G. Torta, L. Ardissono, D. Fea, L. La Riccia, A. Voghera A Semantic Approach to Constraint-Based Reasoning in Geographical Domains. In: Fred A. et al. (eds) Knowledge Discovery, Knowledge Engineering and Knowledge Management. IC3K 2018, pp. 202-227. ISBN: 978-3-030-49558-9, DOI: 10.1007/978-3-030-49559-6_10, Springer Verlag, 2020. The Editorial Version of this paper is published in https://link.springer.com/chapter/10.1007/978-3-030-49559-6_10.

A Semantic Approach to Constraint-based Reasoning in Geographical Domains*

Gianluca Torta[1][0000−0002−4276−7213], Liliana Ardissono[1][0000−0002−1339−4243], Daniele Fea[1], Luigi La Riccia[2][0000−0002−4800−2641], and Angioletta Voghera[2][0000−0002−0166−3303]

[1] Dipartimento di Informatica, Università di Torino, Italy,
gianluca.torta,liliana.ardissono@unito.it daniele.fea@edu.unito.it
www.unito.it
[2] Dipartimento Interateneo di Scienze, Progetto e Politiche del Territorio, Torino, Italy,
luigi.lariccia,angioletta.voghera@polito.it
www.polito.it

Abstract. Various models have been developed to manage geographic data but most of them integrate heterogeneous techniques to support knowledge representation and reasoning. This is far from optimal because it requires mapping data between different representation formats; moreover, as it fragments knowledge, it limits the possibility to use complete information about the problem to be solved for the execution of inferences.
In order to address this issue, we adopt a unified approach, in which we use Semantic Web techniques to manage both knowledge representation and reasoning rules with particular attention to constraint verification that is central to several geographic reasoning tasks. Our model exploits an ontological description of spatial constraints which supports the specification of their properties, facilitating the automated selection of the relevant ones to be applied to a given problem. The model supports different types of inferences, such as checking the compliance of a given geographical area to a set of constraints, or suggesting a suitable aggregation of land patches that satisfy them.
We test our model by applying it to the management of Ecological Networks, which describe the structure of existing real ecosystems and help planning their expansion, conservation and improvement by introducing constraints on land use.

Keywords: Geographic Knowledge · Geographical Constraints · GeoSPARQL, Ecological Networks · Urban Planning.

1 Introduction

With the convergence of GIS and Semantic Web, various models have been developed to manage geographic information; however, most of them integrate different techniques to support knowledge representation and reasoning. For instance, they describe the application domain by means of ontologies but they use specialized reasoners, such as

* This work is funded by the University of Torino, projects “Ricerca Locale” and “Ricerca Autofinanziata”.

constraint solvers or rule-based systems, to make inferences. This approach is far from optimal because it fragments knowledge in multiple data sources characterized by heterogeneous representation formats. Therefore, it limits the possibility to use complete information about the problem to be solved for the execution of inferences.

In order to address this issue, we propose a geographic reasoning model that adopts a unified knowledge management approach to represent both geographic information and reasoning rules with particular attention to constraint verification, which is central to several geographic reasoning tasks. We adhere to the GeoSpatial Semantic Web paradigm [29] that promotes standard knowledge representation languages to maximize the interoperability of data and applications. Specifically, our model uniformly represents the application domain and its constraints as OWL ontologies [58] in order to benefit from the expressiveness and reasoning tools provided by standard Semantic Web languages. Moreover, it offers a set of specialized reasoners that take as input these ontologies in order to solve different types of constraint verification problems on a selected geographical area. As a proof-of-concept, the current implementation offers two reasoners optimized to solve specific types of tasks: i.e., finding paths that connect geographical areas by traversing land patches that satisfy a given set of constraints, and clustering land patches that satisfy the same constraints to summarize the distribution of homogeneous areas in a territory.

A novel aspect of our model is the representation of constraint types as classes of an OWL ontology. In this way, we employ a single, well-known knowledge representation standard and we avoid the introduction of a new language that would require ad-hoc tools to manage constraint information. Our Constraint ontology supports a detailed description of the different kinds of constraints (e.g., soft and hard, part-of and relational, aggregation and individual) by qualifying their scope, purpose and relationships. This meta-information enables the development of automated reasoners that can autonomously retrieve and apply the relevant constraints for the task to be completed.

We test our model on the management of Ecological Networks (ENs) [6], which describe the structure of existing real ecosystems and help planning their expansion, conservation and improvement by imposing restrictions on land use. ENs have been traditionally specified as large sets of Natural Language guidelines requiring a manual design of public policies for the proposal of land use transformations. We aim at providing an interactive tool that helps the human decision-maker by automatically designing the structure of the EN of a geographic region and by suggesting suitable aggregations of land patches that satisfy a given set of constraints; e.g., having medium or low levels of irreversibility and extroversion. These functions help the design of the structure of the EN and of urban and regional transformation plans and projects to improve the compliance of a geographical area with the EN specifications. Those activities are needed to construct the social awareness on bindings and opportunities related to Ecological Networks for quality of life. The present paper brings the following main contributions:

- It provides an ontological representation of Ecological Networks (EN ontology) and of the related constraints on land use (Constraint ontology) which supports knowledge sharing and semantic reasoning.
- It presents an extensible framework for reasoning about geographical constraints, based on a semantic knowledge representation.

The present paper extends the work described in [49] in the following ways: first, it refines the EN ontology defined in that article by describing different types of linear infrastructures; e.g., roads ranging from natural paths to highways that can represent serious obstacles to connecting land patches. Moreover, the present paper significantly extends the Constraint ontology proposed in [49] by defining more complex types of conditions to be used in constraints. Finally, by exploiting a number of examples that make use of the extensions to the ontology, this paper provides more detail about the reasoners we developed.

In the following, Section 2 provides background information and positions our work in the related one. Section 3 presents our knowledge representation and reasoning model. Section 4 describes the reasoners we developed. Section 5 describes the framework implementation. Section 6 concludes the paper and presents our future work.

2 Background and Related Work

2.1 Semantic Knowledge Representation

According to Gruber, an ontology is an explicit specification of a conceptualization [24]. Moreover, in [25], Guarino et al., explain that ontologies "may be classified into different types, depending on the way they are used. For instance, the primary purpose of *top-level ontologies* lies in providing a broad view of the world suitable for many different target domains. *Reference ontologies* target the structuring of ontologies that are derived from them. The primary purpose of *core ontologies* derives from the definition of a super domain. *Application ontologies* are suitable for direct use in reasoning engines or software package." In this paper, we define two application ontologies: the former, henceforth denoted as the EN ontology, specifies the types of elements that constitute an Ecological Network; the latter, denoted as Constraints ontology, defines geographical constraints.

The GeoSpatial Semantic Web vision advocates for representing geographical information by means of ontologies suitable for explicitly describing concepts and relations among concepts [29]. This supports a conceptual notion of data interoperability, which goes beyond the adoption of a common representation format and is aimed at enabling correct data interpretation and inferences in geographical reasoning. The interoperability issue has been studied in other works, related to information sharing and retrieval: for instance, Fonseca et al. propose an ontology to classify geographic elements with respect to geometric characteristics and attribute values [17]. They also analyze the impact of semantic granularity, i.e., the level of detail at which geographic objects are described, on interoperability [18]. On a similar perspective, Mauro et al. analyze the impact of semantic granularity on geographic information search support [36] and Palacio et al. investigate spatial granularity to describe toponyms at different levels of detail [40]. Some ontologies support the sharing of toponyms and generic geographic concepts; e.g., the GeoNames Mappings ontology [19] based on the GeoNames database [20]. Furthermore, specific ontologies describe fine-grained aspects of geographical objects; for instance, GeoSPARQL [38] defines geographical objects supporting the specification of their geometry, as well as topologic relations among different objects. Finally, other ontologies provide a semantics of Volunteer Geographical

Information; for instance, LinkedGeoData links crowsourced OpenStreetMap (OSM) information to GeoNames and others ontologies [30].

