Ontological Formalisation of Mathematical Equations for Phenomic Data Exploitation

Luis Felipe Vargas-Rojas

Ontological Formalisation of Mathematical Equations for Phenomic Data Exploitation

Felipe Vargas-Rojas[1,2(✉)]

[1] LEPSE, Université Montpellier, INRAE, Institut Agro, Montpellier, France
[2] MISTEA, Université Montpellier, INRAE, Institut Agro, Montpellier, France
luis-felipe.vargas-rojas@inrae.fr

Abstract. In recent years, plant phenomics community has adopted Semantic Web technologies in order to harmonise heterogeneous, multi-scale and multi-source datasets. Semantic Web provides inference services for representing logic relationships in an unambiguous, homogeneous and clean manner, which enhances data harmonisation. However, mathematical relationships involving numerical attributes are poorly formalised, despite the fact that they are supported for a theoretical and well-defined structure. For instance, whilst unit ontologies (e.g. UO, OM, QUDT) provide relationships and annotations to perform unit conversion, they are not effectively used for automating the integration of heterogeneous measurements. Here we propose an ontological framework for representing mathematical equations supporting the automatised use of inference services, metadata, domain ontologies, and the internal structure of mathematical equations. This approach is evaluated using two plant phenomics case studies involving the calculation of unit conversions and thermal time.

Keywords: Semantic Web · Plant phenomics · Ontological reasoning · Mathematical equations

1 Introduction

Plant Phenomics (PP) has produced massive datasets involving experiments performed in the field and controlled conditions, concerning hundreds of genotypes at different scales of organisation. These datasets are unprecedented resources for identifying and testing novel mechanisms and models [17]. Assembling and organising such datasets is not straightforward because of the heterogeneous, multi-scale and multi-source nature of data.

Supported by INRAE and #DigitAg. This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004

Category: Early Stage Ph.D. **Topic:** Deductive Reasoning, Neuro-symbolic reasoning, Inductive Reasoning.

R. Verborgh et al. (Eds.): ESWC 2021 Satellite Events, LNCS 12739, pp. 176–185, 2021.
https://doi.org/10.1007/978-3-030-80418-3_30

Recently, the PHIS[1] [11] ontology-driven information system based on FAIR principles [18] has been proposed as a tool for managing phenomics data. PHIS allows expressing a number of relationships implicit in the data, like hierarchies, mappings and constraint values. However, some numerical relationships cannot be expressed using this mechanism neglecting a number of data often used in PP (observations and measurements). State-of-the-art in PP is populated by mathematical equations relating different plant and environmental traits in different scales, invoking arithmetic and series operations (summations, aggregations).

In this paper we propose an ontological framework for representing mathematical equations and exploiting inference services. Our main contributions are: (i) a model for representing mathematical equations, (ii) a reasoning-based mechanism to compute the equations, (iii) a module to automate unit conversion based on unit ontologies.

The paper is organised as follows. Section 2 discusses the related work describing mathematical equation representation in Semantic Web (SW) while Sect. 3 presents the problem and contributions. Sections 4 and 5 are the main part of the paper, presenting a preliminary methodology and evaluation plan. Finally, Sect. 6 presents the conclusions.

2 State of the Art

Although there is not an ontological framework that addresses directly the proposed features, a number methods allow computing mathematical expressions related to SW technologies.

- **Ontology-based information representation**: the expression is represented in some formal language but the machinery for evaluating is not associated
- **Ontological reasoning**: the expression is evaluated as part of the reasoning task
- **SPARQL extension**: an SPARQL [4] function facilitates the expression evaluation
- **Ontology-based delegated computing**: the expression is evaluated by an external tool and the necessary information is structured using ontologies

The following approaches are organised by the used method and reviewed taking into account these criteria: (i) how is the information represented, (ii) what is the expressive power for each approach, (iii) where is the computation executed, (iv) how are the inference services used.

