# A Pattern for Features on a Hierarchical Spatial Grid


Cogan Shimizu
coganmshimizu@ksu.edu
Kansas State University
Manhattan, KS, USA

Rui Zhu
ruizhu@ucsb.edu
UC Santa Barbara
Santa Barbara, CA, USA

Gengchen Mai
maigch@cs.stanford.edu
Stanford University
Stanford, CA, USA

Colby K. Fisher
cfish25@gmail.com
Hydronos Labs LLC
Princeton, NJ, USA

Ling Cai
lingcai@geog.ucsb.edu
UC Santa Barbara
Santa Barbara, CA, USA

Mark Schildhauer
schild@nceas.ucsb.edu
National Center for Ecological Analysis & Synthesis
Santa Barbara, CA, USA

Krzysztof Janowicz
janowicz@ucsb.edu
UC, Santa Barbara
Santa Barbara, CA, USA

Pascal Hitzler
hitzler@ksu.edu
Kansas State University
Manhattan, KS, USA

Lu Zhou
luzhou@ksu.edu
Kansas State University
Manhattan, KS, USA

Shirly Stephen
shirlystephen@ucsb.edu
UC Santa Barbara
Santa Barbara, CA, USA



## ABSTRACT

The integration of data along a common spatial component remains an obstacle in many problem spaces. One promising method for integrating data in such a way is through the use of a common, underlying spatial reference system, such as a Discrete Global Grid (e.g., the S2 Grid System), and pre-computing spatial relations between features and the constituent components at a spatial resolution appropriate for the data and use case. That is, by emphasizing the notion of the cell, we can examine what is in a cell, predict contents of its parent and child cells, and quickly get an overview of spatially co-located features and regions of interest without having to directly compute spatial interactions. This paper provides an ontology design pattern, to be used as a structural template, for modeling how features or regions map onto a hierarchical grid system and addresses how the attributes of these features may be inherited upwards or downwards through the hierarchy. We furthermore provide a motivating example and implementation.



## CCS CONCEPTS

• **Information systems** → **Geographic information systems**; **Semantic web description languages**; **Ontologies**.

## KEYWORDS

geoinformation science, ontology design pattern, ontology engineering

**ACM Reference Format:**
Cogan Shimizu, Rui Zhu, Gengchen Mai, Colby K. Fisher, Ling Cai, Mark Schildhauer, Krzysztof Janowicz, Pascal Hitzler, Lu Zhou, and Shirly Stephen. 2021. A Pattern for Features on a Hierarchical Spatial Grid. In *The 10th International Joint Conference on Knowledge Graphs (IJCKG'21), December 6–8, 2021, Virtual Event, Thailand*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3502223.3502236



Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).


*IJCKG'21, December 6–8, 2021, Virtual Event, Thailand*


© 2021 Copyright held by the owner/author(s).


ACM ISBN 978-1-4503-9565-6/21/12.
https://doi.org/10.1145/3502223.3502236


## 1 INTRODUCTION

Integrating data along a common spatial component remains an obstacle in many problem spaces, especially when using a knowledge graph (KG). In traditional linked data approaches, such data is represented as geometries directly on the Earth's surface (e.g., using GeoSPARQL [2]). However, this can result in complex queries to the KG with high execution times as the number of overlapping geometries grows. This is due to necessarily computing the spatial relations between the geometries in the query result on the fly.

Another challenge is integrating geospatial vector and raster data into a common schema. Many approaches focus on modeling discrete geographic features (e.g., cities, countries, and lakes), and thus do not readily support modeling raster data (e.g., remote sensing images, interpolated temperature or precipitation observations, or gridded model weather outputs). Prior work, such as RDF Data Cube [4] attempts to model raster data as a set of cells, but may result in exponential growth of the graph. Moreover, since different raster datasets have various spatial (and temporal) resolution, spatial reference systems, and other sensor metadata, the triplified raster cell information is actually less useful, difficult to reuse, and not easily integrated with other datasets. In this work, we mainly

focus on answering the following core question: *What is a common geospatial representation framework for both geospatial vector and raster data that can facilitate easy geospatial data integration, meaningful regional summaries, and can be queried efficiently?*

We propose to use a Discrete Global Grid (DGG) [3] as the common spatial data frame to achieve a compromise between data precision (i.e., the fidelity of the geometries underlying geospatial phenomena), access speed, and ease of data integration among multiple vector and raster data sets, all while supporting globally unique identifiers.

A Discrete Global Grid is a hierarchical partitioning of the surface of the Earth. That is, the Earth is partitioned into some number of "top level" cells, which are further partitioned into child cells, and so on until the desired level of spatial resolution is achieved. The core motivation for this work was the S2 Grid System[1], but other DGGs will also hold (e.g., Uber's H3[2]).

