Fourth International Conference
Modelling and Development of Intelligent Systems
October 28 - November 1, 2015
"Lucian Blaga" University
Sibiu - Romania

Data structuring for the ontological modelling of wind energy systems

Adrian Groza

Abstract

Small wind projects encounter difficulties to be efficiently deployed, partly because wrong way data and information are managed. Ontologies can overcome the drawbacks of partially available, noisy, inconsistent, and heterogeneous data sources, by providing a semantic middleware between low level data and more general knowledge. In this paper, we engineer an ontology for the wind energy domain using description logic as technical instrumentation. We aim to integrate corpus of heterogeneous knowledge, both digital and human, in order to help the interested user to speed-up the initialization of a small-scale wind project. We exemplify one use case scenario of our ontology, that consists of automatically checking whether a planned wind project is compliant or not with the active regulations.

1 Introduction

Small wind projects in the sector of renewable energies encounter difficulties to be efficiently deployed, partly because the wrong way data and information are managed [12, 3]. Ontologies can overcome the drawbacks of partially available, noisy, inconsistent, and heterogeneous data sources [15]. Domain specific ontologies have already been developed for the renewable energy sector [14, 10, 13].

In this line, we aim to develop an ontology for wind energy domain. The ontology was developed in the RacerPro (Renamed ABox and Concept Expression Reasoner Professional) knowledge representation and reasoning system [9]. In our view, RacerPro and the corresponding Knowledge Representation System Specification (KRSS) syntax for Description Logic axioms are powerful technical instrumentation that support ontology engineering behind the basic capabilities provided by GUI-based ontology editors. Compared to the ontologies listed in Table 1, the particularity of our ontology is to complement knowledge extracted from various maps with local observations about the location. Aiming to deal with small wind projects, we rely on 30-50 meter height wind maps and we also include community-scale wind resource maps to quantify the wind resource.

The remaining of the paper is structured as follows: Section 2 briefly introduces the KRSS syntax in which the ontology was developed. Section 3 illustrates how the available knowledge was reused. Section 4 shows the main engineering steps of the ontology. Section 5 depicts how the ontology can be interrogated, while section 6 concludes the paper.

2 Technical instrumentation

The wind data is modelled in an ontology, which is a semantic framework for organising information. We formalise the wind ontology in Description Logic (DL). In the description logic $\mathcal{ALC}$, concepts are built using the set of constructors formed by negation, conjunction, disjunction, value restriction, and existential restriction [2], as shown in Table 2. Here, C and D represent concept descriptions, while r is a role name. The semantics is defined based on an interpretation $I = (\Delta^I, \cdot^I)$, where the domain Δ^I of

Table 1: Examples of ontology-based systems in the renewable energy sector.

Ontology	*Ref.*	*Short description*
OpenWatt	[12]	Global schema for data about solar energy, wind energy, and biomasses.
EMA	[4]	RDF/OWL model of the French electricity company. The ontology-based Energy Management Adviser (EMA) provides personalised tips for 300,000 clients of the company.
WONT	[11]	Semi-automatic created ontology from the Wiki articles in the domain of wind energy (http://www.ceng.metu.edu.tr/we120329/wont.owl).
TurbMon	[17]	Ontology for wind turbines' condition monitoring: focus on wind turbine components and fault detection by means of SPARQL queries.
SEMANCO	[5]	Ontology for urban planning and energy management that describes regions, cities, and buildings; energy consumption and CO2 emission indicators (based on ISO/CD 16346, ISO/CD 16343), climate and socio-economic factors that influence energy consumption.

Table 2: KRSS syntax and semantics of $\mathcal{ALC}$ description logic.

