Supporting Companion Planting with the CoPla Ontology

Giacomo Zamprogno[1,•], Márk Adamik[1,•], Ritten Roothaert[1,•], Ameneh Naghdipour[1,•], Lise Stork[1,*,•], Patrick Koopmann[1,•], Romana Pernisch[1,2,•], Benno Kruit[1], Jieying Chen[1], Ilaria Tiddi[1] and Stefan Schlobach[1]

[1]*Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, The Netherlands*
[2]*Discovery Lab, Elsevier, Amsterdam, The Netherlands*

Abstract
Sustainability in agriculture is crucial for environmental conservation and ecosystem resilience. Within this context, companion planting stands out as a key practice, leveraging synergies between plants for enhanced growth and pest control. However, its broader adoption is hindered by large-scale knowledge and data integration challenges. To bring companion planting forward and closer to interested users, we engineered a semantically rich ontology, CoPla. We used several automated techniques to extract knowledge from various sources and capture different companion and anti-companion mechanisms.We demonstrate CoPla's versatility through three applications using different reasoning mechanisms: identifying plant companionships, evaluating, and optimising garden layouts.

Keywords
Sustainable Agriculture, Ontology-Based Decision Support, Knowledge Representation, Reasoning, Permaculture, Agroecology, Companion Planting, Knowledge Integration, Ontology Engineering

1. Introduction

Sustainability stands at the forefront of global concerns, particularly within the realms of environmental preservation and agricultural innovation.Central to this discourse is the concept of sustainable gardening, which is paramount for fostering green, resilient ecosystems across the globe [1]. Companion planting, traditionally widely adopted in sustainable agricultural movements, such as permaculture [1] and agroecology [2], leverages the synergistic interactions between different plant species and plays a pivotal role in sustainable farming [3, 2]. The main idea is to place together plants that support each other (*companions*), based on a variety of different interaction mechanics—e.g. because a plant attracts insects that help pollinate other

The 2nd International Workshop on Knowledge Graphs for Sustainability (KG4S2024) – Colocated with the 21st Extended Semantic Web Conference (ESWC2024), May 27th, 2024, Hersonissos, Greece.
*Corresponding author.
•These authors contributed equally.
g.zamprogno@vu.nl (G. Zamprogno); m.adamik@vu.nl (M. Adamik); h.m.roothaert@vu.nl (R. Roothaert); a.naghdipour@vu.nl (A. Naghdipour); l.stork@uva.nl (L. Stork); p.k.koopmann@vu.nl (P. Koopmann); r.pernisch@vu.nl (R. Pernisch); b.b.kruit@vu.nl (B. Kruit); j.y.chen@vu.nl (J. Chen); i.tiddi@vu.nl (I. Tiddi); k.s.schlobach@vu.nl (S. Schlobach)
0009-0009-6474-1987 (G. Zamprogno); 0009-0008-7843-6513 (R. Roothaert); 0000-0002-2146-4803 (L. Stork); 0000-0001-5999-2583 (P. Koopmann); 0000-0001-8590-1817 (R. Pernisch); 0000-0002-0228-4823 (B. Kruit); 0000-0002-2497-645X (J. Chen); 0000-0001-7116-9338 (I. Tiddi); 0000-0002-3282-1597 (S. Schlobach)

plants. We similarly consider *anti-companions*; the case where one plant can harm another, for instance, when a plant attracts animals that are pests of other plants. Companion planting not only promotes plant growth and health [4, 5] but also serves as a natural pest deterrent [6] and soil enhancer [5, 7, 8], thus significantly contributing to the viability of agricultural ecosystems.

The current landscape of companion planting knowledge is dispersed across various platforms, including agricultural manuals, scholarly articles, and digital databases, making it challenging for practitioners to access and synthesise this information effectively. This dispersion presents a significant obstacle to the efficient integration and broader application of companion planting strategies. Due to this, people aiming at planting their own garden often lack an overview of companion planting mechanisms, ultimately limiting their impact on sustainability. In particular, gardeners who are new to companion planting might need help with the following tasks: (1) exploring potential companions and anti-companions of plants they are interested in, (2) understanding the reasons behind (anti-) companion relationships, and (3) analysing and improving garden layouts wrt. companionships.

To address these challenges, we developed a semantically rich ontology called **CoPla** that delineates the complex relationships among plant species and their interactions with the environment. It is built in two phases: first, a core conceptualisation was created through a knowledge acquisition phase, where knowledge of companion planting and its various mechanisms was acquired from relevant literature. Second, subclasses and class axioms were automatically created using a rich variety of (web) technologies—from SPARQL querying to scraping of PDFs—to extract knowledge from openly available multi-modal sources. The result is a semantically rich large-scale ontology for informed decision-making in companion planting. To illustrate CoPla's usefulness, we integrated it in a prototypical system to support gardeners addressing the use cases above. In particular, we showcase how different reasoning mechanisms for OWL—deductive inference, explanations and repair—can be used together with the ontology towards the three problems mentioned above. The application of **CoPla** in these scenarios shows that it is manageable and directly applicable in real-world scenarios. Moreover, our method is designed with scalability in mind, allowing for easy adaptation to broader agricultural contexts by simply expanding the underlying knowledge base.

