Technical Report

UMBEL Ontology

Volume 1

Technical Documentation

TR 08-07-16-A1

July 2008

Acknowledgements

The UMBEL project would like to thank Zitgist LLC for its generous donation of time and resources in programming and writing the documentation for this Technical Report.

UMBEL would also like to thank Cycorp for its support and preparation of a more current OWL version of the OpenCyc knowledge base. We would especially like to thank Larry Lefkowitz for his internal advocacy and answering many questions.

The Cyc Foundation, notably Mark Baltzegar, has been instrumental in helping to guide us through OpenCyc and to share with us many Foundation resources and projects currently in progress. The effort to date would not have been possible without this assistance.

– *Michael K. Bergman*, editor
– *Frédérick Giasson*, editor

Table of Contents

List of Tables

List of Figures

THIS VERSION AND RELATED VOLUMES

This technical documentation is for UMBEL version 0.70, released July 16, 2008. This is the first release of the ontology.

The UMBEL ontology is documented in two volumes:

- *UMBEL Ontology, Vol. 1: Technical Documentation*, **TR 08-07-16-A1**, this volume, that overviews the ontology schema, vocabulary and use, and
- *UMBEL Ontology, Vol. 2: Subject Concepts and Named Entities Instantiation*, **TR 08-07-16-A2**, which is an explanation of the N3 files in the ontology distribution.

In addition, the ontology documentation itself is also accompanied by a three-volume series describing the selection and vetting of UMBEL's 20,000 subject concepts from OpenCyc.[1] [†] These three volumes are:

- *Distilling Subject Concepts from OpenCyc, Vol. 1: Overview and Methodology*, **TR 08-07-16-B1**, the basic introduction and explanation of terminology and the distillation process
- *Distilling Subject Concepts from OpenCyc, Vol. 2: Files Documentation*, **TR 08-07-16-B2**, the listing and description of the various files accompanying this process, and
- *Distilling Subject Concepts from OpenCyc, Vol. 3: Appendices*, **TR 08-07-16-B3**, supporting materials and detailed backup.

DOWNLOAD AND ACCESS

See http://www.umbel.org/documentation.html for how to download the UMBEL ontology under a Creative Commons Attribution-Share Alike 3.0 license. Other documents and backup are also available from that location.

Reference and binding to UMBEL subject concepts occurs through the dereferencable URIs at, for example, http://umbel.org/ns/sc/ExactConceptName. See further *UMBEL Ontology, Vol. 2: Subject Concepts and Named Entities Instantiation*, **TR 08-07-16-A2** (UmbelOntology_20080716vA2.pdf), for details regarding access and use.

INTRODUCTION TO UMBEL

UMBEL (Upper-level Mapping and Binding Exchange Layer) is a lightweight ontology for relating external ontologies and their classes to UMBEL subject concepts. UMBEL subject concepts are conceptually related together using the SKOS[2] and the OWL-Full[3] ontologies. They form a structural 'backbone' comprised of subject concepts and their semantic relationships. By linking external ontologies to this conceptual structure, we *explode the domain* of the linked classes by leveraging this conceptual structure. UMBEL's project Web site is at http://www.umbel.org.

UMBEL defines "subject concepts" as a distinct subset of the more broadly understood *concept* such as used in the SKOS/OWL-Full controlled vocabulary, conceptual graphs, formal concept analysis or

[†] All references are provided in the concluding Endnotes section.

the very general concepts common to many upper ontologies. We define subject concepts as a special kind of concept: namely, ones that are concrete, subject-related and non-abstract.

UMBEL contrasts subject concepts with abstract concepts and with named entities. Abstract concepts represent abstract or ephemeral notions such as truth, beauty, evil or justice, or are thought constructs useful to organizing or categorizing things but are not readily seen in the experiential world. Named entities are the real things or instances in the world that are themselves natural and notable class members of subject concepts. More detailed distinctions are provided under Terminology and Definitions below.

Structure

The UMBEL subject concept structure is, in essence, a content graph of subject nodes related to one another via *skos:broaderTransitive* and *skos:narrowerTransitive* relations. In turn, these internal UMBEL subject concepts may be related to external classes and individual entities (named entities) via a set of relational, equivalent, or alignment predicates. About 740 nodes represent *abstract concepts*, and are included for graph integrity and consistency. The current conceptual structure has 20,093 total subject concepts and 47,293 defined relationships between them.

All of the UMBEL subject concepts and their relationships are derived from the OpenCyc ontology. This means that UMBEL is a clean and 100% subset of OpenCyc.

UMBEL, RDF and OWL

The UMBEL ontology is formally defined as an OWL-Full ontology. This means that UMBEL can take advantage of all OWL language constructs and has a free and unconstrained use of RDF constructs.

UMBEL subject concepts are both classes (*owl:Class*) and instances of the class *umbel:SubjectConcept*, and may also sometimes be instances of other subject concept classes.

In practical terms, these relationships mean that people can re-use UMBEL subject concepts to describe new individuals belonging to these classes of concepts. At the same time, UMBEL's structure is described in SKOS and is consistent with the SKOS data model.

This dichotomy is a direct benefit of the use of the OWL-Full ontology to describe UMBEL subject concepts. This not only enables us to define a subject concept structure, but also to re-use these subject concepts to describe *things* in RDF. This document will describe the many different ways to use UMBEL, its vocabulary, and its technical underpinnings.

Inference and the Open-World Assumption

The "open world" assumption is defined in the SKOS ontology reference documentation[4] as:

> *"RDF and OWL Full are designed for systems in which data may be widely distributed (e.g., the Web). As such a system becomes larger, it becomes both impractical and virtually impossible to "know" where all of the data in the system is located. Therefore, one cannot generally assume that data obtained from such a system is "complete", i.e., if some data appears to be "missing", one has to assume, in general, that the data might exist somewhere else in the system. This assumption, roughly speaking, is known as the "open world" assumption.*
>
> *This means in practice that, for a data model defined as an OWL Full ontology, some definitions*

can have an unexpected meaning. No conclusions can be drawn from "missing" data, and removing something will never make the remaining data inconsistent."

We will see in this document that the UMBEL subject concept structure is used to infer facts between related, semi-related or un-related defined external ontologies (such as FOAF, DOAP, DC, etc.).

Since no conclusion can be drawn from "missing" data, one cannot conclude that facts inferred by UMBEL, or between two external ontologies, are inconsistent by the only fact that the inferred relationship is not defined in either external ontology.

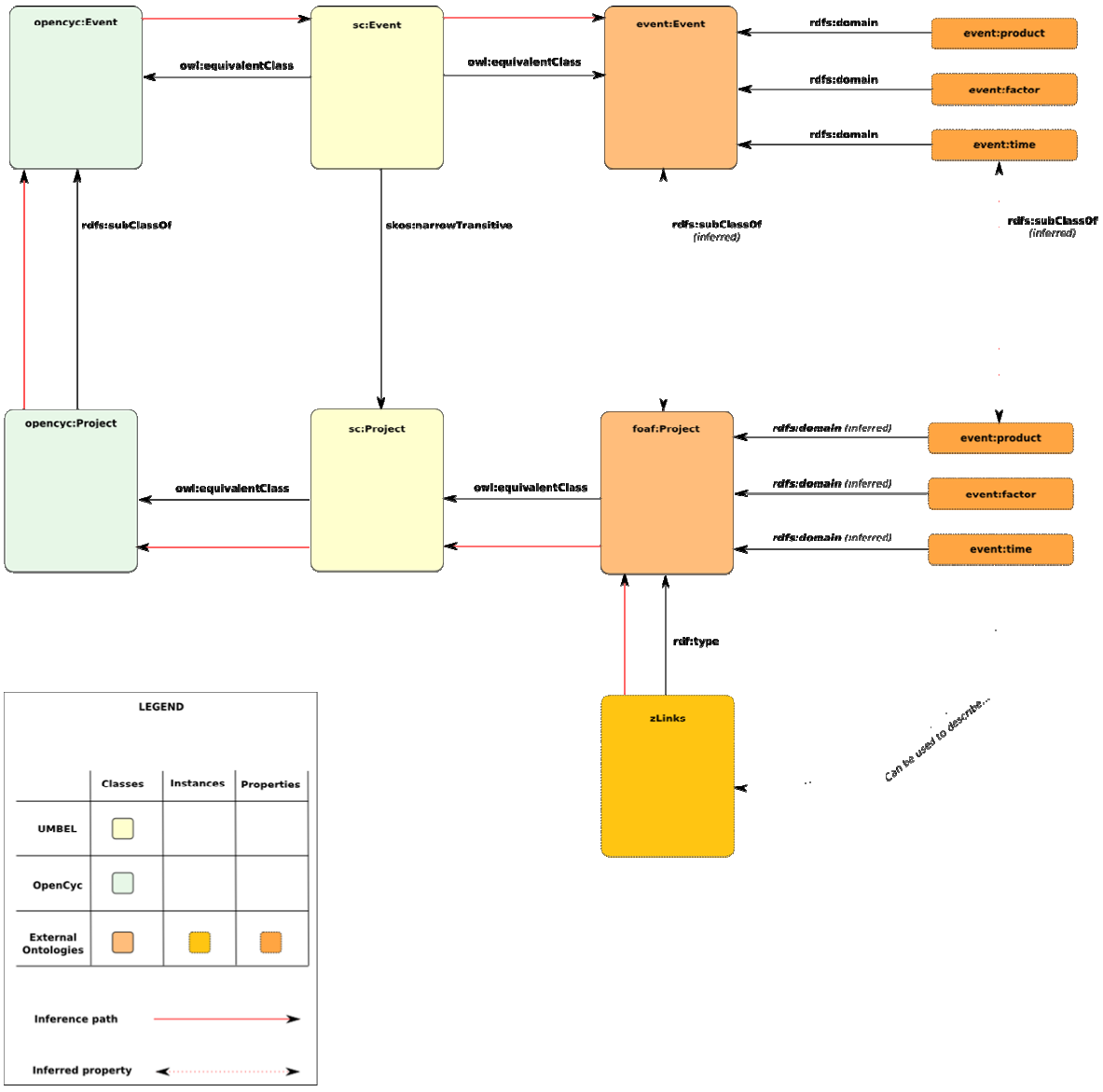

Figure 1. Inferencing with UMBEL

The example in *Figure 1* above shows the inference path (in red) between the class *foaf:Project* and

the class *event:Event*. The UMBEL subject concept structure defines that the subject concept *sc:Event* is a more general concept than the subject concept *sc:Project*. Since *event:Event* is an equivalent class to *sc:Event* and the *foaf:Project* is an equivalent class to *sc:Project*, we can infer that *event:Event* is a super-class of *foaf:Project*. This inferencing is consistent with the UMBEL subject concept structure.

