SOGO: A Scaling Ontology for Geospatial Observations in Knowledge Graphs

Amin Anjomshoaa

Vienna University of Economics and Business, Vienna, Austrian

Abstract

Ontologies and knowledge graphs have emerged as effective tools for data integration and enabling information interoperability, particularly in the context of spatiotemporal observations. In this setting, data aggregation and disaggregation play a crucial role in multilevel modeling, facilitating the transfer and preservation of information across different spatial and temporal scales. While existing methods outside the domain of knowledge graphs address aspects of data scaling, there is currently no standardized or widely accepted approach for modeling and implementing such mechanisms using ontologies and knowledge graphs. This gap highlights the need for a high-level and lightweight ontology capable of describing scalable observations based on their spatiotemporal properties. In this work, we articulate the necessity of such an ontology and introduce a work-in-progress scaling ontology that defines the core classes and properties required to represent scaling processes in spatial knowledge graphs. As a proof of concept, we demonstrate the application of the proposed ontology in upscaling and downscaling scenarios, highlighting its application in domains involving observations and sensor data.

Keywords

Spatial Knowledge Graph, Ontology, Data Observation, Spatial Scaling

1. Introduction

Scale is a foundational concept in the geographical sciences and an inherent property of many spatial systems [1]. It has been extensively employed in quantitative research across geographic, ecological, and statistical domains [2], where it plays a critical role in spatial data analysis. In these contexts, spatial scaling refers to how the distribution of individuals, resources, or phenomena varies across different spatial extents or resolutions. The complexity of the scale concept has led to several definitions and categorizations, reflecting its diverse applications and interpretations across disciplines [2]. In spatial data analysis specifically, scale directly influences both how data is interpreted and the conclusions drawn from analytical processes.

At its core, spatial scaling is shaped by two interrelated factors: how we look at spatial data (i.e., the scale of observation) and how the phenomena are spatially arranged (i.e., the distribution). These two aspects are crucial for accurate modeling and meaningful interpretation. A phenomenon that is clearly observable at one scale may become invisible, distorted, or irrelevant when examined at another scale. This scale-dependent behavior introduces challenges in spatial modeling, especially when attempting to generalize results across different spatial resolutions.

One well-known challenge associated with spatial scaling is the Modifiable Areal Unit Problem (MAUP) which is a statistical bias that arises when spatial data is aggregated into different areal units [3]. MAUP demonstrates how changes in the scale or zoning scheme of spatial units can significantly influence the outcomes of statistical analyses and underscores the importance of careful consideration in how spatial units are defined and used in data modeling approaches.

The process of merging or aggregating properties across distinct spatial regions is therefore non-trivial. Various methods exist to address this challenge, with data aggregation and disaggregation being

SPARK 2026: First International Workshop on Spatial Intelligence and Reasoning Enabled by Knowledge Graphs and Foundation Models, 25-26 October 2026, Bari, Italy, co-located with 25th International Semantic Web Conference (ISWC 2026)

Corresponding author.

amin.anjomshoaa@wu.ac.at (A. Anjomshoaa)

https://anjomshoaa.github.io/ (A. Anjomshoaa)

0000-0001-6277-742X (A. Anjomshoaa)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



published 2026-11-05

CEUR-WS.org/Vol-4313/SPARK-paper8.pdf

the most common ones. For instance, aggregation involves combining data from finer-scale spatial units into larger, coarser units to produce a new representation. This may include techniques such as simple averaging, summing values, or applying weighted averages to derive a representative value for the aggregate region. While these methods are widely used in statistical and spatial analysis to generate summary statistics for spatial data analysis, there remains a lack of standardized, widely adopted approaches for representing and implementing them using ontologies and knowledge graphs.

Furthermore, the choice of spatial scale can critically influence the accuracy of modeled property distributions. Poorly chosen or improperly applied scales can lead to misrepresentation or invalid inferences. For example, while population density may be accurately calculated at a fine scale (e.g., urban block level), its broader spatial distribution might appear uneven or misleading when aggregated to regional or national levels. Given these complexities, we advocate for allowing data owners and data publishers, who possess deep contextual knowledge, to explicitly encode scale and aggregation information within their knowledge graphs, alongside the data itself. By doing so, we can enhance the interpretability, accuracy, and reusability of spatial data in spatial knowledge graphs.

