ECOLOPES

ECOlogical building enveLOPES: a game-changing design approach for regenerative urban ecosystems

H2020-FET-OPEN-2021-2025

Action number 964414

D4.2

Interim EIM Ontology

Dissemination level:	Public
Contractual date of delivery:	Month 30, 30 September 2023
Actual date of delivery:	27 September 2023
Work package:	WP 4
Task:	T4.7
Type:	Report
Approval Status:	Submitted
Version:	v0.1
Number of pages:	71
Filename:	D4.2_Ecolopes_InterimEIMOntology_20230927_v0.1.docx

Abstract

This deliverable outlines advances made in WP4 since D4.1. WP4 and WP5 worked together to develop the EIM ontologies as a key component of the ontology-aided generative computational design process (PART A in this report). WP4 also focused on the development of the ecological model (PART B in this report). PART A comprises development of the EIM Ontologies (1) for preparing datasets for the design generation process via a knowledge graph that is queried by the designer; (2) to aid generating design output for spatial organisation; (3) to aid design output for geometric articulation of selected spatial arrangements; and (4) to facilitate interactions between the components of the ontology-aided generative computational design process, and to integrate the ontology into the ECOLOPES computational design workflow. PART B entails development of (1) a plant community model, (2) an animal home range model, (3) a soil development model, and (4) their integration into the ECOLOPES computational design workflow.

co-funded by the European Union

HISTORY

Version	Date	Reason	Revised by
v0.1	02.08.2023	Draft Internal Review	Jens Joschinski
	02.08.2023	Draft Internal Review	Verena Vogler
	08.08.2023	Draft Internal Review	Surayyn Uthaya Selvan
	23.08.2023	Draft Internal Review	Shany Barath
	03.09.2023	Draft Internal Review	Yasha Grobman

AUTHOR LIST

Organization	Name	Contact Information
TUM	Defne Sunguroğlu Hensel	defne.hensel@tum.de
TU Wien	Albin Ahmeti	aljbin.ahmeti@tuwien.ac.at
TU Wien	Michael Hensel	michael.hensel@tuwien.ac.at
TU Wien	Akif Mehmet Cifci	akif.cifci@tuwien.ac.at
TU Wien	Tina Selami	tina.selami@tuwien.ac.at
TU Wien	Jakub Tyc	jakub.tyc@tuwien.ac.at
TUM	Wolfgang Weisser	wolfgang.weisser@tum.de
TUM/SAAD	Jens Joschinski	jens.joschinski@tum.de jens.joschinski@animal-aided-design.de
UNIGE	Mariasole Calbi	mariasole.calbi@edu.unige.it

EXECUTIVE SUMMARY

The EIM Ontologies and Ecological Model are key components of the ECOLOPES Computational Platform. The successful integration of the latter has been described in detail in D3.3 (M29). The ECOLOPES Computational Platform comprises among integrated technical components also the EIM Ontologies, the ECOLOPES Voxel Model (D5.2, D5.3), and the ECOLOPES Computational Model (D5.4). Together these three deliverables detail the current state of these platform components. In this report we indicate how these components are interlinked, and how they will feed into the ECOLOPES computational system.

In PART A of this deliverable, we describe the development processes with respect to the EIM ontologies. As one of the key components of the ECOLOPES Computational platform, and more specifically of high relevance for the design generation environment that we term "*ontology-aided generative computational design process",* the EIM ontologies are three separate ontologies that fulfil the following goals: EIM Ontology 1 supports the *translational* process in which requirements elaborated in the design brief for a given project and site and additional requirements are analysed, correlated, spatialised and prepared for design generation. This involves the preparation of the datasets that serve as inputs into the design generation process via a knowledge graph that can be queried by the designer and connects with correlations computed in the ML empowered KB. EIM Ontology 2 uses this as an input and aids the generation of variants of spatial organisation (dataset *volumes*). EIM Ontology 3 aids the generation of variants of the specific geometric articulation for selected variants of spatial organisation (dataset *landform*). Part A is organised into seven sections: In section 1, we introduce the ECOLOPES EIM Ontology, and an overview of the content covered in this report. In section 2, we describe the role of the EIM Ontologies in the ECOLOPES Computational Platform and elaborate their conceptual and technical characteristics. This section describes the EIM Ontology in this pipeline and identifies a data-to-design gap that is bridged by the ontology and ontology-aided generative design process. In section 3, we describe the role of the EIM Ontologies in the design generation environment that we term "ontology-aided generative computational design process". The conceptual and technical characteristics of the involved components are elaborated, and we explain how the EIM Ontologies are developed according to their specific tasks. In section 4 of this report, we detail the interfaces, interactions, and interoperability between the three components of the ontology-aided generative computational design process (EIM Ontologies, Rule-Based Systems, and CAD Models). Furthermore, this section describes how this is organised in three process loops. In section 5, we elaborate the ontological output for subsequent computational processes. This includes a description of the different types of data and information (including the input from the Ecological Model and the Knowledge Generation Framework), the uses of the ontology for the inference of decision rules and how these are implemented by algorithms in the generation of the three datasets *networks*, *volumes*, and *landform* in CAD. Finally, this section describes the output of the ontology-aided generative computational design process in CAD for design optimization (WP6). In section 6, we outline the intended future development of the EIM Ontologies, and approaches and methods for validation. The current state and planned further development of the EIM Ontologies within and beyond the project's timeline is addressed. Finally, we provide an overview of the submitted and planned publications on the EIM Ontologies in section 7.

In PART B of this deliverable, we describe the development of the Ecological Model (initially reported in *D4.1 Preliminary EIM Ontology*). This part is intentionally kept short, as the successful integration of the model into the ECOLOPES computational workflow has already been described in D3.3. The advances from M13 to M24 have been summarised in D1.5 (Report of Year 2), so we here provide further details on the modelling and data parametrization efforts of M25 to M29.

ABBREVIATIONS AND ACRONYMS

ASP	Answer set programming
AEC	Architecture, Engineering and Construction
CAD	Computer-Aided Design
CQs	Competency Questions
DFP	Dataflow Programming
EA	Evolutionary Algorithm
ENs	Ecological Networks
EIM	ECOLOPES Information Model
GA	Genetic Algorithm
GDB	Graph Database
GH	Grasshopper
JSON	JavaScript Object Notation
KG	Knowledge Graph
KGF	Knowledge Generation Framework
KPI	Key Performance Indicator
LP	Logic Programming
ML	Machine Learning
MR	Machine Reasoning
Ns	Networks
RDFS	RDF Schema
OWL	Web Ontology Language
RDB	Relational Database
RDF	Resource Description Framework
SQL	Structured Query Language
OBDA	Ontology-based Data Access
UNs	User Networks

WP	Work-package

INTERFACE ABBREVIATIONS (SEE FIG. 2)

DB	Database
V	Voxel
EM	Ecological Model
O	Ontology
CAD	Computer-Aided Design
A	Algorithm
D	Designer
KGF	Knowledge Generation Framework

TABLE OF CONTENTS

GENERAL INTRODUCTION

The ECOLOPES computational framework, its technical components, and data flow between the latter (computational workflow) was elaborated in WP3 and the updated version presented in D3.3. (M29) (Fig. 1).

The ECOLOPES Computational Framework facilitates informed multi species design for ecological building envelopes, that we term *ecolopes* (Fig. 1) (D3.3, Weisser et al. 2022). It includes technical components such as the Ecological Model, the Knowledge Base, the design generation environment, which we term "*ontology-aided generative computational design process*", the Optimization environment (D6.1), and components for validation. The Ecological Model, developed in WP4 (D4.1, D1.5), simulates plant, animal, and soil dynamics. The Ecological Model was integrated in a 3D CAD system (Rhino/ Grasshopper) (D3.3 Chapter 3), which facilitated the generation of relational data (architecture, environmental, and ecology) for building envelopes in a resolution of 1 cubic metre. In the next step, this data was stored in the Knowledge Base (D3.3. Chapter 4). The KB was then analysed using a ML model which extracts rules for decision making for WP5 (D3.3, Chapter 4). The design generation environment (ontology-aided generative computational design process), which is developed in WP5 (*D5.2 ECOLOPES Voxel Mode*l and *D5.3 ECOLOPES Computational Model*) facilitates design generation and the generation of design search space populated with alternative solutions that can be analysed, evaluated, and ranked. The optimization environment, which is developed in WP6 aims to facilitate optimization based on the search space produced by the ontology-aided generative computational design process (WP5) and selection of the final *ecolope* design solution based on KPIs (D6.1). The ECOLOPES Computational Model provides input for optimization, the output of which provides the basis for the overall validation (WP7) of the ECOLOPES Computational Framework.

During development it became clear that the Ecological Model output is far too complex for integration into the design generation algorithms, and that the development of a comprehensive EIM ontology takes time. It was therefore decided to pursue parallel workflows: One workflow focuses on the development of the ontology-aided generative computational design process and the EIM Ontologies (TU Vienna), while workflow uses the Knowledge Generation Framework (KGF, D3.3) to provide correlational (ecology-architecture) information for design decision support (MCNEEL, TUM, SAAD). The KB is the joint interface of the two workflows.