2.1 Ontology-Based Information Representation

In these approaches the system contains annotated datasets, and occasionally information for describing some execution parameters. Hence, the information

[1] http://www.phis.inra.fr.

should be transferred to local scripts for handling the required transformations. As an example the Function Ontology [5] describes functions independently of the programming language, focusing on the function name and attributes information without semantic information about the internal computed mathematical model or the resulting value. A number of studies tackle the problem of representing units of measurement, providing means to describe units and to some extent model conversion between these units (e.g. unit ontologies UO [7] and OM [16]). However, non of these studies specify a concrete machinery to perform unit conversions [2].

In a recent study using two unit ontologies, OM [16] and QUDT [8], Martín-Recuerda et al. [10] evidence the challenges related to the use of metadata for computing unit conversion. Unit conversion is performed in QUDT by using the values of two data properties: *qudt:conversionMultiplier* and *qudt:conversionOffset*. The values of these properties determine how the magnitude of a quantity value can be converted to a base (or reference derived). Conversely, unit conversion in OM it is not straightforward since the conversion factor (or multiplier) and offsets are not available for all derived units. Consequently, it is necessary to navigate along the RDF graph to find an unit that has one or two of these properties to obtain the necessary conversion factors and offsets for a given unit. In this study the conversion is invoked within the query definition in SPARQL.

2.2 Ontological Reasoning

Ontological reasoning allows to define computations in terms of ontology concepts and assigning the resulting value to ontological properties.

In this line Bischof et al. [3] extended the inference services of RDFS allowing axioms about equations by adding the type "equation" to the TBOX. The expressive power of this proposal was limited to simple equations without considering aggregations or summations. In addition, the equations were embedded as strings without semantics inside the components. Besides the former problems, the incorporation of unit names into the properties instead of using unit ontologies (e.g. tempHighF, tempHighC), leads to an excessive proliferation of properties [13].

In another study, Parsia et Smith [13] introduced a method for unit conversions based on a new datatype system for quantities (e.g. "6 feet"^^owl:quantity). They argued that axiomatising quantities leads to performance issues and contaminates the axiomatisation of the domain, whilst a new datatype will enable special syntax and semantic support for the worked out theory about quantities. This approach requires that conversions are calculated during insertion time, missing information about the original quantity form. This is also disconnected from the evolution of units ontologies, since the unit is imposed inside an string as a text and not as a linked resource.

2.3 SPARQL Extensions

SPARQL can be extended to perform calculations on top of the basic graph pattern (BGP) [6]. When a query is executed, all the data matching the pattern are loaded in memory for later operations, allowing operations like aggregation and SPARQL functions to be calculated based on these in-memory stored records. Although it is not possible to invoke inferences in these approaches, they offer a computation environment that could be exploited to evaluate the equations.

Hogan et al. [9] proposed a language aimed to integrate graph querying with analytical tasks supporting custom computations over the existing SPARQL infrastructure. The language increments the expressive power of SPARQL allowing *for loops* and variables to assign subgraphs. The proposed language is far from the mathematical notions and more related to SPARQL queries, and scripts are defined using the RDF structures. Nevertheless, it demonstrates that analytical tasks can be performed on a SPARQL extension.

2.4 Ontology-Based Delegated Computing

In these studies, computations are delegated to external tools such as Matlab, Python, SPSS, or R. This is a complementary approach to Ontology-based information representation adding semantic information about the external execution. For instance, Rijgersberg et al. [15] proposed the Ontology of Quantitative Research (OQR) for annotating scientific data, allowing people and machines to interpret and connect to real-world phenomena as well as metadata for automating invocation of numerical software. In this approach the computations follow a black-box model where is not possible to connect the internal structure with ontology concepts or inference services. Finally, the ontology is defined from scratch for an specific purpose.