In this paper, we first review existing work on spatial knowledge graphs and then discuss common aggregation and disaggregation methods for spatiotemporal observation data. We subsequently introduce SOGO (A Scaling Ontology for Geospatial Observations in Knowledge Graphs), a core and lightweight ontology that captures spatial scaling logic and makes it a first-class citizen of knowledge graphs. By explicitly representing scaling operations, particularly their semantic aspects, SOGO accounts for the nuances of spatial structure and distribution and enables scaling logic to be published alongside the data itself. Finally, as a proof of concept, we present two real-world upscaling and downscaling scenarios using a knowledge graph and demonstrate how SOGO can effectively capture and facilitate spatial scaling operations in knowledge graphs.

2. State of the Art

To represent spatial data in RDF, several existing works and standard vocabularies provide mechanisms for describing the multidimensional structure and spatial characteristics of datasets. Among these, the RDF Data Cube Vocabulary (QB) [4] provides a general framework for representing multidimensional data in RDF. Although originally designed for statistical data, its model is sufficiently general to support other types of multidimensional datasets, including sensor, environmental, Earth-observation, and OLAP data. QB is based on concepts from the multidimensional information model of SDMX (Statistical Data and Metadata eXchange) [5], an international standard for the exchange and dissemination of statistical data and metadata.

A specialized geospatial extension of QB is the RDF Data Cube Vocabulary for Spatio-Temporal Components (QB4ST) [6], which extends the Data Cube model to explicitly represent spatial and temporal dimensions in multidimensional datasets. QB4ST provides mechanisms for describing spatial reference systems, spatial components, coordinate granularity, and other spatial metadata, thereby supporting the semantic representation of datasets at different spatial resolutions.

However, while QB and QB4ST provide the metadata and semantic foundations for describing multidimensional observations and their spatial characteristics at different resolutions, they do not define explicit mechanisms for representing or executing the transformation of observations between spatial resolutions or spatial hierarchies. In particular, they do not specify the semantics of scaling operations, such as upscaling and downscaling, or the associated resampling, aggregation, or disaggregation methods used to derive observations at a target spatial resolution.

A widely adopted approach for capturing spatiotemporal phenomena in knowledge graphs is the use of the SOSA ontology [7], a lightweight, modular framework designed to model observations, sampling, and actuation activities involving sensors and systems that collect or interact with data. SOSA facilitates consistent and interoperable descriptions of observational data across diverse domains such as environmental monitoring, smart cities, and the Internet of Things (IoT). It models observations—such as those from sensors—by linking key elements, including the observed property, the procedure used,

the result, and the time and location of the observation. In SOSA, each observation is represented as an event or activity that yields a result through a defined method.

To spatially organize observations in a scalable and neutral way, without regard for political boundaries, grid-based modeling is often employed in knowledge graphs. This method partitions the Earth's surface into spatial grid cells, allowing observations to be assigned to standardized spatial units. These grid cells can then be associated with overlapping administrative or geographic regions such as communities, cities, and states.

One such grid system is Google's S2 Geometry library [8], which provides tools for representing, indexing, and computing geographic data on the surface of a sphere. S2 divides the Earth into hierarchical cells, where each level offers progressively finer spatial resolution. In the context of knowledge graphs, S2 has been leveraged for efficient multi-source data integration, qualitative spatial querying, and cross-graph reasoning [9].

Another popular spatial indexing system is Uber's H3 [10], which partitions the Earth's surface into hexagonal cells using a hierarchical hexagons. Each hexagon can be recursively subdivided into smaller hexagons, enabling multi-resolution analysis. H3's hexagonal grid structure is particularly effective for applications such as spatial clustering, proximity queries, and geospatial analytics due to its uniform adjacency properties and consistent area coverage.

In spatial data analysis, scale concept has been extensively studied within the fields of remote sensing and geographic information systems (GIS) [11]. Existing conceptualizations of spatial scale have led to several widely recognized categories, including cartographic scale, geographic scale, operational scale, and observational scale [12].

Alongside traditional statistical scaling approaches, machine learning (ML) methods have been increasingly applied for downscaling and gap-filling in remote sensing applications [13]. These methods have shown promising results in producing high-quality remote sensing results. Compared to conventional statistical techniques, ML approaches offer several advantages: they do not require strict statistical assumptions, can efficiently model complex and non-linear relationships, and often yield higher precision in capturing spatial variability [14, 15].