Moving from knowledge representation to reasoning, in [48] and [50] Torta et al. propose the GeCoLan language for constraint-based reasoning on semantic geographical data, applied to the validation of Ecological Networks. Similarly, some Semantic Web languages allow the definition of constraints as generic rules (e.g., SWRL - Semantic Web Rule Language - [27] and RuleML [8]) or as logic formulas (e.g., see logical/functional languages such as the CIF Constraint Interchange Format [22]). All these works are affected by two main limitations: first, they require ad-hoc reasoners to perform inferences. Second, they fail to characterize the properties of constraints as needed to automatically retrieve and apply them to specific reasoning tasks. Our current work addresses both limitations because it represents ENs and constraints in the same language and it supports a detailed specification of the latter by qualifying their scope, purpose, relationships, so that an automated reasoner can retrieve and apply the appropriate ones, given the task to be completed. These characteristics associate our work to some recent research about spatial reasoning that investigates the homogeneous representation of different types of knowledge supporting complex tasks. For instance, Lazoglou and Angelides propose an ontology to model actors and tasks for spatial planning systems [32]; moreover, Abrahao and Hirakawa propose a task ontology for agriculture operations [1]; furthermore, the Spatial Decision Support Consortium promotes an ontology specifying spatial decision making [44]. Indeed, we have a similar perspective but a different purpose because we aim at reasoning about constraints.

Our work also differs from the path finding approaches adopted in location-based services and recommender systems. Given a graph representing the travel map of a geographical area, those models suggest paths suiting individual preferences by composing road segments which, globally, maximize one or more measures associated to the selected properties; e.g., the shortest path between two endpoints, or a path maximizing pleasure, calm, or other properties of an area [43]. Those models solve a specific task by taking a pre-defined set of constraints into account. Differently, we aim at supporting multiple reasoning tasks and at retrieving relevant constraints from a semantic knowledge base by using their description as classes of an OWL ontology.

2.2 Ecological Networks

Urban land use has dramatically extended towards natural spaces: external urban areas, such as uncultivated or abandoned cultivated land, burnt areas and degraded forests, have often been confined from urban and regional planning to a secondary position and sometimes simply considered as “waiting areas for a new urbanization” [54].

Ecological Networks (ENs) have been proposed to preserve biodiversity and enhance ecosystem services [28] by reducing the process of nature and landscape fragmentation and vulnerability caused by the development of new urbanizations, infrastructural networks and intensive agriculture [31]. As reported in [6], ecological networks share two generic goals: firstly, they are aimed at “maintaining the functioning of ecosystems as a means of facilitating the conservation of species and habitats”. Secondly, they are aimed at “promoting the sustainable use of natural resources in order to

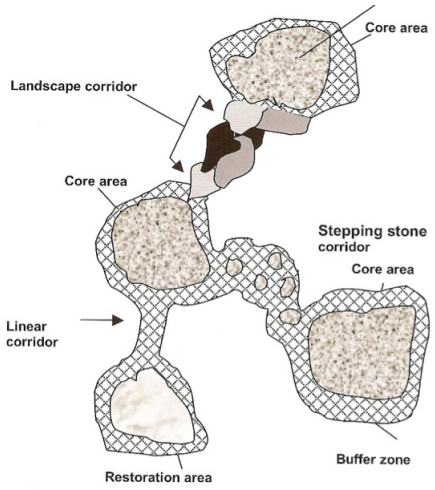

Fig. 1. Ecological Network representation, from [6]

reduce the impacts of new urbanizations on biodiversity and/or to increase the biodiversity value of managed landscapes".

Even though the Protected Areas and *Natura 2000* sites are now considered the backbone of European policy for biodiversity, the increasing expansion of urbanization and infrastructural networks is challenging the conservation of natural habitats for the preservation of animal species and plant varieties. It is thus necessary to develop policies for the improvement of Ecological Networks in order to overcome the fragmentation of habitats and natural areas, which is the main cause of biodiversity loss in Europe. The consequences of these processes can be summarized as follows [5]:

- Degradation of wetlands, which compromise the following ecological functions: control of water flows, ability to block sediments, support to plant and animal species (stepping stone function), ability to provide nutrients for the ecosystems.
- Loss of natural areas due to urban development and fragmentation of natural areas into smaller, disconnected patches that become isolated.
- Inability of ecosystems to respond to changes and find a new ecological balance; the effect is a significant reduction of resilience.
- Loss of ecosystem services: natural systems are considered essential "services", such as the control of water, the filter functions for pollutants, the preservation of climate change and environmental risks.
- The increased economic costs for public services, caused by the response to natural disasters deriving from human footprint.

An Ecological Network can be defined as an interconnected system consisting of territorial areas that include natural and semi-natural habitats. As shown in Figure 1, the typical representation of an EN is a network of core areas interconnected by corridors. According to Bennet and Mulongoy [6]:

- *Core Areas* are the areas "where the conservation of biodiversity takes primary importance, even if the area is not legally protected".
- *Adjoining Areas*, also known as *Buffer Zones* are the areas that "protect the network from potentially damaging external influences and which are essentially transitional areas characterized by compatible land uses". They are important to safeguard and increase the stability of the core areas; see also [54].
- *Corridors* "serve to maintain vital ecological or environmental connections by maintaining physical (though not necessarily linear) linkages between the core areas".
- *Sustainable-use areas* are zones "where opportunities are exploited within the landscape mosaic for the sustainable use of natural resources together with maintenance of most ecosystem services".

Until now, the research on Ecological Network management has focused on the following main topics:

- Providing evaluation frameworks to simulate the evolution of an EN starting from its initial state. This is aimed either at predicting the future state of a given geographical area or at simulating the effects of some planned actions on the area. For instance, some researchers propose mathematical simulations to model the interaction between organisms within an ecosystem, the dynamics of the relations among species, the existence of dynamical bottlenecks in the functioning of the ecosystems, etc.; e.g., see [15, 51, 15, 34, 21, 41]. These works are complementary to our own: in fact, EN simulation helps foresee the consequences of actions on a geographical area; however, it does not support the assessment of the status of the area, given its properties. Therefore, these works could support the identification of land use constraints to be imposed in order to safeguard an ecologic area, but they cannot support the identification of obstacles to the satisfaction of such constraints, or the suggestion of how to resolve the obstacles.
- Implementing ENs at different scales, from European ones down to small-scale ones such as those developed by the municipalities. The results of these implementations, built on the basis of a thorough analysis of the involved geographical areas, are guidelines on land use and planning documents; for instance, see [13, 7, 6] and the example used in this work [11]. Unfortunately all these documents are written in Natural Language, posing different challenges to the human planner: first, the lack of a formal specification makes it difficult to check the consistency among guidelines in the cases where they have to be jointly applied. Second, the EN elements of a geographical area have to be manually identified, posing a heavy burden on the decision-maker and exposing her/him to the risk of making mistakes. A formal representation of the concepts underlying Ecological Networks and of the properties of geographical areas is the missing building block for the development of any automated tool aimed at supporting this type of task.

Our work is concerned with the second topic above and aims at helping human planners through ICT. In our previous work [48, 50], we proposed a semantic representation of ENs for their automated validation. However, as previously discussed, we introduced an ad-hoc constraint satisfaction language for the verification of ENs; e.g., to check whether a certain area, identified as a Buffer zone in a pre-defined EN, complies with

the definition given in the specifications, or not. In the present work, we go one step forward by introducing a uniform representation of domain knowledge and inference rules, based on Semantic Web technologies, in order to provide a unified approach to the management of ENs.

Before concluding this section it is worth noting that, as far as the representation of Ecological Networks is concerned, some ontologies model the types of *land use/cover*; e.g., LBCS-OWL2 [37] and HarmonISA [26]. While those ontologies are interesting models, in our work we use a taxonomy based on the Land Cover Piemonte (LCP) cartography [42] because the experimental data available to us is tagged according to it; see project [11]. However, our approach is general and could be adapted to work on the basis of other specifications.

3 Knowledge Representation: Ontology and Graph Models

3.1 OWL Representation of Ecological Networks

The EN ontology describes the main concepts and relations of Ecological Networks starting from two main sources of information:

- The former is the set of Natural Language specifications produced in project "Experimental activity of participatory elaboration of ecological network" [11]. This project was carried out by the Metropolitan City of Turin (Italy) [12] in collaboration with Polytechnic of Turin and ENEA [14]; it aimed at defining a proposal for the Ecological Network implementation at the local level in two pilot municipalities near Turin. The goals were guiding local Public Administrations with measures to limit anthropogenic land use and, where possible, orienting and qualifying the conservation of ecosystem services.
- The latter is the GeoSPARQL ontology [38], which defines the *Feature* class to represent geographical information. A *Feature* has a *Geometry* on the 2D plane and can thus be used to represent points, lines and areas on a map, known in the literature as Simple Features [39]. GeoSPARQL also defines a set of topological geometric relations between *Feature*s that correspond to basic relations such as *intersects* (to represent geometric intersection), *equals* (to represent equality of geometries) and *contains* (to represent the fact that a geometry includes another one).

