Big Data to Enable Global Disruption of the Grapevine-powered Industries

D3.1 - Data Modelling and Linking Components

DELIVERABLE NUMBER	D3.1
DELIVERABLE TITLE	Data Modelling and Linking Components
RESPONSIBLE AUTHOR	Nikola Tulechki (ONTOTEXT)

Co-funded by the Horizon 2020
Framework Programme of the European Union

GRANT AGREEMENT N.	780751
PROJECT ACRONYM	BigDataGrapes
PROJECT FULL NAME	Big Data to Enable Global Disruption of the Grapevine-powered industries
STARTING DATE (DUR.)	01/01/2018 (36 months)
ENDING DATE	31/12/2020
PROJECT WEBSITE	http://www.bigdatagrapes.eu/
COORDINATOR	Nikos Manouselis
ADDRESS	110 Pentelis Str., Marousi, GR15126, Greece
REPLY TO	nikosm@agroknow.com
PHONE	+30 210 6897 905
EU PROJECT OFFICER	Ms. Annamária Nagy
WORKPACKAGE N. | TITLE	WP3 | Data & Semantics Layer
WORKPACKAGE LEADER	ONTOTEXT
DELIVERABLE N. | TITLE	D3.1 | Data Modelling and Linking Components
RESPONSIBLE AUTHOR	Nikola Tulechki (Sirma AI)
REPLY TO	nikola.tulechki@ontotext.com
DOCUMENT URL	http://www.bigdatagrapes.eu/
DATE OF DELIVERY (CONTRACTUAL)	31 July 2020 (M31)
DATE OF DELIVERY (SUBMITTED)	31 July 2020 (M31)
VERSION | STATUS	3.0 | Final
NATURE	Demonstrator (DEM)
DISSEMINATION LEVEL	Public (PU)
AUTHORS (PARTNER)	Vladimir Alexiev, Nikola Tulechki (ONTOTEXT)
CONTRIBUTORS	Franco Maria Nardini (CNR), Raffaele Perego (CNR), Nicola Tonellotto (CNR), Timos Lanitis, Giannis Stoitshs (Agroknow)
REVIEWER	Giannis Stoitshs (Agroknow)

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Initial draft	01/09/2018	Vladimir Alexiev (ONTOTEXT)
0.7	Complete draft	17/09/2018	Vladimir Alexiev (ONTOTEXT)
0.8	Input from partners	21/09/2018	Franco Maria Nardini (CNR), Raffaele Perego (CNR), Nicola Tonellotto (CNR), Pythagoras Karampiperis (Agroknow), Antonis Koukourikos (Agroknow)
0.9	Internal Review	28/09/2018	Franco Maria Nardini (CNR)
1.0	Final edits after internal review	04/10/2018	Vladimir Alexiev (ONTOTEXT)
1.1	Update draft	18/09/2019	Nikola Tulechki (ONTOTEXT)
1.2	Complete update draft	04/12/2019	Nikola Tulechki (ONTOTEXT)
2.0	Updated Version	30/12/2019	Nikola Tulechki (ONTOTEXT)
2.1	Update draft	1/01/2020	Nikola Tulechki (ONTOTEXT)
2.9	Internal Review	31/07/2020	Giannis Stoitshs (Agroknow)
3.0	Updated Version	31/07/2020	Nikola Tulechki (Sirma AI)

PARTICIPANTS		CONTACT
Agroknow IKE (Agroknow, Greece)		Nikos Manouselis Email: nikosm@agroknow.com
Ontotext AD (ONTOTEXT, Bulgaria)		Todor Primov Email: todor.primov@ontotext.com
Consiglio Nazionale Delle Richerche (CNR, Italy)		Raffaele Perego Email: raffaele.perego@isti.cnr.it
Katholieke Universiteit Leuven (KULeuven, Belgium)		Katrien Verbert Email: katrien.verbert@cs.kuleuven.be
Geocledian GmbH (GEOCLEDIAN Germany)		Stefan Scherer Email: stefan.scherer@geocledian.com
Institut National de la Recherché Agronomique (INRA, France)		Pascal Neveu Email: pascal.neveu@inra.fr
Agricultural University of Athens (AUA, Greece)		Katerina Biniari Email: kbiniari@aua.gr
Abaco SpA (ABACO, Italy)		Simone Parisi Email: s.parisi@abacogroup.eu
SYMBEEOSIS EY ZHN S.A. (Symbeeosis, Greece)		Konstantinos Rodopoulos Email: rodopoulos-k@symbeeosis.com

ACRONYMS LIST

AEO	Agricultural Experiments Ontology
AFEO	Agri-Food Experiment Ontology
AGRO	AgroKnow
AGRO	Agronomy Ontology
AgroBio	Agronomy and Biology data
AT	Agricultural Technology Ontology
AUA	AGRICULTURAL UNIVERSITY OF ATHENS
BCO	Biological Collection Ontology
BFO	Basic Formal Ontology
ChEBI	Chemical Entities of Biological Interest
CLO	Cell Line Ontology
CO	CropOntology: a group (set) of ontologies for specific crops
CO_320	CropOntology: Rice
CO_321	CropOntology: Wheat
CO_322	CropOntology: Maize
CO_356	CropOntology: Vitis (grapes/viticulture)
CO_357	CropOntology: Woody Plants
CSV	Comma-Separated Values
CUBE	W3C ontology for representing multidimensional data cubes
DC	Dublin Core (elements)
DCT	Dublin Core Terms
DOID	Human Disease Ontology
EBI	European Bioinformatics Institute
EC	Electrical conductivity
ECA	Eddy Current Array
EDAC	Earth Data Analysis Center (data produced by Earth, Life and Semantic Web project)
EFO	Experimental Factor Ontology
EM-38	A handheld Geonics electromagnetic soil conductivity meter
EMI	Electromagnetic Induction: used in soil conductivity sensors (see also ECA)
ENVO	Environment Ontology
EO	Environment Ontology
eyeball	A Jena tool for RDF semantic validation (e.g. that no unknown terms are used)
FOODON	Food Ontology
GeoSPARQL	Geographic extensions to SPARQL. Defines representing features, geometries (e.g. asWKT) and spatial relation predicates (e.g. sfContains)
GIS	Geographic Information System

GODAN	Global Open Data for Agriculture and Nutrition
GPS	Global Positioning System
GraphDB	Semantic repository (database) by ONTO
grlc	Git Repository Linked data API Constructor
HDOP	Horizontal Dilution of Precision of a GPS reading
HTML	W3C HyperText Markup Language
IAO	Information Artifact Ontology
INRA	Institut national de la recherche agronomique
LAI	Leaf Area Index
LIRMM	Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier
LOV	Linked Open Vocabularies, a site for discovering ontologies
MMO	Measurement Methods Ontology
NASA	National Aeronautics and Space Administration
NCBITaxon	NCBI Taxonomy
NDRE	Normalized Difference Red Edge
NDVI	Normalized Difference Vegetation Index
NIR	Near-infrared spectral region
NIRi	Incident radiation of the Near-InfraRed spectrum
NIRr	Reflected radiation of the Near-InfraRed spectrum
OBO	Open Biological and Biomedical Ontology
OEPO	Ontology for Experimental Phenotypic Objects
OFPE	Ontology for Food Processing Experiment
OLS	Ontology Lookup Service
OWL	W3C Web Ontology Language, a more complex language for describing ontologies
OxO	Ontology Xref (Cross-Reference) Service
PATO	Phenotypic Quality Ontology
PCO	Population and Community Ontology
PECO	Plant and Environmental Conditions Ontology
PO	Plant Ontology
QB	See CUBE
QUDT	NASA Quantities, Units, Dimensions, and Types Ontology
R	Red spectral region
RDA	Research Data Alliance
RDBMS	Relational Database Management System
RDF	W3C Resource Description Framework, the semantic web data model
RDF Shapes	A way to describe semantic data Application Profiles. Two approaches are SHACL and ShEx
rdfpuml	ONTOTEXT tool for translating RDF to PlantUML, a textual notation for generating UML diagrams

RDFS	W3C RDF Schema, a simple language for describing ontologies
RE	Red-Edge spectral region (spectrum centred around 715 nm)
REDi	Incident radiation of the red spectrum
REDr	Reflected radiation of the red spectrum
REST	Representational State transfer
RIOT	RDF Input/Output Tool, part of Apache Jena. Includes RDF syntax validation
RO	Relations Ontology
SDGIO	SDG-Interface Ontology
SHACL	Shapes Constraint Language, a W3C Recommendation
ShEx	Shape Expressions, a W3C community specification
SKOS	Simple Knowledge Organization System, an ontology for describing thesauri
SPARQL	SPARQL Protocol and RDF Query Language
TO	Trait Ontology
TSV	Tab-Separated Values
Turtle	Terse RDF Triple Language
UML	Unified Modeling Language
UO	Units Ontology
URL	Uniform Resource Locator
VANN	Vocabulary for annotating vocabulary descriptions
W3C	World Wide Web Consortium
WKT	Well-Known Text, a format for describing feature geometries
WP	Work package
XML	W3C eXtensible Markup Language
XO	Experimental condition ontology
XSD	XML Schema Datatypes

EXECUTIVE SUMMARY

WP3 Data & Semantics Layer is a core WP of the project. Within this WP3, task T3.1 Data Modelling over Big Data Infrastructures has the following objectives:

- Explore partner data
- Define competence questions that the data should be able to answer
- Study relevant AgroBio ontologies
- Define semantic modelling principles and specific models
- Study user (researcher) requirements for discovering ontologies, mapping data, aligning data, etc.
- Implement or adopt tools for these requirements

The document has the following structure:

- Chapter 1 Introduction describes fundamental AgroBio data (observations and measurements), outlines the ontological representation of measurements, mentions possible alternatives (e.g. following existing AgroBio patterns vs using the W3C CUBE ontology), describes the steps of semantic data integration, and provides links to consortium resources related to the task.
- Chapter 2 Chosen OntologiesLists the 3 ontologies chosens to form the core of the BDG Semantic Data model
- Chapter 3 Specific Project Data discusses how we adapt the BDG data model to the specifics of the tasks at hand and how we adapt the vast and heterogeneous datasets from the consortium to the harmonized model. We present the data processing requirements and data access requirements based on the quasi-totality of the data collected in the project. We also present several use-cases where particular data issues specific to the project are addressed on a fine-grained scale.
- Chapter 4 Conclusions provides conclusions and a bibliography.

Deliverable D3.1 Data Modelling and Linking Components had 3 iterations at M9, M21, M30. This is the final version.

TABLE OF CONTENTS

1. INTRODUCTION

Deliverable D3.1 is defined as "A tool for creating, maintaining and linking semantic data, customized to serve the needs of the relevant grapevine-powered industries".

This deliverable is part of task T3.1, which is described as:

- Work on the task will initially focus on the provision of a basic integrated model for grapevine-powered industries, facilitating interoperability between the data assets of the different industries and incorporating open data from third-party entities that pertain to use cases specified in T2.1.
- Consequently, the BigDataGrapes model is published as an ontology, and linked with external conceptualizations via a semi-automatic process. The scalable ontology alignment systems envisioned in the project will be implemented and applied for linking the model with significant specifications, either general purpose or domain-specific.
- Furthermore, the task has produced the necessary tools and components for carrying out the aforementioned processes, i.e. an environment for building, reusing and linking disparate conceptualizations.

1.1 FUNDAMENTAL AGROBIO DATA: MEASUREMENTS

The basic data that needs to be represented by the project is AgroBio **measurements/observations**: the measurement of some traits of some objects (e.g. soil or a particular crop) using a certain method, technique, equipment, units of measure, time, place, etc. This sounds simple, but it involves a number of data items to give the observations context and meaning.

