

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

GeCoLan: A constraint language for reasoning about ecological networks in the semantic web

This is the author's manuscript

Original Citation:

Availability:

This version is available <http://hdl.handle.net/2318/1700778> since 2023-02-12T10:04:02Z

Publisher:

Springer Verlag

Published version:

DOI:10.1007/978-3-030-15640-4_14

Terms of use:

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

Torta G., Ardissono L., Corona M., La Riccia L., Savoca A., Voghera A. (2019) GeCoLan: A Constraint Language for Reasoning About Ecological Networks in the Semantic Web. In: Fred A. et al. (eds) Knowledge Discovery, Knowledge Engineering and Knowledge Management. IC3K 2017. Communications in Computer and Information Science, vol 976. Springer, Cham

The editorial version of this paper is published at:

<https://www.springerprofessional.de/en/gecolan-a-constraint-language-for-reasoning-about-ecological-net/16543642>

GeCoLan: a Constraint Language for Reasoning about Ecological Networks in the Semantic Web

Gianluca Torta¹[0000–0002–4276–7213], Lilitana Ardissono¹[0000–0002–1339–4243], Marco Corona¹, Luigi La Riccia²[0000–0002–4800–2641], Adriano Savoca¹, and Angioletta Voghera²[0000–0002–0166–3303]

¹ Dipartimento di Informatica, Università di Torino, Italy,
WWW home page: www.unito.it

² Dipartimento Interateneo di Scienze, Progetto e Politiche del Territorio, Torino, Italy,
WWW home page: www.polito.it

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

Abstract. Ecological Networks (ENs) describe the structure of existing real ecosystems and help planning their expansion, conservation and improvement. While various mathematical models of ENs have been defined, to our knowledge they focus on simulating ecosystems, but none of them deals with verifying whether any transformation proposals, as those collected in participatory decision-making processes for public policy making, are consistent with land usage restrictions.

As an attempt to fill this gap, we developed a model to represent the specifications for the local planning of ENs in a way that can support both the detection of constraint violations within new proposals of expansion, and the reasoning about improvements of the networks. In line with the GeoSpatial Semantic WEB, our model is based on an OWL ontology for the representation of ENs. Moreover, we define a language, GeCoLan, supporting constraint-based reasoning on semantic data. Even though this paper focuses on EN validation, our language can be employed to enable more complex tasks, such as the generation of proposals for improving ENs.

The present paper describes our ontological specification of ENs, the GeCoLan language for reasoning about specifications, and the tools we developed to support data acquisition and constraint verification on ENs.

1 Introduction

The process of fragmentation of nature and the consequent reduction of natural environments directly depend on the development of new urbanizations, infrastructural networks and intensive agriculture. Particular expressions of this fragmentation process are the endangering of plant varieties and the loss of habitats able to sustain the wild species and their migrations.

The planning of Ecological Networks (ENs) has been proposed to model and simulate biotic and abiotic ecosystems and, therefore, understand their changes towards

urban and territorial transformations: a way to overcome the isolation of protected natural areas due to the siege of wild urbanizations. The conservation of nature and the planning of ENs expansions are aimed to preserve biodiversity by reducing the process of fragmentation of nature especially in the most urbanized areas [1].

Although every sectoral approach, every point of view in urban planning maintains its value of knowledge and proposal, transversal and interdisciplinary studies have been encouraged, as far as they are able to restore the complexity of ecosystems. During the past years, several ENs have been implemented at different scales (from European down to municipality), resulting in the production of guidelines, planning documents and projects; e.g., [2–4]. Specifically, between 2014 and 2015 some of the authors of the present paper participated to an “Experimental activity of participatory elaboration of ecological network” [5] conducted by the Metropolitan City of Turin (Italy)³ in collaboration with Polytechnic of Turin⁴ and ENEA⁵. That project aimed at defining a proposal for the Ecological Network implementation at the local level in two pilot municipalities near Turin. The proposed approach was aimed at guiding local Public Administrations with measures to limit anthropogenic land use and, where possible, orient and qualify the conservation of ecosystem services.

We point out that describing ENs in informal ways, through linguistic guidelines, is useful for their presentation but challenges their management because it makes it difficult to compare specifications to one another, finding possible contradictions, and to integrate guidelines in order to obtain a single reference document for land planning. In fact, specifications are dispersed in multiple textual sources, that introduce different concepts and metrics to evaluate the properties of an EN.

We thus aim at providing formal approach to the representation of ENs, suitable for integrating heterogeneous information sources, and for automated analysis. For the specification of ENs, we adopt an ontology-based approach, in order to explicitly define the relations between entities and to formally characterize their attributes, supporting automated reasoning. This is in line with consolidated approaches to the representation of geographical knowledge (see the GeoSpatial Semantic WEB [6] and [7, 8]) and provides a basis for a unified treatment of information in Geographical Information Systems (GIS). However, an important aspect of ENs is the specification and reasoning about constraints associated to the guidelines for land planning. Therefore, our work aims at integrating semantic knowledge representation with constraint-based reasoning [9] into a unified framework.

This paper presents an ontology for the description of ENs, and a constraint specification language supporting automatic reasoning about them:

- Our ontology is based on the outcome of the project [5]. We adopted OWL [10], augmented by the GeoSPARQL ontology [11], to describe the main entities and relations of ENs.
- Regarding the representation of EN specifications, we developed a new language, GeCoLan, guided by two main goals:

³ www.cittametropolitana.torino.it/

⁴ www.polito.it

⁵ www.enea.it

- The expression of constraints on the EN domain in a simple and compact way, which simplifies reading and automated management of information.
- The employment of specifications in other automatic tasks, which can't be implemented by simply querying an existing Knowledge Base (KB). For instance, checking the consistency of proposals for changes and additions to an existing EN, or automatically suggesting optimized changes and additions to it.

In line with recent research on geometric and geographic constraint reasoning, e.g., [12, 13], *GeCoLan* abstracts from the details of the reasoning tasks that it supports and can thus be applied to other domains than the one in the focus of this paper.

GeCoLan is suitable to define complex arithmetic and geometric constraints, and supports constraint verification, as well as other rich types of reasoning, by exploiting the SPARQL query language [14] as a lower-level tool for constraint specification.

We developed a *GeCoLan* prototype reasoner and we tested it on a large dataset of geographic information derived from project [5]. Even though we obtained encouraging results regarding the applicability of our approach (see [15]), they were affected by the fact that the dataset included a certain amount of noise (e.g., overlapping geometries of structural elements that should have been disjoint) and extremely detailed geometries, composed of hundreds of vertexes. We thus decided to develop a data pre-processing tool that, before translating a source dataset (e.g., a set of ESRI shapefiles) to semantic data format (RDF [16] triples), pre-processes it by simplifying the geometries and some irregularities of its geographic objects.

In the following we describe the *GeCoLan* language, the prototype reasoner for checking its specifications on an EN, and our data pre-processing tool. The present paper extends the work described in [15] by providing more detail about our work, with examples, and with the description of the data pre-processing tool.

The remainder of this paper is organized as follows: Section 2 provides some background and positions our work in the related literature. Section 3 describes our representation of ENs specifications and provides details about the *GeCoLan* language. Section 4 describes the storage of information about an individual EN and the data pre-processing tool to acquire it. Section 5 presents our approach to the validation of ENs and the tool we developed for that purpose. Section 6 discusses the broader context of our work, outlining our future research directions. Section 7 concludes the paper.

2 Background and Related Work

Our work builds on existing literature about ENs and knowledge representation, and introduces a novel model for representing and managing constraints in ENs. The next subsections provide some background and position our work in the related research.