Beck et al. [1] proposed an ontology for building simulations in agriculture systems modelling by including several web-based visual design tools where users can create a model and automatically generate the simulation code. Symbols, operators and variables are represented using a proposed ontology. This approach allows to represent the model structure, and to connect with the ontology concepts. Several inference services can be executed like subsumption and classification, but the computations remain delegated to the external software where the generated script will be executed.

3 Problem Statement and Contributions

The state of the art shows a lack of studies exploiting the inference services interconnected with a formalisation of mathematical equations. Despite the availability of several formal languages to represent mathematical formulas (MathML[2], OpenMath[3]), they are merely descriptive and not effectively integrated with the reasoning services. On the other hand, to the best of our knowledge, the few approaches addressing the integration of reasoning tasks do not consider units ontologies annotations neither the expressive power to deal with aggregations and summations (typical SPARQL operations).

[2] https://www.w3.org/Math/.
[3] https://www.openmath.org/.

Our contribution is to propose a framework using the SW stack for representing mathematical equations in terms of PP attributes. Here we will address: (i) how to easily represent mathematical equations independently on the execution engine, in a manner more compliant with symbolic mathematics than programming languages, (ii) how to link or define these equations using the ontology concepts and properties linked to public ontologies, (PO[4], CO[5], AgrO[6]) instead of isolated meaningless variables (x, y, z). (iii) characterise the trade-off when equations are embedded within reasoning tasks, (iv) how to use unit ontologies in order to harmonise numerical data, (v) how to deal with nested equations.

For instance, assuming two environmental datasets (D1 and D2) with different schemes and the following equation:

$$sizeN = size(ex\!:\!dailyPrecipitation) \tag{1}$$

$$ex\!:\!avgMonthPrecipitacion = \frac{\sum(ex\!:\!dailyPrecipitation)}{sizeN} \tag{2}$$

Let us suppose that D1 has an attribute *ex:dailyPrecipitation* and that D2 uses another name convention like *ex2:dailyRaining*. Directly, D2 is not accepted by the equations, however we can state a rule to unify the two datasets through the inference services:

$$ex2:dailyRaining \quad rdfs:subPropertyOf \quad ex:dailyPrecipitation \tag{3}$$

Then, if during the calculation time the equation is interconnected with the reasoning task, the system can apply the computation to both datasets. In this regard, our contribution should also analyse the expected benefits, we present some examples:

1. Define equations close to mathematical structures instead of programming language expressions
2. Define equation variables using ontology terms
3. Apply same equations for heterogeneous datasets schemes using OWL/RDFS inference rules for mapping (owl:sameAs, rdf:subPropertyOf)
4. Offer up-to-date results avoiding proliferation of stored attributes (lazy evaluation)
5. Automate unit conversion harmonisation for heterogeneous observations

4 Research Methodology and Approach

The ontological framework development can be divided into different steps, each one addressing specific features and challenges.

The first step is about *mathematical equation representation*, in this step we will investigate alternatives to represent mathematical equations as shown

[4] http://obofoundry.org/ontology/po.
[5] https://www.cropontology.org/.
[6] http://obofoundry.org/ontology/agro.

in Sect. 2.1. In order to select the most appropriate model, the following criteria will be taken into account: (i) similarity with mathematical notation, (ii) expressive power, (iii) compatibility with RDF and OWL, Finally, if necessary, we should extend and adjust the provided functionalities to support the case study requirements.

The second step considers *revisiting unit ontologies*. As mentioned in Sect. 2.1, unit ontologies are a fundamental resource for exploiting numerical data. In this step, we will revisit unit ontologies in order to define which one among the publicly available is the more suitable to perform unit conversion.