The EN ontology is defined using the OWL language [58] and it is composed of two main portions:

- The EN Domain ontology defines the concepts related to types of land use, land patches, and similar.
- The EN Task ontology describes the types of intervention that can be planned on a geographical region.

Figure 2 shows the main classes of the EN Domain ontology and of the portion of the GeoSPARQL ontology we used.[3] Following the graphic notation described in [52], the arrows with open heads symbolize subclass relationships between classes, while regular arrows connect domains and ranges of properties.

[3] All the graphs describing portions of the EN and Constraint Ontologies have been produced using the Dia Editor [35].

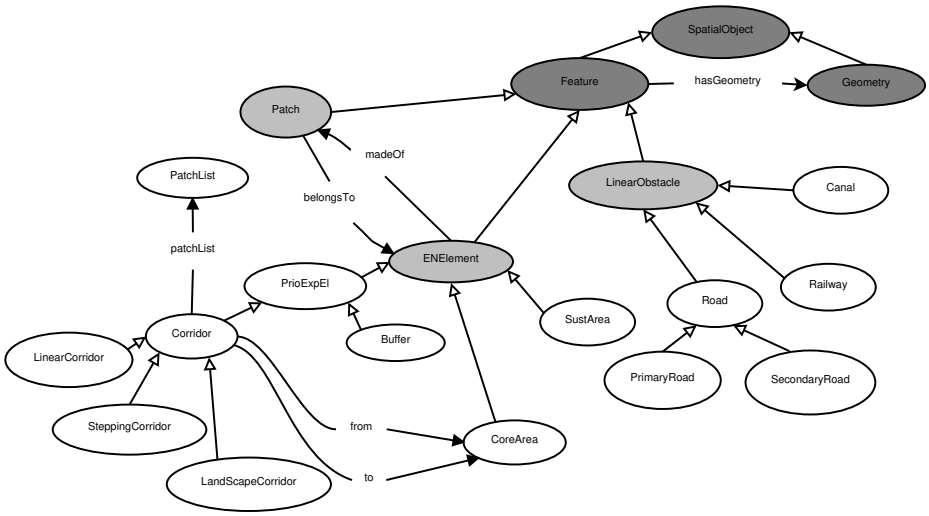

Fig. 2. A portion of the EN Domain Ontology

The top class is *Feature*, imported from the GeoSPARQL ontology. In order to define ENs, we introduce four subclasses of *Feature* that are the roots of the hierarchies of classes describing the core of the domain. In the figure, *Feature* is depicted in dark grey and the roots of the hierarchies of EN elements are depicted in light grey for easy identification. Specifically:

- *ENElement* represents a generic element of the EN and can be either a Core Area (*CoreArea*), a Sustainable-use area (*SustArea*), or a Priority Expansion Element (*PriorExpEl*). In turn, Priority Expansion Element has Corridor (*Corridor*) and Buffer Zone (*Buffer*) as more specific classes.
- *Patch* represents a small geographical area characterized by a specific land use. It is worth noting that each instance of *Patch belongsTo* an instance of *EcologicalNetworkElement*; conversely, each instance of *EcologicalNetworkElement* is *madeOf* one or more instances of *Patch*.
- *LinearObstacle* represents the linear obstacles that can separate land patches. There are various types of linear obstacles, represented as more specific classes. In the EN ontology we represent canals (*Canal*), railways (*Railway*), and roads (*Road*). In turn, roads can be primary ones, to denote highways and other major ones (*PrimaryRoad*) and secondary ones (*SecondaryRoad*).

The *LandUseElement* hierarchy of the EN Domain ontology describes the types of land use: each instance of *Patch* is *describedBy* a *LandUseElement*, i.e., it is associated with a specific land use. See Figure 3, where the *describedBy* relation links the *Patch* class to the *LandUseElement* one. Each class of this hierarchy is a singleton and includes exactly one representative object characterized by:

- A specific type of land use; e.g., wetland (*WetLand*), wooden land (*WoodenLand*), and similar, as defined in the Land Cover Piemonte (LCP) cartography [42]. The

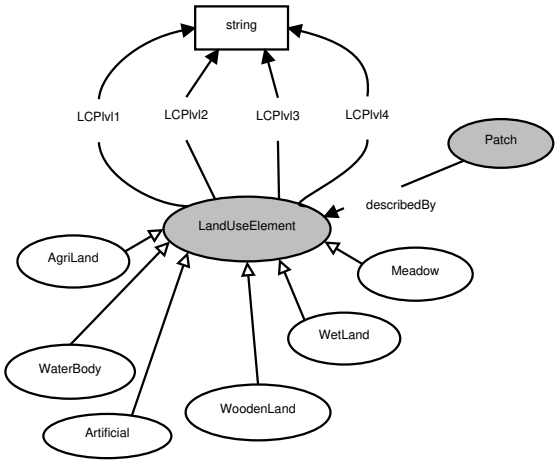

Fig. 3. A portion of the LandUseElement Hierarchy (EN Domain Ontology), from [49]

LCP defines a hierarchy of land use types organized in 4 levels that describe land use at different specificity levels: the first one (*LCPlvl1*) is the less detailed one and includes 5 general classes of land use; the second one (*LCPlvl2*) is more specific and includes 15 classes; the third one (*LCPlvl3*) includes 45 classes; the last one (*LCPlvl4*) is the most specific one and it includes 97 classes of land use.

– The score obtained with respect to five evaluation criteria taken from [53]. We represent these criteria as OWL properties of *LandUseElement* in the ontology but we do not show them in Figure 3 for brevity:
  - *naturalness* (how close the element is to a natural environment);
  - *relevance* (how relevant it is for the conservation of the habitat);
  - *fragility* (how fragile the element is with respect to anthropogenic pressure);
  - *extroversion* (how much pressure it can exert on the neighboring patches);
  - *irreversibility* (how difficult it would be to change the use of the element).

  The value for each criterion ranges from 1 to 5 and 1 is the maximum value.

The EN Task ontology includes the *Intervention* and *Operation* subclasses of *Feature* (see Figure 4), which describe the types of activity related to the planning of improvements and expansions of Ecological Networks. More specifically:

– *Intervention* represents an intervention for building, improving or conserving the Ecological Network.
– *Operation* represents a specific operation of elimination (*Elimination*), construction (*NewPlanting*), or maintenance (*Maintenance*) that is part of an intervention; see the *comprises* relation between *Intervention* and *Operation*.

The current version of the EN ontology models interventions and operations at a coarse granularity level. However, we plan to refine them on the basis of recent work about spatial planning [32] and Spatial Decision Support Systems [44] which supports the automated management of activities by modeling actors and tasks in a detailed way.

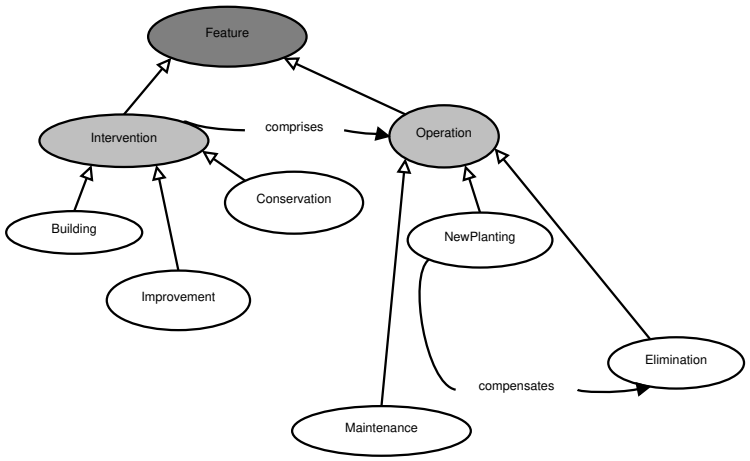

Fig. 4. A portion of the EN Task Ontology

3.2 OWL Representation of Geographical Constraints

As discussed by Louwsma et al. in [33], constraints in a geographical domain must be able to express restrictions on the instances of the classes of the domain ontology by specifying logic, geometric, and numeric requirements. For example, constraints can define the allowed values of categorical attributes of areas, they can be used to compute the sum of the sizes of a set of areas, or they can restrict the topological relations between pairs of areas.

