Mapping of EuroWordnet Top Ontology into Upper Cyc Ontology*

Atanas K. Kiryakov[1] and Kiril Iv. Simov[2]

[1] OntoText Lab., Sirma AI Ltd.
Hr. Botev 38A, Sofia 1000, Bulgaria, naso@sirma.bg
[2] Linguistic Modelling Laboratory, Bulgarian Academy of Sciences
Acad. G. Bontchev Str. 25A, 1113 Sofia, Bulgaria kivs@bgcict.acad.bg

Abstract. A mapping of EuroWordnet Top Ontology into Upper Cyc Ontology is presented. The mapping is expressed in terms of a CycL microtheory encoding of the EuroWordnet Top Ontology, because it is impossible to be made just by means of equivalence and subsumption relations. However we provide also a simplified relational view that is sufficient for many purposes.
The mapping will facilitate a better understanding of those ontologies. It could also be used as a tool for the linking of the actual lexical items in the WordNets for the EuroWordnet covered languages to the knowledge represented in the Cyc knowledge base.

1 Introduction

Understanding a text can require a vast amount of knowledge about the world including common-sense knowledge, factual knowledge about the context of the writing of the text and general knowledge. We can think of the process of understanding as a construction of a knowledge base representing the content of the text. Of course, such a view is an oversimplification of what real understanding is, but in our opinion this is a good starting point. The knowledge base connected to a text can be divided in two parts corresponding to the common understanding that a knowledge base comprises an ontology part (*ontology content*) and a specific knowledge base part (*factual content*). In general, each text represents both kinds of knowledge. The ontology part of the text constrains the interpretations of the content of the text within the context of some domain(s) and the factual knowledge part asserts what state of affairs is the text about. We think a lexical knowledge base (similar to WordNet) can be used as a key tool for constructing the ontology content of a text, especially if linked to a world knowledge base (similar to Cyc knowledge base).

In this paper we will be concerned with mapping of the EuroWordnet Top Ontology (see [7]) to the Upper Cyc Ontology (see [2]). A direct and in the

* This work is partially supported by Tübingen-Sofia International Graduate Programme in Computational Linguistics and Represented Knowledge (CLARK) funded by the Volkswagen-Stiftung.

same time formally correct mapping via equivalence and subsumption relations is impossible because of the complexity of the Upper Cyc Ontology. This is why the mapping is expressed in terms of a CycL (the knowledge representation language of Cyc) microtheory encoding of the EuroWordnet Top Ontology. We also provide, however, a simplified relational view of the mapping that is sufficient for many purposes. The purpose of the mapping is manifold: (1) to ensure linking between the concepts in the two ontologies; (2) to be used as a tool for the linking of the actual lexical items in several languages to the knowledge represented in the Cyc knowledge base; (3) to provide a more detailed semantic context information for the lexical items in EuroWordnet.

2 The Two Ontologies

EuroWordnet (**EWN**) (see [7]) is a collection of lexical knowledge bases in several languages along the lines of the Princeton WordNet [3]. Each WordNet included in EWN is based on the notion of synset — a set of synonyms representing a common sense. Synsets are related by *lexical relations* (see [1][1]). The main structuring relations besides *synonymy* (which is the defining relation for synsets) are: *hyponymy, meronymy, antonymy.* The structure of each WordNet represents the knowledge about the lexical concepts for each specific language. All WordNets in EWN are (loosely) connected to an Inter Lingual Index (**ILI**) that defines a relation of equivalence between synsets in the different WordNets. ILI is defined in terms of synsets from the Princeton WordNet for English (version 1.5) augmented manually with additional concepts during the development of EWN.

A set of base concepts is developed in order to unify the conceptual knowledge represented among the different WordNets. The base concepts were selected from the resources available in each language according to their importance defined by two criteria: (1) the number of relations connected to each concept and (2) its position in the hierarchy of concepts. The set of common ("language independent") base concepts in the Inter Lingual Index is produced by merging (in a complex way) the sets for each language. Additionally, the set of base concepts is grouped in coherent clusters by means of the EuroWordnet Top Ontology (**EwnTO**). It comprises 64 concepts defining the fundamental semantic classes. Because of the choice of the base concepts it is expected that all the words could also be classified under the semantic features of the EwnTO via their relations to the base concepts.