2.1 Ecological Networks and their Representation

An Ecological Network is an interconnected system consisting of territorial areas that include natural and semi-natural habitats. As reported in [4], "although the way in which the model is elaborated and applied reflects certain conceptual and methodological variants and is subject to local and regional circumstances, ecological networks share two

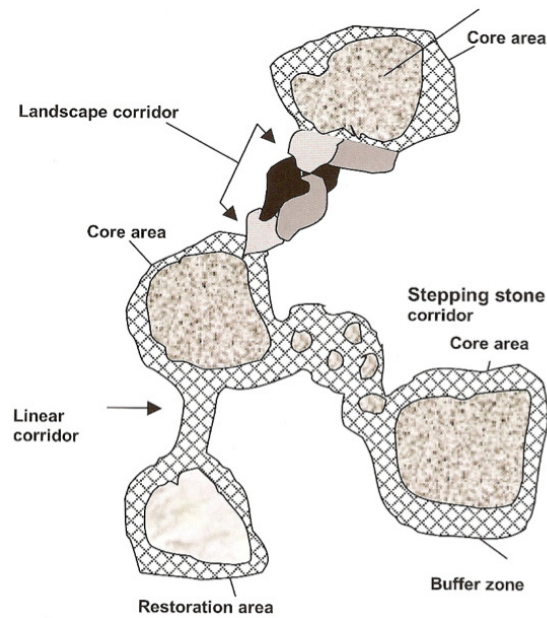


Fig. 1. Diagrammatic representation of the spatial configuration of an Ecological Network, from [4].

generic goals, namely (1) maintaining the functioning of ecosystems as a means of facilitating the conservation of species and habitats and (2) promoting the sustainable use of natural resources in order to reduce the impacts of new urbanizations on biodiversity and/or to increase the biodiversity value of managed landscapes". An EN is represented as a network including various types of elements, described below:

- Structural Elements are the *core areas* of the EN. They denote the areas having a primary ecological function, i.e., the areas with significant “naturalness”, that are relevant to preserve biodiversity. They are represented as nodes of the EN.
- Adjoining Areas, also denoted as *buffers*, are the areas neighboring (≥ 50 meters) the Structural Elements. They represent the areas with the function to safeguard and increase the stability of the core areas.
- Connection Elements, alias *corridors*, are the areas with residual “naturalness” that connect the Structural Elements. They are represented as links of the EN.

Figure 1 depicts the general design of an Ecological Network.

So far, ENs have been largely studied and modeled in order to describe their dynamics. “Ecological network analysis (ENA) is a systems-oriented methodology to analyze within system interactions used to identify holistic properties that are otherwise not evident from the direct observations” [17]. Several mathematical models have been developed to analyze and predict the dynamics of ENs in terms of interactions between organisms in an ecosystem, starting from observational data; e.g., [18, 17, 19–21]. These models are aimed at modeling and simulating the dynamics of the relations among

species, the existence of dynamical bottlenecks in the functioning of the ecosystems, etc., but they do not concern the verification or imposition of urban planning constraints on the ecosystems themselves, which is the objective of our work.

In the context of knowledge engineering, ENs provide a complex, inherently geographic domain that demands an expressive language to be conveniently represented. Moreover, their specifications constitute a challenging problem for representing constraints over complex geographic domains, and evaluating if a given (possibly large) knowledge base satisfies or violates them. To our knowledge, no existing work on ENs focuses on land-usage constraints, and on how the transformations of ENs can be regulated in policy-making, taking such constraints into account.

With the aim of representing knowledge in a formal way that supports the specification of the semantics of concepts, we selected the OWL Web Ontology Language [10] to represent the EN domain, augmented by the GeoSPARQL ontology [11] for representing aspects of geographical knowledge such as topological relations. These languages are well known standards for semantic knowledge representation and they are supported by many tools, including highly scalable and efficient RDF [16] data stores, such as Triple Stores; e.g., Parliament [22]. Being Semantic Web languages, they also seamlessly allow publishing the information on the Web for open access and processing, thus favoring knowledge sharing in the format of Linked Data [23].

2.2 Constraint-based Reasoning on Geographic Information

Several works address the problem of expressing constraints on structured domains, mainly for checking the integrity of data, e.g., [24, 25], and some of them specifically consider knowledge bases with geographic and geometric constraints. [26] introduces the High-Level Constraint Language (HLCL) to define geographic constraints on a conceptual model (i.e., Entity-Relationship) of a database, and proposes to translate them into SQL integrity constraints. Similarly, [13] exploits the Object-Constraint Language (OCL) on UML models. The main difference between the present work and these proposals stems from the fact that we assume that the domain is modeled as an OWL ontology, and that data is stored as RDF triples in a Triple Store.

A way to represent constraints in the Semantic Web, is through *rule languages*, such as SWRL (Semantic Web Rule Language) [27] and RuleML [28]. While the original expressiveness of those languages was not general enough to support constructs such as logic quantifiers, negation, disjunction, numeric and geometric expressions (all of which typically occur in EN specifications), subsequent extensions empowered them with the full expressiveness of First-Order Logic (FOL) [29, 30]. Moreover, some work integrated SWRL with mathematical, geometrical, and other types of functions through built-ins; e.g., [31].

We introduce a new language, instead of extending an existing rule-based one, because of several syntactic aspects that make it particularly succinct and convenient for our needs. However, we don't exclude translating our language to a standard rule-based one in our future research, in order to exploit existing inference engines for those languages. Our language is named *GeCoLan* to recall *CoLan* [25], on which it is strongly based, with restrictions and extensions. *CoLan* is a language initially proposed for expressing integrity constraints in Object-Oriented Databases, and later used as the basis

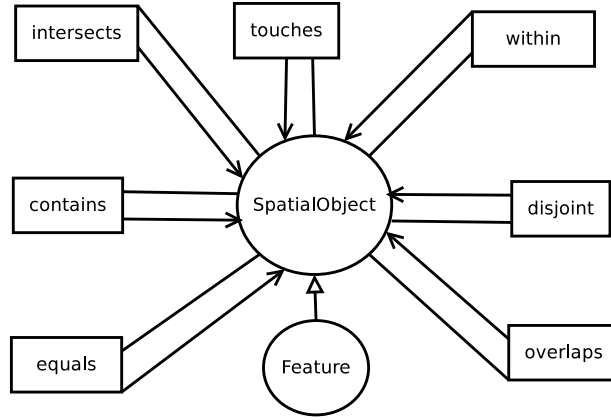


Fig. 2. A fragment of the GeoSPARQL ontology, from [15].

for the Constraint Interchange Format (CIF), an RDF schema for exchanging constraints on the Semantic Web [32].

The SPARQL [14, 33] language has been used for expressing and checking constraints over OWL-based Triple Stores [34]. However, the direct adoption of SPARQL (or of other query languages) for the description of EN specifications fails to support reasoning tasks that cannot be implemented by simply querying an existing KB; e.g., suggesting changes to repair a constraint violation, possibly optimizing some desired measure. While our current work focuses on constraint verification, we introduce GeCoLan to provide a unified support to other pro-active tasks, such as suggesting solutions to violation problems and exploring expansion possibilities for ENs.

3 Conceptual Representation of Ecological Networks

In the following we describe the OWL ontology we defined to represent the main concepts and relations concerning ENs, and the GeCoLan constraint specification language.

3.1 The Ecological Network Ontology

For the representation of Ecological Networks, we selected the OWL2 [35] language, which offers:

- IS-A relations to define *taxonomies* of concepts; e.g., hierarchies of land use types, of planned interventions, and so forth.
- Support to modeling and performing calculations with geographic/geometric shapes of concepts; e.g., elements of the EN and elements of land use maps.
- Specification of restrictions on concepts and on their relations. This is useful to formally represent definitions as the following one: “coppicing is a maintenance intervention that only applies to wooden areas”.

