## Details of the ODPs used are as under:-

***1. Acting For ODP:*** This ODP represents that some agent is acting in order to forward the action of an agent.
***2. Chess ODP:*** The ODP represents a flexible schema for linked data querying of chess games. Players are modeled as AgentRole. In addition, notions such as chess moves, chess tournaments, as well as chess game annotation are included.
***3. Climatic Zone ODP:*** The intent of this ODP is to be able to represent climatic zones for aquatic resources.This pattern allows to query what climatic zones are typical of an aquatic resource.
***4. Description ODP:*** To formally represent a conceptualization or a descriptive context.This ODP allows the designer to represent both a descriptive context and the components that characterize in that context.
***5. Hazardous Situation ODP:*** This ODP provides a building block for modelling situations where one or more object is exposed to one or more hazards to some extent.They also result in some consequences.
***6. Region ODP:*** It is a basic ODP which talks about attributes/parameters/dimensions, while still referring to their values.
***7. DUL ODP:*** The Affordance ODP relies on the descriptions and situations and is combined with a frame-based representation scheme. This allows to extend the notion of affordance not only to physical objects, but also complex situations afford actions.
***8. Activity Pattern ODP:*** It incorporates the general two perspectives of activities: a workflow perspective, which are often observed in planning-related applications, and a spatiotemporal perspective, which are often found in geographic activity analysis.
***9. Born Digital Archives ODP:*** The ODP models the domain of born digital archives.
***10. Classification ODP:*** To represent the relations between concepts (roles, task, parameters) and entities (person, events, values), which concepts can be assigned to.
***11. Componency ODP:*** It represent that objects either are proper parts of other objects, or have proper parts.It allows designers to represent part-whole relations and distinguish between parts and proper parts.
***12. Computer System ODP:*** It models computer systems based on a hardware/software approach.It is expected to facilitate the creation of computer

system domain ontologies that can be exploited in numerous fields.
***13. Constituency ODP:*** Represents the constituents of a layered structure. It models the physical objects and its parts.
***14. Digital Video ODP:*** It models digital video files, their components and other associated entities, such as codecs and containers.This ODP is expected to facilitate the creation of digital video domain ontologies.
***15. Event ODP:*** It provides a minimalistic model of event where it is not always possible to separate its spatial and the temporal aspects, thus can model events that move or possess discontinuous temporal extent. Events according to this model has at least one participant, attached via a participant-role, and may have additional descriptive information through its information object.
***16. GoTop ODP:*** It allows to categorize gene-related types.It represents parts of gene-related entities (transitively).
***17. Gear Species ODP:*** It outlines the relations between aquatic species and types of fishing gear that are suitable for catching. While we do not make this distinction from the species' viewpoint, we do distinguish what species are gear types targeted to and what can be also incidentally caught.
***18. Information Realisation ODP:*** Represents information objects and their physical realization.It distinguishes information objects from their concrete realizations.
***19. Intension Extension ODP:*** Represents the meaning of an information object: the concepts it expresses, the things it is about.It employs a simple set of properties to link information objects to their meanings, and to entities they can be about.
***20. Invoice ODP:*** Heterogeneous models for invoices can be aligned to this ODP, which then acts as a semantic facade to different invoice management applications. The Context class can be used to gather temporal,spatial and organizational data.
***21. LCA Pattern ODP:*** Life Cycle Assesment ODP studies the environmental impact of products taking into account their entire life-span and production chain.It specifies key aspects of LCA/LCI data models, namely the notions of flows, activities, agents, and products, as well as their properties.
***22. List ODP:*** Represents ordered lists, through a specialization of the bag pattern, where each resource in the bag is referred through an item, so that the same item can occur in several places. The usual properties of lists are also there, i.e. the sequence of elements, and references to the first and last item.

