Integrating data and knowledge to support the selection of service plant species in agroecology

Elie Najm, Marie-Laure Mugnier, Christian Gary, Jean-François Baget,
Raphael Métral, Léo Garcia

Elie Najm, Marie-Laure Mugnier, Christian Gary, Jean-François Baget, Raphael Métral, et al.. Integrating data and knowledge to support the selection of service plant species in agroecology. Computers and Electronics in Agriculture, 2024, 217, pp.108594. 10.1016/j.compag.2023.108594 . lirmm-04393663

HAL Id: lirmm-04393663
https://hal-lirmm.ccsd.cnrs.fr/lirmm-04393663v1

Submitted on 14 Jan 2024

Integrating Data and Knowledge to Support the Selection of Service Plant Species in Agroecology

Elie Najm[a], Marie-Laure Mugnier[a,*], Christian Gary[b], Jean-François Baget[a], Raphael Métral[b], Léo Garcia[b]

[a]*LIRMM, Inria, University of Montpellier, CNRS, Montpellier, France*
[b]*SYSTEM, University of Montpellier, CIHEAM-IAMM, CIRAD, INRAE, Institut Agro, Montpellier, France*

Keywords: Functional trait, Ecosystem service, Service crop, Ontology-based data access, Rules, Reasoning

Abstract

There is a crucial need for tools to help researchers, technicians and farmers designing sustainable agroecosystems based on agroecology. Indeed, such agroecosystems are inherently complex and their design requires to integrate various data and unstabilized scientific knowledge. In this paper, we consider the issue of selecting service plant species according to their potential to provide ecosystem services. To tackle that issue, we adopt an approach based both on a formalized representation of domain knowledge, which enables reasoning, and on the exploitation of available data, collected independently of the targeted application. More specifically, we rely on the one hand on recent scientific results in agronomy linking functional traits (i.e., measurable characterics of plant species) to ecosystem services, and on the other hand on data about functional traits collected by the research community in ecology. The architecture of our system is inspired by the ontology-based data access paradigm, which allows to combine data and knowledge in a principled way. We provide a methodology to acquire scientific knowledge in the form of diagrams linked to data sources, as well as a formalization in a logical rule-based language. Importantly, our rules are independent from specific diagrams and data, to ensure genericity and facilitate the evolution of the system. We detail the construction of a knowledge base devoted to vine grassing, i.e., installing herbaceous service plants in vineyards, and present an evaluation of the system's results on this use case. We finally discuss the lessons learned and further challenges to be met.

1. Introduction

During the second half of the 20th century, the increased use of chemical inputs (e.g. pesticides, fertilizers) in agriculture allowed a rapid increase in the productivity of land and labour. However, this caused serious threats for the environment and human health such as soil, water and air pollution, loss of biodiversity, contribution to climate change, etc. It is now established that the restoration of sustainable farming systems requires the adoption of agroecological practices supporting the reintroduction of biodiversity in agroecosystems (Duru et al., 2015). These practices lead to more complex agroecosystems including a higher

---

[*] Corresponding author

Email addresses: `elie.najm@inria.fr` (Elie Najm), `mugnier@lirmm.fr` (Marie-Laure Mugnier), `christian.gary@inrae.fr` (Christian Gary), `jean-francois.baget@inria.fr` (Jean-François Baget), `raphael.metral@supagro.fr` (Raphael Métral), `leo.garcia@supagro.fr` (Léo Garcia)

number of plant species, whether cash crops (providing food, energy, fiber, or raw materials) and service crops (providing various ecosystem services) (Garcia et al., 2018).

There is a crucial need for tools that would assist users in the design of such complex agroecosystems, from researchers in agronomy to agricultural advisors and farmers. A major difficulty to develop such tools is that agroecological engineering requires to integrate knowledge from various domains : agronomy, which provides knowledge on functions, design and management of agroecosystems, and ecology, which studies the relationships between living organisms and their natural environment, as well as other expertise domains like management sciences, and agricultural know-how; moreover, such knowledge is heterogeneous, scattered and not stabilised.

In this paper, we more specifically consider the issue of selecting service plant species according to targeted ecosystem services. To tackle that issue, we adopt an approach based both on a formalized representation of domain knowledge, which enables reasoning, and on the exploitation of data, which has been collected independently of the targeted application.

On the one hand, recent scholarly work identified relationships between plants *functional traits* and the ecosystem *functions* they provide, which support ecosystem *services* (Garcia et al., 2019, 2020; Damour et al., 2015). Functional traits are measurable characteristics of individual organisms (namely, morphological, physiological, or phenological) that are relevant to study the response of these organisms to environmental disturbances and/or their effects on ecosystem properties. Ecosystem functions are often rendered by combinations of traits, which can each contribute positively or negatively to the function. For example, several functional traits of the root system of a plant can be positively associated with the function of soil structural stability, which supports the service of maintenance of soil quality (Garcia et al., 2019).

On the other hand, the ecology community has compiled functional trait values collected from a large number of plants under a wide range of environmental conditions into the TRY database (`www.try-db.org`, (Kattge and al., 2020)). TRY currently integrates experimental observations coming from more than 400 datasets and concerning 4 millions individual plants, 2100 traits and about 280k plant taxa (mostly species). This constitutes a very rich source of information that has not yet been exploited to select service species in agriculture. Hence, we made the hypothesis that, as soon as we were able to associate the expected services and functions with functional trait values, this database, and possibly others, would make it possible to identify species that are favourable to these functions and services.

More generally, our leading question was the following: can we exploit currently available data on plant functional traits and combine it with a suitable representation of scientific knowledge on the trait-function-service relationships, to assess the potential contribution of any plant species to targeted ecosystem services?

While there are numerous decision support tools in agriculture, still few of them integrate the notion of ecosystem service, and to the best of our knowledge, none of them exploits preexisting data sources on functional traits, like TRY in ecology. In particular, available tools for selecting service species rely on local field experiments, farmers' know-how and workshops between experts in agronomy. As collecting the relevant information is time and budget demanding, these tools are typically restricted to a small set of service species. To illustrate, let us cite a tool in agroforestry to select shade tree species in coffee and cocoa agrosystems (Van der Wolf et al., 2019), based on an inventory of local practices; a tool devoted to banana and yam crops with a list of a hundred service plant species and five targeted services (Ozier-Lafontaine et al., 2010); and a tool devoted to rapeseed crops with eleven service plant species and three targeted services (CAPS, 2017). Interestingly, the two latter tools consider functional traits among other criteria, however they do not do it in a principled manner nor exploit preexisting datasets. Moreover, most existing tools are designed for farmers (or agricultural consultants) and give a recommendation adapted to a particular cultivation context, which has to be precisely known. Their objective is to deliver an accurate recommendation, whose computation remains a black-box. In contrast, our operational goal is to support the design activity of researchers and technicians in agroecology, and of farmers, with a tool that (1) widens the space of possibilities beyond the solutions already known to professionals, by suggesting new associations of species to be tried, and (2) provides the reasons behind the solutions suggested by the tool. In other words, innovation and explanability prevail over contextualized accuracy.

To integrate data and knowledge in a principled manner, we built an architecture inspired by the paradigm of *ontology-based data access* (OBDA) (Poggi et al., 2008; Lenzerini, 2018), which makes a fundamental distinction between the data and the conceptual levels. An OBDA system is composed of a data layer, made of one or several data sources that may have been built for independent purposes, and a conceptual layer, which describes knowledge (typically under the form of an ontology) using a vocabulary adapted to the intended application and users; declarative mappings between both levels allow to select and aggregate relevant data and to translate it at the conceptual level. The conceptual layer is formalized in a knowledge representation (KR) language provided with reasoning capabilities. Queries to the OBDA system are expressed at the conceptual level and answers to queries take into account inferences made by the system. This paradigm has several advantages. First, it allows a user to formulate queries using a familiar vocabulary, without knowing how the data is actually encoded and stored. Second, it provides richer answers, since not only facts directly coming from the data are considered, but also those that are inferred from both the data and the ontology. Finally, the conceptual vocabulary can act as a mediating layer to integrate several data sources.

The OBDA paradigm has attracted a lot of interest from the 2010s. Several implementations are now available, from mature systems (Calvanese et al., 2011, 2017) to research prototypes, e.g., (Buron et al., 2020), all based on semantic web languages, namely the lightweight OWL 2 QL profile, or RDF Schema and slight extensions of it. While OBDA has been deployed in various companies and public organizations, it has seemingly not been applied in agriculture / agroecology yet.

The chosen KR language is based on rules in first-order logic, a formalism well suited to the representation of expert knowledge. While being syntactically simple, these rules naturally allow for the representation of complex relationships between entities. They furthermore include computed functions (in the sense of programming), which are in particular used to calculate aggregations of trait values. Since existing OBDA systems do not offer the desired expressivity, we could not use them. Instead, we developed our own prototype based on the open-source software Graal (`https://gitlab.inria.fr/rules/integraal`) dedicated to query answering with rule knowledge bases (Baget et al., 2015).

To study the feasibility of our approach, we focused on the issue of installing service herbaceous species in vineyards (aka *vine grassing*), see Figure 1. We specifically detail in this paper the service of nitrogen supply to the vine. Note that, beyond this specific case study, our approach is in principle applicable to any use case targeting the selection of plant species based on the relationships between functional traits and ecosystem functions and services.

The paper is organized as follows:

- Section 2 gives an overview of the system. It outlines its architecture, which allows us to combine data and knowledge in a principled way. It also introduces the KR language and the query answering technique.
- Section 3 details the methodology for acquiring domain knowledge following the trait-function-service

Figure 1: Examples of service crops sown in inter-rows in vineyards. From left to right, *Hordeum vulgare*, *Phacelia tanacetifolia*, and *Vicia villosa*. Pictures by Alexis Thoumazeau, Léo Garcia and Yvan Bouisson.

approach and linking it to relevant data sources. This phase involves the construction of diagrams by agronomists, based on the scientific and grey literature, which are then translated into logic, yielding so-called "expert facts".

- Section 4 presents a pipeline composed of several types of mappings allowing to select data and transform it into knowledge expressed in logic, yielding so-called "data facts".
- Section 5 focuses on reasoning. It presents the different kinds of rules that exploit expert and data facts to infer new facts about the contribution of species to ecosystem functions and services. We also discuss the qualities of our approach in terms of genericity, evolutivity and explanability.
- Section 6 reports an evaluation of the quality of the obtained results, which was carried out for a specific ecosystem service and a selected set of species.
- Section 7 is devoted to the lessons learned and further challenges to be met.