All resources described in this article, which includes the engineered ontology, the code of the engineering process, the backend and frontend, are available in our online repository.[1]

2. Related Work

We divide the related work into (a) companion planting resources, (b) semantic web efforts from other domains, and (c) AI-based applications to help farming and planting.

Companion Planting and Relevant Resources. The seminal book [9] on companion planting dates back to the 80's. The book recollects decades of practices to provide disease- and pest-free gardens using mutual restoration conditions. Some attempts to formalise companion planting practises in the form of charts and tables can be found online. A lookup summary for

[1] https://github.com/kai-vu/companion-planting-decision-support

plant compatibility, for instance, is published online.[2] Plant Interactions are described by the IDEP foundation[3], which provides worldwide training for Permaculture Design. While being semi-formalised, these sources of information are non-standardised and relatively incomplete.

A collaborative encyclopaedia for plant information, cultivation and association is the Practical Plants database[4]. In addition to description of individual plants and their interaction with other plants, descriptions of plant combinations can be created. This polyculture data collection is in its early stage. The EPPO pest database[5] provides an overview of hosting plants for pests, and can be used to determine combinations of plants to cultivate—namely, hosts of known pests should not be combined with plants that can be affected by that particular pest. The US Natural Resources Conservation Service provides a number of conservation resources for planters, researchers, and landowners, including the Soil Taxonomy for soil properties[6]. The Global Biotic Interactions (Globi) is a large open-access dataset to find symbiotic/opportunistic (e.g. predator-prey, pathogen-host or parasite-host) interactions between biological organisms [10]. Globi is a combination of different datasets and vocabularies that can be used to describe plants and their characteristics relevant to permaculture. The TRY database [11] provides information about plant traits (i.e. morphological, anatomical, physiological, biochemical and phenological characteristics of plants) that help determine how plants respond to different environmental and ecosystem factors. Companion planting aspects are missing and could be integrated as part of the dataset, which is not formalised according to Semantic Web standards.

Semantic Web Resources for Companion Planting. A number of semantic web initiatives exist for the agricultural domain. AgroPortal [12] acts on an hub of semantic resources (vocabularies, terminologies, etc.) for the agricultural domain. Agronomic Linked Data [13] is an integration effort to represent plant science data in a large KG. It includes aspects of plant molecular interactions e.g. genes, proteins, metabolic pathways and plant trait associations, and integrates resources and ontologies including the Ensembl plants, UniProtKB, and the Gene Ontology Annotation. An overview of ontologies for the agricultural domain is given in [14]. The multilingual vocabulary FAO [15] allows data classification, interoperability and reuse across applications. While these works do not focus directly on companion planting, they can be useful to enhance plant information using schema and instance alignments.

Plant and organism dynamics are also important concepts in biology and ecology. [16] provides an overview of ontologies for the ecology domain, such as the Extensible Observation Ontology (OBOE), the Environmental Data Exchange (CEDEX) ontology and the Observations Data Model (ODM). The OpenBiodiv [17] resource integrates biodiversity literature into a knowledge graph, including an ontology based on the Global Biodiversity Information Facility (GBIF) structure. Platforms to support KG constructions for biodiversity include [18] and [19], including a knowledge graph to promote meta-analysis and research for soil ecology. Nutritional values of fruits and vegetables can also be found in the FoodKG[7] dataset.

[2]https://www.thespruce.com/companion-planting-with-chart-5025124
[3]https://www.permaculturenews.org/2010/07/30/companion-planting-guide/
[4]https://practicalplants.org/wiki/practical_plants/
[5]https://gd.eppo.int/taxon/CUUPE/pests
[6]https://www.nrcs.usda.gov/resources/guides-and-instructions/soil-taxonomy
[7]https://foodkg.github.io/

AI-based Planting Applications. A number of tools exist to support smart planting. The Almanac Garden Planner[8] is an application to help designing layouts of a garden, and provides insights including planting/harvesting seasons and crop rotation warnings. As a proprietary product, its reusability and interoperability are limited. Elzeard [9] is a crop planning and management tool, designed to replace spreadsheet-based crop management methods. It features a Crop Planification and Production Process Ontology (C3PO) that describes the agricultural production process. The Planting Planner ([10] provides more information on soil conditions, with a focus on flowers. Most of these and other existing applications (GrowIt, Garden Manager, etc.) mainly focus on garden aesthetics rather than farming and sustainability aspects. Similarly, industrial farming and farm managing applications focus on the optimisation of farming processes.

