An Ontology Pattern for Oceanographic Cruises: Towards an Oceanographer's Dream of Integrated Knowledge Discovery

Adila Krisnadhi[1,7], Robert Arko[2], Suzanne Carbotte[2], Cynthia Chandler[3], Michelle Cheatham[1], Timothy Finin[4], Pascal Hitzler[1], Krzysztof Janowicz[5], Thomas Narock[6], Lisa Raymond[3], Adam Shepherd[3], and Peter Wiebe[3]

[1] Wright State University
[2] Lamont-Doherty Earth Observatory, Columbia University
[3] Woods Hole Oceanographic Institution
[4] University of Maryland, Baltimore County
[5] University of California, Santa Barbara
[6] Marymount University
[7] Faculty of Computer Science, Universitas Indonesia

Abstract. EarthCube is a major effort of the National Science Foundation to establish a next-generation knowledge architecture for the broader geosciences. Data storage, retrieval, access, and reuse are central parts of this new effort. Currently, EarthCube is organized around several building blocks and research coordination networks. OceanLink is a semantics-enabled building block that aims at improving data retrieval and reuse via ontologies, Semantic Web technologies, and Linked Data for the ocean sciences. Cruises, in the sense of research expeditions, are central events for ocean scientists. Consequently, information about these cruises and the involved vessels has to be shared and made retrievable. For example, the ability to find cruises in the vicinity of physiographic features of interest, e.g., a hydrothermal vent field or a fracture zone, is of primary interest for oceanographers. In this paper, we use a design pattern-centric strategy to engineer ontologies for OceanLink. We provide a formal axiomatization of the introduced patterns and ontologies using the Web Ontology Language, explain design choices, discuss the re-usability of our models, and provide lessons learned for the future geo-ontologies.

1 Introduction

Years of research in the ocean science, and the geosciences in general, have yielded an amount of data that is not only huge in volume but also highly heterogeneous both in types and formats, and scattered across distributed data repositories [1]. For individual researchers, this situation presents a difficult challenge regarding discovery, access, and integration of data which they need to conduct scientific inquiries. Not only that, this also introduces difficult knowledge management issues that must be overcome by the whole research community [2].

Sponsored by the National Science Foundation (NSF), the EarthCube initiative[1] brings together the US geoscience research community through a number of funded building blocks, research coordination networks and special interest groups to establish a knowledge infrastructure crucial for enabling cross-discipline scientific endeavors. Intuitively, such an infrastructure can facilitate data discovery and integration through centralized facilities. On the other hand, it is often the case that data quality can be better ensured when local data sources and partners are made an active part of the framework. The challenge is then *how to realize such a centralized discovery framework while maintaining a decentralized nature.*

The OceanLink project[2] is an EarthCube building block aimed at tackling the aforementioned challenge specifically within the area of ocean science [3]. Oceanographic research data in the US are maintained by numerous distributed online repositories, for example, the Biological and Chemical Oceanographic Data Management Office (BCO-DMO)[3], Rolling Deck to Repository (R2R)[4] program, Integrated Earth Data Applications (IEDA)[5], and the Index to Marine and Lacustrine Geological Samples (IMLGS)[6], to name a few. The lack of integrated knowledge infrastructure hampers researchers' ability to realize discovery scenarios possible only when multiple repositories are involved. For example, one may be interested in determining if the Global Multi-Resolution Topography (GMRT)[7] synthesis grid [4] contains high-resolution data from a ship's multibeam sonar in the proximity of a specified physiographic feature such as the Lomonosov Ridge, and returning the list of ship expeditions that contributed high-resolution data to those grid cells. One may then wish to determine which principal investigators and research programs are linked to those expeditions; which journal publications, meeting and/or funding awards contain thematic keywords pertaining to the physiographic feature; and which data sets and research products are available for those expeditions at each online repository. The OceanLink project has set out to facilitate such a discovery scenario, which is a vision many oceanographers would hope to see realized.

However, building an integrated knowledge discovery framework on top of those data repositories is a task that is as much socially challenging as it is technically, because the data not only often does not directly align, but more than that, there are fundamental differences in modeling, leading to insufficient overlap for conducting a meaningful integration. OceanLink addresses this challenge using advances in Semantic Web technologies, particularly Linked Data [5] and Ontology Design Patterns (ODPs) [6]. The former allows the repositories to describe and expose their data in a standard syntax that is natural for linking

[1] http://www.earthcube.org/
[2] http://www.oceanlink.org/
[3] http://www.bco-dmo.org/
[4] http://www.rvdata.us/
[5] http://www.iedadata.org/
[6] http://www.seabedsamples.org/
[7] http://www.marine-geo.org/portals/gmrt/

with other data, possibly in different repositories. The latter enables a horizontal integration where semantic alignment occurs for specific purposes between repositories with potentially independent semantic models. Such a horizontal integration is possible through ODPs because it is not advocating an overarching, upper-level ontology that captures a global agreement on all concepts and relationships across all data repositories, something that is often infeasible even within a single scientific domain [7]. Rather, the ODP approach is to specify a set of the so-called *ontology (design) patterns*, each of which is simply a partial ontology that formalizes only one key notion, and to do it in such a robust way that it is alignable with the differing representation choices which had already been made in different repositories.