Moreover, we imported the GeoSPARQL ontology [22, 36], which defines geographic entities and relations:

- GeoSPARQL defines a *Feature* as a *SpatialObject* that has a *Geometry* on the 2D plane; a *Feature* can be either a *Point*, a *Curve* (in particular, a piecewise linear curve), or a *Polygon*.
- Moreover, GeoSPARQL defines a number of topologic geometric relations between *Features* that correspond to the basic relations identified in the literature about Geographic Information Systems: namely, the Simple Features relations [37], and the equivalent RCC8 [38], and Egenhofer relations [39]. Figure 2 shows the Simple Features topological properties, which relate objects of the *SpatialObject* class. For instance, the *touches* relation between two spatial objects *A* and *B* holds whenever the borders (but not the interiors) of *A* and *B* have a non-empty intersection. Moreover, the *overlaps* represents a non null intersection (but not an inclusion) between the areas of two spatial objects.

The EN ontology is defined around the *Feature* class. By specializing *Feature*, we define the four hierarchies of concepts representing the core of the EN domain:

- *EcologicalNetworkElement*: this is an element of the EN, i.e., either a Structural Element, or a Priority Expansion Element.⁶ The latter is specialized to a Connection Element or Adjoining Area.
- *LandUseElement* represents a Land Use Element (LUE), i.e., a parcel of land defined in the Land Cover Piemonte (LCP) cartography [40], characterized by a specific type of land use; see below. The LCP structures Land Use Elements in 4 specification levels:
 - At level 1 the LCP defines 5 general types of land use: *WaterBody*, *WetLand*, *WoodenLand*, *AgriculturalLand*, and *ArtificialLand*, which we explicitly model as concepts of the EN ontology.
 - Moreover, the LCP includes 15 types of land use at level 2 of the taxonomy, 45 at level 3, and 97 at level 4. We specify the values of levels 2, 3, and 4 as properties of *LandUseElement*: *LCPllevel2*, *LCPllevel3*, and *LCPllevel4*.

Each *LandUseElement* is characterized by five evaluation criteria, which take values in [1, 5], with 1 representing the maximum value and 5 the lowest one:

1. *naturality*: how close the element is to a natural environment;
 2. *relevance*: how relevant it is for the conservation of the habitat;
 3. *fragility*: how fragile it is with respect to the anthropogenic pressure;
 4. *extroversion*: how much pressure it can exert on the neighboring areas;
 5. *irreversibility*: how difficult it would be to change its use.
- *Intervention* represents an intervention for building, improving or conserving the EN.
 - *Operation* represents a specific operation of elimination, construction or maintenance that is part of an intervention.

⁶ The Priority Expansion Elements are the areas with residual ecological functionality where the priority is intervening to increase the functionality of the primary ecological network and implementing conservation measures.

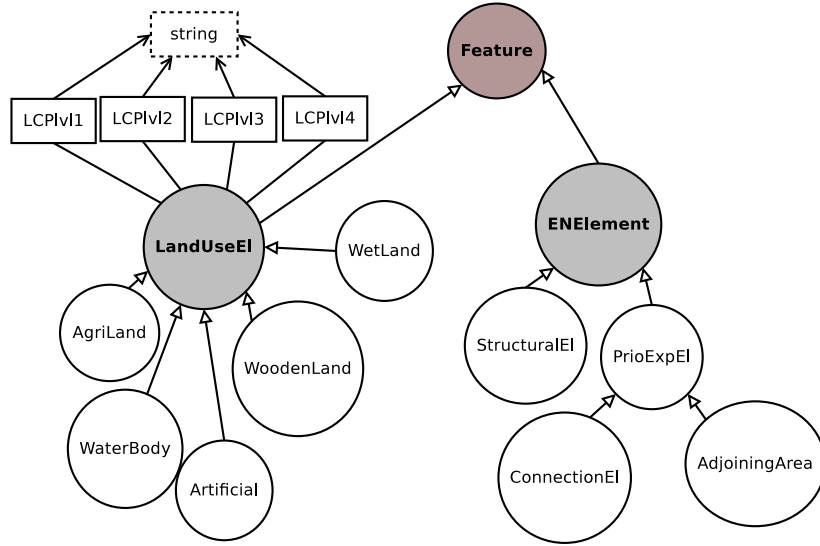


Fig. 3. A fragment of the EN ontology with EN elements and Land Use Elements.

Figure 3 shows a portion of our EN ontology, with the *Feature* class depicted in dark grey and the roots of the *EcologicalNetworkElement* and *LandUseElement* in light grey. All of the concepts of the ontology inherit the topological relations defined by *Feature*. Thus, e.g., an individual *ConnectionElement* may *touch* a number of *LandUseElements*.

Figure 4 shows another portion of our EN ontology with the roots of the *Interventions* and *Operations* hierarchies in light grey.

In order to characterize the classes and properties used in our ontology, we exploit the powerful constructs of the OWL2 language (in particular, Class Expressions) in several places. For example:

- the *hasPlants* property applies to LUEs of types *WoodenLand* and *AgriLand*. The domain of *hasPlants* is therefore the following OWL Class Expression (in functional syntax):

ObjectUnionOf(WoodenLand AgriLand)

- the *hasArea* property applies to *Features* whose geometry is a *Surface*. The domain of *hasArea* is thus:

AllValuesFrom(hasGeometry Surface)

- the *Coppicing* class (subclass of *Maintenance* operation) should have a linear geometry. Thus, Class *Coppicing* is also a subclass of the following Class Expression:

AllValuesFrom(hasGeometry LineString)

- the *Eradication* class (subclass of *Elimination* operation) should only *intersect* LUEs of type *WoodenLands*; moreover, such elements should have an *LCPIlevel2*

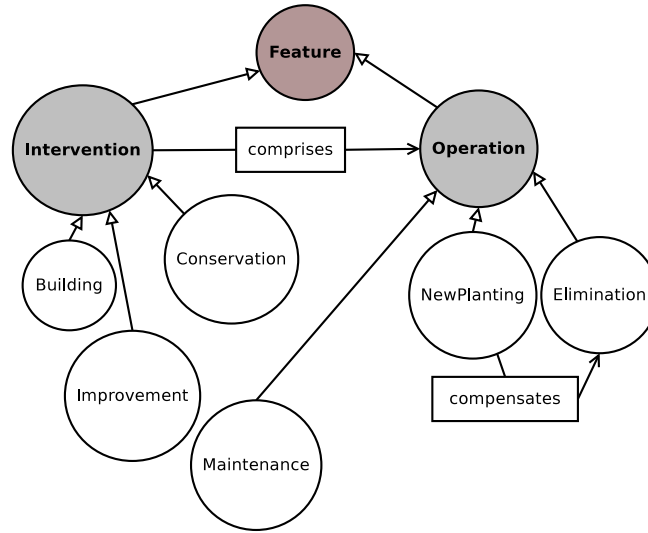


Fig. 4. Another fragment of the EN ontology with interventions and operations.

value of either “ShrubbyGrassyZone” or “OpenZone”. Class *Eradication* is therefore also a subclass of the following Class Expression:

```

AllValuesFrom(sfIntersects
  (ObjectIntersectionOf(WoodenLand
    (DataAllValuesFrom(LCPlevel2
      DataOneOf("ShrubbyGrassyZone" "OpenZone")))))

```

It should be noticed that the above described specifications of ENs represent *integrity constraints* on the classes and properties defined by the ontology; they describe intrinsic properties of the EN. For instance, it would be meaningless to have an element of the *Coppicing* class with a non-linear geometry, or an *Eradication* operation over an urban area. Differently, the EN specifications described in the following section are meant to constrain the way we are *allowed to plan* an Ecological Network, and may possibly be violated by existing ENs that have not been correctly planned. This is an important reason to encode the two types of constraints differently, besides the fact that OWL has limits on the logic constraints that it can express [41].

