Notitie

Anna van Buerenplein 1
2595 DA Den Haag
Postbus 96800
2509 JE Den Haag

www.tno.nl

T +31 88 866 00 00

Aan
DDINGS Stuurgroep

Van
TNO

Onderwerp
Common Greenhouse Ontology manual

Datum
16 juni 2021

Onze referentie
<vnr-ext>

E-mail
romy.vandrie@tno.nl

Contents

CGO manual

1. Common Greenhouse Ontology

An ontology is a formal model of the structure of a system. It captures relevant concepts of a domain and describes the relations between these concepts. Concepts are usually called *classes*, and the relation between them *properties*. The properties are split up into two kinds of properties: *object properties* and *data properties*. Object properties are used to connect classes, and data properties to connect data, such as numerical values, to classes. Finally, the actual data in an ontology is expressed with *individuals.* These concepts are discussed and visualized in section 1.2 and figure 1.

The Common Greenhouse Ontology (CGO) captures information relevant to greenhouses. It is specifically geared towards high-tech greenhouses with modern systems that produce data about the greenhouse. The ontology was developed with domain experts and focuses on greenhouse-related concepts and measurements done in the greenhouse.

1.1 Getting started

The goal of this section is to help you get started with the ontology. If you want to link your data to the ontology, a prerequisite is that all the relevant concepts are in the ontology. Section 1.1.1 gives a 3-step action plan for checking this, and some rules of thumb to make adding new concepts easier. If you are an experienced ontology engineer, you can skip this section. When adding concepts to the ontology, you must choose to add them as a class, object property or data property. Section 1.1.2 serves as an introduction to this distinction. We discuss the design choices between this distinction in the CGO and give examples of a class, object property and data property.

1.1.1 How to find and view the CGO

The latest published version of the CGO can be found on TNO's ontology server[1]. This is where you have access to the ontology's turtle file, which is necessary for viewing and editing the ontology in an ontology viewer such as Protégé and TopBraid. For beginners, we recommend Protégé. The ontology is also made available on WIDOCO.[2] Here, you can inspect the ontology without opening it in an ontology viewer, but if you want to edit, you will need to open the ontology in a viewer.

[1] https://ontology.tno.nl/
[2] https://app-ddinges01.hex.tno.nl:3334/ddings/index-en.html

1.1.2 How to check if relevant concepts are in the CGO

To make sure all relevant concepts are in the ontology, we recommend following the 3 steps below:

1. Formulate what you want to know (in a question form).

> Example:
> I want to measure the humidity in the greenhouse, or
> What is the humidity in the greenhouse

2. Collect the content words from your question.

> Example:
>
> I want to measure the humidity in the greenhouse, or
> What is the humidity in the greenhouse
>
> Note: *measure* could also be marked as a content word, but verbs do not have a one to one mapping to the ontology. You could say that *measure* indicates the use of a sensor.

3. Search for these words in the ontology.
   You can search for these terms using an ontology editor such as Protégé or TopBraid or open the ontology file in a text editor and search there. If you cannot find your concept, please try to search using synonyms or try a different language (English or Dutch). For example, instead of *humidity,* try looking for *vochtigheid* or *luchtvochtigheid.*

If you determine that there are concepts and/or properties that are not part of the CGO, follow 'how to add new concepts or properties to the ontology'

1.1.3 How to add new concepts to the CGO?

When adding new concepts, you can add them as classes, object properties or data properties. Classes (e.g. *greenhouse* or *sensor*) are used for concepts, whereas properties are used to indicate a link between two things, either a concept and another concept, or a concept and a datapoint. For example, a property *has length* can be used to indicate a link between a class *greenhouse* and a datapoint containing the length of the greenhouse. In case the link with a datapoint is a direct one, such as in this example, a data property is used. If the link with the datapoint is not direct, an object property is used. This is the case with the object property *has feature of interest*. This is property is imported from the Semantic Sensor Network Ontology (SSN; Neuhaus, 2009), which is discussed in more detail in section 1.2. For now, it is sufficient to know that this object property indicates a relation between an observation (which is a class) and an entity of which the observation was made (which is also a class).

