Introducing ODIN: Ontological Design grounded in Image-schematic kNowledge

Stefano De Giorgis[1], Aldo Gangemi[1,2]

[1]University of Bologna - FICLIT Department, Via Zamboni 32, 40126 Bologna (BO), Italy
[2]ISTC - CNR, Via S. Martino della Battaglia 44, 00185 Roma (RM), Italy

Abstract
Ontology Design Patterns (ODPs) are defined as “possible solutions against recurrent problems”, and pattern-based design has been recognised as a consolidated good modelling practice for improving ontology engineering quality. Furthermore the adoption of ODPs fosters the practices of ontology modularization and ontology reuse. The increasing number of Content ODPs, as well as their nature of cognitive schematisations, suggests the possibility of some further abstraction, which could provide also an organisation criterion. We propose here ODIN: a knowledge graph for Ontological Design grounded in Image-schematic kNowledge. Image schemas come from the embodied cognition and Cognitive Metaphors Theory tradition, and are recurrent cognitive patterns stemming from human sensori-motor experience. They are considered the basic building blocks for each and every human conceptualisation. They operate as embodied cognitive building blocks and structures to provide meaning to everything, according to human perception-conception loop. In this work we perform an analysis of ODPs’ OWLs comments and descriptions to automatically extract Image Schemas in order to provide, for each ODP, its sensori-motor grounding, bringing out the underlying cognitive structure and allowing possible ODPs cognitive upper layer organization.

Keywords
ontology design, image schemas, knowledge representation, cognitive semantics, embodied cognition,

1. Introduction

Ontology Design Patterns (ODPs) are defined as “possible solutions against recurrent problems” [1, 2], and pattern-based design has been recognised as a consolidated good modelling practice [3] for improving ontology engineering quality [4]. Furthermore, the adoption of ODPs fosters the practices of ontology modularization and ontology reuse.

Image schemas (IS), on the other side, have been proposed within the tradition of embodied cognition as conceptual structures that capture sensorimotor experiences (namely the individual perception of the world via senses and locomotor experience) and shape abstract cognition, including commonsense reasoning and semantic structures of natural language (see e.g. [5, 6]). “Schema” refers to our capability to “conceptualize situations at varying levels of schematicity” [7] and “image” resonates the imagistic capability in the sense of “mental representation”, schematic gestalts that integrate information from various modalities [6], rather than purely

WOP2022: 13th Workshop on Ontology Design and Patterns, October 23-24, 2022, Hangzhou, China
stefano.degiorgis2@unibo.it (S. De Giorgis); aldo.gangemi@unibo.it (A. Gangemi)
https://github.com/StenDoipanni/ (S. De Giorgis); https://www.unibo.it/sitoweb/aldo.gangemi (A. Gangemi)
0000-0003-4133-3445 (S. De Giorgis); 0000-0001-5568-2684 (A. Gangemi)

visual images [8]. Image schemas are furthermore defined as “internally structured gestalts” (unified wholes of meaning) [9], that is, composed of spatial primitives (SP) that make up more complex image schemas [10, 5, 11].

For example, considering the expression “some time has passed”: “time” is conceptualized as a moving entity on a spatial dimension, it refers to the well-known “TIME IS A MOVING OBJECT” cognitive metaphor, and it activates (namely, it is structured referring to elements from some sense-making conceptual structure like) the SOURCE_PATH_GOAL image schema, which is activated in conceptualizing any kind of movement along a path. This IS, in order to realize its semantics, is (gestaltically) composed by the SOURCE, PATH, and GOAL spatial primitives.

While their existence in natural language has been studied by means of corpus-based (e.g. [12, 13]) and machine learning methods (e.g. [14, 15, 16]), few approaches to formalize image schemas (e.g. Image Schema Logic [17]) and connect them to existing resources to capture semantics exist.

The notion of image schema profile is used here as in [18] and [19] to describe the collection of IS which are activated by some entity, sentence, situation, or event.

Therefore, the idea of injecting image-schematic knowledge in the existing ODPs, providing some cognitive grounding to ontological design structures, stems from two main assumptions, one on the pragmatical side and the other on the theoretical level. The first one (i) is that the increasing number of content ODPs proposed (more than one hundred at the actual state of the art) suggests that a further clustering process will be, in the close future, if not needed at least strongly advised; the second one (ii) is that ODPs are schematisations operated by humans while designing architectural solutions to some set of recurrent instantiations of some situation. Therefore the process of cognitive abstraction is intrinsic to the ODPs nature.

This work proposes a first rough attempt to found ODPs on a cognitively valid grounding that is at the same time ’abstract’ enough to support semantic interoperability of ontologies and data, and ’embodied’ in sensorimotor experience. ODIN ontology and image schematic extraction results analysis focus on Content ODPs descriptions, starting from their ontological modules[1].

2. Preliminaries

We introduce here some theoretical background, and present some state of the art tools involved in the analysis that we expose in detail in Sec. 4.

In particular to perform automatic image schema profile extraction we rely on [20], reusing the FRED tool and the ImageSchemaNet ontology, part of the Framester resource.