3.2 Constraints on EN Transformations

For the specification of constraints on the transformations of ENs we defined GeCoLan, having in mind the following requirements: first, the language must use the terms defined in the EN ontology (classes and properties) as its main vocabulary. Second, it must support the expression of *constraints* about the objects of the world modeled by the same ontology. Third, it must allow mixing logic, geometric, numeric, and time requirements into the constraints.

We named our language GeCoLan because it is intended to capture geometric and geographic constraints, beside more traditional types of constraints captured by the CoLan language [25] from which it derives. The terms of our language have one of the following types:

- literal values; e.g., numbers, strings, etc.;
- variables;
- function calls whose arguments are themselves terms.

GeCoLan offers the `and`, `or`, `not`, and `implies` connectives of First Order Logic to compose conditions. Moreover, the atoms can be:

- *property* predicates $p(t_1, t_2)$, where p is the name of a property defined in the EN ontology, t_1 is a term representing an individual in the domain of p , and t_2 is a term that denotes an individual in the range of p . For example, given two variables $g1$ and $g2$, each one taking the value of a Geometry, `sfTouches(g1,g2)` states that the value of $g1$ touches the value of $g2$.
- *filter expressions* $\{e\}$ ⁷; a filter expression e , enclosed in curly brackets, can use the logic connectives `&&` (and), `||` (or), and `!` (not) between conditions. In turn, conditions are defined using relational operators `>`, `<`, `<=`, `>=`, `=`, `!=`. The operands of the relational operators are terms or numeric expressions obtained by combining numeric operands with operators `+`, `-`, `/`, `*`.

Compared to rule-based languages such as SWRL and RuleML (even in their extensions with First Order Logic), GeCoLan has a number of syntactic features that make it particularly convenient for our purposes:

- it allows the use of nested functions that correspond either to functions defined in (Geo)SPARQL, or to functional properties of the ontology classes. For example,

`LCPllevel2(e1)`

denotes the value of the functional property *LCPllevel2* of variable *e1*, while:

`distance(intersection(l1,l2), intersection(l3,l4))`

denotes the distance between the intersection area of *l1* and *l2*, and the intersection area of *l3* and *l4*. The *l1*, ..., *l4* variables are *Well Known Text* (WKT) literals, i.e., serializations of geometries.

- GeCoLan also allows the use of infix relational and arithmetic operators in expressions over complex terms. For example:

`{distance(l1,l2) >= 2*distance(l1,l3)}`

means that the distance between the WKT literal values of variables *l1* and *l2* is at least twice as the distance between those of *l1* and *l3*. Similarly,

`{irreversibility(e1) < 3 || extroversion(e1) < 3}`

imposes that the values of functional properties *irreversibility* and *extroversion* of variable *e1* are less than 3.

⁷ Filter expressions correspond to whole constraints to be evaluated in the *FILTER* clause of SPARQL; see Section 5.1.

- moreover, it allows the quantifiers `forall` and `exists` with *range restricted* variables:

$$\begin{aligned} &\text{forall } C(x)[\text{s.t. } \psi(x)]\phi(x) \\ &\text{exists } C(x)[\text{s.t. } \psi(x)]\phi(x) \end{aligned}$$

where C is an ontology concept, x is a variable that represents an individual, and $\psi(x)$ is an optional formula that further restricts the individuals of C . For example, “forall `NewPlanting(np)`” states that the value of variable np can be any instance of class *NewPlanting*. If we add a condition “s.t. `compensates(np,el)`”, only individuals np that compensate some given elimination el will be considered.

As we aim at expressing constraints that can be either satisfied or violated by the knowledge base, we are only interested in closed formulas, in which all the variables are within the scope of a corresponding quantifier.

3.3 Example Specifications

In the following, we describe the representation of three sample specifications taken from the guidelines for the Local EN implementation devised in project [5].

Specification 1: Connection Elements. *Connection elements must avoid areas with maximum irreversibility and areas with maximum extroversion.*

This constraint is satisfied if a *ConnectionElement* does not *overlap* with any *LandUseElements* whose *irreversibility* or *extroversion* properties have the maximum value, i.e., a value > 1 . The specification can be encoded as follows:

```
forall ConnectionElement(ce)
forall LandUseElement(lue) such that sfOverlaps(ce,lue)
{irreversibility(lue) > 1 && extroversion(lue) > 1}
```

The first universal quantifier restricts the range of variable ce , that must be a *ConnectionElement*. The second one restricts variable lue , that must be a *LandUseElement*, and must have a geometry G^8 that *overlaps* with the geometry of ce .⁹

The filter expression in curly brackets imposes that both the *irreversibility* and the *extroversion* of lue have a value > 1 . In the example we use several features of our ontology and language, e.g.:

- the geometric capabilities of GeoSPARQL enable to automatically find the *LandUseElements* that fall at least partially within a *ConnectionElement*;
- the filter expressions provide the ability to perform numeric comparisons between the values of some functional properties of *LandUseElements* and a numeric constant.

⁸ We write G for short to denote functional property *hasGeometry* that applies to both *ConnectionElements* and *LandUseElements*.

⁹ In GeoSPARQL, the topological relations such as *sfOverlaps* are computed on the geometries associated with the individuals in class *Feature* (lue and ce in our example) through the *defaultGeometry* property.



Fig. 5. Graphical representation of the two cases of the specification for Buffer Zones. The structural elements of the EN are colored in dark green, the Land Use Elements in light green are meadows (with low extroversion) while the LUEs in yellow are crops (with high extroversion). Buffers are depicted as shadowed areas.

The second example constraints the creation of Buffer Zones to protect the elements of the EN:

Specification 2: Buffer Zones. *The creation of protection buffers is done, whenever possible, through interventions of restoration in areas surrounding the structural elements of the EN, with the goal of enhancing and protecting them. In case a structural element is surrounded by areas with maximum extroversion and/or maximum irreversibility, the buffers must be realized within the structural element (due to the impossibility of extending the element itself).*

Using the vocabulary of the EN ontology, a *BufferCreation* is an *Intervention of Conservation* that *touches* a *StructuralElement* of the EN, except when it would *overlap* with one or more *LandUseElements* with maximum *irreversibility* or *extroversion*. In that case, the *BufferCreation* should be *within* the *StructuralElement*. Figure 5 shows the two cases: the lower buffer is within a *StructuralElement*, since otherwise it would overlap with a crop area of maximum extroversion. Differently, the upper buffer touches the *StructuralElement*, as it overlaps with a meadow of limited extroversion.

This specification can be encoded as follows:

```
forall BufferCreation(bc)
forall StructuralElement(sel)
forall LandUseElement(lue) such that (not sfWithin(lue,sel))
  ((sfTouches(bc,sel) and sfOverlaps(bc,lue))
    implies
      {(irreversibility(lue) > 1) and (extroversion(lue) > 1)})
  and
  ((sfWithin(bc,sel) and sfTouches(bc,lue))
    implies
      {(irreversibility(lue) = 1) or (extroversion(lue) = 1)})
```

The formula starts with universal quantifiers specifying the classes of variables *bc*, *sel*, and *lue*. The quantification of *lue* further restricts the attention to *LandUseElements* whose geometry is not *within* the geometry of *StructuralElement sel*. The body is the conjunction of two sub-specifications, corresponding to the two cases of Figure 5: if (the geometry of) *bc* *touches sel*, then each *lue* that overlaps with *bc* must have non-maximum irreversibility and extroversion. Otherwise, if *bc* is *within sel*, then each *lue* touched by *bc* must have maximum irreversibility and/or extroversion.