Despite this rich body of work, the semantic representation of observation scaling within knowledge graphs remains relatively underexplored. To the best of our knowledge, no established ontology or standard vocabulary provides a comprehensive semantic representation of spatiotemporal data-resolution transformations that explicitly captures the possible transformation mechanisms and methods, such as aggregation and disaggregation, their associated parameters, and their applicability, while linking these transformation semantics to the underlying spatial observations. In this research, we aim to shed light on this gap by investigating how spatial scale can be explicitly represented and leveraged in knowledge graphs, and exploring its potential applications in the development and analysis of spatial knowledge graphs.

3. Spatial Scaling of Observations

In the context of quantitative indicators assigned to spatial entities, such as geometric features (e.g., political boundaries) and grid cells (e.g., Google's S2 or Uber's H3 cells) at different resolutions, it is often necessary to upscale or downscale these indicators to match the target area in which an event occurs or an analysis is conducted. This process ensures that the indicators are adjusted to accurately reflect the characteristics of the specific area under consideration.

Suppose R is the region under study composed of sub-regions R_i such that $R = \bigcup_{i=1}^{n} R_i$. For well-defined grid systems, such as S2 and H3, hierarchical levels are inherently established by their underlying algorithms. For political or geographical regions such as municipalities and states, hierarchical levels and associated data can instead be derived from available data sources and structured accordingly.

When aggregating or disaggregating values across spatial sub-regions, there is no single universal formula; the appropriate scaling approach depends primarily on the nature of the property being represented. Furthermore, all indicators should be expressed on a consistent scale and in the same units

to ensure meaningful comparison. The temporal extent and reference period of each data point should also be clearly defined to avoid inaccurate or misleading aggregation or disaggregation.

3.1. Upscaling

In upscaling use cases, data from smaller grid cells are combined into larger units to create a new spatial representation. This can be done by averaging values, taking the sum, or using weighted averages to create a new value for the larger unit. Such methods are commonly used in statistical analysis to create summary statistics for larger regions.

When aggregating indicators across spatial sub-regions, the appropriate aggregation method depends on whether the indicator represents an intensive property or an extensive quantity. Intensive properties, which do not depend on the size of the area, are generally aggregated using averages or other measures of central tendency. In contrast, extensive quantities scale with the size of the region and are therefore typically aggregated by summation. For example, consider population as a property of smaller spatial cells. When population is expressed as a density (e.g., people/km²), it is treated as an intensive property. If two grid cells have population densities of 100 and 200 people/km², the combined region does not have a density of 300 people/km². Instead, its population density would typically be calculated as an area-weighted average of the two cells. In contrast, when population is expressed as the number of people within a spatial region, it represents an extensive quantity and can therefore be summed across spatial units.

Thus, for an extensive property (ρ), such as population, the aggregated value for a larger region (R) is commonly defined as the sum of the corresponding indicator values across its constituent sub-regions:

$$\rho(R) = \sum_{i=1}^{n} \rho(R_i) \tag{1}$$

Additionally, depending on the semantics of the target property—for example, the maximum or minimum temperature across smaller regions—the value for the larger region R may be determined by the corresponding extreme value among its constituent sub-regions. Specifically, the maximum or minimum value for R is equal to the respective maximum or minimum indicator value observed in one of its sub-regions:

$$\rho(R) = \rho(R_i) \quad \text{for some } i \in \{1, \dots, n\} \tag{2}$$

Another important type of aggregation involves weighted averages which is commonly applied to intensive properties. In this case, the property is expressed as a ratio or quotient of two quantities. For example, an indicator such as population density (people/km²) does not represent the total number of people in a region; rather, it describes the population relative to the region's area. To aggregate such an indicator to a higher-level region, the corresponding denominator (in this case area) must be taken into account.

Accordingly, the values of the sub-regions should be weighted so that each contributes proportionally to the aggregated indicator. In many cases, the appropriate weights are determined by a quantity related to the indicator. For example, a mortality rate is typically weighted by population, whereas precipitation expressed as a depth (e.g., liters/m²) is generally weighted by area when aggregating across spatial regions. Thus, to calculate the aggregated indicator for a larger region A, the values of its constituent sub-regions can be combined using a normalized weight function ω, where the weights are derived from the relevant population or area values. When the weights are defined based on the areas of the respective regions, this approach is referred to as area-weighted averaging. More generally, weighted averaging can be expressed using the following formula:

$$\rho(R) = \frac{\sum_{i=1}^{n} \rho(R_i) * \omega(R_i)}{\sum_{i=1}^{n} \omega(R_i)} \tag{3}$$

3.2. Downscaling

In the case of downscaling, properties from higher-level regions are disaggregated to estimate values for smaller, lower-level regions. Downscaling is generally more challenging than upscaling because moving from a coarse spatial resolution to a finer one typically requires additional information or assumptions about how the property is distributed within the larger region. Without such information, the fine-scale values cannot be uniquely determined from the aggregated value alone. Therefore, domain knowledge from data owners, who often have a deeper understanding of the underlying data and its spatial characteristics, can be particularly valuable in selecting an appropriate downscaling method and defining the assumptions required for disaggregation.