Complementary material about the modeling and the evaluation can be found at `https://gitlab.inria.fr/boreal-artifacts/agroecology/`

2. Overview of the System

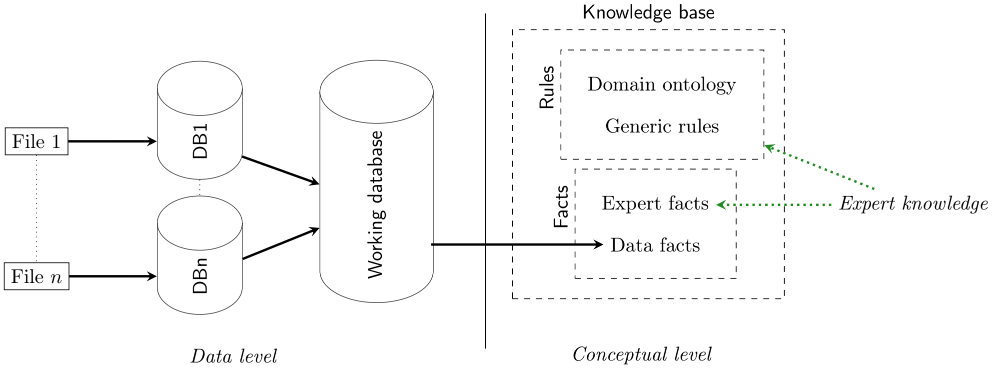


Figure 2: Overview of the global architecture. Cylinders depict databases, black arrows mappings and green dotted arrows the formalization of expert knowledge.

.

Our goal was to provide a tool that fulfills the following requirements:

1. integrating data, which may come from heterogeneous and independent sources,
2. combining data with a formal representation of expert knowledge that enables reasoning,
3. answering user queries while taking into account reasoning,
4. allowing an end-user to formulate queries in a familiar vocabulary,
5. being generic enough to be applicable to other use cases following the trait-function-service approach,
6. allowing for an easy evolution in terms of data sources or expert knowledge,
7. allowing to explain answers to queries.

The architecture of our system, presented next, meets requirements (1) to (4). Requirements (5) to (7) will be discussed in Section 5.7).

2.1. Architecture of the system

As explained in the introduction, the system's architecture is inspired by the *ontology-based data access* (OBDA) paradigm. Its main components are pictured in Figure 2. A global working database is obtained by integrating several data sources, which may themselves have required some preliminary cleaning steps. The data integration step selects and aggregates relevant data from the data sources, and translates it according to the global database schema, while keeping track of the data source provenance. The conceptual layer is made of a *knowledge base* (KB), expressed in a rule-based knowledge representation and reasoning language. The KB comprises two kinds of knowledge: *facts*, which are assertions about specific entities (together forming the fact base) ; and *rules*, which express general knowledge that can be applied to facts to infer/create new facts. We further distinguish between two kinds of facts: *data facts* obtained from the working database via mappings (e.g., the fact that some functional trait for a given species has some value according to a certain data source) and *expert facts* obtained from expert knowledge (e.g., the fact that a given ecosystem service relies on some ecosystem functions, which themselves are supported by some functional traits).

The *domain ontology* provides a vocabulary in terms of concepts and relations that are meaningful to a user (an expert in agroecology or an end-user). The semantic relationships between the elements of this vocabulary are specified by simple rules expressing concept/relation inclusions as well as signatures of relations. More complex relations and rules are used to process data facts and combine them with expert facts to evaluate the contribution of species to ecosystem functions and services. The latter rules are called *generic* in the figure, to emphasize that they are independent from specific functional traits, nor ecosystem function and services.

Mappings (black arrows in Figure 2) allow to select and aggregate information from a structure (a formatted text file, like a .csv file, or a database) and to translate the resulting information into the vocabulary of another structure (a database or a fact base). Mappings are specified in a declarative way. Different types of mappings are used in the system: *cleaning* mappings from a formatted text file to a database ; *database* mappings from a database DB_i to the working database ; and *data-to-knowledge* mappings from the working database to the fact base.

Queries from an end-user are formulated using the domain ontology vocabulary. This allows queries to be formulated at a high level of abstraction, regardless of how the data is actually stored (e.g., "select the k best species for the improvement of the soil structure, together with their score and the reliability of this score"). The answers to such queries are inferred from the KB, i.e., using facts and rules.

```
% 3 data facts
hasTraitValue("specific root length","dactylis glomerata",0.72).
hasTraitValue("root length density","dactylis glomerata", 0.38).
hasTraitValue("relative growth rate","dactylis glomerata", 0.54).
% 2 expert facts
ecoSystemFunction("soil exploration and competition with vines").
isLinkedTo("soil exploration and competition with vines",
"specific root length","root length density","relative growth rate",fct:mean).

% Rule 1:  all traits filled, 100% reliability
isLinkedTo(EcoSystFunction,Trait1,Trait2,Trait3,Aggregation),
hasTraitValue(Trait1,Species,V1), hasTraitValue(Trait2,Species,V2),
hasTraitValue(Trait3,Species,V3)
→ hasFctValue(EcoSystFunction,Species,fct:aggreg3(Aggregation,V1,V2,V3), 100).

% Rule 2:  first trait not filled, 66 % reliability
isLinkedTo(EcoSystFunction,Trait1,Trait2,Trait3,Aggregation),
not hasTraitValue(Trait1,Species,V1), hasTraitValue(Trait2,Species,V2),
hasTraitValue(Trait3,Species,V3)
→ hasFctValue(EcoSystFunction,Species,fct:aggreg2(Aggregation,V2,V3), 66).
```

Figure 3: Five facts and two rules. Lines starting with % are comments.

2.2. The KR language

Most KR languages are based on (first-order) logic, even if they often have a dedicated syntax. This is the case of description logics (Baader et al., 2017), specially designed for the representation of ontologies, W3C semantic web languages like RDFS and OWL 2, and rule-based languages like Datalog (Abiteboul et al., 1994), existential rules (e.g., (Mugnier and Thomazo, 2014)) or Answer Set Programming (Lifschitz, 2019). We did not make any *a priori* choice of a KR language, but rather started by eliciting expert knowledge to identify an appropriate language. By appropriate language, we mean a language allowing to express domain knowledge in a natural way, while having a minimal expressivity in order to avoid needlessly costly inferences. Rule-based formalisms were found to be well-suited, as expert knowledge is often expressed under the form "if condition then conclusion" and complex relationships between entities can be represented. Another important feature was the ability to incorporate computed functions (in a programming sense, i.e., a function outputs a value given a list of parameters) into the logical formalism.

We do not give a full definition of our KR language in this paper but rather illustrate it on the example of Figure 3, composed of five facts and two rules. In short, we consider plain Datalog rules, extended with computed functions and stratified default negation (Abiteboul et al., 1994), which can also be seen as a specific case of Answer Set Programming.

We recall that an *atom* in first-order logic is of the form *predicate(list of arguments)*, where each argument is either a variable, a constant (a specific object or a value), or a function, i.e., a functional symbol with a list of arguments. A *fact* is an instantiated atom of the form *predicate(list of constants)*. In Figure 3, the first three atoms are data facts, which specify values of the traits "specific root length", "root length density" and "relative growth rate" for the species *dactylis glomerata*. Note that values have been normalized here and range on the interval $[0, 1]$. The next two atoms are expert facts and specify that "soil exploration and competition with vines" is an ecosystem function, which is linked to traits "specific root length", "root length density" and "relative growth rate", with the method of aggregation of these trait values being the mean, denoted here by “fct:mean”.

A *rule* has the form $B \rightarrow H$, where B, the body of the rule, is a conjunction of (possibly negated) atoms and H, the head of the rule, is a single atom, whose variables already occur in the body. Intuitively, the rule body represents conditions to be met and the rule head the inferred conclusion; as specified next, applying a rule consists in instantiating its body by facts, thereby substituting its variables by constants, to produce a new fact obtained from the rule head. In our examples, strings starting with a capital letter denote variables, `not` denotes the negation and commas stand for conjunctions; furthermore lines beginning with % are comments. The rules from Figure 3 both consider an ecosystem function EcoSystFunction linked to traits Trait1, Trait2 and Trait3 with Aggregation as the aggregation method of these trait values. The first rule deals with the case where all traits have a value: if Trait1, Trait2 and Trait3 have values V1, V2 and V3, respectively, for a species Species, then the score of Species for EcoSystFunction is the aggregation of V1, V2 and V3 with method Aggregation, and the reliability of this score is 100%; the second rule deals with the case where Trait1 has no value for a species Species, while Trait2 and Trait3 respectively have values V2 and V3: then the score of Species for EcoSystFunction is the aggregation of V2 and V3 with method Aggregation, and the reliability of this score is 66%. Note that, in the 5th fact, fct:mean is a logical constant, denoting the mean function, while in the rules fct:aggreg3 and fct:aggreg2 are functional symbols whose first parameter is a variable acting as a placeholder for an aggregation method. Note also that this is a simplified example intended to illustrate the KR language; the actual atoms have additional arguments to specify the data source and the plant growing conditions for trait values, as well as weights expressing the strength of the participation of a trait to an ecosystem function.

Given a rule $B \rightarrow H$, its positive body B^+ (respectively, negative body B^-) is the subset of B that occurs positively (respectively, negatively) in B. We recall the classical notion of a homomorphism: a *homomorphism* h from a set of atoms B to a set of facts F is a substitution of the variables in B to the constants in F such $h(B) \subseteq F$, where $h(B)$ denotes the set of atoms obtained from B by substituting each variable according to h. A rule $B \rightarrow H$ is *applicable* to a set of facts F if (1) there is a homomorphism h from B^+ to F, and (2) there is no homomorphism extending h that maps an atom $a \in B^-$ to F. The application of the rule according to h yields the fact $h(H)$, obtained by substituting each variable X in H by $h(X)$, then evaluating computed functions. Note that negation is interpreted as default negation, i.e., a fact `not` a is entailed by the KB if a is not entailed by the KB.