In the *Inference* section below we will explain how the OpenCyc ontology is used to make such inferences possible and we will explain why the use of the properties *event:product*, *event:factor* and *event:time*, defined in the Event Ontology[5], can consistently be used on a *foaf:Project* within the UMBEL subject concept structure.

UMBEL is an Infocline

The Greek base *-cline* is often applied to gradual transition layers or changes in gradients or slope. A thermocline, for example, represents the layer between the deep and surface ocean. While there is mixing across this layer, it is slower than within the two parts that it separates. Both parts and the thermocline layer itself have quite different properties and temperatures, even though all are ocean and salty water.

The UMBEL ***infocline*** acts in a similar manner. On one side of the UMBEL layer is the Cyc knowledge base, with its self-contained, more-or-less closed world of higher order logics, microtheories regarding thousands of knowledge domains, rich predicates, and coherence. It is venerable, solid and proven, but with its own language and world view. Its purpose is also directed to reasoning and inference, driven from a foundation of (generally not codified outside of Cyc) common sense. It was designed well in advance of the creation of RDF or OWL, indeed in advance of the Internet and Web itself.

On the other side of the UMBEL ***infocline*** is the entire Web. This is a chaotic, decentralized, distributed knowledge environment representing untold numbers of world views. The specifications of the semantic Web and its languages and vocabularies have been designed expressly with these differences in mind and the means and structures to link and interrelate them. The Web environment — though not exactly incoherent — is also not in its ground state coherent. Indeed, it is the very purpose of existing semantic Web standards and UMBEL to help provide that coherence.

UMBEL thus must act as a mediator, or middleware, in its ***infocline*** role as the interface between these world views.[6]

Relation to the OpenCyc Knowledge Base

The central purpose of UMBEL is to provide a ***context*** for relating information. Once such a purpose for ***context*** is embraced, the natural next question is: And what shall be the basis for this ***context***?

The accompanying technical documentation describes why Cyc was chosen over alternatives as this contextual basis.[7] Ultimately, the reasons for choosing Cyc came down to real practical tools and capabilities such as helping to disambiguate the identities of named entities, mapping ontologies and schema, doing natural language processing, and its proven concept relationships.

However, the sheer size and sophistication of Cyc is too great for easy comprehension and linkage by standard Web resources. Thus, the UMBEL project set out to determine and derive the most fundamental concepts from within OpenCyc. What was desired was a tractable set of subject concept

"hub" nodes from within OpenCyc. A further design criterion was to maintain a 100% consistency with OpenCyc for this subset of subject concepts in order for UMBEL to preserve linkage into the Cyc knowledge base.

This review and vetting process reduced the overall size and complexity of Cyc by one to two orders of magnitude, resulting in a lightweight UMBEL structure about 5-10% of the original size consisting of a simple skein of 20,000 subject concepts and their interrelations. We can show this lightweight structure as a ball of subject concept nodes (in red) connected to one another via their graph edges.

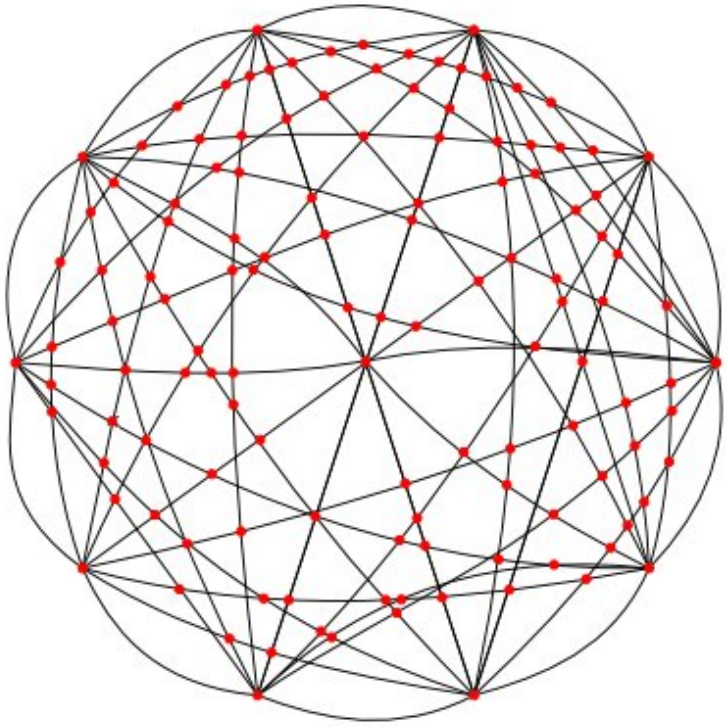

Figure 2. Simple UMBEL Wireframe Structure

Please refer to the *Distilling Subject Concepts from OpenCyc – Volume 3* [8] document for more information about the use of Cyc[9] and OpenCyc in the creation of UMBEL.

This UMBEL lightweight skein or wireframe structure is middleware in its role as an ***infocline***. Each of UMBEL's 20,000 subject concepts is, in effect, a "docking port" to which external Web data can "attach". Once attached, this data can then be related to other Web data via the subject concept relationships in the UMBEL skein.

Because each subject concept ("docking port") has a direct correspondence to Cyc, we can dive more deeply into the Cyc knowledge environment. First through OpenCyc and then (via licensing or other arrangements) into ResearchCyc or the full Cyc, another dimension of tools and capabilities can become available. We now have backup and support to assess mappings and assignments and inferences and reasoning.

Role and Use of Inverse Properties

Thus, the UMBEL skein has the purpose of being available for context reference by external Web data. In this regard, the UMBEL structure is largely passive. Other systems generally refer to it, rather than it directly linking or referring to external sources.

Attention has thus been given to the UMBEL ontology to provide appropriate inverse properties. These inverse properties of *superClassOf* (provided as the inverse predicate for the RDFS *subClassOf*), *linksConcept* and *linksEntity*, 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 discussed under the section, *Interacting with UMBEL – Tools*, below.

Named Entities and the UMBEL Structure

UMBEL thus provides a class structure of subject 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 subject concepts.[10]

These named entities are organized into one of more Named Entity Dictionaries that can be mixed and matched and related to one another via the subject concept structure. The initial set of UMBEL 1.5 million named entities come from Wikipedia.

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

OVERVIEW OF THE UMBEL ONTOLOGY

The UMBEL Ontology defines three classes: *SubjectConcept*, *AbstractConcept* and *Semset*; and nine properties: *superClassOf*, *hasSemset*, *isAligned*, *linksConcept*, *withAlignment*, *isAbout*, *linksEntity*, *isLike*, and *withLikelihood*.

The *Figure 3* below is split into three sections:

- The first section describes the relationship between the classes and properties defined in the UMBEL Ontology. Additionally, it reuses external ontologies properties of a few external ontologies such as SKOS, RDFS, OWL, Lingvoj and DCTERMS
- The second section shows how UMBEL subject concepts and semsets are described in UMBEL. Then it shows how external ontology classes are related to the subject concepts, and
- The third section shows how instances of the subject concept classes and their related semsets are defined in RDF.

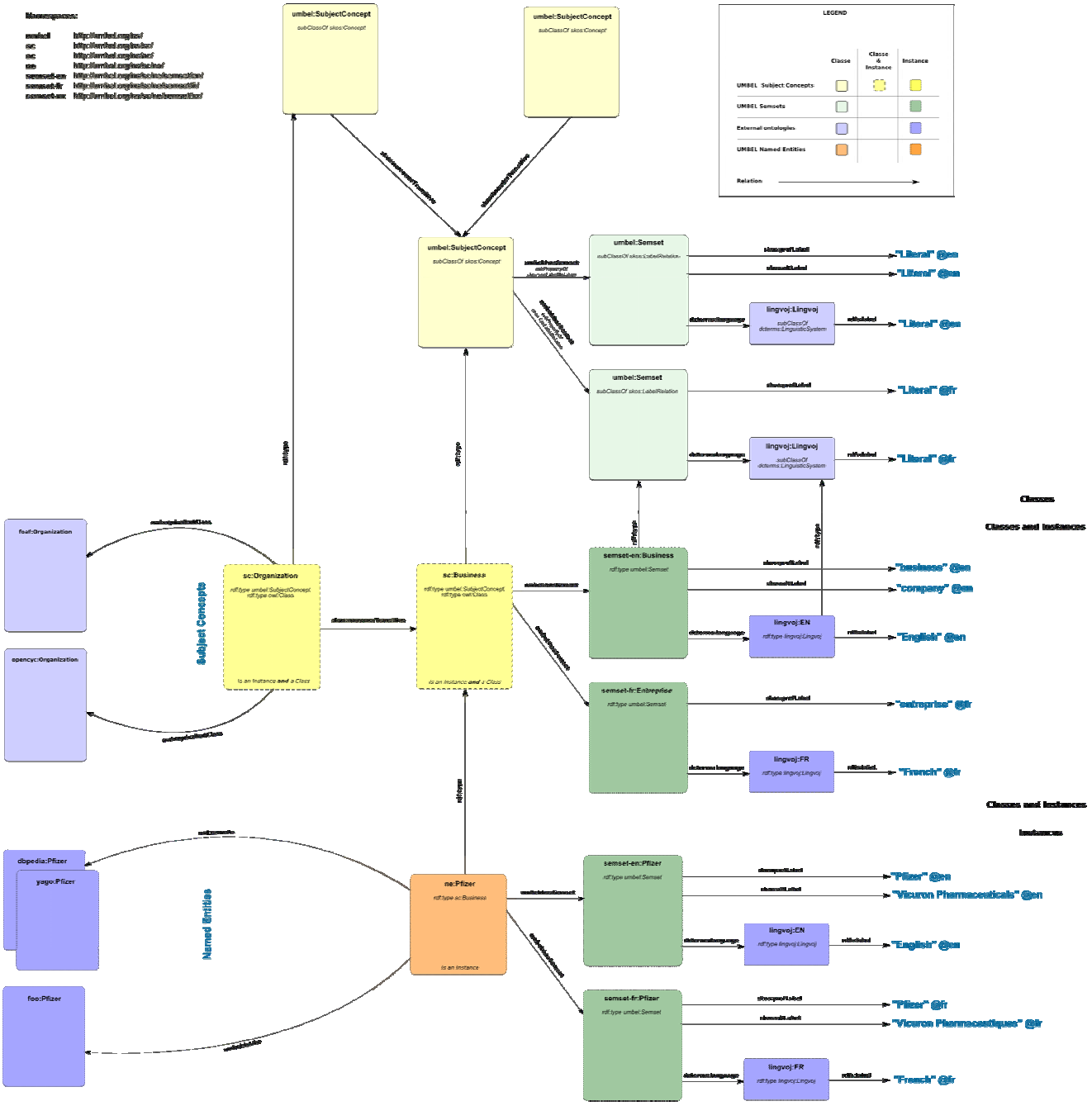

Figure 3. UMBEL Ontology[11]