In order to support the specification of constraints related to Ecological Networks, we define a Constraint ontology whose classes refer to the classes and properties of the EN ontology[4]. Moreover, we provide a flexible representation to compose constraints that allows to define both simple constraints and complex ones. The representation takes inspiration from the typical types of constraints that may appear in a generic configuration knowledge base; e.g., see [45, 46, 16].

Figure 5 shows a portion of the Constraints ontology, which is structured as a hierarchy rooted by class Constraint (in dark gray):

- *PartOfConstraint* (shortened to *PartOfCons* in the figure) describes the constraints that apply to one or more parts of a given object. It should be noticed that a part may be shared by different objects; thus, the semantics of this type of constraint is similar to the *aggregation* of UML [23].
  - *SingleAttributeConstraint* (*SingleAttrCons*) involves a single (part-of) attribute of the object.
  - *MultiAttributeConstraint* (*MultiAttrCons*) involves more than one (part-of) attribute of the object.

[4] In OWL, referring to classes and properties as values of other properties is problematic; see [55]. We avoid these difficulties by only using such references in SPARQL [57] queries.

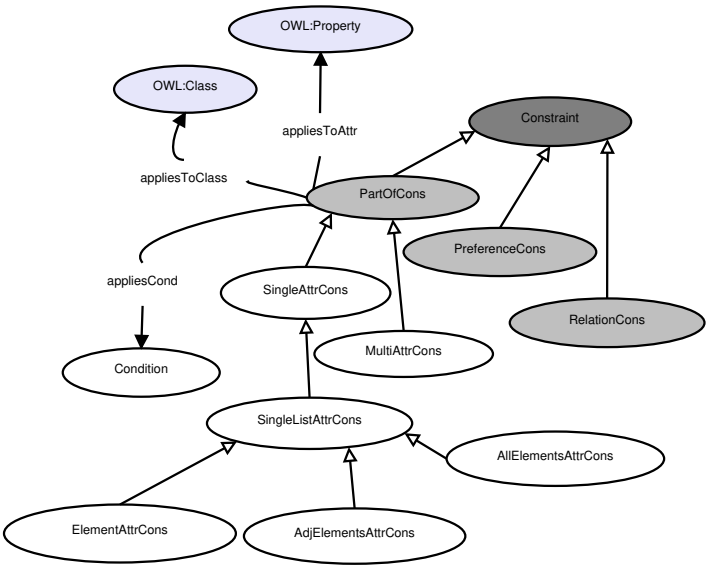

Fig. 5. A portion of the Constraints Ontology

- *RelationConstraint* (*RelationCons*) describes the constraints that apply to a relationship between more than one object.
- *PreferenceConstraint* (*PreferenceCons*) represents soft constraints, which augment regular constraints with functions to be optimized.

Let us focus on the *SingleAttributeConstraint*s, henceforth denoted as SACs. They refer to a single class (*appliesToClass*) and attribute of that class (*appliesToAttribute*). Note that an *attribute* of class *C* is an OWL property with class *C* as a possible domain. A special kind of SAC, *SingleListAttributeConstraint*, applies to attributes that are ordered lists of objects or values. In that case, it is important to distinguish among the cases when the constraint applies to the individual elements of the list (*ElementAttributeConstraint*), to pairs of adjacent elements (*AdjElementsAttributeConstraint*), or globally to all the elements of the list (*AllElementsAttributeConstraint*).

A SAC specifies a condition by means of the *appliesCondition* property; see Figure 6. We distinguisch between two types of conditions:

- *AggregateCondition*s (*AggregateCond*, in gray) specify restrictions on some aggregate quantity computed from the elements of a list attribute or from a subset of them.
- *IndividualCondition*s (*IndividualCond*) apply to each element (or pair of elements) of a list attribute, or to the unique value of a scalar attribute.

Individual conditions can be created by composing *AtomicCondition*s into *CompositeCondition*s with the usual logic connectives: *and*, *or* and *not*. Moreover, *QuantifierCondition*s specify a quantified *variable* (*QVar*) to be used within an inner *subcondition*. A *QVar* has several properties that make it a powerful concept:

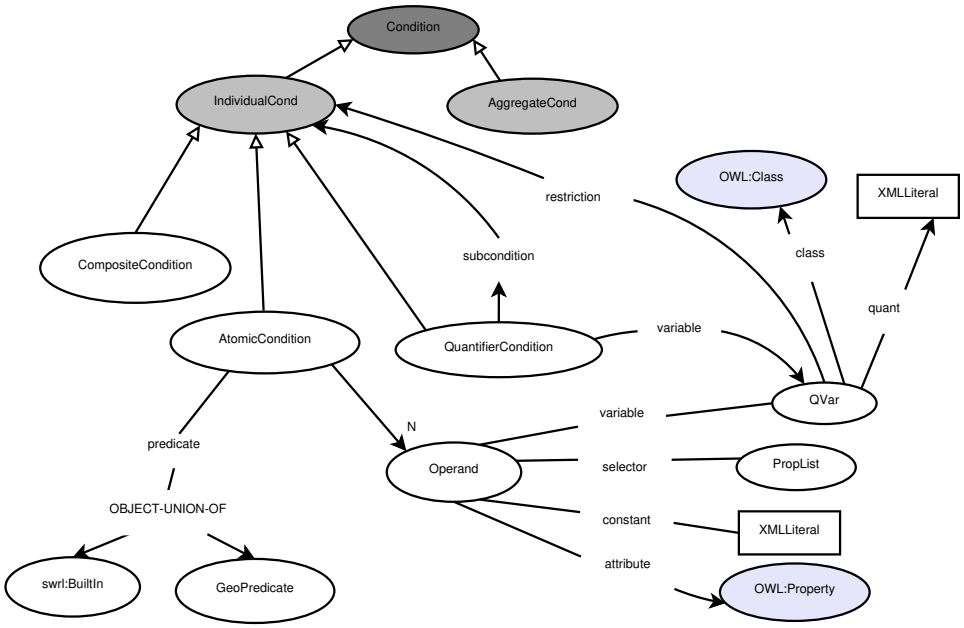

Fig. 6. A portion of the Condition Hierarchy (Constraints Ontology)

- a quantifier *quant* (*forall*, *exists*);
- a *class* from which the variable takes values (this is the domain of the variable);
- a *type* (*part*, *other*), which for *PartOfConstraint*s specifies whether the variable ranges over the parts involved by the constraint or not;
- an optional *restriction* that puts a further condition on the values over which the variable should range.

An *AtomicCondition* has a *predicate*, that can be either an SWRL built-in predicate (e.g., *equal*, *lessThen*, *add*, *subtract*, ...) or a *GeoPredicate*, i.e., a predicate that relates the geometric properties of two or more *Feature*s. The *GeoPredicate* class contains both GeoSPARQL and additional properties defined and implemented in the present work[5]. The condition has one or more *Operand*s, which can specify:

- a quantified *variable* (*QVar*);
- a *selector* modeled as a list of properties;
- a *constant* value; i.e., *XMLLiteral*;
- an *attribute*.

As a SAC applies to an attribute *A* of a class *C*, by default an operand refers to the value of *A* in the objects *O* of class *C*. However, things can be customized by specifying a *constant* value, the *attribute* of *O* to consider, or the *selector* (list of properties) that

[5] So far, we added one custom property (named *separates*) that will be used in the examples below.

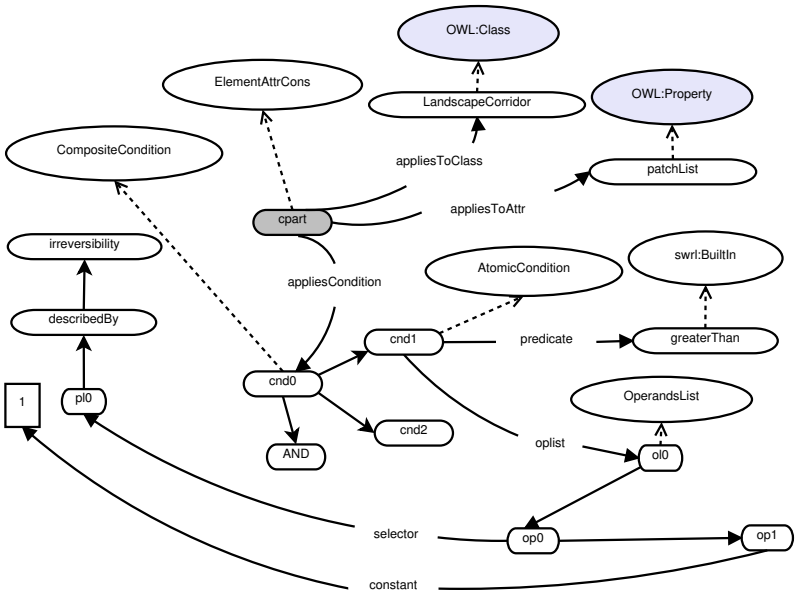

Fig. 7. The *cpart* (Corridor Part) constraint

should be followed from *A* to get to the value of the operand. Moreover, it is possible to specify a quantified *variable* (*QVar*) of an enclosing *QuantifierCondition*; in that case, the operand refers to the value taken by the variable.