3. The CoPla Ontology

The main purpose of the ontology is to structure knowledge about companion planting and assist garden planners in identifying potential companion relationships. Our development process was guided by the METHONTOLOGY [20] methodology because of its focus on knowledge elicitation from knowledge sources rather than4 domain experts. We started from a list of requirements, for which we developed a core ontology offering the main conceptualisation of the ontology. This core ontology was extended with additional axioms that were generated through automated data integration from openly available multi-modal sources.

Sources of Knowledge. The ontology creation process was informed by a combination of books [21, 22, 23], scholarly articles [3, 6] and openly available companion planting charts[11,12]. The core concepts involved in companion planting, including types of mechanisms that can explain a certain companionship between two plants, were manually derived from books on companion planting mechanisms. After understanding various types of companionship mechanisms better, structured knowledge sources were used to automatically populate the different types of companion relationships and their mechanics, as described further below.

3.1. Ontology Requirements

To determine how the knowledge should be used and organised we devised an informal list of requirements, taking into account the use cases mentioned in Section 1. The ontology should:

R1: include the core concepts for the task of companion planting decision support;
R2: describe different types of companion and anti-companion relationship;
R3: define properties describing various mechanisms of companion planting;
R4: describe qualities of optimal and sub-optimal garden configurations;
R5: model specific plant species and their metadata (common and scientific names).

[8] https://gardenplanner.almanac.com/
[9] https://en.elzeard.co/
[10] https://www.plantingplanner.com/
[11] https://https://www.permaculturenews.org/2010/07/30/companion-planting-guide/
[12] https://www.permablitz.net/wp-content/uploads/2016/08/Poster_GDN_Com_Plant.pdf

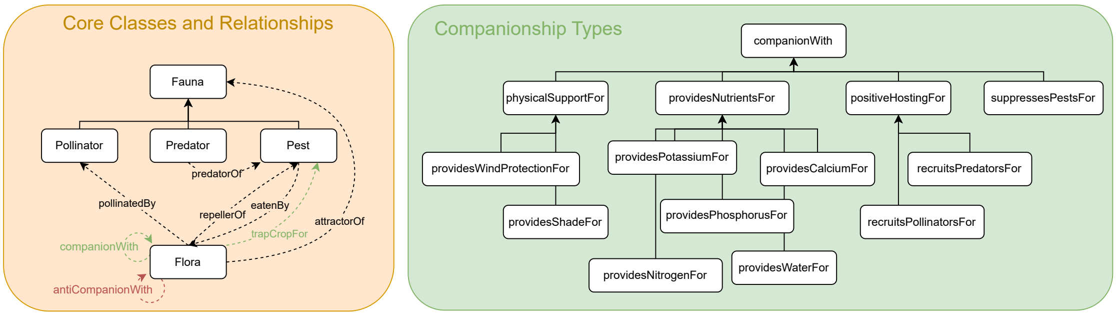


Figure 1: Core classes and relationships of CoPla (left), and the different types of companionship (right).

3.2. Conceptualisation

To address **R1**, the *core ontology* describes classes and relationships that capture the companionship relations (see Figure 1):

Core Classes and Relationships. The core ontology consists of classes `Flora`, `Fauna`, `Predator`, `Pest`, `Pollinator`. `Fauna` refers to the animals involved in companion planting and has three core sub-classes based on the role animals play in determining companionship. Derived from [22], the class `Pollinator` contains all animals that act as pollinator for a given plant. The `Pest` class encompasses all animals that are considered harmful to the plants. These classes are **not** disjoint: an animal can take on various roles, based on the role they play in a companionship mechanism. Members of the `Predator` class represent animals that prey on pests. The `Flora` class contains the different plants that are evaluated for companionship. All flora and fauna are then modelled as subclasses of these core classes.

The core relationships connecting these classes are `eatenBy`, `pollinatorOf`, `predatorOf`, `attractorOf`, `repellerOf`, `trapCropFor`, `companionWith` and `antiCompanionWith`. The `companionWith` and `antiCompanionWith` relationships are defined to express the companionship relation between members of the `Flora class`. To address **R3** and **R2** and expand upon the specific companionship types, some additional relationships are added that describe the interaction between the core class members. The `attractorOf` property is defined between `Flora` and `Fauna`, expressing that some animals are particularly attracted by a specific plant. The `eatenBy` relationship relates animals of type `Pest` to the `Flora` they consume.

The `repellerOf` property describes that some plants prevent pests from feeding on their target plants. `predatorOf` expresses the relationship between members of the `Predator` and the corresponding `Pest` class they feed on. To reflect the interaction between different types of animals that play a role in pollinating certain plants, `pollinatorOf` is used. Lastly, the `trapCropFor` relationship expresses a special type of companion planting technique, which utilises some plant to serve as a decoy and attract pests that can be then effectively exterminated.