We can illustrate it with an example regarding measuring a basic variable: plant height.

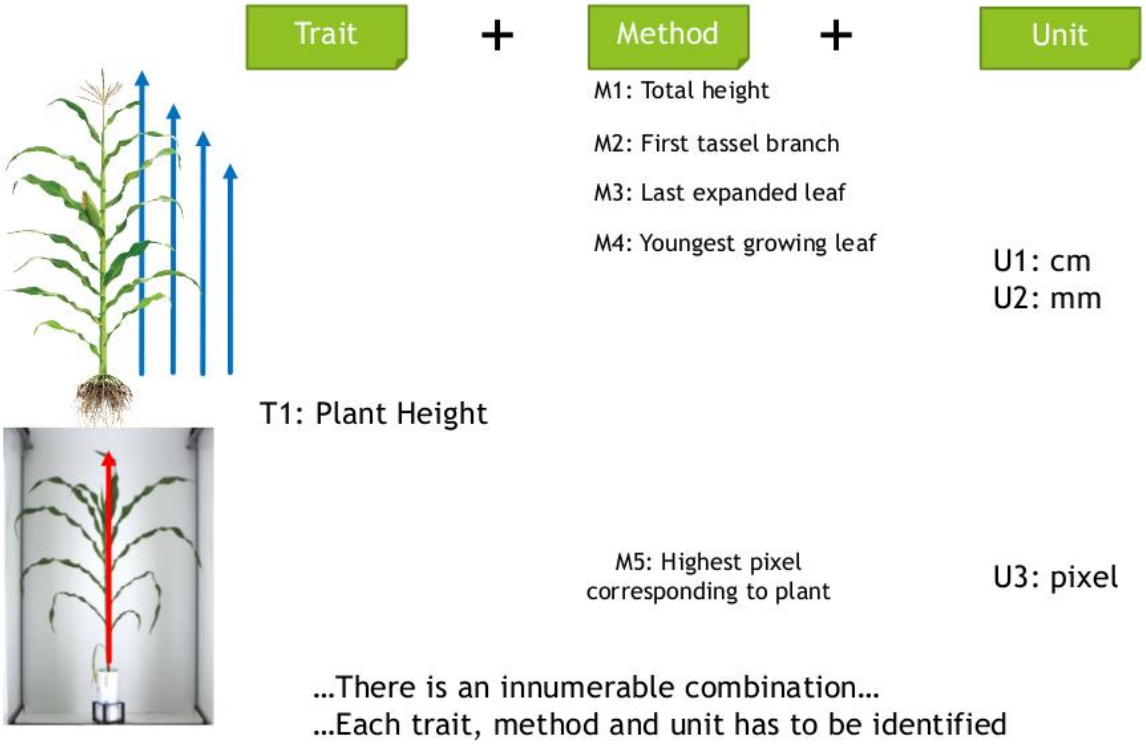


Figure 1 Basic Measurement: Plant Height

A measurement involves the following items:

- **Entity**: thing being measured or observed, such as the soil, weather (temperature, precipitation),a particular crop or plant variety, harvest parameters, etc.
- **Quality**: what is being measured
- **Trait** = entity + quality: what quality of which entity
- **Method**: what exactly are we measuring (e.g. height to youngest growing leaf or total plant height) and how (instrument, technique, etc).
- **Unit** of measure: may include fundamental units (e.g. Meter, Second), derived units (e.g. m/s) or a variety of countable units (e.g. pixels, count, etc).
- **Variable** = trait + method + unit: provides the detailed meaning of the measurement.
- **Context**: circumstances of the observation, e.g. GPS location, estate/plot/subplot, depth of measurement (for soil), datetime, etc. May also include qualifiers, e.g. instrument, which satellite provided GPS location, precision, instrument status at the time the reading was taken, whether there's a metal pole at the location (which makes a conductivity measurement invalid), who took the reading, etc.
- **Value**: the number that was measured/observed
- **Observation** = variable + context + value: all details about a single observation point.

Please note that it is a common practice to measure several variables of the same entity at once (in the same context). Combination instruments make this possible, and it saves time and effort. This leads to the need to share entity and context between observations, which affords the following efficiencies:

- Easier correlation of related observations
- More economical data representation

1.2 Ontological Representation of Measurements

There are various different ways to represent AgroBio measurements using the RDF semantic data model, two of which are:

- Using some of the established AgroBio ontologies. The next chapter introduces such ontologies, but we give below a motivating example of measuring plant height.
- Using the W3C CUBE ontology for representing multidimensional observations, which is described in the next subsection.

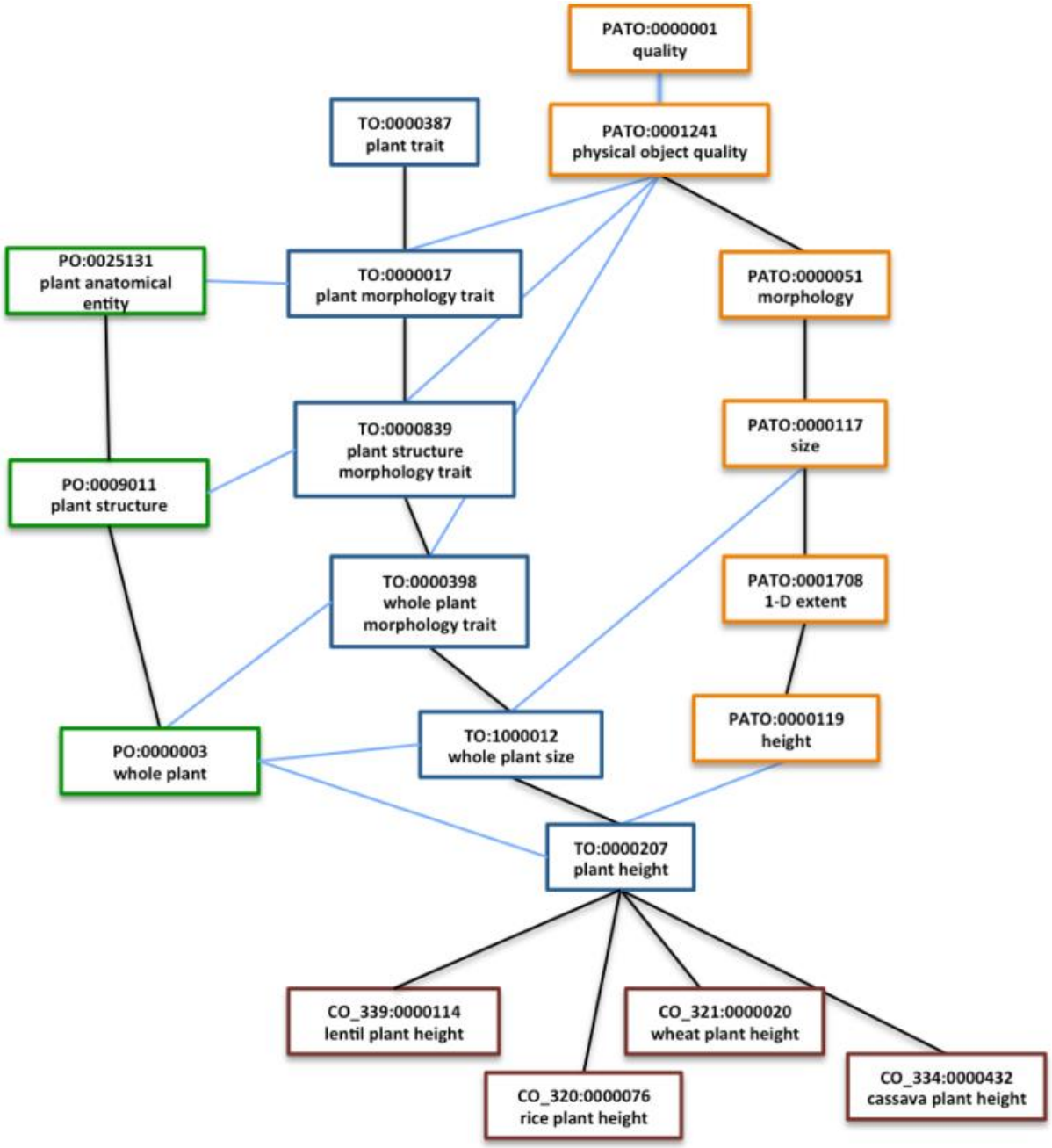


Figure 2 Semantic Classes for Representing Plant Height

- The Phenotypic Quality Ontology (PATO, orange chain) is used to classify the trait (considered as a physical object quality) in a subsumption hierarchy.
- The Plant Ontology (PO, green chain) is used to describe plant anatomical parts, i.e. sub-entities that can be measured
- The Trait Ontology (TO, blue chain) is used to describe a particular plant morphology, i.e. tie the trait to an anatomical part
- The Crop Ontologies (CO_nnn, brown subclasses) specialize the trait to particular crops or varietals

We identify a problem with tying up a trait that is quite universal (height) to such a specific degree. A height is a height, no matter whether you measure lentils, rice, wheat, any plant, or a skyscraper. It's true that measurement methods often vary per entity, i.e. are applicable only to certain kinds of entities. But that

restricted applicability does not mean that every variable should be replicated to every crop that it applies to, which leads to a combinatorial explosion.

We locate this problem (improper level of abstraction) many times in the Crop Ontologies, for example:

- Normalized difference vegetation index (NDVI) is defined in CO_322 Maize, so we can't use it for Grapes. NDVI is not defined in CO_356 Vitis. But rather than replicating NDVI in Vitis, its proper place is in the general Crop Ontology (CO), not a sub-ontology of CO.
- The "grams" unit of mass is bound to some Woody Plant trait, so we can't use it for Grapes.

We believe that by "regrouping the factors" in the equations outlined in section 1.1, we can avoid such combinatorial explosions:

- **Current:** Trait = entity + quality; Variable = trait + method + unit; Observation = variable + context + value
- **Future:** Variable = quality + method + unit; Observation = entity + variable + context + value. Quality defines which entities it is applicable to but is not subjugated to Entity.

2. CHOSEN ONTOLOGIES

2.1 AFEO

The The Agri-Food Experiment Ontology[1] (AFEO) is a specific ontology representing the transformation processes involved in food production. A considerable amount of work has been dedicated by one of the project partners, INRA, in modelling the transformation of grapes in wine using this ontology.

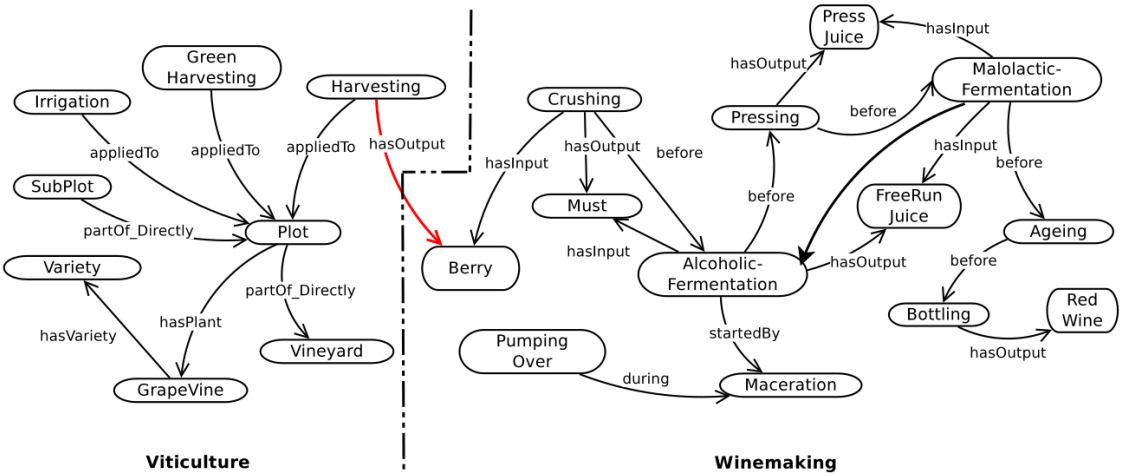


Fig. 3. A part of AFEO demonstrating viticulture and winemaking integration.