Depending on the data type and structure, the required level of disaggregation, and the analysis objectives, different methods can be applied. Common approaches to achieve this include:

- statistical estimation which involves using statistical techniques to estimate values for smaller sub-regions based on data from larger aggregated areas. This can be done by using interpolation or regression techniques to estimate values for each sub-region.
- weighted downscaling which involves dividing an aggregated value among sub-regions based on a higher-resolution auxiliary information. For example, if the population of a city is known, the population of each district can be estimated based on its proportion of the total area or density of street networks.
- spatial clustering which involves grouping areas based on similarities in their properties. This can be done using statistical techniques such as k-means clustering or hierarchical clustering.

4. SOGO Data Model

SOGO (A Scaling Ontology for Geospatial Observations) models spatial-resolution transformations as explicit semantic entities within spatial knowledge graphs. It is designed as a lightweight and generic ontology for representing the semantics of spatial scaling, including the relationships between source and target spatial features, scaling properties, and the methods used to transform observations across spatial resolutions. Building on the principles of SOSA for modeling observations and their association with spatial entities such as grid cells and regions, SOGO defines the necessary classes and properties to formally represent both downscaling and upscaling processes. The SOGO ontology is organized around three main pillars:

- **Spatial Map** defines the mechanism for traversing between higher- and lower-level regions, both of which are modeled as subclasses of geo:Feature, as defined by the OGC GeoSPARQL standard for representing and querying geospatial information in RDF/SPARQL [16]. Because hierarchical region structures and the mappings between their levels can be represented in various ways, SOGO adopts a flexible approach by implementing spatial mappings as SPARQL queries associated with the Spatial Map entity. This allows the ontology to accommodate different hierarchical models and data representations while providing a uniform mechanism for navigating relationships between spatial regions. Furthermore, spatial mappings can be reused across different scaling procedures, reducing redundancy and promoting consistency in the representation of spatial relationships.
- **Scaling Method** includes specific upscaling and downscaling methods, which may be implemented either as native SPARQL functions or as external functions. These methods are represented as instances of the **sogo:Upscaling** or **sogo:Downscaling** classes and can be extended as needed to accommodate additional scaling strategies.
- **Scaling procedures** form the core element of SOGO, linking a spatial mapping to a specific scaling method. They are modeled by extending the **sosa:Procedure** class and are designed to transform a given **sosa:ScalingProperty**, which is associated with spatial observations, into a scaled property value. A scaling procedure may optionally specify one or more auxiliary properties to support