A DGG allows for a common underlying spatial reference system for both geospatial vector and raster datasets, for which we may pre-compute spatial relations between different features or regions for vector data and pre-compute summary statistics for a given cell at a certain level from different raster datasets. By emphasizing the notion of a cell, we can examine what exactly is in a cell, predict or infer contents of its parent and child cells, and quickly get an overview of spatially co-located features and regions of interest without having to directly compute spatial relations.

As such, we have developed an ontology design pattern for quickly and intuitively adapting new data for the DGG paradigm.

The pattern is meant to be used as a "structural template" which can be easily adapted to novel use cases through template-based instantiation [7]. An in-depth discussion of each of the classes and our modeling decisions is provided in the next section. We also provide an alternate version of the pattern which has been extended with semantic shortcuts, and is provided in Section 2.1.

To our knowledge, there is no other pattern that models a DGG alongside features and regions.[3] While there are several common ontologies for modeling features and regions, we are aiming for an abstract generalization. Indeed, in Section 3, we provide an example instantiation of the pattern, also called a module, where we have used existing standards or vocabularies, where possible, as well as an example population of said module. Finally, in Section 4, we conclude with next steps.

## 2 THE HIERARCHICAL CELL FEATURE PATTERN

The purpose of this pattern is to model how features and regions interact with an underlying Discrete Global Grid. Further, it clarifies for how the attributes of these features and regions may be inherited through the hierarchy of cells.

Our pattern was developed using the Modular Ontology Modeling workflow [19], but scoped only to a single pattern. This means that our approach was data driven and use case driven. The axiomatization was done in two steps. First, we followed the systematic

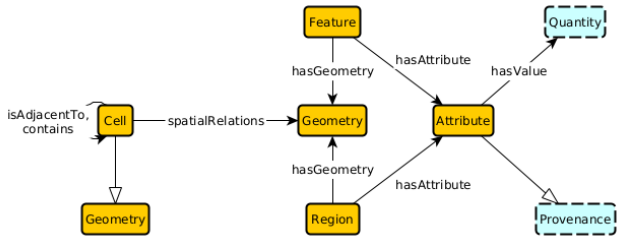



**Figure 1: Schema diagram for the Cell Feature Pattern. Yellow boxes are classes; blue dashed boxes are also classes, but are an acknowledgement of some external dependency or complexity (i.e., they are left un-modeled in the pattern); edges with filled arrows are object properties; and edges with open arrow heads represent subclass relations.**

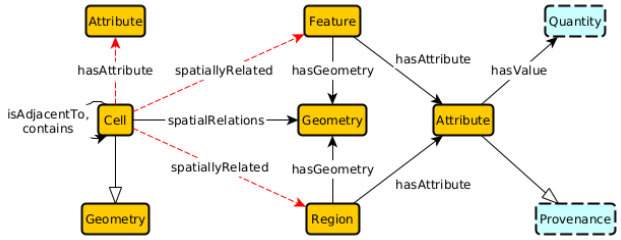



**Figure 2: Schema diagram for the Cell Features Pattern including shortcuts. This uses the same graphical syntax as Figure 1. Red dashed arrows indicate the presence of a shortcut, which are characterized by a role chain, are shown in Section 2.1.**

axiomatization process as outlined in the MOMo workflow. This means for each node-edge-node construct in the schema diagram, we measured the applicability of each of 17 axiom patterns [5], retaining those that do apply. Secondly, we examined available data and discussed with domain experts to generate additional axioms that may involve more than one predicate.

As previously stated, we have developed two patterns: the base pattern, and an extended version with semantic shortcuts. Figure 1 shows the schema diagram for our base pattern. Figure 2 shows the shortcuts that we have added to the base pattern, which are displayed using a red dashed arrow. The purpose of these is to allow an easier mapping between published data and an ontology utilizing the pattern. These are further discussed in Section 2.1.

The resources for this pattern can be found in our online repository.[4] The pattern has been annotated with the Ontology Design Pattern Representation Language (OPLa) [9] and its extension [8] for documenting its use cases using the OPLaTool [20] for Protégé[5]. Documentation has been generated for the pattern using [22] according to community guidelines for documenting patterns [15].

[1] https://s2geometry.io/
[2] https://eng.uber.com/h3/
[3] One existing ontology that provides an OWL version of the abstract specs for DGG is https://github.com/surroundaustralia/dggs-as-ont.
[4] https://github.com/KnowWhereGraph/hierarchical-cell-features
[5] https://protege.stanford.edu/

The rest of this section describes each of the classes present in the pattern. Where applicable, we provide any considerations for each class regarding the modeling decisions behind related aspects of the pattern, which we believe will help prospective users of the pattern to modify the pattern to suit their needs.

We only list each axiom exactly once in Description Logic [1], under the class where it is first applicable, followed by a natural language description. In Section 2.1, we discuss the shortcuts that have been added to the base pattern.