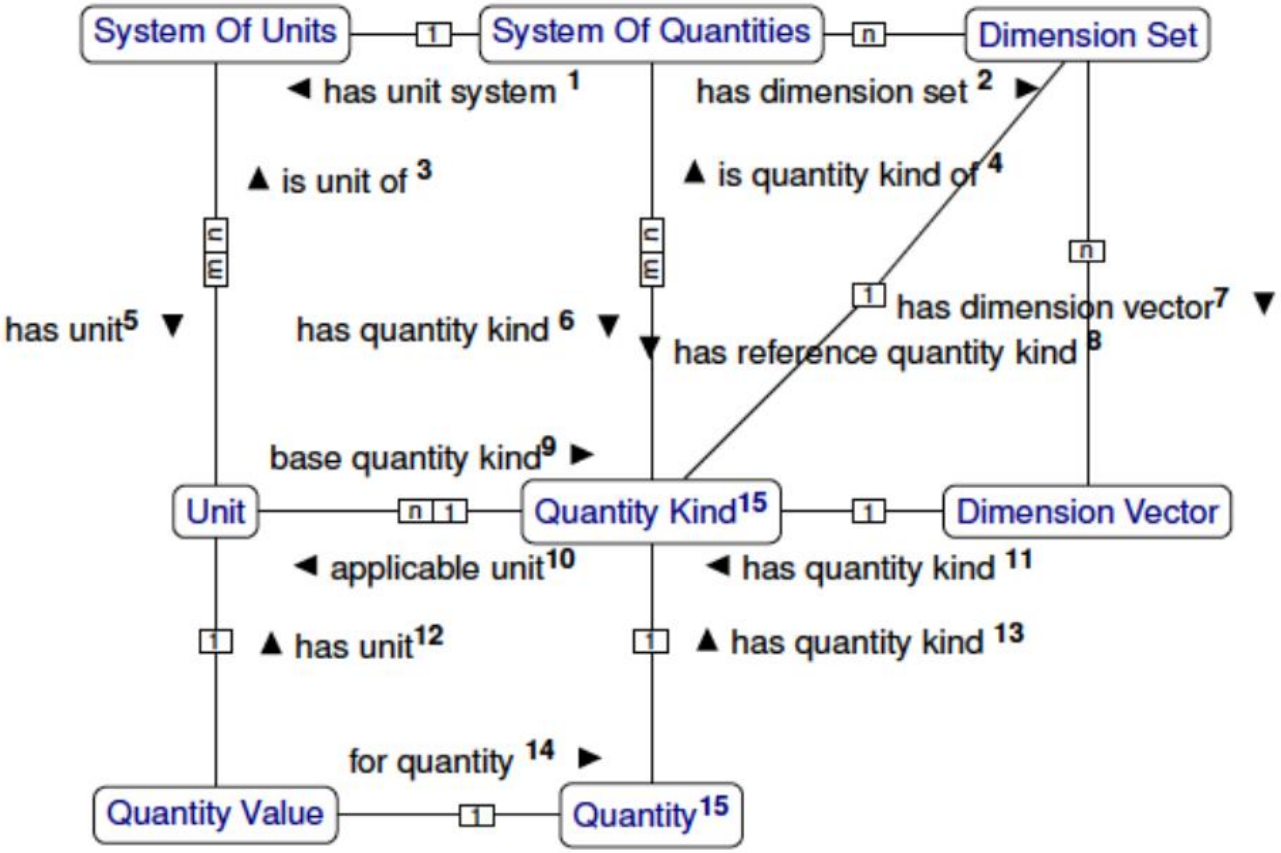
2.2 QUDT 2.0

Th QUDT ontology is used to represent units of measurement. It uses a dimensional approach relating each unit to a system of base units using numeric factors and a vector of exponents defined over a set of fundamental dimensions

Figure 1[2] is very useful, showing the main 8 classes and relations between them, and describes the rationale for the cardinalities. Also see the diagrams in the QUDT Overview, which show the attributes of each class, but can't fit all classes on one diagram.

[1] A generic ontological network for Agri-food experiment integration – Application to viticulture and winemaking https://doi.org/10.1016/j.compag.2017.06.020

[2] http://www.qudt.org/release2/qudt-catalog.html



2.3 W3C CUBE

The W3C CUBE ontology (QB) captures multidimensional observations (data cubes) using the following terminology (in bold). We roughly map these QB terms to the data items discussed in previous sections. The first 3 are called "components".

- **Dimension:** entity, quality, method
- **Attribute:** unit, context
- **Measure:** value
- **Observation** = Dimensions + Attributes + Measures

QB defines what components are expected in a specific *qb:DataSet* by using a *qb:DataStructureDefinition*. QB provides some flexibility that affords data efficiencies, and avoiding combinatorial explosion:

- QB allows using several dimensions per observation, without tying them up together. E.g. you can use 3 dimensions entity="plant", quality="height", method="whole height"
- We could also use several measures per observation (e.g. as taken by a combination instrument), although this is less commonly used.

- One could split a dataset into Slices (or other kinds of ObservationGroups) by fixing some of the dimensions, so one doesn't need to repeat them for every observation.

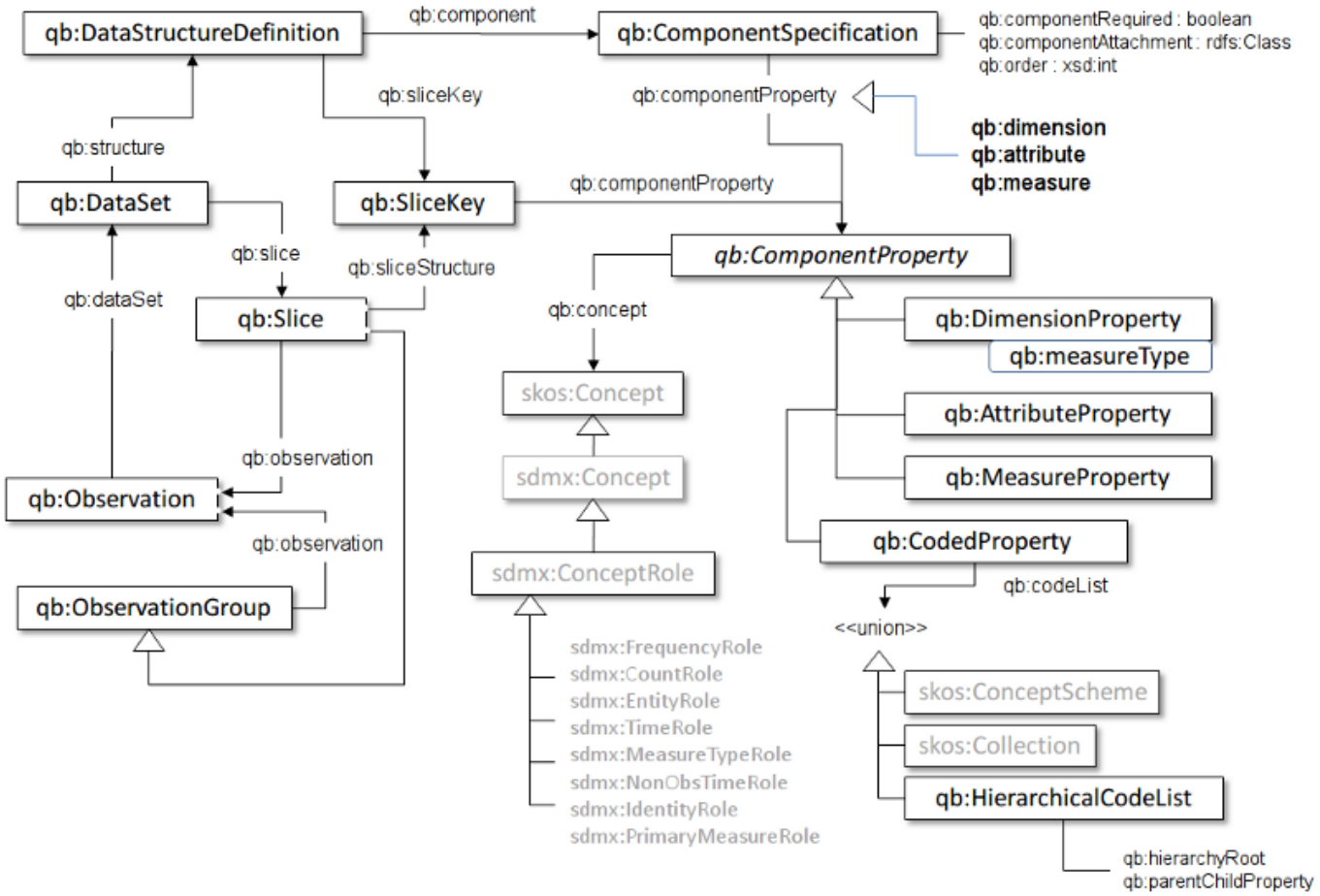


Figure 3 W3C CUBE Ontology

2.3.1 Creating New Ontology Terms

We will often need to define new terms. Some examples:

- There is nothing about "the number of grapes" in Vitis. We could create this trait using "grape" in Agricultural Experiments Ontology (AEO) and "amount" in Phenotypic Quality Ontology (PATO).
- CV1m: "soil conductivity at depth 1 meter in millisiemens per metre (mS/m)" encapsulates 4 factors: entity: soil; variable: electrical conductivity; context: depth=1m; unit: mS/m. There's nothing about soil conductivity in CO or Vitis. The closest we can find is ENVO:09200016 conductivity of soil. We could use that, and then construct extra terms to specify the unit (mS/m) and context (1m vs 0.5m depth)
- The closest we can find to specific-spectrum measurements (Near-infrared, Red, Red-Edge) is FIX:0000641, but that only has "far-, mid- and near-infrared spectroscopy". For some AUA data (see sec 4.2.2) we need to express more specific spectrum measurements.
- We can find NDVI in CO_322 Maize, but not in CO_356 Vitis. Should we create another term "NDVI for grapes", thus perpetuating the increase of number of terms? We believe that CO should define NDVI in a crop-independent manner, then we can just use that rather than making a number of crop-dependent terms.

However, the traditional approach of creating new terms for every combination will possibly lead to a combinatorial explosion in the number of terms. If we vary any one of these factors, we will need another term.

Our solution to the problem of a potential combinatorial explosion variables is to always aim at the most generic representation of variable and then derive more specific ones from them if they are needed. We always keep the link between the generic and specific variables via a *bdg:derivedForm* predicate. We materialize the aspect anchoring the specificity via a number of predicates such as

- ***bdg:hasFeatureOfInterest*** for variables measuring a specific feature such as soil or air
- ***bdg:measurementContext*** for variables capturing measurements within a set of parameters such as wind speed from a specific direction.
- ***bdg:statisticalSummary***: when aggregated values are summarized using a statistical operation such as min, max, average etc...

This allows us to compute correspondence for values and maximizes the interoperability between datasets. Our system if variables is described in detail in the data model section on GitHub and currently the vocabulary contains 225 distinct variables. New elements can be added easily and the vocabulary is evolving all the time.

We can illustrate the progressive specification of variables with the set of variables regarding moisture, where currently we maintain a 3-level hierarchy.