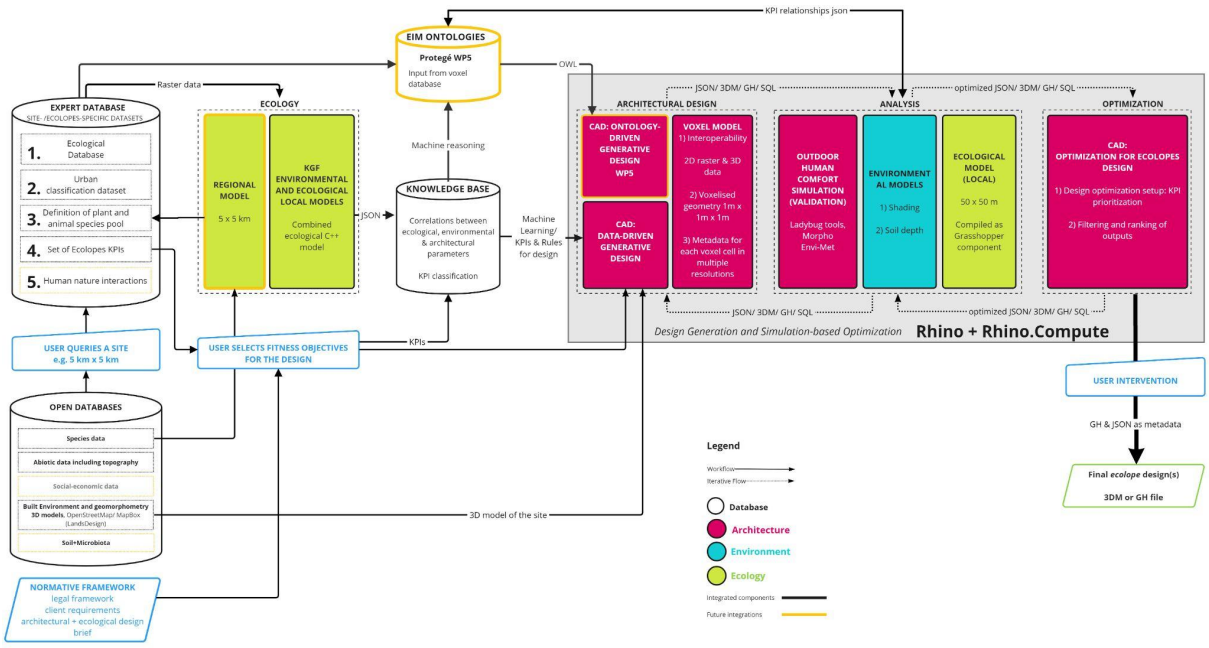

Fig. 1: Ecolopes computational framework showing integrated (black frame) and non-integrated technical components (yellow frame) (D3.3).

PART A - EIM Ontology

1. INTRODUCTION

1.1 Introduction to PART A

Decision support is required to utilise data for solving multi-domain design problems, especially if the aim is to exceed post-optimization in the later stage design phases as is typical for instance for engineering knowledge. This requirement constitutes a data-to-design gap in this project, which concerns the transition from ecological data to architectural design. Computational ontologies have been introduced in the urban environment domains in diverse ways (Pruski & Sunguroğlu Hensel, 2022). However, to our knowledge no ontology and rule-based reasoning has been developed in the domain of generative design that brings ecological requirements into architectural decision-making in the early design stages. Therefore, the EIM Ontologies and the ontology-aided generative computational design process (Fig. 2) is developed as a computational system for decision support to bridge this data-to-design-gap. In this project this entails fusion of heterogeneous ecological, environmental, and architectural data, voxel-based data structuring, context-information at regional/urban and local/architectural scales, and modelling information according to decision needs. Furthermore, this can benefit from a graph-based representation and integration of different domain knowledge, as well as an ontology for machine reasoning to derive design decision rules that can be implemented by algorithms in the CAD environment to generate design solutions. Note that ecological analysis is complex and that in this deliverable we show a way in which ecological knowledge can be integrated, illustrated by special cases. Including all

ecological information needed for holistic multi-species design is a task that will need to be done over the next years.

This section introduces the ECOLOPES Information Model (EIM Ontologies) in the context of the ECOLOPES computational framework in general (Fig. 1), as well as in the context of the ontology-aided generative computational design process (Table 1, Fig. 2). The EIM Ontology is modularized consisting of three linked ontologies (Table 1) that interface with CAD and aid the design of ecological building envelopes. In so doing we address a critical "data-to-design gap" (Sunguroğlu Hensel, 2017; Sunguroğlu Hensel et al, 2022) in computational architecture and multi-species architectural design.

The EIM Ontologies and the ontology-aided generative computational design process combine Logic Programming (LP) (here ASP) for rules and constraints, Dataflow Programming (DFP) (Grasshopper), machine reasoning (ontology-based reasoning, OBR), i.e., TBox only (subsumption of classes) or TBox + ABox consistency checks (Protégé reasoner, e.g., HermiT), SPARQL-endpoint reasoning (RDFS or OWL profiles in GraphDB) and Machine Learning (ML). The EIM Ontologies are located between the Knowledge Base and ECOLOPES Computational Model. An interim KB currently contains the KGF results, a structured analysis of ecological models run on multiple input geometries, as well as architectural input and regional data.

Table 1: Overview of the three key stages and involved components of the generative computational design process, including purpose of each stage, as well as involved datasets, inputs, outputs, involved computational components and degree of designer involvement in each stage.

Ontology-aided generative comp. design process	**Purpose**	**Datasets**	**Inputs**	**Outputs**	**Involved Comp. Components**	**Designer involvement**
Loop 1 Translational Process	Translation of design brief and designer defined requirements into inputs for the generative process	Datasets maps and networks, RDFs, (Open) Knowledge Graphs	Design Brief, Designer Inputs, etc.	Datasets maps and networks in CAD environment	EIM Ontology 1, Voxel Model, Ecological Model, CAD 1 algorithms: GraphDB querying and reasoning	high
Loop 2 Generative Process 1	Computational generation of spatial organisation	Volumes	Constraints, Maps, Networks, etc	Volume distribution in CAD environment Voxel data Ontological output	EIM Ontology 2, Volume distribution in CAD environment Voxel data Ontological output ASP	variable
Loop 3 Generative Process 2	Computational generation of geometric articulation	Landform	Constraints, Maps, Networks, Volumes, etc	Site and building geometry in CAD environment, Voxel data Ontological output	EIM Ontology 3, Site and building geometry in CAD environment, Voxel data Ontological output, KGF, ASP	variable