## Cell

$$\mathsf{Cell} \sqsubseteq \mathsf{Geometry} \tag{1}$$
$$\mathsf{Cell} \sqsubseteq \forall \mathsf{spatialRelations.Geometry} \tag{2}$$
$$\mathsf{Cell} \sqsubseteq \geq 0 \mathsf{spatialRelations.Cell} \tag{3}$$
$$\mathsf{Cell} \sqsubseteq \geq 0 \mathsf{isAdjacentTo.Cell} \tag{4}$$
$$\mathsf{Cell} \sqsubseteq \forall \mathsf{isAdjacentTo.Cell} \tag{5}$$
$$\exists \mathsf{isAdjacentTo.Cell} \sqsubseteq \mathsf{Cell} \tag{6}$$
$$\mathsf{Cell} \sqsubseteq \geq 0 \mathsf{contains.Cell} \tag{7}$$
$$\mathsf{Cell} \sqsubseteq \forall \mathsf{contains.Cell} \tag{8}$$
$$\exists \mathsf{contains.Cell} \sqsubseteq \mathsf{Cell} \tag{9}$$

$$\mathsf{isFullyContainedIn} \sqsubseteq \mathsf{contains}^{-} \tag{10}$$
$$\mathsf{contains} \circ \mathsf{contains} \sqsubseteq \mathsf{contains} \tag{11}$$
$$\mathsf{isFullyContainedIn} \circ \mathsf{isFullyContainedIn} \sqsubseteq \mathsf{isFullyContainedIn} \tag{12}$$

*Description.* A Cell is the primary component that comprises a hierarchical grid system. They are spatially related to other geometries, including other cells. Axiom 1 states that every Cell is a Geometry. Axiom 2, 5, and 8 are scoped range axioms, while axioms 3 and 9 are scoped domain axioms. Notably, we do not restrict the domain of our spatial relations to Cell. Axioms 3, 4 and 7 are *structural tautologies*, which are meaningless in the context of logic, but should be used instead to guide the usage of the predicate. This is similar to Schema.org's predicates, schema:domainIncludes and schema:rangeIncludes. Axiom 10 defines the inverse of contains as isFullyContainedIn. Axioms 11 and 12 state that both properties are transitive.

*Considerations.*

- The spatialRelations property is a placeholder for any number of spatial or topological relational systems (e.g., RCC5 [14], RCC8 [17], or DE-9IM [24]).
- Note that a Cell is a Geometry. This means that it wholly exists as an abstraction, it does not technically exist in the real world, but is really a projection of a boundary onto a surface. Indeed, this is the reason why the shortcuts (as further explained in Section 2.1), do not appear in the base pattern. The Cell, itself, does not have, for example, any non-abstract attributes.
- We do not explicitly model the number of subcells that a cell may have. For example, in the S2 Grid System, each Cell has four children, whereas H3[6] has six children. This axiom can be specified using an exact cardinality constraint, such as the following.

$$\mathsf{Cell} \sqsubseteq =4 \mathsf{hasSubcell.Cell}$$

However, this would not hold for the lowest level of spatial resolution.
- We do not provide a mechanism by which to model the *level* of the cell. That is, "how many parents does the cell have?" If desired, however, it can be added in a few different ways. First, one can add in a datatype property with filler: hasLevel.xsd:nonNegativeInteger. Secondly, one may utilize an axiom to specify the maximum and minimum level (i.e., the cells with no parents or with no children), but this does not model any intermediate level. Finally, one can leave the information implicit in the minted URI for the cell.

[6] https://eng.uber.com/h3/

## Feature

$$\mathsf{Feature} \sqsubseteq \exists \mathsf{hasGeometry.Geometry} \tag{13}$$
$$\mathsf{Feature} \sqsubseteq \exists \mathsf{hasAttribute.Attribute} \tag{14}$$

*Description.* A Feature encapsulates the notion of a feature, characteristic, or aspect of the surface upon which the hierarchical grid system has been applied. A Feature is constituted by Attributes, which can be viewed as the more particular data about the feature. For example, a wildfire may be considered to be a Feature, but there are many different attributes that it may have beyond its geometry, such as the date of fire ignition or the amount of damage it has caused, to date. Axioms 13 and 14 state that every Feature has at least one Geometry and Attribute.

*Considerations.* The existential axioms are strong ontological commitment. However, we believe a Feature without a Geometry or any Attributes would not very interesting from the perspective of the pattern. In case it is too strong a commitment, we recommend replacing the axioms with structural tautologies instead. Note that (due to the Open World Assumption) having a geometry differs from knowing the geometry or whether such geometry is crisp or fuzzy.

$$\mathsf{Feature} \sqsubseteq \geq 0 \mathsf{hasGeometry.Geometry}$$
$$\mathsf{Feature} \sqsubseteq \geq 0 \mathsf{hasAttribute.Attribute}$$

## Region

$$\mathsf{Region} \sqsubseteq \exists \mathsf{hasGeometry.Geometry} \tag{15}$$
$$\mathsf{Region} \sqsubseteq \geq 0 \mathsf{hasAttribute.Attribute} \tag{16}$$

*Description.* A Region is a socioculturally or geopolitically significant area. For example, Kansas is a state (administrative *region*) in the United States. It can also be regarded as a Feature. However, we leave it distinct at this abstract level.