One such key notion that occurs across many ocean science repositories is the notion of *Cruise*. Roughly speaking, a cruise in ocean science, or an oceanographic cruise, is an expedition conducted on a vessel to the ocean or other navigable water body for particular purposes, mostly related to oceanographic research activities. Cruises holds a critical role in ocean science because most oceanographic research activities such as field observations, data acquisition, and scientific experiments can be accomplished only when researchers gain direct access to the oceans using vessels [8]. Note that a cruise should, e.g., be distinguished from the corresponding vessel as the latter is an actual physical object, whereas the former concerns not just the vessel, but also the corresponding activities carried out while the vessel traverses the route from the starting port to the end port, the project award paying for the costs of the cruise, etc. Specifically, there may be two different cruises conducted on the same vessel, but scheduled for different time periods and possibly traveling along different routes. The US academic research fleet currently possesses over 20 research vessels[8] whose usage is shared and managed among 61 US academic institutions and national laboratories, all members of the University-National Oceanographic Laboratory System (UNOLS)[9].

From a data integration perspective, the importance of the notion of Cruise also lies in the fact that it acts as a type of "glue" that may connect all data about and results from the activities carried out during a cruise. This is also clearly reflected in the earlier example discovery scenario whereby, from GMRT data about a specified physiographic feature at some point-of-interest, one can obtain information about research programs relevant to the data. Hence, formalizing the notion of Cruise would be an important step towards data integration as envisioned by the OceanLink project.

In this paper, we describe an *ontology pattern for oceanographic cruises* that formalizes the notion of Cruise for data integration in the OceanLink project. The remainder of this paper is organized as follows. Section 2 will briefly explain what exactly an ontology design pattern is, discuss different types of ontology design pattern, and the differences between an ontology design pattern and an upper-level ontology. We start the formalization of an ontology design pattern

[8] http://www.unols.org/info/vessels.htm
[9] http://www.unols.org/

for oceanographic cruises in Section 3 by elaborating generic use cases that guide the design choices we need to make in specifying the pattern. Making use of these generic use cases, we formally specify the pattern in Section 4. The formalization is presented using a combination of graphical depictions and logical assertions expressed in the Web Ontology Language (OWL) [9]. This is then followed by a discussion in Section 5 on how the pattern can actually be used in applications, especially within the context of the OceanLink project. Finally, Section 6 concludes.

2 Ontology Design Patterns (ODPs)

Intuitively speaking, an *ontology design pattern* (ODP) is a reusable solution to some frequently occurring ontological modeling problem that emerges in different domains and can act as a building block for more complex ontologies [6]. The scope of modeling problems an ODP may address is quite broad, leading to different kinds of ODPs which are developed to solve them. This ranges from *logical patterns* which model certain logical constructs in a particular formal ontology language, to *alignment patterns* which act as templates representing commonly occurring types of alignments between ontologies, to *content patterns* which encapsulate generic notions within a particular domain of discourse

In the context of horizontal data integration, content patterns are useful to provide unified perspectives over the data while still allowing a rather significant degree of semantic independence between the data repositories. Concretely, each content pattern typically focuses only on one generic notion (e.g., event, organization, or *trajectory* [10]) realized as a self-contained, highly modular ontology that contains some axiomatization (preferrably using a standard like OWL) defining the formal semantics and relationships between the vocabulary items used in it. It represents what constitutes the given notion and what important and widely reusable aspects about it the domain experts have agreed upon. The axiomatization is carefully formulated such that no overly strong (i.e., application specific) ontological commitment is made by the pattern. In comparison to a monolithic, foundational ontology, a content pattern can thus be seen as a snippet that defines only one particular notion without excessive intricacies which a foundational ontology may entail. Relationships to other patterns which define different, but related, notions will still be provided, but not specified in detail. Such characteristics make content patterns more suitable than monolithic foundational ontologies for heterogeneity preservation when integrating knowledge.

3 Cruise: Generic Use Cases

Intuitively, the notion of oceanographic cruise is rather specific compared to the general notion of cruise, since one can obviously also think of sight-seeing cruises, pleasure cruises, or even science cruises which are not used for ocean science purposes. From this perspective, to develop a pattern that is highly

reusable even outside the ocean sciences, a generic notion of cruise would have to be modeled, rather than just the notion of ocean science cruise. However, for the purpose of the OceanLink project, i.e., for the integration of oceanographic data, the more specific notion of adequate. Of course, rather than developing such a pattern from scratch, we will reuse, adjust, combine and extend existing ontology patterns. This is done through established modeling practices while keeping the amount of abstract ontological commitments to a minimum.

In the context of ocean science data repositories within the OceanLink building block, a cruise can be seen as an abstract record that can act as a glue between otherwise separate pieces of information that ocean science data repositories may store. Those pieces of information are derived from generic use cases which guide which existing patterns we can reuse to develop the Cruise pattern. We describe such generic use cases through a number of *competency questions* which represent queries to the pattern.

One kind of query relevant to a cruise concerns the spatiotemporal information contained within the cruise route or trajectory. For example,

(1) "Find all cruises passing through Gulf of Maine in August 2013."
(2) "Show the trajectories of all cruises that are in operation between September and December 2013."