2.1. ImageSchemaNet and IS Automatic Detection

Frames in a most general notion are (cognitive) representations of typical features of a situation. Fillmore’s frame semantics [21] has been most influential as a combination of linguistic

[1]OWL files bootstrapped from http://ontologydesignpatterns.org/wiki/Submissions:ContentOPs when available.

descriptions and characterisation of related knowledge structures to describe cognitive phenomena. Words or phrases are associated with frames based on the common scene they evoke. In FrameNet, frames are also explained as *situation types*. In Framester semantics [22] observed/recalled/anticipated/imagined situations are consequently occurrences of frames.

Fillmore explicitly compares frames to other notions, such as experiential gestalt [23], stating that frames can refer to a unified framework of knowledge or a coherent schematization of experience. Thus, widely acknowledged frames provide a theoretically well-founded and practically validated basis for commonsense knowledge patterns.

Framester [24, 22] is a linked data hub that provides a formal semantics for frames [22], based on Fillmore's frame semantics [21]. It creates/reengineers linked data versions of linguistic resources, such as WordNet [25], OntoWordNet [26], VerbNet [27], BabelNet [28], etc, jointly with factual knowledge bases (e.g. DBpedia [29], YAGO [30]). Framester also includes ImageSchemaNet.

ImageSchemaNet [31] is a formal and re-usable representation of image schemas as Semantic Web technology in form of an ontological layer. It presents a formal representation of image schemas as a new layer of the Framester hub. Since a major flaw in current image schema theory was the lack of agreement about the lexical coverage of image schemas, ImageSchemaNet introduces an image-schematic layer linking IS and SP to FrameNet frames and frame elements, WordNet synsets (sets of contextual synonyms) and word supersenses, VerbNet verbs, etc., thereby creating a formal, lexicalized integration of cognitive semantics, enactive theories, and frame semantics. Currently, ImageSchemaNet provides lexical coverage and formalization for the following six image schemas:

- SOURCE_PATH_GOAL: cognitive representation of any kind of movement
- CONTAINMENT: activated for any kind of containment situation
- CENTER_PERIPHERY: used to conceptualize the notion of centrality and the relation with more peripheral elements or landmark areas
- PART_WHOLE: any mereological relation
- SUPPORT: cognitive representation of a force relation for which an object is supporting another one
- BLOCKAGE: the resistance encountered during some attempt of exerting any force e.g. pushing, pulling, moving, lifting etc.

Framester can be used to jointly query (via a SPARQL endpoint[2]) all the resources aligned to its formal frame ontology[3]. Finally Framester has been used [32] to formalize the MetaNet resource of conceptual metaphors[4], based on FrameNet frames as metaphor sources and targets (frame-based), as well as to uncover semantic puzzles emerging from a logical treatment of frame-based metaphors.

[2]http://etna.istc.cnr.it/framester2/sparql
[3]The Framester Schema is available at: https://w3id.org/framester/schema/
[4]The MetaNet schema in Framester's OWL is at https://w3id.org/framester/metanet/schema/.

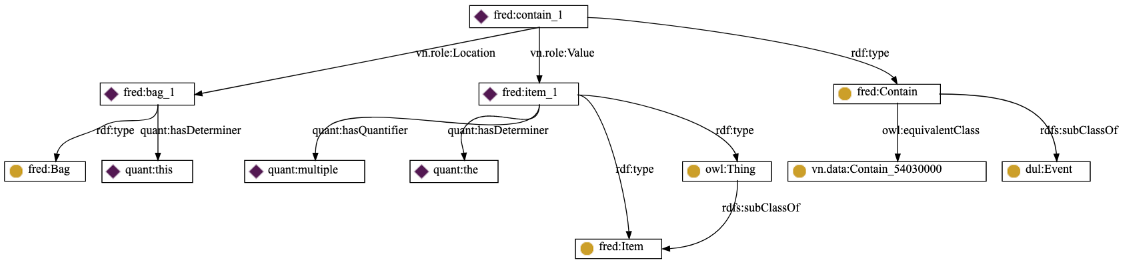

Figure 1: Knowledge graph produced by FRED for *What are the items contained in this bag?*

2.2. FRED Tool

FRED [33] is a hybrid knowledge extraction system to generate knowledge graphs directly from natural language taking as input text from natural language. It is built with a pipeline that includes both statistical and rule-based components, aimed at producing RDF and OWL knowledge graphs, with embedded entity linking, word-sense disambiguation on WordNet synsets, and frame/semantic role detection, aligning entities from the produced graph directly to entities from FrameNet, VerbNet, DBpedia, PropBank etc., in turn aligned one with the other in the Framester resource. FRED is preferred here to other tools due to its importance in the automatic IS profile extraction (see Sec. 2.3) and its native integration with the Framester resource.

2.3. Automatic Image Schema Profile Extraction

The workflow to extract IS and SP from natural language is as follows: (i) the first step is to take a sentence and pass it to FRED tool to generate a knowledge graph from text. Let's take as example "*What are the items contained in this bag?*", which shows the presence of an activator of CONTAINMENT. Figure 2.2 shows the knowledge graph automatically generated as output by FRED[5].

