

Upper Mapping and Binding Exchange Layer (UMBEL) Specification

UMBEL Specification Document - 20 April 2015

Latest version

http://techwiki.umbel.org/index.php/UMBEL_Specification

Last update

\$Date: 2015/4/20 14:28:36 \$

Version

Version No.: 1.20

Volume

TR 12-5-21

Editors

[Frédéric Giasson](#) - [Structured Dynamics](#)

[Michael Bergman](#) - [Structured Dynamics](#)

Authors

[Michael Bergman](#) - [Structured Dynamics](#)

[Frédéric Giasson](#) - [Structured Dynamics](#)



UMBEL: Upper Mapping and Binding Exchange Layer by [Structured Dynamics](#)

[LLC](#) is provided under the

[Creative Commons Attribution 3.0](#) license. See the [attribution section](#) for how to cite the effort.



Copyright © 2009-2015 by Structured Dynamics LLC.

Contents

- [1 ABSTRACT](#)
- [2 PRELIMINARY MATERIALS](#)
 - [2.1 Status of this Document](#)
 - [2.2 Download and Access](#)
 - [2.3 Versioning](#)
 - [2.4 Attribution](#)
 - [2.5 Summary of Changes](#)
- [3 OVERVIEW](#)
 - [3.1 Background and Rationale](#)
 - [3.2 Introduction to UMBEL](#)
 - [3.3 Key Design Aspects](#)
 - [3.3.1 Purpose](#)
 - [3.3.2 Use of OpenCyc](#)
 - [3.3.3 Separate Instances/Individuals from Concepts](#)
 - [3.3.4 Emphasize the Open-World Approach](#)
 - [3.3.5 Based on OWL 2 and Metamodeling](#)
 - [3.3.6 Extends SKOS Annotations and Structure](#)
 - [3.3.7 Organizes Reference Concepts into SuperTypes](#)
 - [3.4 Designed and Intended Uses](#)
 - [3.4.1 Mapping to Other Ontologies](#)
 - [3.4.2 Domain Ontologies](#)
 - [3.4.3 Inferencing](#)
 - [3.4.4 Semantic Tagging](#)
 - [3.4.5 Linked Data Mining](#)
 - [3.4.6 Creating a Computable Wikipedia](#)
 - [3.4.7 Categorizing Instances and Named Entities](#)
 - [3.5 Terminology Challenges](#)
 - [3.6 Best Practices](#)
- [4 PART I: VOCABULARY SPECIFICATION](#)
 - [4.1 Namespaces](#)
 - [4.2 Classes](#)
 - [4.2.1 Specifications](#)
 - [4.3 Properties](#)
 - [4.3.1 Use of Inverse Properties](#)
 - [4.3.2 Non-Normative Representation](#)
 - [4.3.2.1 UMBEL "Class" View](#)
 - [4.3.2.2 UMBEL "Individual" View](#)
 - [4.3.2.3 The "Punning" Relationship](#)
 - [4.3.2.4 External "Class" View](#)
 - [4.3.2.5 External "Individual" and Properties View](#)
 - [4.3.3 Specifications](#)
 - [4.4 Incorporation of External Properties](#)
 - [4.4.1 Listing of Properties](#)
 - [4.4.2 Language Annotations of Classes](#)
 - [4.5 Linking to Instances and Named Entities](#)
- [5 PART II: REFERENCE CONCEPTS SPECIFICATION](#)
 - [5.1 Reference Concepts](#)

- [5.1.1 Basis in OpenCyc](#)
 - [5.1.2 Structural Predicates](#)
 - [5.1.3 Metamodeling and Punning](#)
 - [5.1.4 Use of OpenCyc External IDs](#)
 - [5.1.5 Modularization](#)
 - [5.1.6 Not Determinative for Separate UMBEL Domain Ontologies](#)
- [5.2 SuperTypes](#)
 - [5.2.1 Non-disjoint SuperTypes](#)
 - [5.2.2 Disjoint SuperTypes](#)
 - [5.2.3 SuperType Dimensions](#)
 - [5.2.4 Combination \(Overlapping\) SuperTypes](#)
- [5.3 Linking Predicates](#)
 - [5.3.1 External Mappings](#)
- [5.4 Instances and Named Entities](#)
- [6 LINKING AND MAPPING PREDICATES](#)
 - [6.1 A Comparison of Options](#)
 - [6.2 Narrative Discussion](#)
 - [6.2.1 Equivalent Properties](#)
 - [6.2.1.1 owl:equivalentClass](#)
 - [6.2.1.2 owl:sameAs](#)
 - [6.2.2 Membership Properties](#)
 - [6.2.2.1 rdfs:subClassOf](#)
 - [6.2.2.2 rdf:type](#)
 - [6.2.3 Approximation Properties](#)
 - [6.2.3.1 umbel:correspondsTo](#)
 - [6.2.3.2 umbel:isAbout](#)
 - [6.2.3.3 umbel:isLike](#)
 - [6.2.3.4 umbel:relatesToXXX](#)
 - [6.2.4 Descriptive Properties](#)
 - [6.2.4.1 umbel:isCharacteristicOf](#)
 - [6.3 Qualifying the Mappings](#)
- [7 EXAMPLES](#)
 - [7.1 Reference Concept Description](#)
 - [7.2 Named Entity Instantiation](#)
 - [7.3 Linkage Between External Classes and Reference Concepts](#)
 - [7.4 Linkage Between Named Entities](#)
- [8 CHANGELOG](#)
 - [8.1 Running Changelog](#)
 - [8.1.1 Version 1.20 Changes \(current\)](#)
 - [8.1.2 Version 1.10 Changes](#)
 - [8.1.3 Version 1.05 Changes](#)
 - [8.1.4 Version 1.00 Changes](#)
 - [8.1.5 Summary of Version 0.80 Changes](#)
- [9 ANNEXES](#)
- [10 ENDNOTES](#)

ABSTRACT

UMBEL (*Upper Mapping and Binding Exchange Layer*) has two purposes. Its first purpose is to provide a general vocabulary (the UMBEL "vocabulary") of classes and predicates for describing domain ontologies, with the specific aim of promoting interoperability with external datasets and domains. The second purpose is to provide a coherent framework of broad subjects and topics (the UMBEL "reference concepts"), suitable as binding nodes for grounding relevant Web-accessible content, also with the specific aim of promoting interoperability and to reason over a coherent reference structure and its linked resources. UMBEL presently has about 35,000 of these reference concepts drawn from the Cyc knowledge base, split into 'core' and a series of optional modules, which are organized into more than 30 mostly disjoint SuperTypes.

This document provides the specifications to fulfill these twin purposes. The UMBEL vocabulary and reference concepts ontology are OWL 2-compliant.

PRELIMINARY MATERIALS

Status of this Document

NOTE: *This section describes the status of this document at the time of its publication. Other documents may supersede this document.*

A full copy of these UMBEL specifications may be [downloaded in PDF from here](#).

UMBEL was first publicly released as version 0.70 on July 16, 2008. The last public release was version 1.10 on September 19, 2014. Original background information and links to early details are recorded in the accompanying UMBEL historical documentation in [Annex B](#).

This document may be updated or added to based on implementation experience, but no commitment is made by the authors regarding future updates.

Download and Access

See <https://github.com/structuredynamics/UMBEL/tree> for how to download the UMBEL Vocabulary or the UMBEL Reference Concept ontology, both under a Creative Commons Attribution 3.0 license. Other documents and backup are also available from that location.

Reference and binding to UMBEL Reference Concepts occurs through dereferencable URIs at, for example, <http://umbel.org/umbel/rc/ExactConceptName>. See further *UMBEL Ontology, Vol. 2: Subject Concepts and Named Entities Instantiation*, [TR 08-07-16-A2](#), for historical details regarding access and use.

Versioning

This is **version 1.20** of the UMBEL specification. The last published version was 1.10.

UMBEL was first conceived in April 2007. Initial versions of UMBEL were in draft and principally involved selecting and cleaning reference concepts from OpenCyc (see [below](#)). Sufficient iterations occurred such that the first public release of UMBEL, in July 2008, was version 0.70. The explanation of version numbering and rationale is provided in the historical volume, *Distilling Subject Concepts from OpenCyc, Vol. 1: Overview and Methodology*, [TR 08-07-16-B1](#).

Version 1.00 signaled release of a production-grade system. The Vocabulary is largely stable with regard to concepts, but is still under active development in the attributes area. The scope of Reference Concepts and its ontology is also newly in flux given the role UMBEL is assuming as an organizing framework for Wikipedia and via that as a bridge to OpenCyc (see below). We anticipate future refinements in the scope and number of RCs to perform these roles, in addition to changes resulting from continued mapping of UMBEL to external ontologies, schema and knowledge bases.

Attribution

We are pleased to license the UMBEL ontology under terms that explicitly encourage you to take, modify, reuse, re-purpose, and remix our work as you see fit. The content of UMBEL and its documentation are licensed under the [Creative Commons Attribution 3.0 License](#).

Proper attribution is required when you use UMBEL or its documentation, whether without change or when you reuse or create modified versions of it, according to the terms of this Creative Commons Attribution license. The complete requirements for attribution can be found in section 4b of the [Creative Commons legal code](#).

If you use UMBEL or its documentation, as is or with modifications, we ask that you provide attribution with Web links to the best of your ability for your given medium as follows:

UMBEL and its documentation is the creative work of [Structured Dynamics LLC](#), which grants free use rights thereto, only limited by the attribution terms described in the [Creative Commons 3.0 Attribution License](#). The copyrights to UMBEL and its documentation remain the sole right of Structured Dynamics LLC.

If you have questions or suggestions regarding UMBEL or these license policies, please email us at umbel@structuredynamics.com.

Summary of Changes

These are the principal changes between the last public release, version 1.10, and this version 1.20 (see further the [Annex L](#) discussion for more details on this version update). In summary, those changes include:

- Expanded mappings to OpenCyc to better capture coverage of Wikipedia content; there are now 35,533 reference concepts (RCs) in UMBEL, 35,302 of which are mapped to OpenCyc (the unmapped RCs are mostly used for organizational purposes in the Attributes Ontology – see below – and OpenCyc mismatches with key external ontologies)
- Created a new *Entities* SuperType, with 20,393 RCs designated. The Entities ST is by definition non-disjoint with the other SuperTypes
- Created a new Entities module, with 9,317 RCs assigned; the remainder of the Entities RCs are in core
- Created a new Attributes Ontology (AO), with the purpose of enabling property (attribute) mappings to UMBEL
- Created a new Attributes module, with 1,002 RCs assigned
- Expanded the direct UMBEL RC to Wikipedia page mappings, with 25,582 currently mapped, or nearly three-quarters (72%) of RCs now assigned
- Created a new [Annex Z](#) to hold updated statistics about UMBEL
- Deprecated the *Workplaces* SuperType, and merged with the *Facilities* ST
- Deprecated the *MarketIndustries* SuperType, and merged with the *Attributes* ST
- Reviewed and greatly improved ST assignments across the board; notably, the distinction between the *Events* and *Activities* SuperTypes was improved. See [Annex Z](#) for the updated ST assignment statistics
- Greatly expanded and improved the UMBEL generator to handle satisfiability tests and modules creation
- Expanded and updated the [UMBEL.org](#) Web site.

Note, for legacy reasons, you may still encounter reference to 'subject concepts'. Please consider that term as interchangeable with the current 'reference concepts'.

Details of changes for prior versions may be found under the [Changelog](#) section at the conclusion of these specifications.

OVERVIEW

This section provides the background, rationale and use cases for **UMBEL**, the *Upper Mapping and Binding Exchange Layer*. UMBEL's project Web site is at <http://www.umbel.org>.

This Overview introduces the two-part formal specification that follows. After the specifications, some examples are presented. The material herein is supported by a set of [annex documents](#) that provide additional methodology and use

guidance.

Background and Rationale

The Web and enterprises in general are characterized by growing, diverse and distributed information sources and data. Some of this information resides in structured databases; some resides in schema, standards, metadata, specifications and semi-structured sources; and some resides in general text or media where the content meaning is buried in unstructured form. The premise of the *semantic Web* and the *semantic enterprise*, respectively,^[1] is to provide the languages and methods by which this information can be represented, combined and made meaningful. Making heterogeneous information interoperable -- no matter the format or provenance -- is the promise offered by semantic technologies and approaches.

Yet all of this information in its diverse forms and its disparate sources has been developed for different contexts and purposes. Different world views have underpinned its creation and different needs and uses govern how it might be combined and made interoperable. Making parts of this information interoperable resides in aligning contexts and the semantics of the information.

By definition, alignments between different sources and perspectives will rarely be exact. Data heterogeneity from the structural to the conceptual govern the diversity of information.

Thus, given these huge amounts of information, how can one bring together what subsets are relevant? And, then for candidate material that does appear relevant, how can it be usefully combined or related given its diversity? In short, how does one go about actually combining diverse information to make it interoperable and coherent?