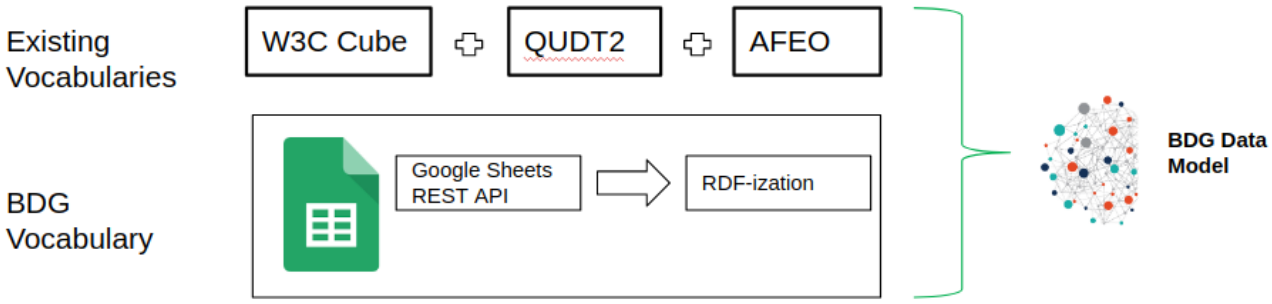
2.3.2Creating the BDG Semantic Model

The *BDG Semantic Model* is a harmonized data shema aiming at rendering interoperable as much of the partner data as possible. The model is available in the BDG Github.

The model is built around the 3 established ontologies, to which we add the BDG Specific vocabulary, built as part of the project. The ontologies are:

- QUDT2 for units of measurements
- AFEO - a wine processing specific ontology developed by INRA
- W3C Cube, for representing multidimensional observations

These 3 ontologies cover the core issues and processes needed to describe the data generated by the panthers, but finer specificities require an extension to the classes and properties provided by them. This is handled by the BDG Vocabulary.



2.3.3 Collaboratively generating the BDG specific vocabulary

While the core BDG semantic data model provides the framework onto which we map the partner's data, the project also requires a substantial number of specific entities in order to efficiently describe it. This is the BDG specific vocabulary. These entities are of several types:

- measures and variables for describing the analytical data
- categories such as grape varieties
- geographical data such as the plots and subplots of the actual fields
- units of measurement not covered by QUDT2

It is easy to see why the BDG vocabulary is a critical part of the BDG Semantic data model and it's generation was a challenging task. The challenge comes from the fact that the specificities of each need to be discussed with the project partners and the entities themselves need to be crafted in a way that they make sense to the end users. For us as ontologists this meant that we had to come up with a methodology allowing a maximum of interactivity over very granular instances. In other words we had to be able to potentially discuss the specificities of every entity with a project partner and incorporate the resulting decision in a subsequent version of the vocabulary. This, coupled with the distributed nature of the project proved a challenging task and we solved it by devising a mechanism for vocabulary generation based on shared Google Sheets. This process is pioneered in the GLAM domain and we transposed it to data driven agriculture.

The principle is as follows. The implementation is on the project github. This is the rationale behind it. The master data for the vocabulary resides in a shared Google Sheet. Google sheets is a marvelous tool which is adapted to handle much of the difficulties of remote collaboration:

- It discharges us of the responsibility to watch for data integrity and data versioning.
- it maintains a full log off all user actions.
- It is trivial to rollback to the previous version or to attribute an error to a user. This allows us to be sufficiently agile while ensuring that the final data meets the rigorous consistency required for generating RDF.
- it has a collaboration mechanism with comments and tasks on specific cells allowing for quick and efficient remote communication over specific data items.
- linking to cells and ranges also helps communication .

Data from the google sheet is consumed via *HTTP* in a plain tabular format CSV and is converted to rdf using a custom SPARQL query and the TARQL tool. TAQRL (or tabular sparql) allows us to define a mapping from tabular to RDF in a sparql query. This allows us to maintain the mappings in a entirely declarative fashion having the different elements independent of each other and in version control (github)

Here is an example of how the Vocabulary generation works:

This line in the variable sheet defines the variable density_MAX

uri	label	unitMeasure	codedList	featureOfInterest	measurementContext	derivedFrom	statisticalSummary
density_MAX	**Maximum Density**	**bdg-unit:G-PER-LT**				**density**	**Maximum**

It gives us the URI, the human readable label, the unit in which density is measured (grammes per liter), as well as additional elements such as the fact that it is derived from the more abstract measure density and that it is built via the statistical operation "maximum"

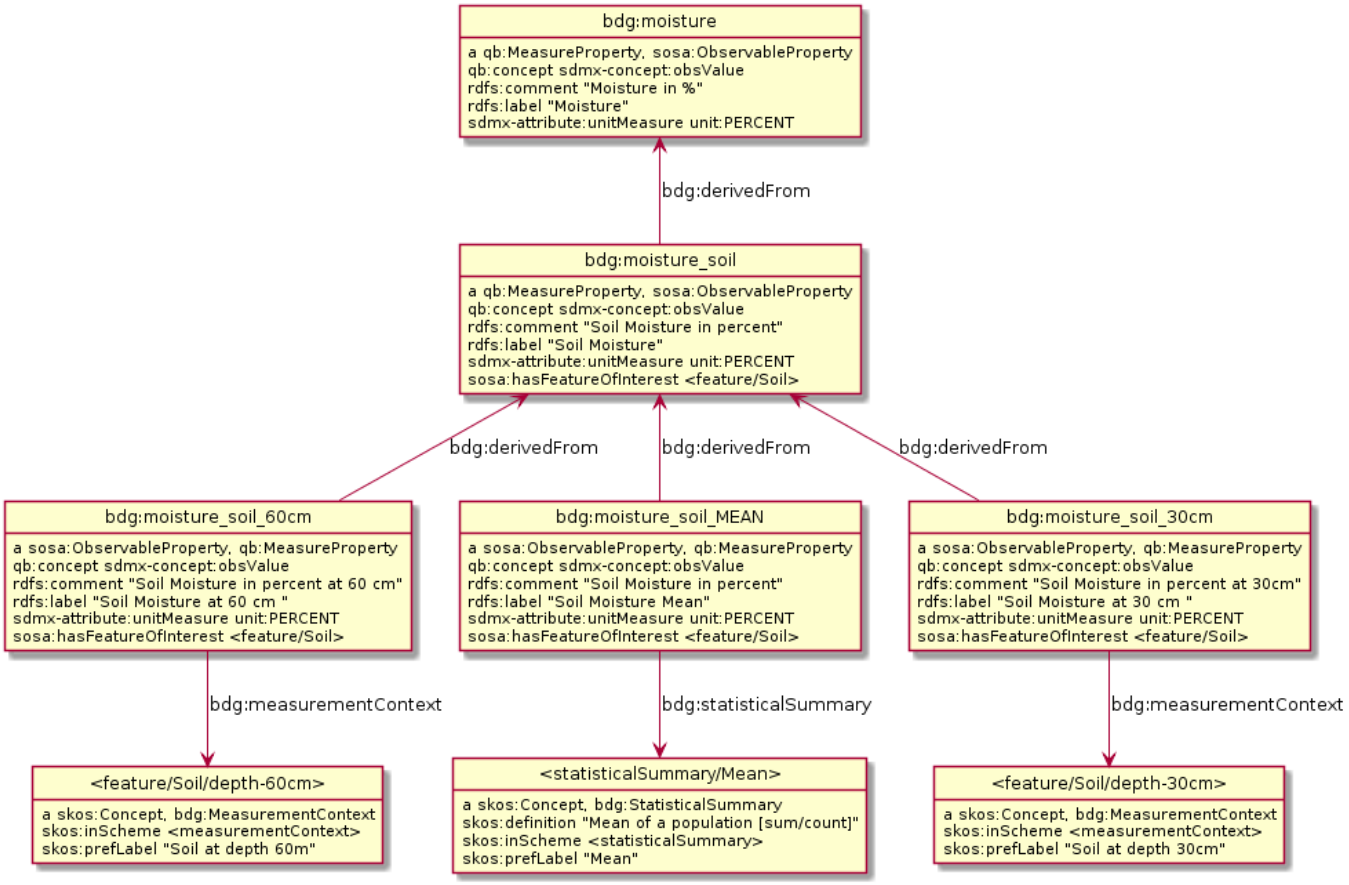
This line is piped through this mapping function, defined as a SPARQL Construct query. Here for clarity we will only show the CONSTRUCT part of the query:

```
CONSTRUCT {
    ?URI a qb:MeasureProperty, sosa:ObservableProperty, ?CODED_PROP_TYPE;
        rdfs:label                  ?label ;
        rdfs:comment                ?description ;
        sdmx-attribute:unitMeasure  ?UNIT ;
        sosa:hasFeatureOfInterest   ?FEATURE ;
        bdg:measurementContext      ?MEASUREMENT_CONTEXT ;
        bdg:derivedFrom             ?DERIVED_FROM ;
        bdg:statisticalSummary      ?STATSUM ;
        qb:codeList                 ?CODEDLIST ;
        qb:concept                  sdmx-concept:obsValue ;
        rdfs:range                  ?CODEDCLASS ;
        ?EXTRA_P                    ?EXTRA_O ;
.
}
```

This will produce the following resulting RDF triples

```
bdg:density_MAX rdf:type        qb:MeasureProperty ;
    rdf:type                sosa:ObservableProperty ;
    rdfs:label              "Maximum Density" ;
    sdmx-attribute:unitMeasure  bdg-unit:G-PER-LT ;
    bdg:derivedFrom         bdg:density ;
    bdg:statisticalSummary  <http://data.bigdatagrapes.eu/resource/statisticalSummary/Maximum> ;
    qb:concept              sdmx-concept:obsValue .
```

The resulting RDF is thus ensured to be always consistent and easily generated. Given that the size is not



First is the most generic variable, *bdg:moisture* it can be used in cases when only one feature is measured and the feature is specified at the dataset level (cf Infra). In cases where we need to specify the feature (in this case soil) at the variable level, we derive the more specific *bdg:moisture_soil* variable. It concerns measurement of the moisture of the soil. From it, we can then derive even more specific variables concerning different modes to measurement of the moisture of the soil, such as in this case measurements at different depths.

3. SPECIFIC PROJECT DATA

This section introduces data, the modelling methodology, the data processing requirements and data access requirements that are specific to the project.

3.1 RESOURCES

We have created a public GitHub repository https://github.com/BigDataGrapes-EU/ontology for WP3 work. It contains the bulk of the resources that are mentioned in this section. It has the following folders:

- data: semantic data (for now mostly samples)
- ttl: relevant ontologies, converted to turtle (and added prefixes) for easier reading
- misc: ontology materials in miscellaneous formats (eg xlsx, obo)
- notes: various notes on ontologies and data. In particular, see README: Github preview and Rendered HTML version

3.2 COMPETENCE QUESTIONS AND MODELLING METHODOLOGY

Developing semantic models or ontologies of some domain hinges on several aspects:

- What **data** you have
- What data **needs** you have, or what questions the data should be able to answer

Given the abundance of available data and the over-abundance of AgroBio ontologies, the latter aspect is crucial in order to keep the modelling effort focused. It should drive the following tasks:

- Seeking more data for specific questions
- Deciding which ontologies to involve and whether more ontological work is needed
- Structuring the data in an appropriate form (semantic modelling)
- Defining data tasks: conversion, clean-up/filtering, discretization…
- Creating sample queries to help data consumers

3.2.1 Data Domains

Data Domains defines the sort of data that we need to represent.