Constructor	*Syntax*	*Semantics*
negation	`(not C)`	$\Delta^I \setminus C^I$
conjunction	`(and C D)`	$C^I \cap D^I$
disjunction	`(or C D)`	$C^I \cup D^I$
existential restriction	`(some r C)`	$\{x \in \Delta^I \mid \exists y : (x,y) \in r^I \wedge y \in C^I\}$
value restriction	`(all r C)`	$\{x \in \Delta^I \mid \forall y : (x,y) \in r^I \rightarrow y \in C^I\}$
individual assertion	`(instance a C)`	$\{a\} \in C^I$
role assertion	`(related a b r)`	$(a^I, b^I) \in r^I$

I contains a non-empty set of individuals, and the interpretation function $\cdot^I$ maps each concept name C to a set of individuals $C^I \in \Delta^I$ and each role r to a binary relation $r^I \in \Delta^I \times \Delta^I$. The last column of Table 2 shows the extension of $\cdot^I$ for non-atomic concepts.

An ontology consists of terminologies (or TBoxes) and assertions (or ABoxes). A terminology $TBox$ is a finite set of terminological axioms of the form `(equiv C D)` or `(implies C D)`. An assertional box $ABox$ is a finite set of concept assertions `(instance i C)`, role assertions `(related i j r)`, or attribute fillers `(attribute-filler i value a)`, where C designates a concept, r a role, a an attribute, and i,j are two individuals. Usually, the unique name assumption holds within the same $ABox$. A concept C is satisfied if there exists an interpretation I such that $C^I \neq \emptyset$. The concept D subsumes the concept C, represented by `(implies C D)` if $C^I \subseteq D^I$ for all interpretations I. Constraints on concepts (i.e. `disjoint`) or on roles (`domain`, `range` of a role, `inverse` roles, or `transitive` properties) can be specified in more expressive description logics. We provide only some basic terminologies of DL in this paper to make it self-contained. For a detailed explanation about families of description logics, the reader is referred to [2].

3 Reusing related ontologies

The process of engineering the ontology was started by: 1) specifying use cases of the ontology, 2) defining a set of competency questions, and 3) analysing the existing ontologies for possible reuse.

First, the *use cases* of the ontology include i) "the assessment of the feasibility to install a particular small-scale wind turbine in a given location" or ii) "checking whether a small wind project is compliant with current regulations.

Second, a solution to narrow the scope of an ontology is to start by defining a list of *competency questions* (CQs) [16]. CQs are questions that an ontology should be able to answer in order to satisfy use

Table 3: Sample of competency questions for the wind turbine domain.

CQ_1	Which is the most adequate wind turbine class for a given location?
CQ_2	Is it norm-compliant to install a wind turbine of a specific class in a given location?
CQ_3	Which is the wind rose distribution for a specific location?
CQ_4	What type of vegetation does exist within a radius of 200 meters?
CQ_5	Which wind turbine types are installed nearby?
CQ_6	Which turbines have alarms for the generator component occurred after a maintenance activity?

Table 4: Reusing related ontologies.

Ontology	*URI*	*Short description*
GeoNames	http://www.geonames.org/	Contains 8.3 million geonames toponym.
BayesOWL	http://semanticweb.org/wiki/Bayes_OWL	A probabilistic extension to the Ontology Language OWL.
Sensor	http://www.w3.org/2005/Incubator/ssn/ssnx/ssn	Describes various sensors and observations.

cases. Thereby, CQs represent initial requirements and they can be used to validate the ontology. Having the role of a requirement, each CQs are initially written in natural language (see Table 3). Then, CQs are formalised in the new Racer Query Language (nRQL) [9] for the task of ontology validation, .