It should be noticed that the the main goal of this representation is the specification of various metadata about constraints; for instance, see the distinction between part-of, relation and preference constraints. The ontology can be extended and refined as needed to express additional metadata. This is a key element for the development of reasoners that automatically retrieve suitable constraints to perform constraint solving, given the characteristics of the input problem.

Example 1. Let us consider the *LandScapeCorridor* class of the EN Domain ontology (see Figure 2). The guidelines for the Local EN implementation devised in project [11] state that:

> *Corridors avoid areas with maximum irreversibility and areas with maximum extroversion.*

A landscape corridor is therefore made of patches that must exhibit the specified characteristics. Figure 7 depicts the specification of the constraint that enforces these prescriptions: we associate the constraint *cpart* with class *LandScapeCorridor*[6]. Constraint

[6] Following the graphic notation described in [52], the rounded rectangles represent individuals, while dashed arrows symbolize instance-of relationships.

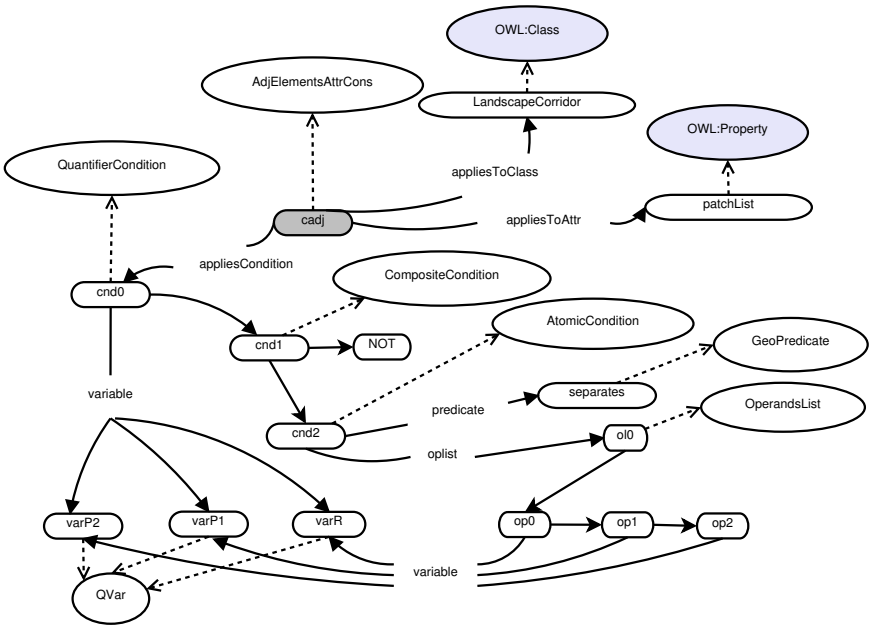

Fig. 8. The *cadj* (Corridor Adjacency) constraint

cpart is an *ElementAttributeConstraint* because it applies to each element of the *patchList* property of *Corridor*. It specifies a *CompositeCondition cnd0* that consists of the conjunction of two *AtomicCondition*s:

- The former *AtomicCondition*, *cnd1*, requires a non-maximum irreversibility and it specifies:
  - the *predicate* as *swrlb:greaterThan*;
  - the operation list (*ol0*) that specifies the two operands of *cnd1* as a linked list. Specifically:
    - the first *Operand* is element *op0* in *oplist ol0*; *op0* is a *selector* and points to a *PropertyList pl0* that contains as its elements properties *describedBy* (from *Patch* to *LandUseElement*) and *irreversibility* (from *LandUseElement* to the value of the irreversibility criterion);
    - the second *Operand op1* is the *constant* value 1.
- The second *AtomicCondition cnd2* is similar to the first one but it requires non-maximum extroversion and is not detailed in Figure 7 for shortness.

Example 2. Let us consider again the *LandScapeCorridor* class in the EN Domain ontology. A specification taken from [42] states that:

> *The design of Corridors should avoid major linear obstacles (highways, high-speed railways, large artificial canals).*

We can associate a suitable constraint with class *LandScapeCorridor*. The *cadj* constraint, depicted in Figure 8, has the following traits:

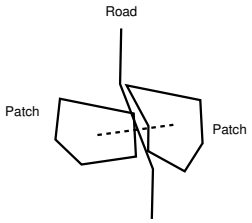

Fig. 9. The *Separates* predicate

- it is an *AdjElementsAttributeConstraint*, because it constrains adjacent elements of the *patchList* property;
- it specifies a *QuantifierCondition cnd0* that quantifies the following *QVar*s:
  - *varR*, ranging *forall* the *Road*s with *hasTraffic* property greater than 2;
  - *varP1*, ranging *forall* the *Patch*es that can constitute *single_part*s of the *patchList*;
  - *varP2*, ranging *forall* the *Patch*es that can constitute *single_part*s of the *patchList*;
- the subcondition of *cnd0* is a *CompositeCondition cnd1* that negates (*NOT*) its own subcondition *cnd2*;
- *AtomicCondition cnd2* specifies:
  - the predicate *separates*, which takes value *true* iff its first operand (a *Road*) separates the second and the third operands (two *Patch*es); this corresponds to checking whether the segment conjoining the centers of the two patches intersects the road, as shown in Figure 9;
  - the *oplist ol0* containing operands that refer to *variable*s *varR*, *varP1*, and *varP2*.

3.3 Representation of Constraints and of Individual Information Items

The instances of the constraints classes representing the actual constraints that apply in the domain, such as the sample ones described at the end of the previous section, are stored in RDF format [56] in a triple store that represents the knowledge base used by the system. The triple store also contains the instances of the classes defined in the EN Domain ontology, such as the *Patch*es of land that form the map of a specific geographic area of interest. As far as geographic items are concerned, the translation from input data-sets (typically available as ESRI shapefiles) to RDF triples is carried out by our data import functions described in Section 5.

3.4 Supporting Efficient Geographic Reasoning: the Adjacency Graph Model

While, starting from the RDF representation of domain knowledge, an automated reasoner can obviously perform the appropriate inferences to solve an input problem, several basic inferences could be pre-compiled to speed up execution. In particular, the adjaciency relations between the land patches of a geographical area are expected to

change very rarely; therefore, they can be pre-processed and made available to the automated reasoners as aggregated data.

In this perspective, beside the OWL representation of knowledge described in the previous sections, we consider a graph model that can be derived from the RDF instances of the knowledge base and is particularly useful for the reasoning features of our system; see Section 4. The graph $\mathcal{G} = (N, E)$, denoted as the Adjacency Graph, is structured as follows:

- the nodes N correspond to areas of a map;
- the arcs E connect nodes whose associated areas are adjacent in the map.

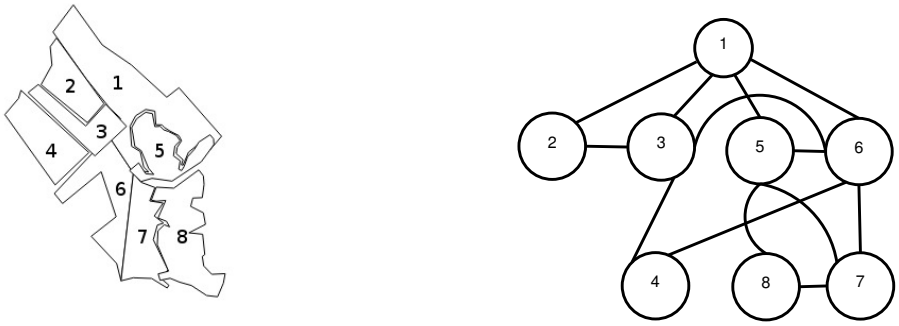

Fig. 10. A map and its corresponding Adjacency Graph, from [49]

Figure 10 shows a map and its Adjacency Graph. Each node of the graph is associated with an area of the map; moreover, areas and nodes are numbered consistently to show their correspondences. For example, node *1* corresponds to an area that is adjacent to the areas associated to nodes *2*, *3*, *5* and *6*. It can be noticed that, in the generation of the Adjacency Graph, some noise in the map data has been removed. For instance, nodes *5* and *8* are connected in the Adjacency Graph even though the borders of their areas are not exactly adjacent in the map. This type of abstraction is needed to deal with real-world, imperfect GIS data, and is a basic pre-processing task that can be performed by our system. Specifically, when a new area A is inserted into the knowledge base of the system, standard geometric algorithms are used to compute an expansion A' of A which extends A with a border of a given thickness, and to determine the adjacent areas as the ones that intersect with A'.