In Figure 3, the first rule can be applied on the facts by the following homomorphism:

```
EcoSystemFunction ↦ "soil exploration and competition with vines",
Trait1 ↦ "specific root length",
Trait2 ↦ "root length density",
Trait3 ↦ "relative growth rate",
Aggregation ↦ fct:mean,
Species ↦ "dactylis glomerata",
V1 ↦ 0.72,
V2 ↦ 0.38,
V3 ↦ 0.54.
```

This rule application leads to evaluate the function fct:aggreg3(fct:mean,0.72,0.38,0.54), i.e., to execute the associated program code, and produces the new fact `hasFctValue("soil exploration and competition with vines", "dactylis glomerata", 0.55, 100)`. The second rule cannot be applied on the facts because the only homomorphism from its positive body to the facts can be extended to also map the atom `hasTraitValue(Trait1,Species,V1)` that belongs to its negative body.

The *saturated* fact base is obtained from the initial fact base by applying rules until a fixpoint is reached. A condition on the set of rules (known as stratifiability) ensures that a unique saturated fact base is obtained (Abiteboul et al., 1994).

2.3. Query Answering

Query answering is performed on the saturated fact base. E.g., on the example from Figure 3, a query asking for all species with a value greater than 0.5 for the ecosystem function "soil exploration and competition with vines" will admit *dactylis glomerata* as an answer.

It should be mentioned that query answering in OBDA usually follows a mediating (aka virtualization) approach, i.e., the fact base remains virtual, and user queries are first reformulated with the ontology, then rewritten with the mappings, to yield queries that are directly evaluated on the data. In contrast, we follow here a materialization approach: the fact base is first materialized (by triggering the mappings to materialize data facts and adding expert facts), then saturated by rule applications. Moreover, we finally store the part of the saturated fact base that is relevant to the end-user as a relational database, in order to benefit from the whole expressive power of SQL. There are several reasons for the choice of materialization: first, mediation has been mainly developed for simple queries (essentially unions of conjunctive queries), while our user queries are more complex (e.g., may involve aggregations); second, some features of our KR language (computed functions, default negation) do not allow to use off-the-shelf query reformulation techniques; third, most queries of interest require to rank species (e.g., find the k-best species for some service) and materialization is more appropriate to answer such queries efficiently. Finally, the main advantage of virtualization is the independence with respect to data updates, yet this is not an issue in our use cases, where it is not necessary to take into account data evolution in real time.

3. From expert knowledge to formalized facts

Expert knowledge is formalized into two kinds of constructs, namely facts and rules. On the one hand, *(expert) facts* are associated with specific ecosystem services; these facts describe the relationships between a specific service, relevant ecosystem functions and functional traits, as well as links to databases. On the other hand, rules describe in a generic manner how to compute the service value for a species, given relevant trait values; these rules are detailed in Section 5.

In this section, we present our methodology to get expert facts. In a nutshell, the main steps are the following:

1. The agronomists build diagrams that describe the relationships between specific functional traits, ecosystem functions and services, based on the scientific and the grey literature (Section 3.1).
2. The agronomists identify relevant databases and link the traits from the diagrams to these databases with the help of computer scientists (Section 3.2).
3. Diagrams and their links to databases are automatically translated into logic, which yields so-called expert facts (Section 3.3).

3.1. Expert diagrams

An expert diagram is devoted to a specific ecosystem service and describes which functional traits contribute (positively or negatively) to some ecosystem functions, which themselves contribute (positively or negatively) to the targeted service. Hence, it is structured in three levels respectively composed of traits, functions and the service, with (positive or negative) edges that connect elements from one level to elements from the next level. More specifically, a positive (resp. negative) edge from a trait to a function denotes a positive (resp. negative) statistical or causal relation between the trait value and the function value (which reflects how well the function is rendered). And similarly from functions to services. Moreover, the diagram specifies which aggregation methods have to be used to aggregate trait or function values. This is illustrated in Figure 4, which depicts the diagram for the service of nitrogen supply to the vine. Positive edges are colored in green and negative edges in red. Diagrams for two other services (soil structuration, water storage and supply to the vine) are provided in the complementary material.

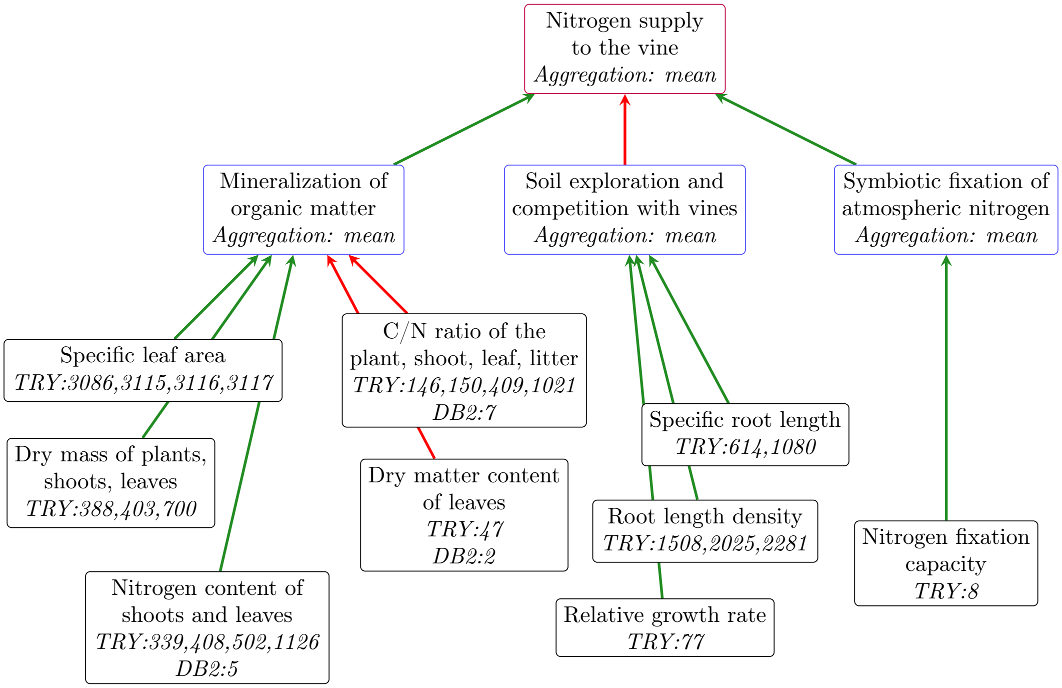


Figure 4: Trait-Function-Service diagram for the service "Nitrogen supply to the vine". Traits, functions and service are surrounded in black, blue and purple, resp. Green and red arrows indicate positive and negative impact, resp. Numbers are identifiers of traits in databases TRY and DB2.

To define services relevant to agroecosystems, the agronomists relied on the reference study EFESE (a French national initiative to assess ecosystems and ecosystem services: `https://www.inrae.fr/en/news/assessing-services-provided-agricultural-ecosystems-improve-their-management`). Then, for each service of interest they defined the underpinning functions and, for each function, relevant functional traits. The identification of these elements and their relationships was based on scientific papers as well as grey literature.

To illustrate, let us comment on the construction of the diagram in Figure 4. One of the main ecosystem services expected by farmers from service plants is the regulation of nutrient availability to crops, in particular nitrogen availability. Nitrogen supply to the vine can be provided by cover crops through two functions:

the mineralization of their biomass (organic matter) when buried in the soil, and the symbiotic fixation of atmospheric nitrogen (Cherr et al., 2006). A third function plays an additional role in nitrogen availability, namely the exploration of soil by the root system of cover crops, which determines the level of competition for soil resources with the grapevine. Hence, the three functions pictured in Figure 4, with positive and negative links to the service. The level of activation of these functions correlates to the values of some traits of plant species. High nitrogen release by mineralization is associated with high values of specific leaf area, dry mass and nitrogen content of plants, and low values of dry matter content of leaves and carbon to nitrogen ratio (Abalos et al., 2019; Damour et al., 2015; Hanisch et al., 2020). This leads to five relevant traits, positively or negatively connected to the function called "Mineralization of organic matter" in the Figure. An efficient exploration of the soil by roots and therefore a high competition with the grapevine is associated with high values of specific root length and root length density (Damour et al., 2015; Funk and Wolf, 2016; Freschet and Roumet, 2017). The relative growth rate correlates with the species' rate of establishment, especially during its initial growth phase (Poorter and Garnier, 2007). We posit that early species establishment is associated with rapid resource depletion and heightened competition. At last, the symbiotic fixation of nitrogen is associated with a functional trait that is specific to legumes, namely the ability to fix atmospheric nitrogen (Abalos et al., 2019; Damour et al., 2015).

Next, the agronomists were asked to specify *methods of aggregation* to pass from traits to functions (resp. from functions to services). Since no precise criteria could be derived from state-of-the-art domain knowledge, they decided to consider the mean of the normalized trait values (see Section 4.4 about normalization), thereby giving the same importance to all traits and functions (as specified in Figure 4). Note however that our formalization allows one to weight traits' and functions' contribution and to specify any method of aggregation.

3.2. Matching diagram traits to database trait IDs

Then, each functional trait in a diagram has to be associated with one or several trait IDs in databases.

As illustrated in Figure 4, our prototype currently uses two databases, namely TRY, the large plant trait database presented in the introduction, and DB2, a French database (see Section 4 for details). Some traits of the diagram are associated with a single database trait ID, like "Relative growth rate" associated with ID 77 in TRY, or with a single ID in each database, like "Dry matter content of leaves" associated with ID 47 in TRY and ID 2 in DB2. However, most traits have more than one match in TRY (note that this does not happen with DB2). The reason for this is that traits can be measured according to different techniques, which are reported in TRY. Hence, the same trait for the same observed individual plant may have different values depending on the measurement techniques. For instance, the trait "Specific leaf area" is linked to four traits in TRY, which are all deemed relevant:

- ID 3086: Leaf area per leaf dry mass: petiole, rhachis and midrib excluded,
- ID 3115: Leaf area per leaf dry mass: petiole excluded,
- ID 3116: Leaf area per leaf dry mass: petiole included,
- ID 3117: Leaf area per leaf dry mass: undefined if petiole is included or excluded.

Trait IDs associated with the same expert trait are called *exchangeable*. Since there is no universally preferred measurement technique and since we normalize trait ID values, it is pertinent to aggregate the values of exchangeable trait IDs for a species, regardless of the measurement technique. This aggregation is useful to mitigate the impact of missing values in databases. To perform this aggregation, the agronomists chose to uniformly consider the maximum value among the normalized values of exchangeable IDs in a given database (and for the same growing condition of plants, see Section 4.1).