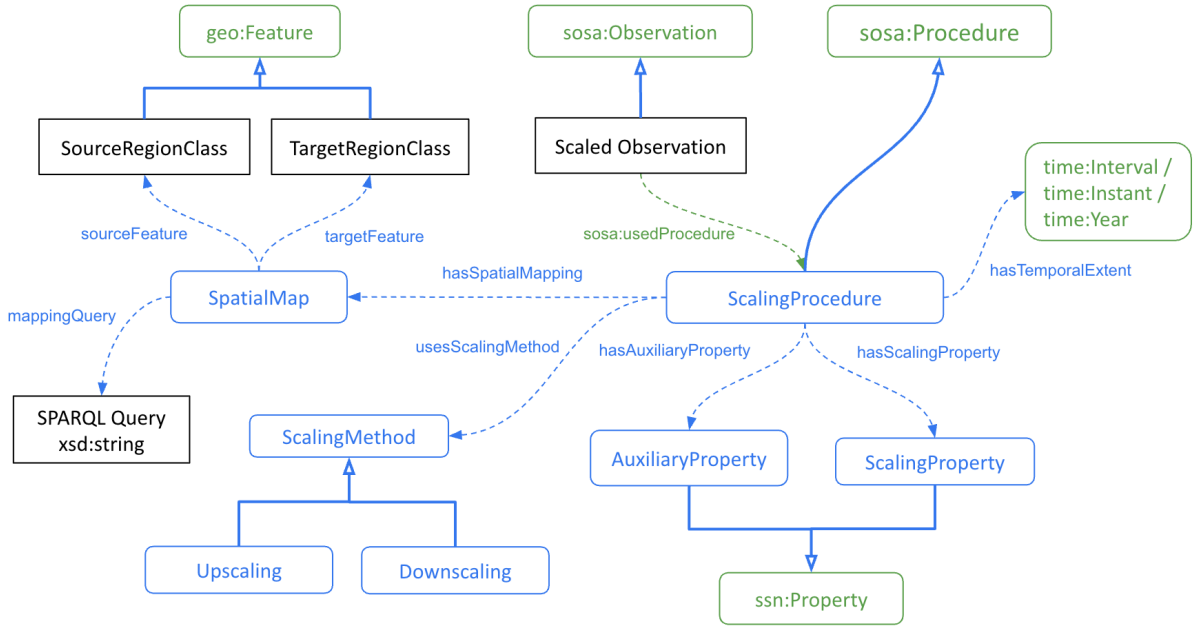


Figure 1: Overview of SOGO classes and properties.

weighted scaling use cases. It also defines the allowed temporal extent values during which the aggregation or disaggregation is applied, thereby restricting the observations considered when deriving the scaled property value. The temporal extent may be specified as a time interval or as a specific point in time, such as a particular timestamp or year.

The formal definitions of these core concepts are presented in Table 1, while Figure 1 illustrates the core components of SOGO and their relationships.

Table 1
Formal definitions of the core SOGO classes.

Class	Definition
sogo : ScalingProcedure	sosa : Procedure $\sqcap$ (= 1 hasScalingProperty.ScalingProperty) $\sqcap$ ($\geq$ 0 hasAuxiliaryProperty.AuxiliaryProperty) $\sqcap$ (= 1 usesScalingMethod.ScalingMethod) $\sqcap$ (= 1hasSpatialMap.SpatialMap) $\sqcap$ ($\geq$ 1 hasTemporalExtent.(time : Instant $\sqcup$ time : Interval $\sqcup$ time : Year))
sogo : SpatialMap	(= 1 sourceFeature.geo : Feature) $\sqcap$ (= 1 targetFeature.geo : Feature) $\sqcap$ (= 1 mappingQuery.xsd : string)
sogo : ScalingMethod	UpscalingMethod $\sqcup$ DownscalingMethod UpscalingMethod $\sqcap$ DownscalingMethod $\sqsubseteq$ $\bot$

Based on this data model, scaling use cases can be implemented directly using the information stored alongside the data in the knowledge graph. To illustrate this process, suppose we are looking for the value of a specific property $\rho(R)$ for a given temporal extent, but no explicit observation of this property exists for region R in the knowledge graph. In this example, the region R is defined as a **geo:Feature** and belongs to a specific spatial class (e.g., State or Municipaality). A representative procedure for deriving the missing observation consists of the following steps:

- **Step 1:** Identify the defined **Scaling Procedure** instances that specify the given property as their scaling property.

- **Step 2:** Among the matching Scaling Procedures, identify those that reference a **Spatial Map** for which the spatial class of given region *R* is defined as the target feature.
- **Step 3:** Using the selected Spatial Map, retrieve the set of source regions involved in the scaling process, together with the required properties for the calculation, including the scaling property and, where applicable, the auxiliary property. The observations are selected according to the temporal extent and grouping specified by the Scaling Procedure.
- **Step 4:** Apply the scaling method specified by the Scaling Procedure to the retrieved property values and calculate the corresponding scaled value.
- **Step 5:** Use the resulting value to construct a synthetic observation for region *R*. The observation is associated with the corresponding SOGO-defined Scaling Procedure through **sosa:usedProcedure**, thereby preserving the provenance of the derived value.

Finally, Table 2 summarizes the classes defined in SOGO, while Table 3 presents the corresponding properties and their semantics.

Table 2
SOGO ontology classes and their descriptions.

