Cyc Upper Ontology Enhancement

Our steady stream of work provides ongoing feedback on the Cyc ontology, causing revisions, combinations, and new terms to be defined at the lower, (more rarely) middle, and (even more rarely) upper levels.

One of the most recent examples of the latter, e.g., was the concept of an abstract container, an object that can encapsulate intangible things and model the containment behavior of UML packages and classifiers. That concept was necessary to introduce new Cyc collections (classes) and predicates (properties) generalized for commonsense mathematical modeling, and specialized for the straightforward mapping of UML instances.

Important new specializations of this new term will be:

```
MathematicalModel,
  (genls MathematicalThing and Individual)
MathematicalContainer,
  (like Container which is for tangible objects)
mathematicallyContains
  (like physciallyContains)
mathematicalParts (like parts)
ObjectOrientedComponent
  (genls ComputationalObject)
inheritObjectOrientedFeatures
  (somewhat like inheritIdentityToSubscenes)
```

Unlike most UML tools which hard-code (or ignore) the UML well-formedness constraints, we can directly map and represent these constraints as first order logical rules in Cyc. For example the first UML V1.4AS Core Well-Formedness rule for Association is: "The AssociationEnds must have a unique name within the Association.

UML OCL (Object Constraint Language) source rule:

```
self.allConnections →
    forAll( r1, r2 | r1.name = r2.name → r1 = r2)
```

This can be quite easily mapped into Cyc as:

```
(implies
  (and
    (connectionOf-UMLComponent ?R1 ?ASSOC)
    (connectionOf-UMLComponent ?R2 ?ASSOC)
    (equals
      (NameOf-UMLComponent ?R1)
      (NameOf-UMLComponent ?R2)))
  (equals ?R1 ?R2))
```

Argument type constraints on the predicate `connectionOf-UMLComponent` ensure that the first argument is a `AssociationEnd` and that the second argument is a `Association-UMLComponent`. So we omit the following redundant conjuncts from the rule's antecedent:

```
(isa ?ASSOC Association-UMLComponent)
(isa ?R1 AssociationEnd)
(isa ?R2 AssociationEnd)
```

Structured Text Ontology Integration

During a recent workshop for INSCOM (US Army Intelligence and Security Command), we mapped instances of structured text into Cyc. The source ontology was in a format similar to that established by the MUC (Message Understanding Conference), with named entities having relationships and descriptions. Here is an example of a source ontology instance, in which the named entity "North Africa" was extracted from a web document:

```
(LOCATION-null-1 ENTITY LOCATION
                 (NAME "North Africa")
                 (TYPE LOCATION)
                 (SUBTYPE REGION)
                 (COUNTRY "North Africa"))
```

Because the source ontology is semi-structured (having text elements as values), mapping the instances into Cyc requires NL phrase parsing. The Cyc lexicon contains an exact match for the phrase "North Africa". We mapped the instance to the Cyc assertions:

```
(isa NorthernAfrica GeographicalRegion)
(isa LOCATION-null-1 NorthernAfrica)
```

In another case, the source ontology instance was a named entity for which the `NAME` attribute was "car full of explosives". Our NL phrase parser was able to parse the concept Automobile, but not the complete meaning of the phrase. Well formed formula (wff) constraints, present in the reference Cyc ontology, prevent the mapping of semantically incorrect facts, discarding for example the mapping of the `NAME` "Algerian" to a geographical location. Cyc parses that phrase as AlgerianPerson, and Cyc has a high level assertion from which it can deduce that all instances of Person are disjoint from all instances of IndependentCountry. In the justification below, the *genls* predicate is Cyc's term for the subclass relationship.

```
"No person is a country."
(disjointWith Person IndependentCountry)

Justification :
(genls Person Primate)
(genls Primate Eutheria)
(genls Eutheria Mammal)
(genls Mammal Homeotherm)
(genls Homeotherm Animal)
(genls Animal EukaryoticOrganism)
(genls EukaryoticOrganism Organism-Whole)
(genls Organism-Whole
  BiologicalLivingObject)
(disjointWith BiologicalLivingObject
  Artifact-Generic)
(genls Organization Artifact-Generic)
(genls GeopoliticalEntity Organization)
(genls Country GeopoliticalEntity)
(genls IndependentCountry Country)
```

Our experience demonstrates that structured text in a source ontology can be utilized to parse facts, with the drawbacks that imprecision is introduced and that only a portion of the text meaning will be likely understood.

Structured Knowledge Source Integration

When mapping a structured knowledge source (SKS) such as a web service or relational database, both the structural schema and the access protocol are mapped into Cyc, thus enabling dynamic access and mapping from the source to satisfy a request. We recently added two mapping predicates to Cyc's ontology, to facilitate SKS integration: `synonymousListFields` and `overlappingListFields`

These predicates associate a Cyc term with a list of strings from the structured knowledge source that jointly map to the term. The latter predicate indicates that the meaning of the Cyc term overlaps the meaning of the source concept.