Finally, our modeling includes *preferences* between databases: there is a global default total order, which can be overwritten for specific traits (here, TRY is globally preferred to DB2). This allows to give a higher priority to a more reliable source. Then, the value of a trait for a species will be given by the highest priority data source that provides such value (see Section 5.3 for details).

To sum up, the acquisition of expert knowledge about specific ecosystem services follows three steps:

1. Build trait-function-service diagrams, based on the literature.
2. Identify relevant databases and associate diagram traits with relevant trait IDs in these databases, with the choice of an aggregation technique if there are exchangeable IDs.
3. Define priority among databases, globally and possibly for specific traits.

Back and forth between steps (1) and (2) is necessary, as traits in the diagrams have to find counterparts in the databases.

3.3. From diagrams to logical facts

It remains to translate experts diagrams into logical facts. For that, we rely on the vocabulary provided by a domain ontology. This vocabulary consists of general concepts and relationships that are independent of any particular case study. The main concepts and relations are depicted in Figure 5. Among the main concepts, `tfsObject` (where `tfs` stands for Trait-Function-Service) has subconcepts `trait`, `function` and `service`, and `source` (which refers to sources of information) has subconcepts `dataSource` (data sources) and `userSource` (users). With regard to relations, note that only the relations listed in Points 1 and 2 are used to represent expert diagrams; relations listed in Point 3 are used in end-user queries (see Section 5.6). From a logical viewpoint, concepts and relations are seen as predicates: unary predicates for concepts and predicates of the same arity for relations. Then, the translation of a diagram into a set of logical facts is straightforward, as illustrated by Figure 6, which lists some of the facts obtained from the example diagram (Figure 4).

We actually used an intermediate KR language to go from experts diagrams to logical facts, namely conceptual graphs (Sowa, 1984; Chein and Mugnier, 2009). Indeed, a diagram can be directly encoded as a conceptual graph, using the concepts and relations of the domain ontology, and then a conceptual graph has a well-defined and automated translation into logic. The interest of this intermediate step that conceptual graphs are easily understood by agronomists, who can then check how diagrams are precisely encoded. See the appendix for details.

4. From Data to Facts

In this section we detail how facts are built from data using different kinds of mappings. Generally speaking, a mapping from a structure S_1 to a structure S_2 is a pair of queries: the first query on S_1 returns tuples of values selected from S_1, and the second query on S_2 inserts these tuples into S_2. As outlined in Section 2, we distinguish between three types of mappings: cleaning mappings from a formatted text file to a

Main concepts

```
tfsObject
    trait      % functional trait
    function   % ecosystem function
    service    % ecosystem service
source         % information source
    dataSource % data source
    userSource % user
species
growingCondition
aggregation    % aggregation method
```

Main relations

1. **Relations common to expert diagrams and end-user queries**

```
isLinkedToFunction(Trait,Function)
    isPositivelyLinkedToFunction(Trait,Function)
    % equivalent to:  isWeightedLinkedToFunction(Trait,Function,1)
    isNegativelyLinkedToFunction(Trait,Function)
    % equivalent to:  isWeightedLinkedToFunction(Trait,Function,-1)
isWeightedLinkedToFunction(Trait,Function,Weight)
isLinkedToService(Function,Service)
    isPositivelyLinkedToService(Function,Service)
    % equivalent to:  isWeightedLinkedToService(Function,Service,1)
    isNegativelyLinkedToService(Function,Service)
    % equivalent to:  isWeightedLinkedToService(Function,Service,-1)
isWeightedLinkedToService(Function,Service,Weight)
hasAggregationMethod(TFSObject,Aggregation)
```

2. **Relations used solely in expert diagrams**

```
hasMatchingTraitID(Trait,TraitID,DataSource)
hasGlobalPriority(DataSource,DataSource)
hasPriorityForTrait(Trait,DataSource,DataSource)
```

3. **Relations used solely in end-user queries**

```
hasSingleDBTraitValue(Trait,Species,GrowingCondition,Value,Source)
hasTraitValue(Trait,Species,GrowingCondition,Value)
hasFunctionValue(Function,Species,Value,Reliability,GrowingCondition)
hasUserValueOfFunction(Function,Species,Value,Reliability,GrowingCondition)
hasComputedValueOfFunction(Function,Species,Value,Reliability,GrowingCondition)
hasServiceValue(Service,Species,Value,Reliability,GrowingCondition)
```

Figure 5: Main concepts and relations of the domain ontology. The name given to a relation's argument reflects its type (e.g., `Trait` has type `trait`), except for literal values: `Weight`, `Value` and `Reliability` are decimal numbers and `TraitID` is a string.

```
%Upper part of the diagram:
service("nitrogen supply to the vine").
function("mineralization of organic matter").
function("soil exploration and competition with vines").
function("symbiotic fixation of atmospheric nitrogen").
isPositivelyLinkedToService("mineralization of organic matter",
                            "nitrogen supply to the vine").
isNegativelyLinkedToService ("soil exploration and competition with vines",
                            "nitrogen supply to the vine").
isPositivelyLinkedToService ("symbiotic fixation of atmospheric nitrogen",
                            "nitrogen supply to the vine").
hasAggregationMethod("nitrogen supply to the vine", fct:mean).
...
% Links between trait “nitrogen content of shoots and leaves" and database trait IDs:
trait("nitrogen content of shoots and leaves").
dataSource(try).
dataSource(db2).
hasMatchingTraitID("nitrogen content of shoots and leaves","339",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","408",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","502",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","1126",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","5",db2).
```

Figure 6: (Partial) logical translation of the diagram from Figure 4

source database, database mappings from a source database to the working database, and data-to-knowledge mappings from the working database to the fact base.

We illustrate our methodology on our use-case, considering the database TRY as well as a database that has been built within a French study on the interests of intermediate crops (`https://methode-merci.fr`). As the latter is confidential, we refer to it as “DB2”. DB2 provides records for 58 species according to 7 traits. Although very small, this database has the advantage of being devoted to herbaceous species grown in agroecosystems (which is pertinent for the issue of selecting service crops in vineyards) and most trait values provided for these species are not filled in TRY. Besides, it allows us to implement the principle of a multi-database setting. Note that both databases use the same standard names for plant species, as defined in *The WFO Plant List* (`http://www.worldfloraonline.org/`), with the help of the *Taxonomic Name Resolution Service* (`https://tnrs.biendata.org/`).

4.1. From formatted text data to clean structured data

TRY data is available on request for specific traits (we asked for 52 traits, which yielded about 151k species—request made in July 2021) and comes as a formatted text file (similar to a csv file). This file contains so-called *observations* (about 1.6M); each observation (identified by an ID) corresponds to measurements of traits (identified by IDs) on an individual plant (associated with a species or genus ID). Actually, a trait is itself divided into subgroups called “Data Names”, identified by “Data ID”s, and the measures are given for a subgroup. For the sake of simplicity, we will ignore this level in our explanations.

An observation is described by several lines, which provide the measurements themselves and their units of measure, as well as contextual information (called covariates) and original dataset provenance. Furthermore, for quantitative traits with at least 1000 occurrences, some data quality checks and standardizations of measure units were made; in particular an estimation of the (un)reliability of the values (called error risk) is given.

However, among our 52 traits, only 8 have all their values expressed in a standardized measure unit. In general, cells have heterogeneous contents in terms of measure units, values taken by a non-numerical field, kind of contextual information, etc. To illustrate, the trait Plant Life Span takes string values among ["Bisannual", "Annual", "Biennial", "Perennial"] but one also finds a lot of other values like "perennial < 20 years", "biasannual", "pere", "nope", "from few decades to more than 60 years", "1", "2", "3", "winter annual", "shrub", "woody", etc. Hence, a step of cleaning is mandatory before this rich source of information can be exploited in an automated way. This step, mainly based on string search, discards irrelevant, doubtful (according to the error risk when available) or unusable information, and transforms values for the retained fields.

We also observed that the growing conditions of the observed plants may lead to large variations of trait values, hence, again by string search, we made the distinction between "natural growing conditions" and "experimental growing conditions".

The abstract form of our *cleaning mappings* is $q_{\text{csv}} \rightarrow q_{\text{DB}}$, where

$q_{\text{csv}} = \text{SELECT } \overrightarrow{X} \text{ FROM line}(\overrightarrow{Y}), \text{ valid}_1(\overrightarrow{Y_1}), \ldots, \text{valid}_k(\overrightarrow{Y_k}),$
$q_{\text{DB}} = \text{INSERT } (f_1(\overrightarrow{X_1}), \ldots, f_p(\overrightarrow{X_p})),$
$\overrightarrow{X}$, $\overrightarrow{Y_1}$, ..., $\overrightarrow{Y_k}$ are sublists of $\overrightarrow{Y}$ and $\overrightarrow{X_1}$, ..., $\overrightarrow{X_p}$ are sublists of $\overrightarrow{X}$.

The query q_{csv} reads each line of the csv file, where $\overrightarrow{Y}$ is the list of values in this line, then checks if the line satisfies all the validity tests expressed by Boolean functions $valid_1(\overrightarrow{Y_1}), \ldots, valid_k(\overrightarrow{Y_k})$, where each $\overrightarrow{Y_i}$ is a sublist of $\overrightarrow{Y}$. If the validity check succeeds, the sublist of values $\overrightarrow{X}$ is passed to the query q_{DB}, which inserts a list of p values into the database DB, each of these values being built from a sublist $\overrightarrow{X_i}$ of $\overrightarrow{X}$ using a transformation function f_i.

At the end of this step, each tuple (observation ID, species ID, trait ID, growing conditions) occurring in the obtained database is associated with a single trait value expressed in a standardized unit (i.e., we chose a unit of measure for each trait).

Although cleaning could have comprised the whole data received from TRY and be independent of the specific vine grassing use case, we performed some selective cleaning for time reasons. In particular, only herbaceous species are relevant for the use case; we constructed this category (which is not a well-defined category with respect to plant taxonomy) from specific values of trait IDs, in order to distinguish species that are herbaceous from the others. Finer categories could be built for other use cases.

The data source DB2 considers solely herbaceous species with natural growing conditions and required no cleaning.

4.2. From databases to the working database

To build the working database, we selected the herbaceous species from our local TRY database (about half of the species) and aggregated all the trait values coming from different observations for a given tuple (species ID, trait ID, growing conditions, source database). Aggregation is performed by taking the mean of the values. DB2 is just imported into the working database. The general form of a *database mapping* is $q_{\text{DB}_i} \rightarrow q_{wDB}$, where q_{DB_i} is a selection query on database DB_i and q_{wDB} inserts the tuples returned by q_{DB_i} into the working database (wDB).