Use-case Specific Classes and Relationships. For **R4**, we use classes and relationships relating to the use-case of garden planning. This allows individual plants within the garden to be connected with `neighbour`, `companionNeighbour` and `incompatibleNeighbour` relationships, that define the garden configurations. Specific garden configurations are described with the class `Garden`. Based on those properties, we introduce classes for plants that are well-placed or badly placed, and gardens that represent good configurations or bad configurations.

3.3. Data Integration

Detailed plant classifications, and complex axioms describing them, were extracted from a variety of multimodal sources, resulting in an ontology with 1405 classes and 5264 axioms, enabling us to derive (anti-)companionships between a variety of plants, as well as explanations.

Companion Planting Charts. For **R2**, we retrieved companions and anti-companions from an openly available planting chart[13]. The high-level relationships (`companionWith` and `anticompanionWith`) were directly captured in the ontology. For a further understanding of mechanisms of companion and anticompanion relationships, we extracted companionship mechanisms from literature [3]. Those more complex mechanisms are modeled using property chains over more fine-grained relationships (such as `pollinatedBy` or `eatenBy`). This allows to derive and explain the different types of companionship relationships (Figure 1) also through reasoning. Moreover, these axioms can be used to infer novel companionships.

Wikidata. We linked plants extracted from the companion planting chart to metadata found on Wikidata[14] (**R5**). From Wikidata, we extracted the taxon name, plant product, and common name. These can be used to adapt an interface to different user types. Some users might be interested in harvesting plant products (i.e. *tomatos*), while others might take a scientific perspective (taxon name *Solanum lycopersicum*).

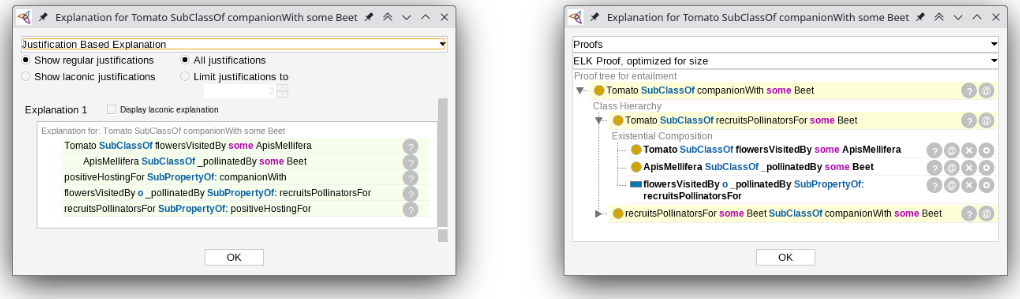
Globi. The Global Biotic Interactions (GloBI) database [10] integrates various existing open datasets describing interactions, such as *eats*, *visitedBy*, *parasiteOf*, between species that inhabit our planet, addressing **R3**. An example is: "Chelonia mydas *eats* Dictyota cervicornis". Such interaction data form a basis for certain companionship mechanisms, mainly submechanisms of `positiveHostingFor` and `supressesPestsFor`. We therefore extracted relationships between plants and pollinators, predators and pests using the Globi API,[15] and integrated them into our ontology by automatically turning them into class axioms such as:

```
Tomato subClassOf flowersVisitedBy some Apis Melifera .
```

These axioms served as the basis for the generation of property chains, such as:

```
visitedBy o eats o parasiteOf SubPropertyOf reqruitsPredatorsFor .
```

[13]https://www.permaculturenews.org/2010/07/30/companion-planting-guide/
[14]https://www.wikidata.org/wiki/Wikidata:Main_Page
[15]https://api.globalbioticinteractions.org/



(a) Justification-based explanation service (Protégé)

(b) Proof-based explanations (Evee)

Figure 2: Explaining companionships, for users familiar with Semantic Web technologies.

4. Use-Cases and Proof of Concept User Interface

CoPla can be used to find (anti-)companionships, analyse and suggest garden configurations, and provide explanations. To illustrate these functionalities, we implemented a simple web-front end (Figure 3). First, let us discuss the three use cases previously introduced.

4.1. Finding and Explaining (Anti-)Companions

If we are interested in a specific plant and its (anti-)companions, we can use a subclass query, e.g. in Protégé, using a class expression "`companion some PLANT`" (or "`anti_companion some PLANT`"), where `PLANT` identifies the corresponding plant. For each obtained (anti-)companionship, we can ask for *explanations* in form of justifications [24, 25], using the functionality of Protégé (Figure 2a). All relevant axioms are in the fragment of OWL-EL that is supported by the ELK reasoner[26], allowing us to use the advanced explanation services based on *proofs* ELK. This includes the protege-proof-explanation plugin [27], Evee [28] and Evonne [29], which gives a clearer explanation on why a companionship holds (see Figure 2b).