1.2 Architecture

The CGO contains 382 classes, 99 properties (both data and object properties) and 23 individuals. This includes the classes and properties from the SSN ontology[3]. We can divide the content into four main modules: the greenhouse, which is the center concept of the ontology, features, which are set properties of the greenhouse such as its dimensions or the location, parts, which are the objects that can be found in greenhouses, and finally, measurements. The greenhouse contains many sensors and systems which measure parts, and possibly features. For instance, the thickness of the stem of a flower can be measured and is relevant to keep track of its growth. The modules are shown in blue in Figure 1. The measurements in the greenhouse are modelled using two import ontologies: the SSN, which is shown in yellow in Figure 1, and the Ontology of units of Measure (OM; Rijgersberg, 2011), shown in pink. OM is used to model the results of measurements and observations, SSN to model the different aspects of the observations.

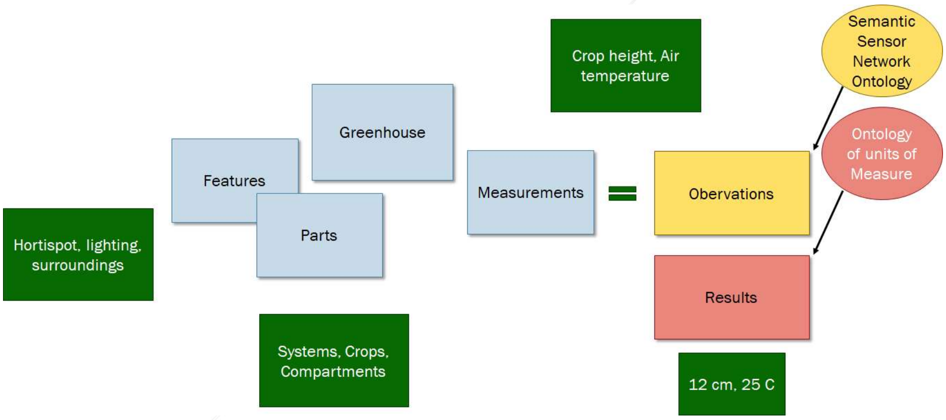
Crop height, Air temperature
Semantic Sensor Network Ontology
Greenhouse
Features
Measurements
=
Obervations
Ontology of units of Measure
Parts
Hortispot, lighting, surroundings
Results
Systems, Crops, Compartments
12 cm, 25 C

Figure 1. CGO modules (blue), examples (green), and imported ontologies (yellow, pink).

Observations are an important aspect of the CGO, but not easy to implement. An observation always has multiple aspects, such as the thing one observes, what is observed, how it is observed, etc. The SSN ontology models those aspects and is therefore reused in the CGO. In Figure 2, a simplified picture of the underlying architecture of observations according to SSN is visualized with an example. The example observation here is the thickness of a stem (of a flower). This observation can be split into multiple parts: the stem, the thickness, and the result. The stem (of a flower) is the specific part that is being observed, this is called the Feature of Interest. We are looking at a specific property of this stem: the thickness. This is the observable property. Other examples of observable properties would be the color or the length. These two classes are coupled to the main observation, the center of the architecture to which all elements are connected, including the result of the observation. The result can be a simple result, such as a string, or a value-unit combination, such as *8mm*. There are other aspects of observations, such as a method, or the way the observation was made (e.g. a sensor). Those are all included in SSN[4], but not shown here because they are used less in the CGO.

[3] https://www.w3.org/TR/vocab-ssn/
[4] https://www.w3.org/TR/vocab-ssn/#SOSASensor

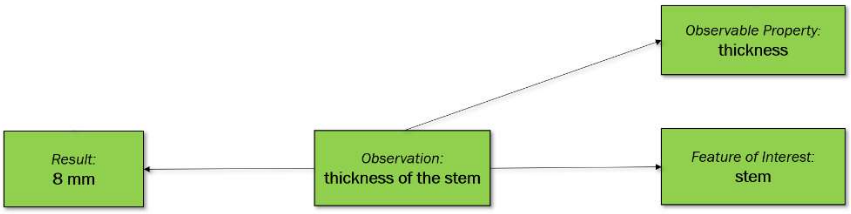

Figure 2. Part of the SSN ontology regarding observations.