Another kind of query involves querying the vessel on which a cruise is operated.

(3) "List all cruise vessels that departed from Woods Hole in 2012."

Also relevant to a cruise are queries for finding the people who serve in some capacity during the cruise's operation. For example,

(4) "Find the chief scientists of any cruise that collected samples of carbon-isotope data in Lake Superior."

Activities on a cruise may output some dataset and other digital objects stored in libraries or data repositories. Such repository objects may be of interest to some users who may issue a query such as:

(5) "What datasets were produced by the cruise AE0901?"

Finally, some party may also be interested in some administrative information about a cruise, exemplified by the following queries:

(6) "Which cruises are funded by the NSF award DBI-0424599?"
(7) "List all cruises under the Ocean Flux Program."

The above questions illustrate different pieces of information that are related to the notion of Cruise. From Questions 1, 2 and 3, we know that trajectory and vessel are two important components of a cruise. A closer observation would lead us to an understanding that the trajectory and vessel of a cruise are indispensable: there is no cruise without a vessel and a trajectory. From Question 4, we understand that a cruise involves people who hold particular roles in its operation. To answer Question 5, information about an ocean science cruise clearly

has to be related to the data and documents the cruise generated during its operation. Furthermore, due to Questions 6 and 7, it also needs to be related to the information about funding award and program which support the activities embodied by the cruise. In principle, all of these pieces of information are described by their own separate patterns which may possess more detailed information that need not be formulated explicitly in the cruise pattern.

4 Formal Conceptualization Using OWL

The use cases from Section 3 give us an insight that the notion of Cruise can essentially be viewed from three different angles: (1) as the route or trajectory a vessel traverses, hence providing the *spatiotemporal boundary* of a cruise; (2) as the collection of activities performed by *actors* which can be humans or other kinds of agents; and (3) as a placeholder to various pieces of explanatory information that fit neither the trajectory nor the constituting activities, e.g., funding award, cruise type, etc. Points (1) and (2) motivate us to understand a cruise as a type of *event* since events are things that happen at some place and time whereby actors participate by performing some activities or roles. Moreover, by point (3), a cruise is not just a simple event; it is an event adorned with other explanatory information. Specifically, we conceptualize a cruise as *an adorned event undertaken by a vessel traversing through a particular trajectory*. This motivates a design choice where we formalize the Cruise pattern through reusing, adjusting, combining, and extending several already-existing patterns, including the Semantic Trajectory [10], Simple Event Model [11], and the Information Object pattern derived from DOLCE [12].

The following convention is used for all graphical depictions of the pattern, such as in Figure 1. Yellow, rounded square nodes denote classes where a dotted line border means that the class also represents an external pattern whose details are unnecessary to specify within the Cruise pattern itself, i.e., they would be specified elsewhere, in the definition of that pattern. White, oval nodes denote instances defined explicitly in the pattern as controlled terms. All arrows, except the ones with an open arrowhead (denoting subclass relationship) or an angled arrowhead (denoting `rdf:type`-ing), denote (object or data) properties where the direction is from the domain to the range of the denoted property. A dotted line arrow means the property is defined in an external pattern.

In addition to visual depictions (which remain somewhat ambiguous and cannot convey more complex relationships), the pattern is formalized as a set of axioms in the Web Ontology Language (OWL) [9]. We use, however, the more concise, description logic (DL) notation [13] whereby each axiom is of the form $C \sqsubseteq D$, where C is a subclass of D with C, D possibly being non-atomic classes; $C \equiv D$, where C is an equivalent class of D; $C(a)$, where a is an instance of C; or $R_1 \circ \cdots \circ R_k \sqsubseteq S$, which means if x connects to y via a property chain using $R_1, \ldots, R_k$, $k \geq 1$, then x connects to y via the property S. Please see [13] for specifics of this notation and of the formal relationship between DLs and OWL.

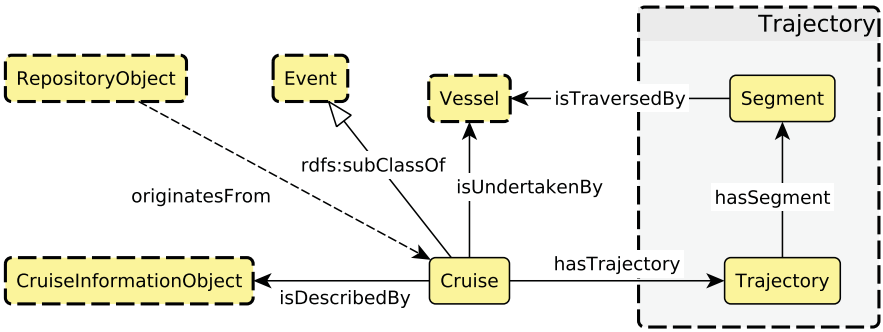

Fig. 1: Overview of the Cruise pattern; dotted box is a snippet of Fig. 2

Figure 1 depicts a high level overview of the Cruise pattern, omitting some details that will be explained and visualized in the remainder of this section. Notice that the relationship between the classes Cruise, Trajectory, and Vessel involves an internal class of the trajectory subpattern, hence the Trajectory class is not drawn with a dotted line border, but rather we put it together with the internal class Segment within the large rectangle with a dotted line border representing the whole trajectory subpattern. Since a cruise is a kind of event, we specify that Cruise is a subclass of the more generic class Event. Adornments to the Cruise pattern are attached through an instance of the CruiseInformationObject class. In addition, Figure 1 also depicts a relationship between library digital objects (represented by the RepositoryObject class that covers datasets, papers, cruise logs, etc.) and cruises through the originatesFrom property. This property is not part of the Cruise pattern, but rather defined in the RepositoryObject pattern which is also begin developed as part of the OceanLink project, but the specification of RepositoryObject pattern is out of scope of this paper. Nonetheless, this relationship allows one to answer queries such as the one in Question 5.