Third, several types of ontologies are needed to merge the input data which is collected in different format and several measurements types are exploited. Thus, existing knowledge bases like wind ontology or measurement ontologies can be exploited. For handling data directly coming from sensors, knowledge about the type of sensor or error and transmission rate are needed, which can be extracted from a specific sensor ontology. Merging different GIS maps requires geographical knowledge. The reused ontologies are listed in Table 4. First, from the GeoNames ontology we exploited: i) hypsographic features (concepts like `Mountain, Hill, Valley, Slope, Mesa, Cliff`), ii) road features (concept like `Junction`, or roles like `roadWidth`), iii) populated places (`City, Area, ResidentialArea`), or iv) vegetation features (`Forrest, Orchard, Scrubland, Vineyard`). Second, we used BayesOWL ontology [6] to represent random variables such as `WindRose` or `WindShear`. Third, we reused concepts from the sensor ontology to describe knowledge about wind-related sensors, as in the following definitions:

```
(implies (or Anemometer WindProfiler WindVane) Sensor)
(implies (or CupAnemometer PropellerAnemometer SonicAnemometer) Anemometer)
```

4 Engineering the wind potential assessment ontology

To develop the wind ontology, we follow the methodology in [16] and we also enact various ontology design patterns [18, 8]. The ontology is a modular one, consisting of a core formalisation and T-boxes for modeling various aspects in the wind energy domain: sub-components of a turbine, classes of turbines, potential of a location, etc.

The sub-componets of a wind turbine are represented by the transitive role `hasPart`:

```
(define-primitive-role hasPart :transitive t)
```

A turbine has one base, one tower, one nacelle, and several blades:

```
(implies WindTurbine (and (=1 hasPart Base)
                          (=1 hasPart Tower)
                          (=1 hasPart Nacelle)
                          (some hasPart Blade)))
```

Given the transitivity of the role `hasPart`, the system is able to deduce all the sub-components of a turbine. The nacelle houses a gearbox and generator, which can be either variable or fixed speed:

```
(implies Nacelle (and (=1 hasPart Gearbox) (=1 hasPart Generator)))
(implies Generator (or VariableGenerator FixedSpeedGeneartor))
(disjoint VariableGenerator FixedSpeedGeneartor)
```

Component prices and technical specification are attached to instances of the concept `WindTurbine`:

```
(instance whisperH20 (and WindTurbine (= 595 hasPrice)))
(instance towerKit1 (and Tower (= 450 hasPrice)))
(related whisperH20 towerKit1 hasPart)
(attribute-filler whisperH20 20 hasBladeArea)
```

Small wind turbines, defined as having a swept area less than 200 m^2, are usually installed between 15 and 40 m high:

```
(implies SmallWindTurbine (and WindTurbine
                               (min sweptArea 200)
                               (min high 15)
                               (max high 40)))
```

A good wind resource is one where wind speeds average 16 mph or more over the course of a year:

```
(define-concrete-domain-attrbiute speedAverage :domain Location :type integer)
(implies GoodWindResource (and WindResource (min speedAverage 16)))
```

Wind speeds are categorized by class, from a low of class 1 to a high of class 7. Wind speeds of class 4 or greater are used for wind power production:

```
(implies GoodWindResource (or WindClass4 WindClass5 WindClass6 WindClass7)))
```

The power of the wind is measured in watts per square meter, and this increases by the cube of the wind speed:

```
(implies top (= windpower (* windspeed windspeed windspeed)))
```

GIS maps are the main source of data fed in the Aboxes of the ontology: (1) wind potential map, (2) wind power map, (3) topographical map, (4) open street map (OSM), (5) transmission lines, (6) archaeological map, (7) vegetation map, (8) rivers and lakes map, (9) digital elevation models. For instance, for converting OSM into KRSS syntax we developed a Java-based converter based on the Osmosis API to import facts about roads. From the wind power map, we defined the potential of a particular wind resource in three classes for the wind speed measured at 50m: marginal, promising, or excellent:

```
(implies MarginalPotentialat50 (and (= hasMeasuredHeight 50) (< 7.5 windSpeed)))
(implies PromisingPotentialat50 (and (= hasMeasuredHeight 50) (> 7.5 windSpeed)
                                     (< 10 windSpeed)))
(implies ExcellentPotentialat50 (and (= hasMeasuredHeight 50) (> 10 windSpeed)))
```

The general concept `MarginalPotential` is the union of the marginal potentials at different heigths:

```
(equivalent MarginalPotential (or MarginalPotentialat15
                                  MarginalPotentialat50
                                  MarginalPotentialat100))
```

These concepts can be used to make rough production estimates at sites, given that the accuracy is +/- 10 to 15% what the actual winds at a site may be.