The Upper Cyc Ontology (**UCO**) (see [2]) is the publicly available part (3000 constants) of the Cyc Knowledge Base and is devoted to the representation of language independent encyclopaedic knowledge. Many of the constants are defined as unary predicates and could be viewed as concepts. Others denote relations, logical operators and so forth. The name of each Cyc constant begins

[1] in WordNet the relations between synsets are called semantic. Only the relations between the words and the synsets are referred as lexical

with the string `#$`. The set of constants in UCO is hierarchically organized by means of two structuring relations:

- `#$isa` - a relation connecting an *instance* with a *collection* or class. Formally, `(#$isa X Y)` means exactly that X is an element of the set Y.
- `#$genls` - subsumption relation connecting one collection with another, more general one. Formally, `(#$genls X Y)` means exactly that X is a subset of Y.

The two relations are very different in their features. `#$genls` is a transitive relation. For instance, from

`(#$genls #$Dog #$Mammal)` and
`(#$genls #$Mammal #$Vertebrate)` follows that
`(#$genls #$Dog #$Vertebrate)`.[2]

The `#$isa` relation is similar to `member_of` relation and thus it is not transitive. For example, from

`(#$isa #$Pufy #$Dog)` and
`(#$isa #$Dog #$BiologicalClass)` it doesn't follow that
`(#$isa #$Pufy #$BiologicalClass)`.

Clearly, the two relations are not independent from each other - an instance of a collection is also an instance of all super-collections. In our example, from

`(#$isa #$Pufy #$Dog)` and
`(#$genls #$Dog #$Mammal)` follows that
`(#$isa #$Pufy #$Mammal)`.

Each concept in UCO is defined in the following terms: (1) a set of `#$genls` (or some other generalization) statements which determine the most specific concepts which the current concept specializes; (2) a set of `#$isa` statements which determine the most specific concepts of which the current constant is an instance; (3) a comment in English which gives a humanly understandable description of the intend interpretation of the constant. It should be mentioned also that multiple-inheritance is vastly used. Another feature of UCO is that in the hierarchy there are classifications of collection of collections and in this way one can state properties of predicates, attributes and others.

The important point about UCO is that it contains (according to its developers) enough concepts to structurally classify properly each new concept.

3 Representation of EwnTO in CycL

The mapping of EwnTO top concepts to the constants of UCO is defined by the means of assertions made in CycL knowledge representation language. All the assertions encoding the mapping are made in a microtheory `#$EuroWordnetMt` which extends the Cyc `#$BaseKB` microtheory. This way all constants defined in UCO are visible within `#$EuroWordnetMt` but it is still a separated theory. In

[2] The relations are written in the prefix format of the Cyc knowledge representation language (CycL).

order to avoid the name clashes between the name of top concepts in EwnTO and the constants in UCO we added to each name of a concept in EwnTO the suffix TC. Formally each top concept in EwnTO is represented as a Cyc predicate. The following CycL expressions state these assertions[3]:

```
(#$genlMt #$EuroWordnetMt #$BaseKB)
(#$comment #$EuroWordnetMt "EuroWordnetMt microtheory ...")

(#$ist #$EuroWordnetMt (#$isa #$EuroWordnetTCType
                              #$PredicateCategory))
(#$ist #$EuroWordnetMt (#$comment #$EuroWordnetTCType
               "Collection of the EuroWordnet top-concepts ..."))
```

The highest concept in EwnTO (Top) is stated to be the same as the peak of the hierarchy of UCO:

```
(#$ist #$EuroWordnetMt (#$equals #$TopTC #$Thing))
(#$ist #$EuroWordnetMt (#$isa #$TopTC #$EuroWordnetTCType))
```