Subject Concepts

Subject concepts are a special kind of concept: namely, ones that are concrete, subject-related and non-abstract. Note in other systems or ontologies, similar constructs may alternatively be called *topics*, *subjects*, *concepts* or perhaps *interests*. UMBEL has adopted the term *subject concept* to distinguish from these uses, which have different nuances of meaning and use, as well as to highlight the subject or topic nature of UMBEL's concrete concepts.

All subject concepts are a class and while they have a preferred label (using SKOS terminology), they are *representative* or a *proxy* for that concept, and not to be confused with the thing itself. Every UMBEL subject 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 subject concept itself. This labeling entity is known as *hasSemset* and is defined below. Each subject concept is related to at least one s*emset*.

Subject concepts are the core constituents to the UMBEL framework. **All subject concepts are based on existing concepts in OpenCyc**, the open source version of the Cyc knowledge base. About 20,000 of them have been distilled and are part of the UMBEL backbone.

The second section of the *Figure 3* above shows how these subject concepts are created and put in relation with other subject concepts or external ontology classes.

Subject Concepts are Classes and Individuals of a Class

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

Subject Concepts are Related to Semsets

All subject concepts are related to at least one *Semset* by the property *umbel:hasSemset*.

Subject Concepts are Related to Other Subject and Abstract Concepts

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

Subject Concepts are Linked to External Ontology Classes

Subject concepts are linked to external ontology classes by using properties such as *rdfs:subClassOf*, *owl:equivalentClass, umbel:superClassOf* and *umbel:isAligned*. Inferencing is also possible on this "extended" UMBEL subject concepts structure.

Subject Concepts are Linked to Individuals

Subject concepts are linked to individuals belonging to any external classes using the properties *umbel:isAbout* and *umbel:linksEntity*.

Abstract Concepts

Abstract concepts are the second new class added to the UMBEL vocabulary.

Abstract 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 concepts common to some upper ontologies.

Abstract concepts are a special kind of concept: they represent abstract or ephemeral notions such as truth, beauty, evil or justice, or are thought constructs useful to organizing or categorizing things but are not readily seen in the experiential world (such as, `PartiallyTangibleThing`). Abstract concepts are further contrasted with subject concepts and with named entities, which are the real things or instances in the world that are members of subject concept classes.

Abstract concepts are included in the UMBEL concept graph for inferencing and graph connectivity purposes, but can not themselves be displayed or used as binding classes in the UMBEL "backbone" for external classes or instances (or named entities).

Like all such distinctions, there is a degree of grayness or uncertainty to assigning a concept as either *subject-oriented* or *abstract*. An affirmative assignment as an *AbstractConcept* in UMBEL's was guided by some heuristic tests:

- Is the concept an arbitrary way to organize the concrete classes or things of the world, and nothing more and nothing less? If so, assign as *abstract*
- Is the concept unlikely to be a suitable concept for an *equivalentClass* mapping to an external ontology under open world assumptions? If so, assign as *abstract*
- Is the mental image or definition of the concept "squishy" and unlikely to be shared and understood broadly? Is It difficult or impossible to enumerate instances or set members of that concept class? If so, assign as *abstract*
- Is the concept apparently mostly useful for tying together or organizing other concrete concepts, rather than an identifiable concept of the world with an identity in its own right? If so, assign as *abstract*
- Does the concept seem to be an internal Cyc construct or one with unusually large numbers of connections within the Cyc knowledge base? If so, assign as *abstract.*

A complete listing of the roughly 800 *AbstractConcepts* used in UMBEL is provided in Appendix B of the accompanying volume, *Distilling Subject Concepts from OpenCyc, Vol. 3: Appendices*, **TR 08-07-16-B3.** Here are some additional considerations regarding this class:

Abstract Concepts are not Related to Semsets

All abstract concepts **are not** related to *Semsets*.

Abstract Concepts are Related to Other Subject and Abstract Concepts

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

Abstract Concepts are not Linked to External Ontology Classes

Abstract concepts **are not** linked to external ontologies.

Abstract Concepts are not Linked to Individuals

Abstract concepts **are not** linked to individuals belonging to any external ontology.

Semsets

Semsets are semantically close terms or phrases synonymous or nearly so with the meanings of a subject concept. *Semsets* are akin to WordNet *synsets* or Cyc *aliases*, but can also include more contemporary jargon or slang as may be drawn from Web tagging or folksonomies. (For example, Web 2.0, Web 20, web20, web_20, web-20, etc., can be expanded variants.) The term *semset* has been chosen to distinguish this consolidated meaning.

Semsets may apply to either *subject concepts* or *named entities*. In the latter case, their use is closer to the sense of an alias (such as nicknames, or "great Satan" or "uncle Sam" for the "United States").

Semset Description

Semsets are related to subject concepts and named entities by the property *umbel:hasSemset*. As shown in the *Figure 3* above, each semset has one, and only one, preferred label, which is the standard handle for referring to its governing subject concept or named entity. Alternatively, a semset can have zero or more alternative labels.

One important feature of a Semset is its relation to a language. Each semset is related to a language resource that defines the language in which the labels are written it. This means that a semset can be seen as a bag of related labels and that this bag of labels is related to a language. The goal is to relate a semset to a language instead of relating each label to the proper datatype. This means that each semset has a relation to a Lingvoj[12] instance that describe the language used to write labels belonging to the semset.

Named Entities

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

Please refer to the section *Using UMBEL to Describe Things* to see the description of the Muhammad Ali named entity, and refer to the section *Interacting with Named Entities and NE Dictionaries* to see how some named entities are defined, and used, within UMBEL.

Versioning

This first release of UMBEL is versioned at 0.70. The explanation of version numbering and rationale for this initial version number is provided in the accompanying volume, *Distilling Subject Concepts from OpenCyc, Vol. 1: Overview and Methodology*, **TR 08-07-16-B1.**

INFERENCING – *'EXPLODING THE DOMAIN'*

The goal of the UMBEL subject concept structure is to define a structure of upper concepts for a myriad of domains, and to link them to external ontologies (external conceptual structures). This creates contexts for *things* and kinds of *things*. The inferencing capabilities of this UMBEL subject concept structure extends the context of each subject concept.

Inferencing Implications

By inferring facts and creating a context for any given subject concept, we *explode their domain*. This explosion of facts gives us a much richer understanding of these subject concept classes and individuals belonging to these classes. *It is how and why UMBEL creates value*.

Subject Concepts Relations can be Inferred

UMBEL is a subject concepts structure. Hierarchical relations between concepts of this structure are described using the SKOS Ontology **based on the concept relationships already extant in the OpenCyc starting basis**. The SKOS data model enables us to infer facts between different subject concepts. We are using two different properties defined in the SKOS Ontology to describe two different hierarchical relationships between the UMBEL subject concepts:

1. *skos:narrowerTransitive* – used to say that a subject concept A is a more specific concept than subject concept B.
2. *skos*:*broaderTransitive* – used to say that a subject concept C is a more general concept than subject concept D.

By taking a look at the *Figure 4* below, we can infer that the subject concept *sc:Business* is a more specific concept than the subject concept *sc:CommercialOrganization* and the subject concept *sc:Organization*.

External Class Relationships can be Inferred

One of the characteristics of the UMBEL subject concept structure is to be able to link external ontology classes to UMBEL subject concept classes. This characteristic enables us to perform inferencing between external ontology classes using the UMBEL subject concept structure. This inferencing is possible according to the consistency/inconsistency principles introduced by the OWL-Full schema language.

The hierarchical relationship between UMBEL subject concepts and external ontology classes is done using two properties: *rdfs:subClassOf* and *umbel:superClassOf*. These properties are used to assert that an external ontology class is a sub-class of, or a super-class of, a UMBEL subject concept. The *umbel:superClassOf* property is the inverse of the *rdfs:subClassOf* property.