Access roads of at least 4m wide are assumed necessary, given by:

```
1. (define-concrete-domain-attribute width :domain Road :type real)
2. (implies TurbineAccessRoad (and Road (>= width 4.0)))
```

Wind turbines are efficient in coast, hills, and mountains regions:

```
3. (implies WindTurbine (all efficientIn (or Coast Hill Mountain)))
```

Theoretically, wind turbines can extract up to 59% from the energy which passes through it. Practically, an efficient turbine extracts around 40% from the wind potential [3]. The individual `wt1` is an instance of the concept `WindTurbine` that is able to extract 38% from the wind energy:

```
4. (define-concrete-domain-attribute extracts :domain WindTurbine :type real)
5. (instance teoreticalLimit (= extracts 0.59))
6. (instance practicalLimit (< extracts 0.40))
7. (instance wt1 (and WindTurbine (= practicalLimit 38)))
```

The transitive role `isLocated` connects instances of type `Entity` with instances of the concept `Location` (line 21). ”Partition ontology design pattern” [18] is used to define various types of locations (lines 22-23). The attributes `hasLatitude` and `hasLongitute` are introduced to define a point in space (lines 25-26). Because GIS maps are usually achieved for small regions, we defined aboxes for each region (line 27). Axiom 29 connects the wind turbine `wt1` with the point `p1`. Because the role `isLocated` is transitive the system is able to infer that `wp1` is located in all concepts representing more general locations than `p1` (i.e. Dobrogea, Romania, given that `(related Dobrogea Romania isLocated)`.

```
21. (define-primitive-role isLocated :domain Entity :range Location :transitive t)
22. (implies (or Point Area) Location)
23. (disjoint Area Location)
24. (implies IndustrialArea (and Area (some has IndustrialActivity)))
25. (define-concrete-domain-attribute hasLatitude :type real)
26. (define-concrete-domain-attribute hasLongitude :type real)
27. (init-abox dobrogea-wind-assesment)
28. (instance p1 (and Point (= hasLatitude 44.56) (= hasLongitude 27.54)))
29. (related wt1 p1 isLocated)
```

When setting a home-sized wind turbine, the rule of thumb states that turbine’s rotor should be at least 10 meters above anything within 150 meters of the tower:

```
(define-rule (?wt ProperWindTurbine) (and (?wt WindTurbine)
                                          (?wt ?h1 hasHeight)
                                          (?x top) (?x ?h2 hasHeight)
                                          (?dist DistanceBetween2Objects)
                                          (?dist ?wt hasObject)
                                          (?dist ?x hasObject)
                                          (?dist ?distance hasDistance)
                                          (< distance 150)
                                          (> ?h1 (+ h2 10)))
```

Here, the wind turbine ?wt is proper if its heigth ?h1 is greater then the height h2 of any object ?x in the ontology (?x top). We enacted the ”n-ary ontology design pattern” [18] to store the distance between two objects:

```
(equivalent DistanceBetween2Objects (and (=2 hasObject) (=1 hasDistance)))
(define-concrete-domain-attribute hasDistance :domain DistanceBetween2Objects :type real)
```

The role hasWindShear is used to describe the differences in wind speed at two different heights (line 12) In general, turbulence decreases and wind speed increases as height increases. A WindRose (line 13) shows the direction that the wind blows and the frequency of that direction at a particular location.

```
11. (define-concrete-domain-attribute hasAverageWindSpeed :domain Location :type real)
12. (define-concrete-domain-attribute hasWindShear :domain Location :type real)
13. (implies WindRose (and (=1 hasDirection Direction) (=1 hasFrequency Frequency)))
14. (instance wr (and (= NW hasDirection) (= hasFrequency 0.6)))
```