- Observations: when (timestamp), where (geo-reference), what (measure, dimension, attribute, and observation). These are the core of the data processed so far and are represented using the W3C CUBE Ontology.
- Estates and plots, including geospatial data, modelled using the GeoSPARQL extension, allowing native querying of topological relationships such as inclusion, distance and adjacency
- Measurement equipment
- Experiments
- Static nomenclature data, e.g.: varieties, types of measurement, etc
- Photos and other images
- Grape and wine operations (AFEO)

3.2.2 Data Questions

Data questions are at the core of our modelling methodology. We maintain a list of competency questions, questions that still need to be elaborated and validated by the partners and uses cases, to ensure they indeed are valuable research questions. Some examples of questions are, while the full list is visible at BigDataGrapes WP3: Competence Questions

- Can I retrieve the sub-plots for a given plot?
  - What's the hierarchy? Estate>Plot>Subplot?
  - Do we need/have GeoSPARQL regions for these plots? At what level?
- Which varieties are cultivated in a given plot?
- Can I retrieve weather data for a given plot?
- Which varieties are cultivated in a soil with certain characteristics?
  - How many characteristics are relevant? 10, 100, 500?
  - How are these characteristics grouped?
  - Is it meaningful to know just a few of them, or do you need to know all of them?
  - To select the optimal variety, we guess that not only the soil, but also the weather, precipitation patterns and elevation are important?
  - Will the answer be a sort of decision tree?
- Can I retrieve the origin locale for a given test sample?
  - Most probably, if we can't localize a sample, it is useless. Clarifications:
  - Does sample mean observation, or actual specimen/soil sample?
  - Does locale mean latitude/longitude/elevation? Or can it also mean specialized context, e.g. depth of a soil measurement?
  - Is localization qualifier data important (e.g. satellite number, quality of reception)?
- Can I retrieve images of a plot from which a sample was taken, at the time of collection?
  - Do we need photos of the crop at the actual time of sample taking, or only of the plot?
- Can I retrieve historical yield results for a plot (providing a timestamp)?
- Can I retrieve historical weather data for a plot (providing a timestamp)?
- Find under-performing land plots
- Is there a correlation between soil conductivity and vegetation?

Once established the list of questions is the pivot of the data modelling workflow. For each question we:

- Establish the relevant dataset(s) containing the answer
- Validate that the datasets are modelled in a satisfactory manner, so that the model supports finding the answer to the question within the Knowledge Graph.
- Track that the relevant datasets are ingested
- Track that datasets are relevant.

3.2.3Dataset relevancy and management of modelling commitment

Modelling overcommitment occurs when one attempts to reflect every single nuance of a source dataset in the target data model. When Integrating multiple datasets in a single harmonized Knowledge Graph, modelling overcommitment can rapidly lead to unnecessary complexity of the data model and severely impact processing and data consumption further on. Given that data modelling is done first, and all the subsequent tasks (from data processing to application design) depend critically on output of the modelling task, avoiding overcomplicating the model is important.

However, especially for non specialists of a given domain (such as viticultura), it is difficult to know in advance what is important and what is not. For that reason we rely upon the list of competency questions and only model and process a given dataset when it is mentioned in at least 1 competency question. Likewise we do not model aspects of the data not covered by the questions. Coupled with the fact that the whole methodology is based on shared google sheets, we establish an iterative process where we can repeatedly and efficiently communicate with stakeholders over fine points of modelling and wording of the questions. This proved also efficient to materialize the data questions themselves, as they are the prerequisite to working on a given dataset.

3.2.4 Semantic Data Integration

Semantic Data Integration has proven itself in the last 10 years as one of the best ways to integrate diverse data across institutions and enterprises, and to leverage datasets available in the LOD cloud. Life Science and Biology researchers were one of the early adopters of semantic web techniques, and by now they have found a wide following also in the Agricultural community, who in many cases leverage ontologies developed in the Bio community.

Semantic Data Integration is a holistic activity that aims to harmonize data from different providers, convert it to a semantic form, match (coreference) instances about the same entity coming from different datasets, and create an integrated Knowledge Graph of data in a domain. It involves the following steps, which have informed and will continue to inform WP3 activities:

- Get sample tabular data from partners
- Get sample RDF data from partners
- Analyse the data
- Define competence questions and other data requirements
- Research ontologies sent by partners and other related ontologies
- Report ontology and instance data errors to partners and the AgroBio ontology
- Ontology engineering: selection, combination and extension of ontologies

The consortium's progress to date is somewhere at this point.

- Discuss how to represent various data aspects with partners: estates/plots, measurements/observations, equipment, experiments, etc
- Create a semantic model with rdfpuml and text narrative (see the euBusinessGraph Semantic Model as an example)
- Get the model approved by all partners
- Create application profiles and/or RDF shapes (SHACL and/or ShEx) for validation of semantic data for conformance to the model
- Define URL design and policies
- Semantic conversion using appropriate tools depending on source (CSV/TSV tabular, RDBMS, XML)
- Semantic alignment and instance matching
- Data validation and data quality management/measurement
- Implement proper semantic publishing and content negotiation
- Design and implement data update flows
- Create sample queries
- Deploy predefined queries as REST services
- Create a dataset catalogue and conversion tracking methodology based on competency questions (see section 3.1)

- Process and prepare for integration 44 distinct datasets

3.3 SPECIFIC PROJECT DATA AND CHALLENGES

Besides the common and shared data model, we have performed a thorough analysis of the partners data and specific data related issues such as format incompatibilities and alignment issues. Here are the main parts:

3.3.1 INRA Semantic Data analysis:

INRA has submitted some sample semantic data in Github folders data/INRA/data[345].
data3 and data4 are illustrated as follows. INRA data is the top 4 nodes, and the bottom 4 nodes are from the Vitis ontology.

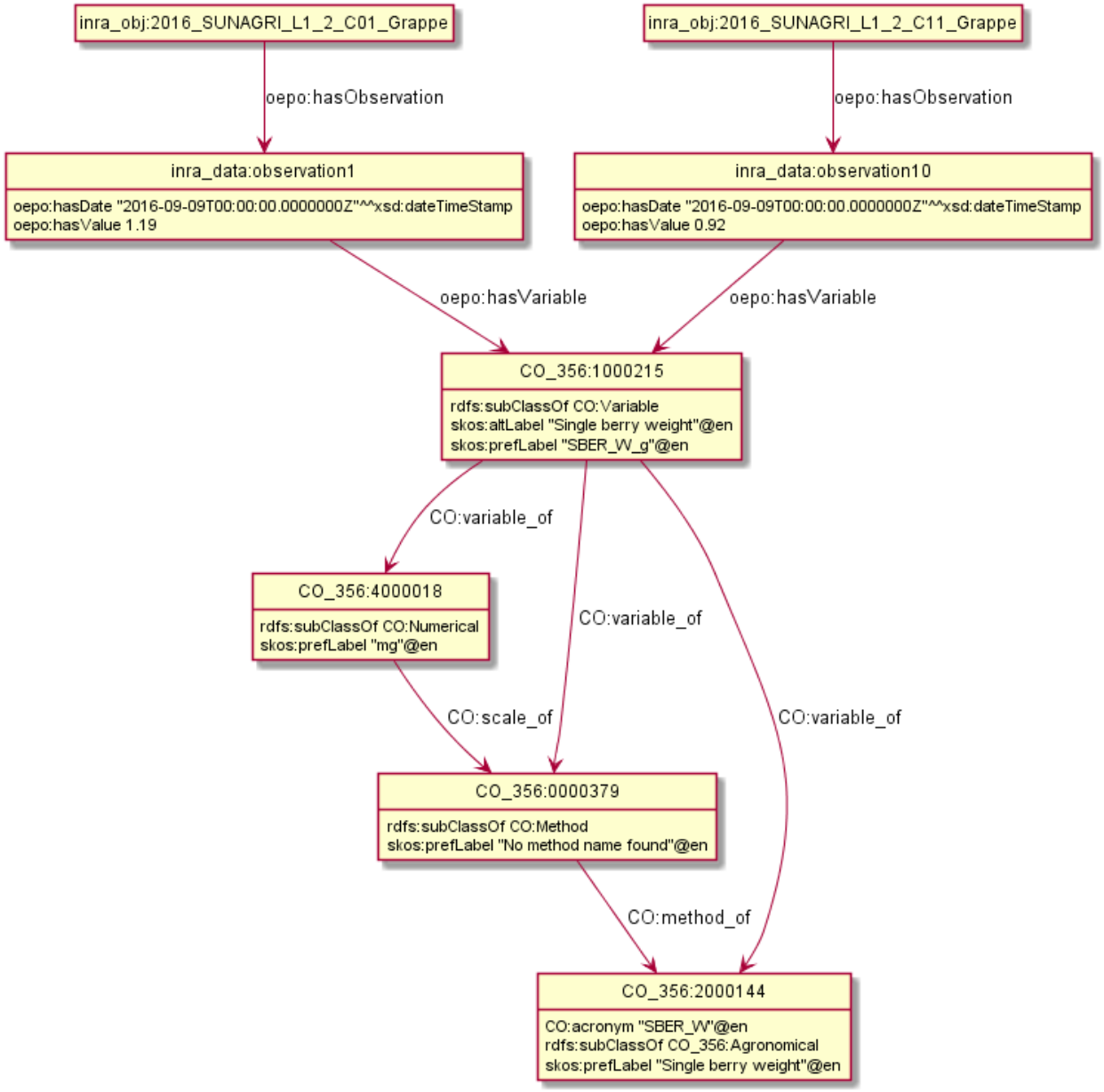


Figure 20 INRA Semantic Data

As we can see, the data consists of observations, in this illustration "single berry weight".

- Should define and use prefixes
- This is invalid datatype, should be xsd:dateTimeStamp. Alternatively, don't pad with a fake time of "0"

```
"2016-09-09T00:00:00.0000000Z"^^xsd:date
```

- http://www.cropontology.org/ontology/CO_356/Vitis#1000215 uses wrong URL, should be http://www.cropontology.org/rdf/CO_356:1000215
- http://vinnotec.supagro.inra.fr/public/Pr/data/observation1 etc are missing rdf:type
- The observed entities, e.g. http://vinnotec.supagro.inra.fr/public/Pr/2016_SUNAGRI_L1_2_C01_Grappe, are not defined in these files
- data5 includes a number of observation files, as follows:
- 2016vendanges_transf_parsed.ttl: Harvest observations: inra_onto:Poidsvendangegpesee (grams harvested).
- ComposantesGrappe_transf_parsed.ttl: Observations: inra_onto:Nbbaiescomptage number of counting bays?
- ComposantesVendanges_transf_parsed.ttl: Observations: inra_onto:Nbgrappescomptage number of counted clusters?
- fieldsLocalisationPR_parsed.ttl: plot geo-references (polygons), uses the GeoSPARQL ontology.
- FinFermentationsAlcoolique_transf_parsed.ttl
- INRA_variables.ttl: Variable definitions
- Maturite_transf_parsed.ttl
- MaturiteAnthocyanes_transf_parsed.ttl
- MaturiteJus_transf_parsed.ttl
- MaturiteSunAgri2B_transf_parsed.ttl
- must_transf_parsed.ttl: Observations: inra_onto:Sucrestotaux.brixrefractometrie Total sugars (BRIX refractometry)
- Suivifermentations_transf_parsed.ttl: Follow-up fermentations of ofpe:IntermediateProduct: observations of "Glucose/fructose g/l sequential enzymatic".

We have examined these files and made a number of recommendations, see google document README (or README.html). Often the same error applies to several terms in the same file, or to several files. E.g. the inapplicability of dct:created to time:Instant is reported for the first observation file 2016vendanges_transf_parsed.ttl but applies to all observation files.