At the end of this step, the working database contains a single trait value for each tuple (species ID, trait ID, growing condition, source database).

4.3. From the working database to logical facts

The computation of species' score for ecosystem functions and services, which is performed at the conceptual level, requires to aggregate values of different traits. To do so, we turn each value associated with a trait ID into a normalized value ranging over the interval $[0, \ldots, 1]$. This is done by the formula $\frac{v - min}{max - min}$, where v is the value of the trait to be normalized for one species and min (resp. max) is the minimum (resp. maximum) value of the trait for all species in the working database. Note that v is actually the value for a Data Name of the trait, and we take for the trait value the mean of the normalized values of its Data Names.

A data-to-knowledge mapping is of the general form $q_{wDB} \rightarrow q_{FB}$, where q_{wDB} is a selection query on the working database (wDB) and q_{FB} inserts the facts obtained from the tuples returned by q_{wDB} into the fact base.

Finally, the *data-to-knowledge* mappings yield facts of the following shape:

```
hasInitialValue(TraitID, SpeciesID, GrowingCondition, NormalizedValue, DB).
hasSpeciesID(Species, SpeciesID, Database).
```

Note that facts with predicate `hasInitialValue` still consider trait IDs and not the traits defined by the agronomists. It will be the role of rules to associate values to expert traits.

4.4. The issue of missing values

Before going into further detail, we have to point out the issue of missing values. Indeed, despite TRY is a very large data source, missing values appeared to be a crucial issue. More precisely, the three modeled services involve 52 traits and the associated observations concern $n = 150976$ species, among which $n_h = 70152$ are herbaceous species. One trait, namely "plant growth form", has a value for 97% of the n species and almost 100% of the n_h herbaceous species. However, all the other traits are only filled for few species: the best case is for the trait "nitrogen fixation capacity", which takes a Boolean value, and is filled for 8,8% of the n species and 6,4% of the n_h species; the worst case is for a trait filled for a single non-herbaceous species. Note that herbaceous are not a disadvantaged category, as for most traits the proportion of species with a trait value is higher among herbaceous than in the global set. As will become clear in the following, we had to make choices to face this issue of missing values.

5. Reasoning

So far, we explained how to build the initial fact base, composed of expert facts (obtained from the expert diagrams) and data facts (obtained by the mappings). We now present the different kinds of rules that allow to do reasoning on the fact base. We remind that these rules are used to saturate the fact base, thereby expliciting inferences. This section also discusses the querying capabilities of the system, as well as its qualities regarding genericity, evolutivity and explanability.

5.1. Overview of the different kinds of rules

As already mentioned (Section 3.3), the *domain ontology* provides a vocabulary consisting of concepts and relations that are meaningful for users, being expert users (formalization of diagrams) or end-users (formulation of queries). The main concepts and relations are listed in Figure 5. The domain ontology furthermore defines the semantic relationships between these concepts and relations, in the form of a set of simple rules expressing specializations between concepts and between relations, as well as relation signatures. For instance, the rule `isPositivelyLinkedToFunction(Trait,Function)` → `isLinkedToFunction (Trait,Function)` says that the relation `isPositivelyLinkedToFunction` is a specialization of the relation `isLinkedToFunction`, while the rules `isLinkedToFunction(Trait,Function)` → `trait(Trait)` and `isLinkedToFunction(Trait,Function)` → `function(Function)` define the signature of the latter relation: the first argument is a trait and the second argument is a function.

The other kinds of rules do not only use the ontological vocabulary but also other predicates needed in intermediate steps of computation. In particular, data facts use the predicate `hasInitialValue` which does not belong to the ontological vocabulary: rules will allow to produce higher-level facts (with ontological predicate `hasTraitValue`) by aggregating and combining facts, as explained next.

Apart from the ontology, the rules can be partitioned into four subsets:

A) Rules that gather into a single fact each element of a diagram with all its sucessors and the associated aggregation method (i.e., a service, function or trait, with all the related functions, traits, or trait IDs, respectively). This first step is mandatory for the computation of aggregations (of function values, trait values or values of traits IDs, respectively).

B) Rules that compute the value of a functional trait for a species, which requires to combine data facts and expert facts; these rules aggregate the values of exchangeable trait IDs associated with the considered trait, while taking database preferences into account.

C) Rules that compute function and service values for a species, with a level of reliability.

D) Rules that account for user feedback, by considering the user as a preferred source of information that may provide trait or function values.

The full list of rules, including the domain ontology, can be found in the complementary material.

5.2. Rules to group expert facts (A)

To compute various aggregations, we need to know the set of traits (resp. functions) linked to a function (resp. service) and its cardinality. The same holds for the set of trait IDs associated with an expert trait. That is why some rules operate on expert facts to produce facts of the following shape:

- `isLinkedTo`k`Traits(Function,Trait1,Weight1,...,Trait`k`,Weight`k`,Aggregation)` to link an ecosystem function (Function) to its k underlying traits (Trait1 ... Trait$k$), with associated weights (Weight1 ... Weight$k$) and the method of aggregation of these trait values (Aggregation). Here, the predicate has arity $2k + 2$.
- `isLinkedTo`k`Functions(Service,Function1,Weight1,...,Function`k`,Weight`k`,Aggregation)`, built similarly to link a service to its k underlying functions.
- `has`k`TraitIDs(Trait, TraitID1, ..., TraitID`k`, DataSource, Aggregation)` to link a trait to its k exchangeable trait IDs in a data source.

Note that these compact relations of high arity could have been directly produced when translating the diagrams, however we preferred to keep expert facts as simple as possible.

5.3. Rules to consolidate trait values (B)

As seen in Section 3.2, a trait (from an expert diagram) can be associated with several exchangeable trait IDs in each database. The aggregation of their normalized values for a species S yields the trait value for S according to the considered database.

The interest of performing this aggregation is that it yields more observations for a species, hence leads to retrieve more species. Consider for instance the service "nitrogen supply to the vine" (Figure 4), which involves nine traits. Among these nine traits, six have several exchangeable trait IDs. Table 1 shows the number of retrieved species for each exchangeable trait ID (third column) and for the set of all trait's exchangeable IDs (last column). Clearly, considering exchangeable traits significantly increases the number of retrieved species. For instance, the trait "specific leaf area" is associated with four IDs in TRY, which are filled in for respectively 403, 7485, 6705 and 12584 species, which at the end yields 16006 species with a value for this trait, that is about 3500 more species than with the most favourable trait ID.

Experts traits	TRY IDs	# species	# species
Specific leaf area	3086	403	16006
	3115	7485	
	3116	6705	
	3117	12584	
Dry mass of plants, shoots, leaves	388	68	883
	403	750	
	700	227	
Nitrogen content of shoots and leaves	339	196	333
	408	126	
	502	2	
	1126	115	
C/N ratio of the plant, shoot, leaf, litter	146	4717	6205
	150	155	
	409	2050	
	1021	16	
Root length density	1508	24	80
	2025	51	
	2281	27	
Specific root length	614	1062	1369
	1080	494	

Table 1: Number of species per trait for each exchangeable ID (3rd column) and for the aggregation of all exchangeable IDs (4th column).

This step produces facts of the following form:

`hasSingleDBTraitValue(Trait,Species,GrowCond,NormalizedValue,DB).`

Finally, additional rules enable one to exploit the preference order on databases to retain the preferred available trait value and compute facts of the following form:

`hasTraitValue(Trait,Species,GrowCond,NormalizedValue).`

5.4. Rules to compute ecosystem function and service values (C)

The aim of these rules is to express that the score of a species for an ecosystem function is the aggregation of its values for all traits participating to the function. And similarly to go from ecosystem functions to

```
% (R1):  both traits filled, 100% reliability
isLinkedTo2Traits(Function,Trait1,Link1,Trait2,Link2,Aggregation),
hasTraitValue(Trait1,Species,V1,GrowCond),
hasTraitValue(Trait2,Species,V2,GrowCond)
→ hasCalculatedValueOfFunction(Function,Species,
      fct:aggreg2Links(Aggregation,V1,Link1,V2,Link2),100,GrowCond).

% (R2):  first trait filled, 50% reliability
isLinkedTo2Traits(Function,Trait1,Link1,Trait2,Link2,Aggregation),
hasTraitValue(Trait1,Species,V1,GrowCond),
not hasTraitValue(Trait2,Species,V2,GrowCond)
→ hasCalculatedValueOfFunction(Function,Species,V1,50,GrowCond).

% (R3):  second trait filled, 50% reliability
% similar to (R2)
```

Figure 7: Rules to compute the value of an ecosystem function (first step)

services. However, we face again here the problem of missing values: it often happens that not all the values of traits participating in a function are filled for some species. The natural way of doing, first recommended by the agronomists, was to discard species for which a required trait value was missing. However, this made almost all the species disappear when coming to ecosystem services.

Therefore, we decided to add a parameter to specify the *reliability* of the computed value of a function (respectively of a service). This allows one to return more species, which is in line with the objective of "widening the space of possibilities". The reliability parameter indicates the proportion of valued traits among all the traits associated with the function (resp. service), i.e., $\frac{n_r}{n_t}$ where n_r is the number of traits with a value and n_t the total number of traits attached to the function (resp. service). To illustrate, Figure 7 depicts the rules allowing to compute the value of an ecosystem function linked to 2 traits: (R1) considers the case where both traits are filled (then the reliability of the result is 100%) and (R2-R3) the case where only one trait is filled (then the reliability is 50%). The produced facts have the shape `hasCalculatedValueOfFunction(Function,Species,Value,Reliability)`.

5.5. Rules to account for user feedback (D)

Finally, feedback from the end-user may palliate missing values for traits or dispute the value assigned to an ecosystem function for a specific species. In the case of traits, the user is simply considered as a data source that has the highest priority, associated with trivial mappings. Facts coming from the user have the following shape: `hasSingleDBTraitValue(Trait,Species,GrowCond,Value,User)`, where the variable `User` denotes the source associated with the user. Then, facts with predicate `hasTraitValue` are produced as previously. In the case of functions, facts coming from the user have the following shape: `hasUserValueOfFunction(Function,Species,Value,Reliability)`. Rules that compute the value of a function give priority to facts with that predicate over those with predicate `hasCalculatedValueOfFunction`, as illustrated by Figure 8.