The second step (ii) consists in taking all the subjects, predicates, and objects of all triples, namely all nodes and arches in the graph, to systematically query the Framester repository, in particular the ImageSchemaNet graph, via SPARQL queries, to check if there is an activation of some Image Schema or Spatial Primitive[6]. The ImageSchemaNet graph contains more than 40k triples which take as subject entities from resources in the Framester hub (FrameNet, VerbNet, WordNet etc.) and declare the activation of some IS or SP. ImageSchemaNet knowledge graph generation is shown in [31].

Finally, for each occurrence of activation retrieved, a triple is added to the original graph declaring the activation. In the above-mentioned example, one IS activator retrieved in the ImageSchemaNet graph is the VerbNet verb `vn.data:Contain_54030000`, resulting in three

[5]This graph can be reproduced using the *What are the items contained in this bag?* sentence as input for the FRED online demo, available here: http://wit.istc.cnr.it/stlab-tools/fred/demo/, a far more extende graph can be obtained by ticking the "Align concepts to Framester" option.

[6]Some useful explorative queries are available at https://github.com/StenDoipanni/ISAAC/tree/main/ImageSchemaNet

occurrences of CONTAINMENT activation. Further examples of IS profile extraction considering both IS and SP are mentioned in [20].

3. ODIN Knowledge Graph

We present here a knowledge graph obtained by performing the above mentioned automatic image schemas extraction on 116 existing ODPs, which allows to explore the image schema profile of each and any ODP analyzed, to correlate the presence of some IS to the `odin:ODPType`, namely its being a “Foundational”, “Core” or “Domain” ODP, and determine its image-schematic nature. The process poses some problems, and some caveats should be kept in mind, as explained in the next Section.

3.1. ODIN Structure

To perform automatic image schema extraction it is necessary a knowledge base to start from. Such a knowledge base is obtained via collecting all the available OWL files from the ODPs official repository[7], resulting in 116 ODPs OWL files to be tested. The full list is provided as additional material in Sec. A, Fig. A in Appendix.

Since image schema extraction, as exposed in Sec. 2.3, takes as input a string of text, we decided to look for entities' annotation already existing in OWL files, and we opted for the most comprehensive form of entity annotation, namely `rdfs:comment`. We are well aware that some noise have to be considered, due to (i) the different modelling / annotating style, including only the comments that are strictly necessary vs extensive comments providing a variety of references, examples, and scenarios; (ii) different amount of verbosity in the modules, and (iii) any kind of `rdfs:comment` on an entity which is not directly related to that entity, but it's e.g. referring to the meta-level explaining some modelling choice or ontological issue. A manual first evaluation of the `rdfs:comment` collected from ODPs' OWLs resulted anyway in a reasonable corpus for the desired task. That said, the ODIN building process starts with SPARQL querying the newly built knowledge base of 116 ODPs, resulting in 621 entries annotated via `rdfs:comment` including general ODPs' URIs annotation - namely description of the whole pattern or ontological module - classes and object properties.

To allow to predicate something like the image schematic activation of the textual description of an entity, being it a class or a property, the punning technique is adopted to materialize the intensional individuals for all the entities annotated with `rdfs:comment`, therefore resulting in 621 individuals maintaining their original URIs. The necessary consequent step - since the image schematic activation is triggered by the textual annotation, and not by the entity itself - is to create a corresponding individual with the same name but ODIN uri and ´´_Descr´´ at the end of the name `odin:entity_Descr`), linked with the intensional individual of the original entity via the property `odin:describedIn`. Finally, automatic IS extraction is performed on each `rdfs:comment` text, and the new knowledge produced is stored in the ODIN graph using classes and properties as described in the next two Sections.

[7] http://ontologydesignpatterns.org

3.2. ODIN Classes

In ODIN ontology there are four main classes:

- `odin:ODP`: This class is used to group all the individuals that are originally URIs of a whole ODP module, we expect their description to be more general and eventually include names of the ODP designers, version, etc.
- `ODPDescription`: This class represents the the intensional meaning of the description of any ODP. It is used as type for the reification of individual's descriptions, eventually activating some IS.
- `odin:ODPElement`: This class is used to group all entities which are a specific class or property of some ODP, therefore offering only a partial view of what the whole pattern is meant for, and we can expect to retrieve here the activation of some spatial primitives, namely image schematic roles.
- `odin:ODPType`: This class is used to classify ODPs between `odin:Foundational` and `odin:Core`. Foundational ODPs are those which are related to domain-independent patterns, namely theories axiomatizing relations of parthood, connection, time, causality, etc.; while Core ODPs are related to the axiomatization of generic theories about some specific domain. The usage of a third independent class like `odin:ODPType`, instead of two subclasses of the `odin:ODP` class is due to the fact that "type" is not intended here as ontological type but it refers to the type of ontologies to which the patterns aims at being a modeling solution for.

3.3. ODIN Object Properties

ODIN introduces some object properties and reuses some others to declare the relation between some `odin:ODP`, `odin:ODPElement` and all the other possible inferences, namely its description, some IS activated and the entities actually activating the IS from the graph generated by FRED.