Introduction to UMBEL

UMBEL (*Upper Mapping and Binding Exchange Layer*) does not alone solve the interoperability problem. Rather, it is designed as a useful set of first steps on the path to doing so.

The first step and purpose of UMBEL is to put forward some predicates -- connecting verbs or properties -- for linking disparate information sources together. This purpose, the [UMBEL Vocabulary](#), is designed to recognize that different sources of information have different contexts and different structures. A meaningful vocabulary is necessary that can express potential relationships between two information sources with respect to their differences in structure and scope. By nature, these connections are not always exact. Means for expressing the "approximateness" of relationships are essential.

The second step and purpose of UMBEL is to provide a fixed set of concepts -- the [UMBEL Reference Concepts](#) -- by which these approximate alignments can be oriented. By design, this set of fixed reference points is not exact nor comprehensive. These reference concepts are not meant to model the world in all of its complexity and nuance. Rather, our goal is to provide a set of fixed references by which we can orient and navigate constituent content. The operative idea here is "aboutness". Can we describe our constituent information in terms of what it is about such that we can begin to gather similar relevant content together?

To achieve this second purpose we must start with an embracing and coherent world view. Then when we combine that with the UMBEL Vocabulary, we can create a reference ontology. This reference structure can then provide initial nodes for mapping relevant Web-accessible content. If shared and widely used, these initial reference concepts can provide a powerful set of groundings for guiding related datasets and content to be discovered.^[2]

For reasons summarized below, the coherent set of UMBEL reference concepts began with the Cyc knowledge base. However, since its scope and sophistication far exceeded what was tractable for a lightweight reference structure, we pruned and cleaned Cyc to a significant degree. All of the UMBEL Reference Concepts and their relationships are derived from the OpenCyc ontology. This means that UMBEL is a clean, 100% subset of OpenCyc. The result is an UMBEL reference structure of about 35,000 concepts, broadly applicable as orienting nodes to any knowledge domain, all coherently structured and linked to one another. This winnowing produced the lightweight UMBEL Reference Concept ontology.

The UMBEL Reference Concept ontology is, in essence, a content graph of subject nodes related to one another via *broader-than* and *narrower-than* relations. In turn, these internal UMBEL Reference Concepts may be related to external classes and individuals (instances and named entities) via a set of relational, equivalent, or alignment predicates. This UMBEL Vocabulary is itself a solid basis for constructing domain ontologies that can also act as reference ontologies within their own domains. You can see a [big graph visualization](#) of this structure on the official [UMBEL Web site](#).

Three years after conception, and two years after first public release, we released version 0.80 in November 2010 after a full year applying and refining the UMBEL framework. Significant use cases were tested, notably with FactForge, and a new development partnership with Ontotext AD emerged.^[3] With version 0.80, the UMBEL reference ontology was better organized and made easier to browse via the addition of 33 new SuperType classes (now 32) clustered into nine dimensions. Many early vocabulary decisions were revised, and substantial improvements across the board were made in terms of structure and documentation. Version 0.80 was also the first fully OWL 2 compliant release.^[4]

By the time of the release of version 1.00 in February 2011, the system had grown to production-grade status. That version was mapped to 60% of Wikipedia and all of the PROTON upper ontology; its size increased by one-third in order that it might fulfill its promise to become a [gold standard](#) for reference linking and interoperability with knowledge bases. Further predicates and refinements were added to the UMBEL Vocabulary to aid this role.

In version 1.05 we began to modularize parts of UMBEL, splitting the Reference Concept ontology into 'core' and 'geo' modules. We also instituted direct mappings to schema.org^[5], the DBpedia ontology^[6] and the GeoNames knowledge base.^[7]

As the size and complexity of UMBEL increased, with more moving parts, it became evident that automated build and testing routines were needed. By version 1.10, a new [UMBEL generator](#) was developed in Clojure that now allows the entire system to be built from a series of simple input files.

By this version, 1.20, the generator is being further refined as we modularize further into 'core', 'geo', 'attributes' and 'entities' modules. Mapping to Wikipedia continued, with now 72% mapped, and an Attributes Ontology was introduced that is

intended to provide a grounding structure for attribute properties, similar to UMBEL's original purpose in providing groundings for concepts.

Key Design Aspects

In pursuing these dual purposes of a linking Vocabulary and a Reference Concept ontology, some key design decisions were made.

Purpose

Vocabularies and ontologies can serve virtually any perspective or any purpose. A focused perspective on inter-linking diverse domains and datasets leads to these clear purposes for UMBEL:

1. A lightweight vocabulary geared to a range of mappings from exact to approximate for diverse content
2. A scaffold of reference concepts sufficient to provide linkage points for any domain and simple inferencing and structure
3. By virtue of the two prior purposes, a re-usable vocabulary and lightweight structural design that can be replicated to provide the same reference benefits for specific domains with their specific terminology,
4. To provide a bridge between the content-rich Wikipedia knowledge base and the logical and computable OpenCyc ontology.

Use of OpenCyc

A central role of the UMBEL reference concepts is to provide **contexts** for relating information. Once the role of context is embraced, the natural next question is: And what shall be the basis for this context?

Historical UMBEL project documentation describes alternatives and why Cyc^[8], and its open source version, OpenCyc, was chosen as this contextual basis.^[9] In essence, the major reasons for this choice were:

- *Venerable and solid* — through an estimated 1000 person years of engineering and effort over more than 20 years, the Cyc structure has been tested and refined through many projects and applications
- *Community* — there is a large community of Cyc users and supporters from academic, government, commercial and non-profit realms
- *Comprehensive* — no existing system has the scope, breadth and coverage of human concepts to match that of Cyc (however, Wikipedia now exceeds Cyc as a source of reference information on instances and individuals)
- *Common sense* — Cyc has set out to capture the common sense at the heart of human reasoning. This objective means codifying generally unstated logic and rules-of-thumb that leads to a solid basis for its reasoning and conceptual relationships
- *Power and inference* — Cyc has about a thousand microtheories governing its inference domains, giving it a contextual scope and power unmatched by other systems
- *Broad functionality* — its knowledge base capabilities can be deeply leveraged in such areas as entity extraction, machine translation, natural language processing, risk analysis or one of the other dozens of specialty modules
- *Free and open* — OpenCyc is a free and open source version that has been downloaded more than 100,000 times, and
- *Upgrade path* — OpenCyc has an upgrade path to the more capable [ResearchCyc](#), full Cyc and the services of [Cycorp](#).

These reasons made Cyc the appropriate choice for UMBEL. But perhaps a more important imperative is to choose an inherently coherent framework. There are alternative ontologies and knowledge bases that could also provide coherent structures. As noted, multiple reference structures, each linked to the other, is ultimately a desirable goal so as to provide alternative perspectives and choice.

One drawback to Cyc is also one of its strengths: its sheer size and sophistication. Cyc's native scope is too great for easy comprehension and linkage for standard Web purposes. Thus, in its first year, the UMBEL project set out to determine and derive the most fundamental concepts from within OpenCyc. What was desired was a tractable set of reference nodes while maintaining 100% consistency with OpenCyc. This consistency preserves the linkage into the Cyc knowledge base.

As a result the overall size and complexity of Cyc was reduced by one to two orders of magnitude. The lightweight UMBEL structure now contains about 5-10% of the original size of Cyc, or about 35,000 Reference Concepts and their interrelations.^[10]

Because each UMBEL Reference Concept has a direct correspondence to a parallel concept in Cyc, it is possible to leverage additional reasoners and the Cyc knowledge base. Thus, through OpenCyc, and then via licensing or other arrangements into ResearchCyc or the full Cyc, another dimension of tools and capabilities can become available for specialized needs.

Separate Instances/Individuals from Concepts

A reference ontology is not a knowledge base. To act as a reference structure of concepts and their relationships, we need not worry about the specific things and individuals that populate a conceptual schema. With UMBEL, we have been able to focus on the schema alone. This clarity of focus naturally leads into the realm of description logics.

Description logics and their semantics traditionally split concepts and their relationships from the different treatment of instances and their attributes and roles, expressed as fact assertions. The concept split is known as the **TBox** (for *terminological* knowledge, the basis for **T** in TBox) and represents the schema or taxonomy of the domain at hand. The TBox is the structural and intensional component of conceptual relationships. The second split of instances is known as the **ABox** (for *assertions*, the basis for **A** in ABox) and describes the attributes of instances (and individuals), the roles between instances, and other assertions about instances regarding their class membership with the TBox concepts.

The TBox portion, or classes (concepts), is the basis of ontologies. Ontologies establish the structure used for governing the conceptual relationships for that domain and in reference to external (Web) ontologies. The ABox portion, or instances

(named entities), represents the specific, individual things that are the members of those classes. Named entities are the notable objects, persons, places, events, organizations and things of the world. Each named entity is related to one or more classes (concepts) to which it is a member. Named entities do not set the structure of the domain, but populate that structure. The ABox and TBox play different roles in the use and organization of the information and structure.^[11]

By definition, then, a knowledge base (as is Cyc or Wikipedia), is the combination of the TBox plus the ABox.^[12] A reference ontology, such as UMBEL, need only focus on the TBox. (Of course, a reference ontology such as UMBEL *could* act as a knowledge base by including instances and related assertions, but that is outside our express purposes herein.)

One could argue (we do) that a strict purpose as a **reference ontology** leads to a simpler and cleaner design for UMBEL than a knowledge base perspective would allow. However, our most recent version 1.00 efforts of mapping UMBEL to Wikipedia helped test and improve the robustness of the UMBEL structure to fulfill its TBox role. We now have real-world validation of the split in UMBEL's design.

Emphasize the Open-World Approach

The circumstance of bringing diverse, heterogeneous information together based on different contexts and world views requires an open world approach. The open world assumption is a logic basis that assumes that what is not stated is not necessarily false, it is simply not known. The open world assumption is a different logic premise than what many enterprises are used to; relational database systems, for example, embrace the alternate closed world premise.^[13]

The traditional relational model works best in well-characterized or self-contained circumstances, such as seats on a plane, books in a library, customers of a company, products sold via distribution channels, etc. A closed-world assumption enables performant transaction operations with easier data validation, since the number of negative facts about a bounded domain is typically much greater than the number of the positive ones. In such cases, it is simpler and shorter to state known "true" statements than to enumerate all "false" conditions.

However, the relational model is a paradigm where the information must be complete and it must be described by a single schema. Traditional databases require an agreement on a schema, which must be made before data can be stored and queried. The relational model assumes that the only objects and relationships that exist in the domain are those that are explicitly represented in the database, and that names uniquely identify objects in this domain. The result of these assumptions is that there is a single (canonical) model for relational systems where objects and relationships are in a one-to-one correspondence with the data in the database.

However, the purposes for UMBEL reside in the broad category of knowledge management. KM includes such applications as search, data federation, data warehousing, enterprise information integration, business intelligence, competitive intelligence, knowledge representation, and so forth. For these applications, the open world approach provides some incredibly important benefits:

- Domains can be analyzed and inspected incrementally
- Schema can be incomplete and developed and refined incrementally
- The data and the structures can be used and expressed in a piecemeal or incomplete manner
- We can readily combine data with partial characterizations with other data having complete characterizations
- Systems built with open world frameworks are flexible and robust; as new information or structure is gained, it can be incorporated without negating the information already resident, and
- Open world systems can readily bridge or embrace closed world subsystems.

Open world does not necessarily mean open data and it does not necessarily mean open source. Open world is merely a way to think about the information we have and how we act on it. An open world assumption accepts that we never have all necessary information and lacking that information does not itself lead to any conclusions.^[14]

The languages used and the design of UMBEL are based on the open world approach. Saying that a concept in an external ontology or information source "is about" a given reference concept in UMBEL does not define or limit what can be said about that external concept in other assertions.

Based on OWL 2 and Metamodeling

It is not unusual to want to treat things either as a class or an instance in an [ontology](#), depending on context. For example, classes and sub-class relationships are natural ways to describe hierarchical and related structure amongst domain concepts. At the same time, we may also want to describe the characteristics of the concepts in this structure as an instance, without the entailments that a class-to-class structure may bring. For example, UMBEL Reference Concepts are both classes (`owl:Class`) and instances of the class `umbel:RefConcept`, and may also sometimes be instances of other Reference Concept classes.

Making statements (asserting properties) in one context as a class and another context as an instance caused OWL version 1 ontologies to be treated as OWL Full. (This was also generally true for most RDF schema.) Reasoning could still be done over OWL Full, but the inherent difficulty is that the ontologies were not decidable. On one practical level, that is not terrible; simple inferencing and other consistency checks could still be reasoned. But on another practical level, such implicit "metamodeling" tends to hide purpose and precision since context of usage is not always clear.^[15]

OWL 2 provides some new capabilities to metamodel this dual class-instance circumstance more explicitly through "punning", which also can make the ontology decidable.^[16] (That is, compliant with OWL DL.) Other sub-languages ("profiles") of OWL 2 (such as OWL RL) are also decidable.