Here are some of the definitions of auxiliary predicates we use in the mapping:

```
(#$genlPreds #$exactType #$isa)
(#$isa #$exactType #$TaxonomicSlotForCollections)
(#$equivalent (#$exactType ?COL ?TYPE)
              (#$equivalent (#$isa ?X ?TYPE) (#$genls ?X ?COL)))
```

`#$exactType` is a specification of the `#$isa` predicate which relates two collection such that the second collection `?TYPE` is a collection of collections and it contains all sub-collections of `?COL` (including `?COL` itself) and only them. Using this predicate we can state as equivalent a part of the collection hierarchy and a collection of collections, i.e. concept and concept type.

```
(#$isa #$specificType #$DefaultMonotonicPredicate)
(#$isa #$specificType #$TaxonomicSlotForCollections)
(#$equivalent (#$specificType ?COL ?TYPE)
              (#$implies (#$isa ?X ?TYPE) (#$genls ?X ?COL)))
```

The predicate `#$specificType` could be used to state that a given collection contains only some collections from the sub-hierarchy. In the above assertions `#$equivalent` is the usual logical operator easily definable in CycL.

[3] The relation `#$ist` states explicitly that the assertion made by the second argument is made in the microtheory given by the first argument. In the first few definitions we will give these statements explicit but in the rest of the paper we will leave these statements implicit.

4 Mapping of the Top Concepts

EwnTO distinguishes three main categories of objects: `1stOrderEntity`, `2ndOrderEntiry` and `3rdOrderEntity`. They divide the entities in the following way: things existing in time and space; situations; and unobservable propositions. The first can be perceivable by the senses, the second occur or take place rather than exist and the third can be evaluated as false or true.

Four concepts are defined one step below `1stOrderEntity` concept - `Origin`, `Form`, `Composition`, `Function`, which determine the main aspects of each time and space thing and correspond to the elements of the Qualia structure presented in [6]. In our view these concepts are more like orthogonal dimensions of description or meta-attributes of the concepts rather than usual concepts (object classes) themselves. We encode them as classes of predicates rather than predicates. The next layer under these four meta-concepts defines the concepts representing the actual values for them. Here we give an example of the encoding of one of these concepts in `#$EuroWordnetMt`:

```
(#$equals #$1stOrderEntityTC #$SomethingExisting)
(#$isa #$1stOrderEntityTC #$EuroWordnetTCType)

(#$isa #$1stOrderEntityTCType #$PredicateCategory)
(#$genls #$1stOrderEntityTCType #$EuroWordnetTCType)
(#$exactType #$1stOrderEntityTC #$1stOrderEntityTCType)

(#$genls #$OriginTCType #$1stOrderEntityTCType)|

(#$equals #$NaturalTC #$NaturalTangibleStuff)
(#$isa #$NaturalTC #$OriginTCType)
```

This example demonstrates the most simple mapping when there exists a Cyc constant with the same meaning as the EWN Top Concept.

The hierarchy below `2ndOrderEntiry` is similar. The first level defines two dimensions for the characteristics of a situation: `SituationTypes` and `SituationComponents`. The former divides situations in dynamic and static while the later defines clustering of the situations according to the presence of a specific aspect in the description of the situation content. Thus we can follow the above pattern in the definition of these concepts also.

```
(#$genls #$SituationTypeTCType #$2ndOrderEntityTCType)
(#$genls #$SituationComponentTCType #$2ndOrderEntityTCType)
```

Below these we often have more complicated mapping. For example, we need to define `MentalTC` as follows

```
(#$equals #$MentalTC (#$UnionFn #$MentalEvent #$MentalAttribute))
```

Here an EWN top concept is represented as a disjunction of two UCO constants because UCO doesn't contain a common concept for dynamic and static situations which also to account for the presence of a mental aspect in the situation.