The third example constrains eradication operations:

Specification 3: Eradication. *Eradication in shrubby-grassy zones and open zones is generally forbidden. In case the eradication is unavoidable, it requires an authorization by the Municipality, and should be compensated by a new planting with an area at least twice as the eradicated area. For the new plantation, autochthonous plants and trees should be employed.*

The norm says that an *Eradication* is an *Operation* that must belong to an *Intervention of Improvement* that *isAuthorized*, and also *comprises* a *NewPlanting* operation that *compensates* the *Eradication*. The planting *hasArea* at least twice as that of the eradication. Finally, the *hasPlant* property of the *NewPlanting* must not contain any plant in the black-list **BL**: this is an individual of type *PlantsList* containing the forbidden exotic species. The norm can be encoded as follows:

```
forall Eradication(er)
exists NewPlanting(np) such that compensates(np,er)
forall Plant(pl) such that hasPlant(np,pl)
(isAuthorized(interventionOf(er))
implies
((not(hasPlant(BL,pl))) and {(area(np) ≥ 2*area(er))}))
```

The formula mixes universal and existential quantifiers, in order to impose a combined restriction on each eradication *er*, the new planting *np* that compensates it (*np* must exist), and all the plants *pl* selected for *np*. The restriction applies only if the *Intervention* including *er* (obtained with functional property *interventionOf*) is authorized. It states that *pl* must not belong to **BL** and that the area of *np* must be at least twice as the area of *er*.

4 Acquisition and Storage of Geographic Information about ENs

4.1 Storage of the Information about an Individual EN

The ontological definition of ENs described in Section 3 supports the representation in RDF format [16] of data about an individual EN; i.e., which are the structural elements of a geographical area.

RDF is standard for knowledge representation in the Semantic Web and prescribes that each piece of information is represented as a <subject, predicate, object> triple. For

instance, a specific Land Use Element is represented as follows:

<i>lueid</i> rdf:type	onto:LandUseElement
<i>lueid</i> rdf:type	geo:Feature
<i>lueid</i> geo:hasGeometry	<i>geoid</i>
<i>lueid</i> onto:area	1152.000000015134
<i>lueid</i> onto:perimeter	191.99999999289867
<i>lueid</i> onto:LCLevel1	"2"
<i>lueid</i> onto:LCLevel2	"2"
<i>lueid</i> onto:LCLevel3	"2"
<i>lueid</i> onto:LCLevel4	"2"
<i>lueid</i> onto:extroversion	2
<i>lueid</i> onto:irreversibility	4
<i>lueid</i> onto:fragility	4
<i>lueid</i> onto:naturality	2
<i>lueid</i> onto:relevance	3

where *lueid* is the unique URI of the Land Use Element, and *geoid* is the unique URI of its associated geometry. Note that the second triple, stating that *lueid* is a *Feature*, is automatically inferred by the Triple Store. This derives from the encoding of the ontology within the Triple Store itself, in particular from the following triple:

```
onto:LandUseElement rdfs:subClassOf geo:Feature
```

Being a standard, RDF is coupled with several tools for data interpretation and management. For instance, various Triple Stores are available to store and manage RDF triples in an efficient way, by answering SPARQL [14], and possibly GeoSPARQL [11], queries. In our experiments, we used the Parliament Triple Store [22] to memorize the data about the EN developed in project [5].

4.2 A Tool for Pre-processing and Acquiring Information about an EN

The process of acquisition of new geometries into the Triple Store is more complex than a simple mapping from their coordinates and (meta) attributes to a set of corresponding RDF triples. With the aim of creating an efficient and robust knowledge base, we introduce a preliminary phase of input processing within which we apply transformations to the geometries and synthesize new attributes from the attributes of the geometry and possibly from other knowledge already contained in the Triple Store.

The geometric transformations can be different depending on the quality and characteristics of the input data. One of such transformations consists in the simplification of a geometry obtained by reducing its number of vertexes. This reduction can yield significant computational benefits, because the time complexity of the algorithms behind the GeoSPARQL functions often directly depends on such a number. Clearly, reducing the vertexes leads to a distortion of the original geometries, and it is important to seek a trade-off between loss of precision and computational gains. Many algorithms for polygons simplification can be found in the literature, with different levels of efficiency and quality of the output; among these, we selected the very well known Douglas-Peucker algorithm [42]. This algorithm takes as input a polyline (in our case, a closed one) and a tolerance distance ϵ , and returns a similar polyline with a subset of the original points.

Similarity means that the Hausdorff distance between the original and the computed polylines is less than the given tolerance. Suppose that we have a polyline described as a list of points; then, the algorithm recursively divides the list in two sublists choosing the split point such that it maximizes the distance d with the segment joining the end points of the list. If the distance d satisfies $d \leq \epsilon$, we can replace the polyline directly with that segment, otherwise the split point p is marked as a point to keep, and the split is recursively invoked on the two sublists determined by p .

Another useful pre-processing operation consists in filtering out all the polygons whose area is below a threshold. This can be done by computing the area of each input geometry with the help of a computational geometry library, skipping all the geometries that are too small for being of interest. When the number of such geometries is large, significant benefits can be obtained in terms of efficiency in the use of the system.

In order to get reliable results during the computation of the areas, in a desired unit of measure (e.g., square meters), it is necessary that the Coordinate Reference System (CRS) used to describe the input geometries uses the same unit of measure; and, more importantly, it preserves the measure of areas when they are projected from the Earth surface to a Euclidean plane. Indeed, some projections preserve the topology of the area without giving any guarantee about the accuracy of their areas (e.g WGS84 projection); and vice-versa (e.g UTM* projections). This is just an example of the need to convert the input geometries from one CRS to another, for specific computations. As these operations are not supported by GeoSPARQL, it is necessary to pre-calculate them and possibly store them for later on-line use.

5 Validation of Ecological Networks

Intuitively, the validation of an individual EN against a set of constraints means analyzing the Triple Store describing the EN (i.e., the knowledge base) and checking that no RDF triple violates them. For instance, let us consider the task of checking the **Connection Elements** specification of Section 3.3. That means considering each *ConnectionElement* ce in the Triple Store, and verifying that all of the *LandUseElements* overlapping with ce satisfy the given restrictions on the *extroversion* and *irreversibility* properties. We assume that all of the relevant facts are present in the store (*closed world* semantics). Therefore, if a required fact is missing, a violation is detected.

5.1 Translating Specifications to GeoSPARQL Queries

In order to check whether the EN specifications are satisfied by an existing knowledge base, we translate them to equivalent queries that can be executed by any engine that supports the GeoSPARQL standard. This approach enables us to exploit the engine to perform the reasoning tasks, instead of developing a GeCoLan interpreter from scratch. Given a closed GeCoLan formula ϕ , we translate it to a GeoSPARQL query q_ϕ such that:

$$KB \models \phi \leftrightarrow \begin{cases} [q_\phi]^{KB} \neq \emptyset & \phi = \text{exists } C(x)\phi(x) \\ [q_\phi]^{KB} = \emptyset & \phi = \text{not exists } C(x)\phi(x) \end{cases} \quad (1)$$

where ϕ is in one of the above forms, KB is the Triple Store against which we want to evaluate ϕ , and $[q_\phi]^{KB}$ denotes the result of applying q_ϕ to KB .