5.6. Querying the system

By querying the saturated fact base, one obtains all the answers that are logically entailed by the knowledge base composed of the initial fact base and the rules. End-user queries are formulated in a

```
% if there is a user value for the function
hasUserValueOfFunction(Function,Species,V,Reliability,GrowCond)
→ hasFunctionValue(Function,Species,V,Reliability,GrowCond).
% otherwise
hasCalculatedValueOfFunction(Function,Species,V,Reliability, GrowCond),
not hasUserValueOfFunction(Function,Species,V2,Reliability2,GrowCond)
→ hasFunctionValue(Function,Species,V,Reliability,GrowCond).
```

Figure 8: Rules to compute the value of an ecosystem function (second step)

privileged way using the vocabulary of the domain ontology (in particular relations listed in Points 1 and 3 of Figure 5). This vocabulary allows to query expert diagrams (e.g., which traits are negatively linked to an ecosystem function that underlies the service "nitrogen supply to the vine"?) as well as the contributions of species to ecosystem functions and services (e.g., which are the 10 best species for the service "nitrogen supply to the vine" with a reliability of the result at least 60%?).

Although we distinguish the vocabulary relevant to an end-user, the whole saturated fact base could be queried. As already mentioned, the saturated fact base is stored in a relational database, which allows to benefit from the whole power of SQL.

Query answering is the basis of higher-level features, which should come with a dedicated user interface. For example, one could provide an overview of relevant functional traits and their role in the various ecosystem services; or compare different species according to their contribution to different functions and services; or, offer the user ways of interacting with the system to study the effects of modifying trait or function values.

5.7. *Genericity, evolutivity and explanability*

At the beginning of Section 2, we listed desired qualities of the constructed system. We mentioned that the first ones were ensured by the OBDA paradigm, which allows one to query data at a conceptual level while taking reasoning into account. We now discuss the other desired qualities: genericity of the formalization, evolutivity of the system and explanability of query answers.

Regarding genericity, expert knowledge about traits, functions and services is formalized as two types of objects:

- *Rules* handling the passage from database values to trait, function and service values. Importantly, these rules are generic in the sense that they do not consider specific traits, functions or services, nor specific aggregation methods, hence they are not tied to specific diagrams, nor to specific data sources.
- *Facts* that describe specific diagrams, including their links to database IDs.

Whereas rule-based expert systems often rely on carefully crafted rules that are tailored for a specific use-case, our rules could in principle be applicable to any use-case that follows the trait-function-service framework.

This formalization moreover allows for a smooth evolution of expert knowledge encoded in diagrams, without impact on the rules. Indeed, updates in diagrams, including the introduction of new diagrams, only leads to updates in expert facts. Similarly, updates in the content of a source file or a database only leads to trigger again the associated mappings. The introduction of a new data source requires to add adequate mappings, independently from those already associated with the other sources, and to link functional traits in expert diagrams to this new data source by identifying relevant trait IDs and establishing priorities among data sources. Obviously, the saturation of the fact base has to be updated when the fact base is updated.

Explanability of answers to queries directly follows from the chosen KR language. More precisely, the successive rule applications can be recorded, so that the way each derived fact has been obtained can be traced back to initial facts. However, we would like explanations that are intelligible to an end-user, which means that they must be structurally simple, short and rely on a vocabulary that is meaningful to her. As we have seen, rules may use predicates that do not belong to the ontological vocabulary. Hence, further theoretical work is required to define an adequate way of composing successive rule applications in order to "forget" such intermediate predicates, an issue related to both axiom pinpointing (Peñaloza, 2020) and forgetting in ontologies (Eiter and Kern-Isberner, 2019). Note however that the main queries of interest are about the contributions of species to services: here, expert diagrams (which are provided with a formal representation based on the ontological vocabulary) naturally offer a view of how the values associated with a given species are computed, from its trait values to its service score. Such visualization is synthetic and understandable by an end-user. The development of a graphical query interface dedicated to end-users and provided with explanation facilities is part of our agenda.

6. Evaluation

In order to assess the applicative validity of the approach, we carried out an evaluation on a specific ecosystem service for a selected set of species, and compared the scores provided on the one hand by the tool and on the other hand by the domain literature.

Specifically, we proceeded as follows:

1. *Selection of a service*: we chose the service of "nitrogen supply to the vine" (Figure 4), since it is one of the most documented ecosystem services regarding the use of service crops in viticulture.
2. *Selection of species*: Based on a literature search, we selected a set of 23 herbaceous species commonly used as service crops in agriculture, and considered to be well-documented in the scientific and grey literature (see the list of species in Table 2).
3. *Species ranking from the literature*: we retained 16 papers from the literature, each one providing a comparison between some of the selected species for the selected service. Note that each paper covers only some of the selected species (from 2 to 8 species). From each paper, we extracted an ordering on the species covered by the paper.[1]

Given two species s_i and s_j, let us note $s_i > s_j$ if s_i is deemed (strictly) better than s_j for the service according to at least one document. To obtain a global ranking, we built a directed graph, whose nodes are the species s_i and there is an edge (s_i, s_j) if $s_i > s_j$ holds for at least one document. This graph is depicted in Figure 9, after removal of all redundant edges for the sake of clarity (an edge (s_i, s_j) is redundant if there is already a path from s_i to s_j in the graph). We can see that the results in the literature are remarkably coherent, since the graph is circuit-free. Hence, the graph provides a partial order on the whole set of species. We then considered all the linear extensions of this partial order (i.e., all the total orders compatible with this partial order), and assigned to each species a score equal to the average of its ranks in the extensions.

---

[1] For details, see additional documentation: file "literature-ranking.pdf" at `https://drive.google.com/drive/folders/1srgjT6AoH9PktaniqtbwYwCXXNr7_z4q?usp=share_link`.

For instance, if we only consider the subgraph induced by nodes 21, 19 and 23, we have $21 > 23$ and $19 > 23$, hence 2 linear extensions: $21 > 19 > 23$ and $19 > 21 > 23$. The whole graph admits more than 44 millions linear extensions. The obtained scores are shown in Table 2, column "Expert ranking". Note that species 21 and 19 obtain the same score, whereas they are incomparable in the graph. The service value computed by the tool for each species, and the reliability of the computation are shown in columns "Tool's value" and "Tool's reliability", respectively. We recall that the reliability depends on the proportion of traits for which a value could be computed.

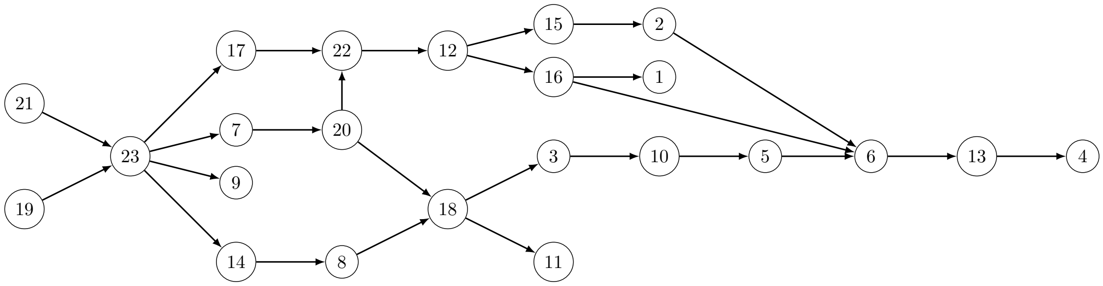
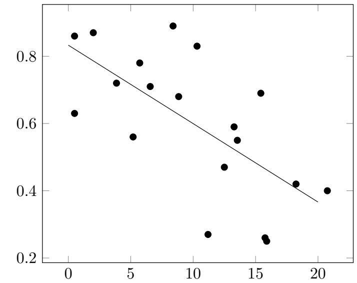


Figure 9: Ranking graph of species for the service of "nitrogen supply to the vine", based on the literature.. Each node represents a species and is labeled by the corresponding species# from Table 2. An edge from s_i to s_j means that $s_i > s_j$.

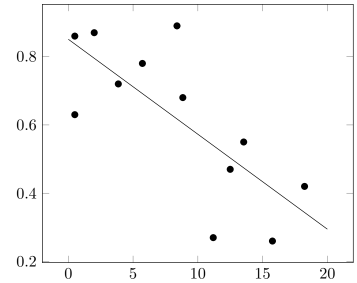
Species#	Species	Experts' ranking	Tools' value of service	Tools' Reliability
1	*Avena sativa*	18.55	—	—
2	*Brassica napus*	16.45	0.26	56
3	*Dactylis glomerata*	11.65	0.27	78
4	*Fagopyrum esculentum*	21.68	0.40	44
5	*Hordeum vulgare*	16.58	0.25	44
6	*Lolium multiflorum*	19.05	0.42	67
7	*Lotus corniculatus*	3.95	0.72	67
8	*Lupinus albus*	6.79	0.71	33
9	*Medicago lupulina*	12.5	0.47	67
10	*Medicago sativa*	14.12	0.55	67
11	*Melilotus officinalis*	16.09	0.69	33
12	*Pisum sativum*	10.73	0.83	44
13	*Raphanus sativus*	20.37	—	—
14	*Rumex acetosa*	4.4	—	—
15	*Secale cereale*	13.84	0.59	44
16	*Sinapsis alba*	14.1	—	—
17	*Trifolium alexandrinum*	5.35	0.56	44
18	*Trifolium incarnatum*	9.19	0.68	55
19	*Trifolium pratense*	0.5	0.86	78
20	*Trifolium repens*	5.9	0.78	78
21	*Trifolium subterraneum*	0.5	0.63	55
22	*Vicia faba*	8.71	0.89	56
23	*Vicia villosa*	2	0.87	55

Table 2: Expert and tool rankings for the service of "nitrogen supply to the vine". The 3rd column gives the expert score. The 4th and 5th columns give the value output by the tool and its reliability, resp. Some species do not occur in the databases, hence the —.

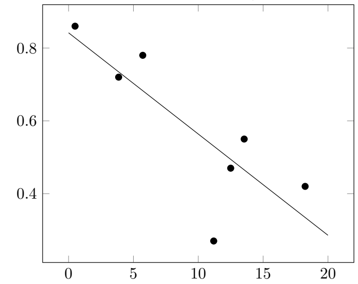
Finally, we studied the correlation between the expert and tool scores, according to Pearson correlation coefficient r. In our case, a perfect correlation is reflected by $r = -1$ (as the tool and the expert rank species in opposite orders), a perfect inverse correlation by $r = 1$, and an absence of correlation by $r = 0$. The



(a) All reliabilities (r-value = -0.667, p-value = 0.002)



(b) Reliability ≥ 50 (r-value = -0.744, p-value = 0.006)



(c) Reliability ≥ 60 (r-value = -0.808, p-value = 0.028)

Figure 10: Correlation between expert scores (x axis) and tool scores (y axis); each point denotes a species. The line is a linear regression line.

correlation is considered good if $r \leq -0.5$. With respect to the p-value, there is a very strong case against the null hypothesis when $p \leq 0.01$ and a strong one when $p \leq 0.05$.