In OWL, the *rdfs:subClassOf* property is defined as:

> "[...] *if the class description C1 is defined as a subclass of class description C2, then the set of individuals in the class extension of C1 should be a subset of the set of individuals in the class extension of C2. A class is by definition a subclass of itself (as the subset may be the complete set).*"

In UMBEL, the *umbel:superClassOf* property is defined as:

> "[...] *is used as the inverse property of the property rdfs:subClassOf. If a class C' is a super-class of a class C, then all instances of C are also instances of C'.*"

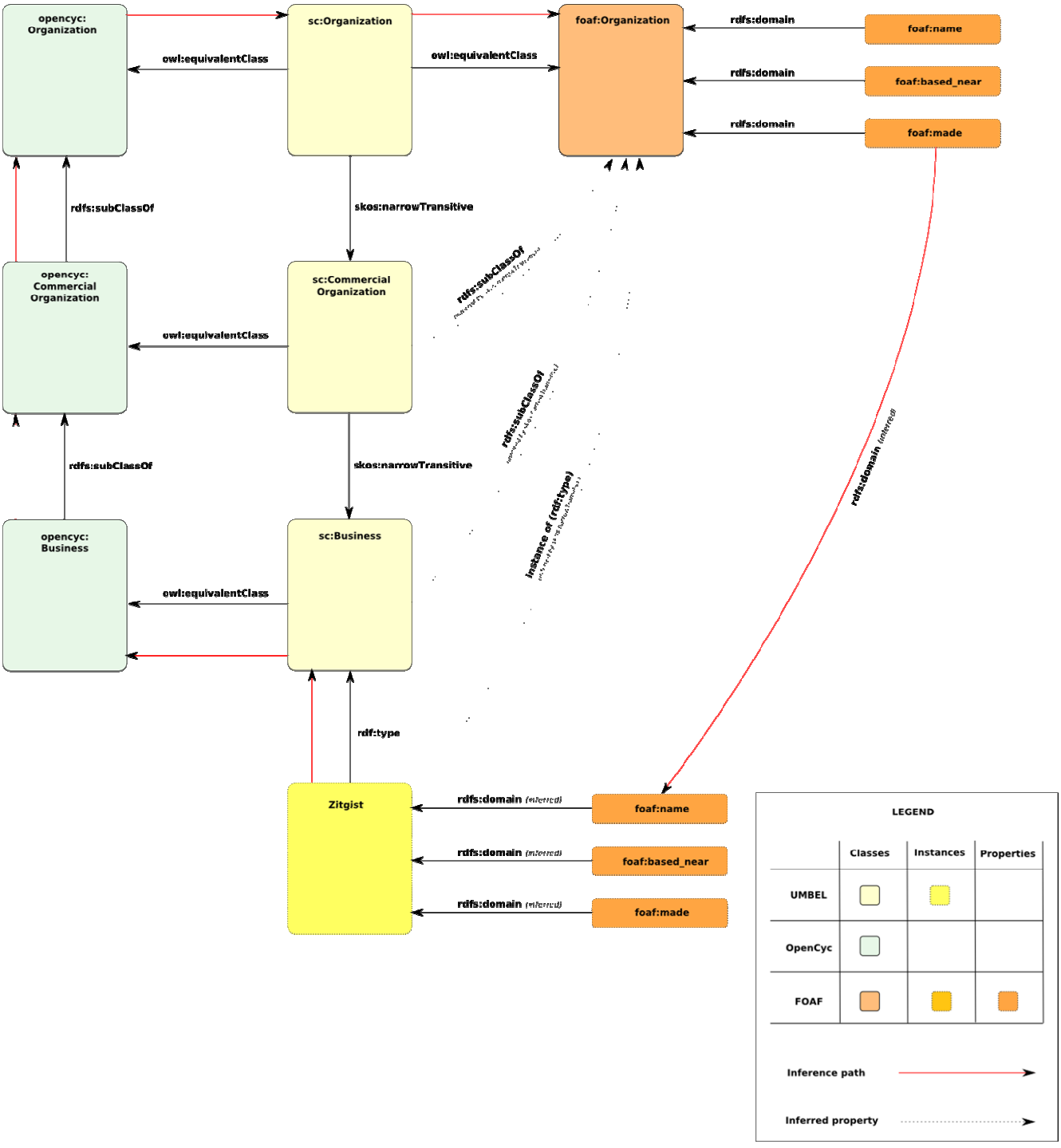

Figure 4. Basic Inference Example

Additionally we can define an equivalency relationship between an external ontology class with an

UMBEL subject concept class by using the *owl:equivalentClass* property. This let us assert that a class A, defined in an external ontology, is an equivalent class to an UMBEL subject concept class B.

In OWL, the *owl:equivalentClass* property is defined as:

> " [...] *owl:equivalentClass is a built-in property that links a class description to another class description. The meaning of such a class axiom is that the two class descriptions involved have the same class extension (i.e., both class extensions contain exactly the same set of individuals).* [...] *NOTE: The use of owl:equivalentClass does not imply class equality. Class equality means that the classes have the same intensional meaning (denote the same concept).* [...] ".

As applied to UMBEL this means that if an external ontology class *foo:Bar* is *owl:equivalentClass* with the UMBEL subject concept class *sc:Person*, then all individuals belonging to the class *foo:Bar* also belong to the class *sc:Person*.

These three properties are used below to prove the consistency of the inference of two external ontology classes using the UMBEL subject concept structure.

Finally we introduce a link relationship property between an external ontology class and an UMBEL subject concept class: the *umbel:isAligned* and its inverse property *umbel:linksConcept* properties. These properties are used to assert an associative link between a subject concept and a RDFS Class. This relationship can be described as a subset of individuals of the class A is equivalent to a subset of individuals of the class B. This asserts that there is a relation between an external ontology class and an UMBEL subject concept; but it doesn't say anything about the semantic nature of the relationship.

Re-use of External Ontology Properties

Once we are able to infer relationships between UMBEL subject concept classes and external ontology classes, we are then able to re-use the defined properties from these external ontologies to describe other instances of UMBEL subject concept classes. This inheritance of linked properties, so to speak, is where one of the main powers of the UMBEL subject concept structure resides.

Re-using properties to describe an individual of a subject concept class

One of the utilities of the UMBEL subject concept structure is that it defines the main concepts applicable to a myriad of domains. Since the current ontological space is weak, UMBEL can be used to describe *things* by using defined subject concepts when no specific ontologies exist for describing these *things*. The real power behind the inference capabilities of UMBEL between subject concept classes and external ontology classes is that it enables us to re-use external ontology properties to describe any instances of an applicable subject concept class.

Figure 4 above shows how an instance of the class *sc:Business* can re-use external properties defined in the FOAF Ontology to describe this instance. Please refer to *Table 11* below for the RDF/N3 code that defines this individual.

However, this approach begs the question: why do we use the *foaf:made* property to describe an individual belonging to the *sc:Business* class?

The clue resides in the domains and ranges of properties.

The domain of this property is a *foaf:Agent* and there are no apparent relationships between a *sc:Business* and a *foaf:Agent*. First we have to know that the class *foaf:Organization* is a sub-class of the class *foaf:Agent*. This means that individuals belonging to the class *foaf:Organization* are all included in the class extension of the indicated class description by the domain of the property *foaf:made*. This means that we can use *foaf:Organization* in the domain of the property *foaf:made*.

From there, we have to check the UMBEL subject concept structure to understand what is happening. The *foaf:Organization* class is equivalent to the *sc:Organization* class. By definition, this means all individuals belonging to both classes are the same. This therefore means that we can use the *sc:Organization* class in the domain of the *foaf:made* property too since it is the same class extension.

We have to keep one important thing in mind: all UMBEL subject concept classes are equivalent to their counterpart OpenCyc classes. It is these OpenCyc classes that describe the relationship, and enable the OWL-Full inference capabilities, between subject concept classes and external ontology classes.

By following the red path in *Figure 4* above, we notice that *sc:Business* is a sub-class of *sc:Organization*. Since we know that *sc:Organization* is an equivalent class with *foaf:Organization*, this means that *sc:Business* is a sub-class of *foaf:Organization*. This means that we can define an instance of the class *sc:Business* using properties where the *foaf:Organization* belongs to the class extension of the indicated class description of the domain, or range, of certain properties.

Formally, we can demonstrate that fact this way:

1. sc:Organization = foaf:Organization
2. sc:Organization = opencyc:Organization
3. opencyc:CommercialOrganization $\subseteq$ opencyc:Organization
4. opencyc:Business $\subseteq$ opencyc:CommercialOrganization
5. sc:Business = opencyc:Business

So we can infer that:

6. sc:Business $\subseteq$ opencyc:CommercialOrganization *(4, 5)*
7. sc:Business $\subseteq$ opencyc:Organization *(3, 6)*
8. sc:Business $\subseteq$ sc:Organization *(2, 7)*
9. sc:Business $\subseteq$ foaf:Organization *(1, 8)*

Given the inference step 9, we know that *sc:Business* is a subset of *foaf:Organization*. This means that all individuals belonging to *sc:Business* also belong to *foaf:Organization*.

The definition of *rdfs:domain* says: *"An rdfs:domain axiom asserts that the subjects of such property statements must belong to the class extension of the indicated class description."*

This means:

10. foaf:Organization $\subseteq$ *Class extension* of Properties X
    a. *Note: Properties X are properties such as foaf:made, foaf:name, etc.*
11. sc:Business $\subseteq$ *Class extension* of Properties X *(9, 10)*

In conclusion, we can use *Properties X* to describe instances of the class *sc:Business*.

These inference steps are consistent given the UMBEL subject concept structure and its defined linkage to external ontology classes.

Re-using properties to describe an Individual of a external ontology class

In the example above, we showed how we can use the inference capabilities of UMBEL to know which properties, defined in external ontologies, can be used to describe instances of subject concept classes.

Now we will see how we can use the same inferencing power to relate two *a priori* unrelated external classes to re-use properties defined in one ontology to describe the other.

Figure 5 below shows that, according to the UMBEL subject concept structure, the class *foaf:Project* is a sub-class of the class *event:Event*[13]. This means that we can re-use the properties *event:product, event:factor* and *event:time* to describe instances of the class *foaf:Project*. How is this possible?

Formally, we can demonstrate that fact this way:

1. sc:Event = event:Event
2. sc:Event = opencyc:Event
3. opencyc:Project ⊆ opencyc:Event
4. sc:Project = opencyc:Project
5. foaf:Project = sc:Project

Note: step 3 above is inferred. The inference path within OpenCyc is: opencyc:Project ⊆ opencyc:PurposefulAction ⊆ opencyc:Action ⊆ opencyc:Event. However we shortened the path of this step to opencyc:Project ⊆ opencyc:Event for convenience.

So we can infer that:

6. foaf:Project = opencyc:Project *(4, 5)*
7. foaf:Project ⊆ opencyc:Event *(3, 6)*
8. foaf:Project ⊆ sc:Event *(2, 7)*
9. foaf:Project ⊆ event:Event *(1, 8)*

Given the step #9, we know that *foaf:Project* is a subset of *event:Event*. This means that all individuals belonging to *foaf:Project* also belong to *event:Event*.