- `odin:describedIn`: this property takes as domain the individual created via punning, namely the intensional entity of the ODP entity from its original module, and as range the individual created ad hoc to represent the content of the `rdfs:comment` annotation.
- `odin:hasODPComponent`: this property relates some ODP individual and some ODPElement, it is used to maintain, with a form of light semantics, the original relation between the ODP ontology URI and the classes and entities actually constituting the ODP.
- `odin:odpType`: this property is used to declare the domain of application of some ODP, namely Core or Foundational, therefore it takes as domain some `odin:ODP` and as range some `odin:ODPType`.
- `isnet:hasActivator`: This property is reused from the ImageSchemaNet ontology and it is used to declare which entities are activating some Image Schema from the graph generated by FRED taking as input some sentence. It takes as domain some `odin:ODPDescription` and as range the IS activator from existing and well known semantic web resources, namely synsets from WordNet, verbs VerbNet, frames from FrameNet, DBpedia entities but also Framester semantic types, roles and frame elements.

Table 1
ODIN ODPs, comments and IS - SP Activation.

ODPs	Comments	TOT Act	1 Act	2 Act	3 Act	4+Act
116	621	404	180	140	51	32

Table 2
ODIN Image Schemas occurrences via automatic extraction.

SPG	CONTAINMENT	PART_WHOLE	BLOCKAGE	SUPPORT	CENTER_PERIPHERY
126	179	233	23	24	23

- `isnet:activates`: This property is reused from ImageSchemaNet ontology and takes as domain some `ODPDescription` and as range the image schema or spatial primitive activated (e.g. `isaac:CONTAINMENT`, `isaac:PART_WHOLE`, `isaac:PATH` etc.).

4. Discussion and Results

Table 1 shows some data about the IS detection: the total number of ODPs in exam is shown in column “ODPs” (the full list of 116 ODPs is available in Sec. A of Appendix); the total number of entities retrieved being annotated with `rdfs:comment` is shown in column “comments”, these are the 621 sentences given as input to FRED tool[8]; out of these 621, column “TOT Act” shows the total amount of sentences for which at least 1 image schema is retrieved; more in detail, the number of sentences for which there is the activation of one IS is in column “1 Act”, the activation of two in column “2 Act”, three in “3 Act” and four or more in “4+ Act”.

Table 2 shows the distribution of image schemas activation. It is clear from the dominance of PART_WHOLE that the partonomical and mereological aspects of patterns are of great importance while some more specific ones (according to IS literature and consistently with the lower lexical coverage) like SUPPORT and BLOCKAGE are an order of magnitude smaller.

Table 3 shows the distribution of spatial primitives activation. Due to lack of space, those spatial primitives for which the number of occurrences was zero are excluded from this table, but still listed here: SOURCE, INSIDE, PART, SUPPORTED, BLOCKER, BLOCKED. Interestingly, in countertrend with the activation of image schemas, the dominant spatial primitive is CONTAINER, which suggests that the containing element is of particular importance in the structure of the patterns.

Please be reminded that multiple image schemas might be correct for a single sentence, and no inconsistency or incompatibility could come from co-location of different image schemas in the same sentence.

[8]All the FRED graphs generated are available here:
https://github.com/StenDoipanni/ODIN/blob/main/odp_comments_final_fred_graphs.ttl

Table 3
ODIN Spatial Primitives occurrences via automatic extraction.

PATH	GOAL	CONTAINER	WHOLE	SUPPORTER	CENTER	PERIPHERY
8	55	63	7	2	2	7

4.1. ODPs Image Schema Profile

As mentioned in Sec. 2 an image schema profile is the set of IS related to some entity, being it an Event, Scenario, Process etc. In our case a meaningful IS profile can be considered the one related to a whole ODP, considering all the IS retrieved by descriptions of its entities and properties. While the full data are available both as table in Sec. A of Appendix and graph online[9] we propose here two meaningful examples starting from Content ODPs for which it is at least plausible to assume which IS they should be grounded in.

4.2. "Bag" ODP

The first case we take in exam is the bag.owl ODP [34]. The definition taken directly from the file is: "To model bags of items (elements). The Bag is characterized by a collection that can have multiple copies of each object.". From its classes, properties and definitions (e.g. `bag:Item`, `bag:size`, `bag:Bag` it seems clear that the sensorimotor grounding of this module could be ascribed to the mereological aspects of types of clusters, therefore to PART_WHOLE and to the Johnson's cognitive metaphor "CATEGORIES ARE CONTAINERS", therefore to CONTAINMENT. Performing the SPARQL query to extract the IS profile[10], provided in Appendix B our expectations are met with four entities activating CONTAINMENT and three activating PART_WHOLE, for a total of 14 triggers from the FRED graphs.