These improvements add reasoning and inferencing power to OWL 2 ontologies. There are also other advantages in annotations and some other areas. As a result, since version 0.80 the UMBEL Vocabulary and its Reference Concept Structure has been OWL 2-compliant. "Punning" is used to represent Reference Concepts as either classes or instances depending on context. See further the discussion of Figure 1 in Part 1 below.

Extends SKOS Annotations and Structure

SKOS (*Simple Knowledge Organization System*)^[17] ontologies have both strengths and weaknesses as applied to the

UMBEL Reference Concept ontology structure. The strength of SKOS ontologies is that they are explicitly designed for concept schema and knowledge structures, to which the UMBEL reference ontology naturally belongs. A particular expression of this strength are the many and relevant annotation properties within SKOS.

On the other hand, SKOS has no available reasoners, has consistency issues with OWL 1 (OWL 2 is better adapted with some changes^[18]), and has not had its specifications updated since the official release of OWL 2. There remain some minor property definition discrepancies.

As a result, these UMBEL specifications more fully adopt OWL 2 considerations than SKOS. Though many SKOS properties are still recommended and used, they are mostly in the areas of annotation, which are easily accommodated in OWL 2.

The continued use of SKOS predicates from prior UMBEL versions, but only in relation to OWL 2 compliance and adjustments, reflects our belief that SKOS will continue to be a leading concept and terminology vocabulary.

Organizes Reference Concepts into SuperTypes

Shortly after the first public release of UMBEL it was apparent that its reference concepts tended to "cluster" into some natural groupings. Further, upon closer investigation, it was also apparent that most of these concepts were disjoint with one another. As subsequent analysis showed, more fully detailed in the [Annex G](#) document, fully 75% of the reference concepts in the UMBEL ontology are disjoint with one another.

Natural clusters provide a tractable way to access and manage some 35,000 items. And, large degrees of disjointness between concepts also can lead to reasoning benefits and faster processing and selection of those items.

For these reasons a dedicated analysis to analyze and assign all UMBEL reference concepts to a new class of *SuperTypes* was undertaken. *SuperTypes* became a major enhancement to UMBEL in version 0.80. The assignment results and the *SuperType* specification are discussed in [Part II](#).

Designed and Intended Uses

These purposes and design decisions occur within the anticipated and designed uses for UMBEL. Besides the sections below, additional information may be found in [Annex E: Using UMBEL](#).

Mapping to Other Ontologies

The central objective of both the UMBEL vocabulary and the UMBEL reference ontology is to support the mapping of disparate and heterogeneous datasets and ontologies, again relying on an open world approach. Since mapping plays such a central role, a separate document, [Annex F: Mapping with UMBEL](#), covers this topic.

Domain Ontologies

The design pattern of reference concepts with a simple vocabulary that emphasizes "aboutness" and approximate (as well as exact) mappings is also well suited to specific domain ontologies. Using the UMBEL Reference Concept ontology as a template, and employing the UMBEL Vocabulary, a similar reference structure may be constructed for virtually any domain.

Inferencing

The UMBEL reference concept structure is designed for simple inferencing, as more fully described in the [Annex D: Inferencing with UMBEL](#). One example in that document describes how, say, relating via the UMBEL reference concepts classes structure enables the use of the properties `event:product`, `event:factor` and `event:time`, defined in the Event Ontology^[19], to be used in relation to a `foaf:Project`.

Within the UMBEL ontology itself, simple inferencing also enables better semantic search and look-ups via the various hierarchical predicates.

Semantic Tagging

The UMBEL Reference Concept ontology, or similar domain ontologies also built from the UMBEL Vocabulary, can be used as input bases to ontology-based information extraction (OBIE).^[20] Via this method, an information or extraction utility (tagger) is supplied a base ontology, which is then used to tag text or documents.

Some of the best practices designs for UMBEL (see [Annex C](#)) advocate the use of "*semsets*", which are robust specifications of preferred and alternative labels (along with misspelled variants) as matching terms and phrases to be used by such taggers. It is also possible to combine a general reference ontology, such as UMBEL, with specific domain ontologies for tagging content.

Then, via separate utilities, this tagged content may then be injected back into Web pages (via RDFa or microdata) or written to datastores directly as metadata.

Linked Data Mining

Linked data mining refers to discovering and linking relevant or similar data across the heterogeneous Web (or internal intranets). Mining effectiveness is presently limited by the lack of linkage assertions that are made between datasets.

The combination of the UMBEL Reference Concept ontology and semantic tagging (see above) can greatly improve this situation. Linked data sets and their content can be processed via semantic tagging. The resulting reference concept identifications can then be related to the source datasets. Via the reference ontology, direct and related concepts may then be retrieved and mined. This same approach may also be used for mining instance data (see next).

Creating a Computable Wikipedia

If organized properly with the right scope, and with complete mappings to key Wikipedia articles, it is possible to replace the Wikipedia category structure with UMBEL RCs. This structuring would make Wikipedia a computable knowledge source, with immense benefits in artificial intelligence and further improvements. As UMBEL matures, this particular intended use is increasingly guiding next UMBEL refinements.

Categorizing Instances and Named Entities

Tremendous resources ranging from Wikipedia and Flickr to many large datastores on the Web (or on internal intranets) are valuable sources of instance and entity data. However, the biggest problem in retrieval and aggregation for these sources is a lack of consistency and coherence in how their data are organized.

Some of these sources have their own category systems; some have folksonomies or user tags; but all lack a common basis for relating similar content.

Semantic tagging with reference ontologies provides a means for asserting a common organizational framework for these sources. Over time, such linked entity "dictionaries" may become one of the most valuable uses of reference ontologies.

Terminology Challenges

The sources and uses of UMBEL bridge across a variety of logic structures, often with different terminology. On major structural bases, here are high-level comparisons across some of these:

| UMBEL | RDF/OWL | Cyc | Frame Logic |
|--|------------|---------------------------|------------------|
| Reference Concept Subject Concept Class | Class | Collection | Class |
| Named Entity Individual | Individual | Individual | Individual-Thing |
| Instance | Individual | Instance | Instance |
| Property Predicate Relation Attribute | Property | Relationship Predicate | Relation |

Table 1. Comparison of Structural Terminology

The use of "individual" in standard semantic Web languages (RDF and OWL) can be particularly confusing. Rather than actual individuals -- that is, something which is itself not a collection of members and can be singly identified and named -- an "individual" in an RDF/OWL sense may also refer to an instance or a member of a class (or set). As the table shows, other terminology issues may also arise in other categories.

From the standpoint of this UMBEL specification, we prefer to use "instance" when referring to class membership, and reserve individual or "named entity" to refer to the definition above. (And, of course, an "instance" in this sense can also be an individual if it meets the definition.) Further, as the design discussion above indicated, the UMBEL Reference Concept ontology tends to focus solely on concepts (as classes), even though in a "punning" sense with regard to metamodeling the referent might be treated as an instance depending on context.

Where terminology may be somewhat different or confusing with regard to external uses, please refer to the UMBEL column in Table 1 above for the preferred use in these specifications.

Best Practices

Please refer to the separate section on best practices, [Annex C: Best Practices using UMBEL](#).

PART I: VOCABULARY SPECIFICATION

The first purpose of UMBEL is to provide a general vocabulary (the *UMBEL Vocabulary*) of classes and predicates for describing domain ontologies, with the specific aim of promoting interoperability with external datasets and domains. This document provides the specifications for this UMBEL Vocabulary.

The UMBEL Vocabulary defines three classes: `RefConcept`, `SuperType`, and `Qualifier`; and 38 properties: `correspondsTo`, `isAbout`, `isRelatedTo`, `relatesToXXX` (31 variants), `isLike`, `hasMapping`, `hasCharacteristic` and `isCharacteristicOf`. In addition, UMBEL re-uses certain properties from external vocabularies. These classes and properties are used to instantiate the UMBEL Reference Concept ontology (see the **UMBEL Specification**), and to link Reference Concepts to external ontology classes. Below we describe each of these classes and properties.

Namespaces

Here are the URIs of the namespaces used to describe the UMBEL Vocabulary, the Reference Concepts ontology, the named entities defined in UMBEL and the `SuperTypes` for both the Reference Concept classes and named entities.

The folder structure of these classes of URIs has been generalized to meet the design goals of using UMBEL with domain extensions. The portion `"/umbel/"` in the URIs is a placeholder for the name of these extensions. Each extension, including UMBEL itself, will share the same identification structure. An example for a 'Foo' domain ontology at an alternative example.com domain using the `"/foo/"` folder extension is shown in the table below.

The UMBEL Vocabulary URI uses a "[hash URI](#)" for convenience purposes. This facilitates the retrieval of the document of the

descriptions of the vocabulary for tools that consume such documents. However considering the size of the reference concepts descriptions files and related named entities files, we choose to use "[slash URLs](#)" so that consumer tools do not have to download the description of everything when they only want descriptions of one of these resources.

| Name | Abbreviation | URI |
|---|--------------|---|
| UMBEL Ontology | umbel: | http://umbel.org/umbel# |
| Reference Concepts | rc: | http://umbel.org/umbel/rc/ |
| Named Entities | ne: | http://umbel.org/umbel/ne/ |
| FOO Ontology (a domain ontology based on UMBEL) | foo: | http://example.com/foo# |

Table 2. UMBEL Namespace

Classes

The UMBEL vocabulary contains three classes, `RefConcept`, `SuperType` and `Qualifier`.

Specifications

Here are the specifications for the three UMBEL classes in the vocabulary.

| | |
|---------------------|--|
| Class name | umbel:RefConcept |
| Description | Reference Concepts are a distinct subset of the more broadly understood concept such as used in the SKOS RDFS controlled vocabulary or formal concept analysis or the very general or abstract concepts common to some upper ontologies. Reference Concepts are selected for their use as concrete, subject-related or commonly used notions for describing tangible ideas and referents in human experience and language. Reference Concepts are classes, the members of which are nameable instances or named entities, which by definition are held as distinct from these concepts. The UMBEL ontology is a coherently organized structure (or reference "backbone") of these Reference Concepts. |
| In-domain-of | umbel:isRelatedTo, skos:prefLabel, skos:altLabel, skos:hiddenLabel, skos:definition |
| In-range-of | umbel:isAbout, umbel:correspondsTo |
| Sub-class-of | skos:Concept |
| Status | Stable |

Table 3. Reference Concept Class

| | |
|---------------------|---|
| Class name | umbel:SuperType |
| Description | SuperTypes are a collection of (mostly) similar Reference Concepts. Most of the SuperType classes have been designed to be (mostly) disjoint from the other SuperType classes. SuperTypes thus provide a higher-level of clustering and organization of Reference Concepts for use in user interfaces and for reasoning purposes. |
| In-domain-of | skos:prefLabel, skos:altLabel, skos:hiddenLabel, skos:definition |
| Status | Stable |

Table 4. SuperType Class

| | |
|--------------------|--|
| Class name | umbel:Qualifier |
| Description | The Qualifier class is a set of descriptions that indicate the method used in order to establish an <code>isAbout</code> or <code>correspondsTo</code> relationship between an UMBEL reference concept (RC) and an external entity. This description should be complete enough to aid understanding of the nature and reliability of the "aboutness" assertion and to be usable for filtering or user interface information. The descriptions may be literal strings or may refer to literal numeric values resulting from an automated alignment technique. |
| In-range-of | umbel:hasMapping |
| Status | Stable |

Table 5. Qualifier Class

Properties

The UMBEL predicates, and those used from external vocabularies, provide the mapping basis for the vocabulary. Predicates (properties) are provided for all structural combinations of classes, individuals and properties.

Use of Inverse Properties

Ontologies based on the UMBEL vocabulary have the purpose of being available for context reference by external Web data. In this regard, ontologies based on the UMBEL vocabulary are largely passive. Other external systems generally refer to them, rather than embedding those mappings within the UMBEL-based ontology (though that design is legal, as well).

However, mapping may be done either from the perspective of the local domain ontology based on the UMBEL vocabulary or from the perspective of an external ontology aligning to an UMBEL-based reference ontology. In the latter case, a tagger might be run so as to map to the reference ontology.

Attention has thus been given within the UMBEL vocabulary to provide appropriate inverse properties. These inverse properties of `isRelatedTo` and `isCharacteristicOf`, in combination with the OWL equivalency predicates of `sameAs` and `equivalentClass`, provide the mechanisms whereby external references can be traced back from UMBEL to find additional corresponding external data.

These capabilities are readily demonstrated via some of the UMBEL Web services (including taggers) discussed in [Annex E: Using UMBEL](#).

Non-Normative Representation

Figure 1 below describes the available object properties within the UMBEL vocabulary (maroon color) and those re-used from external ontologies (blue color) for mapping and structural relations. Also shown is the twin views of UMBEL concepts as both classes (`RefConcept`) and individuals, achieved via 'punning' in OWL 2 (c.f., [OWL 2 and Metamodeling](#)).

The figure is presented as a quadrant. On the left-hand side (quadrants **A** and **C**) is the "class" view of the structure; the right-hand side is the "individual" (or instance) view of the structure (quadrants **B** and **D**). These two views represent alternative perspectives for looking at the UMBEL reference concepts based on metamodeling.