The following step concerns the *reasoning implementation*. Several approaches were mentioned in Sect. 2.2 to perform reasoning coupled with computable expressions. In this step we will implement the code to embed equations within the inference engine. The following possibilities will be tested:

- **Modify a query rewriting algorithm:** given a user's query and a set of equations, rewrite the query to perform the calculations
- **Create a new literal data type:** a new data type such as *xsd:float* or *xsd:double* could be defined, e.g. *owl:equation*, then implement a machinery able to handle the data type
- **Extending SWRL rules:** this language to define rules can be extended to handle equations
- **A module extension of SPARQL:** create a module that recognises the equation and defines the calculations

In this step, we will face the computational boundaries of each approach and will test which of the former possibilities is more suited to exploit the inference services.

4.1 Case Studies

In order to prove the feasibility of the proposal, this research will focus in two concrete case studies from PP, each of them increases the complexity and the functionalities required to reach the task:

Perform Unit Conversions. The aim is to automate the unit conversion using a formal definition of formulas like:

$$1\,\mathrm{m}^2 = 10000\,\mathrm{cm}^2, 1\,\mathrm{cm}^2 = 1\,\mathrm{m}^2 \times 10^{-4} \tag{4}$$

As a result, the user could query the data asking in either centimetres or metres. Whilst this may seem trivial as simple equations allowing unit conversions are broadly known (e.g. cm^2 to m^2) and these are routine operations performed by users, they are often an important source of errors. This is particularly the case for complex unit conversions involving different concepts, units and dimensions, and when heterogeneous datasets should be harmonised (e.g. light units) [14]. For instance, combining data from a pyranometer (measuring global solar radiation (R_s)) and a quantum sensor (measuring photosynthetically active radiation

(PAR)) is not straightforward since both sensors measure different variables in different units. R_s is often expressed using multiple units (e.g. W m^{-2}, J cm^{-2} s^{-1}, MJ m^{-2} d^{-1}), and PAR data is usually provided in μmol m^{-2} s^{-1} thus requiring unit conversions and aggregation of data.

As an example, the conversion of 80 J cm^{-2} of solar radiation to μmol m^{-2} s^{-1} of PAR considering the time of 30 min (1800 s) involves a number of steps [14]:

First the conversion factor for solar radiation (kJ m^{-2} $time^{-1}$) to (W m^{-2}) is:

$$1\frac{kJ}{m^2 \cdot s} = 10^3\frac{W}{m^2} \tag{5}$$

For a period of 30 min (1800 s):

$$\frac{80\ J}{cm^2 \cdot 1800\ s} \rightarrow \frac{10^{-3}\ kJ}{1\ J} \times \frac{1\ cm^2}{10^{-4}\ m^2} \times 10^3\frac{W}{m^2} \rightarrow 444.4\frac{W}{m^2} \tag{6}$$

Then the conversion factor for solar radiation (R_s in W m^{-2}) to PAR in μmol m^{-2} s^{-1} is:

$$1\frac{W}{m^2} = 2.02\frac{\mu mol}{m^2 \cdot s} \tag{7}$$

Finally:

$$444.4\frac{W}{m^2} \rightarrow 444.4 \times 2.02\frac{\mu mol}{m^2 \cdot s} \rightarrow 897.8\frac{\mu mol}{m^2 \cdot s} \tag{8}$$

Main challenges here are related to the specific designs of unit ontologies and the fact that each ontology individual can have distinct units.

Calculation of Thermal Time. Thermal time (i.e. growing degree units) is one of the common processes currently handled by biologists and agronomists which is used to normalise several temperature-dependent processes such as leaf-progression. It can be either calculated using a simple linear model and a species-specific base temperature parameter, a bilinear model with some optimum and minimum temperature parameters or even using a process-based bell-shaped model [12]. Its calculation requires then a number of steps and necessary meta-data (e.g. input temperatures, species, parameters, integration time, interval of calculation).