***23. n-ary Participation ODP:*** Represents events with their participants, time, space, etc.All sorts of relations denoting events with multiple participants, space-time indexing, etc. can be represented with this pattern.
***24. News Reporting Event ODP:*** It can be used for modelling situations in which we are not certain that a particular actual event has the properties which were described in a news message. It defines the properties of an actual event which were reported (time, place, actors, subevents, cause, effect etc.), but not to treat them as universal, verified knowledge.
***25. Object Role ODP:*** Represents objects and the roles they play.
***26. Place ODP:*** Models places of things and represents, transitively, where something is located. It remains unspecified what kind of location relation to represent: reference location, partial location, physical location, social or metaphoric location, etc.
***27. Policy ODP:*** It models policies, their characteristics and their associated entities, such as processes and agents.
***28. Reaction ODP:*** Models dynamic situations, tracking agents and actions they produce, events that are results of some action(s), and consequences as new actions.
***29. Task Role ODP:*** Represents the assignment of tasks to roles.It allows to put roles in the domain of discourse.
***30. Set ODP:*** It is a collection that cannot contain duplicate elements. A Set is expressed by linking to it directly all the members (elements), multiple identical values of members (elements) will be eliminated because by default they are treated as a set.
***31. Simple Aggregated ODP:*** It represents objects that can be simple or aggregated (that is, several objects gathered in another object acting as a whole).The aggregated members should belong to the same concept. For example, a turbine is part of an engine.
***32. Species Bathemetry ODP:*** Represents species together with their typical environment in terms of bathymetric range and water area.
***33. Tagging ODP:*** Rrepresents a tagging situation, in which someone uses a term, from a list, to tag something (or the content of something).It also represents the time and the polarity of the tagging.
***34. Task Execution ODP:*** Represents actions through which tasks are executed.It allows designers to make assertions on roles played by agents without involving the agents that play that roles, and vice versa.

***35. Time Indexed Participation ODP:*** To represent participants in events at some time.
***36. Topic ODP:*** To represent topics and their relations.Topics are modelled as conceptual complexes with part of (containment), overlap, and vicinity relations, and can be related to any kind of entity.
***37. Transition ODP:*** Represents basic knowledge about transitions (events, states, processes, objects).
***38. Vertical Distribution ODP:*** The intent of the ODP is to be able to represent vertical distribution for aquatic resources.This pattern allows to query what vertical distribution is typical of an aquatic resource. Whereas such values can be subject to observation, another pattern based on the generic 'observation' pattern should be used.
***39. Vessel Species ODP:*** Provides a direct relation between aquatic species and vessels that are able to catch them, regardless of the fishing gear used.
***40. Trajectory ODP:*** It provides a model of trajectory, which is understood as a sequence of spatiotemporal points.This pattern is suitable for a variety of trajectory datasets and locations and is easily extendible by by aligning to or matching with existing trajectory ontologies, foundational ontologies, or other domain specific vocabularies.
***41. Toco ODP:*** It is an ODP for telecommunication networks. It has properties like- hasDevice, hasLink,hasService etc.
***42. Interaction ODP:*** To model different interactions where the interactors can have different roles.
***43. Manchester Sequence ODP:*** Models the sequence of events, one after the other. This ODP has been taken from manchester.odp site.
***44. Manchester List ODP:*** The List is used to model ordered elements, representing the semantics of the order. This ODP models the list of genes.
***45. AdaptedSEP ODP:*** Models selective transitive propagation in the biomedical domain.
***46. Exception ODP:*** Models exceptions in biomedical field without breaking the strict class-subclass hierarchy: for example the class MammalianRedBloodCell (with restriction of hasNuclues as 0)would be a subclass of EukaryoticCell (with the restriction HasNucleus exactly 1).
***47. Data Type Relationship ODP:*** Represents a datatype value with more than one aspect. Numerical values can have different aspects. For example, a boiling point has a temperature value, a pressure, etc.