The top side of the diagram (quadrants **A** and **B**) is an internal view of UMBEL reference concepts (`RefConcept`) and their predicates (properties). This internal view applies to the UMBEL Reference Concept ontology or to domain ontologies based on the UMBEL Vocabulary. These relationships show how `RefConcepts` are clustered into `SuperTypes` or how hierarchical relationships are established between Reference Concepts (via the `skos:narrowerTransitive` and `skos:broaderTransitive` relations). The concept relationships and their structure is a "class" view (quadrant **A**); treating these concepts as instances in their own right and relating them to SKOS is provided by the right-hand "individual" (instance) view (quadrant **B**).

The bottom of the diagram (quadrants **C** and **D**) shows either classes or individuals in external ontologies. The key mapping predicates cross this boundary (the broad dotted line) between UMBEL-based ontologies and external ontologies:

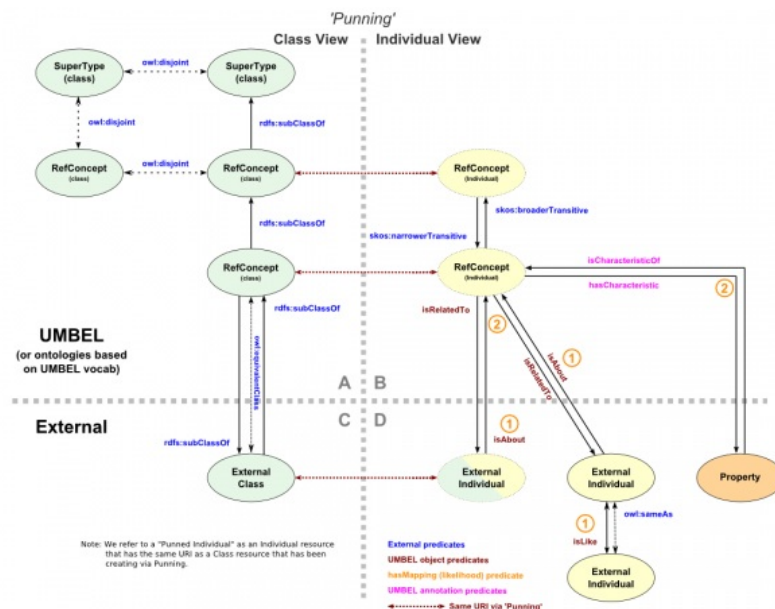


Figure 1. UMBEL Predicates

UMBEL "Class" View

This view is quadrant **A** in the above diagram.

In this view, the reference concepts and their relationships are treated entirely in a class manner. Given reference concepts may be related to one another in `subClass` relationships, or multiples of them may be clustered into `superTypes`, the relationships between which may be established as `disjoint` (or not).

The class view provides the major reasoning structure for the ontology, and corresponds (mostly) to what is known as the TBox (terminological box) in description logics. [\[21\]](#)

UMBEL "Individual" View

This view is quadrant **B** in the above diagram.

The individual (or instance) view parallels all of the reference concepts, only portrays the relationships as instances, not classes. An instance view allows mapping to the SKOS conceptual structure, provides a more flexible means for describing relationships, and is the preferred way to relate the subject matter of external content to the reference vocabulary. The individual view also provides good hooks for relating the attributes of external datasets to their governing concepts in the vocabulary.

The "Punning" Relationship

In OWL 2, these dual views as either instances (individuals) (quadrant **B**) or classes (quadrant **A**) occur via a metamodeling technique called "[punning](#)". When a given concept is used in both an instance and class mode, the object is "punned" as both a class and instance. Both views of the concept share the same URI. Then, based on use context, the object is treated either as an individual or a class.

This "punning" relationship is shown by the maroon dotted lines in Figure 1. Generally, each reference concept is punned in this dual manner.

Though the object shares the URI, for reasoning and consistency purposes the object is actually treated as two different things (class or individual) depending on usage and context. It is this trick that enables `RefConcepts` to both model structural relationships and provide instance hooks to external vocabularies in OWL 2 while being decidable via description logics and reasoners.

External "Class" View

The external class view is shown by quadrant **C** in the above diagram. It represents a "standard" mapping where an **exact** assertion of the mapped relationship between internal (UMBEL vocabulary) and external classes can be made.

The two possible mapping predicates are `rdfs:subClassOf` or `owl:equivalentClass`.

External "Individual" and Properties View

The major uses of the reference vocabularies are reflected in the quadrant **D** mappings in the above diagram.

In most uses, the most prevalent property to be used is the `umbel:isAbout` assertion. This predicate is useful when tagging external content with metadata for alignment with an UMBEL-based reference ontology. The reciprocal assertion, `umbel:isRelatedTo` is when an assertion within an UMBEL vocabulary is desired to an external ontology. Its application is where the reference vocabulary itself needs to refer to an external topic or concept.

For the approximate mapping relations there is also the optional `umbel:hasMapping` property to describe the "degree" of the assertion. (This degree could be quantified, such as from some analytical technique, or use textual qualifiers.)

`umbel:hasMapping` may be reified for the `umbel:isAbout` or `umbel:isLike` properties (#1 in the diagram) assertion, or the `umbel:correspondsTo` and `umbel:relatesToXXX` predicates (#2 in the diagram, and the [LINKING AND MAPPING PREDICATES](#) section below).

Two annotation properties are used to describe the attribute characteristics of a `RefConcept`, namely `umbel:hasCharacteristic` and its reciprocal, `umbel:isCharacteristicOf`. These properties are the means by which the external properties to describe things are able to be brought in and used as lookup references (that is, metadata) to external data attributes. As annotation properties, they have weak semantics and are used for accounting as opposed to reasoning purposes.

Metadata or descriptive characteristics of external concepts could also be related using the `umbel:isAbout` relationship, which is legal given the domain and range definitions of the `umbel:isAbout` predicate.

However, where you want to clearly relate to datasets and data attributes, the `umbel:hasCharacteristic` property provides a means for such refined distinctions, which can be quite helpful when feeding widgets or data displays with metadata not explicit modeled in the domain ontology.

In all cases, the linkage from either external classes or instances occur via the "punned" individual in UMBEL. If the external object is a class, it is therefore punned within the system (shown as the dual color bubble).

More discussion and use guidance (#2 in the Figure 1 diagram, including for the `umbel:correspondsTo` and `umbel:relatesToXXX` predicates) are provided under the [LINKING AND MAPPING PREDICATES](#) section below.

Specifications

Here are the specifications for the UMBEL properties in the vocabulary. Most properties have a status of 'Testing'.^[22]

| Property name | <code>umbel:correspondsTo</code> |
|---------------|--|
| Description | The property <code>umbel:correspondsTo</code> is used to assert a close correspondence between an external class, named entity, individual or instance with a Reference Concept class. <code>umbel:correspondsTo</code> relates the external class, named entity, individual or instance to the class through the basis of both its subject matter and intended scope. This predicate should be used where the correspondence between the two entities is felt to be nearly equivalent to a <code>sameAs</code> assertion, and is reflexive, but without the full entailments of intensional class memberships. In these cases, both entities are understood to have the same type and intended scope. |

| | |
|------------------|---|
| | <p>but without asserting a full class-level or sameAs individual relationship.</p> <p>This predicate is designed for the circumstance of aligning two different ontologies or knowledge bases based on node-level correspondences, but without entailing the actual ontological relationships and structure of the object source. For example, the umbel:correspondsTo predicate is used to assert close correspondence between UMBEL Reference Concepts and Wikipedia categories or pages, yet without entailing the actual Wikipedia category structure.</p> <p>This property asserts a different and stronger relationship than umbel:isAbout. One practical use is to guide specific instance member determinations when, say, the native structure of the external ontology or knowledge base is to be analyzed and replaced with an UMBEL-based structure.</p> <p>This property is therefore used to create a nearly equivalent assertion (however, with the degree of that equivalence being unknown or unknowable) between an external instance or class and a Reference Concept class.</p> |
| Domain | owl:Thing |
| Range | umbel:RefConcept |
| Reflexive | True |
| Status | Testing |

Table 6. correspondsTo Property

| | |
|----------------------|--|
| Property name | umbel:isAbout |
| Description | <p>The property umbel:isAbout is used to assert the relation between an external named entity, individual or instance with a Reference Concept class. umbel:isAbout relates the external named entity, individual or instance to the class through the basis of its subject matter. The relation acknowledges that the scope of the class can not be determined solely by the aggregation or extent of its associated individual entity members, and that the nature of the Reference Concept class may not alone bound or define the individual entity.</p> <p>This property is therefore used to create a topical assertion between an individual and a Reference Concept.</p> |
| Domain | owl:Thing |
| Range | umbel:RefConcept |
| Inverse-of | umbel:isRelatedTo |
| Status | Testing |

Table 7. isAbout Property

| | |
|----------------------|--|
| Property name | umbel:isRelatedTo |
| Description | Check the definition of umbel:isAbout for the definition of this property; isRelatedTo is the inverse property of isAbout. |
| Domain | umbel:RefConcept |
| Range | owl:Thing |
| Inverse-of | umbel:isAbout |
| Status | Testing |

Table 8. isRelatedTo Property

| | |
|----------------------|---|
| Property name | umbel:relatesToXXX |
| Description | <p>The various properties designated by umbel:relatesToXXX are used to assert a relationship between an external instance (object) and a particular (XXX) umbel:SuperType. There may be as many umbel:relatesToXXX properties as there are numbers of SuperTypes (see next table for the listing of all specific umbel:relatesToXXX predicates).</p> <p>The assertion of this property does not entail class membership with the asserted SuperType. Rather, the assertion may be based on particular attributes or characteristics of the object at hand. For example, a</p> |

| | |
|---------------|--|
| | British person might have an <code>umbel:relatesToXXX</code> asserted relation to the SuperType of the geopolitical entity of Britain, though the actual thing at hand (person) is a member of the Person class SuperType. |
| | This predicate is used for filtering or clustering, often within user interfaces. Multiple <code>umbel:relatesToXXX</code> assertions may be made for the same instance. |
| Domain | <code>owl:Thing</code> |
| Range | <code>umbel:SuperType</code> |
| Status | Testing |

Table 9. `relatesToXXX` Property

Each of the 32 UMBEL SuperTypes has a matching predicate for external topic assignments (`umbel:relatesToOtherOrganism` shares two SuperTypes, leading to 31 different predicates):

| SuperType | Mapping Predicate | Comments |
|---------------------|--|---|
| NaturalPhenomena | <code>relatesToPhenomenon</code> | This predicate relates an external entity to the SuperType (ST) shown. It indicates there is a relationship to the ST of a verifiable nature, but which is undetermined as to strength or a full <code>rdf:type</code> relationship |
| NaturalSubstances | <code>relatesToSubstance</code> | same as above |
| Earthscape | <code>relatesToEarth</code> | same as above |
| Extraterrestrial | <code>relatesToHeavens</code> | same as above |
| Prokaryotes | <code>relatesToOtherOrganism</code> | same as above |
| ProtistsFungus | | |
| Plants | <code>relatesToPlant</code> | same as above |
| Animals | <code>relatesToAnimal</code> | same as above |
| Diseases | <code>relatesToDisease</code> | same as above |
| PersonTypes | <code>relatesToPersonType</code> | same as above |
| Organizations | <code>relatesToOrganizationType</code> | same as above |
| FinanceEconomy | <code>relatesToFinanceEconomy</code> | same as above |
| Society | <code>relatesToSociety</code> | same as above |
| Activities | <code>relatesToActivity</code> | same as above |
| Events | <code>relatesToEvent</code> | same as above |
| Time | <code>relatesToTime</code> | same as above |
| Products | <code>relatesToProductType</code> | same as above |
| FoodorDrink | <code>relatesToFoodDrink</code> | same as above |
| Drugs | <code>relatesToDrug</code> | same as above |
| Facilities | <code>relatesToFacility</code> | same as above |
| Geopolitical | <code>relatesToGeoEntity</code> | same as above |
| Chemistry | <code>relatesToChemistry</code> | same as above |
| AudioInfo | <code>relatesToAudioMusic</code> | same as above |
| VisualInfo | <code>relatesToVisualInfo</code> | same as above |
| WrittenInfo | <code>relatesToWrittenInfo</code> | same as above |
| StructuredInfo | <code>relatesToStructuredInfo</code> | same as above |
| NotationsReferences | <code>relatesToNotation</code> | same as above |
| Numbers | <code>relatesToNumbers</code> | same as above |
| Attributes | <code>relatesToAttribute</code> | same as above |
| Entities | <code>relatesToEntity</code> | same as above |
| Abstract | <code>relatesToAbstraction</code> | same as above |
| TopicsCategories | <code>relatesToTopic</code> | same as above |
| MarketsIndustries | <code>relatesToMarketIndustry</code> | deprecated as of version 1.20 |
| Workplaces | <code>relatesToWorkplaces</code> | deprecated as of version 1.20 |