4.3. "Transition" ODP

Transition ODP [2] is dedicated to the formal representation of the notion of "transition" as being composed by three time intervals, a transitional event in time, an initial state and a resulting state and some object undergoing to the transition process. From these elements we could make the hypothesis that the sensorimotor grounding would be in SOURCE_PATH_GOAL, as representing a motion event whose particular focus is on initial location in time (SOURCE) and final location in time (GOAL); PART_WHOLE, for the compound structure of the process, and eventually some form of CONTAINMENT, due to the metaphorical way of expressing the idea of "being in a state" as "being *in* a location". Performing the query to extract IS profile[11] we retrieve 38 triggering entities from FRED graphs, for a total of 8 PART_WHOLE; 9 CONTAINMENT and 3 CONTAINER; 2 SOURCE_PATH_GOAL and 8 GOAL, confirming the idea of having a strong focus on the ending point (resulting state) of the process; and 8

[9]ODIN graph is available at https://github.com/StenDoipanni/ODIN

[10]Results are provided as example on the ODIN GitHub: https://github.com/StenDoipanni/ODIN/blob/main/bag_is_profile.csv

[11]Query results are available as example on ODIN GitHub: https://github.com/StenDoipanni/ODIN/blob/main/transition_IS_profile.csv

SUPPORT, which comes a bit unexpected and is activated mainly from frames, meaning that more than specific disambiguated entities it is activated by the occurrence of more general frame.

5. Conclusion and Future Works

We presented in this work ODIN, a module for Ontological Design grounded in Image schematic kNowledge. The two ODPs' image schema profiles exposed above in Sec. 4.2 and 4.3 show how this initial rough work could drive to a more refined sensorimotor grounding of foundational, as well as core and domain, ontology design patterns, introducing an upper layer of "cognition schemata" describing how the conceptualization of modelling solutions stems from our embodied cognition and can be traced back to our sensorimotor experience. As future developments ODIN ontology could be improved by refining the lexical material used to perform the image schema profile extraction: one hypothesis is to consider and weigh differently the `cpannotationschema:` properties according to the semantics expressed by different entities and properties in the ODP module. Furthermore, the image schematic grounding could be used to cluster ODPs referring to the same sensori-motor experience, helping in avoiding duplicates, providing a common upper layer to ODPs referring e.g. to the same or similar knowledge domain.

A. Appendix: ODIN ODPs

In Fig. 2 you find the full list of ODPs used to build ODIN ontology and on which it is performed the image-schematic analysis.

B. Appendix: Explore the Resource

Here you find some useful queries to fully exploit the resource.

IS profile: Query to retrieve the image schema profile of some ODP module (namely to retrieve all IS / SP activated for each and every of the elements of some specific ODP. In the following example the URI used is the one of the bag.owl ODP.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX odin: <http://www.ontologydesignpatterns.org/ont/odin/odin.owl#>
PREFIX isaac: <http://www.ontologydesignpatterns.org/ont/is/isaac_vanilla.owl#>
PREFIX isnet: <http://www.ontologydesignpatterns.org/ont/is/isnet.owl#>