Axiom 15 states that a Region must have a Geometry. Axiom 16 is a structural tautology that indicates Regions should have Attributes.

*Considerations.*

- As noted above, a Region is typically considered to be a Feature. For example, we could model it as a Feature with an

Attribute that states its sociocultural importance, or simply the name of the region.
- The existentiality in Axiom 15 can lead to problems when dealing with fuzzy regions. For example, what is the exact geometry of Southern California? In the event that these are the regions of interest, we would suggest using a structural tautology in its place.

### Attribute

$$\text{Attribute} \sqsubseteq \text{Provenance} \tag{17}$$
$$\text{Attribute} \sqsubseteq \forall \text{hasValue.Quantity} \tag{18}$$
$$\text{Attribute} \sqsubseteq \exists \text{hasValue.Quantity} \tag{19}$$

*Description.* An Attribute is a generic term for a characteristic of the Feature. Axiom 17 states that every Attribute is also a Provenance node. Axiom 18 constrains the range to Quantity and Axiom 19 states that every Attribute must have at least Quantity filler for hasValue.

### Quantity

*Description.* A Quantity acts as a placeholder for an arbitrary conceptualization of a quantity. This "external hook" node is included here as a prompt for consideration during the knowledge graph development process. We provide a possible instantiation for Quantity in Section 3.2.

*Considerations.*

- Depending on the nature of the data, we may simply want to replace the Attribute-hasQuantity-Quantity construct with a simple data property reflecting the captured Attribute. This, however, does preempt the ability record provenance directly.
- Additionally, we may also replace Quantity with a typed literal, such as cdt:ucum[16], which uses a controlled syntax to represent values and units together in a string.

### Provenance

*Description.* The Provenance acts as a placeholder for an arbitrary conceptualization of the provenance of the Attribute. This "external hook" node is included here as a prompt for consideration during the knowledge graph development process. We provide a possible instantiation for the Provenance in Section 3.2.

*Considerations.*

- This style of provenance may not suit all use cases, especially as the number of Attributes grows in the knowledge graph.
- One possible instantiation for the Provenance class may be the EntityWithProvenance pattern from the Modular Ontology Design Library [21], which is representative of the core of the Provenance Ontology [18].

## 2.1 Semantic Shortcuts

We define the semantic shortcuts using role chains, as follows.

$$\text{spatialRelations} \circ \text{hasGeometry}^{-} \sqsubseteq \text{spatiallyRelated} \tag{20}$$
$$\text{spatiallyRelated} \circ \text{hasAttribute} \sqsubseteq \text{hasAttribute} \tag{21}$$

Notably, it would be useful to have also the converse of Axioms 20 and 21, but unfortunately these can not be expressed in OWL [10], as it does not allow for right hand side role chains.

$$\text{Cell} \sqsubseteq \geq 0 \text{hasAttribute.Attribute} \tag{22}$$
$$\text{Cell} \sqsubseteq \geq 0 \text{spatiallyRelated.Feature} \tag{23}$$
$$\text{Cell} \sqsubseteq \geq 0 \text{spatiallyRelated.Region} \tag{24}$$
$$\text{Cell} \sqsubseteq \forall \text{hasAttribute.Attribute} \tag{25}$$
$$\text{Cell} \sqsubseteq \forall \text{spatiallyRelated.Feature} \tag{26}$$
$$\text{Cell} \sqsubseteq \forall \text{spatiallyRelated.Region} \tag{27}$$

Axioms 22-24 are structural tautologies for guiding how the shortcuts should be used. Axioms 25-27 restrict the ranges, as shown in the schema diagram.

## 2.2 Spatial Considerations

Of particular interest are how spatial relations between Cells and Features and Regions can be inherited. The exact nature of this inheritance depends on the type of spatial relation. For example, we know that if Cell is spatiallyRelated to a Feature, then the parent of that Cell is also spatiallyRelated. We specify this with the following axiom.

$$\text{contains} \circ \text{spatiallyRelated} \sqsubseteq \text{spatiallyRelated} \tag{28}$$

This axiom also covers the notion of a Cell that is fully contained by a Feature or Region, then all subcells are fully contained in the Feature or Region. This is important for semantically compressing the graph.

# 3 UTILIZING THE PATTERN

## 3.1 Scenario

The underlying motivation of this work is driven by our project, KnowWhereGraph[7] [11, 13]. It provides a densely connected, cross-domain knowledge graph to support applications in environmental intelligence via on-demand access to area briefings at a high spatial and temporal resolution for any location on the Earth's surface. To do so, data about extreme events, administrative boundaries, soils, crops, climate, and so on, is incorporated into a knowledge graph and pre-integrated so as to to provide answers to questions such as 'What is here?', 'What happened here before?', 'How does this region compare to other regions?', and so on.