Cruise Trajectory and Vessel Trajectory and vessel are two indispensable, interrelated parts of a cruise. In the OceanLink context, a cruise has exactly one trajectory and is undertaken by exactly one vessel as formalized in axiom (1) below. This vessel must of course be the one that traverses the trajectory. To formalize this, however, we shall first define the notion of trajectory of a cruise.

$$\mathsf{Cruise} \sqsubseteq (=1\ \mathsf{hasTrajectory}.\mathsf{Trajectory}) \sqcap (=1\ \mathsf{isUndertakenBy}.\mathsf{Vessel}) \quad (1)$$

Our modeling of the cruise trajectory is realized through *reusing the Semantic Trajectory pattern* [10] which is very versatile and easily adaptable to many application contexts including ours. A semantic trajectory, as defined in [10], is essentially a sequence of "points", called *fixes*, each of which possesses, at least, a position information and a timestamp. Generally, fixes and *segments* (pairs of consecutive fixes) can additionally be adorned with various geographic

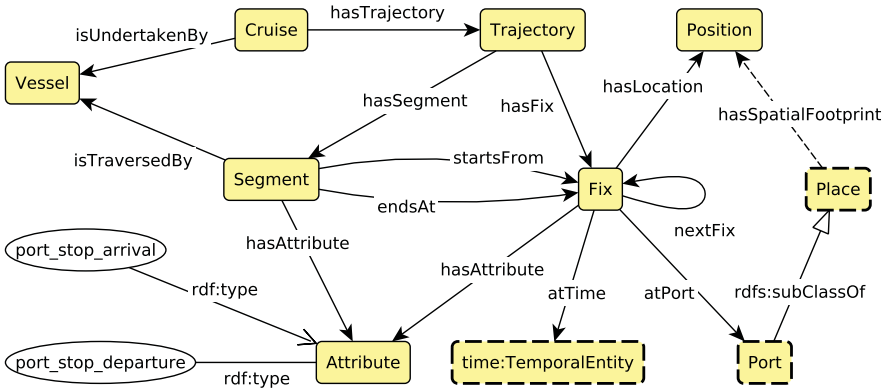

Fig. 2: Trajectory for Cruise

information and domain knowledge enabling a richer information discovery. For OceanLink, reusing the Semantic Trajectory pattern as cruise trajectory leads us to Fig. 2 and a set of DL axioms described below. These axioms are similar to the ones in [10] — axiom (3) is in fact equivalent to axioms (2)-(5) of that paper. There is, however, an important difference leading to a slightly different axiomatization: the ordering of fixes in [10] using the nextFix property is entailed from the given two fixes and the corresponding segment; while here, the ordering is already explicit from the data and segments are auto-instantiated from it.

Note that pairwise-disjointness between classes as well as domain and range restrictions for all properties here are asserted as discussed in the explanation of axioms (24) and (25) further below.

Each fix has a position, a timestamp, and possibly some additional attributes; belongs to a trajectory; and is followed (through the nextFix property) by at most one other fix (axiom (2)). Each segment starts from exactly one fix, ends at exactly one fix, and belongs to a trajectory (axiom (3)). If a fix f is followed by another fix, then exactly one segment starts from f (axiom (4)). Likewise, if a fix f is preceded by another fix, then exactly one segment ends at f (axiom (5)). Axioms (4) and (5), however, do not guarantee that there is only one segment between two consecutive fixes. We can achieve this by ensuring that, whenever a segment s starts from a fix f whose next fix is f', then s must end at f' (i.e., a rule of the form $\mathsf{startsFrom}(x, y) \wedge \mathsf{Fix}(y) \wedge \mathsf{nextFix}(y, z) \rightarrow \mathsf{endsAt}(x, z)$). Since there is exactly one segment ending at f' by axiom (5) and domain/range restrictions for startsFrom and nextFix, the segment auto-instantiated by this axiom will be identified with s. The aforementioned rule is in principle translatable into a description logic property chain axiom (see [14]).

$$\mathsf{Fix} \sqsubseteq \exists \mathsf{hasLocation}.\mathsf{Position} \sqcap \exists \mathsf{atTime}.\mathsf{time{:}TemporalEntity} \sqcap (=1\ \mathsf{hasFix}^{-}.\mathsf{Trajectory}) \sqcap (\leqslant 1\ \mathsf{nextFix}.\mathsf{Fix}) \tag{2}$$

$$\mathsf{Segment} \sqsubseteq (=1\ \mathsf{startsFrom}.\mathsf{Fix}) \sqcap (=1\ \mathsf{endsAt}.\mathsf{Fix}) \sqcap \exists \mathsf{hasSegment}^{-}.\mathsf{Trajectory} \tag{3}$$

$$\exists \mathsf{nextFix}.\top \sqsubseteq (=1\ \mathsf{startsFrom}^{-}.\mathsf{Segment}) \tag{4}$$

$$\exists \mathsf{nextFix}^{-}.\top \sqsubseteq (=1\ \mathsf{endsAt}^{-}.\mathsf{Segment}) \tag{5}$$

$$\mathsf{startsFrom} \circ \mathsf{nextFix} \sqsubseteq \mathsf{endsAt} \tag{6}$$