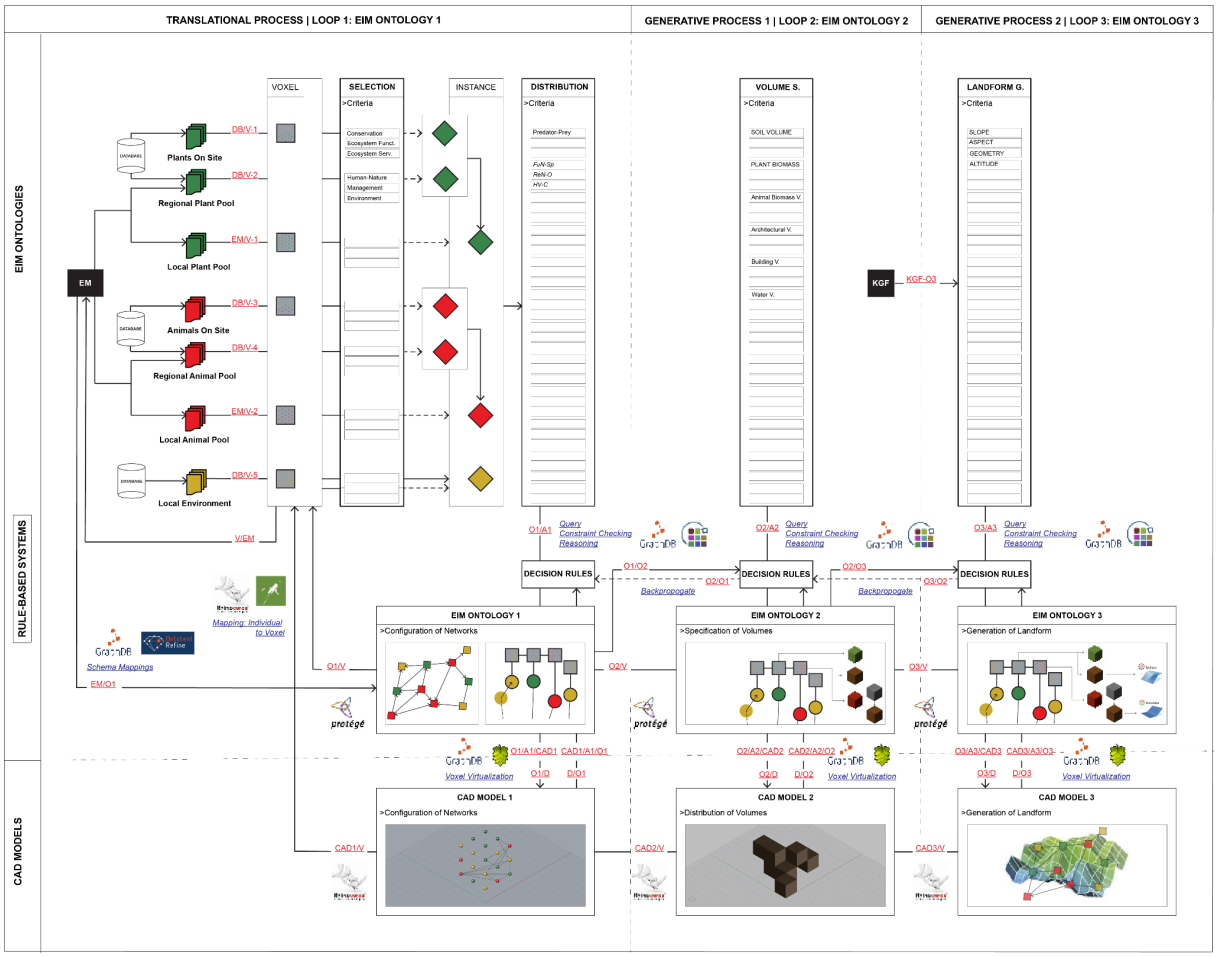

Fig. 2: Ontology-aided generative design process for ECOLOPES showing the system architecture with its main components and interfaces. This includes the three ontologies (EIM Ontology (1, 2, 3)) and process loops (Loop (1, 2, 3)) with the main interfaces (in red) and allocated methods (in blue) mapped along the three design stages (Translational Process, Generative Process 1, Generative Process 2) (x-axis); and the main components (EIM Ontologies, Rule-Based System, CAD models) of each loop (y-axis). The individual loops are described in Fig. 4 to 6 below.

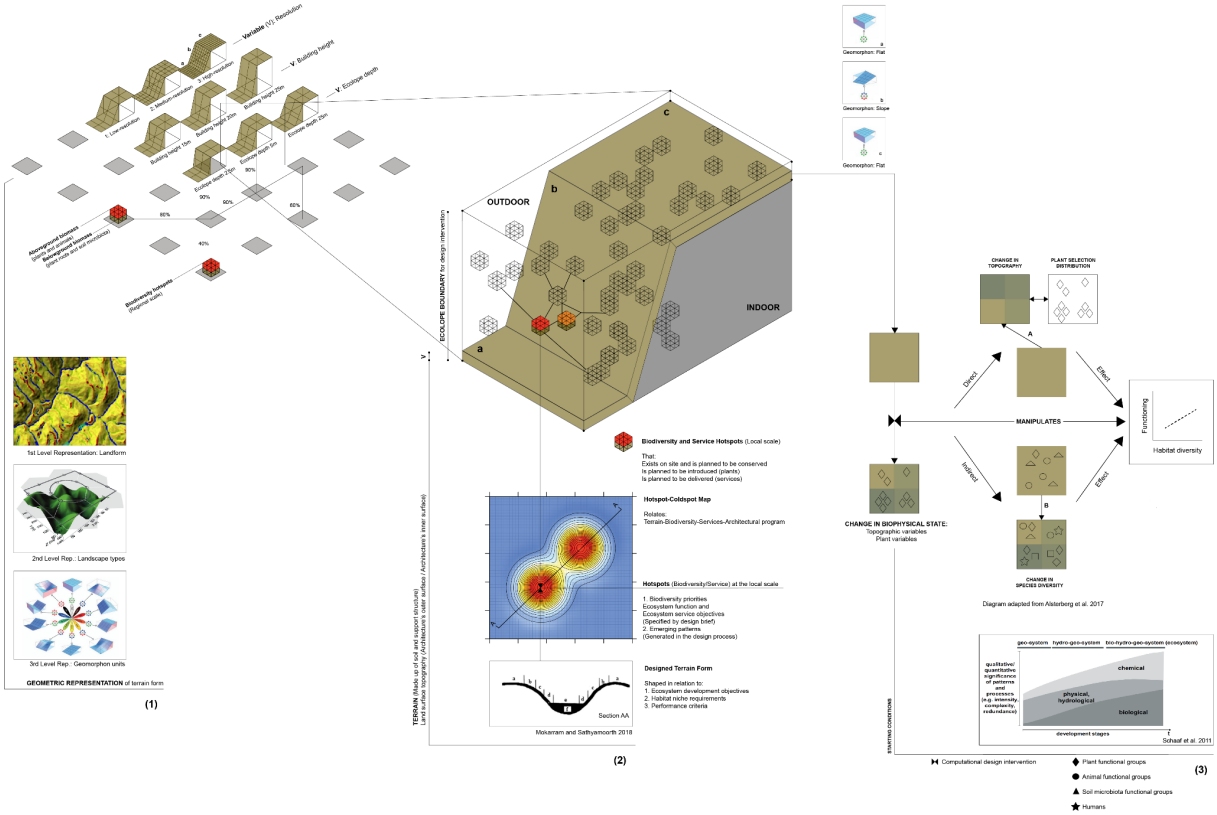

Fig. 3: Examples for ontology-aided generative design inputs and outputs and links to regional and local scales (D4.1 Preliminary EIM Ontology). This figure shows: 1) the larger, regional context of an ecolope, represented as a unified terrain (bottom) and a network of the existing ecosystem (top); b) a representation of a potential initial design state of an ecolope at the local scale, which could be used as an input for initialising the process; and c) a representation of the central design questions how an initial situation could be transformed through manipulations in terrain, soil, plants and management variables to achieve certain objectives for the different stakeholders human, plants, animals and microbes.

Two processes make up the ontology-aided generative computational design process: (1) the *translational* process (Loop 1) and (2) the *generative* process (Loop 2, Loop 3) (*D5.1 Development Process for ECOLOPES Algorithms*).

The EIM Ontologies aid the generative computational design process in Loop 1, Loop2 and Loop 3. Figure 2 shows the loops organised vertically. Each loop corresponds with a specific task and features specific interfaces between computational components and their interactions, including feedback, thereby enabling vertical and lateral information flow. Each runs sequentially from left to right in the workflow and produces CAD Model output derived from algorithmic processes aided by the EIM Ontologies, related data, query results and design rules which are implemented in a sequential way (Fig. 2). This process requires interfaces with the ECOLOPES Voxel Model (*D5.2 ECOLOPES Voxel Model*) and ECOLOPES Computational Model (*D5.3 ECOLOPES Computational Model*).

In the *translational* process requirements elaborated in the design brief for a given project and site and additional requirements are analysed, correlated, spatialised and prepared for design generation. In this process, the requirements given by the design brief for a given project and site, and additional requirements are analysed, correlated, spatialised, and prepared. This entails different datasets and designer input in the form of User Networks for initialising Loop 1. This involves the preparation of datasets referred to as *maps* and *networks* (*D 5.1 Development Process for ECOLOPES Algorithms*). EIM ontologies are defined by their specific roles in the generative design process. In support of the *translational step*, we are building EIM Ontology 1 to aid the configuration of *Networks* (Ns) in the 3D CAD model as to guide volume specification and landform generation in the next design steps. EIM Ontology 1 will be used to represent and reason over Knowledge Graphs (KG) of *Ecological Networks* (existing-regional/local ENs and potential-designed/planned ENs) and *User Networks* (UN) (designer-configured UNs based on i.e., design brief, intentions, constraints, and building and planning regulations), and facilitate the synthesis of ENs and UNs in the configuration of *Networks* (Ns) in the 3D CAD environment. A network–in the general form– can be thought of as a graph that consists of nodes and edges. This makes the mapping and translation to the KGs straightforward given that they have the same data model in common. This requires an approach to ontology building that adapts, as far as possible, the semantics, classification schemes, and data schemas that are already formalised and used in the relevant domains (in AEC - BIM and Smart City models for instance i.e., BOT (Rasmussen et al., 2020) , BRICK (Balaji et al, 2016); in ecology, i.e., ENVO and EN Ontology; in geography, i.e., LFRO (The Landform Reference Ontology).