- **Turtle prefix format.** The files use the SPARQL syntax for prefixes

```
PREFIX  inra_obj:  <http://vinnotec.supagro.inra.fr/public/Pr/>
```

  - While this is not an error (Turtle 1.1 supports this syntax), the older syntax supports wider interoperability:

```
@prefix inra_obj:  <http://vinnotec.supagro.inra.fr/public/Pr/> .
```

- **Check against prefixes.ttl.** Use exactly the same prefixes as defined in prefixes.ttl. Consult http://prefix.cc for the most popular prefixes to use, and add to prefixes.ttl as needed.
  - Use dct: not dcterms: for DC Terms: both are valid, but the former is more popular
  - Use geo: not gsp: for GeoSPARQL, the former is a lot more popular

- **Namespaces are not suggestive.** These namespaces do not suggest they hold time and observations respectively:

```
PREFIX context:   <http://www.phenome-fppn.fr/m3p/eventInsertion_ARCH2017-03-30>
PREFIX inra_data:  <http://vinnotec.supagro.inra.fr/public/Pr/data/>
```

- **URLs should be resolvable.** These files use the following INRA ontologies/resources. The URLs don't resolve, and return error "Veuillez vous connecter pour avoir accès à cette page". The project should publish the data in proper semantic format, and the URLs should become resolvable.

```
inra_obj:  <http://vinnotec.supagro.inra.fr/public/Pr/>
inra_data:  <http://vinnotec.supagro.inra.fr/public/Pr/data/>
inra_agent: <http://vinnotec.supagro.inra.fr/public/Pr/agent/>
inra_code:  <http://vinnotec.supagro.inra.fr/public/Pr/code/>
inra_onto:  <http://vinnotec.supagro.inra.fr/public/Pr/onto/>
```

- **syntax error (unquoted string)**

```
[line: 183, col: 24] Unrecognized: divers
inra_obj:JARDIN-AMPELO divers rouge rdf:type aeo:Plot ;
```

- **dct:created is inappropriate**: one can't "create" a time instant (it just exists), so dcterms:created is inappropraite. To express when an event was converted (vs occurred), we could use the PROV ontology.

```
context:instant_e1ba2667-2a37-4a42-b157-7ac07bfc458e rdf:type time:Instant ;
  time:inXSDDateTimeStamp "2016-08-24T12:00:00+01:00"^^xsd:dateTimeStamp ;
  dcterms:created "2018-07-12T18:52:00.012981"^^xsd:dateTime .
```

- **aeo:involvedIn is inappropriate**. Plots are **part of** Lots, they are not **involved in** lots. aeo:involvedIn is defined as "AgriExperiment involves different instances of AgriActivity and AgriEntity")

```
inra_obj:81-CHARDONNAY rdf:type aeo:Plot ;
  aeo:involvedIn inra_code:Lot_FV-2016-002 ;
```

- **Class vs Property**. This is a class not a property, so it can't be used like this. (In general, I notice that all AgroBio ontologies have lots of classes but few properties).

```
ofpe:Operator inra_agent:fabien.robert ;
```

- **rdf:value?** I can't verify whether oepo:Observation can take rdf:value because OEPO doesn't define this. Using rdf:value this way could be ok, but we should specify it with an RDF Shape.
- **invalid DateTimeStamp**, as reported by Jena RIOT.

```
[line: 16, col: 28] Lexical form '09/09/16' not valid for datatype xsd:DateTimeStamp
```

- **missing rdf:value**. Jena RIOT reports an error, which is caused by a missing rdf:value in the observation.

```
[line: 491, col: 47] Triples not terminated by DOT
inra_data:4e1956e2-eceb-477f-97a4-d22a919970b1 rdf:type oepo:Observation ;
  time:hasTime context:instant_39dec42b-9d84-4269-96f6-289d0d0ee782 ;
  oepo:hasVariable inra_onto:Nbbaiescomptage ;
```

- **Indicate grape variety**. Plots don't seem to indicate the grape variety, except in the URL, but a URL should be interpreted as opaque and not information-bearing.

```
inra_obj:22-SYRAH rdf:type aeo:Plot .
inra_obj:68-COLLECTION-BLANCS rdf:type aeo:Plot .
```

- **Use QUDT**. Plot areas are described using DBpedia and the Telegraphis Quantity ontology (which returns 404 Not Found). However, we better use the QUDT ontology that is more popular and has a full complement of SI and other kinds of units, including expression of units in terms of fundamental quantities (time, mass, length, etc) and conversion factors between units.

```
PREFIX dbo:     <http://dbpedia.org/ontology/>
PREFIX quty:    <http://www.telegraphis.net/ontology/measurement/quantity#>

inra_obj:81-CHARDONNAY rdf:type aeo:Plot ;
  oepo:hasObservation inra_data:6870097e-13b9-4179-83c3-78450c0bb8ce .
inra_obj:81-CHARDONNAY rdf:type aeo:Plot ;
  quty:area "1.20600"^^dbo:hectare .
```

- **Fix polygon geometry**. Plot polygons as defined include just 4 coordinates. Even for a simple box you need 4 corners, i.e. 8 coordinates. Coordinates should be +-180 degrees longitude and +-90 degrees latitude, but these are very big numbers. There are two pairs of the same number, but these should be "lat lon" pairs.

```
inra_obj:81-CHARDONNAY gsp:hasGeometry inra_gis:polygon_81-CHARDONNAY .
inra_obj:81-CHARDONNAY rdf:type aeo:Plot ;
gsp:asWKT "POLYGON ((710743.61182814 710743.61182814, 6226766.01933858 6226766.01933858 ))"^^gsp:wktLiteral .
```

- After coordinates are fixed, we need to check them for validity:
  - Order of latitude/longitude
  - That it indicates a place in France
  - That the given area in hectares corresponds to the polygon's area
- **gsp:Polygon vs gsp:Geometry**. There's no class gsp:Polygon. Use gsp:Geometry instead
- **Declare geo:Feature**. geo:hasGeometry has domain geo:Feature, so it should be declared, e.g. as

```
inra_obj:22-SYRAH rdf:type aeo:Plot, geo:Feature.
```

- **Namespace hijacking**. Don't define terms of other ontologies:

```
CO:variable_of  rdfs:subProperty_of  skos:related ;
     rdf:type            owl:ObjectProperty .
```

- **Use English class names**. To make ontologies that are more easily understood and reusable, we should use English

```
inra_onto:Poidsvendangegpesee # weight as measured at vine picking
```

- **Define labels**. E.g. inra_onto:Glucose.fructoseg.lsequentiel.enzymatique needs a label such as "Glucose/fructose g/l sequential enzymatic"
- **Can't use CO_UO "gram"**. Checking whether inra_onto:Poidsvendangegpesee defines everything required to interpret the number, we find the following data.

```
inra_onto:Poidsvendangegpesee CO:variable_of CO_356:2000168 , CO_UO:0000021 , MMO:0000157 .
CO_356:2000168 rdfs:label "Yield"@en .
CO_UO:0000021 rdfs:label "g"@en; CO:scale_of CO_357:2000105 .
CO_357:2000105 rdfs:label "Ratio shoot root protocol"@en .
MMO:0000157 rdfs:label "digital scale post excision weight measurement" .
```

  - CO_UO:0000021 "gram" is defined as a scale of "ratio shoot root" (some Woody Plant feature), so it cannot be used for grapes. This is yet another example of over-specialization (improper lack of abstraction) in AgroBio ontologies.
  - Note: one can get the whole CO_UO from neither http://www.cropontology.org/rdf/UO: nor http://www.cropontology.org/rdf/UO. But individual terms are returned, e.g. http://www.cropontology.org/rdf/UO:0000021 returns Turtle.
- **Missing CO_UO Term**. http://www.cropontology.org/rdf/UO:0000175 is missing: unlike the above UO:0000021, this one returns nothing.

```
inra_onto:Glucose.fructoseg.lsequentiel.enzymatique CO:variable_of
 CO_356:2000057, CO_UO:0000175, MMO:0000388 .
```

- **Reflexive subclass.** AEO defines a reflexive subclass relation (last pair in the chain below), which is implied by RDFS and is useless

```
aeo:Plot < aeo:CultivatedLand < aeo:Area < aeo:AgriEntity < aeo:AgriEntity
```

- **Syntax error.** The problem is missing a prefix of the subject.

```
[line: 28, col: 1 ] Broken token (newline): VIP_Sauvignon rdf:type afeo:Must ;
```

Syntax error.

```
[line: 144, col: 26] Unrecognized: sec
```

- **Class vs Property**. oepo:Observation needs some link to Agent, be that Operator or Organization. But foaf:Organization is a class not a property so it can't be used like this.

```
inra_data:32757c4a-15dd-4896-a3b9-970f33e6f756 rdf:type oepo:Observation ;
  foaf:Organization inra_code:16-1841 ;
```

- **Where are inra_codes defined?** These codes are used by the data, but are not defined anywhere.

```
inra_code:Cuve_BB1010 # FinFermentationsAlcoolique_transf_parsed
inra_code:BB1010      # Suivifermentations_transf_parsed
```

- **Organization individuals.** Organization URLs (e.g. inra_code:16-1841) use some codes. These URLs should be defined as proper individuals and may be better to use some more suggestive URLs.

3.3.2 AUA Tabular Data analysis

AUA has submitted tabular observation data (soil, plant canopies, spectral vegetation indexes) about table grapes.

- See the data in WP8/Table Grapes Pilot- AUA/Data. See Photos for some images.
- See D8.1 Piloting Plan (specifically BigDataGrapes_Piloting Plan-AUA) for descriptions of the equipment and measured indicators
- The measurements are made with 4 kinds of equipment: EM38, RapidScan, SpectroSense, Crop Circle:
  - Measurements for Soil Electrical Conductivity are taken with an **EM38** device
  - Measurements include information from plant canopies and classic spectral vegetation index data (NDVI, NDRE etc.) with **RapidScan, SpectroSense** and **Crop Circle**
- There about 10 measurements per measurement spot
- The measurements are Geo-referenced (longitude, latitude, altitude) and timestamped
- Includes 3 estates: Fasoulis, Kontogiannis, Palivou. Each estate is subdivided into a number of plots. The plots are named after:
  - Grape varieties: mavroudi, roditis, savatiano, soultanina (Kontogiannis Estate); Merlot (Palivou Estate)
  - Nearby settlements: solomos (Kontogiannis estate)
  - Names given by the owners or relative to the location: Geotrisi, IFG, Kato (Fasoulis Estate); Alekos, dipla oinopoiio, kato, mesi, pano (Palivou Estate)
- **Boundaries** and **Elevation** files give the plot spatial coordinates, e.g.: Fasoulis_RTKGPS_Boundaries.csv, Kontogiannis_RTKGPS_Boundaries.csv, Kontogianis_RTKGPS_Elevation.csv, Palivou_RTKGPS_Boundaries(all).csv, Palivou_RTKGPS_Elevation(all).csv

For example, file "5. Fasoulis_IFG_RapidScan.xlsx" includes tabular info like this (22 columns):

PLOT	NDRE	NDVI	RE	NIR	R	LATITUDE	LONGITUDE	ELEVATION	HDOP	FIXTYPE	DATE

37	0.2252	0.7376	20.836	33.084	5.132	37.81713	22.58971	291.5	2.8	GPS	5/23/2018

TIME	N	MAXNDRE	MAXNDVI	MINNDRE	MINNDVI	STDNDRE	STDNDVI	CVNDRE	CVNDVI
10:12:50	256	0.3423	0.8872	-0.3207	-0.0788	0.0784	0.1675	0.3479	0.2271

See AUA Table Grapes Data for some notes on measurement equipment and specific measurements