4.2. Analysing Garden Configurations

We can also analyse specific garden configurations by representing them as knowledge graphs. For this, we use an instance of the class `Garden` to which we link its plants. The property `neighbour` models which plants are placed next to each other. To determine whether a plant has a neighbour that is an (anti-)companion, CoPla contains axioms of the following form:

```
hasNeighbour some PLANT and companion some PLANT
                           SubClassOf companionNeighbour some PLANT

hasNeighbour some PLANT and anti_companion some PLANT
                           SubClassOf incompatibleNeighbour some PLANT
```

Different concepts in the ontology define desirable properties of plants based on their neighbourhood—e.g. having one or more `companionNeighbours`—and of the garden—having at least 3 well-placed plants. For instance, the class `BadGarden` is defined as a garden in which at least one plant is placed next to an anti-companion. Through reasoning on the Garden individual, users can now determine which properties their garden possesses—whether it is a bad garden or a particularly well-organised garden—and obtain detailed explanations—e.g. which neighbours are currently anti-companions, and why.

4.3. Suggesting Garden Configurations

A lesser known reasoning service called *ontology repair* can be used to not only analyse existing garden configurations, but also suggesting optimal placements of plants. The use case here is that the user specifies what plants they want to have in their garden. Through reasoning, we then determine which plants should be placed next to each other.

Given an OWL knowledge base $\mathcal{K}$ and some undesired entailment α, a *repair* of that knowledge base is a maximal subset of statements in $\mathcal{K}$ such that α is not entailed anymore, but the properties maximising companionship β is preserved. Given a list of plants a naive approach would be to start with a “maximal” knowledge graph in which instances of all of the plants are placed next to each other using the `neighbour` property, and then use repair to remove those neighbourhood relations that cause the garden to be a `BadGarden`. The resulting knowledge graph is then a configuration in which no anti-companions are placed next to each other.

This approach has two limitations: 1) the repair might remove statements other than the neighbour-relation, for instance elements from the TBox such as axioms from the plant ontology, and 2) we not only want to eliminate anti-companionships, but also maximise companionships. We thus use a different type of repairs defined as follows:

Definition 1. *Let $\mathcal{K}$, $\mathcal{K}_f$ be knowledge bases s.t. $\mathcal{K}_f \subseteq \mathcal{K}$ and α, β axioms s.t. $\mathcal{K} \models \beta$ and $\mathcal{K}_f \not\models \alpha$. A* repair of $\mathcal{K} \models \alpha$ preserving $\mathcal{K} \models \beta$, *with* fixed $\mathcal{K}_f$, *is a knowledge base $\mathcal{K}_R$ s.t.*

1. $\mathcal{K}_f \subseteq \mathcal{K}_R \subseteq \mathcal{K}$, 2. $\mathcal{K}_R \not\models \alpha$, and 3. $\mathcal{K}_R \models \beta$.

Clearly, such a repair need not always exist. If it does, it can be computed similar to classical repairs by using justifications [30]. In order to suggest a garden configuration for a given set of plants, we construct a maximal garden configuration as described above, add all statements except for the neighbourhood relations into the fixed component $\mathcal{K}_f$, and repair the entailment

$$\mathcal{K} \models \texttt{garden Type BadGarden} \quad \textit{preserving} \quad \mathcal{K} \models \texttt{garden Type NiceGarden},$$

where `BadGarden` is the Garden class we aim to avoid (e.g. one with neighbouring anti-companions) and `NiceGarden` is some Garden class that we would like to preserve (e.g. one that has many companionship neighbours). In our implementation, we rank different such preferable Garden-classes, and start with the highest ranked class when trying to compute the repair, and then step-wise change to lower-ranked classes until a repair is found, this way computing the best garden configuration that avoid anti-companionships.

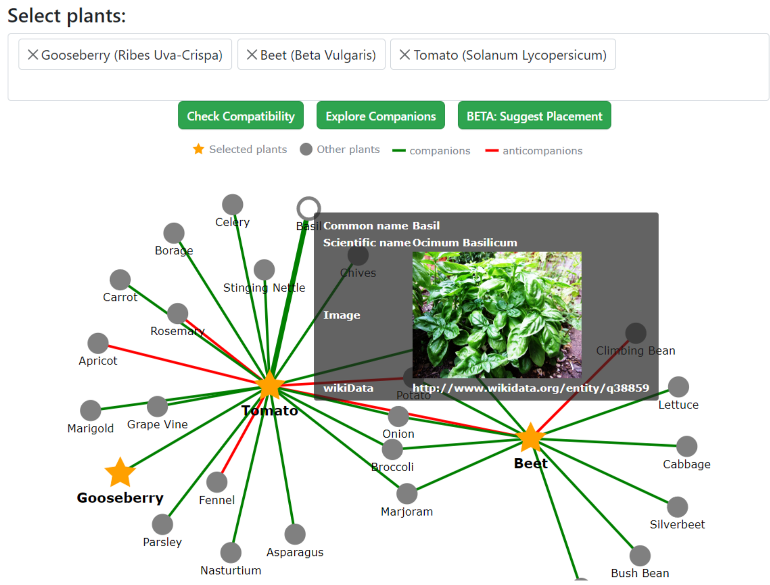


Figure 3: Our prototype used to explore companions and anti-companions of selected plants.