To represent the random variable WindRose given the variable Day ($P(WindRose|Day)$) we enact the ConditionalProbability concept from the BayesOWL ontology [6]:

```
41. (define-primitive-role hasCondition :domain ConditionalProbability)
42. (define-primitive-role hasVariable :domain ConditionalProbability)
43. (define-concrete-domain-attribute hasProbabilityValue :domain ConditionalProbability
                                                           :type real)
44. (instance cp1 bayesOWL:ConditionalProbability)
45. (instance wr1 (and WindRose RandomVariable))
46. (instance 27June Day)
47. (related cp1 27June bayesOWL:hasCondition)
48. (related cp1 wr1 bayesOWL:hasVariable)
49. (attribute-value cp1 0.6 bayesOWL:hasProbabilityValue)
```

5 Retrieving information from Aboxes

Once a user’s situation has been fully described using DL assertions, the RacerPro [9] reasoner was used to classify the location as an instance of one or more of the situation concepts described in the ontology.

Ontology reasoning can be used to check if the location of a wind turbine does not breach active regulations. The following axioms checks whether the wind turbine is located at a minimum distance of 300m from a residential area.

```
31. (instance v1 Village)
32. (implies Village (and geonames:PopulatedPlace geonames:ResidentialArea))
33. (instance a1 (and Area (= hasContourPoint p2) (= hasCountourPoint p3)
                       (= hasContourPoint p4)))
34. (related v1 a1 isLocated)
35. (define-concrete-domain-attribute hasValue :domain Distance :type integer)
36. (instance d1 (and Distance (= hasObject wt1) (= hasObject v1)))
37. (attribute-filler d1 280 hasValue)
38. (instance minimumDistance (= distance 300))
```

Given the village v1 (line 31), we use axioms from GeoNames ontology (axiom 32) to deduce that a village is a residential area. We enact AllegroGraph [1] to perform geospatial reasoning on various tasks, such as computing the distance between a point and a geographic area. Using also the Harversine formula, the minimum distance d1 between potential wind turbine location p1 and residential area a1 of `v1` (line 34) is asserted in the ontology (line 37). Here, area `a1` is a triangular shape defined by three points (line 33) and `d1` is an instance of the concept `Distance` between the starting point the turbine `wp1` and the ending point the village `v1` (line 36). Given that the minimum distance between a turbine an a residential area should be 300m (fact 38) the ontology is able to infer that the turbine `v1` breaches the above normative condition.

Four types of queries on the wind-turbines ontology are illustrated: i) checking the ontology consistency (line 71), ii) retrieving information about individual `wt1` (line 72), iii) identifing the sub-concepts of the various concepts in the ontology (lines 73-74) and iv) retrieving all wind turbines located in Dobrogea region (line 75).

```
71. (tbox-cyclic?)  (tbox-coherent?) (abox-consistent?)
72. (describe-individual wt1)
73. (concept-children WindTurbine)
74. (concept-descendents Wind Turbine)
75. (concept-instances (and WindTurbine (some isLocated Dobrogea)))
```

6 Conclusion and ongoing work

The proposed ontology captures knowledge in the wind energy domain. The knowledge was formalised in Description Logic that provides a rich, flexible and fully declarative language for modelling knowledge about environment. The specific reasoning services on DL were exploited to deduce new knowledge and to classify a location according to its wind potential. The ontology can assist the wind farm operators lowering costs and complying with current regulations in the wind energy sector. The wind ontology will help wind farm operators make informed, evidence based decisions about deployment and maintanance of a wind project.

We did not focus here on the problem of reasoning on data streams [7] continously collected from sensors. Ongoing work regards extending the ontology with terrain analysis, infrastructure, environment and complete legal constraints, into a single predictive map that shows the most suitable site to explore for wind energy.