A position information attached to a fix can be, e.g., geospatial coordinates, and the position acts as an interface to richer geographic information about points-of-interest (POIs). For our need, we simply assume a generic class Place that represents a POI and has the position as its spatial footprint (realized through the hasSpatialFootprint property). Some of the fixes may be of particular interest as they represent ports where the cruise stops during its travel. A port is then here simply modeled as a kind of place (axiom (7)). A fix corresponds to such a port if it has one of the following attributes: `port_stop_arrival` — the fix's timestamp is the arrival time — and `port_stop_departure` — the fix's timestamp is the departure time (axioms (8a,b),(9),(10)). Also, the spatial footprint of the port gives us the fix's location (axiom (11)).

$$\mathsf{Port} \sqsubseteq \mathsf{Place} \tag{7}$$

$$\mathsf{Attribute}(\texttt{port_stop_arrival}), \quad \mathsf{Attribute}(\texttt{port_stop_departure}) \tag{8a,b}$$

$$\exists \mathsf{hasAttribute}.\{\texttt{port_stop_arrival}\} \sqsubseteq \mathsf{Fix} \sqcap \exists \mathsf{atPort}.\mathsf{Port} \tag{9}$$

$$\exists \mathsf{hasAttribute}.\{\texttt{port_stop_departure}\} \sqsubseteq \mathsf{Fix} \sqcap \exists \mathsf{atPort}.\mathsf{Port} \tag{10}$$

$$\mathsf{atPort} \circ \mathsf{hasSpatialFootprint} \sqsubseteq \mathsf{hasLocation} \tag{11}$$

Finally, the vessel by which the cruise is undertaken must be the vessel that traverses the segments in the trajectory of the cruise (axiom (12)).

$$\mathsf{hasTrajectory} \circ \mathsf{hasSegment} \circ \mathsf{isTraversedBy} \sqsubseteq \mathsf{isUndertakenBy} \tag{12}$$

Cruise as Event We realize the modeling of cruises as events (Fig. 3) by reusing the Simple Event Model (SEM) [11]. As in SEM, an event consists of three essential components: place, time and actors. The grey rectangle within the figure represents an Event pattern inspired by SEM and covers the classes and properties that would have been defined there. Information about time and place is omitted there since for the Cruise pattern, they are already inherent within the trajectory. Any property within the Event pattern intended to giving spatiotemporal information in this context can thus be written as a query to the trajectory information of the cruise.

We proceed with modeling the actors within a cruise. Note that SEM does not provide any OWL axiomatization, hence we also axiomatize the part of an event

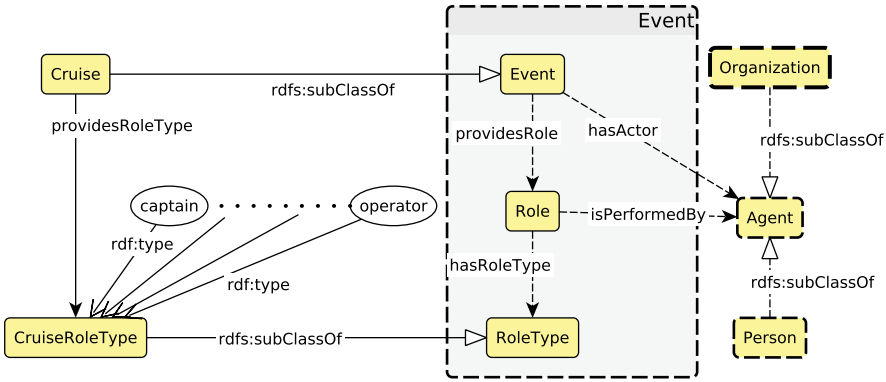

Fig. 3: Cruise as a kind of event.

that concerns the actors. First, we do not enforce an event to always provide a role, but any role it provides must have exactly one type and be performed by some agent (axiom (13)). Also, if a role provided by an event is performed by an agent, then this agent is an actor of the event (axiom (14)). We make no assumption about agents except that people and organizations are considered agents, and these are asserted by the Person and Organization patterns which would be described separately.

$$\mathsf{Role} \sqcap \exists \mathsf{providesRole}^{-}.\mathsf{Event} \sqsubseteq (=1\ \mathsf{hasRoleType}.\mathsf{RoleType}) \sqcap \exists \mathsf{isPerformedBy}.\mathsf{Agent} \tag{13}$$

$$\mathsf{providesRole} \circ \mathsf{isPerformedBy} \sqsubseteq \mathsf{hasActor} \tag{14}$$