Thermal time using a species-dependent base temperature (T_0) and an observed temperature (T):

$$ThermalTime = T - T_0 \tag{9}$$

With some boundary conditions:

$$if\ T < T_0 \rightarrow T = T_0 \tag{10}$$

Thermal time using a species-dependent base, optimum and maximum temperatures (T_0, T_{opt}, T_{max})

$$if\ T > T_0 \leq T_{opt} \rightarrow ThermalTime = T - T_0 \tag{11}$$

$$if\ T > T_{opt} \leq T_{max} \rightarrow ThermalTime = T - T_{max} \tag{12}$$

With some boundary conditions:

$$if\ T < T_0 \rightarrow T = T_0 \tag{13}$$

$$if\ T > T_{max} \rightarrow T = T_{max} \tag{14}$$

The main challenges with thermal time calculation are related to the combination of if-then rules with equations, and the necessity to reuse the unit conversion module because the attributes are often in different units. Variable T_0 depends on the plant species that can be identified using ontologies to automatically assign T_0 values. As mentioned previously, the equations here should use concepts from domain ontologies instead of the generic variables used in the examples.

Figure 1 summarises the two case studies. On top, a model composed of different elements and below two specific models specifying the elements to be executed.

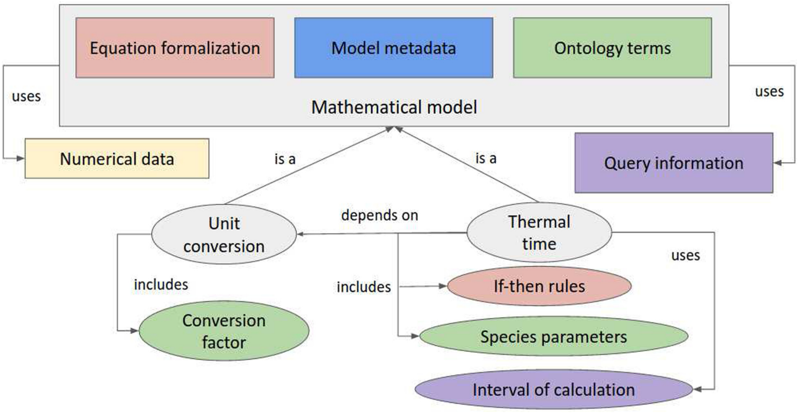

Fig. 1. Elements involved in each case study

5 Evaluation Plan

We will conduct some experiments to assess the efficacy of the framework for representing and computing the two case studies. For this aim, the experiments will require three main resources, (i) datasets, (ii) unit ontologies and (iii) a machine to perform computations. All the resources will be explained in this section.

In the unit conversion case study, two real-world datasets will be used. The first one contains plant measurements (leaf length and leaf width) annotated in different units. The second one contains weather measurements made by three different light sensors involving different light units and time granularities. The thermal time case study will use temperature data and plant annotations (e.g. events and species-specific factors). Datasets are annotated using the units of measurement ontology (UO).

For all the experiments data are stored in a GraphDB database and the experiments will be executed in an average machine with a Linux operating system, 16 GB RAM, and a processor of 1.8 GHz, 8-cores.

Assessing the equation representation. A qualitative evaluation comparing some representation methods will be proposed. This evaluation will consider criteria like the number of instructions necessary to express the case studies.

Assessing the unit conversion module. In this part we will evaluate the efficacy of the unit conversion module using the proposed case studies, and we will define the more appropriated ontology for unit conversion that does not affect the inference capabilities (e.g. such as ontology alignment).

Assessing the nested equation. The thermal time study case will be used to evaluate nested equations. Input data for the thermal time equation will be in different temperature units, thereby depending on the unit conversion module.

All the case studies will be evaluated considering the data volume and the equation complexity.

6 Conclusions and Lessons Learned

In this paper, we propose an ontological framework for representing and computing mathematical equations using Semantic Web technologies. In contrast to the state-of the art, this framework will offer an unified mechanism to represent common Plant Phenomics equations. By using this framework, we expect to have more linked models, more explainable equations, and a more effective use of unit ontologies. In this way, the neglected numerical relationships will be easier to express.