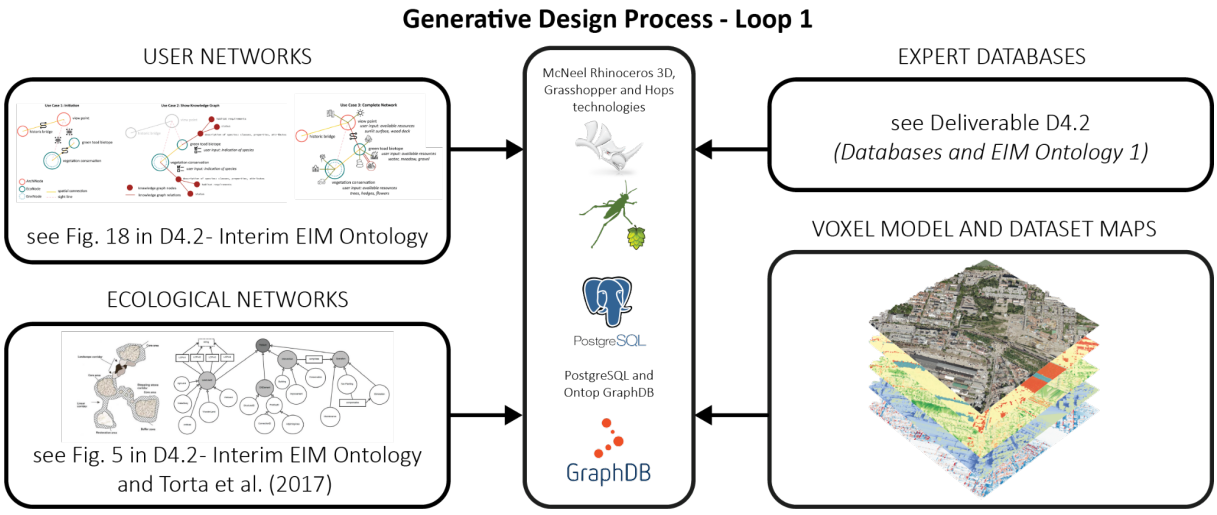

Fig. 4: Loop 1 in the generative design process enables combination of Dataset Networks, Dataset Maps, and data contained in the expert databases into 3D CAD based representation. Designer provides input by configuring User Networks in the McNeel Rhinoceros 3D interface. Dataset Networks consists of User Networks and Ecological Networks interactively queried from GraphDB, while the Dataset Maps is created by interactively querying the ECOLOPES Voxel Model.

In the *generative* process variants of spatial organisation and geometric articulation for the different design outputs are generated. This entails numerous design outputs that can be evaluated and ranked. Loop 2 facilitates generating spatial organisation via the distribution of

architectural, biomass, and soil volumes in a voxelized 3D space, materialised in the Rhinoceros CAD environment. This involves EIM Ontology 2, ASP, and CAD Model 2.

For the purpose of spatial organisation, we initially conceptualised three types of volumes: (1) architectural volumes, (2) plant biomass volumes, and (3) soil volumes. Note that in this step plant biomass is not the result of running the ecological model on an architectural form with a certain amount of soil (as in the Knowledge Generation Framework where plant biomass is the result of the other two variables), but here it is set directly as an objective.

Together these three types of volumes constitute the spatial organisation of an *ecolope* and depending on the design case also of the entire plot or extended site (*D5.1 Development Process for ECOLOPES Algorithm*). During further development steps it transpired that in future more diverse types of volumes may be required. Currently we consider distinguishing between different types of green volumes, e.g., dense biomass and sparse biomass (for instance as corridors for movement of animal species). Furthermore, it is useful to distinguish in future steps different types of architectural volumes, e.g., fully enclosed space and transitional space, and to assign further attributes such as including openings in the surfaces or not. Similarly, more elaborate ecological objectives can be included, also with respect to animals and microbes.

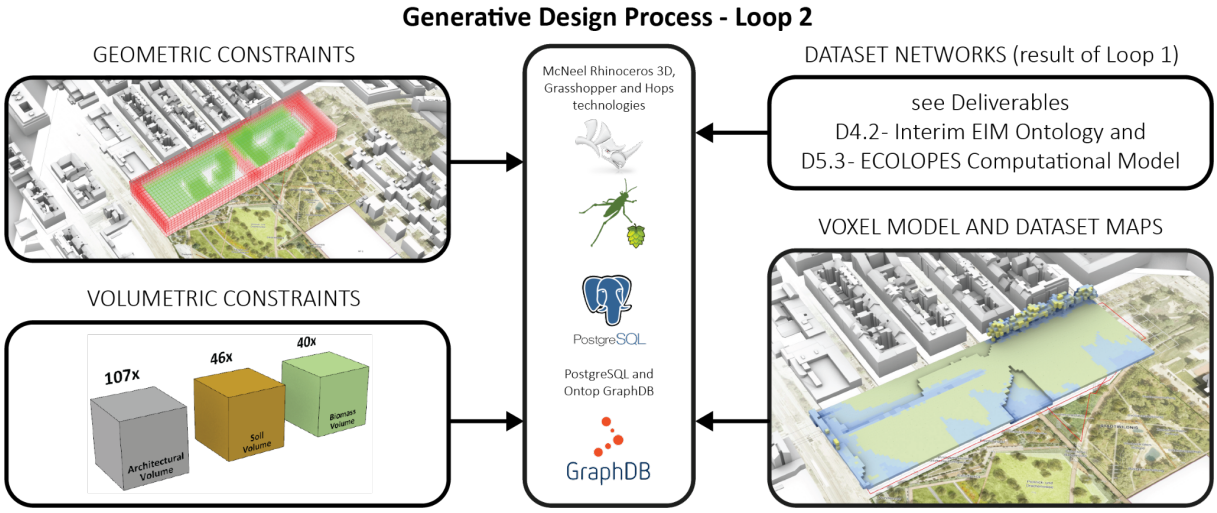

Fig. 5: In Loop 2 the generative design process is initiated to enable spatial organisation by way of distributing architectural, biomass and soil volumes. The inputs for Loop 2 are geometric constraints (e.g., project-specific site borders), volumetric constraints (e.g. as specified by site-specific planning regulations), dataset maps (project-specific datasets contained in the ECOLOPES Voxel Model), and the designer-defined dataset networks.

Loop 3 facilitates the generation of geometric articulation of selected spatial arrangements derived in Loop 2 in a voxelized 3D space, materialised in the Rhinoceros 3D CAD environment. It involves EIM Ontology 3, ASP, and CAD Model 3. For the purpose of initial geometric articulation we developed the notion of *urban landform* (*D5.1 Development Process for ECOLOPES Algorithm*), which is conceptualised to consists of specific terrain features that serve the purpose to instrumentalise recent research on the correlation between geodiversity, microclimate variation (Vernham et al., 2023), biodiversity (Brazier et al., 2012; Tukiainen et al., 2019, 2022) and ecosystem services (Alahuhta et al., 2018) (*D5.1 Development Process for ECOLOPES Algorithm*) as an example to show the functionality of our approach. For this

purpose, we initially selected the *geomorphons* approach, a pattern-recognition based approach to classify and map landforms (Jasiewicz & Stepinski, 2013) (*D5.1 Development Process for ECOLOPES Algorithm*). Geomorphons are organised as a library of terrain features (e.g., flat, valley, shoulder, ridge, etc) and are based on a 2.5D definition of the terrain surface. We seek to modify this approach with the aim to enable the design of a continuous landform geometry while at the same time deriving a systemic approach to terrain features for the purpose of design. In this context, we realised that it is disadvantageous to consider terrain features as a set of components or tiles, since the edges of neighbouring geomorphons will likely not align and therefore not result in a continuous surface without significant modification of tiles, thereby leading to suboptimal results. We are currently reconceptualising our approach, based on reverse engineering the analytical process of geomorphons. The process of geometric articulation will therefore commence from a "generic" condition of horizontal and vertical surfaces that are transformed in a hierarchical manner into coherent *urban landform* characterised by geodiversity.

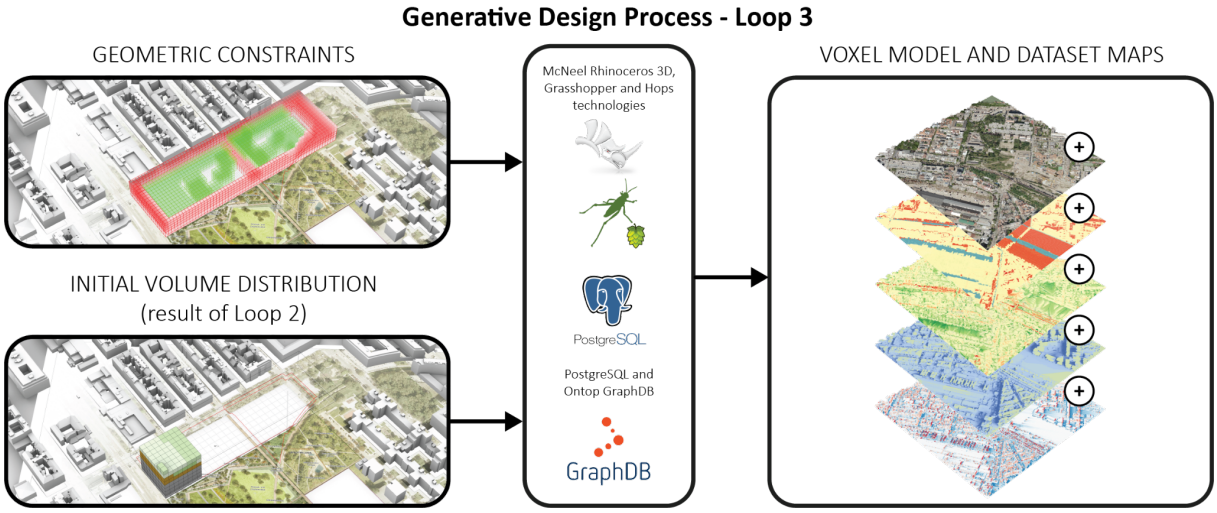

Fig. 6: In Loop 3 the generative design process is initiated to enable detailed geometric articulation for selected spatial organisation. Initial volume distribution created in the previous loop is supplemented by the definition of geometric constraints that are used as inputs to this process. As a result, updated spatial representation of the design process results is created and introduced into the ECOLOPES voxel model.