Normalization of GeCoLan constraints. Similar to [43, 44], we define the translation inductively. Firstly, we consider some rewritings of ϕ into equivalent GeCoLan formulas to reduce the number of cases we have to explicitly translate:

$\text{forall } C(x) \text{ s.t. } \psi (\phi)$	$\mathcal{T}(\text{forall } x$ $((C(x) \text{ and } \psi) \text{ implies } \phi))$
$\text{forall } C(x) \phi$	$\mathcal{T}(\text{not exists } C(x) (\text{not } \phi))$
$\text{exists } C(x) \text{ s.t. } \psi (\phi)$	$\mathcal{T}(\text{exists } x$ $(C(x) \text{ and } \psi \text{ and } \phi))$
$\phi \text{ implies } \psi$	$\mathcal{T}(\text{not } \phi \text{ or } \psi)$

After these rewritings, formula ϕ starts either with the “not exists $C(x)$ ”, or with the “exists $C(x)$ ” prefix. It does not include any occurrences of the forall quantifier or of the implies connective.

Translation of normalized formulas to GeoSPARQL queries. Before starting the translation of the normalized formulas to GeoSPARQL queries, we remove the prefix from ϕ , so that the values returned by the query are not completely projected out; see the example below. At this point, by translating ϕ to q_ϕ according to Equation (1) and by applying q_ϕ to KB , we can check whether KB satisfies ϕ or not.

For finite domain classes C_1, \dots, C_k , we define the cross product of their domains as follows:

$$ADom(x_1, \dots, x_k) = \{x_1 \text{ rdf : type } C_1. \\ \dots \\ x_k \text{ rdf : type } C_k.\}$$

We then define the following translations:

$\text{exists } x \phi$	$\text{SELECT } (free(\phi) - \{x\}) \text{ WHERE } \{\mathcal{T}(\phi)\}$
$\phi \text{ or } \psi$	$\mathcal{T}(\phi) \text{ UNION } \mathcal{T}(\psi)$
$\phi \text{ and } \psi$	$\{\mathcal{T}(\phi) \mathcal{T}(\psi)\}$
$\neg \phi$	$\{ADom(free(\phi)) \text{ FILTER NOT EXISTS } \mathcal{T}(\phi)\}$

The translation of exists involves the projection of variable x through a subquery, as x should no longer appear in outer formulas. The translations of or and and connectors are mapped, respectively, to the UNION clause of SPARQL and to a sequence of graph patterns. Finally, the negation of ϕ maps to a FILTER NOT EXISTS clause between the domains of the free variables and their assignments satisfying ϕ .

The atoms of GeCoLan formulas are translated as follows:

$C(x)$	$\{x \text{ rdf : type } C.\}$
$p(t_1, t_2)$	$\{\mathcal{T}(t_1) \mathcal{T}(t_2) x_1 p x_2.\}$
$\{e(t_1, \dots, t_n)\}$	$\mathcal{T}(t_1) \dots \mathcal{T}(t_n) \text{ FILTER } e(x_1, \dots, x_n)$

- The class C of a variable x is translated as a graph pattern with predicate rdf : type .
- Let us consider the translation of $p(t_1, t_2)$. First of all, we need to translate the terms t_1, t_2 . For example, let the property atom be:

`member(hasPlants(lue), "Rudbekia")`

where t_1 is *hasPlants(lue)*, i.e., the application of a functional property *hasPlants* to a *LandUseElement lue*; the *hasPlants* property maps *lue* to a *Collection* element

col. We need to introduce a new variable x_1 for representing the value of such property, and add the pattern $\{lue \text{ hasPlants } x_1\}$ to the translation. Finally, we add the pattern $\{x_1 \text{ member "Rudbekia"}\}$ to state that Rudbekia is a member of the x_1 collection.

- The translation of filter expressions requires a similar approach to generate variables corresponding to applications of functional properties.

5.2 Example

We will now apply the above described translation method to the **Connection Elements** example of Section 3.3:

```
forall ConnectionElement(ce)
forall LandUseElement(lue) such that sfOverlaps(ce,lue)
  {irreversibility(lue) > 1 && extroversion(lue) > 1}
```

First, we remove the forall quantifiers:

```
not exists ConnectionElement(ce)
exists LandUseElement(lue)
  not (sfOverlaps(ce,lue) implies
    {irreversibility(lue) > 1 && extroversion(lue) > 1})
```

Then, we remove the implies connective:

```
not exists ConnectionElement(ce)
exists LandUseElement(lue)
  (sfOverlaps(ce,lue) and
    not{irreversibility(lue) > 1 && extroversion(lue) > 1})
```

Let us consider the translation of the filter expression:

$$\tau_0 = \{ \{ \text{lue irreversibility ir.} \} \\ \{ \text{lue extroversion ex.} \} \\ \text{FILTER } ((\text{ir} > 1) \ \&\& \ (\text{ex} > 1)) \}$$

Two new variables (named *ir* and *ex* for readability) have been added to hold the values of the *irreversibility* and *extroversion* properties of *lue*. Then, the filter expression (in which the property applications have been replaced with *ir* and *ex*) becomes the argument of a FILTER clause. The range expression for *ce*, *lue* is:

$$\tau_R = \{ \text{ce rdf : type ConnectionElement.} \\ \text{lue rdf : type LandUseElement.} \}$$

Therefore, the translation of the negation (not) of the filter expression is:

$$\tau_1 = \{ \tau_R \text{ FILTER NOT EXISTS } \tau_0 \}$$

By applying the translation rules for the property predicates and conjunctions, we obtain the following translation for the body of the formula:

$$\tau_2 = \{ \{ \text{ce sfOverlaps lue.} \} \tau_1 \}$$

Finally, we translate the existential quantifier on *lue* and class restrictions on *lue*, *ce* using the associated rules:

$$\tau_3 = \text{SELECT } (ce) \text{ WHERE } \{\tau_R \tau_2\}$$

As already mentioned, we do not translate the first existential quantifier on *ce*, because this would leave no variables.

According to Equation (1), given that the original formula started with `not exists`, it is satisfied if query τ_3 returns \emptyset .

5.3 A Tool for the Automated Translation and Execution of GeCoLan Formulas

We developed a translator from GeCoLan formulas to GeoSPARQL queries based on the ANTLR parser generator¹⁰, which we selected because of its power and ease of use. Based on the formal specification of the GeCoLan grammar, ANTLR generated a generic parser based on the Visitor design pattern [45]. We then extended this parser to produce the correct GeoSPARQL fragments according to the grammar rules.

After translating a formula to GeoSPARQL, our tool submits it to the Parliament Triple Store through the Jena¹¹ library for execution. The result returned by the Triple Store is analyzed to determine whether the data satisfies the EN specification or not.

5.4 Experiment

We tested our translator on a portion of the dataset produced in project [5]. That dataset consisted of a set of ESRI shapefiles. We converted them to RDF using our pre-processing tool, which allows the user to define an appropriate mapping in order to associate shapefiles attributes to RDF properties.

Figure 6 shows a portion of the map we used. In the figure, land use elements are colored according to their first level Land Cover Piemonte type; e.g., artificial land is gray and water bodies are light blue.

We uploaded to Parliament the data located within a circle of 10km diameter around a city in the neighborhood of Turin. Overall, the data consisted of 183,752 triples describing 5,162 *LandUseElements*, over which lay the elements of the EN: i.e., 579 *StructuralElements* and 1,054 *ConnectionElements*.

We first tested the **Connection Elements** example from Section 3.3. The query is challenging because it could require to consider all of the pairs formed by a *LandUseElement* and a *ConnectionElement*. Overall, the Triple Store contains almost 5.5M such pairs, and complex operations have to be performed on each one, such as checking whether the *LandUseElement* and *ConnectionElement* geometries overlap. The execution of the query took about 5.5 minutes on a medium-end laptop, pointing out that 595 *ConnectionElements* of the EN intersected at least a *LandUseElement* of maximum irreversibility or extroversion. Thus, in the project data, more than 55% of the *ConnectionElements* (595 out of 1,054) violate the EN specification.