Given the definition of the *rdfs:domain* property, we then have:

10. event:Event ⊆ *Class extension* of Properties X
    a. *Note: Properties X are properties such as event:product, event:factor, event:time, etc.*
11. foaf:Project ⊆ *Class extension* of Properties X *(9, 10)*

In conclusion, we can use *Properties X* to describe instances of the class *foaf:Project*.

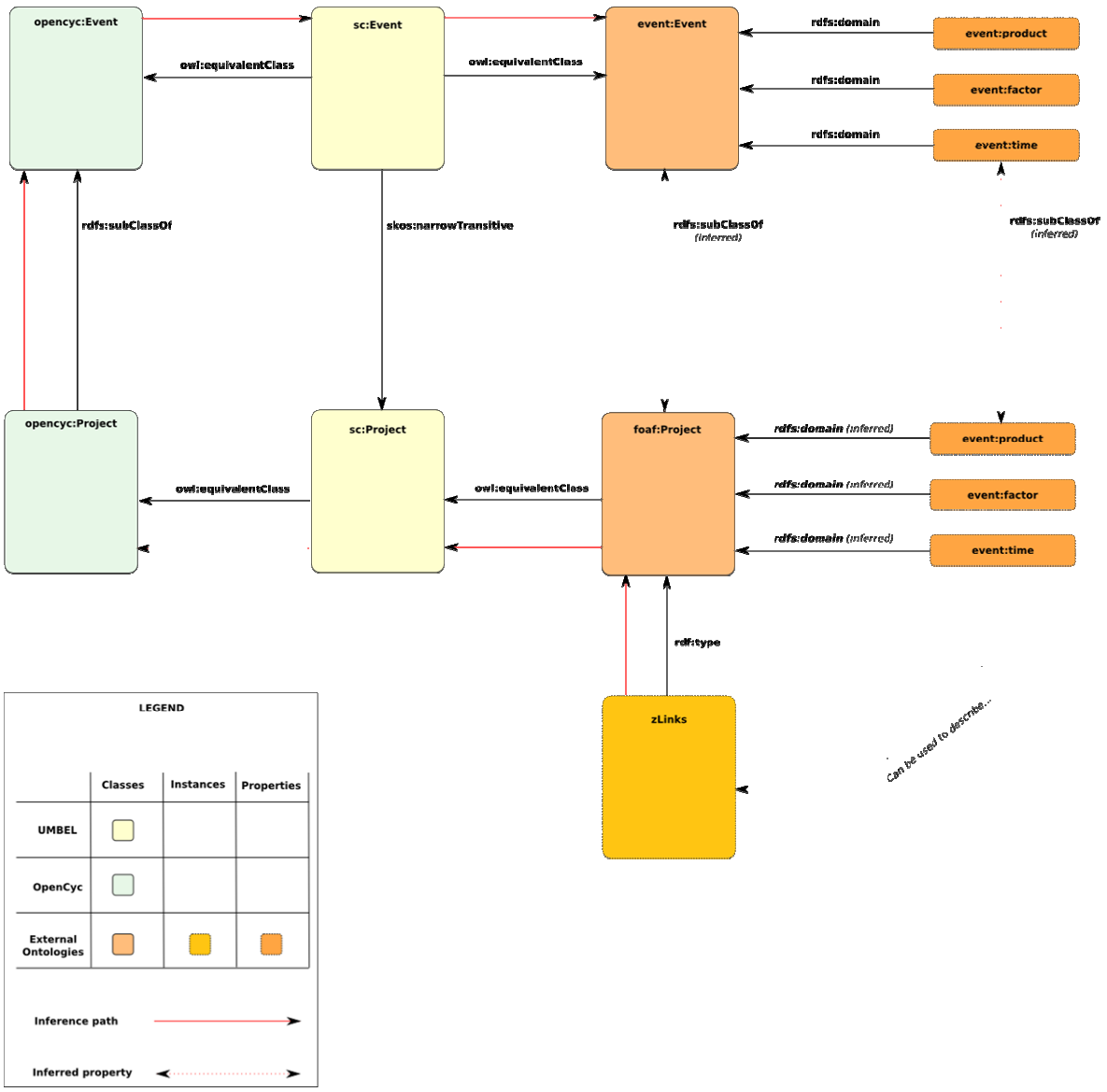

Figure 5. Re-use of Properties on External Ontology Classes

Again, these inference steps are consistent given the UMBEL subject concept structure and its defined linkage to external ontology classes.

Caveats for Class Relationship Assignments

Clearly, these powers may be easily misused and mis-applied if the actual class relationships do not meet the required instance membership requirements. Great care must be made in analyzing carefully these three types of class relationships.

Fortunately, once made, these relations occur between UMBEL and external ontologies and will

pertain to all uses of those external ontologies. Also, only a very few assignments need to emerge in order to see the exploding the domain phenomenon.

For the dozen or so external ontologies related to UMBEL to date, there have been on average roughly two *equivalentClass* and two *subClassOf* assignments per ontology. This relatively small degree of linkage has a multiplier effect, however, that also increases as a function of the number of linked external ontologies.

INTERACTING WITH UMBEL

There are many ways to interact with the UMBEL subject concept structure. This section describes three ways to interact with UMBEL: (1) by linking external classes to UMBEL subject concepts (2) by using subject concepts to describe *things* and (3) to help the development of new ontologies.

Linking External Ontologies Classes to UMBEL Subject Concepts

In the sections above, we described the inferential benefits of linking external ontology classes to UMBEL subject concepts. This linkage gives us the possibility to re-use external ontology classes and properties to describe instances of subject concept classes and instances of external ontology classes.

However, to make this happen, we needed to create the linkage between these external ontology and subject concept classes. Additionally, this linkage had to be consistent with the UMBEL structure.

Tools

We have created a series of Web services to help ontology developers and maintainers to link external classes to subject concept classes. These Web services help find possible related subject concepts, view their relationship with other external ontologies, and infer facts to validate the consistency of the UMBEL structure when creating a new linkage.

The series of UMBEL Web services[14] are defined in three main categories:

Finding Subject Concepts

1. Find Subject
    a. This Web service is used to find UMBEL subject concepts that match a search string. This is the primary tool for finding available subject concepts.

Visualizing/Exploring Subject Concepts

2. Subject Concept Report
    a. This Web service is used to get basic information about a specific UMBEL subject concept. The information shown in the report is a definition of the subject concept and its semset related to that concept. This semset is an aggregation of different labels with similar meaning that is used in practice to designate the concept.
3. Subject Concept Detailed Report
    a. This service provides detailed information about a specific UMBEL subject concept. This user interface is a special kind of service. It uses all existing UMBEL Web services to create a detailed report about a UMBEL subject concept and its relations to other UMBEL subject concepts, external ontology classes, and existing named entities from different kinds of data sources.

4. Subject Concepts Explorer
    a. This service is a visualization interface that lets you browse the UMBEL graph of subject concept relationships. You can navigate from one node to another by clicking any of the circles. Each circle is an UMBEL subject concept or an external ontology class.

Inferring Subject Concepts

5. List Sub-Concepts & Sub-Classes
    a. This Web service is used to get a list of more specific concepts (UMBEL subject concepts or external ontology classes) for a given UMBEL subject concept URI or an external ontology class URI.
6. List Super-Concepts & Super-Classes
    a. This Web service is used to get a list of more general concepts (UMBEL subject concepts or external ontology classes) for a given UMBEL subject concept URI or an external ontology class URI.
7. List Equivalent External Classes
    a. This Web service is used to get a list of equivalent concepts (UMBEL subject concepts or external ontology classes) for a given UMBEL subject concept URI or an external ontology class URI.
8. Verify Sub-Class Relationship
    a. This Web service is used to get a "true" or "false" answer to the question: is this UMBEL subject concept URI or external ontology class URI a sub-concept of this other UMBEL subject concept URI or external ontology class URI?
9. Verify Super-Class Relationship
    a. This Web service is used to get a "true" or "false" answer to the question: is this UMBEL subject concept URI or external ontology class URI a sub-concept of this other UMBEL subject concept URI or external ontology class URI?
10. Verify Equivalent Class Relationship
    a. This Web service is used to get a "true" or "false" answer to the question: is this UMBEL subject concept URI or external ontology class URI equivalent to this other UMBEL subject concept URI or external ontology class URI?

Method

The method consists of a series of steps to guarantee the consistency of the UMBEL structure once the linkage between an external class and a subject concept class is made. To help users to perform these steps, we will illustrate with the use of the Web services that have been described in the section above.

Step 1: Search for a related subject concept

When you try to link an external ontology class <A> to the UMBEL subject concepts structure, the first thing you have to do is to try to find a related subject concept.

To find a list of potential candidates, we suggest using the *Find Subject* Web service to find a list of subject concepts that can use the search label to denote the subject concept. Most of the time, we use the name of a class, and some of its synonyms, to find potential subject concept candidates. Additionally, we can use the *Subject Concept Detailed Report* to check the list of more general and more specific subject concepts related to the subject concepts returned by the *Find Subject* Web service. Inspecting this list helps refine the search for the right subject concept.

Once we have a list of such potential candidates, we have to find the right one to make the linkage.

Step 2: Find the right subject concept

To find the right subject concept class <B> to link to a class <A>, we have to determine the right relation that exists between class <A> and a given subject concept. There exists three different kind of relations, and not all have the same importance. The relationships, in order of importance, are:

1. Equivalence relationship; *owl:equivalentClass*
2. Sub-class of relationship; *rdfs:subClassOf, umbel:superClassOf*
3. Part-of relationship; *umbel:isAligned, umbel:linksConcept*

The subject concept that shares the highest importance relationship with the class <A> will be linked to it.