4.4. Our User Interface

Using Protégé for the described use cases requires familiarity with Semantic Web technologies. Therefore, we implemented some of these use cases with a proof-of-concept user interface using JavaScript and Spring Boot [31], using the REST API for communication with the backend. The backend was implemented in Java 17, using the OWL API 5 [32] (5.1.9) for all OWL related services (including explanation services). The tool can be deployed using Docker Compose [33]. All code, including docker files, are available in our repository (see introduction), which also includes instructions on how to run our tool.

The frontend uses Javascript and Node.js. We made use of AnyChart[16] to display simple but interactive graphs to the user. The user interface provides three distinct functionalities: (1) exploring companions of selected plants (2) Checking garden configurations (based on a list of plants) and (3) Suggesting Placement of Plants. We describe the first functionality—the others are described in the appendix.

Figure 3 shows the output of the companion exploration after the user presses the "Explore Companions" button. We use a simple network graph with coloured edges. Green signals a companion relationship and red is for an anti-companion relationship. The presented graph is interactive and the user can click on any nodes or edges, as well as drag the nodes around in the graph canvas. The tooltips of the graph provide further information about the plants: the scientific name, a wikidata link to the plant, and an image of the plant retrieved from wikiData. In the future, we plan to additionally implement an explanation functionality when clicking on edges, to explain in more detail why the connected plants are (anti-) companions.

[16]https://www.anychart.com/

5. Limitations and Future Work

Provenance in the Ontology. The source knowledge on which we based our design may have limitations: A lot of the knowledge regarding companion planting is anecdotal, and the resources describing these relationships are not always supported by scientific claims. Future work could address this issue by indicating and tracking the provenance of the sources for companionship interactions, both to spot contradicting information and to highlight the relationships that are verified by scientific evidence. Additional future work will aim at including additional factors in the ontology, such as environmental factors and soil quality.

Explanations in User interface. We want to integrate further explanation services into our prototype. In the current version, we only show explanations for the analysis of the garden configuration. Those are just displayed in OWL syntax but but not visualised further. In the future, we would also like to provide explanations for companionships. We also plan to integrate the interactive proof visualisations of Evonne [29], which is also based on Javascript, to allow for a more user friendly and interactive explanations.

Further Improving the User Interface. As the name suggests, we engineered a proof of concept, which means that our tool is neither finished nor did we perform any evaluation with users. For future work, we plan to elicit requirements for the user interface with domain experts and hobby gardeners who are unfamiliar with ontologies to go the extra mile and make this tool accessible. This also includes an evaluation of this interface to show its usefulness. Only by making it usable by laymen can we fully overcome the challenge that companion planting is currently facing in terms of heterogeneous and often inaccessible information.

Evaluation with Human Users. We plan to conduct evaluations of the tool with human users in real-world use cases. We will solicit feedback from domain experts and hobby gardeners who are unfamiliar with ontologies to refine our user interface. This user-centered design approach aims to ensure that our tool is accessible and practical, even for those without technical expertise in ontologies. The evaluation will specifically aim to assess the tool's usability and effectiveness in supporting sustainable gardening practices.

6. Conclusions

Companion planting is a key practice in sustainable agriculture and a way for laymen to contribute with their own garden. However, the information on companion planting is scattered and often inaccessible, which calls for integration and publication in a more digestible format. Our contributions to this challenge are three-fold: (1) we engineered the semantically rich companion planting ontology, integrating information from various sources on plants and their effects on each other; (2) we leveraged various reasoning paradigms to infer additional information as well as to check plant constellations against certain garden criteria; (3) we build a simple yet useful decision support system which makes use of the ontology and reasoning abilities for users unfamiliar with Semantic Web technologies. Through our contributions, we showcase the usefulness of these technologies outside of research bounds and also bring them closer to everyday users.

References

[1] S. Hirschfeld, R. V. Acker, Permaculture farmers consistently cultivate perennials, crop diversity, landscape heterogeneity and nature conservation, Renewable Agriculture and Food Systems 35 (2020) 342–351. doi:10.1017/S1742170519000012.