Even more interesting case is the mapping of the EWN's `PartTC`. In UCO there are some constants devoted to distinguish some special kinds of parts such as parts of organisms, parts of buildings and others but obviously there is no general definition of *part*. Therefore all we can point out are examples (specific cases) of parts. We can also say that in principle only in individual things can constitute a part of something:

```
(#$genls #$PartTC #$Individual)
(#$genls #$OrganismPart #$PartTC)
(#$genls #$CellPart #$PartTC)
(#$genls #$PartOfBuilding #$PartTC)
(#$isa #$PartTC #$CompositionTCType)
```

This definition reflects another characteristic of EwnTO which we will discuss in detail below. As a part of the Composition dimensions `PartTC` could be applied to many entities below `1stOrderEntity`, but its value is not significant for many of the concepts: "It is not the case that all persons will be classified as Parts because they may be part of group." (see [7]). The notion of "intensional" significance is important in the EWN classification of word meanings but it is very hard to represent them on a general level in Cyc. Thus, we decided to leave this concept underspecified.

One even harder problem is the definition of EWN top concept `TimeTC`. It is defined in [7] as: "Situations in which duration or time plays a significant role; Static yesterday, day, pass, long, period, Dynamic e.g. begin, end, last, continue." This gives us a hint at how to constrain the concept from above by the disjunction of two UCO constants (`#$Event` or `#$StaticSituation`) both of which have a temporal aspect. However this is still too general to cover the EwnTO meaning of `TimeTC` which doesn't include a lot of events and states which are classified under `#$Event` or `#$StaticSituation`. The only appropriate concept in UCO is `#$TemporalRelation` but it is more specific and the mapping is problematic because it is not a specialization of `#$SituationType`.

5 Handling the Differences in Representation and Structure

Some typical phenomena considered during the mapping are discussed here.

5.1 Mismatching Taxonomic Structure

There are concepts which glosses match but their formal definitions in the UCO and the EwnTO (with respect to the taxonomic relations) differ. In those cases the mapping was made according to the glosses.

Lets take as an example `MoneyRepresentationTC` that is subsumed in EWN by `RepresentationTC`. These two top concepts are wired into UCO as follows:

```
(#$equals #$RepresentationTC #$InformationBearingObject)
(#$equals #$MoneyRepresentationTC #$TenderObject)
```

However it is NOT true in UCO that

```
(#$genls #$TenderObject #$InformationBearingObject)
```

The reason for this is the different structuring of the conceptualizations used by the creators of both ontologies. Thus `#$Currency` (a specialization of `#$TenderObject`) is an `#$InformationBearingObject` but fails to cover some of the meanings of `MoneyRepresentationsTC`, for example "shares". The closest UCO concept for the latter one is `#$Stock`, but it is a specialization of `#$SalesAgreement` and it is not declared as a kind of `#$TenderObject`. The reason for this is that `#$Stock` covers only the abstract aspect of the *stock* without its material (paper) media which is represented by `#$StockCertificate`. Unfortunately, the later one is not declared to be a specialization of `#$TenderObject`. The mismatch can be partially corrected if `#$TenderObject` could be classified in UCO as a kind of `#$InformationBearingObject`. The last assertion would be correct because each of `#$TenderObject` instances could play this role. However the correctness and relevance of the last assertion is arguable. Finally, the comparison would be easier if `#$StockCertificate` be classified as a `#$TenderObject`, but the last is not obvious for some kinds of `#$Stock`.

This was a typical example in which we mapped `MoneyRepresentation` to `#$TenderObject` following the matching glosses. Our motivation for this decision was that it could be expected that the knowledge enterers using the ontologies (especially those with linguistic background) are more likely to also give preference to the meaning stated in the gloss. It is also a fact that the formal meaning encoded by the taxonomic relations is just a small fraction of what is described in the gloss.

5.2 Concept Types instead of Concepts

Some UCO constants stand for concept types (collections of collections) rather than for concepts themselves (i.e. collections). Such concept types include, for example, the constant `#$PositionType`. It represents the collection of all concepts (predicates) about occupations (`OccupationTC`) but it is not a concept itself. The mapping is even harder when such a concept type covers just part of the sub-concepts of a top concept. Such is the case for example between `#$SocialAttaributeType` and `SocialTC`. The former covers only the static situations clustered under `SocialTC`.