To determine the relation that applies, we apply this procedure:

- Do all individuals that belong to the subject concept class <B> also belong to the class <A>?
  - If yes, do all individuals that belong to the class <A> also belong to the subject concept class <B>?
    - If yes, then the two classes are equivalent
    - If no, then the subject concept class <B> is a sub-class of class <A>
  - If no, do all individuals of the class <A> belong to the subject concept class <B>?
    - If yes, then the class <A> is a sub-class of the subject concept class <B>
    - If no, is there a non-empty intersection between the class <A> and the subject concept class <B>?
      - If yes, then the class <A> is-about the subject concept class <B> given a certain confidence percentage value
      - If no, there is no relationship between the class <A> and the subject concept class <B>.

Additionally, we have to read the description of the class <A> and the subject concept class <B> to make sure that both classes share the same semantic meaning. This description is normally written in the documentation of the ontologies.

Once we think we have found the right subject concept class <B> to link to the class <A> (if any candidates do indeed exist), we next have to make sure that the UMBEL data model remains consistent after making the linkage.

Step 3: Analyze the relationship between the two classes

This third step is performed to make sure that the UMBEL subject concept structure remains consistent after asserting that the class <A> is related, in some way, to the subject concept class <B>.

The analysis will differ depending on the kind of relationship that exists between the class <A> and the subject concept class <B>.

This analysis is based on the OWL-Full description of the external ontology classes and UMBEL subject concept classes. As we noted in the *Inference and Open-World Assumption* section above,

we can only conclude things according to what is known (so what is defined in these different ontologies).

If we state that <A> is equivalent to <B>, then the following assertions have to be true:

1. All individuals that belong to <A> also belong to <B>
2. All individuals that belong to <B> also belong to <A>
3. All sub-classes of <A> have to be sub-classes of <B>
4. All sub-classes of <B> have to be sub-classes of <A>
5. All super-classes of <A> have to be super-classes of <B>
6. All super-classes of <B> have to be super-classes of <A>
7. All equivalent classes of <A> have to be equivalent classes of <B>
8. All equivalent classes of <B> have to be equivalent classes of <A>.

If any of these assertions is ***false***, then <A> is not equivalent to <B> and the linkage is dropped.

If we state that <A> is a sub-class of <B>, then the following assertions have to be true:

1. All individuals that belong to <A> also belong to <B>
2. All super-classes of <B> have to be super-classes of <A>

If any of these assertions is false, then <A> is not a sub-class of <B> and the linkage is dropped.

If we state that <A> is-aligned with <B>, then the following assertion has to be true:

1. Individuals that belong to the intersection of <A> and <B> should not belong to any set of disjoint set of individuals with neither <A> nor <B>.

If this assertion is false, then <A> is not aligned with <B> and the linkage is dropped.

The analysis of the OWL-Full class definition, and the current linkage between UMBEL subject concepts and other external ontology classes, will tell us if one of these assertion is true, or false.

The critical analysis task thus determines, according to what is defined within the UMBEL subject concept structure, if these relationships are true or false.

Step 4: Make the linkage

Once we determine which relation holds with which subject concept class, we can assert this fact in RDF, such that:

- <A> owl:equivalentClass <B>
- <A> rdfs:subClassOf <B>
- <A> umbel:isAligned <B>

Then the linkage is published via the UMBEL Linkage Ontology Extension, or via an external ontology definition.

See *Appendix A: Listing of Linked External Ontologies* for the listing of the current 12 external ontologies linked to UMBEL.

Interacting with Named Entities and NE Dictionaries

The section *Using UMBEL to Describe Things* below talks about how one can use UMBEL subject concepts to describe things on the semantic Web. In this section we are interested in how UMBEL defines dictionaries of named entities and how these dictionaries can be used.

Zitgist uses publicly available data sources such as Wikipedia (via its Yago and DBpedia incarnations) and others to create and publish UMBEL named entities dictionaries. These dictionaries are composed of UMBEL named entities (individuals belonging to UMBEL subject concept classes) and their links to numerous identities for a given named entities. This linkage is created and published by Zitgist. However the same principles described in this section can be used by other organizations or individuals to create other named entities dictionaries that use their own linkage methodologies and procedures.

The Goal

The goal is to link external named entities together. The result of this linkage is to aggregate identities of a same thing together. For example, this named entity would link all the Muhammad Ali identities, described in a myriad of external data sources, together:

```
<http://umbel.org/ns/sc/ne/example/> a sc:Boxer ;
      owl:sameAs <http://www.ali.com/me/> ;
      owl:sameAs <http://dbpedia.org/resource/Muhammad_Ali> ;
      owl:sameAs <http://www.mpii.de/yago/resource/Muhammad_Ali > ;
      umbel:isLike <http://yet-another-ne.com/resource/Muhammad_Ali_junior> ;
      umbel:hasSemset <http://umbel.org/ns/sc/ne/semset/example/> .
```

Table 1. Muhammad Ali – the linkage of identities

The Method

Such UMBEL named entities have three main characteristics:

1. They are individuals belonging to one or more UMBEL subject concept class(es).
2. They are linked to individuals of external ontologies classes with the properties *owl:sameAs* or *umbel:isLike*.
3. They are related to a Semset by the property *umbel:hasSemset*.

Named entities created with such a procedure are published in the namespace:

http://umbel.org/ns/sc/ne

Normally one named entities dictionary is created for each data source. For example, named entities that come from Wikipedia are published under this namespace:

*http://umbel.org/ns/sc/ne/*wikipedia/

Also note that each named entity is mapped to a governing subject concept for ontology purposes. As noted, named entities may also have semset aliases.

The third section of the *Figure 3* above shows how named entities relate to subject concepts, to other named entities and how they in turn relate to their associated semset.

USING UMBEL

UMBEL can be use in a myriad of ways; here we describe some of them, but its utility is not limited to them.

Using UMBEL to Describe Things

UMBEL can be used to describe *things*. Considering the current sparse nature of the ontology space, UMBEL can be used to fill some gaps. The goal is to use UMBEL subject concepts to describe certain individuals belonging to their governing subject concept classes.

By using UMBEL, you can find a subject concept related to any *thing* you may wish to describe. Once the proper subject concept is found (if it exists), you can then check the structure to try to re-use classes and properties defined in external ontologies to describe that *thing* you want to describe in RDF.

In such a scenario, it is the "upper ontology" facet of UMBEL that is exploited. There is an example:

We want to describe the person of Muhammad Ali. However, we don't want to describe him only as a simple person; since he is the best **boxer** of the 20th century, we want to describe him as a **boxer**! But we can't find any "boxing ontology". Normally, this would mean that we would have to describe Ali as a person (*foaf:Person*), or we would have to define a new "boxing" ontology. But, with the capabilities of UMBEL, we can now search for a subject concept "boxer" in UMBEL and use it as well to describe Muhammad Ali.

If we use the *Find Subject* UMBEL Web service, we will find that the subject concept "boxer" exists in UMBEL[15]. We know that a *sc:Boxer* is a *foaf:Person*. We have a list of properties, defined in external ontologies, that we can use to describe Ali. This means that we can describe Muhammad Ali this way:

```
<http://www.ali.com/me/> a sc:Boxer ;
      foaf:name "Muhammad Ali" ;
      foaf:gender "male" ;
      foaf:birthday "1942-01-17" ;
      ...
```

Table 2. Muhammad Ali – the Boxer

This is a good example of how UMBEL can be used as an "upper ontology" to fill the gaps when specialized ontologies don't exist for a given domain.

Using UMBEL to Help Develop New Ontologies

The notion of inconsistency introduced by the use of OWL-Full applied the UMBEL subject concept structure is quite useful when the need arises to develop new ontologies. UMBEL helps to learn more about a domain of knowledge, and that then leads to develop better ontologies.

Since that most of the time an ontology interacts with external ontologies by referring to them, or by re-using some of their classes and properties, having a consistent framework that puts all the ontologies in context helps a lot to define new ontologies and to fix existing ones.

The SKOS specification document[16] says:

> "*When OWL Full is used as a data modeling (i.e., schema) language, the notion of inconsistency is again useful, but in a different way. Here we are not concerned with the logical consistency of human knowledge itself. We are simply interested in formally defining a data model, so that we can establish with certainty whether or not some given data conforms to or "fits" with the given data model. If the data is inconsistent with respect to the data model, then it does not "fit".*"

The OWL-Full schema language helps us to link external ontology classes to UMBEL subject concepts. The consistency/inconsistency concepts are useful to find if a linkage fits, or not, with the defined UMBEL subject concept structure. The UMBEL Web services and the linkage method helps to develop new ontologies and to fix existing ones.

Using UMBEL to Put Individuals in Context

An individual belongs to a class. Since UMBEL is a subject concept structure of well-defined classes with links to classes defined in external ontologies, we can learn a lot about an individual simply by looking-up its type(s) in UMBEL.

This is what we sometimes call the "context" of an individual.

Let's again take the example of Muhammad Ali. Think about someone that gets the RDF description of Muhammad Ali shown in *Table 2* above. This person may not know what a "boxer" really is. Then he chooses to look for the type *sc:Boxer* in UMBEL[17]. Only with this search within UMBEL, he knows that: a *sc:Boxer* is a *foaf:Person*, a *foaf:Agent* and a *cyc:Boxer*. He also finds that Antonio Ayala Jr., Galveston Giant and John Ruiz were boxers too. He now knows that Ali is an *athlete* and a *social being*; and so on. With a single lookup in UMBEL, the context around "Muhammad Ali" starts to emerge and its *domain explodes*.

The Ali example took a subject concept as the lookup basis. However, we could also take an external ontology class to get the same result.

Try the same experience with a *po:Radio*[18]. We find that it is a *sc:RadioStation-Organization*[19], a *foaf:Organization*; we find that there exist different types of radio stations such as local radios, regional radios and national radios; we find that WNSC, WRHS, WKHX, WALR-FM are radios organizations, etc. In a single lookup, we get the context of a class described in an external ontology and we *explode its domain*.

SPECIFICATION DOCUMENTATION

The UMBEL Ontology defines three new classes: SubjectConcept, AbstractConcept and Semset; and nine properties: superClassOf, hasSemset, isAligned, linksConcept, withAlignment, isAbout, linksEntity, isLike, and withLikelihood. These classes and properties are used to instantiate the

UMBEL subject concept structure, and to link subject concepts to external ontology classes. Below we describe each of these classes and properties.