The classes and properties of the SSN architecture are integrated in the CGO. The CGO modules involved with SSN are all connected as subclasses of SSN classes. The Parts module contains subclasses that fall under the Feature of Interest class of SSN, therefore assuring that all classes of the Parts module can be used as Features of Interest in observations. For instance, the parts module is a superclass that has subclasses such as *Construction Part*, and *Plant Part*. Construction parts are the floor, the roof, of the ventilation windows in a greenhouse. Plant parts include the leaves of the plant, the stem, or the branches. All those parts are subject to measurements, in the above example in figure 2 the stem, which is a part of the crop, is the object of our observation. But to use them in the observation according to SSN, they have to be a Feature of Interest. By making these parts a subclass of Feature of Interest, this usage is made possible.

The Measurement module is a subclass of SSN's Observation. The SSN Observation class is very broad, the CGO measurement class is a specific kind of observation that is made in the Greenhouse. For instance, SSN observations include all kinds of observations made by humans, computers, sensors, etc., in all places, from all objects. CGO Measurements is more narrow, this class only includes observations made inside and around greenhouses.

Finally, CGO classes that represent Observable Properties, such as thickness, are added as subclasses of SSN's Observable Property. These connections can be seen in Figure 3. The result of an observation is implemented using OM[5]. Compound results, that is, results that are expressed with a value and a unit such as 8mm, are expressed by using an object property 'has value', which connects to a value. This value has in turn two data properties: in our case 'has numerical value: 8' and 'has unit: millimeter'.

For an overview of all the classes in the ontology and their hierarchy, you can consult the ontology itself or view its documentation on the WIDOCO page of the CGO.[6]

[5] For the complete OM ontology, the architecture, and examples, see https://github.com/HajoRijgersberg/OM

[6] https://app-ddinges01.hex.tno.nl:3334/ddings/index-en.html

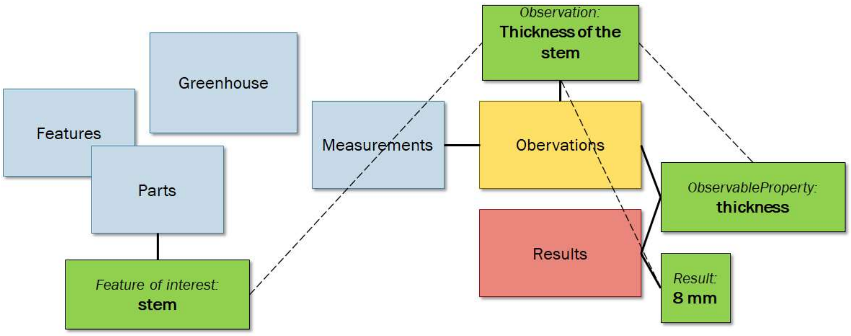

Figure 3. CGO modules and SSN observation connections.

1.3 Individuals

The CGO is specialized in expressing measurements inside the greenhouse. Measurements in greenhouses are often made with sensors and climate computers. This means that the results of these measurements are usually available in a database. To connect the database to the ontology for querying your data, the data needs to be triplified, that is, it needs to be transformed to a format that is compatible with the ontology. More information on how to do this can be found in the manual of the datahub. For this documentation, we assume that the data has the right format. The data can now be regarded as individuals. Individuals are the basic components of an ontology. They are the incarnate classes. For instance, *human* is the class categorizing humans, and I am an individual, or instance, of that class. In CGO terms, *greenhouse* is the class describing all greenhouses, and the greenhouse named X standing on location Y is an individual of that class.

Measurements in the greenhouse have results, as described above in the architecture. This result, in our previous example 8mm, is an instance of the OM class *Measure*, which in turn is a subclass of the SSN class *Result*. The CGO contains 27 example individuals. The other individuals from databases are gathered in a triple store inside the data hub.