***48. Entity Feature Value ODP:*** Models features with the simplest structure possible.This ODP is used to represent modifiers with multiple aspects, thus features (e.g. colour with a certain brightness and saturation).
***49. Selector ODP:*** To recreate selectors in the biomedical field, that is refining entities that can be used to choose between to alternatives: for example, right or left hand. A selector is a modifier that can be used to select between identical entities, e.g. right and left hand.
***50. Normalisation ODP:*** To untangle a polyhierarchy, coding the subsumption relationships using restrictions rather than class-subclass relationships. The application example for this ODP is adapted from the Cell Type Ontology. In the example, the subsumption relationships that already are in the Cell Type Ontology are inferred by the reasoner instead of hard-coded.
***51. Upper Level ODP:*** Creates an ontology that can integrate different ontologies in itself.
***52. Closure ODP:*** Simulates the closed world assumption in a concrete class.One of the examples of such problem is the fact that plenty of users think that asserting an existential restriction is enough to close a relationship, when in fact a universal restriction is also needed: it is not enough to say that carnivore eats some meat, as that is equivalent to saying that it can eat another things apart of meat.
***53. Entity Quality ODP:*** To model qualities without relying in a proliferation of object properties. This ODP is modeled for biomedical field.
***54. Value Partition ODP:*** Models values of attributes. In this example we model biological regulation, being negative or positive.
***55. Entity Property Quality ODP:*** To model qualities of independent entities (e.g. position, colour).
***56. Description ODP:*** To simulate an If-Then of the type: if something fulfills certain conditions, it should have a further given attribute.
***57. Agent Role ODP:*** This ODP has been obtained from MODL.It represents the roles of an agent. It models its tasks, events, location etc.
***58. Bag ODP:*** The pattern for an Aggregation, Bag, or Collection is relatively simple. The Bag is a type of unordered collection. This pattern demonstrates a more approachable interface for the partonymy pattern, with respect to membership.
***59. Explicit Typing ODP:*** This ODP is used when there is a finite, but mutable number of types of a thing. We find this easier to maintain than a series of

subclass relationships.
***60. Identifier ODP:*** It is used for associating some sort of identifier and metadata with a thing. It can associate additional information aside from its type with a thing, e.g. an identifier may be a URL or a primary key value in a database.
***61. Name Stub ODP:*** The NameStub ODP is a specialization of the Stub Pattern.
***62. Participant Role ODP:*** It is a specialization of the AgentRole Pattern, many axioms are inherited due to this.This pattern has additional synergies with the Event.
***63. Provenance ODP:*** It suffices to align a sub-pattern to the core. The EntityWithProvenance class is any item of interest to which a developer would like to attachprovenance information.It is interested in capturing, who or what created that item, what was used to derive it, and what method was used to do so.
***64. Quantities ODP:*** It allows a developer to express a quantity of some stuff. The nature of quantities is rather complex, due to the fact that there are a multitude of dimensions, unit types, and ways to measure quantities. The Quantity class is used to express the nature of the quantity via its QuantityKind. A QuantityValue expresses the magnitude of the Quantity via an xsd:double and a Unit.
***65. Reification ODP:*** It is essentially a metapattern. It represents a set of axioms that will allow a developer to quickly reify a concept by specializing the framework.

***66. Spatial Extent ODP:*** It is characterized by a set of Interiors, which are in turn characterizedby a PointInSpace-Sequence.A PointInSpace-Sequence consists of PointInSpace-SequenceElements, which are constituted by PointInSpace.A PointInSpace is described by a value and a reference system.
***67. Spatio-Temporal Extent ODP:*** It wraps the Trajectory Pattern. Essentially, it uses the Trajectory patterns ability to capture discrete snapshots of something moving along some dimension, but casts it into the familiar physical dimensions, plus time. This is done by adding the atPlace and atTime properties.It is used when it is difficult to separate space and time when talking about a concept.
***68. Stub ODP:*** This ODP defines the meta-data of the attributes.

***69. Temporal Extent ODP:*** It is composed of a number of ComplexTimeIntervals, which may be intervals of non-zero length (i.e. TimeIntervals) or intervals of length 0 (i.e. PointsInSpace).
***70. Tree ODP:*** It allows a developer to organize data into a tree data structure.An example- the organization of organisms into a phylogenetic tree.