It might be questioned why, instead of exploiting standard geographic reasoning functions such as those offered by GeoSPARQL, we developed our own ones. Indeed, while GeoSPARQL provides a set of functions to compute the *Simple Features* topological relations it defines, those functions require that the involved geometries exactly satisfy the corresponding relations. For instance, according to GeoSPARQL, an area *touches* another area *iff* they share some common points on their borders, but they do not share any internal points. This function is clearly too restrictive to determine the

adjacency of two geographical areas in a meaningful way for our purposes because the specification of both areas could be noisy.

Overall, we use the Adjacency Graph of a geographical area to store more information than the association between nodes and areas. Specifically:

- Each node $n \in N$ can have attributes representing meta-information about the area $\mathcal{A}_n$ associated with it. In the EN domain, this may include:
  - the values of the evaluation criteria and the LCP levels of the *LandUseElement* describing $\mathcal{A}_n$;
  - information such as the area size and perimeter of $\mathcal{A}_n$;
  - the identity of the EN element to which $\mathcal{A}_n$ belongs; e.g., a *CoreArea*.
- Each arc $e = (n_i, n_j) \in E$ can have attributes that represent meta-information about the relationship between $\mathcal{A}_{n_i}$ and $\mathcal{A}_{n_j}$. For example, the arc can have:
  - the length of the perimeter shared by the two areas;
  - a numeric "cost" that describes how difficult is to move from $\mathcal{A}_{n_i}$ to $\mathcal{A}_{n_j}$. This cost is determined by the presence of an obstacle (instance of class *LinearObstacle* of the EN Domain ontology) between the two areas.

4 A Portfolio of Reasoners

4.1 Reasoning Tasks

Our model supports reasoning tasks based on the following kinds of inputs:

- the EN and Constraints ontologies, which describe the domain concepts, the constraints hierarchy and their relationships;
- the RDF data representing the instances of domain classes of the EN ontology; e.g., individual land patches included in the geographic area of interest;
- the RDF data representing the instances of the constraints (classes of the Constraints ontology) that apply to the specific domain;
- further requirements provided by the user to specify the desired reasoning task and its parameters.

Ideally, we would like that the system automatically extracts all the data needed to perform a requested task from the above listed sources of information, and use it to drive a generic reasoning engine that computes the answer by exploiting the RDF domain instances. However, this type of generality would be extremely hard if not impossible to achieve in practice. Therefore, we equip the system with a pre-defined (but extensible) set of reasoning capabilities that can be reused in different tasks and fill the details of specific reasoning task requests.

Definition 1. *A* Reasoner $\mathcal{R}(\Omega, \Delta, \rho)$ *is a function that takes as inputs an OWL ontology* Ω*, a RDF graph* Δ*, and a request* ρ*, where* Ω *is partitioned in two sets* Ω_{DOM} *(EN Domain ontology classes) and* Ω_{CONS} *(EN Constraint ontology classes), and, similarly,* Δ *is partitioned in two sets* Δ_{DOM} *and* Δ_{CONS}*. The reasoner performs the following steps:*

1. *it extracts from* Δ_{CONS} *(driven by ontology* Ω_{CONS} *and request* ρ*), a relevant set of constraints:*
$$\mathcal{C} = \mathcal{C}_G \cup \mathcal{C}_R$$
denoting, respectively, graph constraints and reasoning constraints;
2. *using* $\mathcal{C}_G$*, it extracts from* Δ_{DOM} *the data* $\mathcal{D}_G$ *useful for building the Adjacency Graph model;*
3. *using* $\mathcal{D}_G$*, it builds the Adjacency Graph model* $\mathcal{G}$*;*
4. *using* $\mathcal{C}_R$*, it extracts from* Δ_{DOM} *the additional data* $\mathcal{D}_R$ *useful to support the reasoning task;*
5. *using* $\mathcal{D}_R$*, it performs a reasoning task on* $\mathcal{G}$ *by enforcing the constraints in* $\mathcal{C}_R$*;*
6. *it returns a result* α *that* answers *the request* ρ*, given* Ω *and* Δ*.*

The extraction of constraints (step 1 above) is done by issuing SPARQL queries [57] on the RDF data Δ_{CONS} using the vocabulary of ontology Ω. The retrieved constraints are represented as internal data structures that the reasoner can use to perform steps 2, 4 and 5 above. Specifically:

- $\mathcal{R}$ uses the constraints $\mathcal{C}_G$ to automatically generate SPARQL queries that extract from Δ_{DOM} the data $\mathcal{D}_G$ needed to build the nodes of graph $\mathcal{G}$;
- $\mathcal{R}$ uses the constraints $\mathcal{C}_R$ to automatically generate SPARQL queries that extract from Δ_{DOM} other, additional data $\mathcal{D}_R$ needed by the reasoner;
- finally, $\mathcal{R}$ directly evaluates in-memory the constraints $\mathcal{C}_R$ by using data $\mathcal{D}_R$ while it searches for a solution by visiting graph $\mathcal{G}$.

Specific tasks are requested by executing *Commands* that are translated to one or several invocations of the reasoner with specific values of request ρ.

Currently, we have implemented the following two reasoners:

- $\mathcal{R}_{CLUST}$: starting from a given *Patch*, it computes a clustering of the surrounding patches that satisfy the constraints associated with a given property. The reasoner can be used to implement the command *BUILD(CoreArea, id)*, which creates an instance of the *CoreArea* class by clustering the patches that have high or medium *ecological functionality*. The ecological functionality depends on their *naturality* and *relevance*.
- $\mathcal{R}_{PATH}$: given two *CoraArea*s, this reasoner computes a path that is composed of patches satisfying the constraints associated with a given property. The next section describes $\mathcal{R}_{PATH}$ in detail.

4.2 The $\mathcal{R}_{PATH}$ Reasoner

The $\mathcal{R}_{PATH}$ reasoner receives (through the request ρ) two identifiers id_s and id_e of *CoreArea*s that aggregate *Patch*es, and the name of a property *prop* that is a list of *Patch*es. It then computes a path of adjacent elements from element id_s to id_e taking into account the constraints associated with property *prop*.

This reasoner can be used to implement the command *BUILD(LandscapeCorridor, id_s, id_e)* which assigns id_s, id_e to the *from* and *to* attributes of *LandscapeCorridor*, and computes the value of the *patchList* attribute by invoking reasoner $\mathcal{R}_{PATH}$ with $\rho = (id_s, id_e, patchList)$.

Fig. 11. Sample map with patches and roads

1. First of all, the reasoner issues a number of SPARQL queries to retrieve the constraints associated with *LandscapeCorridor* and retrieves the following constraints:
   - a *ElementAttributeConstraint cpart* associated with *patchList* described in Example 1;
   - an *AdjElementsAttributeConstraint cadj* associated with *patchList* described in Example 2.