Table 10. Listing of `relatesToXXX` Properties

| | |
|----------------------|---|
| Property name | <code>umbel:isLike</code> |
| Description | The property <code>umbel:isLike</code> is used to assert an |

| | |
|------------------|--|
| | <p>associative link between similar individuals who may or may not be identical, but are believed to be so. This property is not intended as a general expression of similarity, but rather the likely but uncertain same identity of the two resources being related.</p> <p>This property may be considered as an alternative to <code>sameAs</code> where there is not a certainty of sameness, and/or when it is desirable to assert a degree of overlap of sameness via the <code>umbel:hasMapping</code> reification predicate. This property can and should be changed if the certainty of the sameness of identity is subsequently determined.</p> <p><code>umbel:isLike</code> has the semantics of likely identity, but where there is some uncertainty that the two resources indeed refer to the exact same individual with the same identity. Such uncertainty can arise when, for example, common names may be used for different individuals (e.g., John Smith).</p> <p>It is appropriate to use this property when there is strong belief the two resources refer to the same individual with the same identity, but that association can not be asserted at the present time with full certitude.</p> |
| Domain | owl:Thing |
| Range | owl:Thing |
| Symmetric | True |
| Status | Testing |

Table 11. isLike Property

| | |
|----------------------|--|
| Property name | umbel:hasMapping |
| Description | <p>This property is used to reify a <code>umbel:isAbout</code>, <code>umbel:isRelatedTo</code> or an <code>umbel:isLike</code> property assertion with a statement as to its degree of mapping or relationship between subject and object.</p> <p>The <code>hasMapping</code> property may be expressed as a mapping percentage value, some quantitative metric value, or a qualitative descriptor characterizing the linkage degree or overlap between the two classes, predicates, individuals or datatypes. This value might be calculated from some external utility, may be free form, or may be based on some defined listing of mapping values expressed as literals.</p> |
| Domain | rdf:Statement |
| Range | owl:Thing |
| Status | Testing |

Table 12. hasMapping Property

| | |
|----------------------|--|
| Property name | umbel:hasCharacteristic |
| Type | owl:AnnotationProperty |
| Description | <p>The property <code>umbel:hasCharacteristic</code> is used to assert the relation between a Reference Concept, or any other classes, and external properties that may be used in external ontologies to characterize, describe, or provide attributes for data records associated with that concept or that class. It is via this property or its inverse, <code>umbel:isCharacteristicOf</code>, that external data characterizations may be incorporated and modeled within a domain ontology based on the UMBEL vocabulary.</p> |
| Inverse-of | umbel:isCharacteristicOf |
| Status | Testing |

Table 13. hasCharacteristic Property

| | |
|----------------------|---|
| Property name | umbel:isCharacteristicOf |
| Type | owl:AnnotationProperty |
| Description | <p>The property <code>umbel:isCharacteristicOf</code> is used to assert the relation between a property and a Reference Concept (or its punned individual), or any other classes, to which it applies. Such properties may be used in external ontologies to characterize, describe, or provide attributes for data records associated with that concept or that class. It is via this property or its inverse,</p> |

| | |
|-------------------|---|
| | umbel:hasCharacteristic, that external data characterizations may be incorporated and modeled within a domain ontology based on the UMBEL vocabulary. |
| Inverse-of | umbel:hasCharacteristic |
| Status | Testing |

Table 14. isCharacteristicOf Property

Incorporation of External Properties

The UMBEL vocabulary re-uses several properties from external vocabularies. Here is the list of such properties that are recommended for or used by the UMBEL Reference Concept ontology (see [Part II](#)).

Listing of Properties

| | |
|-----------------------|---|
| RDFS | <ul style="list-style-type: none"> • rdfs:subClassOf |
| OWL | <ul style="list-style-type: none"> • owl:equivalentClass |
| SKOS-Preferred | <ul style="list-style-type: none"> • skos:prefLabel • skos:altLabel • skos:hiddenLabel • skos:definition • skos:broaderTransitive • skos:narrowerTransitive |
| SKOS-Optional | <ul style="list-style-type: none"> • skos:note • skos:changeNote • skos:editorialNote • skos:example • skos:historyNote • skos:scopeNote • skos:broader • skos:narrower • skos:related |

Table 15. Recommended External Properties for UMBEL

Language Annotations of Classes

The idea of Reference Concepts is that they are referents to real-world concepts with exact definitions and instance members, the combination of which (through both intension and extension^[23]) defines what the concept means. The label for the concept is merely a useful handle to the class membership and meaning of that concept, and has no further meaning in and of itself.

What this means is that the various labels for a given concept may be readily switched out to reflect that concept in different languages. To enable multiple language use, then, all annotations and labels should be labeled with their source language.

Earlier versions of UMBEL used the lingvoj ontology for this purpose. However, that ontology has been deprecated by its editor.

Now, each label and definition related properties should be defined by a language identifier, as defined by IETF [RFC 4646](#) or successor. Different methodologies should be used depending on the serialization language used to serialize the UMBEL reference concepts structure.

Linking to Instances and Named Entities

UMBEL thus provides a class structure of Reference Concepts to which the actual instance data of the world, what we term named entities, can interact. *Named entities* are the real things or instances in the world that are themselves natural and notable class members of Reference Concepts. ^[24]

Named entities that are instances of external ontologies can be related to UMBEL via their class relationships. In addition, UMBEL provides the predicate `umbel:isLike` (to supplement the `owl:sameAs`) for relating named entities from disparate sources to one another.

PART II: REFERENCE CONCEPTS SPECIFICATION

The second purpose of UMBEL is to provide a coherent framework of broad subjects and topics (the *UMBEL Reference Concept* ontology), suitable as binding nodes for mapping relevant Web-accessible content. This reference ontology has a specific aim of promoting interoperability, including linkages to various domain ontologies. The UMBEL Reference Concept ontology is itself based on the UMBEL Vocabulary specified in Part I.

This Part II specifies the UMBEL Reference Concept ontology, with its 35,000 reference concepts and relationships drawn from the Cyc knowledge base, which are organized into 32 mostly disjoint SuperTypes.

Reference Concepts

Reference Concepts are a distinct subset of the more broadly understood *concept* such as used in the SKOS RDFS controlled vocabulary or formal concept analysis or the very general or abstract concepts common to some upper ontologies. Reference Concepts are selected for their use as concrete, subject-related or commonly used notions for describing tangible ideas and referents in human experience and language. Note in other systems or ontologies, similar constructs may alternatively be called *topics*, *subjects*, *concepts*, *classes* or perhaps *interests*. UMBEL has adopted the term *Reference Concept* to distinguish from these uses, which have different nuances of meaning and use, as well as to highlight the use of them as referents to which topics or subjects in other ontologies may be mapped. The shorthand for *Reference Concept* is *RefConcept* or simply *RC*.

Reference Concepts are classes, the members of which are nameable instances or named entities, which by definition are held as distinct from these concepts. The UMBEL Reference Concept ontology (or "UMBEL Ontology") is a coherently organized structure (or reference "backbone") of these Reference Concepts. Via "punning" for some uses (see below), RefConcepts are also instantiated as instances (individuals).

While all Reference Concepts have a preferred label (using SKOS terminology a `skos:prefLabel`), as objects they should be understood as a *representative* or a *proxy* for that concept, and not to be confused with the thing itself. Every UMBEL Reference Concept can be expressed and referred to by a different preferred label in alternate languages.

Indeed, in a given language, different preferred labels may be swapped out without affecting the identity or use of the Reference Concept itself. Each Reference Concept should be described in its entirety by multiple `skos:altLabel`. (If misspellings are also desired as a means to map to a Reference Concept, the `skos:hiddenLabel` property should be used.) We colloquially refer to a Reference Concept's mappable terms and phrases as its *semset*, which by convention is the union of the `skos:prefLabel`, and all `skos:altLabels` and `skos:hiddenLabels` for that concept. Consistently using and emphasizing such *semsets* is one of the [best practices](#) documented in a separate annex.

Reference Concepts are the core constituents to the UMBEL Reference Concept ontology.

Basis in OpenCyc

For the reasons as stated in the [use of OpenCyc as a building block](#) above, **all reference concepts are based on existing concepts in OpenCyc**, the open source version of the Cyc knowledge base. About 35,000 of them have been distilled and are part of the UMBEL Reference Concept ontology.^[25]

Structural Predicates

Reference Concepts are related to one another by the properties `skos:broaderTransitive` and `skos:narrowerTransitive`. These two relations create a taxonomic structure within UMBEL's concepts. We can define which reference concept is a more general, or more specific, than other concepts. This enables us to do inferencing on the taxonomic structure of UMBEL.

Reference Concepts are also aggregated into (mostly) disjoint *SuperTypes*. This provides improved organization of the large number of reference concepts and, by use of disjoint assertions, faster structure processing and enhanced reasoning. Reference Concepts are assigned to a *SuperType* by the `rdfs:subClassOf` property.

Metamodeling and Punning

Every Reference Concept in the UMBEL ontology is "punned" to also have an individual (instance) instantiation. Each instance shares the same annotation properties (`skos:definition`, `skos:prefLabel`, `skos:altLabel`, `skos:hiddenLabel`, etc.) as its "punned" class. The instance is assigned to its associated class and any other classes to which it has a `subClass` relationship. The punning that is taking place is shown in the Figure 1 above.

This metamodeling convention enables the ontology structure to be treated as either a class or instance structure, depending on context, and allows the ontology to be decidable.

Use of OpenCyc External IDs

UMBEL reference concept names have been used for convenience only. When a new version of UMBEL is created, the "external IDs" of the OpenCyc classes are used to link these classes to UMBEL reference concepts. That way, if their naming conventions change from an OpenCyc version A to a version B, then we are still able to update the proper UMBEL concepts according to their new OpenCyc definitions.

Note that the OpenCyc external IDs are only used when we create a new version of UMBEL. Otherwise the IRIs of the UMBEL reference concepts use the "human readable" `skos:prefLabel` to refer to the concept.

Modularization

The UMBEL Reference Concept ontology has been modularized into 'core', 'geo', 'entities', and 'attributes' modules (with the latter also being presented with additional structure as the Attributes Ontology). See further [Annex J](#) and [Annex L](#) for how these modules and ontology are constructed. The move to modularity allows more flexibility in the use of UMBEL, and the ability to swap out capable external ontologies -- in this case GeoNames^[7] -- where specific capabilities are desired.

Additional modularization of the UMBEL reference concepts may be forthcoming over time.

Not Determinative for Separate UMBEL Domain Ontologies

The assignments and uses of UMBEL Reference Concepts in its ontology is not determinative for what concepts can or should be used in other domain ontologies based on the UMBEL Vocabulary (Part I).

SuperTypes

The UMBEL Reference Concept ontology presently contains 32 *SuperTypes*, organized into nine dimensions. Details about these *SuperTypes*, statistics, reference concept assignments and methodologies used in their development and creation is provided by the [Annex G: UMBEL SuperTypes Documentation](#). A summary of that material is provided in this section, as has been updated .

Non-disjoint SuperTypes

The four *SuperTypes* of Attributes, Abstract-level, Entities and Topics/Categories are designed to be **fully** non-disjoint, and do **not** participate in any disjoint assertions.^[26] About 10% of all reference concepts fall into this grouping.

Disjoint SuperTypes

The remaining 28 *SuperTypes* are designed to be as disjoint as possible. How disjoint and other commentary is provided in the [Annex G](#) report. These 28 *SuperTypes* designed as **mostly disjoint** are:

- Natural Phenomena
- Natural Substances
- Earthscape
- Extraterrestrial
- Prokaryotes
- Protists or Fungus
- Plants
- Animals
- Diseases
- Person Types
- Organizations
- Finance & Economy
- Society (culture, issues, beliefs)
- Activities
- Events
- Time
- Products
- Food or Drink
- Drugs
- Facilities
- Geopolitical
- Chemistry (n.o.c)
- Audio Info
- Visual Info
- Written Info
- Structured Info
- Notations & References
- Numbers

SuperType Dimensions

In addition, all of these *SuperTypes* are clustered into nine "dimensions", which are useful for aggregation and organizational purposes, but which have no direct bearing on logic assertions or disjointness testing. These nine dimensions, with their associated *SuperTypes*, are:

| | |
|------------------|--------------------|
| Natural World | Natural Phenomena |
| | Natural Substances |
| | Earthscape |
| | Extraterrestrial |
| Living Things | Prokaryotes |
| | Protists & Fungus |
| | Plants |
| | Animals |
| | Diseases |
| | Person Types |
| | Organizations |
| | Finance & Economy |
| Human Activities | Society |
| | Activities |
| | Events |
| Time-related | Time |
| | Products |
| Human Works | Food or Drink |
| | Drugs |
| | Facilities |
| | |

| | |
|----------------|------------------------|
| Human Places | Geopolitical |
| Information | Chemistry (n.o.c) |
| | Audio Info |
| | Visual Info |
| | Written Info |
| | Structured Info |
| | Notations & References |
| | Numbers |
| Descriptive | Attributes |
| Classificatory | Abstract-level |
| | Entities |
| | Topics/Categories |

Note the last two dimensions (and four *SuperTypes*) are by definition non-disjoint.