- EM38 measures apparent soil electrical conductivity (ECa):
  - Longitude
  - Latitude
  - CV1m: conductivity at depth 1 meter in millisiemens per metre (mS/m)
  - CV05m: conductivity at depth 0.5 meter in millisiemens per metre (mS/m)
  - Quality, Satellite, HDOP: related to the GPS signal-explained below
  - Elevation
  - Time and Date given by the GPS
- RapidScan measures Canopy characteristics and vegetation indices:
  - RE: Red-Edge spectral region (spectrum centred around 715 nm)
  - R: Red spectral region
  - NIR: Near-infrared spectral region
  - NDRE: mean value Normalized Difference Red Edge Index, defined using NIR and RE
  - NDVI: mean value Normalized Difference Vegetation Index, defined using NIR and R: (NIR-R)/(NIR+R)
  - Latitude
  - Longitude
  - Elevation
  - HDOP, FIXTYPE: related to the GPS signal-explained below
  - Date, Time
  - MAXNDRE, MAXNDVI: maximum values for NDRE and NDVI
  - MINNDRE, MINNDVI: minimum values for NDRE and NDVI
  - STNDVI, STNDRE: standard deviation for NDRE and NDVI
  - CVNDRE, CVNDVI: coefficient of variation for NDRE and NDVI
- Both equipment record a GPS and datetime fix:
  - Longitude, Latitude, (or Northing and Easting on a UTM projection ZONE 34N) Elevation
  - Time, Date
  - HDOP: horizontal dilution of precision, a factor in determining the relative accuracy of a horizontal GPS fix
  - Quality: quality of the GPS receiver (EM38 only)
  - Sat: which satellite provided the GPS fix (EM38 only)
  - PLOT: sequential measurement number in this run (RapidScan only). Note: this is **not** a plot number
- SpectroSense measures canopy characteristics and vegetation indices:
  - Context:
    - Northing, Easting: a specific way of expressing coordinates
    - Elevation
    - Satellite
    - HDOP
    - Date and Time
    - Mod: related to the GPS signal
  - Canopy characteristics:

- REDi: Incident radiation of the red spectrum
- REDr: Reflected radiation of the red spectrum
- NIRi: Incident radiation of the Near-InfraRed spectrum
- NIRr: Reflected radiation of the Near-InfraRed spectrum

- Then we calculate the following:
  - NIR: NIRr / NIRi
  - RED: REDr / REDi
  - NDVI: Normalized Difference Vegetation Index = (NIR - RED) / (NIR + RED)
  - LAI: Leaf Area Index = 0.0148*(EXP(6.192*NDVI))
- Optical Measurement Bands: (SF1-SF3 User definable and SF4, SF5 calculated by the sensor)
  - SF1 - channel with 670 nm (BW ±11 nm) interference filter
  - SF2 - 730 nm (BW ±10 nm) interference filter
  - SF3 - 760 nm (LWP) interference filter
  - SF4 and SF5

To tie measurements to a specific plot, geo-coordinates need localization within the plot (GeoSPARQL **within** predicate).

3.3.3 Natural Cosmetics Data

SYMBEEOSIS provide experimental data consisting of laboratory analysis of grape byproducts. The data is in csv format and consists of products from 16 estates tested over 10 variables.
The result form RDFization and linking is available on the BDG GitHub repository

An example line form the natural cosmetics data file

Sample	pH	Refractive Index	Total microbial count	Yeasts and moulds	Antioxidant activity DPPH (Mg/mL trolox)	Antioxidant activity ABTS (Mg/mL trolox)	Total phenolic content, TPC (Mg/mL gallic acid)	Total flavonoid content, TFC (Mg/mL quercetin)
I.A.1_M	5.38	20.47	<10	<10	25.26	12.35	41.62	45.67

And the resulting RDF

```
<http://data.bigdatagrapes.eu/resource/data/cosmetics/2018/IA1M>
    rdf:type            qb:Observation ;
    bdg:pH              5.38 ;
    qb:dataSet          <http://data.bigdatagrapes.eu/resource/data/cosmetics/2018> ;
    bdg:extractionMethod    <http://data.bigdatagrapes.eu/resource/extractionMethod/Masseration> ;
    bdg:refractiveIndex     20.47 ;
    bdg:antioxidantActivityDPPHTrolox  25.26 ;
    bdg:antioxidantActivityABTSTrolox  12.35 ;
    bdg:sample          "I.A.1_M" ;
    bdg:totalMicrobialCount  <http://data.bigdatagrapes.eu/resource/microbialCount/LT10> ;
```

```
bdg:yeastsAndMoulds     <http://data.bigdatagrapes.eu/resource/microbialCount/LT10> ;
bdg:TPCGallicAcid       41.62 ;
bdg:TFCQuercetin        45.67 .
```

The natural cosmetics pilot provided us also with a specific use case involving querying disproportionately voluminous data. We solved the used cas using distributed inference techniques, interfacing between a natoive RDF repository and an auxiliary document store. The Results for this particular use case are detailed in Section 8 of D4.2.

3.3.4 ABACO field sensor data

Abacco have provided data from field sensors. The data exists both in JSON and CSV format. A sample is visible in the github repository and the raw data is available form the Pessl FieldClimate datacloud http://fieldclimate.com/

The data model is available in the github reposotory and is used as part of the wind speed and direction demo (section 3.4.4.1 Wind speed and direction data transformation)

3.3.5 Geocledian parcel data analysis

Geocledian provide geotagged data concerning the plots and parcel of wineries. Their data is modelled and integrated in the BDG semantic model and can be queried using geosparql. An example of the use of GeoSPAQRL is shown in the Geo Aggregation uses case in section 3.4.5 Data Localization - a use case

3.3.6 AGRONOW Risk management data

The risk management pilot has provided data for food incident tracking . Besides the need for reconciliation of some geographical entities (such as "Czechia" → Check republic) The bulk of the data maps nicely to the BDG semantic data model and uses persistent URI wherever needed in the shared taxonomies it employs.

3.4 Data Processing Requirements

This section outlines some specific data processing requirements to be taken into account by WP3.

3.4.1Data Validation and Handling

Based on the syntactic and semantic errors observed above, we employ a guideline for data validation and handling. It covers:

Started rules on:

- How to submit files. We currently use Github, which is synchronized with Google Drive, but should select only one of them.
- How to use and update prefixes.ttl, a common prefixes file to be used consistently by all project partners.
- How to validate RDF file syntax using Jena **RIOT** (and maybe Jena **Eyeball**)

prefixes.ttl

The project keeps a single master prefixes file: prefixes.ttl (this is currently in ontology/model, but will definitely move to a more meaningful location).

- All partners should ensure they use the same namespaces and prefixes (e.g. dct: not dcterms: for Dublin Core Terms, and geo: not gsp: for GeoSPARQL)
- Check your prefixes against prefixes.ttl: if there's a discrepancy, discuss with Ontotext
- If you need a new prefix: consult http://prefix.cc for the most popular one, add it to prefixes.ttl and commit.

As a best practice, do not include individual prefixes in Turtle files, instead always prepend prefixes.ttl. This is especially important if you exchange a large number of small/example files.

Syntax Validation

- Use RIOT (part of Apache Jena) to validate the syntax of your files, e.g.

```
riot --validate 2016vendanges_transf_parsed.ttl
```

- If you prepend prefixes to Turtle files, use the script **riotval.pl:** it prepends prefixes, calls RIOT validation, then subtracts the number of lines in prefixes.ttl from error messages.
- For more extensive experimentation, also try Jena Eyeball that performs deeper validation (e.g. that unknown class/property names are not used). However, there is no Apache release of Eyeball and the code has not been updated for Jena3.

3.4.2 Data Cleaning

Use case A. Data Anomaly Detection & Classification defines some needs for data cleaning. E.g. see this row:

- Name: Eca sensing;
- Description: Georeferenced soil electrical conductivity data;
- Operations Performed: Data filtering for outliers:
- Provenance: Proximal sensors

EM38 is affected by metal pillars (poles), so soil conductivity readings near such poles make the measurement invalid. E.g. on Fasoulis_Kato_EM38_map (metal vineyard pillars).jpg, red readings show the position of pillars, and only the green readings should be retained. Readings over the value 100 should be discarded.

Another example is: **RapidScan** needs some time to establish a GPS connection. See file 6. Fasoulis_Geotrisi_RapidScan.xlsx for some examples. The following kinds of measurement should be discarded because they don't have a valid geo-reference:

- Readings with "FIXTYPE: Fix not valid" (missing geo-coordinates)
- Readings with negative ELEVATION (invalid geo-coordinates)

3.4.3 Alignment of aggregation across datasets - a use-case

This case compares two datasets containing similar meteorological information, but recording it in different manners. The Climate data from INRA's Pech Rouge weather station and data from a Pessl Weather station at Casato Prime Donne estate (from Abaco) contain information about the wind speed and direction. They differ in several aspects at the same time:

- *Granularity*: Abaco's data is a series of observations every 30 seconds, INRA's data is a daily summary.
- *Nature of the measurements*: Abaco measure the speed of the wind in metres per second and the direction. INRA report the total wind (in KM) for the day
- *Discretisation of the directions:* Abaco use the magnetic bearing (in degrees) to represent wind direction INRA use the 8 main compass directions (N,NE,E etc..)

The objective is to demonstrate how semantic technology allows conversion between the two representations in a 100% declarative manner.

Datasets

Here are subsets of both datasets illustrating the relevant entities:

```
curl
   "https://docs.google.com/spreadsheets/d/1e3KHXUCC6jwM7tTQURYPWi5OEkXRvyTH6J3orL8btlA/g
   viz/tq?tqx=out:csv" | csvcut -c "YEAR,MONTH,DAY,TNW,TNEW,TEW,TSEW,TSW,TSWW,TWW,TNWW" |
   head -n 10 | csvtomd
```

We can see in the table that on new year's day 2012, at Pech Rouge a total of 16km wind blew from the north (TNW). This can be the result of (for example) 4 hours of 4km/h north wind.

YEAR	MONTH	DAY	TNW	TNEW	TEW	TSEW	TSW	TSWW	TWW	TNWW
2012	1	1	16	5	0	0	1	31	96	54
2012	1	2	44	10	0	0	1	19	139	120
2012	1	3	14	1	1	1	2	21	65	95
2012	1	4	31	2	0	1	4	25	94	175
2012	1	5	13	1	0	1	4	53	261	213
2012	1	6	66	1	0	0	0	4	77	313
2012	1	7	24	1	1	0	1	12	110	251
2012	1	8	19	1	0	0	1	26	105	155
2012	1	9	13	1	0	0	0	12	149	257

This is the RDF resulting from the first line

```
<wineMaking/PechRouge/climaticData/11170004/2012-01-01>
        rdf:type                        qb:Observation ;
        qb:dataSet                       <data/wineMaking/PechRouge/climaticData/11170004>
   ;
        bdg:date                        "2012-01-01"^^xsd:date ;
        bdg:total_wind_E                "0"^^xsd:decimal ;
        bdg:total_wind_N                "16"^^xsd:decimal ;
        bdg:total_wind_NE               "5"^^xsd:decimal ;
        bdg:total_wind_NW               "54"^^xsd:decimal ;
```

```
        bdg:total_wind_W                "96"^^xsd:decimal ;
        bdg:total_wind_S                "1"^^xsd:decimal ;
        bdg:total_wind_SE               "0"^^xsd:decimal ;
        bdg:total_wind_SW               "31"^^xsd:decimal ;
```

Here is a sample from Abaco's data, showing that on 2019-05-23, between 10:00:24 and 10:30:24 the wind speed was an average of 0.9m/s and the average direction was 225°

Date	dir	speed
2019-05-23 10:00:24	208	0.7
2019-05-23 10:30:24	225	0.9
2019-05-23 11:00:23	202	1
2019-05-23 11:30:24	195	1.1
2019-05-23 12:00:24	220	1
2019-05-23 12:30:23	234	1
2019-05-23 13:00:24	211	1
2019-05-23 13:30:24	234	0.8
2019-05-23 14:00:23	126	1.1

And the resulting RDF from the first line:

```
<data/farmManagement/windDemo/2019-05-23T10:30:24>
    rdf:type                qb:Observation ;
    qb:dataSet              <data/farmManagement/WindDemo> ;
    bdg:dateTime            "2019-05-23T10:30:24"^^xsd:dateTime ;
    bdg:speed_wind_MEAN     "0.9"^^xsd:float ;
    bdg:direction_wind_MEAN "225"^^xsd:float .
```

The target model in this case is INRA's data because it is the least granular one. One line of INRA data corresponds to 48 lines of Pessl data.