Further, a cruise is an event (axiom (15)) that also provides a predefined set of role types. For OceanLink, there are 20 cruise role types (axiom (16) and all of (17a-t)) represented as the following named individuals (*): `captain`, `chief_engineer`, `scientist`, `chief_scientist`, `cochief_scientist`, `postdoc_scientist`, `student`, `graduate_student`, `undergraduate_student`, `k12_student`, `higher_ed_educator`, `k12_educator`, `technician`, `marine_technician`, `lead_marine_technician`, `inspector`, `observer`, `foreign_observer`, `other_observer`, `scheduler`, `operator`, and `other_role`. All cruise role types have to be provided by any cruise. This can be expressed as the rule $\mathsf{Cruise}(x) \wedge \mathsf{CruiseRoleType}(y) \rightarrow \mathsf{providesRoleType}(x, y)$, and translated into OWL this becomes axioms (18) and (19a,b) where R_{Cruise} and $R_{\mathsf{CruiseRoleType}}$ are additional object properties needed to encode the atoms $\mathsf{Cruise}(x)$ and $\mathsf{CruiseRoleType}(y)$ in the above rule using OWL's self-restriction (`owl:hasSelf`), and `owl:topObjectProperty` is the predefined OWL object property that connects all pairs of individuals.

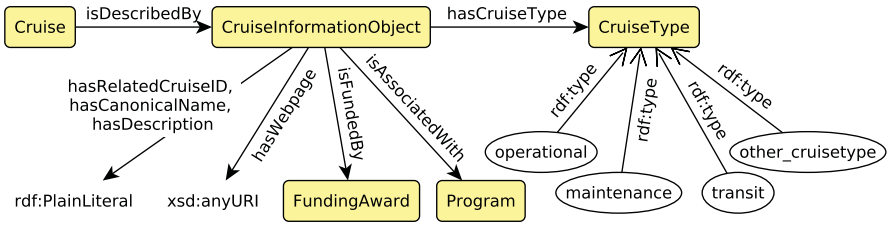

Fig. 4: Cruise Information Object.

$$\mathsf{Cruise} \sqsubseteq \mathsf{Event} \tag{15}$$

$$\mathsf{CruiseRoleType} \sqsubseteq \mathsf{RoleType} \tag{16}$$

$$\mathsf{CruiseRoleType}(x) \text{ for every role type } x \text{ in } (*) \tag{17a-t}$$

$$R_{\mathsf{Cruise}} \circ \mathsf{owl{:}topObjectProperty} \circ R_{\mathsf{CruiseRoleType}} \sqsubseteq \mathsf{providesRoleType} \tag{18}$$

$$\mathsf{Cruise} \equiv \exists R_{\mathsf{Cruise}}.\mathsf{Self}, \quad \mathsf{CruiseRoleType} \equiv \exists R_{\mathsf{CruiseRoleType}}.\mathsf{Self} \tag{19a,b}$$

Cruise Information Object Apart from spatiotemporal information and actor information, there is other explanatory information important for a cruise such as the funding award, cruise webpage, etc. These pieces of information are aggregated into an information object (Figure 4). Each cruise is then described by exactly one instance of such an information object (axiom (20)). Most explanatory information is optional, however, for OceanLink, exactly one cruise type is required for each cruise information object (axiom (21)) and the set of cruise types is predefined (axiom (22)).

$$\mathsf{Cruise} \sqsubseteq (=1\ \mathsf{isDescribedBy.CruiseInformationObject}) \tag{20}$$

$$\mathsf{CruiseInformationObject} \sqsubseteq (=1\ \mathsf{hasCruiseType.CruiseType}) \tag{21}$$

$$\mathsf{CruiseType} \equiv \{\mathtt{operational}, \mathtt{transit}, \mathtt{maintenance}, \mathtt{other_cruisetype}\} \tag{22}$$

Finally, in the OceanLink context, a cruise is operational if, and only if, it has a chief scientist and is funded by some funding award. That is,

$$\begin{aligned} &\mathsf{Cruise} \sqcap \exists \mathsf{isDescribedBy}.\exists \mathsf{hasCruiseType}.\{\mathtt{operational}\} \\ &\equiv \exists \mathsf{providesRole}.(\mathsf{Role} \sqcap \exists \mathsf{hasRoleType}.\{\mathtt{chief_scientist}\}) \\ &\quad \sqcap \exists \mathsf{isFundedBy}.\mathsf{FundingAward} \end{aligned} \tag{23}$$

Class Pairwise-Disjointness, Domains and Ranges of Properties We assert that all classes in the pattern are pairwise disjoint, except for each of the following pairs: (Cruise, Event), (Port, Place) and (CruiseRoleType, RoleType)

— each is a subclass-superclass pair. The following axiom exemplifies pairwise-disjointness of Cruise and Vessel.

$$\mathsf{Cruise} \sqcap \mathsf{Vessel} \sqsubseteq \bot \tag{24}$$

Also, the unique name assumption is made for named individuals, e.g., `port_stop_arrival` and `port_stop_departure` refer to different individuals. In addition, we also assert domain and range restrictions to all of the properties in the pattern. We provide the following as example how to enforce this. For object properties, we include *guarded* domain and range restrictions, e.g., for the hasFix property, they are, respectively,

$$\exists \mathsf{hasFix}.\mathsf{Fix} \sqsubseteq \mathsf{Trajectory} \text{ and } \exists \mathsf{hasFix}^{-}.\mathsf{Trajectory} \sqsubseteq \mathsf{Fix}. \tag{25}$$