In the agricultural sector, KnowWhereGraph is used to enhance assessment and strategic planning during real-time hazard events affecting the food supply chain by providing online analysis, forecasting, and alerts that are enriched with location and context-specific intelligence, to ensure key stakeholders throughout the supply chain are ready with backup strategies to keep products moving. It also allows farmers and growers to identify how they can be better prepared to mitigate and build resilience in the face of such events.

In the next sections, we describe our modularization of the pattern, as relating to our use case scenario, and, as an example, a (curated) instance graph that is returned from a series of queries intended to assess the impact of ongoing wildfires, smoke plumes,

[7] https://knowwheregraph.org/

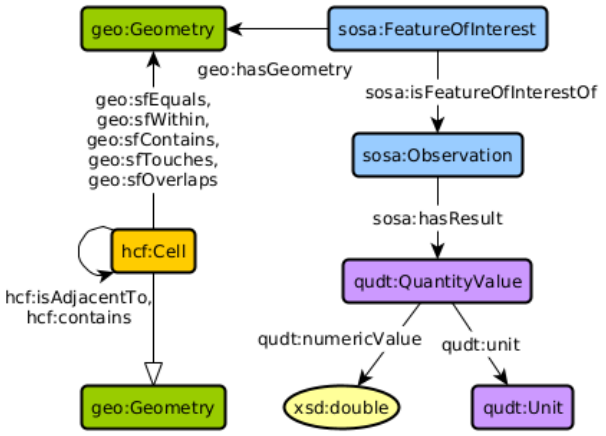



**Figure 3: The schema diagram for the instantiation, or modularization, of the Hierarchical Cell Features Pattern. The rectangular cells are color coded according to their namespaces. We use geo: for the GeoSPARQL standard; sosa: for the Sensors, Observations, Samples, and Actuators ontology; and qudt for the Quantity, Units, Dimensions, and Types ontology. The yellow ellipse indicates a datatype.**

and ash fall on crop supply chains. The results may answer such competency questions as:

CQ1. What S2 cells are under a particular smoke plume, that have a certain crop type?

CQ2. What crop types are under a particular smoke plume?

Additionally, we provide some example queries that can be used to answer these questions.

## 3.2 Modularization

| Prefix | Name | URI |
|---|---|---|
| hcf: | This pattern | https://knowwheregraph.org/ontology/hcf/ |
| sosa: | SOSA/SSN | http://w3.org/ns/sosa/ |
| geo: | GeoSPARQL | http://opengis.net/ont/geosparql/ |
| qudt: | QUDT | http://qudt.org/schema/qudt/ |
| xsd: | XML Schema | http://www.w3.org/2001/XMLSchema# |

**Table 1: The namespaces used in our example instantiation of the Hierarchical Cell Features pattern shown in Figure 3.**

Figure 3 shows an example instantiation, or modularization, of the Hierarchical Cell Features pattern. To accomplish this, we have followed the process of template-based instantiation [7]. This means that the new concepts listed here have not been subclassed to the concepts in the pattern. Furthermore, where possible, we have utilized existing standards or ontologies for this module. In particular, we are using SOSA/SSN, the Sensors, Observations, Samples, Actuators Ontology [12] and the Semantic Sensor Network [6], respectively; GeoSPARQL, a standard for representing geospatial information from the Open Geospatial Consortium;[8] and QUDT[9], the Quantities, Units, Dimensions, and data Types ontology. The namespaces and the prefixes that we use for them can be found in Table 1.

The classes in the pattern map, as follows.

$$\begin{aligned} \text{Geometry} &\longrightarrow \text{geo:Geometry} \\ \text{Feature} &\longrightarrow \text{sosa:FeatureOfInterest} \\ \text{Attribute} &\longrightarrow \text{sosa:Observation} \\ \text{Quantity} &\longrightarrow \text{qudt:QuantityValue} \end{aligned}$$

Note also that there are two new nodes, xsd:double and qudt:Unit. These "come with" the qudt:QuantityValue node. For more details see the SSN standard.[10] The placeholder property, spatialRelations, expands to a subset of the RCC8 relations, as follows.

$$\begin{aligned} \text{spatiallyRelations} \longrightarrow & \text{geo:sfEquals} \\ & \text{geo:sfWithin} \\ & \text{geo:sfContains} \\ & \text{geo:sfTouches} \\ & \text{geo:sfOverlaps} \end{aligned}$$

The properties in the pattern map, as follows

$$\begin{aligned} \text{hasGeometry} &\longrightarrow \text{geo:hasGeometry} \\ \text{hasAttribute} &\longrightarrow \text{sosa:isFeatureOfInterestOf} \\ \text{hasValue} &\longrightarrow \text{sosa:hasResult} \end{aligned}$$

Finally, Cell and the associated properties, isAdjacentTo and contains remain in the pattern namespace, but these would be changed to the namespace of the knowledge graph.