Namespaces

There are the URIs of the namespaces used to describe the UMBEL Ontology, the subject concepts structure, the named entities defined in UMBEL and the semsets for both the subject concept classes and named entities.

Name	Abbreviation	URI
UMBEL Ontology	umbel:	http://umbel.org/ns/
Subject Concepts	sc:	http://umbel.org/ns/sc/
Abstract Concepts	ac:	http://umbel.org/ns/ac/
Named Entities	ne:	http://umbel.org/ns/sc/ne/
Semsets	semset-*xyz*	http://umbel.org/ns/sc/ne/semset/*xyz*/
Example, English semset	semset-en	*http://umbel.org/ns/sc/ne/semset/en/*

Table 3. UMBEL Namespace

Classes

Class name	umbel:SubjectConcept
Description	Subject 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 concepts common to some upper ontologies. Subject concepts are a special kind of concept: ones that are concrete, subject-related and non-abstract. We further contrast these with named entities, which are the real things or instances in the world that are members of these subject concept classes. The UMBEL "backbone" is this set of reference subject concepts.
In-domain-of	umbel:hasSemset, umbel:linksConcept, umbel:linksEntity
In-range-of	umbel:isAbout, umbel:isAligned
Sub-class-of	skos:Concept
Disjoint-with	umbel:Semset, umbel:AbstractConcept
Status	Stable

Table 4. Subject Concept Class

Class name	umbel:AbstractConcept
Description	Abstract 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 concepts common to some upper ontologies. Abstract concepts are a special kind of concept: they represent abstract or ephemeral notions such as truth, beauty, evil or justice, or are thought constructs useful to organizing or categorizing things but are not readily seen in the experiential world (such as, PartiallyTangibleThing). Abstract concepts are further contrasted with subject concepts (see their class description) and with named entities, which are the real things or instances in the world that are members of subject concept classes. Abstract concepts are included in the UMBEL concept graph for inferencing and graph connectivity purposes, but can not themselves be displayed or used as binding classes in the UMBEL "backbone" for external classes or instances (or named entities).
Sub-class-of	skos:Concept
Disjoint-with	umbel:Semset, umbel:AbstractConcept
Status	Stable

Table 5. Abstract Concept Class

Class name	umbel:Semset
Description	Semsets are semantically close terms or phrases synonymous or nearly so with the meanings of a subject concept. Semsets are akin to WordNet synsets or Cyc aliases, but can also include more contemporary jargon or slang as may be drawn from Web tagging or folksonomies. The term semset has been chosen to distinguish this consolidated meaning. Semsets may apply to either subject concepts or named entities. In the latter case, their use is closer to the sense of an alias (such as nicknames, or "great Satan" or "uncle Sam" for the "United States").
In-range-of	umbel:hasSemset
Sub-class-of	skos:LabelRelation
Disjoint-with	umbel:SubjectConcept, umbel:AbstractConcept
Status	Stable

Table 6. Semset Class

Properties

Property name	umbel:superClassOf

Description	The property umbel:superClassOf is used as the inverse property of the property rdfs:subClassOf. If a class C' is a super-class of a class C, then all instances of C are also instances of C'.
Domain	rdfs:Resource
Range	rdfs:Resource
Inverse-of	rdfs:subClassOf
Status	Stable
Note	This property is used to explicitly denote a super-class property of a subject concept resource to an external ontology class.

Table 7. superClassOf Property

Property name	umbel:hasSemset
Description	Link a subject concept to its Semset.
Domain	umbel:SubjectConcept
Range	umbel:Semset
Sub-property-of	skos:seeLabelRelation
Status	Stable

Table 8. hasSemset Property

Property name	umbel:isAligned
Description	The property umbel:isAligned is used to assert an associative link between a subject concept and a RDFS Class. This relationship can be described as a subset of individuals of the class A is equivalent to a subset of individuals of the class B. This means that there is a Subset(A) equivalent to a Subset(B) when a <A> <umbel:isAligned> <B>. So there exists a class C that is the intersection of A and B. This is the formal definition of two linked classes. This property is used to link an external ontology class to an UMBEL subject concept class when neither the owl:equilaventClass nor the rdfs:subClassOf can be applied but where umbel:isAligned applies. umbel:isAligned allows one to say that the subset of a class extension (set of instances) X of a class description is equivalent to the subset of a class extension Y of another class description. Great care has to be taken when using this property between two class descriptions. No owl:disjointWith property should be defined, or inferred, between the two class descriptions in

	order to keep the mapping consistent.
Domain	rdfs:Class
Range	umbel:SubjectConcept
Inverse-of	umbel:linksConcept
Status	Experimental - Unstable

Table 9. isAligned Property

Property name	umbel:linksConcept
Description	Check the definition of umbel:isAligned for the definition of this property; linksConcept is the inverse property of isAligned.
Domain	umbel:SubjectConcept
Range	rdfs:Class
Inverse-of	umbel:isAligned
Status	Experimental – Unstable

Table 10. linksConcept Property

Property name	umbel:withAlignment
Description	This property is used to reify a umbel:isAligned or a umbel:linksConcept property to a calculated or estimated overlap percentage value between the two classes (sets).
Domain	rdf:Statement
Range	rdfs:Literal
Status	Experimental – Unstable

Table 11. withAlignment Property

Property name	umbel:isAbout
Description	The property umbel:isAbout is used to assert the relation between a named entity (individual) and a subject concept class. umbel:isAbout relates the named entity (individual) 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 subject concept class may not alone bound or define the individual entity. Named entitles may be related with multiple subject concept classes. The domain of umbel:isAbout defines its class description as the class of all individuals (owl:Thing) and its range as the class of subject concepts (umbel:SubjectConcept), thereby bounding the property's

	proper semantics of associating individuals to their related subject concept class(es). This property is therefore used to create a topical assertion between an individual and a subject concept.
Domain	owl:Thing
Range	umbel:SubjectConcept
Inverse-of	umbel:linksEntity
Status	Experimental - Unstable

Table 12. isAbout Property

Property name	umbel:linksEntity
Description	Check the definition of umbel:isAbout for the definition of this property; linksEntity is the inverse property of isAbout.
Domain	umbel:SubjectConcept
Range	owl:Thing
Inverse-of	umbel:isAbout
Status	Experimental – Unstable

Table 13. linksEntity Property

Property name	umbel:isLike
Description	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 can and should be changed if the certainty of the sameness of identity is subsequently determined. In general, we may not be able to assert that two individuals are the same based solely on current information on hand. However, there may be quite reasonable bases or methods that the two individuals are **likely** the same without being one hundred percent sure. umbel:isLike 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). 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 certitude.
Domain	owl:Thing
Range	owl:Thing
Status	Experimental - Unstable

Table 14. isLike Property

Property name	umbel:withLikelihood
Description	This property is used to reify a umbel:isLike property to an likelihood percentage value between the two individuals.
Domain	rdf:Statement
Range	rdfs:Literal
Status	Experimental – Unstable

Table 15. withLikelihood Property

RE-USE OF EXTERNAL ONTOLOGIES

The UMBEL ontology re-uses several properties and individuals of classes defined in external ontologies to instantiate and describe the UMBEL subject concepts. This is a list of such properties that are or can be used to describe the subject concepts.

Properties

Properties of the Dublin Core[20] ontology:

- dcterms:language

Properties of the SKOS ontology:

- skos:prefLabel
- skos:altLabel
- skos:hiddenLabel
- skos:note
- skos:changeNote
- skos:definition
- skos:editorialNote
- skos:example
- skos:historyNote
- skos:scopeNote
- skos:broader
- skos:narrower
- skos:related
- skos:broaderTransitive
- skos:narrowerTransitive

Properties of the RDFS ontology:

- rdfs:subClassOf

Properties of the OWL ontology:

- owl:equivalentClass

Instances of Classes

All Lingvoj Language instances (such as lingvoj:EN, lingvoj:FR, etc) are re-used by UMBEL. These instances are used to refer each subject concept semset to the language they are written in.

EXAMPLES

We provide some examples below that show how the subject concepts, the semsets and the named entities are described in RDF and serialized in N3[21].

Subject Concept Description

This example is a sample of the *UMBEL Ontology, Vol. 2: Subject Concepts and Named Entities Instantiation* document[22]. This is the RDF description of the *Project* subject concept. Linkage between UMBEL subject concepts is performed in a hierarchical way using properties *skos:broaderTransitive* and *skos:narrowerTransitive*.

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

Table 16. Subject Concept Description Example

Semset Description

This example is a sample of the UMBEL Ontology Instantiation document [22]. This is the RDF description of the *Project* subject concept semset. This semset is related to the *sc:Project* subject concept by using the *umbel:hasSemset* property.

```
semset-en:Project a umbel:Semset ;
                        skos:prefLabel """project"""@en ;
                        skos:altLabel """projects"""@en ;
                        skos:altLabel """undertakings"""@en ;
                        skos:altLabel """undertaking"""@en ;
                        skos:altLabel """enterprises"""@en ;
                        skos:altLabel """enterprise"""@en ;
                        skos:altLabel """programz"""@en ;
                        skos:altLabel """programs"""@en ;
                        skos:altLabel """program"""@en ;
                        skos:altLabel """programme"""@en ;
                        dcterms:language <http://www.lingvoj.org/lingvo/en> .
```

Table 17. Semset Description Example

Named Entity Instantiation

This example shows how the subject concept *sc:Business* is used to describe the Zitgist 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.