We do not enforce unguarded domain and range restrictions, e.g., of the form $\exists \mathsf{hasFix}.\top \sqsubseteq \mathsf{Fix}$ and $\exists \mathsf{hasFix}^{-}.\top \sqsubseteq \mathsf{Trajectory}$, since they constitute very strong ontological commitments which are not required for the modeling; thus we stick to good modeling practice and guard domains and ranges. For data properties, however, only domain restrictions are guarded; range restriction is unguarded because the inverses of data properties cannot be expressed in OWL, e.g., for the hasRelatedCruiseID data property, we assert the triple hasRelatedCruiseID `rdfs : range` xsd:string.

Views for the Cruise Pattern In summary, the Cruise pattern glues together three existing patterns: Trajectory, Event, and Information Object. This combination may make the Cruise pattern a bit complicated, both for data providers as well as for users. To aid them in the readability and ease of use, it is often useful to specify some semantic "shortcuts" that capture some common queries over the pattern. Such shortcuts, called *views*, can be defined depending on the application needs and typically expressed as rules which can be translated into OWL axioms (c.f. how (18), (19a,b) were obtained). For example, hasChiefScientist property connects a cruise and its chief scientist:

$$\begin{aligned} \mathsf{Cruise}(x) \wedge \mathsf{providesRole}(x, y) \wedge \mathsf{hasRoleType}(y, \texttt{chief_scientist}) \\ \wedge\ \mathsf{isPerformedBy}(y, z) \wedge \mathsf{Person}(z) \rightarrow \mathsf{hasChiefScientist}(x, z) \end{aligned} \tag{26}$$

Another example is the starting port of a cruise (the ending port is similar), obtained from axiom (27) and rule (28).

$$\mathsf{Fix} \sqcap \neg \exists \mathsf{endsAt}^{-}.\mathsf{Segment} \sqsubseteq \mathsf{StartingFix} \tag{27}$$

$$\begin{aligned} \mathsf{Cruise}(x) \wedge \mathsf{hasTrajectory}(x, y) \wedge \mathsf{hasFix}(y, z) \wedge \mathsf{StartingFix}(z) \\ \wedge\ \mathsf{atPort}(z, p) \rightarrow \mathsf{hasStartingPort}(x, p) \end{aligned} \tag{28}$$

5 Application Scenarios

In the context of data integration and centralized discovery within the OceanLink project, ontology patterns are very useful to define common vocabularies

for expressing a user's information need. The project envision a knowledge infrastructure whose architecture is divided into four major layers, from top to down: (1) user interface (UI); (2) UI views; (3) a collection of OceanLink patterns, including the Cruise pattern; and (4) data sources which for OceanLink, currently include BCO-DMO, R2R, WHOI library, the American Geophysical Union (AGU)'s conference data and NSF meeting abstracts. The key point here is that the patterns *do not force* the *adoption* of the patterns' vocabularies by each data source, but rather, require the data source to expose its content as an RDF dataset and *provide* a *mapping* from the dataset to the patterns.

Each data source then types its instance data against the classes and properties in the patterns. For example, in R2R repository, all cruises are typed (via `rdf:type`) to `r2r:Cruise`, while in BCO-DMO repository, all cruises are instances of `bcodmo:Deployment` only of platform type `bcodmo:Vessel`. The data provider's task is then to ensure that their cruise instances would also be typed to Cruise class from the pattern. This can be expressed as the following two SPARQL queries for R2R and BCO-DMO, respectively:

```
CONSTRUCT ?x rdf:type :Cruise WHERE { ?x rdf:type r2r:Cruise. }
CONSTRUCT ?x a :Cruise WHERE { ?x a bcodmo:Deployment;
                    bcodmo:ofPlatform [a bcodmo:Vessel]. }
```

Using these kinds of SPARQL queries applied to both classes (e.g., `:Cruise`) and properties in the patterns, each data repository produces a *derived graph* (from the set of triples formed by the `CONSTRUCT` clause) that can be aggregated and cached at layer (3). Such a derived graph intuitively projects the data from a repository according to the structure specified by the patterns, hence realizing the mapping from the data to the pattern. With this in place, a user can then issue a query from the top layers to layer (3) using only vocabularies defined in layer (3) as illustrated in the subsequent paragraphs.

Suppose one is interested in finding all ports at which the researcher named "Mak Saito" stopped by in any of his expedition. This can be expressed as the following SPARQL query over the Cruise pattern as follows, assuming `:hasLegalName` is a property defined in the Person pattern:

```
DESCRIBE ?port WHERE {
  ?port a :Port.
  ?cruise :hasTrajectory [ :hasFix [ :atPort ?port ] ];
          :hasActor [a :Person; :hasLegalName "Mak Saito"]. }
```