Each design outcome consists of (1) a CAD Model, (2) a corresponding dataset contained in the voxel model and (3) ontological output. The generative process is facilitated by Loop 2 and Loop 3 (Fig. 2).

EIM Ontology 1 is designed according to the selection and distribution tasks of the design objectives. EIM Ontology 2 is built on top of EIM Ontology 1. It aids the specification of *volumes* and their spatial organisation in the CAD Model 2 . EIM Ontology 2 is programmed to infer design decision rules based on the configured 3D Network iterations, criteria for spatial organisation and specification of volumes, and related CQs.

The final generative phase provides a project- and context- specific design search space and input data for optimization. This entails the generation of a site and building geometry (dataset *landform*). This iterative process is ontology-aided and culminates in the generation of landform geometry expressed with i.e., geomorphons and parametrically, in CAD Model 3

(the geometric approach is still under development) (*D5.1 Development Process for ECOLOPES Algorithms*). EIM Ontology 3 is built on top of EIM Ontology 1 and EIM Ontology 2 and infers design decision rules based on the (1) 3D network from Loop 1, (2) volume distribution output from Loop 2, (3) criteria for landform generation and form generation (through the articulation of slope, aspect, altitude, geometry parameters for example), and (4) related CQs.

Design output is produced for two distinct design cases that are common in architectural practice (see report *D5.4 ECOLOPES Computational Model Validation*). Case 1 entails the design of a master plan for the development of a given site. In such cases the number and distribution of building volumes, including footprint, floor area ratio, maximum volume, and height, are not yet defined. In the context of this research this entails that spatial organisation is generated through the distribution of architectural, biomass and soil volumes, which we term for case 1 *primary volumes*, as well as geometric articulation of site and buildings leading to what we term for case 1 *primary landform*. Landform can therefore be coherently designed across the entire site, with all volumes adhering closely to the landform scheme. Case 2 entails the design of an individual building for which all constraints, such as footprint, floor area ratio, maximum volume, and height, etc. are already established by a municipal master plan. Since the maximum allowed primary volume is already given by the masterplan, the task is to partition the primary volume into *secondary* and *tertiary* architectural, biomass and soil volumes. To enable different species to inhabit the envelope it is useful to develop the building geometry as a *secondary* and *tertiary landform* (hierarchical nesting of terrain features) to enable accessibility and appropriate provisions for specified species to specified parts of the building envelope.

Section 2 of the report focuses on the role of the EIM Ontologies within the generative Computational Design Process, outlining EIM Ontology 1, EIM Ontology 2 and EIM Ontology 3 and their respective roles in the three stages of the generative computational design process. This involves:

1. development of EIM ontology 1 / Loop1 (knowledge graph) that aids the translational process, and involves the preparation of the datasets that underlie the design generation process via a knowledge graph that can be queried by the designer.
2. development of EIM ontology 2 / Loop 2 to aid the generation of the spatial organisation (dataset *volumes*) for design cases 1 and 2;
3. development of EIM ontology 3 / Loop 3 to aid the generation of the geometric articulation (dataset *landform*) for design cases 1 and 2.

Section 3 of the report focuses on the interfaces and interactions between the EIM Ontologies, Voxel Model and Computational Model (the algorithms that facilitate the design generation), and components of the computational framework, namely ecological model and KGF. This is organised according to the three above described EIM Ontologies and the related loops or stages in the generative design process.

Section 4 of the report elaborates the ontological output for subsequent computational processes. The ontological output serves as a structured and organised representation of knowledge captured within the EIM Ontologies, encompassing several vital components. It comprehensively captures design constraints, guidelines, and regulations specific to the

architectural domain, incorporating essential information on building codes, zoning regulations, environmental standards, and other critical criteria that guide the design process. Furthermore, this includes design parameters and their relationships, including geometric properties, material specifications, spatial arrangements, and functional requirements of design elements.

In section 5, we elaborate the ontological output for subsequent computational processes. This includes a description of the different types of data and information (including the input from the Ecological model and the Knowledge Generation Framework), the uses of the ontology for the inference of decision rules and how these are implemented by algorithms in the generation of the three datasets *networks*, *volumes*, and *landform* in CAD. Finally, this section describes the output of the ontology-aided generative computational design process in CAD for design optimization (WP6).

In section 6, we outline the intended future development of the EIM Ontologies, and approaches and methods for validation. The current state and planned further development of the EIM Ontologies within and beyond the project's timeline is addressed.

In section 7 we provide a summary publication plan.

1.2 Note on the Use of the Term "Networks" in PART A

Note regarding the use of the term network in PART A of this deliverable (and in D5.3 ECOLOPES Voxel Model and D5.4 ECOLOPES Computational Model):

The design of architects, be these buildings or masterplans, are assemblages of items and their relations that are expected to become a coherent whole. In this context architects select and relate items, for instance in terms of tectonics the relation between structure and space, or in terms of building program the relation between different spaces, i.e., kitchen and living room in the context of housing, or reading room and book storage areas in the context of libraries. The network character of these relations is for instance inherent in the type of program diagrams that architects produce as design inputs or outputs.

In recent decades the notion of performance-oriented design in architecture emerged that focuses on what architectures and their component parts do, or, put differently what they affect. In this context, specific approaches to performance-oriented design (Hensel 2010, 2011, 2012, 2013; Hensel and Sunguroğlu Hensel 2020) located this aspect in the context of agency and Actor-Network-Theory (Latour 2005).

Agency indicates the capacity to act in the world. This capacity was initially proposed to exclusively relate to conscious action and hence to humans. However, Actor-Network-Theory (ANT) (Latour 2005) positioned agency also as a non-human trait: "*Any thing* that does modify a state of affairs by making a difference is an actor… Thus the questions to ask about any agent are simply the following: Does it make a difference in the course of some other agent's action or not? Is there some trial that allows someone to detect this difference?" (Latour 2005)

Furthermore, Dwiartama and Rosin elaborated: "Actor-network theory asserts that agency is manifest only in the relation of actors to each other. Within this framing, material objects exert agency in a similar manner to humans… human and nonhuman components (both referred to as actants) have the same capacity to influence the development of social-

ecological systems (represented as actor-networks) by enacting relations and enrolling other actors" (Dwiartama and Rosin 2014)

Agency, or the capacity to influence the development of systems, relates to the notion of *performance* as, for instance, posited in performance-oriented design (Hensel 2010, 2011, 2012, 2013; Hensel and Sunguroğlu Hensel 2020). This places emphasis on the *performance capacity* of a broad range of actants. Moreover, this approach challenges established views on system boundaries in that "the outside of any given entity (what used to be called its 'environment') is made of forces, actions, entities, and ingredients that are flowing through the boundaries of the agent." (Latour 2017a)

This perspective suggests an understanding of architectural objects or urban systems as essentially *non-discrete*, and as embedded in a multitude of spatial and functional relations. Moreover, concepts like *agency* and *performance* can be employed across spatial, temporal, and functional scales. This approach requires to (1) identify local actors and actants, (2) to understand and activate the full range of their performative capacity, (3) to understand interactions between actors and actants, (4) identify tentative system borders for types of actors, actants, and interactions, (5) identify the involved spatial, temporal, and functional scales and scale ranges, and (6) to ground all planning and design in considerations focused on *agency* and *performative capacity*.

In this context we utilise an approach based on networks that describe relations between actants of similar or different types. As such this approach lends itself to a multi-species approach and to relations between biotic and abiotic actants. This allows us to work with a methodological approach that is familiar to architectural designers and shared by the logic of ontologies. This enables to configure both domain specific (i.e., architectural) networks, as well as networks consisting of items across domains (i.e., architectural, and ecological domains). In this context we are aware that for instance ecological networks are often differently understood and instrumentalised in the field of ecology. We do not intend to contradict this understanding in ecology, nor do we intent to promote the understanding that ecological networks constitute the exclusive ecological consideration or objectives that need to be considered. Instead, we use the notion in the context of architectural design and in relation to the operations that are familiar to an architect or planner. In future there may therefore exist a need to carefully clarify and adjust terminology across disciplinary divides to better ground interdisciplinary research. We will bare this in mind as we progress with this research.

2. Role of the EIM Ontologies in the ECOLOPES Computational Framework

The role of the EIM Ontologies in the overall ECOLOPES Computational Framework is to model information regarding the entities and relationships that need to be represented and to aid the design of ecological building envelopes. This is facilitated by machine reasoning via capturing decision rules computed in WP3 that can be implemented in design generation and optimisation. In the following sections we provide a conceptual and technical characterization of the EIM Ontologies.

2.1 Conceptual Characterisation of the EIM Ontologies

EIM Ontology 1 entails a comprehensive knowledge graph that aids the translational process in the generative computational design workflow. It will encompass a steadily growing and updated range of design-related information and relationships, including architectural principles, ecological consequences of architectural forms, models, simulations, and related datasets. EIM Ontology 1 will eventually enable designers to query and retrieve specific information necessary for the design process. It acts as a foundation for data-driven algorithmic processes, enabling designers to utilise domain-specific knowledge.