Compass directions

The conversion between symbolic and numeric directions (225° to "South West") is done via a Concept list representing the 8 directions and their corresponding range of bearings.

```
<resource/compass/southwest>
    rdf:type       skos:Concept ;
    rdf:type       bdg:Compass ;
    skos:prefLabel  "Southwest" ;
    skos:inScheme   <compass> ;
    bdg:compassFrom  202.5 ;
    bdg:compassTo   247.5 .
```

This is crucial not only to convert between the two modes of representing direction but also because the concept gives us the means to select the relevant qb:MeasureProperty, that will be the predicate of the new value, in this case bdg:total_wind_SW, which is also linked to the same concept via the bdg:measurementContext predicate:

```
bdg:total_wind_SW  rdf:type       qb:MeasureProperty ;
    rdf:type               sosa:ObservableProperty ;
    rdfs:label             "Total Wind Sud-West Direction" ;
    sdmx-attribute:unitMeasure  unit:KM ;
    sosa:hasFeatureOfInterest   <feature/Wind> ;
    bdg:measurementContext     <compass/southwest> ;
    bdg:derivedFrom          bdg:total_wind ;
    qb:concept             sdmx-concept:obsValue ;
    skos:notation           "TSWW" .
```

Conversion query

The following query converts between the two representations. The inner query does most the work:

- Calculates the cumulative wind using simple arithmetics
- Groups by date and aggregates the results for each day
- Converts bearing to direction

The outer query selects the relevant qb:measureProperty

```
PREFIX bdg: <http://data.bigdatagrapes.eu/resource/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX qb: <http://purl.org/linked-data/cube#>
select * {
  ?var a qb:MeasureProperty ;
      bdg:derivedFrom bdg:total_wind ;
      bdg:measurementContext ?compass .
  {
     select ?date (sum(?wind_km) as ?total_wind_km) ?compass where {
        ?s bdg:direction_wind_MEAN ?dir ;
           bdg:speed_wind_MEAN ?speed ;
```

```
        bdg:dateTime ?dateTime  .

      ?compass a bdg:Compass ;
            bdg:compassFrom ?from ;
            bdg:compassTo ?to ;
      filter(?dir >= ?from && ?dir < ?to )

      bind(?speed * 1.8 as ?wind_km) #speed in m/sec, 30 min interval, 1800 sec
      bind(strdt(replace(str(?dateTime),"T.*$",""),xsd:date) as ?date)
    } group by ?date ?compass order by desc(?date)
  }
}
```

link

var	date	total_wind_km
bdg:total_wind_E	2019-07-09	104.76
bdg:total_wind_NE	2019-07-09	18.720001
bdg:total_wind_S	2019-07-09	0.17999999
bdg:total_wind_SE	2019-07-09	1.6199999
bdg:total_wind_SW	2019-07-09	3.2399998
bdg:total_wind_W	2019-07-09	2.52
bdg:total_wind_E	2019-07-08	81.0
bdg:total_wind_NE	2019-07-08	25.56
bdg:total_wind_SE	2019-07-08	20.34
bdg:total_wind_SW	2019-07-08	2.6999998
bdg:total_wind_W	2019-07-08	2.8799999
bdg:total_wind_E	2019-07-07	45.539997

bdg:total_wind_NE	2019-07-07	66.78001
bdg:total_wind_S	2019-07-07	4.8599997
bdg:total_wind_SE	2019-07-07	3.78
bdg:total_wind_SW	2019-07-07	2.1599998
bdg:total_wind_W	2019-07-07	0.17999999

3.4.5 Data Localization - a use case

To link metrics to a specific sub-plot, we may need to localize geo-coordinates within a sub-plot. Assuming that we have the sub-plot polygons, we can use the GeoSPARQL predicate **within**. Ontotext GraphDB supports a full complement of GeoSPARQL relations, using 3 different spatial relation algebras.

Mean CV1M per plot

This case study demonstrates the integration between observational and geographical data in the BDG Knowledge Graph. It shows how we can aggregate data, where the aggregation criterion is whether

Observational data consists of several thousand discrete measurements of a given variable (Electrical Conductivity). The measurements are performed by a sensor pulled behind a tractor and record a measurement at roughly 20cm intervals. The resulting table (visible at 3. Kontogiannis_EM38.xlsx) is a list of 8053 measurements which follow the path taken by the tractor. Each measurement is tagged with the precise coordinates within the field. However, given that the measurements are taken for the entire field, there is no straightforward way to differentiate, at a higher granularity which Plot they are part of.
In order to be able to do that, we define the Plots as polygons by their boundaries.
Then we can use GeoSPARQL to perform geographical queries at the same time as standard SPARQL queries. Inclusion of a point within a polygon becomes a fact that can be expressed as an RDF triple pattern. This allows us to use it for various operations such as grouping of point and statistically summarizing their values.

Example query summarizing the values of a variable by Plot and its results

```
base  <http://data.bigdatagrapes.eu/resource/>
PREFIX qb: <http://purl.org/linked-data/cube#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX bdg: <http://data.bigdatagrapes.eu/resource/ontology/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX ext: <http://rdf.useekm.com/ext#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select
```

```
?plot (count(*) as ?n_obs) (avg(?cv1m) as ?cv1m_mean) (avg(?cv05m) as ?cv05m_mean) (min(?dateTime) as ?min_time) (max(?dateTime) as ?max_time) (sample(?area) as ?AREA)
where {
        ?obs a qb:Observation ; qb:dataSet <data/tableGrape/Kontogiannis/EM38-mk2> ; bdg:position ?pos ; bdg:CV1m ?cv1m ; bdg:CV05m ?cv05m ; bdg:dateTime ?dateTime .
        ?plot a bdg:Plot ; geo:defaultGeometry ?plotGeo ; skos:broader <AUA/estate/Kontogiannis> .
  ?plotGeo geo:asWKT ?plotWKT .
  ?plotGeo geo:sfContains ?pos .
  bind(ext:area(?plotWKT) as ?area)
}
group by ?plot order by desc(?AREA)
```

link

plot	n_obs	cv1m_mean	cv05m_mean	min_time	max_time	AREA
estate/Kontogiannis/Mavroudi	2296	51.47	69.72	2018-03-26T16:13:53.31	2018-03-26T16:33:47.01	2.03E-7
estate/Kontogiannis/Soultanina	801	32.56	64.93	2018-03-26T16:11:46.81	2018-03-26T16:21:53.36	1.25E-7
estate/Kontogiannis/Savatiano	942	32.63	44.63	2018-03-26T16:17:09.83	2018-03-26T16:30:55.34	1.13E-7
estate/Kontogiannis/Roditis	609	32.12	53.66	2018-03-26T16:18:56.01	2018-03-26T16:28:45.83	5.39E-8

3.5 Data Access Requirements

There are some impediments to effective use of semantic technologies by AgroBio researchers that we need to address (these are in addition to semantic data integration steps/challenges as outlined in sec 3.1.4 Semantic Data Integration):

- Given the huge number of AgroBio ontologies, it is hard for researchers to find and effectively apply them.
- AgroBio researchers should not be expected (in most cases) to write SPARQL: they need a simpler way to get data out of the semantic Knowledge Graph, i.e. query writing aids and visualization mechanisms.

Regarding the first challenge, we are developing a harmonized data model adapted to the plethora of winemaking activities we have encountered. In it we combine existing ontologies to which we add new terms in an integrated and collaborative manner (see section3.3 Creating the BDG Semantic Model)

Regarding the second challenge, we provide an integrated software stack (see D.3.2) with the BDG Knowledge graph part of the back-end and a number of tools adapted to the needs of researchers deployed on top of it.

4. CONCLUSIONS

This document outlined the progress by WP3 Data & Semantics Layer in the *BigDataGrapes* project. More specifically, it has presented:

- The sort of data to represented in a semantic way
- Specific steps that we intend to follow for Semantic Data Integration
- Relevant AgroBio ontologies and problems that we have found in them
- Specific project data
- The BDG data model and the methodology used to collaboratively build it
- Specific data processing requirements
- Examples of challenges and how the model addresses them
- Specific data access requirements and relevant tools and approaches

The BDG semantic data model is the backbone of the harmonised datasets produced by the BDG project and we firmly believe that it will serve as a foundation for numerous precision and data driven agriculture tasks in the future, both in academia and in the industry.

5. REFERENCES

Cerans, K., Barzdins, J., Sostaks, A., Ovcinnikova, J., Lace, L., Grasmanis, M. and Sprogis, A. (2017). Extended UML Class Diagram Constructs for Visual SPARQL Queries in ViziQuer/web. Available online: http://ceur-ws.org/Vol-1947/paper08.pdf

Cerans, K. and Ovcinnikova, J. (2016). ViziQuer: Notation and Tool for Data Analysis SPARQL Queries. Available online: http://ceur-ws.org/Vol-1704/paper15.pdf

Ferré, S. (2015). Sparklis: An Expressive Query Builder for SPARQL Endpoints with Guidance in Natural Language. Semantic Web Journal. Available online: http://www.semantic-web-journal.net/content/sparklis-expressive-query-builder-sparql-endpoints-guidance-natural-language-0

Marginean, A., Groza, A., Slavescu, R.R., Alfred Letia., I. (2014). International Conference on Development and Application Systems, DOI: 10.1109/DAAS.2014.6842456. Available online: https://www.researchgate.net/publication/263218121_Romanian2SPARQL_A_Grammatical_Framework_approach_for_querying_Linked_Data_in_Romanian_language

Meroño-Peñuela, A., and Hoekstra, R. (2016). grlc Makes GitHub Taste Like Linked Data APIs. ESWC 2016 Satellite Events (SALAD 2016), Heraklion, Crete, Greece, May 29 – June 2, 2016, Revised Selected Papers. LNCS 9989, pp. 342-353 (2016). Available online: https://link.springer.com/chapter/10.1007/978-3-319-47602-5_48

Meroño-Peñuela, A. and Hoekstra, R. (2017). ISWC 2017, 16th International Semantic Web Conference. Lecture Notes in Computer Science, vol 10587, pp. 334-339 (2017). Available online: https://iswc2017.semanticweb.org/wp-content/uploads/papers/MainProceedings/430.pdf

Meroño-Peñuela, A. and Hoekstra, R. (2017a). SPARQL2Git: Transparent SPARQL and Linked Data API Curation via Git. Proceedings of the 14th Extended Semantic Web Conference (ESWC 2017), Poster and Demo Track. Portoroz, Slovenia, May 28th – June 1st, 2017. DOI 10.1007/978-3-319-70407-4_27. Available online: https://pdfs.semanticscholar.org/85cc/73ede853e8f7d9c1c7371c4b435a80123af3.pdf

Soru, T., Marx, E., Moussallem, D., Publio, G., Valdestilhas, A., Esteves, D. and Baron Neto., C. (
2017). Posters and Demos Track of the 13th International Conference on Semantic. Amsterdam, The Netherlands, September 11-14, 2017. Available online: http://ceur-ws.org/Vol-2044/paper14/

Soylu, A., Giese, M., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D. and Horrocks, I. (2017). Universal Access in the Information Society, June 2017, Volume 16, Issue 2, pp 435–467. DOI 10.1007/s10209-016-0465-0. Available online: https://link.springer.com/article/10.1007/s10209-016-0465-0