References

[1] Jans Aasman. Allegro graph: RDF triple database. Technical report, Franz Incorporated, 2006. url: http://www.franz.com/agraph/allegrograph/(visited on 20/04/2015), 2006.

[2] Franz Baader. *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003.

[3] Octavian Capatina and Ioan Letia. Online knowledge based wind assessment tool. In *Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on*, pages 352–355. IEEE, 2012.

[4] Pierre Chaussecourte, Birte Glimm, Ian Horrocks, Boris Motik, and Laurent Pierre. The energy management adviser at edf. In Harith Alani and et., editors, *The Semantic Web ISWC 2013*, volume 8219 of *Lecture Notes in Computer Science*, pages 49–64. Springer Berlin Heidelberg, 2013.

[5] Vincenzo Corrado, Ilaria Ballarini, Leandro Madrazo, and German Nemirovskij. Data structuring for the ontological modelling of urban energy systems: The experience of the SEMANCO project. *Sustainable Cities and Society*, 14:223–235, 2015.

[6] Zhongli Ding, Yun Peng, and Rong Pan. Bayesowl: Uncertainty modeling in semantic web ontologies. In *Soft Computing in Ontologies and Semantic Web*, pages 3–29. Springer, 2006.

[7] Adrian Groza and Ioan Alfred Letia. Plausible description logic programs for stream reasoning. *Future Internet*, 4(4):865–881, 2012.

[8] Adrian Groza, Anca Marginean, and Vlad Muresan. An ontology-based model for vehicular adhoc networks. In *Intelligent Engineering Systems (INES), 2014 18th International Conference on*, pages 83–88. IEEE, 2014.

[9] Volker Haarslev, Kay Hidde, Ralf Möller, and Michael Wessel. The RacerPro knowledge representation and reasoning system. *Semantic Web Journal*, 3(3):267–277, 2012.

[10] Ahmad Kayed. Renewable energy ontology. In *Applications of Information Technology to Renewable Energy Processes and Systems (IT-DREPS), 2013 1st International Conference & Exhibition on the*, pages 144–148. IEEE, 2013.

[11] Dilek Kk and Yusuf Arslan. Semi-automatic construction of a domain ontology for wind energy using Wikipedia articles. *Renewable Energy*, 62(0):484 – 489, 2014.

[12] D Davide Lamanna and Antonio Maccioni. Renewable energy data sources in the semantic web with OpenWatt. In *EDBT/ICDT Workshops*, pages 128–133, 2014.

[13] Trinh Hoang Nguyen, Rocky Dunlap, Leo Mark, Andreas Prinz, Bjørn Mo Østgren, and Trond Friisø. Offshore wind metadata management. *International Journal of Metadata, Semantics and Ontologies*, 9(4):333–349, 2014.

[14] Trinh Hoang Nguyen, Andreas Prinz, Trond Friis, Rolf Nossum, and Ilya Tyapin. A framework for data integration of offshore wind farms. *Renewable Energy*, 60(0):150 – 161, 2013.

[15] Trinh Hoang Nguyen, Andreas Prinz, and Josef Noll. Proactive maintenance of offshore wind turbine blades using knowledge-based force analysis. In *Innovative Computing Technology (INTECH), 2013 Third International Conference on*, pages 239–245. IEEE, 2013.

[16] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[17] P Papadopoulos and L Cipcigan. Wind turbines' condition monitoring: an ontology model. In *Sustainable power generation and supply, 2009. SUPERGEN'09. International conference on*, pages 1–4. IEEE, 2009.

[18] Jeffrey T Pollock and Ralph Hodgson. Ontology design patterns. *Adaptive Information: Improving Business through Semantic Interoperability, Grid Computing, and Enterprise Integration*, pages 145–194.

Adrian Groza
Intelligent Systems Group
Department of Computer Science
Technical University of Cluj-Napoca
Baritiu 26-28, Cluj-Napoca, ROMANIA
E-mail: *adrian.groza@cs.utcluj.ro*