   Constraint *cpart* is a $\mathcal{C}_G$ constraint, i.e., it is used to identify the nodes of the adjacency graph $\mathcal{G}$. Constraint *cadj* is a $\mathcal{C}_R$ constraint, i.e., it is used directly by the reasoner during the search for a solution.
2. Then, the reasoner builds an Adjacency Graph $\mathcal{G}$ in such a way that the nodes of $\mathcal{G}$ are associated to patches that satisfy *cpart*; i.e., they have non-maximum irreversibility and extroversion.
3. After that, $\mathcal{R}_{PATH}$ considers constraint *cadj* and realizes that, for its enforcement, it needs to retrieve the additional data $\mathcal{D}_R$ consisting of all the *Road*s with *hasTraffic* ≥ 2.
4. Finally, the reasoner applies a simple path-finding algorithm based on the well-known *Dijkstra* algorithm [2] to identify a corridor between the id_s and id_e elements, if any. As the *cadj* constraint is of type *AdjElementsAttributeConstraint*, $\mathcal{R}_{PATH}$ applies it whenever it explores any further nodes that are adjacent to the currently considered node. In particular, a node N' is considered adjacent to a node N iff there is no road selected by $\mathcal{D}_R$ that *separates* N and N'. Note that, in order to evaluate this condition, the reasoner needs to invoke an in-memory function that implements the *separates GeoPredicate*, with nodes N and N', as well as data $\mathcal{D}_R$, as arguments.

Example 3. As an example of the use of reasoner $\mathcal{R}_{PATH}$, let us consider the map depicted in Figure 11, where:

- patches 2,3,5,6,8, and 9 (lighter gray) are *describedBy LandUseElement*s of type *WoodenLand*

- patches 1,4, and 7 (darker gray) are *describedBy LandUseElement*s of type *Meadow*
- the dashed-line road is a local road (with *hasTraffic* equal to 1)
- the solid-line road is a secondary road (whose *hasTraffic* equal to 3)

Let us further assume that patches 1 and 8 correspond to two *CoreArea*s. Reasoner $\mathcal{R}_{PATH}$ can then be invoked with $id_s = 1$ and $id_e = 8$ in order to try to find a path of suitable adjacent patches that connects them. First of all, we note that all the numbered patches in Figure 11 satisfy constraint *cpart*, i.e., they have admissible levels of *irreversibility* and *extrovarsion*. Therefore, they are returned as elements of the data set $\mathcal{D}_G$ used to build the Adjacency Graph $\mathcal{G}$. As for the roads in the map, only the solid-line roads are returned as elements of the data set $\mathcal{D}_R$ because the *hasTraffic* property of the dashed-line roads has a value that is too low for such roads to be relevant for constraint *cadj*. When the reasoner starts searching for a path from patch 1 to patch 8, it first considers the patches that are adjacent to patch 1, namely: 2,3,4,5, and 6. However, by applying constraint *cadj*, the reasoner immediately discards patches 2,3, and 4, since they are *separated* from patch 1 by a road belonging to data set $\mathcal{D}_R$. The search for the path to patch 8 has therefore to continue from patches 5 and 6. A possible solution is the following path:

$$(5, 9)$$

that leads from patch 1 to patch 8 by crossing a local road that can be safely ignored according to constraint *cadj*.

5 Implementation

We have implemented the model described in the previous sections as a Java library consisting of the following modules:

- *data-import* contains functions supporting the import/export of shape files to/from a triple store (e.g., Parliament [4]), the pre-processing, optimization, and conversion of the reference system of the geometries associated with geo-SPARQL *Feature*s, and the transfer of RDF triples between disk and the in-memory model of the Jena library [3] used to query the triple store. The metadata associated with the geometries in the shape file is exploited to associate such geometries with the appropriate concepts in the EN ontology (in particular, *Road*s and *Patch*es described by *LandUseElement*s);
- *reasoning* contains the functions for the creation of the Adjacency Graph data model. Moreover, it collects all the specific reasoners provided by the system (currently, the $\mathcal{R}_{PATH}$ and $\mathcal{R}_{CLUST}$ reasoners described above);
- *commands* implements the parsing of commands (currently, the two forms of the *BUILD* command described above) and interfaces with the *reasoning* and *data-import* functions to execute them;
- *shared* provides the definitions and implementations of elements relevant across the other system modules; e.g., the geometric feature and triple store manager, as well as utility functions used by the other modules.

By exploiting the *data-import* module, we have populated the Parliament triple store with 395 patches and 307 roads defined in the shape files of a portion of map situated at the north of the Italian city of Turin. We have then used the implementation of the $\mathcal{R}_{CLUST}$ reasoner contained in the *reasoning* module to generate the Core Areas as clusters of patches with given characteristics. The reasoner has generated 74 clusters. Finally, we have used the implementation of the $\mathcal{R}_{PATH}$ reasoner to generate a number of landscape Corridors between pairs of Core Areas specified by us.

6 Conclusions and Future Work

This paper has presented a semantic framework for the specification and management of constraints on a geographical domain. Our framework supports the validation of conditions on a geographic area, the composition of land patches into broader areas having homogeneous properties and the identification of paths satisfying given sets of constraints for connecting land patches. We represent both the domain knowledge and the constraints as OWL ontologies based on standard languages for knowledge representation and reasoning. This approach has several advantages: first of all, it does not introduce any special language for the management of constraints. Second, it fully exploits the knowledge representation and reasoning interoperability provided by Semantic Web languages. Third, it opens the avenue to the classification of constraints for their automated management within reasoners that can adapt to solve a possibly large range of reasoning problems.

As a test-bed we use the Ecological Networks domain. In this context, we aim at supporting both the compliance verification with respect to a pre-defined Ecological Network and the generation of a new one by suggesting suitable aggregations of land patches into EN elements. Moreover, our approach is designed to support full-fledged implementations of creation and modification tasks in order to enable the automated suggestion of *modifications* to existing EN elements through suitable interventions. Whereas we implemented reasoning about ENs as a stand-alone model, the main motivation and application of our work lies in its possible integration within Participatory Geographical Information Systems (PGIS, [47]), in order to support online interaction with stakeholders in inclusive urban planning and design processes aimed at collecting feedback and EN project proposals from stakeholders.

Our work can be extended in several directions to provide a suitable decision support system . For instance, we plan to:

- Extend the EN ontology to model finer-grained concepts. For instance, we currently describe a land patch by exclusively considering its use; e.g., wetland and wooden land. In our future work we may consider the association of more specific information with patches by exploiting existing ontologies to model further environment and ecology concepts, e.g., ENVO [10] and EcoCore [9].
- Extend the Constraint ontology to specify more types of constraints, such as *soft* constraints for guiding the automated reasoners offered by the framework to compute solutions that maximize some preference criteria, and *geometric* constraints about the shape, size, and other properties of given areas.

– Extend our reasoning framework with the ability to handle soft constraints, and with additional reasoners; e.g., for proposing maintenance and modification interventions on an EN.

7 Acknowledgements

We thank Adriano Savoca and Marco Corona for their contributions to this work.

References

1. Abrahao, E., Hirakawa, A.: Complex task ontology conceptual modelling: Towards the development of the agriculture operations task ontology. In: Proc. of 10th Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD 2018). SCITEPRESS, Seville, Spain (2018)
2. Ahuja, R.K., Mehlhorn, K., Orlin, J., Tarjan, R.E.: Faster algorithms for the shortest path problem. Journal of the ACM (JACM) **37**(2), 213–223 (1990)
3. Apache: Apache jena. https://jena.apache.org/ (2019)
4. Battle, R., Kolas, D.: Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. Semantic Web **3**(4), 355–370 (2012)
5. Benedict, M., McMahon, E.: Green Infrastructure: smart conservation for the 21st century. Watch Clearinghouse Monograph Series, Sprawl (2002)
6. Bennett, G., Mulongoy, K.: Review of experience with ecological networks, corridors and buffer zones. Technical Series **23** (2006)
7. Bennett, G., Wit, P.: The development and application of ecological networks: a review of proposals, plans and programmes. AIDEnvironment (2001)
8. Boley, H., Tabet, S., Wagner, G.: Design rationale of ruleml: A markup language for semantic web rules. In: Proceedings of the First Int. Conf. on Semantic Web Working. pp. 381–401. CEUR-WS. org (2001)
9. Buttigieg, P.L.: Ecology Core Ontology. https://github.com/EcologicalSemantics/ecocore (2018), accessed: 2018-07-28
10. Buttigieg, P.L., Pafilis, E., Lewis, S.E., Schildhauer, M.P., Walls, R.L., Mungall, C.J.: The environment ontology in 2016: bridging domains with increased scope, semantic density, and interoperation. Journal of biomedical semantics **7**(1), 57 (2016)
11. Città Metropolitana di Torino: Misura 323 del PSR 2007-2013. http://www.cittametropolitana.torino.it/cms/territorio-urbanistica/misura-323/misura-323-sperimentale (2014)
12. Città Metropolitana di Torino: Torino Metropoli - Città Metropolitana di Torino. http://www.cittametropolitana.torino.it (2019)
13. Council of Europe: General guidelines for the development of the pan-european ecological network. Nature and environment **107** (2000)
14. ENEA: Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile. http://www.enea.it/it (2019)
15. Fath, B., Sharler, U., Ulanowicz, R., Hannon, B.: Ecological network analysis: network construction. Trends in Ecology & Evolution **208**, 49–55 (2007)
16. Felfernig, A., Friedrich, G., Jannach, D., Zanker, M.: Configuration knowledge representation using UML/OCL. In: International Conference on the Unified Modeling Language. pp. 49–62. Springer (2002)
17. Fonseca, F., Egenhofer, M., Agouris, P., Câmara, G.: Using ontologies for geographic information systems. Transactions in GIS **3**, 231–257 (2002)
18. Fonseca, F., Egenhofer, M., C.A. Davis, Câmara, G.: Semantic granularity in ontology-driven geographic information systems. Annals of Mathematics and Artificial Intelligence **36**(1-2), 121–151 (2002)
19. GeoNames.org: Geonames mappings ontology. http://www.geonames.org/ontology/mappings_v3.01.rdf (2018)
20. GeoNames.org: Geonames. http://www.geonames.org/ (2019)
21. Gobluski, A., Westlund, E., Vandermeer, J., Pascual, M.: Ecological networks over the edge: Hypergraph trait-mediated indirect interaction (TMII) structure. Trends in Ecology & Evolution **31**(5), 344354 (2016)