In this respect, the mapping between `Time` and `#$TemporalRelation` is interesting because the later constant does not represents a concept, but a concept type. That means that the concepts related with it (like `#$after`) are not its

specializations (sub-concepts). They are its instances that in Cyc will be expressed via `(#$isa #$after #$TemporalRelation)` rather then using one of the subsumption predicates (`#$genls`, `#$genlPreds`, etc.).

5.3 Missing Subsumption Relations in UCO

There are subsumption relations that are not precise in UCO. For example `Occupation` could be mapped as a specialization of `#$IntendedFunction`, especially considering the base concepts that are clustered under this top concept. On the other hand `#$PositionType` (that has no relation with `#$IntendedFunction`) is still relevant to `Occupation` because all the instances of `#$PositionType` are specializations of `Occupation`. In this case we included `#$PositionType` as a constant that is not "the Mapping" but still relevant to `Occupation`. This case is getting additionally complicated by the fact that `#$PositionType` itself is a type of concept rather then concept. The definition is:

```
(#$genls #$OccupationTC #$IntendedFunction)
(#$specificType #$OccupationTC #$PositionType)
```

Summarizing all this, we conclude this section with a table (Table 1) showing the relations between an EWN top concepts and UCO constants which are necessary to be used in the mapping.

Table 1. Mapping Relations

Relation name	**Encoding in CycL**	**Comment**
exact mapping	`(#$equals EWNTC CYCC)`	
more general in Cyc	`(#$genls EWNTC CYCC)`	
more specific in Cyc	`(#$genls CYCC EWNTC)`	
instance of	`(#$exactType EWNTC CYCC)`	otherwise equivalent but encoded as a concept type (rather than concept) in UCO
instance of, more general in Cyc	`(#$isa EWNTC CYCC)`	EWNTC is more specific then each of the instances of CYCC
instance of, more specific in Cyc	`(#$specificType EWNTC CYCC)`	EWNTC is more general then some of the instances of CYCC
qualia for	`(#$genls EWNTC CYCC)` [4]	EWNTC is qualia (attribute type) for instances of CYCC

6 Conclusion

The EWN top concepts are classified according to "quality" of their mapping into UCO as follows: *exact mapping*(33 top concepts),*difference*(14), *problematic*(3), *gap*(8), and *qualia*(6). Above we mainly discussed the top concepts classified

as *differences*. *Problematic* are such mappings that are correct according to the glosses and the intuition behind the corresponding concepts but imply contradiction to the formal relations in one or both of the ontologies. *Gaps* are top concepts that do not have even "intuitive" mapping, i.e. such that are not covered in UCO. The judgment between *gap* and *difference* is hard in some of the cases. Finally *qualia* are the top concepts immediately below `XthOrderEntity` level that we already discussed. The classification for each top concept could be seen in the appendix.

The mapping defined here shows the level of compatibility between the EwnTO and UCO. It became clear that there are important aspects that could not be properly covered in UCO. We should say that this mapping could suffer in quality because of two main reasons: (1) the complexity of UCO, that the authors can not pretend to fully understand; (2) underspecification of some of the EWN top concepts, especially some of those below 1stOrderEntity. Thus there is a place for improvement of the mapping.

We plan to use the mapping in two future works. First, it will be extended with the base concepts of EuroWordnet and then used for the creation of a Bulgarian lexical knowledge base connected to EuroWordnet base concepts and thus to UCO. Second, we envisage a use of the mapping for the analysis of text on the idea of lexical chains (see [4]) which will be used to determine the right ontology chunks assigned to the text along the lines of [5].