Class	Definition	Superclass
`ScalingProcedure`	A procedure that transforms observations or properties from one spatial resolution to another.	`Procedure`
`ScalingMethod`	A method specifying how a scaling transformation is performed, with disjoint subclasses representing upscaling and downscaling methods.	–
`UpscalingMethod`	A class that defines aggregation-based scaling methods for deriving observations at a coarser spatial resolution from observations at a finer resolution.	`ScalingMethod`
`DownscalingMethod`	A class that defines disaggregation-based scaling methods for deriving observations at a finer spatial resolution from observations at a coarser resolution.	`ScalingMethod`
`ScalingProperty`	The primary observed property whose values are transformed by a scaling procedure.	`sosa:Property`
`AuxiliaryProperty`	An auxiliary observed property that may be required by specific scaling methods and used as a reference or predictor during a scaling procedure.	`sosa:Property`
`SpatialMap`	A specification of the spatial correspondence between source and target geographic features involved in a scaling procedure.	–

5. Evaluation Use Cases

To provide a concrete example and demonstrate the feasibility of our approach, we present one upscaling and one downscaling use case involving population data. Both use cases are applied to a knowledge graph containing population and street network observations for regions in Austria. The knowledge graph also includes a hierarchy of interrelated spatial regions, comprising communities, districts, states, and their corresponding underlying H3 cells. Listing 1 represents the scaling properties and two sample observations.

Table 3
SOGO object properties and their semantics.

Property	Domain	Range	Card.	Definition
hasScalingProperty	ScalingProcedure	ScalingProperty	1..1	Associates a scaling procedure with the primary property being scaled.
hasAuxiliaryProperty	ScalingProcedure	AuxiliaryProperty	0..*	Associates a scaling procedure with auxiliary properties used to support the scaling procedure.
usesScalingMethod	ScalingProcedure	ScalingMethod	1..1	Specifies the scaling method employed by a procedure.
hasSpatialMap	ScalingProcedure	SpatialMap	1..1	Associates a scaling procedure with the spatial mapping used to establish source–target correspondence.
hasTemporalExtent	ScalingProcedure	time:Instant, time:Interval, time:Year	0..*	Specifies the allowed temporal extent values during which the scaling procedure can be applied.
sourceFeature	SpatialMap	geo:Feature	1..1	Specifies the source geographic feature from which values are scaled.
targetFeature	SpatialMap	geo:Feature	1..1	Specifies the target geographic feature for which the scaled values are generated.
mappingQuery	SpatialMap	xsd:string	1..1	Defines a SPARQL query that specifies the spatial mapping between the source and target regions.

Listing 1: Scaling properties and sample observations

```
:population rdf:type ssn:Property;
        rdfs:range xsd:integer ;
        rdfs:label "population"@en ;
        skos:prefLabel "Population of given area"@en ;
        rdfs:comment "Represents the total population of a given area."@en .

:roadLength rdf:type ssn:Property;
        rdfs:domain :H3Cell;
        rdfs:range xsd:int ;
        rdfs:label "roadLeangth"@en ;
        skos:prefLabel "Total length of roads in grid cells"@en ;
        rdfs:comment "Represents the total length, in meters, of all roads within a given cell."@en .

<id/observation/881e064c83fffff/roads> a sosa:Observation ;
        sosa:resultTime "2011-01-01T00:00:00Z"^^xsd:dateTime ;
        sosa:hasFeatureOfInterest <id/cell/881e064c83fffff> ;
        sosa:observedProperty :roadLength;
        sosa:hasSimpleResult "10807"^^xsd:integer .

<id/observation/92101/POP_2011> a sosa:Observation ;
        sosa:resultTime "2011-10-31T00:00:00Z"^^xsd:dateTime ;
        sosa:hasFeatureOfInterest <id/community/92101> ;
        sosa:observedProperty :population;
        sosa:hasSimpleResult "143962"^^xsd:integer .
```

As the first use case, we consider an upscaling scenario in which the population of a state is derived from the aggregated population of its municipalities and communities. In our dataset, the political hierarchy is structured such that communities (municipalities) are grouped into districts, and districts collectively form a state. This spatial mapping between communities (source geo:Feature) and their corresponding state (target geo:Feature) is represented in Listing 2.