SELECT DISTINCT ?s ?is ?comment
WHERE { ?s rdfs:comment ?comment ; odin:describedIn ?descr .
?descr isnet:activates ?is .
FILTER (regex(str(?s), 'http://www.ontologydesignpatterns.org/cp/owl/bag.owl'))}
```

1	ODP URI	60	https://w3id.org/smartenvironment/patterns/network.owl
2	http://www.ontologydesignpatterns.org/cp/owl/recurrenteventseries.owl	61	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_FeatureOfInterest.owl
3	http://www.ontologydesignpatterns.org/cp/owl/fsdas/aquaticresourceobservation.owl	62	http://ontologydesignpatterns.org/template
4	http://www.ontologydesignpatterns.org/cp/owl/fsdas/vesselspecies.owl	63	http://www.ontologydesignpatterns.org/ont/mario/affordance.owl
5	http://www.ontologydesignpatterns.org/cp/owl/descriptionandsituation.owl	64	http://mklab.iti.gr/pericles/DigitalVideo_ODP
6	http://www.descartes-core.org/patterns/owl/transportPattern	65	http://www.ontologydesignpatterns.org/cp/owl/description.owl
7	http://w3id.org/daselab/onto/trajectory	66	http://www.ontologydesignpatterns.org/cp/owl/participation.owl
8	http://www.ontologydesignpatterns.org/cp/owl/timeinterval.owl	67	https://w3id.org/smartenvironment/patterns/event.owl
9	http://www.ontologydesignpatterns.org/cp/owl/agentrole.owl	68	http://www.irit.fr/recherches/MELODI/ontologies/AAE
10	http://www.ontologydesignpatterns.org/cp/owl/timeindexedpersonrole.owl	69	http://www.ontologydesignpatterns.org/cp/owl/fsdas/climaticzone.owl
11	http://ontologydesignpatterns.org/ekp/Airline.owl	70	http://www.ontologydesignpatterns.org/cp/owl/invoice.owl
12	http://www.ontologydesignpatterns.org/cp/owl/sequence.owl	71	http://www.ontologydesignpatterns.org/cp/owl/fsdas/catchrecord.owl
13	http://www.ontologydesignpatterns.org/cp/owl/move.owl	72	http://www.ontologydesignpatterns.org/cp/owl/topic.owl
14	http://www.ontologydesignpatterns.org/cp/owl/tagging.owl	73	http://www.ontologydesignpatterns.org/cp/owl/basicplan.owl
15	http://www.ontologydesignpatterns.org/cp/owl/fsdas/aquaticresources.owl	74	http://w3id.org/daselab/onto/event
16	http://www.ontologydesignpatterns.org/cp/owl/constituency.owl	75	http://www.ontologydesignpatterns.org/cp/owl/observation.owl
17	http://www.ontologydesignpatterns.org/cp/owl/timeperiod.owl	76	http://www.ontologydesignpatterns.org/cp/owl/fsdas/vesselwaterarea.owl
18	http://delicias.dia.fi.upm.es/ontologies/PeriodicInterval.owl	77	http://mklab.iti.gr/pericles/BornDigitalArchives_ODP
19	http://semantic.cs.put.poznan.pl/ontologies/oshdo/HazardousSituation	78	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Situation.owl
20	http://www.ontologydesignpatterns.org/cp/owl/fsdas/gearvessel.owl	79	http://descartes-core.org/ontologies/lca/1.0/LCAPattern.owl
21	http://mklab.iti.gr/pericles/ComputerSystem_ODP	80	https://w3id.org/smartenvironment/patterns/object.owl
22	http://www.ontologydesignpatterns.org/cp/owl/fsdas/resourceexploitationobservation.owl	81	http://ecareathome-ontology.mpi.aass.oru.se/patterns/time.owl
23	http://ecareathome-ontology.mpi.aass.oru.se/patterns/time.owl	82	http://www.ontologydesignpatterns.org/cp/owl/timeindexedparticipation.owl
24	http://www.ontologydesignpatterns.org/cp/owl/timeindexedclassification.owl	83	http://www.ontologydesignpatterns.org/cp/owl/fsdas/resourceabundanceobservation.owl
25	http://purl.org/net/roles	84	http://www.ontologydesignpatterns.org/cp/owl/fsdas/speciesnames.owl
26	http://www.ontologydesignpatterns.org/cp/owl/fsdas/speciesconservation.owl	85	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Property.owl
27	http://purl.org/net/social-reality	86	http://www.ontologydesignpatterns.org/cp/owl/situation.owl
28	http://semantic.cs.put.poznan.pl/ontologies/reportingevent.owl	87	http://www.ontologydesignpatterns.org/cp/owl/typesofentities.owl
29	http://www.ontologydesignpatterns.org/cp/owl/naryparticipation.owl	88	http://www.ontologydesignpatterns.org/cp/owl/actingfor.owl
30	http://www.ontologydesignpatterns.org/cp/owl/countingas.owl	89	http://www.ontologydesignpatterns.org/cp/owl/coparticipation.owl
31	http://www.ontologydesignpatterns.org/cp/owl/informationobjectsandrepresentationlanguages.owl	90	http://www.ontologydesignpatterns.org/cp/reaction.owl
32	http://www.ontologydesignpatterns.org/cp/owl/fsdas/gearwaterarea.owl	91	http://www.ontologydesignpatterns.org/cp/owl/region.owl