Combination (Overlapping) SuperTypes

This convention has been deferred to a later UMBEL v. 1.xx release.

Aside from the four *SuperTypes* that are designed as non-disjoint, reference concepts in the remaining *SuperTypes* are largely (65%) disjoint. And, where that minority is non-disjoint, the overlap is minor and at most limited to one or two *SuperTypes*.

As a result of this very narrow overlap, the UMBEL Reference Concept ontology adopts a convention of assigned a combined *SuperType* class name that concatenates the names of the overlapping *SuperTypes*. Thus, for example, the `umbel:RefConcept` of Weaving, which is assigned to both the Activities and Events *SuperTypes*, is assigned to the synthetic *SuperType* class of ActivitiesEvents. By convention, these synthetic overlap assignments are limited to pairwise with the order of *SuperType* class concatenation alphabetical.

Linking Predicates

The internal structural properties for building the UMBEL Reference Concept ontology are described under [Structural Predicates](#) above.

Reference Concepts classes are linked to external ontology *classes* by using a variety of properties, such as `rdfs:subClassOf` and `owl:equivalentClass`.

Otherwise, any classes, properties, individuals and datatype can be linked to one, or multiple reference concepts by using the `umbel:isAbout` predicate and its inverse `umbel:isRelatedTo`, or using the `umbel:correspondsTo` predicate.

For more detail on the linking properties and their use in the UMBEL ontology, see the [LINKING AND MAPPING PREDICATES](#) section below.

External Mappings

In addition, the UMBEL Reference Concept ontology is mapped to a number of external ontologies. For a listing of these, see [Annex A: Listing of Linked External Ontologies](#).

Instances and Named Entities

Named entities are the real things or instances in the world that are themselves natural and notable class members of Reference Concepts. Named entities are the instances of the Reference Concepts in the standard definition of the term.

Please refer to the section *Using UMBEL to Describe Things* in [Annex E](#) to see the description of the Muhammad Ali named entity, and refer to the section *Linking External Entities* [Annex F](#) to see how some named entities are used and mapped to within UMBEL.

LINKING AND MAPPING PREDICATES

The guiding purpose of the UMBEL Vocabulary and its associated Reference Concept Structure is to provide a coherent means to link and interoperate Web-based content. Since there is a huge diversity of organizational structure and world views on the Web, the linking and mapping predicates to fulfill this purpose must also capture that diversity. Relations between things on the Web can range from the exact and identity, to the approximate, descriptive and casual.^[27]

The large-scale and diverse mapping of UMBEL to Wikipedia undertaken for version 1.00 (as well as the other mappings conducted) provided real-world tests and challenges for how to capture this diversity. The need is to find the range of mapping options that can reflect and capture quality, accurate mappings. Further, because mappings also can be aided with a variety of techniques from the manual to the automatic, it is important to characterize the specific mapping methods used whenever a linking predicate is assigned. Such qualifications are important to distinguish mapping trustworthiness, plus to also enable later segregation for the application of improved methods as they may arise.

As a result, the UMBEL Vocabulary and its use of external vocabularies offers a diversity of linking and mapping predicates.

Guidelines for how these differ, how they are used, and how they are qualified is described in this section.

A Comparison of Options

Properties for linking and mapping need to differ more than in name or intended use. They must represent differences that affect inferences and reasoners, and can be acted upon by specific utilities via user interfaces and other applications. Furthermore, the diversity of mapping predicates should capture the types of diverse mappings and linkages possible between disparate sources.

Sometimes things are individuals or instances; other times they are classes or groupings of similar things. Sometimes things are of the same kind, but not exactly aligned. Sometimes things are unlike, but related in a common way. (Everything in Britain, for example, is a British "thing" even though they may be as different as trees, dead kings or cathedrals.) Sometimes we want to say something about a thing, such as an animal's fur color or age, as a way to further characterize it, and so on.

The OWL 2 language and existing semantic Web languages give us some tools and existing vocabulary to capture some of this diversity. How these options have been applied to the UMBEL mapping and linking predicates is shown by this table:

| Property | Relative Strength | Usage | Standard Reasoner? | Inverse Property? | Kind of Thing | | Symmetrical? | Transitive? | Reflexive? |
|---------------------------------------|-------------------|--------------|--------------------|-------------------|---------------|---------------|--------------|-------------|------------|
| | | | | | It is | It Relates to | | | |
| <code>owl:equivalentClass</code> | 10 | equivalence | X | N/A | class | class | yes | yes | yes |
| <code>owl:sameAs</code> | 9 | identity | X | N/A | individual | individual | yes | yes | yes |
| <code>rdfs:subClassOf</code> | 8 | subset | X | | class | class | no | yes | yes |
| <code>umbel:correspondsTo</code> | 6 | ~equivalence | + / - | | anything | RefConcept | yes | yes | yes |
| <code>rdf:type</code> | 5 | membership | X | | anything | class | no | no | no |
| <code>umbel:isAbout</code> | 4 | topical | | X | anything | RefConcept | perhaps | not likely | not likely |
| <code>umbel:isLike</code> | 3 | similarity | | | anything | anything | yes | no | not likely |
| <code>umbel:relatesToXXX</code> | 2 | relationship | | | anything | SuperType | no | no | not likely |
| <code>umbel:isCharacteristicOf</code> | 1 | attribute | | X | anything | RefConcept | no | no | no |

Table 16. Comparison of Linking and Mapping Predicates

We discuss each of these predicates below. But, first, let's discuss what is in this table and how to interpret it. [\[28\]](#)

- *Relative power* - is meant to capture the inferencing power (*entailments*) embodied in the predicate. Identity (equivalence), class implications, and specific predicate properties that can be acted upon by reasoners are given higher relative power
- *Standard reasoner?* - indicates whether standard reasoners [\[29\]](#) draw inferences and entailments from the specific property. A "+" / "-" indication indicates that reasoners do not recognize the specific property per se, but can act upon the predicates (such as symmetric, transitive or reflexive) used to define the predicate
- *Inverse property?* - indicates whether there is an inverse property in UMBEL that is not listed in the table. In such cases, the predicate shown is the one that treats the external entity as the subject
- *It is a kind of thing* - is the same as *domain*; it means the kind of thing to which the subject belongs
- *It relates to a kind of thing* - is the same as *range*; it means the kind of thing to which the object of the subject belongs
- *Symmetrical?* - describes whether the predicate for an s - p - o relationship can also apply in the o - p - s manner
- *Transitive?* - is whether the predicate interlinks two individuals A and C whenever it interlinks A with B and B with C for some individual B
- *Reflexive?* - By that is meant whether the subject has itself as a member. In a reflexive closure between subject and object the subject is fully included as a member. Equivalence, subset, greater than or equal to, and less than or equal to relationships are reflexive; not equal, less than or greater than relationships are not.

The *Usage* metric is described for each property below.

Narrative Discussion

To further aid the understanding of these properties, we can also group them into equivalence, membership, approximate or descriptive categories.

Equivalent Properties

Equivalent properties are the most powerful available since they entail all possible axioms between the classes.

`owl:equivalentClass`

Equivalent class means that two classes have the same members; each is a sub-class of the other. The classes may differ in terms of annotations defined for each of them, but otherwise they are axiomatically equivalent.

An `owl:equivalentClass` assertion is the most powerful available because of its ability to 'Explode the Domain'. [\[30\]](#) Because of its entailments, `owl:equivalentClass` should be used with great care.

`owl:sameAs`

The `owl:sameAs` assertion claims two instances to be an identical individual. This assertion also carries with it strong entailments of symmetry and reflexivity.

`owl:sameAs` is often misapplied.^[31] Because of its entailments, it too should be used with great care. When there are doubts about claiming this strong relationship, UMBEL has the `umbel:isLike` alternative (see below).

Membership Properties

Membership properties assert that an instance is a member of a class.

`rdfs:subClassOf`

The `rdfs:subClassOf` asserts that one class is a subset of another class. This assertion is transitive and reflexive. It is a key means for asserting hierarchical or taxonomic structures in an ontology. This assertion also has strong entailments, particularly in the sense of members having consistent general or more specific relationships to one another.

Care must be exercised that full inclusivity of members occurs when asserting this relationship. When correctly asserted, however, this is one of the most powerful means to establish a reasoning structure in an ontology because of its transitivity.

`rdf:type`

The `rdf:type` assertion assigns instances (individuals) to a class. While the idea is straightforward, it is important to understand the intensional nature of the target class to ensure that the assignment conforms to the intended class scope. When this determination can not be made, one of the more approximate UMBEL predicates (see below) should be used.

Approximation Properties

For one reason or another, the precise assertions of the equivalent or membership properties above may not be appropriate. For example, we might not know sufficiently an intended class scope, or there might be ambiguity as to the identity of a specific entity (is it Jimmy Johnson the football coach, race car driver, fighter, local plumber or someone else?). Among other options -- along a spectrum of relatedness -- is the desire to assign a predicate that is meant to represent the same kind of thing, yet without knowing if the relationship is an equivalence (identity, or `sameAs`), a subset, or merely just a member of relationship. Alternatively, we may recognize that we are dealing with different things, but want to assert a relationship of an uncertain nature.

This section presents the UMBEL alternatives for these different kinds of approximate predicates.^[27]

`umbel:correspondsTo`

The most powerful of these approximate predicates in terms of alignment and entailments is the `umbel:correspondsTo` property. This predicate is the recommended option if, after looking at the source and target knowledge bases,^[32] we believe we have found the best equivalent relationship, but do not have the information or assurance to assign one of the relationships above. So, while we are sure we are dealing with the same kind of thing, we may not have full confidence to be able to assign one of these alternatives:

```
rdfs:subClassOf
owl:equivalentClass
owl:sameAs
superClassOf
```

Thus, with respect to existing and commonly used predicates, we want an umbrella property that is generally equivalent or so in nature, and if perhaps known precisely might actually encompass one of the above relations, but we don't have the certainty to choose one of them nor perhaps assert full "sameness". This is not too dissimilar from the rationale being tested for the `x:coref` predicate from the UMBC Ebiquity group.^{[33][34]} The `umbel:correspondsTo` predicate is used, for example, in the mapping of UMBEL to GeoNames feature codes; see further [Annex J](#).

The property `umbel:correspondsTo` is thus used to assert a close correspondence between an external class, named entity, individual or instance with a Reference Concept class. It asserts this correspondence through the basis of both its subject matter and intended scope.

This property may be reified with the `umbel:hasMapping` property to describe the "degree" of the assertion.

`umbel:isAbout`

In most uses, the most prevalent linking property to be used is the `umbel:isAbout` assertion. This predicate is useful when tagging external content with metadata for alignment with an UMBEL-based reference ontology. The reciprocal assertion, `umbel:isRelatedTo` is when an assertion within an UMBEL vocabulary is desired to an external ontology. Its application is where the reference vocabulary itself needs to refer to an external topic or concept.

The `umbel:isAbout` predicate does not have the same level of confidence or "sameness" as the `umbel:correspondsTo` property. It may also reflect an assertion that is more like `rdf:type`, but without the confidence of class membership.

The property `umbel:isAbout` is thus used to assert the relation between an external named entity, individual or instance with a Reference Concept class. It can be interpreted as providing a topical assertion between an individual and a Reference Concept.

This property may be reified with the `umbel:hasMapping` property to describe the "degree" of the assertion.

`umbel:isLike`

The property `umbel:isLike` is used to assert an associative link between similar individuals who may or may not be identical, but are believed to be so. This property is not intended as a general expression of similarity, but rather the likely but uncertain same identity of the two resources being related.

This property may be considered as an alternative to `sameAs` where there is not a certainty of sameness, and/or when it is desirable to assert a degree of overlap of sameness via the `umbel:hasMapping` reification predicate. This property can and should be changed if the certainty of the sameness of identity is subsequently determined.

It is appropriate to use this property when there is strong belief the two resources refer to the same individual with the same identity, but that association can not be asserted at the present time with full certitude.

This property may be reified with the `umbel:hasMapping` property to describe the "degree" of the assertion.

umbel:relatesToXXX

At a different point along this relatedness spectrum we have *unlike* things that we would like to relate to one another. It might be an attribute, a characteristic or a functional property about something that we care to describe. Further, by nature of the thing we are relating, we may also be able to describe the kind of thing we are relating. The UMBEL `SuperTypes` (among many other options) gives us one such means to characterize the thing being related.

UMBEL presently has 31 predicates for these assertions relating to a `SuperType`.^[35] The various properties designated by `umbel:relatesToXXX` are used to assert a relationship between an external instance (object) and a particular (XXX) `SuperType`. The assertion of this property does not entail class membership with the asserted `SuperType`. Rather, the assertion may be based on particular attributes or characteristics of the object at hand. For example, a British person might have an `umbel:relatesToXXX` asserted relation to the `SuperType` of the geopolitical entity of Britain, though the actual thing at hand (person) is a member of the Person class `SuperType`.