References

1. Cruse, D.A., *Lexical Semantics.* Cambridge University Press. Cambridge, England, 1986.
2. Cyc Public Ontology. `http://www.cyc.com/cyc-2-1/`
3. Fellbaum, Christiane (editor), *WORDNET: an electronic lexical database.* MIT Press, 1998.
4. Graeme Hirst and David St-onge, *Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms.* In [3]. MIT Press, 1998.
5. Atanas Kiryakov and Kiril Iv. Simov. *Ontologically Supported Semantic Matching.* In the proceedings of NoDaLiDa'99 conference. Trondheim, Norway, 1999.
6. James Pustejovsky. *Generative Lexicon.* The MIT Press. Cambridge, MA, 1995.
7. Vossen Piek (ed.), *EuroWordnet General Document.* Version 3, Final, July 19, 1999, `http://www.hum.uva.nl/ ewn`

7 APPENDIX: Relational mapping between EWN Top Concepts and Upper Cyc Ontology constants

The type of mapping could be found in brackets next to each EWN top concept (put in **bold face**) in the table below. The related UCO constants are listed below the concept together with the type of relation for each one and comments. The type of relation is omitted when it is *exact mapping* or *qualia* — in both cases the related UCO constant is put on the same line.

1stOrderEntity (exact mapping) — `#$SomethingExisting`

2ndOrderEntity (exact mapping) — `#$Situation`

3rdOrderEntity (exact mapping) — `#$PropositionalInformationThing`

Agentive (gap)
`#$Event`— much more general in Cyc

Animal (exact mapping) — `#$NonPersonAnimal`

Artifact (exact mapping) — `#$Artifact`

BoundedEvent (difference)
`#$Event` — more general in Cyc
`#$TemporalObjectType` — instance of, much more general in Cyc

Building (exact mapping) — `#$Building`

Cause (gap)
`#$Event` — much more general

Comestible (exact mapping) — `#$FoodAndDrink`

Communication (difference)
`#$Situation` — much more general in Cyc
`#$Communicating` — more specific in Cyc, it requires exchange of information between at least two agents
`#$ibtHasInfoAbout` — more specific in Cyc, covers "to be about" sense of "communicate", but only for non-abstract `#$InformationBearingThings`
`#$propositionalInfoAbout` — more specific in Cyc, covers "to be about" sense of "communicate" for abstract things, e.g. theories

Composition (qualia) — `#$SomethingExisting`

Condition (difference)
`#$Situation` — much more general in Cyc
`#$WeatherAttribute` — more specific in Cyc
`#$PhysiologicalCondition` — more specific in Cyc
`#$TangibleStuffStateType` — instance of, more specific in Cyc
`#$StateOfMatter-SolidLiquidGaseous` — more specific in Cyc

Container (exact mapping) — `#$ContainerProduct`[5]

Covering (gap)
`#$SomethingExisting` — much more general in Cyc

Creature (gap)
`#$BiologicalLivingObject` — much more gen. in Cyc, missing, contradictory in EWN

Dynamic (exact mapping) — `#$Event`

Existence (diffence)
`#$Event` — much more general in Cyc
`#$CreationOrDestructionEvent` — more specific in Cyc, covers only changes in the state of existence[6]

Experience (difference)
`#$Situation` — much more general in Cyc

[5] Seems a bit more specific in Cyc but it is not formally specified in EWN in order compare precisely

[6] Thanks to Wim Peters who stressed this aspect. It was initially judged as an exact mapping

#$Perceiving — more specific in Cyc, covers only the phisical experiences, but not mental such as "desire"
#$FeelingAttribute — more specific in Cyc, covers only the mental experiences, but not physical, such as "hear"

Form (qualia) — #$SomethingExisting

Function (qualia)
#$SomethingExisting — qualia for
#$IntendedFunction — more specific in Cyc
#$Role — more specific in Cyc

Furniture (exact mapping) — #$FurniturePiece

Garment (exact mapping)
#$ClothingItem

Gas (exact mapping)
#$GaseousTangibleThing

Group (exact mapping)
#$Group

Human (exact mapping)
#$Person

ImageRepresentation (exact mapping) — #$VisualInformationSource

Instrument (difference)
#$SomethingExisting — much more general in Cyc
#$PhysicalDevice — more specific in Cyc. But it is too underspecified in EWN in order to compare properly

LanguageRepresentation (exact mapping) — #$TextualMaterial

Liquid (exact mapping)
#$LiquidTangibleThing

Living (exact mapping)
#$BiologicalLivingObject

Location (difference)
#$Situation — more general in Cyc
#$SpatialPredicate — instance of, more specific in Cyc, covers Location + Static
#$MovementEvent — more specific in Cyc, covers Location + Dynamic