¹⁰ www.antlr.org/

¹¹ <https://jena.apache.org/>

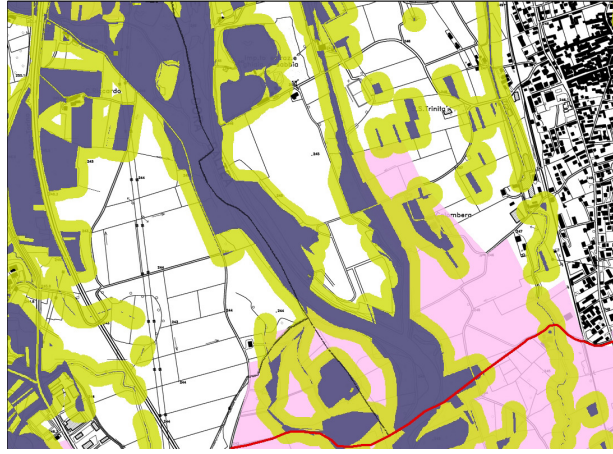


Fig. 6. A portion of the map of the area covered by the Local EN proposal.

We then tested the **Buffer Zones** example. The translation of the specification yields a GeoSPARQL query with four (partially nested) SELECTs and 44 lines. However, the execution took only a few seconds before answering that there are no violating *Buffer Creations* in our data. We explain the efficiency of this query with the fact that the number of pending *Interventions* of type *Buffer Creation* is usually small at a given time. For instance, in our dataset, there was only one intervention. The number of *LandUseElements* and *StructuralElements* that must be considered by the query is therefore strongly limited by the fact that they must be “close” to the *Buffer Creations*; i.e., they touch, or overlap with them.

5.5 Beyond Constraint Verification

A natural extension of the EN validation task, which requires the full power of a language like GeCoLan, is the suggestion of how to fix the violations detected in a knowledge base. For example, it is quite plausible that the check of the **Connection Elements** specification may return a list of pairs (ce, lue) that indicate which *LandUseElements* *lue* overlapping with some *ConnectionElements* *ce* have invalid *extroversion* and/or *irreversibility*. Different actions could be identified to solve the problem; e.g., the *ConnectionElement* *ce* might be removed from the KB, or the geometry of *ce* could be reduced so that it no longer overlaps with the “bad” *LandUseElement* *lue*. However, in both cases, the proposed changes might generate new inconsistencies, caused by the fact that some elements of the EN become disconnected. Thus, in general, the identification of hypotheses of transformation of an EN have to be tightly integrated with the verification of their eligibility.

An even more challenging task would be to ask an automated reasoner to suggest how to connect a new *StructuralElement* *sel* to the rest of the EN, by proposing a path of new *ConnectionElements* that lead from *sel* to an existing element of the EN.

The **Connection Elements** specification is relevant to all of the above tasks, but clearly not all of them can be solved by querying the KB through a language like SPARQL. Some tasks may require reasoning engines such as, e.g., Prolog [46], Answer Set Programming [47], and Constraint Programming [48], or even specialized libraries for 2D computational geometry [49, 50]. For example, suppose that some EN specifications were translated to a Constraint Satisfaction Problem (CSP, [48]) for an EN containing a pair (ce, lue) which violates **Connection Elements**. If the following facts were retracted from the CSP:

```

ConnectionElement ce.
overlaps(ce,lue).
irreversibility(lue) = 1.
extroversion(lue) = 1.

```

a CSP solver might be able to suggest a change to the type of ce , and/or to its overlapping with lue , and/or to the attributes of lue , that satisfies the **Connection Elements** specification while preserving the connection of the EN elements (if this latter constraint is encoded in the CSP).

Of course, this is just an illustrative example. However, the point that we want to stress here, is that by adopting a higher-level language such as GeCoLan to encode the specifications, we can then (re-)use them as inputs to an appropriate combination of reasoners in order to solve more complex tasks.

6 Future Work

The study and implementation of further reasoning tasks based on GeCoLan represents one of the most important directions of our future work, in order to enable the interactive verification of the applicability of transformation proposals in a crowdsourcing context.

Whereas, at the current stage, we implemented EN validation as a stand-alone prototype, we aim at integrating it with a Participatory Geographical Information System (PGIS) to support online interaction with stakeholders in inclusive processes aimed at collecting feedback and project proposals for redesigning a geographical area. This would be a novel feature, as current Collaborative Web GIS (e.g., OnToMap [51–53], Ushahidi [54], PlanYourPlace[55] and other Collaborative Web GIS [56, 57]) support a free introduction of geographic information, that has to be separately checked to evaluate the consistency and acceptability of user contributions. Basically, these applications only support feedback collection, and fail to provide validation functions to check the feasibility of the proposed actions.

In order to facilitate the convergence towards mutually agreed and feasible plans, it might be interesting to know how far the constraints on land usage are satisfied in a certain area, or whether any hypothetical actions, e.g., related to redevelopment project plans, are compatible with them. The present work is suitable for answering this kind of question by proposing a model for verifying the compliance of a set of geo-data with specifications concerning the related geographical area. Although in this paper we focus on the preservation and reconstruction of Ecological Networks, our model can be applied in rather different contexts, including the management of city plans, where

detailed constraints have to be satisfied when building new elements or remodeling existing ones.

Another line of research we will pursue the improvement of the efficiency of validity checks. For instance, we plan to cache some types of pre-computed information (e.g., the intersections between elements of the EN), and to introduce an automatic optimization of the translations of GeCoLan formulas, in order to better exploit the operational semantics of GeoSPARQL and of its implementations.

7 Conclusions

This paper described an OWL ontology for the representation of Ecological Networks (ENs), and a semantic language (GeCoLan) for the specification and verification of EN transformation proposals. The language is aimed at enabling an automatic check of the consistency of transformation actions with the defined land usage restrictions, as well as at identifying existing inconsistencies in the ENs. Moreover, the language is sufficiently expressive to support other reasoning tasks, such as the generation of proposals for amending inconsistencies in ENs, and the generation of optimized solution proposals.

We developed a prototype reasoner for the automatic validation of OWL-based representations of ENs: the reasoner automatically translates GeCoLan formulas to GeoSPARQL queries to efficiently implement the validation checks. A first test on the data collected in project [5] provided encouraging results; however, we observed that the noise present in the data affected performance. In order to support an efficient execution of reasoning tasks, we thus developed a data pre-processing tool that simplifies geographic datasets before translating them to the RDF representation needed for semantic reasoning. This paper describes both tools and their application to the dataset of project [5].

8 Acknowledgements

This work is partially funded by project MIMOSA (MultiModal Ontology-driven query system for the heterogeneous data of a Smartcity, “Progetto di Ateneo Torino_call2014_L2_157”, 2015-17), and by “Ricerca Locale” and “Ricerca Autofinanziata” of the University of Torino.