22. Gray, P., Hui, K., Preece, A.: An expressive constraint language for semantic web applications. In: E-Business and the Intelligent Web: Papers from the IJCAI-01 Workshop. pp. 46–53 (2001)
23. Object Management Group, O.: Unified Modeling Language (UML. http://www.uml.org (2008)
24. Gruber, T.: Towards principles for the design of ontologies used for knowledge sharing. Int. Journal of Human-Computer Studies **43**(5-6), 907–928 (1995)
25. Guarino, N., Oberle, D., Staab, S.: What is an ontology? In: Staab, S., Studer, R. (eds.) Handbook of Ontologies, pp. 1–17. Springer (2009)
26. Hall, M., Mandl, P.: Spatially extended ontologies for a semantic model of harmonised landuse and landcover information (04 2006)
27. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission **21** (2004)
28. Ingaramo, R., Salizzoni, E., Voghera, A.: La valutazione dei servizi ecosistemici forestali per la pianificazione e il progetto del territorio e del paesaggio. Valori e valutazioni **19**, 65–78 (2018)
29. Janowicz, K., Scheider, S., Pehle, T., Ha, G.: Geospatial semantics and linked spatiotemporal data past, present, and future. Semantic Web - On linked spatiotemporal data and geo-ontologies **3**(4), 321–332 (2012)
30. Janowicz, K., Scheider, S., Pehle, T., Ha, G.: LinkedGeoData: A core for a web of spatial open data. Semantic Web Interoperability, Usability, Applicability **3**(4), 333–354 (2012)
31. Jongman, R.: Nature conservation planning in Europe: developing ecological networks. Landscape and urban planning **32**(3), 169–183 (1995)
32. Lazoglou, M., Angelides, D.C.: Development of an ontology for modeling spatial planning systems. Current Urban Studies **4**(02), 241 (2016)
33. Louwsma, J., Zlatanova, S., van Lammeren, R., van Oosterom, P.: Specifying and implementing constraints in GIS - with examples from a geo-virtual reality system. GeoInformatica **10**(4), 531–550 (2006)
34. Lurgi, M., Robertson, D.: Automated experimentation in ecological networks. Automated Experimentation **3**(1) (2011)
35. Macke, S.: Dia diagram editor. http://dia-installer.de/ (2019)
36. Mauro, N., Di Rocco, L., Ardissono, L., Bertolotto, M., Guerrini, G.: Impact of semantic granularity on geographic information search support. In: Proc. of 2018 IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI). pp. 323–328. IEEE, Santiago, Chile (2019)
37. Montenegro, N., Gomes, J., Urbano, P., Duarte, J.: An OWL2 land use ontology: LBCS. In: International Conference on Computational Science and its Applications. pp. 185–198. Springer (2011)
38. OGC: Geosparql vocabulary. http://schemas.opengis.net/geosparql/1.0/geosparql_vocab_all.rdf (2012)
39. Open Geospatial Consortium, et al.: OpenGIS Implementation Standard for Geographic information-Simple feature access-Part 1: Common architecture (2011)
40. Palacio, D., Derungs, C., Purves, R.: Development and evaluation of a geographic information retrieval system using fine grained toponyms. Journal of Spatial Information Science JoSIS **11**, 1–29 (2015)
41. Pilosof, S., Porter, M., Pascual, M., Kefi, S.: The mulutilayer nature of ecological networks. Nature ecology&evolution **1**, article 101 (2017)
42. Provincia di Torino: Linee guida per le reti ecologiche (in italian). http://www.provincia.torino.gov.it/territorio/file-storage/download/pdf/pian_territoriale/rete_ecologica/lgsv_lgre.pdf (2014)

43. Quercia, D., Schifanella, R., Aiello, L.: The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In: Proc. of the 25th ACM Conference on Hypertext and Social Media. pp. 116–125. HT '14, ACM, New York, NY, USA (2014)
44. SDS Consortium: Spatial decision support ontology. http://sdsportal.sdsconsortium.org/ontology/ (2017)
45. Soininen, T., Tiihonen, J., Männistö, T., Sulonen, R.: Towards a general ontology of configuration. Ai Edam **12**(4), 357–372 (1998)
46. Stumptner, M., Friedrich, G.E., Haselböck, A.: Generative constraint-based configuration of large technical systems. AI EDAM **12**(4), 307–320 (1998)
47. Sun, Y., Li, S.: Real-time collaborative GIS: a technological review. ISPRS Journal of Photogrammetry and remote sensing **115**, 143–152 (2016)
48. Torta, G., Ardissono, L., Savoca, A., Voghera, A., La Riccia, L.: Representing ecological network specifications with semantic web techniques. In: Proc. of 9th Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD 2017). pp. 86–97. SCITEPRESS, Funchal, Madeira, Portugal (2017)
49. Torta, G., andM. Corona, L.A., La Riccia, L., Voghera, A.: Ontological representation of constraints for geographic reasoning. In: Proc. of 10th Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KEOD 2018). pp. 136–147. SCITEPRESS, Seville, Spain (2018)
50. Torta, G., Ardissono, L., Corona, M., La Riccia, L., Savoca, A., Voghera, A.: GeCoLan: A constraint language for reasoning about ecological networks in the semantic web. In: Fred, A., Aveiro, Davidand Dietz, J.L.G., Liu, K., Bernardino, J., Salgado, A., Filipe, J. (eds.) Knowledge Discovery, Knowledge Engineering and Knowledge Management. pp. 268–293. Springer International Publishing, Cham (2019)
51. Ulanowicz, R.: Quantitative methods for ecological network analysis. Computational biology and chemistry **28**, 321339 (2004)
52. Van Hage, W.R., Malaisé, V., Segers, R., Hollink, L., Schreiber, G.: Design and use of the simple event model (sem). Web Semantics: Science, Services and Agents on the World Wide Web **9**(2), 128–136 (2011)
53. Voghera, A., La Riccia, L.: Ecological networks in urban planning: Between theoretical approaches and operational measures. In: Calabrò, F., Della Spina, L., Bevilacqua, C. (eds.) New Metropolitan Perspectives. pp. 672–680. Springer International Publishing, Cham (2019)
54. Voghera, A., La Riccia, L.: Ecological networks in urban planning: Between theoretical approaches and operational measures. In: Calabrò, F., Della Spina, L., Bevilacqua, C. (eds.) New Metropolitan Perspectives. Local Knowledge and Innovation Dynamics Towards Territory Attractiveness Through the Implementation of Horizon/E2020/Agenda2030, pp. 672–680. Smart Innovation, Systemsa and Technologies, Springer (2019)
55. W3C: Representing classes as property values on the semantic web. https://www.w3.org/TR/2005/NOTE-swbp-classes-as-values-20050405/ (2017)
56. W3C: Resource description framework (RDF). https://www.w3.org/RDF/ (2017)
57. W3C: SPARQL query language for RDF. https://www.w3.org/TR/rdf-sparql-query/ (2017)
58. W3C: Web ontology language (OWL). https://www.w3.org/OWL/ (2017)