EIM Ontology 2 captures and represents spatial organisation principles and guidelines within the generative computational design process. It currently contains a set of rules and constraints for spatial organisations that govern the distribution and arrangement of architectural, biomass and soil volumes. This ontology incorporates considerations such as architectural and ecological requirements, to guide the ASP-based process that generates spatial configurations. EIM Ontology 2 enables designers to generate a broad variety of spatial organisation outcomes and evaluate their suitability for the given design brief and site.

EIM Ontology 3 aids the geometric articulation (dataset *landform*) of selected variations of spatial organisation generated in Loop 2. It defines rules and parameters that influence the shape, form and detailing of architectural elements within the generated design. EIM Ontology 3 enables designers to generate a broad range of geometric outcomes and evaluate their suitability for a given design brief and site.

Collectively the EIM Ontologies provide a knowledge driving foundation for the generative computational design process.

2.2 Technical Characterisation of the EIM Ontologies

Ontology is a technical term denoting an artefact that is *designed* for the purpose of modelling knowledge about *some* domain (Gruber, 2016). In the context of Semantic Web and AI, one of the most concise definitions of ontology was provided by Gruber (1993), who stated that an ontology is an explicit, formal specification of a shared conceptualisation. A conceptualisation is an abstract and simplified model of some domain of application that needs to be represented. Computational ontologies make it possible to formally model the structure of that system including the domain, the identified relevant concepts, and the relations between them in a consensual manner. This model must be explicit, meaning that all concepts must be expressed in a formal machine-understandable and machine-actionable format grounded in mathematical logic for automated reasoning.

Ontologies are formally represented in languages such as RDFS/OWL in which it is possible to state different axioms or statements. Using an ontology makes it possible to formally describe a domain, increase interoperability, and reason on top. Furthermore, ontologies are used to integrate data from different sources into one unified schema. Ontologies comprise of *classes*, *object properties* and *datatype properties* (so-called *attributes*). *Classes* are used to group individuals in a category (e.g., species), *properties* are used to establish relationships between classes (e.g., speciesHasHabitat), and *datatype properties* or *attributes* are used to describe an individual of a class (e.g., height, size, angle etc.).

The term *Ontology* is in our context considered interchangeable with *Knowledge Graphs (KGs)*. In some cases, ontology is only referred to the schema (Terminological component - TBox), and it is distinguished by instances (Assertional component - ABox). The KG contains both ABox and TBox statements, that is both individuals and schema, including controlled vocabularies. The statements in KGs can be represented using the RDF framework, for both ABox and TBox, using Subject-Predicate-Object (SPO) notation. SPO positions can consist of URIs, blank nodes (anonymous URIs) or Literals (e.g., string), which are allowed depending on the position.

In the following we refer to the query endpoint of KG using the */sparql* endpoint notation, and similarly for rules and constraint reasoning to */asp* endpoint.

EIM Ontology 1 guides the designer in decision making by integrating datasets from networks, maps, objectives as well as species from both local and regional pools. The ontology can be queried by either posing queries to the /sparql endpoint.

EIM Ontology 2 contains the relations that represent architectural, biomass and soil volumes. The selection and distribution of entities in EIM Ontology 1 is used as input for EIM Ontology 2. After the algorithm generates a set of possible variants, based on the constraints and decisions in EIM Ontology 1, data from EIM Ontology 1 will be verified against the rules and constraints of spatial organisation (dataset volumes) in EIM Ontology 2 that are evaluated in an ASP solver. This will create a search space for design variety that will be evaluated by the designer.

EIM Ontology 3 formalises the geometric articulation. It describes geometry in a more abstract form so that it is possible to reason. For instance, by using a parameter such as ‘angle’ one can deduce whether animals can access a surface by using the rules in ASP in combination with Algorithm 3. The output from EIM Ontology 2 serves as input to EIM Ontology 3, creating design outcomes that will be evaluated by the designer and used in the optimization process.

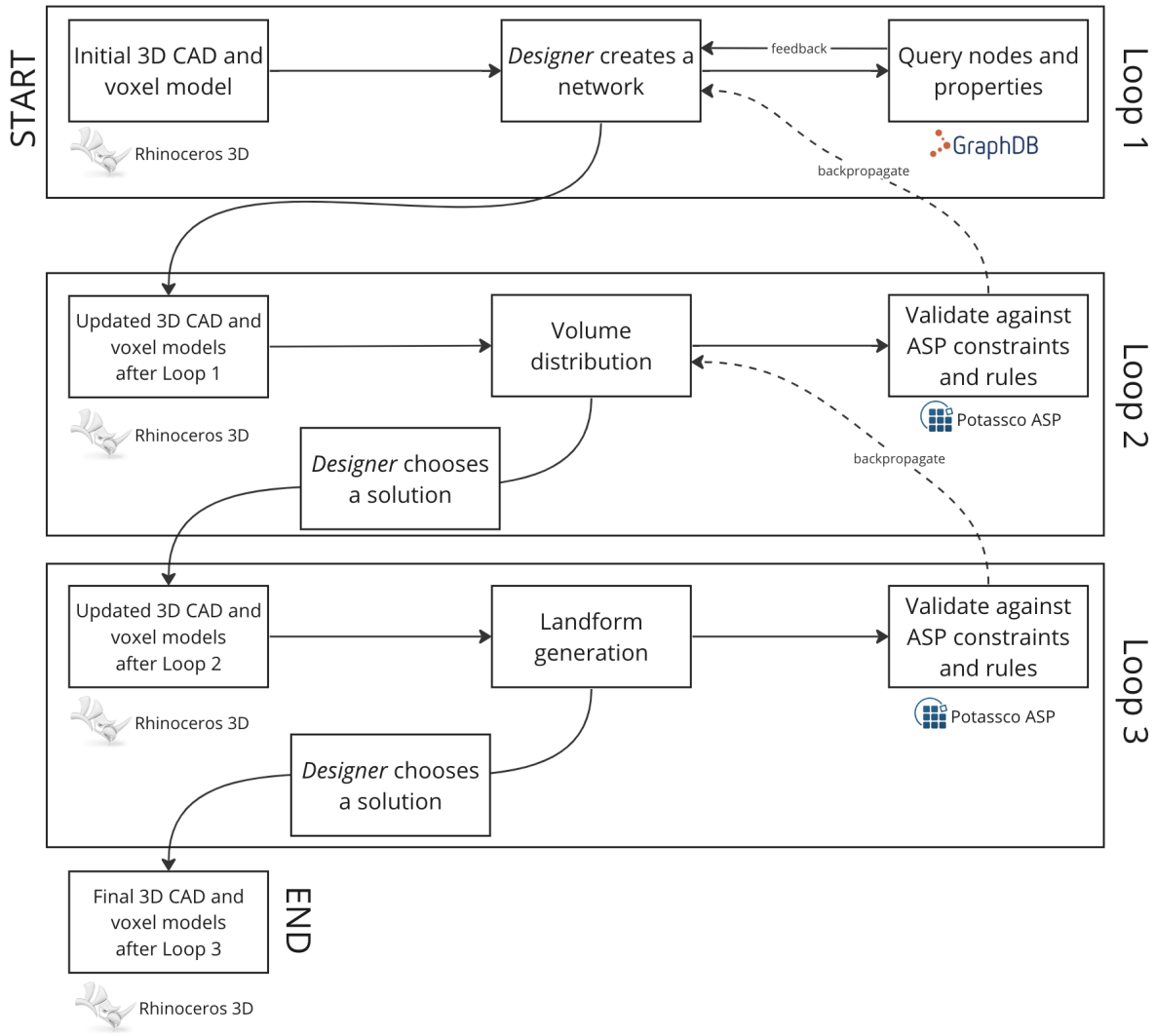

Fig 7: Workflow describing the designer input and the interaction with the respective algorithms in the different loops.

3. ROLE OF THE EIM ONTOLOGIES IN THE GENERATIVE COMPUTATIONAL DESIGN PROCESS

In this section the EIM Ontologies and other system components including the Rule-Based Systems and CAD models, are introduced according to their specific role in the three different design stages and corresponding process loops of the ontology-aided generative computational design process.

3.1 EIM Ontology 1 and Components of Loop 1

In this subsection the role of EIM Ontology 1 and other components of Loop 1 (Fig. 4) of the generative computational design process are discussed in the context of the generative computational design process. This is done with respect to the conceptual, methodological,

and technical approaches that are being explored and introduced. Selected examples demonstrate aspects of the components and how they operate.

3.1.1 The Conceptual Approach

EIM Ontology 1 represents Ecological Networks (ENs) as a knowledge graph. The latter can be reasoned and integrates User Networks (UNs) to generate query results for Competency Questions related to the Selection and Distribution Tasks according to specific criteria established in Loop 1. Furthermore, it enables the implementation of rules inferred from the ontology to aid the iterative configuration of CAD Networks (Ns) in the *translational process*.

EIM Ontology 1 is a knowledge graph that serves to convert design requirements and constraints into computationally interpretable data. It includes various design-related information, such as project-specific requirements, site characteristics, environmental factors, and additional design constraints. EIM Ontology 1 provides a structured representation of this data, allowing designers to access and query relevant information to aid in the design generation process. By using this ontology, designers can effectively navigate and utilise domain-specific knowledge that enables informed decision-making throughout the design workflow.