Manner (difference)
#$Situation — more general in Cyc
#$ScriptPerformanceAttribute — more specific in Cyc, covers the Static situations with Manner aspect
#$LocomotionEvent — more specific in Cyc

Mental (difference)
#$Situation — more general in Cyc
#$MentalAttribute — more specific in Cyc, covers Mental + Static
#$MentalEvent — more specific in Cyc, covers Mental + Dynamic

Modal (exact mapping) — #$ModalRelationship

MoneyRepresentation (exact mapping) — #$TenderObject

Natural (exact mapping) — #$NaturalTangibleStuff

Object (exact mapping)
#$ExistingObjectType — instance of

Occupation (difference)
`#$IntendedFunction` — more general in Cyc
`#$PositionType` — instance of, more specific in Cyc, related only to people

Origin (qualia) — `#$SomethingExisting`

Part (difference)
`#$Individual` — much more general in Cyc
`#$PartOfBuilding` — more specific in Cyc
`#$CellPart` — more specific in Cyc
`#$OrganismPart` — more specific in Cyc

Phenomenal (gap)
`#$Event` — much more general in Cyc

Physical (difference)
`#$Situation` — much more general in Cyc
`#$PhysicalAttribute` — more specific in Cyc, covers the Static + Physical situations
`#$PhysicalEvent` — more specific in Cyc, covers the Dynamic + Physical situations

Place (exact mapping)
`#$Place`

Plant (exact mapping) — `#$PlantBLO`

Possession (difference)
`#$Situation` — more general in Cyc
`#$ChangeInUserRights` — more specific in Cyc, covers Dynamic + Possession
`#$userRightsRelation` — more specific in Cyc, partially covers Static + Possession
`#$hasOwnershipIn` — more specific in Cyc, should be related to `#$userRightsRelation`, but it is not in UCO
`#$UserRightsAttribute` — more specific in Cyc, partially covers Static + Possession

Property (problematic)
`#$AttributeValue` — it is not a `#$Situation` in Cyc
`#$StaticSituation` — more general in Cyc

Purpose (exact mapping) — `#$PurposefulAction`

Quantity (gap)
`#$Situation` — more general in Cyc

Relation (problematic)
`#$Relationship` — it is not a `#$Situation` in Cyc
`#$StaticSituation` — more general in Cyc

Representation (exact mapping) — `#$InformationBearingObject`

SituationComponent (qualia) — `#$Situation`

SituationType (qualia) — `#$Situation`

Social (difference)
`#$Situation` — more general in Cyc
`#$SocialOccurrence` — more specific in Cyc, covers Social + Dynamic
`#$SocialAttributeType` — instance of, more specific in Cyc, covers Social + Static

Software (exact mapping) — `#$ComputerProgram`[7]

Solid (exact mapping) — `#$SolidTangibleThing`

[7] in Cyc it is an IBO, i.e. tangible object that bears an information that could be interpreted as a computer program. In EWN it is not determined that it is tangible

Static (exact mapping)
`#$StaticSituation`
`#$Relationship` — more specific in Cyc, not related to `#$StaticSituation` in Cyc
`#$AttributeValue` — more specific in Cyc, not related to `#$StaticSituation` in Cyc

Stimulating (gap)
`#$Event` — much more general in Cyc

Substance (exact mapping) — `#$ExistingStuffType`

Time (problematic)
`#$Event` — much more general in Cyc
`#$TemporalRelation` — instance of, more specific in Cyc, partially covers Time + Static, it is not a situation type in Cyc
`#$StaticSituation` — much more general in Cyc

Top (exact mapping) — `#$Thing`

UnboundedEvent (difference)
`#$Event` — more general in Cyc
`#$TemporalStuffType` — instance of, much more general in Cyc

Usage (gap)
`#$Situation` — more general in Cyc
`#$ConsumingFoodOrDrink` — more specific in Cyc

Vehicle (exact mapping) — `#$TransportationDevice-Vehicle`