33	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Object.owl	92	http://www.ontologydesignpatterns.org/cp/owl/bag.owl
34	http://www.ontologydesignpatterns.org/cp/owl/fsdas/gearspecies.owl	93	http://www.ontologydesignpatterns.org/cp/owl/fsdas/speciesconditions.owl
35	http://www.ontologydesignpatterns.org/cp/owl/set.owl	94	http://www.ontologydesignpatterns.org/cp/owl/fsdas/verticaldistribution.owl
36	http://www.ontologydesignpatterns.org/cp/owl/taskrole.owl	95	http://www.ontologydesignpatterns.org/cp/owl/planconditions.owl
37	http://www.ontologydesignpatterns.org/cp/owl/list.owl	96	http://www.ontologydesignpatterns.org/cp/owl/fsdas/specieseat.owl
38	http://www.ontologydesignpatterns.org/cp/owl/persons.owl	97	http://descartes-core.org/ontologies/activities/1.0/ActivityPattern
39	http://www.ontologydesignpatterns.org/cp/owl/objectrole.owl	98	https://w3id.org/smartenvironment/patterns/sensing
40	http://www.semanticweb.org/hirtq/ontologies/2018/4/EntityWithProvenanceOntologyPattern	99	http://www.ontologydesignpatterns.org/cp/owl/collectionentity.owl
41	http://www.ontologydesignpatterns.org/cp/owl/taskexecution.owl	100	http://purl.org/NET/rdfchess/ontology/
42	http://delicias.dia.fi.upm.es/ontologies/ObjectWithStates.owl	101	http://www.ontologydesignpatterns.org/cp/owl/informationrealization.owl
43	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Network.owl	102	http://www.ontologydesignpatterns.org/cp/owl/controlflow.owl
44	http://www.ontologydesignpatterns.org/cp/owl/place.owl	103	http://www.ontologydesignpatterns.org/cp/owl/topic.owl
45	http://www.ontologydesignpatterns.org/cp/owl/parameter.owl	104	http://www.ontologydesignpatterns.org/cp/owl/intensionextension.owl
46	http://www.ontologydesignpatterns.org/cp/owl/price.owl	105	http://www.ontologydesignpatterns.org/cp/owl/transition.owl
47	http://www.ontologydesignpatterns.org/cp/owl/communities.owl	106	http://www.ontologydesignpatterns.org/cp/owl/eventprocessing.owl
48	http://ontologydesignpatterns.org/ekp/owl/EthnicGroup.owl	107	http://semantic.cs.put.poznan.pl/ontologies/newsreportingevent.owl
49	http://w3id.org/daselab/onto/spatiotemporalextent	108	https://w3id.org/smartenvironment/patterns/place.owl
50	http://www.ontologydesignpatterns.org/cp/owl/linnaeantaxonomy.owl	109	http://mklab.iti.gr/pericles/Policy_ODP
51	http://www.ontologydesignpatterns.org/cp/owl/timeindexedpartof.owl	110	http://www.ontologydesignpatterns.org/cp/owl/gotop.owl
52	https://w3id.org/smartenvironment/patterns/geometry.owl	111	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Geometry.owl
53	http://www.ontologydesignpatterns.org/cp/owl/classification.owl	112	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Sensing.owl
54	http://www.ontologydesignpatterns.org/cp/owl/fsdas/speciesbathymetry.owl	113	http://www.ontologydesignpatterns.org/cp/owl/fsdas/rtmsmapping.owl
55	http://www.ontologydesignpatterns.org/cp/owl/timeindexedsituation.owl	114	https://w3id.org/smartenvironment/patterns/situation.owl
56	http://www.ontologydesignpatterns.org/cp/owl/componency.owl	115	https://w3id.org/food/ontology/dir
57	http://ecareathome-ontology.mpi.aass.oru.se/patterns/SmartHome_Place.owl	116	http://www.ontologydesignpatterns.org/cp/owl/partof.owl
58	http://www.ontologydesignpatterns.org/cp/owl/fsdas/specieshabitat.owl	117	http://delicias.dia.fi.upm.es/ontologies/SimpleOrAggregated.owl
59	http://www.ontologydesignpatterns.org/cp/owl/basicplanexecution.owl	118	
60	https://w3id.org/smartenvironment/patterns/network.owl	119	

Figure 2: ODPs bootstrapped from ontologydesignpatterns.org

IS profile including activators: Same query as before but including also the activators entities (e.g. WordNet synsets, VrbNet verbs, FrameNet frames etc.)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX odin: <http://www.ontologydesignpatterns.org/ont/odin/odin.owl#>
PREFIX isaac: <http://www.ontologydesignpatterns.org/ont/is/isaac_vanilla.owl#>
PREFIX isnet: <http://www.ontologydesignpatterns.org/ont/is/isnet.owl#>

SELECT DISTINCT ?s ?comm ?is ?tr
WHERE { ?s odin:describedIn ?descr ; odin:odpType ?type ; rdfs:comment ?comm .
?descr isnet:activates ?is ; isnet:hasActivator ?tr .
```

```
FILTER (regex(str(?s), 'http://www.ontologydesignpatterns.org/cp/owl/bag.owl'))}
```