Knowledge Source Navigation / Access Protocol

Navigation and access protocols for SKSI are to be handled as cases: The most common are relational databases that are accessed via SQL. Many web services are lightweight interfaces using a relational database backend. Cyc uses its knowledge of the mapped DB schema to create `SELECT` statements or to populate the parameters of DB stored procedures, when satisfying requests.

Increasingly, web services have structured knowledge source schema based on XML (including DAML and RDF), and where the access protocol is SOAP or another web services standard. Building upon our experience with SQL, we intend to develop a generic driver program for web services whose behavior is governed in a declarative fashion by assertions in the Cyc KB.

The more complex case of navigation and access protocols arises from the need to interface software systems that have system specific APIs. We can model these protocols via agent conversation vocabulary, in which performatives model API function calls.

The Role of English Dialogue-Based Tools

The burden of inputting individual software system APIs into Cyc as modeled schema and access protocols, can be addressed by tools that carry on a clarification back-and-forth dialogue with a subject matter expert (SME) – e.g., a chemist or a tank commander – who does not need to have training in formal logic, ontologies, programming, etc. In other words, Natural Language parsing, understanding, and generation are employed to insulate the SME from having to read or write CycL (the predicate calculus representation language used throughout this paper, and by Cyc itself.)

Cyc now has such dialogue-based tools, written for the DARPA Rapid Knowledge Formation (RKF) project.[1] These tools have been used to guide SMEs through the acquisition of new domain knowledge, including terms, facts, rules, and scripts. Using mixed initiative, they ask the SME about salient properties of a new term. As an example of such "guiding", a SME says "TR1 can kill people", and Cyc asks (1) if TR1 can kill some broader category (primates, mammals, etc.), and (2) the manner in which the killing occurs (infection, wounding, etc.)

We plan to refactor the toolkit for the input of structured knowledge source schema and access protocol. RKF allows the testing of new knowledge as it is input. Similarly, RKF style SKSI tools will enable the testing of the source access protocol during the knowledge entry process, detecting errors at the earliest stage. Based on our experience during 2001 (RKF Year 1), we would expect dialogue-based tools for SKSI to work best at the fringe of what Cyc already knows. So we need to build up Cyc's knowledge of system structure and access protocols, which complements the mapping of UML models to Cyc.

Conclusion

Progressing through increasingly complex ontology mappings into Cyc, we find that the major barrier to adoption of ontology mapping for sophisticated web services will be the requirement for someone to input the source schema and access protocols. Our experience with the Open Directory linking tool, which incorporated some parsing of RDF tag names, shows that thousands of terms can be mapped into Cyc with a team of skilled knowledge workers. We need to combine the best features of this approach with RKF tools that enable SMEs, as opposed to skilled knowledge workers, to perform the mapping effort.

Further research and development is required to enhance Cyc's commonsense knowledge of system interfaces and behavior. Perhaps the Unified Modeling Language can serve as Cyc's reference ontology for system and service description, gaining leverage from an industry standard and numerous CASE tools that exchange system design information in the UML XMI format. The ontology DAML-S, whose instances describe web services, will certainly be integrated into Cyc's ontology during the DAML program.

[1] Some of our RKF tools, including some which were built by Ken Forbus at NWU, enable SMEs to describe new knowledge using analogy. Some still-in-development RKF tools include those with sketching tablet interfaces.

References

Central Intelligence Agency. 2001. *The CIA World Factbook*
URL: http://www.cia.gov/cia/publications/factbook/

Features of CycL. Cycorp
URL: http://www.cyc.com/cycl.html

Federal Information Processing Standards Publication 10-4. 1995. Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions.

Doerr, M. 2001. Semantic Problems of Thesaurus Merging. *Journal of Digital Information*. 1(8)

Fellbaum, C., editor. 1998. ***WordNet*** *An Electronic Lexical Database*. The MIT Press.

Knight, K and Luk, S. K. 1994. Building a LargeScale Knowledge Base for Machine Translation. *Proceedings of the American Association of Artificial Intelligence Conference*, Seattle, Washington: AAAI Press.

Lenat, D. B. 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. *The Communications of the ACM* 38(11):33-38

National Library of Medicine. 2002. Medical Subject Headings. URL:
http://www.nlm.nih.gov/pubs/factsheets/mesh.html

Pinto, H. S., Gomez-Perez, A., and Martins, J. P. 1999. Some issues on ontology integration. *Proceedings of the Workshop on Ontologies and Problem Solving Methods during*
IJCAI-99, Stockholm, Sweden.
URL: http://citeseer.nj.nec.com/pinto99some.html

Rumbaugh, J., Jacobson, I. and Booch, G. 1998. *The Unified Modeling Language Reference Manual*. Addison-Wesley

Wiederhold, G. 1994. An Algebra for Ontology Composition. *Proceedings of 1994 Monterey Workshop on Formal Methods, Sept 1994*, U.S. Naval Postgraduate School, Monterey CA :56-61. URL:
http://citeseer.nj.nec.com/wiederhold94algebra.html