When considering the whole set of species (Graph 10a), the correlation coefficient is already significant ($r = -0.667$ and $p = 0.002$). We furthermore observe that the reliability of the tool's score is a crucial parameter. Indeed, the correlation coefficient reaches -0.74 (with $p = 0.006$) when the set of species is restricted to those whose computed value has reliability at least 50% (Graph 10b) and -0.81 (with $p = 0.028$) for reliability at least 60% (Graph 10c). In the case of this service, a reliability higher or equal to 60% for a species means that at least 6 of the 9 traits actually have a value by the predicate `hasTraitValue` for this species. On the graph associated with reliability at least 60% (Graph 10c), we can see that only one species (*dactylis glomerata*) is quite far from the regression line. If we go back to the TRY data, we notice that for this species several traits have important value variations, which might be explained by variations of the observation contexts (e.g., elevation and phosphorus content of the soil in this case). Finally, note that the improvement of the correlation coefficient goes with a less good p-value, which is due to a reduced sample size.

7. Discussion and Conclusion

In this paper, we studied the feasibility of an original approach for the selection of service plant species in agroecology. The main idea was to combine a formal representation of scientific knowledge in agronomy on the relationships between functional traits, ecosystem functions and services, with independently-built data sources, in particular the database TRY that already integrates the main data sets about functional traits collected by the research community in ecology. Data is selected and integrated through declarative mappings and knowledge is described by logical rules independent from the use-case. These two features provide greater genericity and facilitate the evolution of the system. They also make it possible to follow the reasoning steps, hence to explain the results to a user.

The implementation of this approach however faced two difficulties. First, data cleaning was a mandatory step before any automated exploitation of the data. Second, the problem of missing data turned out to be crucial. This issue, extensively analyzed in (Kattge and al., 2020), is probably exacerbated by the fact that data was collected for research in ecology, which focuses more on natural and spontaneous species than on species bred for agricultural purposes. Despite this, the evaluation we carried out shows that very satisfactory results can be obtained as long as the proportion of missing values is not too high. Since the TRY initiative is developing, in both volume and standardization of the data, and there is a growing effort to produce trait data in agronomy (Blesh, 2018; Garcia et al., 2020, 2019; Wood et al., 2015; Martin and Isaac, 2015), one can only expect more accurate results in the next years.

Outside missing values, a reason for observed discrepancies between the tool's results and results from the literature seems to be, for some traits, a high variation of the values measured for individual plants of the same species. Besides potential problems of reliability of some observations, it is likely that such variations are related to differences in the contexts of the observations, be it the cultivation context (like characteristics of the soil, climate and plant management, ...) or the plant growth stage. Apart from the growing conditions of plants, we did not take context into account for two reasons: first, it would have reduced the number of relevant observations, hence worsen the problem of missing values; second, many observations in TRY are not provided with adequate context information (beyond the problem of automatically exploiting such information). Once again, the development of data on traits should allow to leverage contextual information in the near future.

Contextual information could be considered to homogenise the conditions of the measurements themselves, like the plant growth stage. More importantly, it could be exploited in relationship with information about the soil properties and climate, and/or cropping practices. Indeed, if the context of measurement of functional traits in the databases is important to limit the variability of their values, the context of installation of service plants in the field (soil, climate, cropping practices) is also important to ensure that they will actually provide the expected functions and services. As already mentioned, functional traits are relevant to the effects of plants on ecosystem properties (effect traits) and/or the response of plants to the environment (response traits) (Damour et al., 2018). In other words, traits also give us information about the ability of a species to grow in a specific environment. Our expert diagrams involve effect traits: for these traits, the pedoclimate can be considered *a priori* to restrict relevant data observations. Technically, this requires to specialize mappings so that contextual information of interest is imported into the working database, and to extend the logical vocabulary to represent it (for instance, by increasing the arity of predicates to include new parameters, as we did for the plant growing conditions). Response traits (like resistance to frost or drought, or the ability to grow in a nutrient-poor soil) can be taken into account *a posteriori* to filter species returned as answers. To conclude, our framework allows for such extensions, the limiting factor being the availability of data.

Similarly, our modeling allows to refine expert diagrams, in particular by weighting traits' participations and making aggregation methods more complex, the limiting factor being whether state-of-the-art domain knowledge enables such refinements. This will be possible within the framework of design workshops (Notaro et al., 2022; Naulleau et al., 2022), where participants (researchers, technicians, farmers) will be able to parametrize and refine the diagrams, and also to feed their thinking with the tool outputs, which will provide innovative options while making the underlying reasoning explicit.

Finally, this work also supports future research about the combination of various plant species to provide the expected functions and services, in comparison to single species (Storkey et al., 2015; Finney et al., 2017).

Acknowledgements

The authors are grateful to Gaëlle Damour, Eric Garnier, Elena Kazakou and Aurélie Métay for fruitful discussions and contributions to trait-function-service diagrams, as well as to Sébastien Minette for providing the second traits database. This work was supported by a governmental grant managed by the Agence Nationale de la Recherche (ANR) within the framework of the "Investissements d'Avenir" program under the reference ANR-16-CONV- 0004 (`https://www.hdigitag.fr/en/`).

References

Abalos, D., van Groenigen, J., Philippot, L., Lubbers, I., and De Deyn, G. (2019). Plant trait-based approaches to improve nitrogen cycling in agroecosystems. *Journal of Applied Ecology*, 56(11):2454–2466. 10

Abiteboul, S., Hull, R., and Vianu, V. (1994). *Foundations of Databases*. Addison Wesley. 6, 8

Baader, F., Horrocks, I., Lutz, C., and Sattler, U. (2017). *An Introduction to Description Logic*. Cambridge University Press. 6

Baget, J., Leclère, M., Mugnier, M., Rocher, S., and Sipieter, C. (2015). Graal: A toolkit for query answering with existential rules. In Bassiliades, N., Gottlob, G., Sadri, F., Paschke, A., and Roman, D., editors, *Proceedings of RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*, volume 9202 of *Lecture Notes in Computer Science*, pages 328–344. Springer. 3

Blesh, J. (2018). Functional traits in cover crop mixtures: Biological nitrogen fixation and multifunctionality. *Journal of Applied Ecology*, 55:38–48. 23

Buron, M., Goasdoué, F., Manolescu, I., and Mugnier, M. (2020). Ontology-based RDF integration of heterogeneous data. In *Proceedings of EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020*, pages 299–310. 3

Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., and Xiao, G. (2017). Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487. 3

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R., Ruzzi, M., and Savo, D. F. (2011). The mastro system for ontology-based data access. *Semantic Web*, 2(1):43–53. 3

CAPS (2017). Outil CAPS - colza associé. 2

Chein, M. and Mugnier, M. (2009). *Graph-based Knowledge Representation - Computational Foundations of Conceptual Graphs*. Advanced Information and Knowledge Processing. Springer. 11, 27

Cherr, C. M., Scholberg, J. M. S., and McSorley, R. (2006). Green Manure Approaches to Crop Production: A Synthesis. *Agronomy Journal*, 98(2):302–319. 10

Damour, G., Garnier, E., Navas, M. L., Dorel, M., and Risède, J.-M. (2015). Chapter three - using functional traits to assess the services provided by cover plants: A review of potentialities in banana cropping systems. volume 134 of *Advances in Agronomy*, pages 81–133. Academic Press. 2, 10

Damour, G., Navas, M. L., and Garnier, E. (2018). A revised trait-based framework for agroecosystems including decision rules. *Journal of Applied Ecology*, 55(1):12–24. 24

Duru, M., Therond, O., Martin, G., Martin-Clouaire, R., Magne, M.-A., Justes, E., Journet, E.-P., Aubertot, J.-N., Savary, S., Bergez, J.-E., and Sarthou, J.-P. (2015). How to implement biodiversity-based agriculture to enhance ecosystem services: a review. *Agronomy for Sustainable Development*, 35. 1

Eiter, T. and Kern-Isberner, G. (2019). A brief survey on forgetting from a knowledge representation and reasoning perspective. *Künstliche Intell.*, 33(1):9–33. 20

Finney, D. M., Murrell, E. G., White, C. M., Baraibar, B., Barbercheck, M. E., Bradley, B. A., Cornelisse, S., Hunter, M. C., Kaye, J. P., Mortensen, D. A., Mullen, C. A., and Schipanski, M. E. (2017). Ecosystem Services and Disservices Are Bundled in Simple and Diverse Cover Cropping Systems. *ael*, 2(1):0. 24

Freschet, G. T. and Roumet, C. (2017). Sampling roots to capture plant and soil functions. *Functional Ecology*, 31(8):1506–1518. 10

Funk, J. and Wolf, A. (2016). Testing the trait-based community framework: Do functional traits predict competitive outcomes? *Ecology*, 97. 10

Garcia, L., Celette, F., Gary, C., Ripoche, A., Valdés-Gómez, H., and Metay, A. (2018). Management of service crops for the provision of ecosystem services in vineyards: A review. *Agriculture, Ecosystems & Environment*, 251:158–170. 2

Garcia, L., Damour, G., Gary, C., Follain, S., Le Bissonnais, Y., and Metay, A. (2019). Trait-based approach for agroecology: contribution of service crop root traits to explain soil aggregate stability in vineyards. *Plant and Soil*, 435. 2, 23

Garcia, L., Metay, A., Kazakou, E., Storkey, J., Gary, C., and Damour, G. (2020). Optimizing the choice of service crops in vineyards to achieve both runoff mitigation and water provisioning for grapevine: a trait-based approach. *Plant and Soil*. 2, 23

Hanisch, M., Schweiger, O., Cord, A., Volk, M., and Knapp, S. (2020). Plant functional traits shape multiple ecosystem services, their trade-offs and synergies in grasslands. *Journal of Applied Ecology*, 57:1535–1550. 10

Kattge, J. and al. (2020). Try plant trait database—enhanced coverage and open access. *Global Change Biology*, 26(1):119–188. 2, 23