Listing 2: Examlpe of spatial mapping between communities and their corresponding state

```
crisp:CommunityStateMapping rdf:type owl:NamedIndividual, sogo:SpatialMap ;
        sogo:sourceFeature crisp:Community ;
        sogo:targetFeature crisp:State ;
        sogo:mappingQuery """
            PREFIX geo: <http://www.opengis.net/ont/geosparql#>

            SELECT * WHERE {
                ?parts geo:sfWithin/geo:sfWithin {SOGO_WHOLE} . } """ .
```

In our upscaling example, the knowledge graph contains population observations for individual communities, but the population of the corresponding states is not explicitly represented. To derive population observations for the states, we define the Scaling Procedure shown in Listing 3

Listing 3: Scaling Procedure for generating state populations

```
crisp:populationUpscaling rdf:type owl:NamedIndividual, sogo:ScalingProcedure ;
    sogo:hasScalingProperty crisp:population ;
    sogo:hasSpatialMap crisp:CommunityStateMapping ;
    sogo:hasTemporalExtent [ rdf:type time:Year ; time:year "2011"^^xsd:gYear] ;
    sogo:usesScalingMethod sogo:SumAggregation ;
    rdfs:comment "A scaling procedure for calculating population of states based on community population data."
     @en ;
    rdfs:label "Population Upscaling Procedure"@en .
```

After applying the scaling procedure the state population observation will be generated as shown in Listing 4. Notably, the resulting scaled observations include an additional property, **sosa:usedProcedure**, which indicates that the observations were generated through a scaling process. This property also distinguishes explicitly observed population data from population data derived using the scaling procedure.

Listing 4: Generated population observation for State of Burgenland

```
<id/observation/state/1/population> a sosa:Observation ;
       sosa:resultTime "2011-10-31T00:00:00Z"^^xsd:dateTime ;
       sosa:hasFeatureOfInterest <id/state/1> ;
       sosa:observedProperty :population;
       sosa:hasSimpleResult "285685"^^xsd:integer ;
   sosa:usedProcedure sogo:populationUpscaling .
```

As the second use case we demonstrate a downscaling example where we have the population o a municipality and municipality is overlappping with a colllection of H3 Cells. The task is to estimate the population in these H3 Cellls based on the population of the upper level region which in this case is the Municipality. Since the population is not distribute uniformly in the municipality, we would need an auxiliary property to estimate the weighted cell populations according to that. In this example we assume the length of road network in each cell indicates more people and residential areas in a specific cell. As such the Scaling Procedure defined in Listing 5 includes an axiom about length of roads in each cell using **sogo:hasAuxiliaryProperty**.

As the second use case, we demonstrate a downscaling example in which the population of a municipality is known, and the municipality overlaps with a collection of H3 cells. The objective is to estimate the population of each H3 cell based on the population of the upper-level region. Since the population is not uniformly distributed across the municipality, an auxiliary property is required to estimate the population of each cell in a weighted manner. In this example, we assume that the length of the road network within each cell serves as an indicator of population density, with cells containing longer road networks assumed to have more residential areas and, consequently, larger populations. Accordingly, the Scaling Procedure defined in Listing 5 includes an axiom that uses the length of the road network in each cell as an auxiliary property through **sogo:hasAuxiliaryProperty** property. Furthermore, this scaling procedure uses the spatial map **H3CellCommunityMapping** to establish the mapping between H3 cells and their corresponding upper-level region, while also using a specific timestamp to filter the relevant population observations.

Listing 5: Scaling Procedure for generating H3 Cell populations

```
crisp:populationDownscaling rdf:type owl:NamedIndividual, sogo:ScalingProcedure ;
   sogo:hasAuxiliaryProperty crisp:roadLength ;
   sogo:hasScalingProperty crisp:population ;
   sogo:hasSpatialMap crisp:H3CellCommunityMapping ;
   sogo:hasTemporalExtent [ rdf:type time:Instant ; time:inXSDDateTimeStamp "2011-10-31T00:00:00Z"^^xsd:
     dateTimeStamp] ;
   sogo:usesScalingMethod sogo:WeightedDisaggregation ;
   rdfs:comment "A scaling procedure for estimating H3 cell populations from community-level populations,
     weighted proportionally by the length of roads within each cell."@en ;
   rdfs:label "Population Downscaling Procedure"@en .
```

Based on the definition provided in the scaling procedure, the population observation for each H3 cell is calculated using a weighted disaggregation scaling method, defined through an instance of **sogo:WeightedDisaggregation**. The resulting population observations for the H3 cells are then generated as shown in Listing 6.

Listing 6: Generated population observation for H3 Cells

```
<id/observation/881e064c83fffff/population> a sosa:Observation ;
       sosa:resultTime "2011-10-31T00:00:00Z"^^xsd:dateTime ;
       sosa:hasFeatureOfInterest <id/cell/881e064c83fffff> ;
       sosa:observedProperty :population;
       sosa:hasSimpleResult "857"^^xsd:integer ;
   sosa:usedProcedure sogo:populationDownscaling .
```

The implementation of the scaling functions for both upscaling and downscaling procedures can be integrated into the backend system to automate the generation of the required observations on demand based on the SOGO specifications. This approach can abstract the underlying scaling steps from end users. As a proof of concept, a simple Python implementation of the scaling functions for the use cases described in this paper is provided in the SOGO GitHub Repository.