## 3.3 Example Triples

The below is an example population of the pattern, using the modularization and namespaces in Section 3.2.

```
# Cells and Spatial Relations
kwgr:cell.L13.0 a hcf:Cell .
kwgr:cell.L14.0 a hcf:Cell .
kwgr:cell.L14.1 a hcf:Cell .
kwgr:cell.L14.2 a hcf:Cell .
kwgr:cell.L14.3 a hcf:Cell .

kwgr:cell.L13.0 hcf:contains kwgr:cell.L14.0 ,
 kwgr:cell.L14.1 ,
 kwgr:cell.L14.2 ,
 kwgr:cell.L14.3 .

kgr:cell.level14.0 hcf:isAdjacentTo kwgr:cell.L14.1 ,
    kwgr:cell.L14.2 .

# Crop Land
# Note:
# kwg-ont:CropLandType rdfs:subclassOf
# sosa:FeatureOfInterest
kwgr:geo.0 a geo:Geometry .
```

[8] https://www.ogc.org/standards/geosparql
[9] http://qudt.org/
[10] https://www.w3.org/TR/vocab-ssn/#quantity-values-and-unit-of-measures

```
kwgr:obs.0 a kwg-ont:TomatoObservation .
kwgr:qv.0 a qudt:QuantityValue ;
          qudt:numericValue "9.9e+03"^^xsd:double ;
          qudt:unit unit:M2 .
kwgr:obs.0 sosa:hasResult kwgr.qv.0 .
kwgr:foi.0 a kwg-ont:CropLandType ;
           geo:hasGeometry kwgr:geo.0 ;
           sosa:isFeatureOfInterestOf kwgr:obs.0 .

# Smoke Plume
# Note:
# kwg-ont:SmokePlume rdfs:subclass
# sosa:FeatureOfInterest
kwgr:geo.1 a geo:Geometry .
kwgr:obs.1 a kwg-ont:SmokePlumeExtentObservation .
kwgr:foi.1 a kwg-ont:SmokePlume ;
           geo:hasGeometry kwgr:1 ;
           sosa:isFeatureOfInterestOf kwgr:obs.1 .

kwgr:cell.L14.0 geo:sfWithin kwgr:geo.0 ;
                hcf:spatiallyRelated kwgr:foi.0 ,
                                     kwgr:foi.1 .
kwgr:cell.L13.0 hcf:spatiallyRelated kwgr:foi.0 .
```

The representative queries that can be used to closely generate this worked example are as follows.

The SPARQL query that can answer **CQ1**.

```
select  * as ?cell_wkt) where {
?cell a kwg-ont:KWGCellLevel14 ;
  rdfs:label ?label;
  sosa:isFeatureOfInterestOf ?cell_obs_collection ;
  geo:hasGeometry ?cell_geom .

?cell_geom geo:asWKT ?cell_wkt .

?cell_obs_collection sosa:hasMember ?cell_obs .

?cell_obs a ?cell_obs_type ;
  sosa:hasResult ?cell_obs_result .

?cell_obs_type rdfs:label ?cell_obs_type_label .

?cell_obs_result  qudt:value ?cell_obs_result_val ;
                  qudt:unit ?cell_obs_result_unit .

?cell_geom geo:sfWithin 'ex-geo'^^geo:wktLiteral .

FILTER(
 ?cell_obs_type_label = 'Tomatoes Observation' ||
 ?cell_obs_type_label =
   'Fallow/Idle Cropland Observation' ||
 ?cell_obs_type_label = 'Grapes Observation'
)
}
```

The SPARQL Query that can be used to answer **CQ2**.

```
SELECT * WHERE {}{
?cell a kwg-ont:KWGCellLevel14 ;
  sosa:isFeatureOfInterestOf ?cell_obs_collection ;
  rdfs:label ?cell_id;
  geo:hasGeometry ?cell_geom .

?cell_geom geo:asWKT ?cell_wkt .

?cell_obs_collection sosa:hasMember ?cell_obs .

?cell_obs a ?cell_obs_type ;
      sosa:hasResult ?cell_obs_result .

?cell_obs_type rdfs:label ?cell_obs_type_label .

?cell_obs_result  qudt:value ?cell_obs_result_val ;
          qudt:unit ?cell_obs_result_unit .

?cell_geom geo:sfWithin 'ex-geo'^^geo:wktLiteral .

FILTER(?cell_obs_type_label != 'Background Observation')
}
```

## 4 CONCLUSION

In this paper, we introduced an ontology design pattern for modeling features and regions and their interactions with an underlying discrete global grid, as well as provided an example instantiation and population of the pattern via our motivating scenario.