IS Activation and ODPType: Query to retrieve all `odin:ODPs` and `odin:ODPElements` and of a specific `odin:ODPType` - in the following example "Foundational" - and their IS Activation.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX odin: <http://www.ontologydesignpatterns.org/ont/odin/odin.owl#>
PREFIX isaac: <http://www.ontologydesignpatterns.org/ont/is/isaac_vanilla.owl#>
PREFIX isnet: <http://www.ontologydesignpatterns.org/ont/is/isnet.owl#>

SELECT DISTINCT ?s ?comm ?is ?tr
WHERE { ?s a odin:ODP; odin:odpType odin:Foundational ;
  rdfs:comment ?comm ; odin:describedIn ?descr .
  ?descr isnet:activates ?is . }
```

References

[1] A. Gangemi, V. Presutti, Ontology design patterns, in: Handbook on ontologies, Springer, 2009, pp. 221–243.

[2] P. Hitzler, A. Gangemi, K. Janowicz, Ontology engineering with ontology design patterns: foundations and applications, volume 25, IOS Press, 2016.

[3] E. Blomqvist, A. Gangemi, V. Presutti, Experiments on pattern-based ontology design, in: Proceedings of the fifth international conference on Knowledge capture, 2009, pp. 41–48.

[4] E. Blomqvist, K. Hammar, V. Presutti, Engineering ontologies with patterns-the extreme design methodology., Ontology Engineering with Ontology Design Patterns (2016) 23–50.

[5] J. M. Mandler, C. Pagán Cánovas, On defining image schemas, Language and Cognition (2014) 1–23. doi:10.1017/langcog.2014.14.

[6] L. Talmy, The fundamental system of spatial schemas in language, in: B. Hampe, J. E. Grady (Eds.), From perception to meaning: Image schemas in cognitive linguistics, volume 29 of *Cognitive Linguistics Research*, Walter de Gruyter, 2005, pp. 199–234.

[7] R. W. Langacker, Foundations of cognitive grammar: Theoretical prerequisites, volume 1, Stanford university press, 1987.

[8] G. Lakoff, Women, Fire, and Dangerous Things. What Categories Reveal about the Mind, The University of Chicago Press, 1987.

[9] M. Johnson, The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason, The University of Chicago Press, Chicago and London, 1987.

[10] B. Hampe, Image schemas in cognitive linguistics: Introduction, From perception to meaning: Image schemas in cognitive linguistics 29 (2005) 1–14.

[11] M. M. Hedblom, O. Kutz, F. Neuhaus, Choosing the right path: image schema theory as a foundation for concept invention, Journal of Artificial General Intelligence 6 (2015) 21–54.

[12] A. Papafragou, C. Massey, L. Gleitman, When English proposes what Greek presupposes: The cross-linguistic encoding of motion events, Cognition 98 (2006) B75–B87.
[13] J. A. Prieto Velasco, M. Tercedor Sánchez, The embodied nature of medical concepts: image schemas and language for pain., Cognitive processing (2014). doi:10.1007/s10339-013-0594-9.
[14] D. Gromann, M. M. Hedblom, Body-mind-language: Multilingual knowledge extraction based on embodied cognition, in: AIC, 2017, pp. 20–33.
[15] D. Gromann, M. M. Hedblom, Kinesthetic mind reader: A method to identify image schemas in natural language, in: Proceedings of Advancements in Cogntivie Systems, 2017.
[16] L. Wachowiak, D. Gromann, Systematic analysis of image schemas in naturallanguage through explainable multilingualneural language processing, in: Proceedings of the 56th Linguistic Colloquium, Peter Lang Group, Forthcoming.
[17] M. M. Hedblom, O. Kutz, T. Mossakowski, F. Neuhaus, Between contact and support: Introducing a logic for image schemas and directed movement, in: Conference of the Italian Association for Artificial Intelligence, Springer, 2017, pp. 256–268.
[18] M. M. Hedblom, O. Kutz, R. Peñaloza, G. Guizzardi, Image schema combinations and complex events, KI-Künstliche Intelligenz 33 (2019) 279–291.
[19] T. Oakley, Image schemas, The Oxford handbook of cognitive linguistics (2007) 214–235.
[20] S. De Giorgis, A. Gangemi, D. Gromann, The racing mind and the path of love: automatic extraction of image schematic triggers in knowledge graphs generated from natural language (2022).
[21] C. J. Fillmore, Frame semantics, in: Linguistics in the Morning Calm, Seoul: Hanshin, 1982, pp. 111–138.
[22] A. Gangemi, Closing the loop between knowledge patterns in cognition and the semantic web, Semantic Web 11 (2020) 139–151.
[23] G. Lakoff, M. Johnson, Metaphors we live by, University of Chicago press, 1980.
[24] A. Gangemi, M. Alam, L. Asprino, V. Presutti, D. R. Recupero, Framester: a wide coverage linguistic linked data hub, in: European Knowledge Acquisition Workshop, Springer, 2016, pp. 239–254.
[25] G. A. Miller, Wordnet: A lexical database for english, Communications of the ACM 38 (1995) 39–41.
[26] A. Gangemi, R. Navigli, P. Velardi, The ontowordnet project: extension and axiomatization of conceptual relations in wordnet, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, 2003, pp. 820–838.
[27] K. K. Schuler, VerbNet: A broad-coverage, comprehensive verb lexicon, University of Pennsylvania, 2005.
[28] R. Navigli, S. P. Ponzetto, Babelnet: Building a very large multilingual semantic network, in: Proceedings of the 48th annual meeting of the association for computational linguistics, 2010, pp. 216–225.
[29] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: The semantic web, Springer, 2007, pp. 722–735.
[30] F. M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: Proceedings of the 16th international conference on World Wide Web, 2007, pp. 697–706.

[31] S. De Giorgis, A. Gangemi, D. Gromann, Imageschemanet: Formalizing embodied commonsense knowledge providing an imageschematic layer to framester, Semantic Web Journal, forthcoming (2022).
[32] A. Gangemi, M. Alam, V. Presutti, Amnestic forgery: An ontology of conceptual metaphors, in: S. Borgo, P. Hitzler, O. Kutz (Eds.), Proceedings of the 10th International Conference, FOIS 2018, volume 306 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2018, pp. 159–172. URL: https://doi.org/10.3233/978-1-61499-910-2-159. doi:10.3233/978-1-61499-910-2-159.
[33] A. Gangemi, V. Presutti, D. R. Recupero, A. G. Nuzzolese, F. Draicchio, M. Mongiovì, Semantic web machine reading with FRED, Semantic Web 8 (2017) 873–893. URL: https://doi.org/10.3233/SW-160240. doi:10.3233/SW-160240, 10.3233/SW-160240.
[34] P. Ciccarese, S. Peroni, The collections ontology: creating and handling collections in owl 2 dl frameworks, Semantic Web 5 (2014) 515–529.