Lenzerini, M. (2018). Managing data through the lens of an ontology. *AI Mag.*, 39(2):65–74. 3

Lifschitz, V. (2019). *Answer set programming*. Springer Heidelberg. 6

Martin, A. R. and Isaac, M. E. (2015). Functional traits in agroecoogy: a blueprint for research. *Journal of Applied Ecology*, 52:1425–1435. 23

Mugnier, M.-L. and Thomazo, M. (2014). An introduction to ontology-based query answering with existential rules. In *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014. Proceedings*, volume 8714 of *Lecture Notes in Computer Science*, pages 245–278. Springer. 6

Naulleau, A., Gary, C., Prévot, L., Vinatier, F., and Hossard, L. (2022). How can winegrowers adapt to climate change? a participatory modeling approach in southern france. *European Journal of Agronomy*, 203(103514). 24

Notaro, M., Deheuvels, O., and Gary, C. (2022). Participative design of the spatial and temporal development of improved cocoa agroforestry systems for yield and biodiversity. *European Journal of Agronomy*, 132(126395). 24

Ozier-Lafontaine, H., Blazy, J.-M., Publicol, M., and Melfort, C. (2010). SIMSERV - Expert system of selection assistance of service plants. 2

Peñaloza, R. (2020). Axiom pinpointing. In Cota, G., Daquino, M., and Pozzato, G. L., editors, *Applications and Practices in Ontology Design, Extraction, and Reasoning*, volume 49 of *Studies on the Semantic Web*, pages 162–177. IOS Press. 20

Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati, R. (2008). Linking data to ontologies. In Spaccapietra, S., editor, *Journal on Data Semantics X*, pages 133–173, Berlin, Heidelberg. Springer Berlin Heidelberg. 3

Poorter, H. and Garnier, E. (2007). Ecological Significance of Inherent Variation in Relative Growth Rate and Its Components. In Pugnaire, F. I. and Valladares, F., editors, *Functional Plant Ecology (2nd ed)*, volume 000, pages 67–100. CRC Press, Boca Raton. 10

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley. 11, 27

Storkey, J., Döring, T., Baddeley, J., Collins, R. P., Roderick, S., Jones, H., and Watson, C. (2015). Engineering a plant community to deliver multiple ecosystem services. *Ecological Applications*, 25(4):1034–1043. 24

Van der Wolf, J., Jassogne, L., Gram, G., and Vaast, P. (2019). Turning local knowledge on agroforestry into an online decision-support tool for tree selection in smallholders' farms. *Experimental Agriculture*, 55(S1):50–66. 2

Wood, S. A., Karp, D. S., DeClerck, F., Kremen, C., Naeem, S., and Palm, C. A. (2015). Functional traits in agriculture: Agrobiodiversity and ecosystem services. *Trends in Ecology and Evolution*, 30(9):531–539. 23

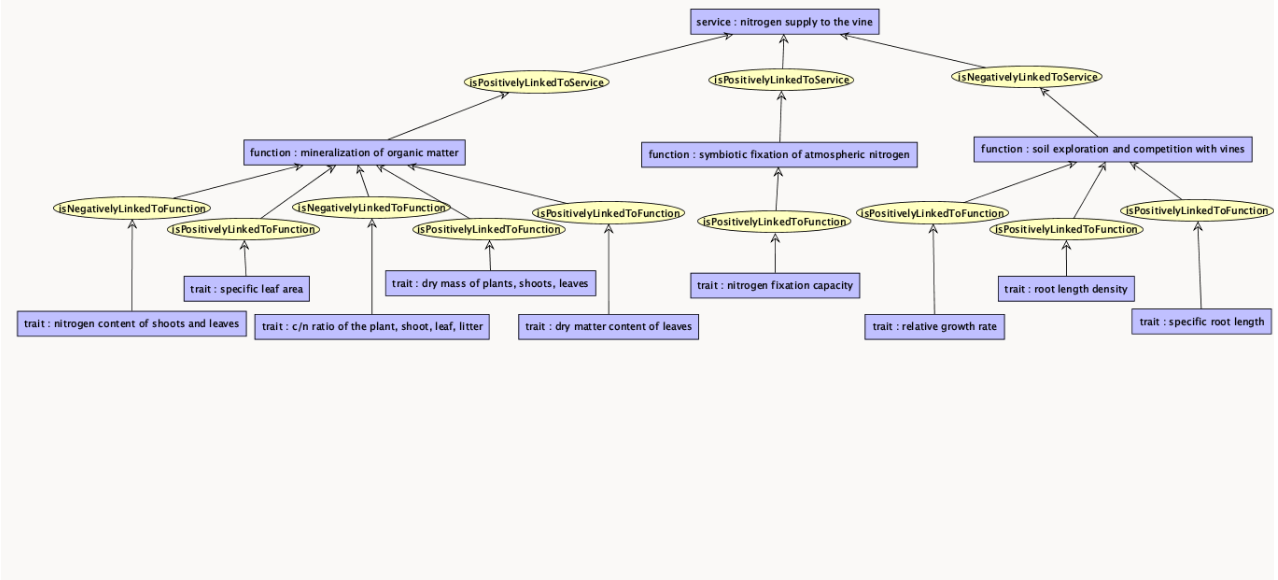
Appendix A. From diagrams to formal conceptual graphs

Conceptual graphs are a KR language in which knowledge is encoded as labeled graphs (in the sense of graph theory) provided with a translation in first-order logic (Sowa, 1984; Chein and Mugnier, 2009). We used them as an intermediate representation between expert diagrams and logics, which has the advantage of being both easy to understand by domain experts and formal. To draw and manage conceptual graphs, we used the visual tool Cogui (`https://www.lirmm.fr/cogui/`), which implements the framework developed in (Chein and Mugnier, 2009). This framework allows to represent knowledge bases composed of an ontological vocabulary, facts and rules built on this vocabulary, and to do logically-founded reasoning based on graph mechanisms. It does however not provide negation nor computed functions, which is required in some of our rules, hence we only used Cogui to build the expert facts and export them into the logical framework (using Cogui export functionality).

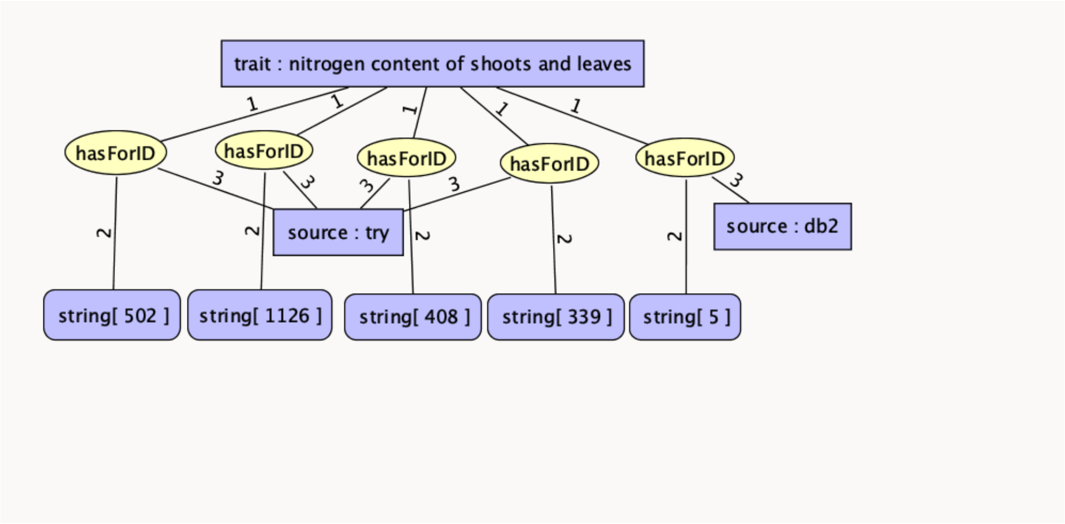
A conceptual graph is made of two kinds of nodes: concept nodes with labels of the form $type : marker$, where *type* is a (list of) concept(s) and *marker* is a constant or the symbol $\star$ denoting an unknown entity (here, we only use constants); and relation nodes labeled by a relation. Each relation node labeled by a relation of arity k is incident to k edges, labeled from 1 to k, that link it to concept nodes.

Figure A.11 depicts conceptual graphs that partially translate the diagram from Figure 4: the top graph describes the trait-function-service relationships and the bottom graph the links between the trait "nitrogen content of shoots and leaves" and the associated trait IDs in TRY and BD2. In the case of binary relations (as in the top graph), Cogui replaces edge labels 1 and 2 by directed edges. Note that different conceptual graphs are implicitely joined on the nodes that share the same constant. Generally speaking, the ability to split graphs into several components facilitates the design activity and provides better readability.

A conceptual graph has a natural translation into a set of logical facts. Concepts and relations from the vocabulary are seen as predicates: unary predicates for concepts and predicates of the same arity for relations. Then, each concept node with label *type:constant* yields a fact of the form *type(constant)*, and each relation node with label r and arity k yields a fact of the form $r(c_1, ..., c_k)$, where c_i is the constant that labels its i-th neighbor. See Figure A.12 for an illustration.



(a) Trait-Function-Service



(b) Nitrogen content of shoots and leaves

Figure A.11: Conceptual graphs associated with the expert diagram from Figure 4. Rectangles (in purple) denote concept nodes and oval shapes (in yellow) relation nodes. Numbers on edges specify the order of arguments for non-binary relations.

```
% Top graph (a)
service("nitrogen supply to the vine").
function("mineralization of organic matter").
function("soil exploration and competition with vines").
function("symbiotic fixation of atmospheric nitrogen").
isPositivelyLinkedToService("mineralization of organic matter","nitrogen supply to the
vine").
isNegativelyLinkedToService("soil exploration and competition with vines","nitrogen
supply to the vine").
isPositivelyLinkedToService("symbiotic fixation of atmospheric nitrogen","nitrogen supply
to the vine").
hasAggregationMethod("nitrogen supply to the vine", fct:mean).
...
% Bottom graph (b)
trait("nitrogen content of shoots and leaves").
dataSource(try).
dataSource(db2).
hasMatchingTraitID("nitrogen content of shoots and leaves","339",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","408",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","502",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","1126",try).
hasMatchingTraitID("nitrogen content of shoots and leaves","5",db2).
```

Figure A.12: (Partial) logical translation of the conceptual graphs from Figure A.11