As we saw in the *Figure 4* above, a *sc: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 *sc:Business.*

```
<http://zitgist.com/about/> a sc:Business ;
    foaf:name "Zitgist LLC." ;
    foaf:birthday "2006-10-20" ;
    foaf:logo <http://zitgist.com/imgs/zitgistlogo2_110_55.gif> ;
    foaf:fundedBy <http://www.openlinksw.com> ;
    bio:olb "Zitgist (pronounced "zeitgeist") is an industry standards compliant Semantic
    Web Query Service. Its goal is to help Web users locate data, information, and
    knowledge on the Web. Zitgist is based on a new search paradigm: users describe
    characteristics of their search target, instead of relying entirely on content keywords."
    @en ;
    foaf:based_near [ geo:Point [geo:lat "42.455", geo:long "-71.218"] ] ;
    foaf:homepage <http://zitgist.com> ;
    foaf:phone "+1 781 273 0900" ;
    foaf:made <http://browser.zitgist.com/about/> ;
    foaf:made <http://pingthesemanticweb.com/about/> ;
    foaf:made <http://musicontology.com/about/> ;
    foaf:made <http://bibliontology.com/about/> ;
    foaf:made <http://talkdigger.com/about/> .
```

Table 18. Named Entity Instantiation Example

Linkage Between External Classes and Subject Concepts

There are three ways to describe the relation between a subject concept and an external ontology class. An example for each way is described in the *Table 12* below.

The first example shows how the *owl:equivalentClass* property is used. Then we see how properties *umbel:superClassOf* and *rdfs:subClassOf* are used.

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

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

<http://umbel.org/ns/sc/SpatialThing>
<http://umbel.org/ns/superClassOf>
<http://www.geonames.org/ontology#Feature> .

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

Table 19. equivalentClass, superClassOf and subClassOf Examples

Another way to link an external ontology class to an UMBEL subject concept is to use the umbel:isAligned property and to reify the statement with a confidence value.

As we said above, the property *umbel:isAligned* tells us that there exist some non-empty intersection between the class *foo:Bar* and *sc:Person*. This mean that some individuals of the class *foo:Bar* are also individuals of the class *sc:Person*. The reification statement *_:rei12345* with the property *umbel:withAlignment* gives an approximation of the size of the intersection in percentage.

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

foo:Bar umbel:isAligned  sc:Person .

_:rei12345 rdf:type rdf:Statement .
_:rei12345 rdf:subject foo:Bar .
_:rei12345 rdf:predicate umbel:isAligned .
_:rei12345 rdf:object sc:Person .
_:rei12345 umbel:withAlignment "0.53" .
```

Table 20. isAligned and withAlignment Example

Linkage Between Named Entities

Firstly, the linkage of named entities is done using the *owl:sameAs* property. Each named entities are individuals belonging to a subject concept class. As written in the OWL Ontology document:

"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 distinct 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 21. 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 22. 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.

Appendix A: Listing of Linked External Ontologies

This is a list of the external ontologies classes that are linked to UMBEL subject concepts. This list will evolve over time and the RDF/N3 serialization document will be published at this URL: http://umbel.org/ontology/umbel_external_ontologies_linkage.n3. Note that the current linkages are partial.

FOAF (Friend-of-a-Friend Ontology)

- sc:PropositionalConceptualWork owl:equivalentClass foaf:Document
- sc:Person owl:equivalentClass foaf:Person
  - Note: this assertion holds by following the paths: (Person – subClassOf → SocialBeing – subClassOf → IntelligentAgent – subClassOf → Agent-Generic) **and** (foaf:Person – subClassOf → foaf:Agent)
- sc:Agent-Generic owl:equivalentClass foaf:Agent
  - Note: this assertion holds by following the paths: (MultiIndividualAgent – subClassOf → Agent-Generic) **and** (foaf:Group – subClassOf → foaf:Agent)
- sc:MultiIndividualAgent-Intelligent owl:equivalentClass foaf:Group
  - Note: this assertion holds by following the paths: (Organization – subClassOf → MultiIndividualAgent-Intelligent – subClassOf → MultiIndividualAgent – subClassOf → Agent-Generic) **and** (foaf:Organization – subClassOf → foaf:Agent)
- sc:Organization owl:equivalentClass foaf:Organization
- sc:Picture-Image owl:equivalentClass foaf:Image
- sc:InternetService owl:equivalentClass foaf:OnlineAccount
  - Note: because of this assertion, foaf:OnlineChatAccount, foaf:OnlineEcommerceAccount and foaf:OnlineGamingAccount becomes subClassOf sc:InternetService. This is what is explicated bellow
- sc:InternetService umbe:superClassOf foaf:OnlineChatAccount
- sc:InternetService umbe:superClassOf foaf:OnlineEcommerceAccount
- sc:InternetService umbe:superClassOf foaf:OnlineGamingAccount
- sc:Project umbe:superClassOf foaf:Project

FOAF Yandex (Friend-of-a-Friend Yandex extension Ontology)

- sc:PropositionalConceptualWork umbel:superClassOf foaf-yandex:Posts

RSS 1.0 (Really Simple Syndicate Ontology)

- sc:List-Information owl:equivalentClass rss:channel
- sc:Picture-Image owl:equivalentClass rss:image
- sc:PropositionalConceptualWork umbel:superClassOf rss:item

SIOC (Socially Interlinked Online Communities Ontology)

- sc:PropositionalConceptualWork umbel:superClassOf sioc:Item
- sc:InternetService umbel:superClassOf sioc:User
- sc:PropositionalConceptualWork umbel:superClassOf sioc:Post
- sc:MessageThread owl:equivalentClass sioc:Thread

- sc:ComputerBulletinBoard owl:equivalentClass sioc:Forum
- sc:Role umbel:superClassOf sioc:Role
- sc:WebSite umbel:superClassOf sioc:Site
    - Note: this assertion holds since: (sc:WebSite – subClassOf → sc:InformationStore)
- sc:InformationStore umbel:superClassOf sioc:Space
- sc:UserAccountGroup owl:equivalentClass sioc:Usergroup

DOAP (Description of a Project Ontology)

- sc:Project umbel:superClassOf doap:Project
    - Note: this assertion holds since: (doap:Project – subClassOf → foaf:Project) **and** (foaf:Project – equivalentClass → sc:Project)

GEO WGS84 (Geo wgs84 Ontology)

- sc:SpatialThing owl:equivalentClass geo:SpatialThing
- sc:GeographicalPlace owl:equivalentClass geo:Point

GEONAMES (Geonames Ontology)

- sc:Country owl:equivalentClass geonames:Country
- sc:SpatialThing umbel:superClassOf geonames:Feature

CC (Creative Commons Ontology)

- sc:CopyrightLicense owl:equivalentClass cc:License

EVENT (Event Ontology)

- sc:Event owl:equivalentClass event:Event

PO (Programme Ontology)

- sc:Broadcasting owl:equivalentClass po:Broadcast
    - Note: this assertion holds because: (po:Broadcast – subClassOf → event:Event) **and** (umbel:Broadcasting – subClassOf → umbel:Event)
- sc:Broadcasting umbel:superClassOf po:FirstBroadcast
- sc:Broadcasting umbel:superClassOf po:RepeatBroadcast
- sc:BroadcastMediaProduct owl:equivalentClass po:Programme
- sc:BroadcastSeriesProduct owl:equivalentClass po:Series
    - Note: this assertion holds because: (BroadcastSeriesProduct – subClassOf → BroadcastMediaProduct) **and** (po:Series – subClassOf → po:Programme)
- sc:BroadcastMediaProduct umbel:superClassOf po:Episode
- sc:CommunicationsOrganization umbel:superClassOf po:Service
- sc:RadioStation-Organization owl:equivalentClass po:Radio
- sc:RadioStation-Organization umbel:superClassOf po:LocalRadio
- sc:RadioStation-Organization umbel:superClassOf po:RegionalRadio
- sc:RadioStation-Organization umbel:superClassOf po:NationalRadio
- sc:TelevisionStation umbel:superClassOf po:TV

MO (Music Ontology)

- sc:MusicPerformanceAgent owl:equivalentClass mo:MusicArtist
- sc:MusicalPerformer owl:equivalentClass mo:SoloMusicArtist
- sc:Band-MusicGroup owl:equivalentClass mo:MusicGroup
- sc:RecordedSoundProduct owl:equivalentClass mo:Record
- sc:MusicTrack owl:equivalentClass mo:Track
- sc:Multi-MovementComposition owl:equivalentClass mo:Movement
- sc:AudioConceptualWork owl:equivalentClass mo:MusicalWork
- sc:Sound owl:equivalentClass mo:Sound
- sc:MusicalPerformance owl:equivalentClass mo:Performance
- sc:InformationRecordingProcess owl:equivalentClass mo:Recording

FRBR

- sc:ConceptualWork owl:equivalentClass frbr:Work

ENDNOTES

[1] http://opencyc.org

[2] http://www.w3.org/TR/2008/WD-skos-reference-20080125/

[3] http://www.w3.org/TR/owl-ref/

[4] http://www.w3.org/TR/2008/WD-skos-reference-20080125/#L881

[5] http://motools.sourceforge.net/event/

[6] More on this role of UMBEL is presented in M. Bergman, *The Role of UMBEL: Stuck in the Middle with You . . .*, May 11, 2008, http://www.mkbergman.com/?p=441. Some additional narrative that follows is also taken from this post.

[7] M. Bergman and F. Giasson, eds., *Distilling Subject Concepts from OpenCyc – Volume 1,* Technical Report **TR 08-07-16-B1**, July 2008, 37 pp. See SubjectConcepts_20080716vB1.pdf. 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.

[8] M. Bergman and F. Giasson, eds., *Distilling Subject Concepts from OpenCyc – Volume 3,* Technical Report **TR 08-07-16-B3**, July 2008, 20 pp. See SubjectConcepts_20080716vB3.pdf.

[9] http://cyc.com

[10] 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.

[11] See http://umbel.org/images/080605_umbel_techdoc_fig3.png.

[12] http://www.lingvoj.org/

[13] http://motools.sourceforge.net/event/

[14] http://umbel.zitgist.com

[15] http://umbel.zitgist.com/view_detailed.php?concept=Boxer

[16] http://www.w3.org/TR/2008/WD-skos-reference-20080125/

[17] http://umbel.zitgist.com/view_detailed.php?concept=Boxer

[18] http://www.bbc.co.uk/ontologies/programmes/2008-02-28.shtml#Radio

[19] http://umbel.zitgist.com/view_detailed.php?concept=RadioStation-Organization

[20] http://dublincore.org/

[21] http://www.w3.org/DesignIssues/Notation3.html

[22] 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. See UMBELOntology_20080716vA2.pdf.