[2] M. D. Hathaway, Agroecology and permaculture: addressing key ecological problems by rethinking and redesigning agricultural systems, Journal of Environmental Studies and Sciences 6 (2016) 239–250. doi:10.1007/s13412-015-0254-8.

[3] P. P. Reddy, P. P. Reddy, Companion planting, Agro-ecological Approaches to Pest Management for Sustainable Agriculture (2017) 149–164.

[4] J. Griffiths-Lee, E. Nicholls, D. Goulson, Companion planting to attract pollinators increases the yield and quality of strawberry fruit in gardens and allotments, Ecological Entomology 45 (2020) 1025–1034. doi:10.1111/een.12880.

[5] A. Konkolewska, A. Piechalak, L. Ciszewska, N. Antos-Krzemińska, T. Skrzypczak, A. Hanć, K. Sitko, E. Małkowski, D. Barałkiewicz, A. Małecka, Combined use of companion planting and pgpr for the assisted phytoextraction of trace metals (zn, pb, cd), Environmental Science and Pollution Research 27 (2020) 13809–13825. doi:10.1007/s11356-020-07885-3.

[6] R. Ben-Issa, L. Gomez, H. Gautier, Companion plants for aphid pest management, Insects 8 (2017) 112. doi:10.3390/insects8040112.

[7] J. Gao, F. Zhang, Influence of companion planting on microbial compositions and their symbiotic network in pepper continuous cropping soil, Journal of Microbiology and Biotechnology 33 (2023) 760–770. doi:10.4014/jmb.2211.11032.

[8] C.-l. Chang, X.-p. Fu, X.-g. Zhou, M.-y. Guo, F.-z. Wu, Effects of seven different companion plants on cucumber productivity, soil chemical characteristics and pseudomonas community, Journal of Integrative Agriculture 16 (2017) 2206–2214. doi:https://doi.org/10.1016/S2095-3119(17)61698-8.

[9] G. Franck, Companion planting: successful gardening the organic way, Thorsons Publishing Group, 1983.

[10] J. H. Poelen, J. D. Simons, C. J. Mungall, Global biotic interactions: An open infrastructure to share and analyze species-interaction datasets, Ecological Informatics 24 (2014) 148–159.

[11] J. Kattge, G. Bönisch, S. Díaz, S. Lavorel, I. C. Prentice, P. Leadley, S. Tautenhahn, G. D. Werner, T. Aakala, M. Abedi, et al., Try plant trait database–enhanced coverage and open access, Global change biology 26 (2020) 119–188.

[12] C. Jonquet, A. Toulet, E. Arnaud, S. Aubin, E. D. Yeumo, V. Emonet, J. Graybeal, M.-A. Laporte, M. A. Musen, V. Pesce, et al., Agroportal: A vocabulary and ontology repository for agronomy, Computers and Electronics in Agriculture 144 (2018) 126–143.

[13] P. Larmande, K. Todorov, Agrold: A knowledge graph for the plant sciences, in: proceedings of ISWC, Springer, 2021, pp. 496–510.

[14] A. Goldstein, L. Fink, G. Ravid, A framework for evaluating agricultural ontologies, Sustainability 13 (2021) 6387.

[15] I. Subirats-Coll, K. Kolshus, A. Turbati, A. Stellato, E. Mietzsch, D. Martini, M. Zeng, Agrovoc: The linked data concept hub for food and agriculture, Computers and Electronics in Agriculture 196 (2022).

[16] J. S. Madin, S. Bowers, M. P. Schildhauer, M. B. Jones, Advancing ecological research with