Loop 1 involves several components interacting with EIM Ontology 1 to drive the *translational* process. These components include the Query Interface, EIM Ontology 1 (Knowledge Graph), ECOLPOES Voxel Model and the Data Preparation Module. The Query Interface allows designers to pose queries to retrieve specific design-related information, acting as a bridge between the designer's input and the underlying knowledge graph. The EIM Ontology 1 (Knowledge Graph) is the structured design data repository, comprehensively representing the design domain and capturing all necessary data for subsequent algorithmic processes. The Data Preparation Module takes the retrieved information from EIM Ontology 1. It prepares the datasets required for further design generation, ensuring that the data is properly formatted and ready for input into subsequent computational algorithms.

Together, these components and EIM Ontology 1 facilitate Loop 1 in the generative computational design process. Their use enables the seamless translation of design requirements into computationally interpretable data, setting the stage for subsequent algorithmic processes and the generation of design variations.

This process involves Ecological Networks (ENs), User Networks (UNs), and Networks (Ns). Currently we are examining the state-of-the-art in the implementation of computational ontologies and knowledge graphs that apply ENs in real-life decision-making. ENs are typically used as a way to describe and compare the structures of real ecosystems and the complex biotic and abiotic interactions within them. This entails, for instance, in which way species (instances of ontological schemas that represent nodes of KGs) might be connected through pairwise relationships (along object properties of ontology that represent edges of KGs) via trophic relationships. A range of methods exist that are used to model ENs for various purposes, such as guiding the planning and management of landscapes based on the concept of connectivity and investigating the effects of network structure on ecosystem stability. Recently, ontologies are used for representing and reasoning over existing ENs to support the planning and expansion, conservation, and improvement of Local ENs (Torta et al. 2017). We use one of those ontologies, geonames, as the base for the EN; a future version may

incorporate a data driven EN that is based on relationships uncovered by the KGF and the ECOLOPES Ecological Model.

The first use of EIM Ontology 1 is to facilitate the application of the concept of ENs to describe and infer ecological networks that can potentially be created and expanded, and to integrate existing or real local ecological networks in building envelopes. ENs are expressed as KGs, where nodes are instances that are derived from a selection process with edges that connect the nodes in the network obtained from the distribution process. The second use of EIM Ontology 1 is to facilitate the initialisation of the *translational* process via user-information and UNs configured by the designer in the CAD environment based on the design brief, building and planning regulations, existing conditions, and project constraints, etc to initialise the *translational* process. The constraints contained in planning regulations and the architectural and ecological requirements contained in the project brief are localised as items and relations (Networks) in the CAD environment by the designer, who queries EIM ontology 1 (KG) to ascertain that implemented configurations are permissible and who utilises data contained in the ECOLOPES Voxel Model to inform the placement of items and relations. Hence all ecological objectives that can be described as networks (see note on the term networks in the introduction) can be incorporated.

3.1.2 The Technical Approach

The first step towards establishing EIM Ontology 1 is the naming scheme of URIs. We defined the URI generation method for identifying entities in the ontology in an unique way , namely “classes”, “properties”, “instances”, etc. It is possible to deduce from the URI whether a concept is a “class”, “property”, or “instance”. This follows a best practice approach for designing a knowledge graph. The URI generation is structured as follows:

For “classes”, “properties” and “datatype properties” we use a pattern that starts with “schema” followed by the domain and separated with “#”: https://schema.dap.tuwien.ac.at#{Class/Property/DatatypeProperty}

The following are examples that demonstrate the approach by using “classes”, “properties” and “datatype properties” respectively:

https://schema.dap.tuwien.ac.at#Habitat

https://schema.dap.tuwien.ac.at#comprises

https://schema.dap.tuwien.ac.at#impactHeight

For instances, we have a pattern that starts with “resource” followed by the domain and separated with “/”:

https://resource.dap.tuwien.ac.at/{Resource}

The following is a concrete example (Note that we additionally use “/Species/” in order to be able to differentiate individuals based on a group based on the URI):

https://resource.dap.tuwien.ac.at/Species/bromusCommutatus

To explain the querying and reasoning interface of KG, it is useful to consider sample excerpt data that is stored in GraphDB.

There are different serialisations to express statements in RDF. For simplicity we use Turtle syntax (omitting the defined prefixes for readability):

TBox:

```
:Animal rdfs:subClassOf :Species . :Plant rdfs:subClassOf :Species .
:hasKingdom rdfs:domain :Species ; rdfs:range :Kingdom .
```

ABox:

```
:bufotes_viridis a :Animal ; rdfs:label "Bufotes viridis" ; :hasKingdom :Animalia .
```

Note that in this KG we have described "Bufotes viridis" in a set of assertional facts. Together with the TBox statements we infer that:

`:bufotes_viridis a :Species .` (following subclass axioms/relationships)

`:Animalia a :Kingdom .` (following range axioms/relationships)

Queries are written in SPARQL language using SPO notation with variables, which has the similarity and expressivity of SQL. For instance, in a SPARQL endpoint where reasoning is enabled, this query will also return `:bufotes_viridis`.

```
SELECT * WHERE { ?X a :Species }
```

Otherwise, if the SPARQL endpoint has no reasoning enabled, then in order to get the same result we have to rewrite the query so that it takes reasoning into account using property paths. This can be achieved in the following ways:

```
SELECT * WHERE { ?X a/rdfs:subClassOf* :Species }
```

Figure 8 shows a screenshot of EIM Ontology 1 that is maintained in Protégé. The ontology is exported to GraphDB in order to be queried using the /sparql endpoint with reasoning enabled.

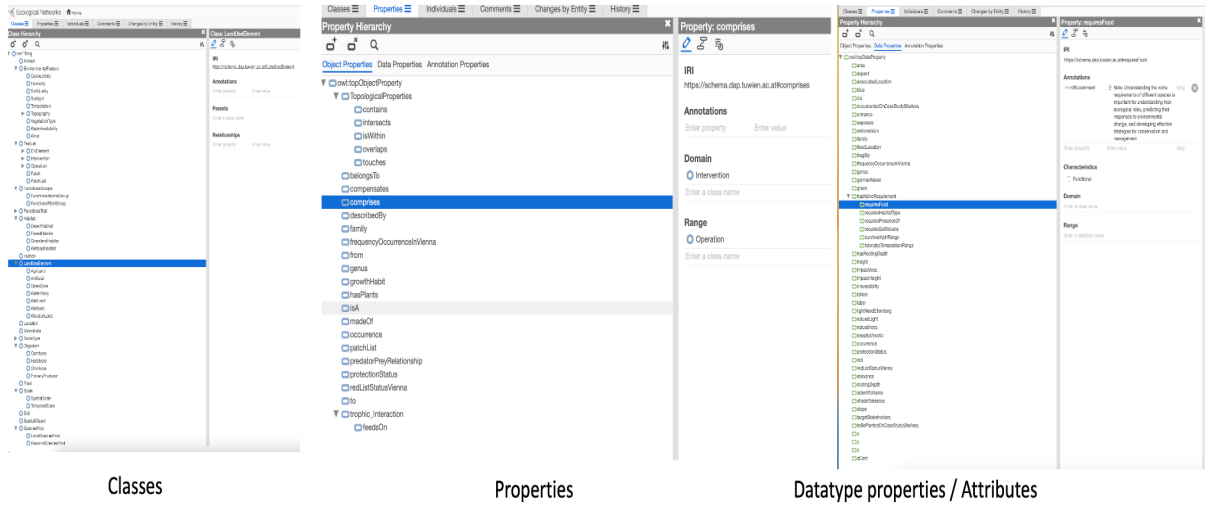

Fig. 8: Ecological Networks Ontology maintained in Protege, exported to GraphDB for querying and reasoning via the /sparql endpoint.

The visualisation of the Ecological Networks Ontology using VOWL Plugin in Protege is shown in the following figures.

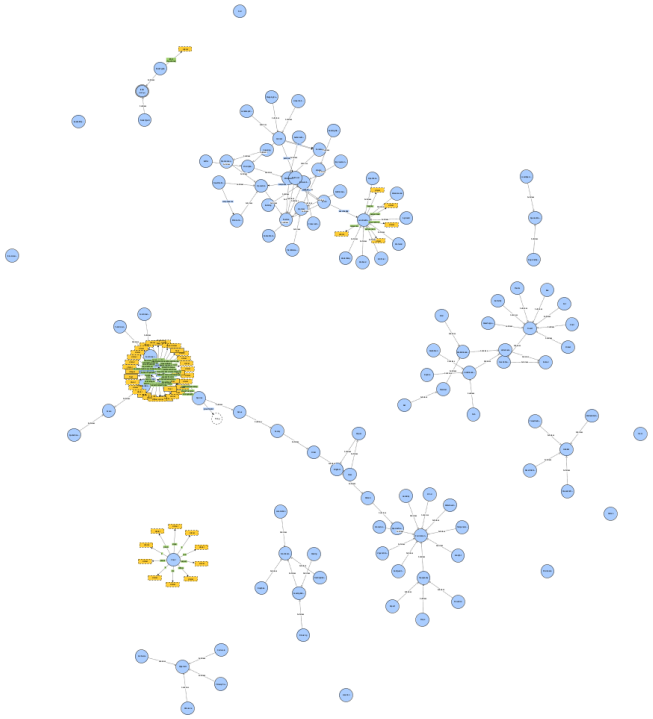

Fig 9. An example of an entire Ecological Networks Ontology visualised using VOWL Plugin in Protege.