1.4 Competency questions

For testing the CGO, we created multiple competency questions (CQ). Competency questions are for an ontology what requirements are for software. The answer to "Is the right system build?" should be sought primarily in the CQs which ask for facts and figures about the who, what, when, what for, where, whence and alike that are required in order to generate the business value as described by the motivating scenario. Competency questions can be low, middle, and high level. The low-level competency questions ask questions about the data. Ontologically, this level describes property/object relations. Middle level CQs describe relations between several concepts along several property/object relations. Ontologically they describe relations among at least three concepts. High-level CQs describe the questions that show the added value to the ontology, in this case financial value, but it can also be otherwise. Below, two examples of each category are shown. The complete set can be found in the CGO.

Low - What is the location of sensor X?
- When did observation X occur?
- How many sensors are inside the greenhouse?

Middle - Which sensors hang next to the ridge foils?
- What crop uses ridge foils?
- What are the temperature measurements between 112020 and 122020?

High - Is there a relation between sensor height and measured temperature?
- What is the effect of ridge foils on the temperature?
- Do ridge foils influence the growth of the crops?

2. SPARQL queries

To interrogate the model, it is common to use the SPARQL query language. Section 2 gives some example queries and helps you getting started creating SPARQL queries.

The most basic (and useful) SPARQL query is structured as:

```
SELECT [variables] WHERE { [graph pattern] }
```

Where [variables] is replaced with a space-separated list of variables that you want in the result, and [graph pattern] is replaced by list of triples that define the shape of the data you want to capture. For more on SPARQL, refer to the W3C recommendation[7].

2.1 Getting started

To show how to create SPARQL queries for the CGO, we list a few questions in natural language, and show the corresponding SPARQL queries that answer them.

Which subclasses of features of interest exist in the ontology?

```
SELECT ?foi WHERE {
  ?foi rdfs:subClassOf sosa:FeatureOfInterest .
}
```

What were the temperatures of observations at which time?

```
SELECT ?time ?value
WHERE {
  ?observation a sosa:Observation .
  ?observation sosa:observedProperty cgo:Temperature .
  ?observation sosa:hasSimpleResult ?value .
  ?observation sosa:resultTime ?time .
}
```

[7] https://www.w3.org/TR/sparql11-query/, specifically section 2 about simple queries.

What parts does the greenhouse ex:greenhouse have?

```
PREFIX ex: <https://www.example.org/>

SELECT ?part
WHERE {
  ex:greenhouse1 cgo:hasPart ?part .
}
```

What sensors does the greenhouse ex:greenhouse have?

```
PREFIX ex: <https://www.example.org/>

SELECT ?sensor
WHERE {
  ex:greenhouse1 cgo:hasPart ?sensor .
  ?sensor a sosa:Sensor .
}
```

References

de Almeida Falbo, R. (2014). SABiO: Systematic Approach for Building Ontologies. In ONTO. COM/ODISE@ FOIS.

Neuhaus, H., & Compton, M. (2009). The semantic sensor network ontology. In AGILE workshop on challenges in geospatial data harmonisation, Hannover, Germany (pp. 1-33).

Noy, N., & McGuinness, D. (2001). *Ontology development 101*: A guide to creating your first ontology. Development, 32, 1–25.

Peroni, S. (2017). *A simplified agile methodology for ontology development*. In M. Dragoni, M. Poveda-Villalón, & E. Jimenez-Ruiz (Eds.), *OWL: Experiences and Directions – Reasoner Evaluation (13th International Workshop, OWLED 2016, and 5th International Workshop, ORE 2016, Revised Selected Papers)* (Vol. LNCS-10161, pp. 55–69). Bologna, Italy.

H. Rijgersberg, M.L.I. Wigham, J.L. Top, (2011). “How semantics can improve engineering processes. A case of units of measure and quantities.” *Advanced Engineering Informatics*, 25, 2, 2011, pp. 276-287.