This predicate is used for filtering or clustering, often within user interfaces. Multiple `umbel:relatesToXXX` assertions may be made for the same instance.

This property may be reified with the `umbel:hasMapping` property to describe the "degree" of the assertion.

Descriptive Properties

Descriptive properties are annotation properties.

umbel:isCharacteristicOf

Two annotation properties are used to describe the attribute characteristics of a `RefConcept`, namely `umbel:hasCharacteristic` and its reciprocal, `umbel:isCharacteristicOf`. These properties are the means by which the external properties to describe things are able to be brought in and used as lookup references (that is, metadata) to external data attributes. As annotation properties, they have weak semantics and are used for accounting as opposed to reasoning purposes.

These properties are designed to be used in external ontologies to characterize, describe, or provide attributes for data records associated with a given `RefConcept`. It is via this property or its inverse, `umbel:hasCharacteristic`, that external data characterizations may be incorporated and modeled within a domain ontology based on the UMBEL vocabulary.

Qualifying the Mappings

UMBEL, for its current mappings and purposes, has adopted a controlled vocabulary for characterizing the `umbel:hasMapping` predicate.^[36]

This controlled vocabulary is based on instances of the `Qualifier` class. This class represents a set of descriptions to indicate the method used when applying an approximate mapping predicate (see above). Here is the current listing:

| Qualifier | Description |
|-------------------------------|---|
| Manual - Nearly Equivalent | The two mapped concepts are deemed to be nearly an equivalentClass or sameAs relationship, but not 100% so |
| Manual - Similar Sense | The two mapped concepts share much overlap, but are not the exact same sense, such as an action as related to the thing it acts upon |
| Heuristic - ListOf Basis | Type assignment based on Wikipedia ListOf category; not currently used |
| Heuristic - Not Specified | Heuristic mapping method applied; script or technique not otherwise specified |
| External - OpenCyc Mapping | Mapping based on existing OpenCyc assertion |
| External - DBOntology Mapping | Mapping based on existing DBOntology assertion |
| External - GeoNames Mapping | Mapping based on existing GeoNames assertion |
| Automatic - Inspected SV | Mapping based on automatic scoring of concepts using Semantic Vectors , with specific alignment choice based on hand selection |
| Automatic - Inspected S-Match | Mapping based on automatic scoring of concepts using S-Match , with specific alignment choice based on hand selection; not currently used |
| Automatic - Not Specified | Mapping based on automatic scoring of concepts using a script or technique not otherwise specified; not currently used |

EXAMPLES

We provide some examples below that show how Reference Concepts and named entities are described in RDF and serialized in N3 [\[37\]](#).

Reference Concept Description

This example is a sample of the *UMBEL Ontology, Vol. 2: Subject Concepts and Named Entities Instantiation* document [\[38\]](#). This is the RDF description of the *Project* Reference Concept. Linkage between UMBEL Reference Concepts is performed in a hierarchical way using properties `skos:broaderTransitive` and `skos:narrowerTransitive`.

```
rc:Project a umbel:RefConcept ;
  a owl:Class ;
  a rc:TemporalStuffType ;
  umbel:definition ""An organized endeavor with a set goal""@en ;
  skos:broaderTransitive rc:SocialOccurrence ;
  skos:broaderTransitive rc:PurposefulAction ;
  skos:narrowerTransitive rc:ConstructionProject ;
  skos:narrowerTransitive rc:Adventure ;
  skos:narrowerTransitive rc:ResearchProject ;
  skos:narrowerTransitive rc:Campaign ;
  skos:narrowerTransitive rc:Enterprise-Project ;
  skos:narrowerTransitive rc:GovernmentProgram ;
  skos:narrowerTransitive rc:RetirementSystem ;
  skos:narrowerTransitive rc:MassTransitSystem .
```

Table 18. Reference Concept Description Example

Named Entity Instantiation

This example shows how the Reference Concept `rc:Business` is used to describe the Structured Dynamics LLC business. Currently, most businesses are described using the `foaf:Organization` class of the Friend-of-a-Friend Ontology. However an organization can be many things and this makes such a description less useful than if we would do it with a more specific concept. The idea is to use the UMBEL subject structure to describe *things* in RDF. Here what we want is to describe a business, and not an organization.

For example, a `rc:Business` is a sub-class of `foaf:Organization`. This means that we can re-use all properties defined in the FOAF Ontology to describe this `rc:Business`.

```
<http://structureddynamics.com/about/> a rc:Business ;
  foaf:name "Structured Dynamics LLC." ;
  foaf:birthday "2008-11-25" ;
  bio:olb "Structured Dynamics is an innovative technology and consulting company helping to catalyze
the semantic enterprise. Our open semantic framework is the full expression of these innovations." @en ;
  foaf:homepage <http://structureddynamics.com> .
```

Table 19. Named Entity Instantiation Example

Linkage Between External Classes and Reference Concepts

There are three ways to describe the relation between a Reference Concept and an external ontology class. An example for each way is described in the *Table 20* below.

The first example shows how the `owl:equivalentClass` property is used. Then we see how the property `rdfs:subClassOf` is used.

Remember that the section *Linking to UMBEL Reference Concepts* explains how to properly make this linkage to remain consistent with the UMBEL Reference Concept structure.

```
<http://umbel.org/umbel/rc/Organization>
<http://www.w3.org/2002/07/owl#equivalentClass>
<http://xmlns.com/foaf/0.1/Organization> .

<http://www.geonames.org/ontology#Feature>
<http://www.w3.org/2000/01/rdf-schema#subClassOf>
<http://umbel.org/umbel/rc/SpatialThing> .
```

Table 20. equivalentClass and subClassOf Examples

Another way to link an external ontology class to an UMBEL Reference Concept is to use the `umbel:isAbout` property and to optionally reify the statement with a confidence value defined by the `hasMapping` predicate.

In the example below, the `umbel:isAbout` relationship between the `foo:Bar` and the `rc:Person` classes tell us that there is a certain degree of similarity between the two classes. This degree is defined by the `umbel:hasMapping` predicate. This mapping level can be interpreted, in the case of a class-to-class relation, as the size of the intersection of the classes extensions.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```

@prefix umbel: <http://umbel.org/umbel/> .
@prefix rc: <http://umbel.org/umbel/rc/> .
@prefix foo: <http://purl.org/ontology/foo/> .

foo:Bar umbel:isAbout rc:Person .

_:rei12345 rdf:type rdf:Statement .
_:rei12345 rdf:subject foo:Bar .
_:rei12345 rdf:predicate umbel:hasMapping .
_:rei12345 rdf:object rc:Person .
_:rei12345 umbel:hasMapping"0.53" .

```

Table 21. isAbout and hasMapping Example

Linkage Between Named Entities

Firstly, the linkage of named entities is done using the `owl:sameAs` property. Each named entity is an individual belonging to a Reference Concept class. As written in the OWL 1 Ontology document:^[39] *The built-in OWL property `owl:sameAs` links an individual to an individual. Such an `owl:sameAs` statement indicates that two URI references actually refer to the same thing: the individuals have the same "identity".*

This is the only property that makes sense to use in this context. A named entity **is the same as** this other named entity; otherwise they are distinctly different named entities.

```

<http://www.mpii.de/yago/resource/Abraham_Lincoln>
<http://www.w3.org/2002/07/owl#sameAs>
<http://dbpedia.org/resource/Abraham_Lincoln> .

```

Table 22. owl:sameAs Linkage of Named Entities Example

If two named entities are distinct, then the relationship between the two named entities will be described using other ontologies properties such as: `foaf:knows`, `dcterms:hasPart`, `sioc:link`, etc. So, virtually all-existing properties defined in external ontologies.

Additionally, the property `umbel:isLike` can be used to state that two named entities "likely" have the same identity.