We believe that the use of a DGG for the purposes of integrating spatial data within a knowledge graph is a promising method, and furthermore believe that this pattern will help future knowledge graphs perform this integration easily and more quickly.

In fact, we are using S2 and the presented work heavily for the 600M triples that currently comprise the KnowWhereGraph. While we mostly looked at S2 cells here in our isolated example, it is interesting to note that these cells can also replace classical vector geometries in the sense that each polygon can be represented by a set of cells at different levels. As each cell has an unique IRI, we can directly associate other triples with it and thereby describe any region on earth together with direct inferences about the properties of its super- and subcells.

## Future Work

There is plenty of adjacent work to be accomplished with both this pattern and the surrounding paradigm. In particular, we wish to

(1) define a SHACL[11] shape for the pattern, to be used in validating the data mapped using this pattern;
(2) incorporate temporality, as it may not be the case that a feature *always* is spatially related to a Cell;
(3) incorporate the notion of an Event, which will greatly improve its applicability and usefulness, as well as immediately incorporate how persons may interact with the DGG; and, finally,
(4) connect this pattern to the Causal Events Pattern [23], so as to enrich the relationship between where something may occur and how it affects that place, using the DGG as a medium.
(5) provide a human-centered evaluation of the robustness of the pattern, as the pattern is currently used in prototype applications, the evaluation itself is still ongoing.

*Acknowledgements.* The authors acknowledge support by the National Science Foundation under Grant 2033521 A1: KnowWhereGraph: Enriching and Linking Cross-Domain Knowledge Graphs using Spatially-Explicit AI Technologies. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[11] https://www.w3.org/TR/shacl/

# REFERENCES


[1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. 2010. *The Description Logic Handbook: Theory, Implementation and Applications* (2nd ed.). Cambridge University Press, USA.

[2] Robert Battle and Dave Kolas. 2011. Geosparql: enabling a geospatial semantic web. *Semantic Web Journal* 3, 4 (2011), 355–370.

[3] Ben Bondaruk, Steven A Roberts, and Colin Robertson. 2020. Assessing the state of the art in Discrete Global Grid Systems: OGC criteria and present functionality. *Geomatica* 74, 1 (2020), 9–30.

[4] D Brizhinev, S Toyer, K Taylor, and Z Zhang. 2017. Publishing and using Earth observation data with the RDF data cube and the discrete global grid system. *W3C Working Group Note and OGC Discussion Paper W3C* 20170928 (2017), 16–125.

[5] Aaron Eberhart, Cogan Shimizu, Sulogna Chowdhury, Md. Kamruzzaman Sarker, and Pascal Hitzler. 2021. Expressibility of OWL Axioms with Patterns. In *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12731)*, Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Óscar Corcho, Petar Ristoski, and Mehwish Alam (Eds.). Springer, 230–245. https://doi.org/10.1007/978-3-030-77385-4_14

[6] Armin Haller, Krzysztof Janowicz, Simon JD Cox, Maxime Lefrançois, Kerry Taylor, Danh Le Phuoc, Joshua Lieberman, Raúl García-Castro, Rob Atkinson, and Claus Stadler. 2019. The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web* 10, 1 (2019), 9–32.

[7] Karl Hammar and Valentina Presutti. 2016. Template-Based Content ODP Instantiation. In *Advances in Ontology Design and Patterns [revised and extended versions of the papers presented at the 7th edition of the Workshop on Ontology and Semantic Web Patterns, WOP@ISWC 2016, Kobe, Japan, 18th October 2016] (Studies on the Semantic Web, Vol. 32)*, Karl Hammar, Pascal Hitzler, Adila Krisnadhi, Agnieszka Lawrynowicz, Andrea Giovanni Nuzzolese, and Monika Solanki (Eds.). IOS Press, 1–13. https://doi.org/10.3233/978-1-61499-826-6-1

[8] Quinn Hirt, Cogan Shimizu, and Pascal Hitzler. 2019. Extensions to the Ontology Design Pattern Representation Language. In *Proceedings of the 10th Workshop on Ontology Design and Patterns (WOP 2019) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 27, 2019 (CEUR Workshop Proceedings, Vol. 2459)*, Krzysztof Janowicz, Adila Alfa Krisnadhi, María Poveda-Villalón, Karl Hammar, and Cogan Shimizu (Eds.). CEUR-WS.org, 76–75. http://ceur-ws.org/Vol-2459/short2.pdf

[9] Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Alfa Krisnadhi, and Valentina Presutti. 2017. Towards a Simple but Useful Ontology Design Pattern Representation Language. In *Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017) co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. (CEUR Workshop Proceedings, Vol. 2043)*, Eva Blomqvist, Óscar Corcho, Matthew Horridge, David Carral, and Rinke Hoekstra (Eds.). CEUR-WS.org. http://ceur-ws.org/Vol-2043/paper-09.pdf

[10] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph (Eds.). 2012. *OWL 2 Web Ontology Language Primer (Second Edition.* W3C Recommendation 11 December 2012. Available from http://ww.w3.org/TR/owl2-primer/.