ontologies, Trends in ecology & evolution 23 (2008) 159–168.
[17] L. Penev, M. Dimitrova, V. Senderov, G. Zhelezov, T. Georgiev, P. Stoev, K. Simov, Openbiodiv: a knowledge graph for literature-extracted linked open data in biodiversity science, Publications 7 (2019) 38.
[18] S. Babalou, E. Kleinsteuber, B. El Haouni, F. Zander, D. S. Costa, J. Kattge, B. König-Ries, iKNOW-a knowledge graph management platform for the biodiversity domain, in: proceedings of ISWC 2022, Springer, 2022.
[19] N. Le Guillarme, W. Thuiller, A practical approach to constructing a knowledge graph for soil ecological research, European Journal of Soil Biology 117 (2023) 103497. doi:https://doi.org/10.1016/j.ejsobi.2023.103497.
[20] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: From ontological art towards ontological engineering, in: Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series, AAAI, 1997. URL: https://oa.upm.es/5484/.
[21] H. Philbrick, R. Gregg, Companion Plants and How to Use Them, Devin-Adair Publishers, 1966.
[22] A. Greer, Companion Planting for the Kitchen Gardener, Skyhorse Publishing, 2014.
[23] M. Jeffreys, A Beginners Guide to Companion Planting: Companion Gardening with Flowers, Herbs & Vegetables, 2013.
[24] S. Schlobach, R. Cornet, et al., Non-standard reasoning services for the debugging of description logic terminologies, in: Ijcai, volume 3, 2003, pp. 355–362.
[25] M. Horridge, Justification based explanation in ontologies, Ph.D. thesis, University of Manchester UK, 2011.
[26] Y. Kazakov, M. Krötzsch, F. Simancik, The incredible ELK — from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies, J. Autom. Reason. 53 (2014) 1–61. doi:10.1007/s10817-013-9296-3.
[27] Y. Kazakov, P. Klinov, A. Stupnikov, Towards reusable explanation services in Protégé, in: Description Logic Workshop, volume 1879 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017.
[28] C. Alrabbaa, S. Borgwardt, T. Friese, P. Koopmann, J. Méndez, A. Popovič, On the eve of true explainability for owl ontologies: Description logic proofs with evee and evonne, Proc. DL 22 (2022).
[29] J. Méndez, C. Alrabbaa, P. Koopmann, R. Langner, F. Baader, R. Dachselt, Evonne: A visual tool for explaining reasoning with OWL ontologies and supporting interactive debugging, Computer Graphics Forum (2023). doi:https://doi.org/10.1111/cgf.14730.
[30] B. Parsia, E. Sirin, A. Kalyanpur, Debugging OWL ontologies, in: A. Ellis, T. Hagino (Eds.), WWW-05, ACM, 2005, pp. 633–640.
[31] C. Walls, Spring Boot in action, Simon and Schuster, 2015.
[32] M. Horridge, S. Bechhofer, The OWL API: A Java API for OWL ontologies, Semantic web 2 (2011) 11–21.
[33] D. Merkel, Docker: lightweight linux containers for consistent development and deployment, Linux Journal 2014 (2014) 2.

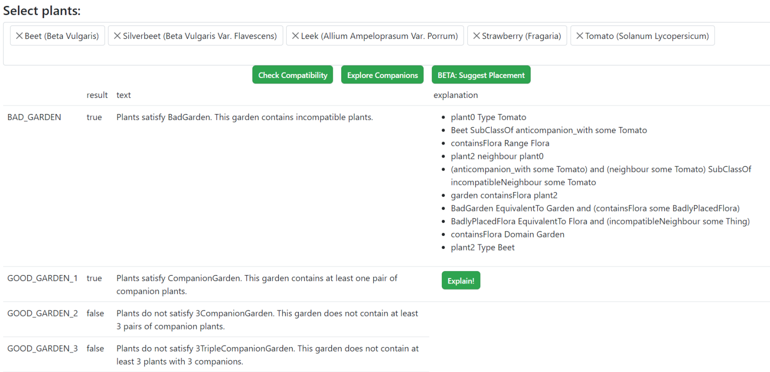


Figure 4: Check the compatibility of selected plants, with an option of retrieving explanations for satisfied garden properties.

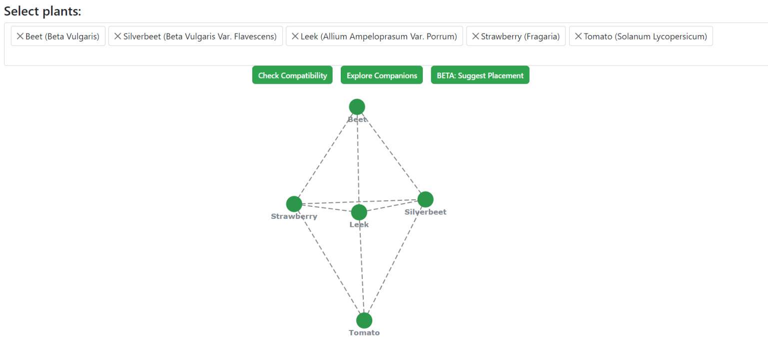


Figure 5: Suggestion of plant placement of the selected plants. This is especially intersting if anti-companions are present in the set of plants.

A. Appendix

Figure 4 shows the output of the compatibility checking function, by listing the different garden properties, their result and a short description of what the result means. If a property evaluates to 'true', then an explain button is displayed that the user can click to retrieve an explanation for this property. By only showing a button when an explanation can be generated, we prevent the user for trying to retrieve an explanation where none can be given. Once the button is clicked, the explanation will be inserted instead of the button, once received from the API.

Figure 5 displays the last functionality, the suggested placement of plants. It is visualised in a simple but interactive network graph. A connection between plants from the selected plant list is shown if these plants can be placed next to each other in the garden. These plants do not necessarily have to be companions. However, if two plants are anti-companions, no connection is drawn, suggesting that these plants are placed far apart from each other to optimize the placement of the plants in terms of positive interactions.