References

1. Jongman, R.: Nature conservation planning in europe: developing ecological networks. *Landscape and urban planning* **32**(3) (1995) 169–183
2. Council of Europe: General guidelines for the development of the pan-european ecological network. *Nature and environment* **107** (2000)
3. Bennett, G., Wit, P.: The development and application of ecological networks: a review of proposals, plans and programmes. *AIDEnvironment* (2001)
4. Bennett, G., Mulongoy, K.: Review of experience with ecological networks, corridors and buffer zones. *Technical Series* **23** (2006)

5. Città Metropolitana di Torino: Misura 323 del psr 2007-2013 (in italian). <http://www.cittametropolitana.torino.it/cms/territorio-urbanistica/misura-323/misura-323-sperimentale> (2014)
6. 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 ge-ontologies* **3**(4) (2012) 321–332
7. Fonseca, F., Egenhofer, M., C.A. Davis Jr., Borges, K.: Ontologies and knowledge sharing in urban GIS. *Computers, Environment and Urban Systems* **24**(3) (2000) 251–272
8. Fonseca, F., Egenhofer, M., Agouris, P., Câmara, G.: Using ontologies for geographic information systems. *Transactions in GIS* **3** (2002) 231–257
9. Dechter, R.: Constraint networks. *Encyclopedia of Artificial Intelligence* (2nd ed.) (1992) 276–285
10. W3C: Web ontology language (OWL), <https://www.w3.org/OWL/> (2017)
11. OGC: GeoSPARQL - a geographic query language for RDF data, <http://www.opengeospatial.org/standards/geosparql> (2017)
12. Ajit, S., Sleeman, D., Fowler, D.W., Knott, D.: Constraint capture and maintenance in engineering design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **22**(04) (2008) 325–343
13. 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) (2006) 531–550
14. W3C: SPARQL query language for RDF, <https://www.w3.org/TR/rdf-sparql-query/>
15. Torta, G., Ardissono, L., Savoca, A., Voghera, A., Riccia, L.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)*, Funchal, Madeira, Portugal, SCITEPRESS (2017) 86–97
16. W3C: Resource description framework (RDF), <https://www.w3.org/RDF/> (2017)
17. Fath, B., Sharler, U., Ulanowicz, R., Hannon, B.: Ecological network analysis: network construction. *Trends in Ecology & Evolution* **208** (2007) 49–55
18. Ulanowicz, R.: Quantitative methods for ecological network analysis. *Computational biology and chemistry* **28** (2004) 321–339
19. Lurgi, M., Robertson, D.: Automated experimentation in ecological networks. *Automated Experimentation* **3**(1) (2011)
20. 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) (2016) 344–354
21. Pilosof, S., Porter, M., Pascual, M., Kefi, S.: The multilayer nature of ecological networks. *Nature ecology&evolution* **1** (2017) article 101
22. Battle, R., Kolas, D.: Enabling the geospatial Semantic Web with Parliament and GeoSPARQL. *Semantic Web* **3**(4) (2012) 355–370
23. W3C: Linked data, <https://www.w3.org/standards/semanticweb/data> (2018)
24. Urban, S.: Alice: An assertion language for integrity constraint expression. In: *Proc. Computer Software and Applications Conference*. (1989) 292–299
25. Bassiliades, N., Gray, P.: Colan: A functional constraint language and its implementation. *Data & Knowledge Engineering* **14**(3) (1995) 203–249
26. Christensen, J.V., Johnsen, M.: Formalizing constraints for geographic information. In: *Advances in Information Systems Development*. Springer (2006) 657–667
27. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosz, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML. W3C Member submission **21** (2004)

28. Boley, H., Tabet, S., Wagner, G.: Design rationale of ruleml: A markup language for semantic web rules. In: Proceedings of the First International Conference on Semantic Web Working, CEUR-WS. org (2001) 381–401
29. Boley, H., Dean, M., Grosz, B., Sintek, M., Spencer, B., Tabet, S., Wagner, G.: FOL RuleML: The first-order logic web language. Document located on-line at <http://www.ruleml.org/fol> (2004)
30. W3C: A Proposal for a SWRL Extension towards First-Order Logic. <https://www.w3.org/Submission/SWRL-FOL/> (2005)
31. Kessler, C., Raubal, M., Wosniok, C.: Semantic rules for context-aware geographical information retrieval. In: Proc. of the 4th European Conference on Smart Sensing and Context. EuroSSC'09, Springer-Verlag (2009) 77–92
32. 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. (2001) 46–53
33. Arenas, M., Gottlob, G., Pieris, A.: Expressive languages for querying the semantic web. In: Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM (2014) 14–26
34. Fürber, C., Hepp, M.: Using sparql and spin for data quality management on the semantic web. In: Proc. Business Information Systems. (2010) 35–46
35. Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Web Semantics: Science, Services and Agents on the World Wide Web **6**(4) (2008) 309–322
36. OGC: Geosparql vocabulary. http://schemas.opengis.net/geosparql/1.0/geosparql_vocab_all.rdf (2012)
37. Open Geospatial Consortium, et al.: OpenGIS Implementation Standard for Geographic information-Simple feature access-Part 1: Common architecture. (2011)
38. Cohn, A., Bennett, B., Gooday, J., N., G.: Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica* **1**(3) (1997) 275–316
39. Egenhofer, M.: A formal definition of binary topological relationships. In: Proc. Int. Conf. on Foundations of Data Organization and Algorithms. (1989) 457–472
40. 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/lgs_v_lgre.pdf (2014)
41. Krötzsch, M.: Description Logic Rules. Volume 8 of Studies on the Semantic Web. IOS Press (2010)
42. Saalfeld, A.: Topologically consistent line simplification with the douglas-peucker algorithm. *Cartography and Geographic Information Science* **26**(1) (1999) 7–18
43. Angles, R., Gutierrez, C.: The expressive power of sparql. In: Proc. International Semantic Web Conference. (2008) 114–129
44. Kostylev, E., Reutter, J., Ugarte, M.: Construct queries in sparql. *LIPIcs-Leibniz International Proceedings in Informatics* **31** (2015)
45. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1994)
46. Lloyd, J.W.: Foundations of logic programming. Springer Science & Business Media (2012)
47. Marek, V.W., Truszczyński, M.: Stable models and an alternative logic programming paradigm. In: The Logic Programming Paradigm. Springer (1999) 375–398
48. Brailsford, S.C., Potts, C.N., Smith, B.M.: Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research* **119**(3) (1999) 557 – 581
49. Fabri, A., Giezeman, G.J., Kettner, L., Schirra, S., Schönherr, S.: On the design of cgal a computational geometry algorithms library. *Software: Practice and Experience* **30**(11) (2000) 1167–1202

50. Shekhar, S., Xiong, H. In: Java Topology Suite (JTS). Springer US, Boston, MA (2008) 601–601
51. Voghera, A., Crivello, R., L.Ardissono, Lucenteforte, M., Savoca, A., L. La Riccia: Production of spatial representations through collaborative mapping. an experiment. In: Proc. of 9th Int. Conf. on Innovation in Urban and Regional Planning (INPUT 2016). (2016) 356–361
52. Ardissono, L., Lucenteforte, M., Mauro, N., Savoca, A., Voghera, A., Riccia, L.L.: Ontomap - semantic community maps for knowledge sharing. In: Proc. of Hypertext 2017, ACM (2017) 317–318
53. Ardissono, L., Ferrero, M., Petrone, G., Segnan, M.: Enhancing collaborative filtering with friendship information. In: Proc. of UMAP 2017, Springer (2017) 353–354
54. Ecosystem, T.U.: Ushahidi, <http://www.ushahidi.com/about> (2015)
55. Hunter, A., Steiniger, S., Sandalack, B., Liang, S., Kattan, L., Shalaby, A., Alaniz Uribe, F., Bliss-Taylor, C., Martinson, R.: Planyourplace - a geospatial infrastructure for sustainable community planning. *International Journal of Geomatics and Spatial Analysis* **22**(2) (2012) 223–253
56. Sun, Y., Li, S.: Real-time collaborative gis: a technological review. *ISPRS Journal of Photogrammetry and remote sensing* **115** (2016) 143–152
57. Hu, Y., Lv, Z., Wu, J., Janowicz, K., Zhao, X., Yu, B.: A multistage collaborative 3D GIS to support public participation. *Int. Journal of Digital Earth* **8**(3) (2015) 212–234