To assess the feasibility of the framework, the different experiments and case studies will evaluate specific computing boundaries. From these results, we will assess the performance of the model and the capability to express all the required functionalities. For each case study we will demonstrate that this representation facilitates the data exploitation and reduces the user's time-effort in favour of a simplified data retrieval process.

Although the case studies presented here belong to Plant Phenomics community (offering massive semantic datasets), the framework can be used for another domains dealing with numerical attributes and mathematical equations.

References

1. Beck, H., Morgan, K., Jung, Y., Grunwald, S., Kwon, H.y., Wu, J.: Ontology-based simulation in agricultural systems modeling. Agric. Syst. **103**(7), 463–477 (2010)
2. Bischof, S., Harth, A., Kämpgen, B., Polleres, A., Schneider, P.: Enriching integrated statistical open city data by combining equational knowledge and missing value imputation. J. Web Seman. **48**, 22–47 (2018)
3. Bischof, S., Polleres, A.: RDFS with attribute equations via SPARQL rewriting. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 335–350. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38288-8_23
4. Consortium, W.W.W., et al.: Sparql 1.1 overview (2013)
5. De Meester, B., Dimou, A., Verborgh, R., Mannens, E.: An ontology to semantically declare and describe functions. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenić, D., Auer, S., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9989, pp. 46–49. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47602-5_10
6. DuCharme, B.: Learning SPARQL: Querying and Updating with SPARQL 1.1. O'Reilly Media, Inc., Sebastopol (2013)
7. Gkoutos, G.V., Schofield, P.N., Hoehndorf, R.: The units ontology: a tool for integrating units of measurement in science. Database **2012**, bas03 (2012)
8. Hodgson, R., Keller, P.J., Hodges, J., Spivak, J.: Qudt-quantities, units, dimensions and data types ontologies. USA Available http://qudt. org March 156 (2014)
9. Hogan, A., Reutter, J.L., Soto, A.: In-database graph analytics with recursive SPARQL. In: Pan, J.Z., Tamma, V., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) ISWC 2020. LNCS, vol. 12506, pp. 511–528. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62419-4_29
10. Martín-Recuerda, F., Walther, D., Eisinger, S., Moore, G., Andersen, P., Opdahl, P.-O., Hella, L.: Revisiting ontologies of units of measure for harmonising quantity values – a use case. In: Janowicz, K., et al. (eds.) ISWC 2020. LNCS, vol. 12507, pp. 551–567. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62466-8_34
11. Neveu, P., et al.: Dealing with multi-source and multi-scale information in plant phenomics: the ontology-driven phenotyping hybrid information system. New Phytol. **221**(1), 588–601 (2019)
12. Parent, B., Turc, O., Gibon, Y., Stitt, M., Tardieu, F.: Modelling temperature-compensated physiological rates, based on the co-ordination of responses to temperature of developmental processes. J. Exp. Bot. **61**(8), 2057–2069 (2010)
13. Parsia, B., Smith, M.: Quantities in Owl. Clark & Parsia LLC, Washington, DC (2008)
14. dos Reis, M.G., Ribeiro, A.: Conversion factors and general equations applied in agricultural and forest meteorology. Agrometeoros **27**(2), 227–258 (2020)
15. Rijgersberg, H., Top, J., Wielinga, B.: Towards conceptual representation and invocation of scientific computations. Int. J. Seman. Comput. **6**(04), 447–489 (2012)
16. Rijgersberg, H., Van Assem, M., Top, J.: Ontology of units of measure and related concepts. Semant. Web **4**(1), 3–13 (2013)
17. Tardieu, F., Cabrera-Bosquet, L., Pridmore, T., Bennett, M.: Plant phenomics, from sensors to knowledge. Curr. Biol. **27**(15), R770–R783 (2017)
18. Wilkinson, M.D., et al.: The FAIR guiding principles for scientific data management and stewardship. Sci. Data **3**(1), 1–9 (2016)