For another example, suppose one wishes to find out who joined any cruise that went through Gulf of Maine, what their role was in the cruise, and what funding award did support their trip. This can be expressed using SPARQL as:

```
SELECT ?name ?role ?fund WHERE {
  ?cruise :isDescribedBy [ :isFundedBy [ :hasAwardID ?fund ] ];
          :providesRole  [ :hasRoleType ?role;
              :isPerformedBy [a :Person; :hasLegalName ?name]];
          :hasFix [ :hasLocation ?pos ].
```

```
?pl :hasSpatialFootprint ?pos; rdfs:label ?plname.
FILTER regex(?plname, "Gulf of Maine", "i").
```

Clearly, satisfactory answer to such queries depend on how complete the derived graphs are constructed by the data provider. A rather crude alignment using SPARQL's `CONSTRUCT` clause above works only when a straightforward correspondence between the data and the patterns can be obtained. Otherwise, more expressive alignment schemes, possibly involving complicated inferencing may need to be employed, and further discussion on how such inferencing can be done is out of scope for this paper. On the other hand, this alignment-based approach is highly flexible because if a new data source needs to be added, the data provider simply has to establish the alignment to the pattern. Furthermore, there is no obligation for the data providers to completely specify such an alignment to every vocabularies in the patterns. It is up to them to choose which vocabulary items in the patterns they want to map with. The only consequence is that the less complete their alignment is, the less amount of data can be discovered from their repositories.

6 Conclusions

In this paper, we presented an ontology design pattern for oceanographic cruises. We showed how this pattern was specified as a combination and reuse of existing patterns: trajectory, event and information object. We then demonstrated the applicability of this pattern in integrated knowledge scenarios within the OceanLink project and also argued for the general reusability of the pattern.

One direction for future work is regarding the actual implementation and application of this pattern within OceanLink's integrated knowledge discovery service that is currently being implemented. In particular, we plan to study the effectiveness and ease of use from a user's perspective in serving a variety of information needs through the OceanLink service. From data providers' perspective, we will study the ease of use in aligning their data to the pattern as well as the practical extensibility of the pattern.

Another direction for future work that is not constrained within the OceanLink project we also wish to pursue concerns a number of more fundamental, theoretical questions arising from our experience in specifying and implementing this pattern. This includes, among others, problems regarding the use of such a pattern for data integrity; the expressivity and computational issues surrounding views; also, how flexible the pattern is for data integration, especially if one wish to add new data sources.

Acknowledgement. The presented work has been primarily funded by the National Science Foundation under the OceanLink project, award 1354778 *EAGER: Collaborative Research: EarthCube Building Blocks, Leveraging Semantics and Linked Data for Geoscience Data Sharing and Discovery.* Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Malik, T., Foster, I.: Addressing data access needs of the long-tail distribution of geoscientists. In: Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International. (July 2012) 5348–5351
2. Mascarelli, A.L.: Data's shameful neglect. Nature **461**(7261) (2009) 145
3. Committee on an Ocean Infrastructure Strategy for U.S. Ocean Research in 2030; National Research Council: Critical Infrastructure for Ocean Research and Societal Needs in 2030. The National Academies Press (2011)
4. Ryan, W.B.F., Carbotte, S.M., Coplan, J.O., O'Hara, S., Melkonian, A., Arko, R., Weissel, R.A., Ferrini, V., Goodwillie, A., Nitsche, F., Bonczkowski, J., Zemsky, R.: Global multi-resolution topography synthesis. Geochemistry, Geophysics, Geosystems **10**(3) (2009)
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems **5**(3) (2009) 1–22
6. Gangemi, A.: Ontology design patterns for semantic web content. In Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A., eds.: The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings. Volume 3729 of Lecture Notes in Computer Science., Springer (2005) 262–276
7. Janowicz, K., Hitzler, P.: The digital earth as knowledge engine. Semantic Web **3**(3) (2012) 213–221
8. Committee on Evolution of the National Oceanographic Research Fleet; National Research Council: Science at Sea: Meeting Future Oceanographic Goals with a Robust Academic Research Fleet. The National Academies Press (2009)
9. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S., eds.: OWL 2 Web Ontology Language: Primer. W3C Recommendation 27 October 2009 (2009) Available from http://www.w3.org/TR/owl2-primer/.
10. Hu, Y., Janowicz, K., Carral, D., Scheider, S., Kuhn, W., Berg-Cross, G., Hitzler, P., Dean, M.: A geo-ontology design pattern for semantic trajectories. In Tenbrink, T., Stell, J.G., Galton, A., Wood, Z., eds.: Spatial Information Theory - 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013. Proceedings. Volume 8116 of Lecture Notes in Computer Science., Heidelberg, Springer (2013) 438–456
11. van Hage, W.R., Malaisé, V., Segers, R., Hollink, L., Schreiber, G.: Design and use of the Simple Event Model (SEM). Journal of Web Semantics **9**(2) (2011) 128–136
12. Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Vembu, S., Baumann, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Porzel, R., Zorn, H.P., Micelli, V., Schmidt, C., Weiten, M., Burkhardt, F., Zhou, J.: DOLCE ergo SUMO: On Foundational and Domain Models in the SmartWeb Integrated Ontology (SWIntO). Journal of Web Semantics **5**(3) (2007) 156–174
13. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
14. Krisnadhi, A., Maier, F., Hitzler, P.: OWL and Rules. In Polleres, A., et al., eds.: Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, Tutorial Lectures. Volume 6848 of Lecture Notes in Computer Science., Springer, Heidelberg (2011) 382–415