```

<http://dbpedia.org/resource/United_States_of_America> umbel:isLike
<http://dbpedia.org/resource/Great_Satan> ;
umbel:isLike <http://dbpedia.org/resource/Uncle_Sam> .

```

Table 23. umbel:isLike Linkage of Named Entities Example

With the relatively small amount of Linked Data at present, most `umbel:isLike` relations will occur via aliases or synonyms where different labels may be used for the identical entity. At scale, though, this predicate is apt to become more important when there is some uncertainty or ambiguity that two references indeed belong to the same thing (e.g., a person called 'John Smith'). In those cases, statistical matching techniques based on attributes in the references may be the basis for the assertion of sameness, to which some confidence value may be assigned.

CHANGELOG

Running Changelog

Version 1.20 Changes (current)

See further the [Annex L](#) discussion for more details on this version update. In summary, those changes include:

- Expanded mappings to OpenCyc to better capture coverage of Wikipedia content; there are now 35,533 reference concepts (RCs) in UMBEL, 35,302 of which are mapped to OpenCyc (the unmapped RCs are mostly used for organizational purposes in the Attributes Ontology – see below – and OpenCyc mismatches with key external ontologies)
- Created a new `Entities` SuperType, with 20,393 RCs designated. The `Entities` ST is by definition non-disjoint with the other SuperTypes
- Created a new `Entities` module, with 9,317 RCs assigned; the remainder of the `Entities` RCs are in core
- Created a new `Attributes Ontology` (AO), with the purpose of enabling property (attribute) mappings to UMBEL
- Created a new `Attributes` module, with 1,002 RCs assigned
- Expanded the direct UMBEL RC to Wikipedia page mappings, with 25,582 currently mapped, or nearly three-quarters (72%) of RCs now assigned
- Created a new [Annex Z](#) to hold updated statistics about UMBEL
- Deprecated the `Workplaces` SuperType, and merged with the `Facilities` ST
- Deprecated the `MarketIndustries` SuperType, and merged with the `Attributes` ST
- Reviewed and greatly improved ST assignments across the board; notably, the distinction between the `Events` and `Activities` SuperTypes was improved. See [Annex Z](#) for the updated ST assignment statistics
- Greatly expanded and improved the UMBEL generator to handle satisfiability tests and modules creation
- Expanded and updated the [UMBEL.org](#) Web site.

Version 1.10 Changes

- Developed an [UMBEL generator](#) that enables UMBEL to be created and loaded from scratch using a new Clojure scripting framework. It is further described in [Annex K](#)
- Updated mappings to [OpenCyc](#) (27,691 links) (version 4.0, dated October 12, 2012), [GeoNames](#) (688 links) (version 3.1, dated October 29, 2012), [schema.org](#) (754 links) (version 1.9, dated August 19, 2014) and the [DBpedia ontology](#)

(682 links) (version 3.9, retrieved August 18, 2014)

- Added reference concept definitions where missing
- Added additional altLabels (to the semsets) in many cases
- Checked graph integrity for relationships between concepts
- Reviewed and corrected prefLabels to make them unique (for more usefulness in autocompletion)
- Checked assignments of all reference concepts to a parent SuperType
- Reviewed SuperType assignment inconsistencies and removed some disjoint assertions
- Developed and released an new [UMBEL.org](http://umbel.org) Web site.

Version 1.05 Changes

- Mapped UMBEL to schema .org^[6] with a total of 298 type mappings; added new UMBEL reference concepts to provide complete mapping coverage; see further [Annex I](#)
- Split the base UMBEL Reference Concepts into a 'core' module and a 'geo' module
- The 'core' module has a total of 26,046 reference concepts, including some minor additions for the GeoNames mapping purposes^[7]
- The 'geo' module has a total of 1,854 reference concepts
- These reference concepts have been fully mapped to the GeoNames ontology using the `umbel:correspondsTo` predicate linked to 671 GeoNames feature codes
- These changes are detailed more fully in [Annex I](#)
- Upgraded the UMBEL `umbel:correspondsTo` property from a status of 'Experimental - Unstable' to 'Testing'
- All specification documents have been updated and moved to UMBEL's wiki for easier maintenance, and
- A [big graph visualization](#) has been created of the 'core' UMBEL structure.

Version 1.00 Changes

- In its role as a central mapping vocabulary, the number of UMBEL reference concepts was expanded from 20,512 to 27,917. These are all fully integrated into the UMBEL ontology with one of 33 SuperTypes (ST) assigned
- 444 PROTON classes were directly mapped to corresponding UMBEL reference concepts
- 257 DBpedia ontology classes were directly mapped to corresponding UMBEL reference concepts
- Across all mappings, 60% of all UMBEL reference concepts (or 16,884) are now linked directly to Wikipedia via the new `umbel:correspondsTo` property. Additional related breakdowns and Wikipedia linkages are:
 - 2,130,021 unique Wikipedia pages are now accessible and linked to the UMBEL structure
 - All of these Wikipedia pages are related to one or more UMBEL STs
 - 876,125 of these Wikipedia pages are assigned a specific `rdf:type`; the remaining have a less certain relationship (`umbel:relatesToXXX` predicate)
 - Across all of these mappings, nearly 4 million predicate relations (3,935,148) link UMBEL to Wikipedia
- UMBEL has been mapped to 444 PROTON classes
- UMBEL has been mapped to 671 GeoNames feature codes
- The UMBEL RefConcepts have been re-organized, with most local, geopolitical entities moved to a supplementary module. 577 prior (version 0.80) UMBEL RCs and a further 3204 new RCs have been added to this geopolitical module. This module is **not** being released for the current version because testing is incomplete
- A new `correspondsTo` predicate has been added for nearly or approximate sameAs mappings (symmetric, transitive, reflexive)
- A controlled vocabulary of qualifiers was developed for the `hasMapping` predicate
- 31 new `relatesToXXX` predicates have been added to relate external entities or concepts to UMBEL SuperTypes
- Some disjointness assertions between SuperTypes were added or changed
- Added a new section on [LINKING AND MAPPING PREDICATES](#)
- Switched former UMBEL predicates that duplicated ones in SKOS because SKOS has now been changed to accommodate OWL DL
- Upgraded most UMBEL properties from a status of 'Experimental - Unstable' to 'Testing', and
- Added an [Annex H](#) that describes the version 1.00 changes and methods.

Note: for more detailed discussion of the version 1.00 changes, see [Annex H](#) and the [UMBEL Specification 20110210](#) and similarly dated annexes.

Summary of Version 0.80 Changes

- Changed the name of 'Subject Concepts' (`SubjectConcept`, or SC) to 'Reference Concepts' (`RefConcept`, or RC). The `umbel:SubjectConcept` class got deprecated, and the `umbel:RefConcept` got added
- Deprecated the `umbel:AbstractConcept` class
- Dropped the Abstract Concepts individuals
- Added SuperTypes for high-level organization of the Reference Concepts
- Deprecated the `SemSet` class
- Dropped all `SemSet` individuals
- Deprecated the `umbel:hasSemset` object property
- Deprecated the `umbel:superClassOf` object property
- Deprecated the `umbel:isAligned` object property
- Deprecated the `umbel:linksEntity` object property
- Deprecated the `umbel:linksConcept` object property
- Deprecated the `umbel:withAlignment` object property; replaced with `umbel:hasMapping`
- Deprecated the `umbel:withLikelihood` object property; replaced with `umbel:hasMapping`
- Added an `umbel:isRelatedTo` inverse property for `umbel:isAbout`
- Added the `umbel:hasCharacteristic` and the `umbel:isCharacteristicOf` annotation property
- Made the ontology OWL-2 compliant
- Some refinements in the descriptions, ranges and domains of properties
- Better and cleaner organization of the documentation
- Many general updates and changes.

Note: for more detailed discussion of the version 0.80 changes, see the [UMBEL Specification 20101115](#).

ANNEXES

These are the current UMBEL annexes:

- [Annex A: Listing of Linked External Ontologies](#)
 - [Annex B: Historical UMBEL Documentation](#)
 - [Annex C: Best Practices using UMBEL](#)
 - [Annex D: Inferencing with UMBEL](#)
 - [Annex E: Using UMBEL](#)
 - [Annex F: Mapping with UMBEL](#)
 - [Annex G: UMBEL SuperTypes Documentation](#)
 - [Annex H: Version 1.00 Changes](#)
 - [Annex I: schema.org Mapping](#)
 - [Annex J: Geo Module and GeoNames Mapping](#)
 - [Annex K: UMBEL Generator](#)
 - [Annex L: Attributes Ontology and Version 1.20](#)
 - [Annex Z: Current UMBEL Statistics](#)
-

ENDNOTES

1. ↑ For general background on the semantic Web, begin with this [Wikipedia category](#). For background on the semantic enterprise, the [MIKE2.0 offering](#) is also a good place to start.
2. ↑ There is not a single "best" set of reference concepts. The methods and specifications presented herein could be applied to many similar reference structures. Indeed, a desirable outcome is that multiple structures may emerge for reference and interoperability, each of which also is mapped to the other reference structures.
3. ↑ [FactForge](#) is a large, searchable knowledge base of linked data selected and vetted from what is currently available. It has been developed and maintained by [Ontotext AD](#). UMBEL is one of the governing class structures within FactForge. Ontotext has also partnered with Structured Dynamics to support ongoing development and extensions of UMBEL, including its mapping to the [PROTON](#) ontology and future enhancements for FactForge.
4. ↑ The best entry point for learning about OWL 2 is, [OWL 2 Web Ontology Language Document Overview](#), which provides a listing of all of the OWL 2 specification documents and background materials.
5. ↑ [5.0 5.1](#) For more information on schema.org, see <http://schema.org/>. The complete mapping to schema.org is based on its (then) 298 types; properties (attributes) will be mapped at a later time. Details are provided in [Annex I: schema.org Mapping](#).
6. ↑ The DBpedia ontology is introduced at <http://wiki.dbpedia.org/Ontology> with more detail provided at the [mapping wiki](#). The basic mapping process is similar to that for schema.org (see [Annex I](#)).
7. ↑ [7.0 7.1 7.2](#) For more information on GeoNames, see <http://www.geonames.org/>. The complete mapping to GeoNames is based on its 671 feature codes, which describe natural, geopolitical, and human activity geo-locational information; see further <http://www.geonames.org/statistics/total.html>. Details are provided in [Annex J: Geo Module and GeoNames Mapping](#).
8. ↑ <http://cyc.com>
9. ↑ M. Bergman and F. Giasson, eds., *Distilling Subject Concepts from OpenCyc Volume 1*, Technical Report [TR 08-07-16-B1](#), July 2008, 37 pp. Also, see M. Bergman, *Basing UMBEL's Backbone on OpenCyc: Part 4 of 4 on Foundations to UMBEL*, April 2, 2008, <http://www.mkbergman.com/?p=433>.
10. ↑ M. Bergman and F. Giasson, eds., *Distilling Subject Concepts from OpenCyc Volume 3*, Technical Report [TR 08-07-16-B3](#), July 2008, 20 pp.
11. ↑ For further background on this rationale, see the four-part description logics series from M. K. Bergman, 2009. "[Making Linked Data Reasonable using Description Logics, Part 1](#)," *AI3::Adaptive Information* blog, Feb. 11, 2009; "[Making Linked Data Reasonable using Description Logics, Part 2](#)," *AI3::Adaptive Information* blog, Feb. 15, 2009; "[Making Linked Data Reasonable using Description Logics, Part 3](#)," *AI3::Adaptive Information* blog, Feb. 18, 2009; and "[Making Linked Data Reasonable using Description Logics, Part 4](#)," *AI3::Adaptive Information* blog, Feb. 23, 2009.
12. ↑ Actually, one criticism of Wikipedia is its weak conceptual structure. Its rich instance content is not matched with a coherent categorization structure, which ultimately limits its usefulness as a knowledge base. Of course, as a reference **source** it is unparalleled.
13. ↑ For a description of the open world approach, see http://en.wikipedia.org/wiki/Open_world_assumption; for a description of the closed world premise, see http://en.wikipedia.org/wiki/Closed_world_assumption
14. ↑ More detailed discussion of the open world approach may be found in, M. K. Bergman, 2009. "[The Open World Assumption: Elephant in the Room](#)," *AI3::Adaptive Information* blog, Dec. 21, 2009.
15. ↑ [Metamodeling](#) can be accomplished in a number of ways.
16. ↑ The newest version of the [Web Ontology Language, OWL 2](#), provides useful technique for doing this called "[punning](#)". The trick with "punning" is to evaluate the object based on how it is used contextually; the IRI is shared but its referent may be viewed as either a class or instance based on context. Thus, objects used both as concepts (classes) and individuals (instances) are allowed and standard OWL 2 reasoners may be used against them. For additional background on metamodeling in the context of UMBEL, see M.K. Bergman, 2010. "Metamodeling in Domain Ontologies," *AI3::Adaptive Information* blog, September 20, 2010; see <http://www.mkbergman.com/913/metamodeling-in-domain-ontologies/>.
17. ↑ <http://www.w3.org/TR/skos-reference/>
18. ↑ For a discussion of language mismatches and ways to overcome them, see [SKOS in OWL 2](#).
19. ↑ <http://motools.sourceforge.net/event/>
20. ↑ One source, among many, for [OBIE](#) is the [GATE](#) natural language processing platform from the University of Sheffield.
21. ↑ A general design approach is to make explicit a split between the "[ABox](#)" (for instance data) and "[TBox](#)" (for ontology schema) in accordance with our [working definition](#) for [description logics](#):
Description logics and their semantics traditionally split concepts and their relationships from the different treatment of instances and their attributes and roles, expressed as fact assertions. The concept split is known as the TBox (for

terminological knowledge, the basis for T in TBox) and represents the schema or taxonomy of the domain at hand. The TBox is the structural and intensional component of conceptual relationships. The second split of instances is known as the ABox (for assertions, the basis for A in ABox) and describes the attributes of instances (and individuals), the roles between instances, and other assertions about instances regarding their class membership with the TBox concepts.

22. ↑ See <http://www.w3.org/2003/06/sw-vocab-status/note>.
23. ↑ The *intension* of a concept is provided by its definition as the described scope and coverage; the *extension* of a concept is the set of its members, which collectively imply its scope and coverage. Use of both approaches is encouraged and allowed in the idea of a "reference concept".
24. ↑ Op cit., *Distilling Subject Concepts from OpenCyc Volume 1*. Also, see M. Bergman, *Subject Concepts and Named Entities: Part 3 of 4 on Foundations to UMBEL*, April 1, 2008, <http://www.mkbergman.com/?p=432>.
25. ↑ This restriction only applies to the UMBEL Reference Concept ontology. There is no limit for Cyc concepts for domain ontologies based on the UMBEL Vocabulary. The reason for limiting UMBEL reference concepts to Cyc is judgmental, but not arbitrary. The judgment is that maintaining synchronicity with the Cyc knowledge base outweighs case-specific exceptions. If allowed to perpetuate, exceptions to Cyc would dilute its usefulness as a reasoning foundation to UMBEL. Fortunately, Cycorp has been responsive to suggestions from the UMBEL community for new or edited concepts and relations in Cyc. So long as this responsiveness is the case, the judgment is that coherency and consistency outweigh possible oversights.
26. ↑ See http://en.wikipedia.org/wiki/Disjoint_set_data_structure for a discussion of disjointness.
27. ↑ [27.0 27.1](#) M.K. Bergman, 2010. "The Nature of Connectedness on the Web," *AI3::Adaptive Information* blog, November 22, 2010; see <http://www.mkbergman.com/935/the-nature-of-connectedness-on-the-web/>.
28. ↑ A good starting reference for some of these concepts is Pascal Hitzler et al., eds., 2009. *OWL 2 Web Ontology Language Primer*, a W3C Recommendation, 27 October 2009; see <http://www.w3.org/TR/owl2-primer/>.
29. ↑ Such as the *semantic reasoners* FaCT++, Racer, Pellet, Hermit, etc.
30. ↑ Fred Giasson first coined this phrase; see F. Giasson, 2008. "Exploding the Domain: UMBEL Web Services by Zitgist," blog posting on April 20, 2008; see <http://fgiasson.com/blog/index.php/2008/04/20/exploding-the-domain-umbel-web-services-by-zitgist/>.
31. ↑ Among many, many references, see a fairly comprehensive listing at http://ontologydesignpatterns.org/wiki/Community:Overloading_OWL_sameAs.
32. ↑ This predicate is designed for the circumstance of aligning two different ontologies or knowledge bases based on node-level correspondences, but without entailing the actual ontological relationships and structure of the object source. For example, the `umbel:correspondsTo` predicate is used to assert close correspondence between UMBEL Reference Concepts and Wikipedia categories or pages, yet without entailing the actual Wikipedia category structure.
33. ↑ Jennifer Sleeman and Tim Finin, 2010. "Learning Co-reference Relations for FOAF Instances," *Proceedings of the Poster and Demonstration Session at the 9th International Semantic Web Conference*, November 2010; see http://ebiquity.umbc.edu/_file_directory/_papers/522.pdf.
34. ↑ For example, in the words of Tim Finin of the Ebiquty group:
The solution we are currently exploring is to define a new property to assert that two RDF instances are co-referential when they are believed to describe the same object in the world. The two RDF descriptions might be incompatible because they are true at different times, or the sources disagree about some of the facts, or any number of reasons, so merging them with `owl:sameAs` may lead to contradictions. However, virtually merging the descriptions in a co-reference engine is fine – both provide information that is useful in disambiguating future references as well as for many other purposes.

See quote on <http://www.semanticoverflow.com/questions/1095/alternatives-to-owlsameas-for-linked-data>.

35. ↑ The same vocabulary construct can be applied to other domain ontologies based on the UMBEL Vocabulary.
36. ↑ Other ontologies or purposes based on the UMBEL vocabulary may assign their own controlled vocabularies.
37. ↑ <http://www.w3.org/DesignIssues/Notation3.html>
38. ↑ M. Bergman and F. Giasson, eds., *UMBEL Ontology, Vol. 2: Subject Concepts and Named Entities Instantiation*, **TR 08-07-16-A2**. July 2008, 5 pp.
39. ↑ <http://www.w3.org/TR/owl-ref/>