[11] Krzysztof Janowicz. 2021. KnowWhereGraph Drives Analytics and Cross-Domain Knowledge. *ArcUser* (2021), 16–19.

[12] Krzysztof Janowicz, Armin Haller, Simon JD Cox, Danh Le Phuoc, and Maxime Lefrançois. 2019. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* 56 (2019), 1–10.

[13] Krzysztof Janowicz, Pascal Hitzler, Wenwen Li, Dean Rehberger, Mark Schildhauer, Rui Zhu, Cogan Shimizu, Colby Fisher, Ling Cai, Gengchen Mai, Joseph Zalewski, Lu Zhou, Shirly Stephens, Seila Gonzalez, Anna Lopez Carr, Andrew Schroeder, Dave Smith, Dawn Wright, Sizhe Wang, Yuanyuan Tian, and Zilong Liu. 2021. now, Know Where, KnowWhereGraph: A Densely Connected, Cross-Domain Knowledge Graph and Geo-Enrichment Service Stack for Applications in Environmental Intelligence. *AI Magazine* (2021). Under Review.

[14] Peter Jonsson and Thomas Drakengren. 1997. A Complete Classification of Tractability in RCC-5. *J. Artif. Int. Res.* 6, 1 (June 1997), 211–221.

[15] Nazifa Karima, Karl Hammar, and Pascal Hitzler. 2017. How to Document Ontology Design Patterns. In *Advances in Ontology Design and Patterns*, Karl Hammar, Pascal Hitzler, Agnieszka Lawrynowicz, Adila Krisnadhi, Andrea Nuzzolese, and Monika Solanki (Eds.). Studies on the Semantic Web, Vol. 32. IOS Press, Amsterdam, 15–28.

[16] Maxime Lefrançois and Antoine Zimmermann. 2018. The Unified Code for Units of Measure in RDF: cdt: ucum and other UCUM Datatypes. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11155)*, Aldo Gangemi, Anna Lisa Gentile, Andrea Giovanni Nuzzolese, Sebastian Rudolph, Maria Maleshkova, Heiko Paulheim, Jeff Z. Pan, and Mehwish Alam (Eds.). Springer, 196–201. https://doi.org/10.1007/978-3-319-98192-5_37

[17] David A. Randell, Zhan Cui, and Anthony G. Cohn. 1992. A Spatial Logic based on Regions and Connection. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, USA, October 25-29, 1992*, Bernhard Nebel, Charles Rich, and William R. Swartout (Eds.). Morgan Kaufmann, 165–176.

[18] Satya Sahoo, Deborah McGuinness, and Timothy Lebo. 2013. *PROV-O: The PROV Ontology.* W3C Recommendation. W3C. http://www.w3.org/TR/2013/REC-prov-o-20130430/.

[19] Cogan Shimizu, Karl Hammar, and Pascal Hitzler. 2021. *Modular Ontology Modeling.* Technical Report. Kansas State University. https://daselab.cs.ksu.edu/publications/modular-ontology-modeling Under Review.

[20] Cogan Shimizu, Quinn Hirt, and Pascal Hitzler. 2018. A Protégé Plug-In for Annotating OWL Ontologies with OPLa. In *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11155)*, Aldo Gangemi, Anna Lisa Gentile, Andrea Giovanni Nuzzolese, Sebastian Rudolph, Maria Maleshkova, Heiko Paulheim, Jeff Z. Pan, and Mehwish Alam (Eds.). Springer, 23–27. https://doi.org/10.1007/978-3-319-98192-5_5

[21] Cogan Shimizu, Quinn Hirt, and Pascal Hitzler. 2019. MODL: A Modular Ontology Design Library. In *Proceedings of the 10th Workshop on Ontology Design and Patterns (WOP 2019) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 27, 2019 (CEUR Workshop Proceedings, Vol. 2459)*, Krzysztof Janowicz, Adila Alfa Krisnadhi, María Poveda-Villalón, Karl Hammar, and Cogan Shimizu (Eds.). CEUR-WS.org, 47–58. http://ceur-ws.org/Vol-2459/paper4.pdf

[22] Cogan Shimizu and Pascal Hitzler. In Press. Automatically Generating Human Readable Documentation for Ontology Design Patterns. In *20th International Semantic Web Conference Poster and Demos.*

[23] Cogan Shimizu, Rui Zhu, Mark Schildhauer, Krzysztof Janowicz, and Pascal Hitzler. In Press. A Pattern for Modeling Causal Relations Between Events. In *12th Workshop on Ontology Design and Patterns.*

[24] Christian Strobl. 2008. *Dimensionally Extended Nine-Intersection Model (DE-9IM).* 240–245. https://doi.org/10.1007/978-0-387-35973-1_298