6. Conclusions and Future Work

Scaling is a common yet intricate task in spatial data analysis. It demands not only technical methods but also contextual understanding of the underlying data, including its distribution, limitations, and potential biases. We argue that data owners are often best positioned to provide this insight. In this work, we introduced a work-in-progress scaling ontology that builds on established best practices for modeling spatial observations, such as those found in SOSA, and extends them with the ability to formally represent scaling procedures.

As a proof of concept, we demonstrated how the proposed ontology can be applied to population upscaling and downscaling scenarios involving different types of spatial regions, including administrative boundaries and grid cells. These examples illustrate how spatial knowledge graphs can benefit from semantically rich, ontology-driven scaling processes.

In future work, we plan to extend our approach to more complex scaling methods, including interpolation and extrapolation, which are widely used in spatial data analysis. Furthermore, we plan to develop an initial generic Python library for SOGO to automate scaling tasks and facilitate the integration and implementation of scaling functionality in spatial knowledge graphs.

Declaration on Generative AI

Following the CEUR GenAI Usage Taxonomy, ChatGPT was used for ”grammar and spelling check” and ”improving writing style”. Author has reviewed all AI outputs and take full responsibility for the publication.

References

[1] N. Tate, P. M. Atkinson, Modelling scale in geographical information science, John Wiley & Sons, 2001.

[2] T. M. Oshan, L. J. Wolf, M. Sachdeva, S. Bardin, A. S. Fotheringham, A scoping review on the multiplicity of scale in spatial analysis, Journal of Geographical Systems 24 (2022) 293–324.

[3] D. W. Wong, The modifiable areal unit problem (maup), in: WorldMinds: geographical perspectives on 100 problems: commemorating the 100th anniversary of the association of American geographers 1904–2004, Springer, 2004, pp. 571–575.

[4] J. Tennison, R. Cyganiak, D. Reynolds, The RDF Data Cube Vocabulary, Technical Report, Technical report, W3C Working Draft 05 April, 2012., 2012.

[5] R. Stahl, P. Staab, History of sdmx, in: Measuring the Data Universe: Data Integration Using Statistical Data and Metadata Exchange, Springer, 2018, pp. 73–83.

[6] R. Atkinson, QB4ST: RDF Data Cube Extensions for Spatio-temporal Components, https://www.w3.org/TR/qb4st/, 2017. W3C Working Group Note.

[7] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, M. Lefrançois, Sosa: A lightweight ontology for sensors, observations, samples, and actuators, Journal of Web Semantics 56 (2019) 1–10.

[8] Google, S2 geometry, 2017. URL: http://s2geometry.io/, accessed: 2025-08-05.

[9] S. Stephen, M. Faulk, K. Janowicz, C. Fisher, T. Thelen, R. Zhu, P. Hitzler, C. Shimizu, K. Currier, M. Schildhauer, et al., The s2 hierarchical discrete global grid as a nexus for data representation, integration, and querying across geospatial knowledge graphs, arXiv preprint arXiv:2410.14808 (2024).

[10] I. Brodsky, H3: Uber’s hexagonal hierarchical spatial index., 2018. URL: https://www.uber.com/en-AT/blog/h3/, accessed: 2025-08-05.

[11] D. A. Quattrochi, M. F. Goodchild, Scale in remote sensing and GIS, CRC press, 1997.

[12] Z. Dabiri, T. Blaschke, Scale matters: a survey of the concepts of scale used in spatial disciplines, European Journal of Remote Sensing 52 (2019) 419–434.

[13] Z. Tang, Gap-filling using machine learning: implementations and applications in remote sensing., Ph.D. thesis, University of Helsinki, Finland, 2022.
[14] D. Bzdok, N. Altman, M. Krzywinski, Statistics versus machine learning, Nature methods 15 (2018) 233.
[15] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and machine learning forecasting methods: Concerns and ways forward, PloS one 13 (2018) e0194889.
[16] R. Battle, D. Kolas, Geosparql: enabling a geospatial semantic web, Semantic Web Journal 3 (2011) 355–370.

A. Online Resources

The source code for the SOGO ontology and its proof-of-concept implementation is available in the project's GitHub Repository.