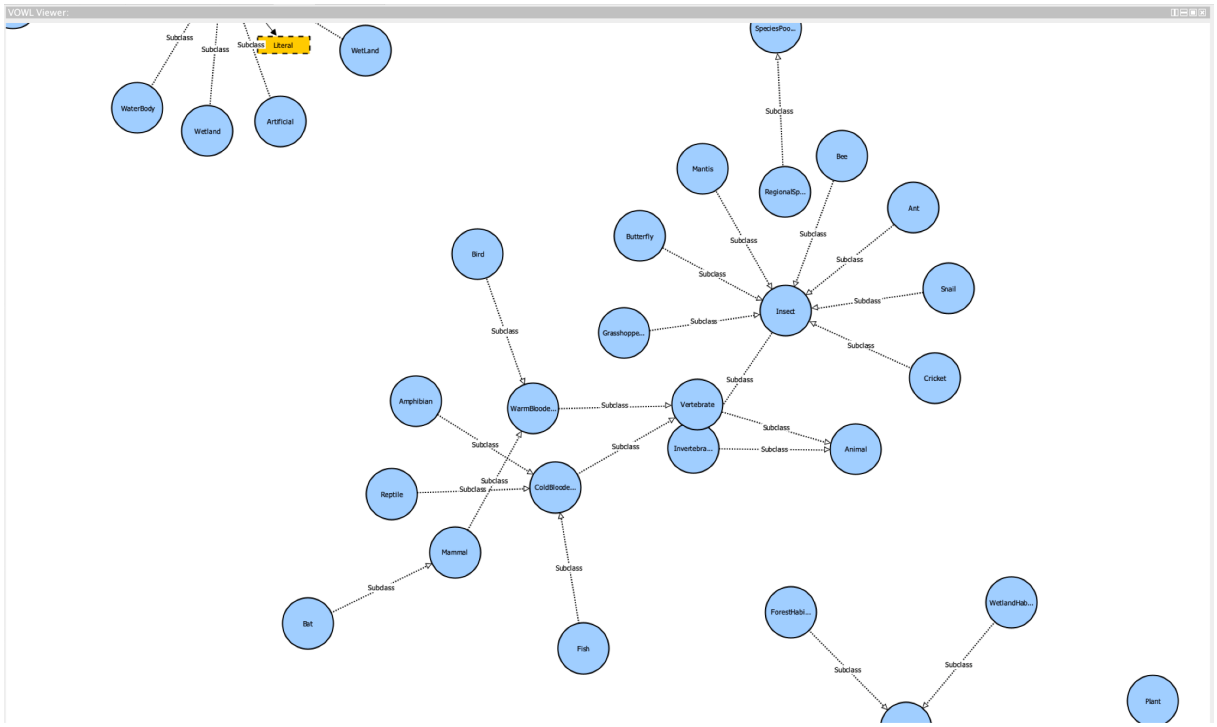

Fig 10.: This illustration shows a detailed part of ontology visualised using classes and subclass relationships in Protege.

A part of the queries comes from the defined competency queries that can be used as templates. These competency queries are defined such as "*Give me all species that are threatened?*".

```
SELECT * WHERE { ?s a :Species ; :status :threatened }
```

To address this, we must adjust the gap in the data and see whether we need to integrate databases that can help us with this question. Hence, these competency questions help us as a gap analysis.

We have created each competency query in .rq format (SPARQL), altogether with the comment starting with # that describes the competency question, the actual SPARQL query, and they are all stored in Github for further reuse. For example, one can use a CURL command to call a query from the list of competency questions against the GraphDB's SPARQL endpoint to see if the queries are respected by the repository, i.e., they return non-empty bindings. In an analogous way, we created ASK queries as a counterpart that checks only for existence of bindings as a boolean answer, but not does not return the actual bindings itself.

```
ASK WHERE { ?s a :Species ; :status :threatened }
```

There are many ontologies published on the Web where one can reuse them to describe one's domain in a standardised form, e.g., schema.org, geonames, PROV-O etc. In our case, for EIM Ontology 1 we have reused geonames ontology and we use it to describe features, together with the geo locations.

In addition, EIM Ontology 1 contains all the classes, relations and attributes that are used to describe a local plants and animals dataset, which was downloaded from the environment map of the City of Vienna "Stadt Wien - Umweltgut" (https://www.wien.gv.at/umweltgut/public/grafik.aspx?bookmark=Ibk4RllSHEaPGzVGXUS6RTnC-cs6-crOsX3Z-cJ1b3VCe85UpT2o-cKSUyDCj9L6i0kPs1sMk0mqGZi4MR2OtT6Z0YAnvXpmj1OY4b4EuXJ7jPaPzYYMoOMC-cp-aLmAD1Aw-b-b&bmadr=65346909). Also habitats and which species live in those habitats, as well as prey relations. In the context of the planned Vienna Case Study, which will serve the purpose of validation, it also contains birds located in the region of Vienna, represented in RDF, which can be queried using GeoSPARQL, in this way retrieve all birds that are close to the specific point or site. These are all classes, properties and attributes that are extracted from the datasets in bottom-up fashion.

To make the approach more generalizable in the example of the Ecological Network considered here, also the ontologies in this domain are studied and they are reused whenever possible, adhering to Linked Data and Semantic Web principles where reuse and using common vocabularies and ontologies is always strongly suggested. The advantage of this approach is that if the data is contextualised and reuses common vocabularies, then it is better understood for both humans and machines alike. In future the PFG/AFG definitions may be added to the ontology, as they are written into machine-readable JSON files. One can query for instance "all plant functional groups that have minimum soil depth requirement < 10 cm" and do the same for the regional functional group pool information.

3.1.3 Open Questions

Open questions related to EIM Ontologies in Loop 1 include:

- Integration of Plant Functional Groups to enable the designer to select and distribute plants based on their role in the ecosystem (functional groups);
- Extension of competency questions based on the Vienna use case.

Open questions related to User Networks include:

- Specification of edge types to express different relations between nodes.
- Specification of technical interface and between user network and GraphDB through sequence diagrams.

3.2 EIM Ontology 2 and Components of Loop 2

In this subsection, the role of EIM Ontology 2 and other components of Loop 2 (Fig. 5) of the generative computational design process are discussed in the context of the generative computational design process. This is done with a focus on the conceptual, methodological, and technical approaches that are being explored and introduced. Selected examples demonstrate aspects of the components and how they operate.

3.2.1 The Conceptual Approach

EIM Ontology 2 builds on EIM Ontology 1 and is developed to generate query results for Competency Questions related to the volume specification task according to specific criteria established in Loop 2, and to enable the implementation of rules inferred from the ontology to aid the iterative distribution of CAD Volumes in the Generative Process_1.

3.2.2 The Technical Approach

The purpose of EIM Ontology 2 is to aid in the spatial organisation, i.e., volume distribution process as explained previously. In the previous Loop 1, we had selection and distribution with the interaction coming from the designer and KG. After the initial set of graph configurations in the Loop 1, we check whether the volumes are subject to whether they satisfy the constraints laid by the general rules of functional ecosystems. In this case, the feedback is not as involved as in the Loop 1, as we must check against a general form of rules.

For demonstration purposes of rules, let us take an example: "*if volume is Biomass, then below it, the volume should be either Soil or Biomass.*"
This expressed in logical terms will be formally defined as:

```
volume(X, Y-1, Z, b) v volume(X, Y-1, Z, s) :- volume(X, Y, Z, b)
```

- **v** — refers to 'logical or';
- **:-** — refers to 'if then condition'

The verification and the return of solutions is done in the Algorithm 2, that are implemented using `/asp` stable model semantics by incorporating rules of the above form with a different syntax. Each stable model is a solution to a search problem that can be considered in the variety set of Generative Design Process.

The purpose of EIM Ontology 2 is to bring the data as input for verification in order to check for conformance with the ecosystem rules (such as above) and see if there are a set of solutions. For this the data is fetched from the `/sparql` endpoint (the results of Loop 1) by issuing a query to GraphDB and further expressed as input to Algorithm 1.

The component that is used herein is the one that translates the RDF data and makes them as input to ASP solver, thus creating a bridge. This transformation takes care of translating *binary* predicates (e.g., `hasCoordinateX(a, 1), hasCoordinateY(a, 1), hasCoordinateZ(a, 1), hasVolumeType(a, s)` to a form that is n-ary (e.g. 4-ary as seen volume predicate above - `volume(1,1,1,s)`.

Like Loop 1 the components of GraphDB such as querying, reasoning and constraint checking can be used in case the designer wants to be more involved in the design process with further questions.

3.2.3 Open Questions

A key question regarding EIM Ontology 2 in Loop 2 comprises the placement of architectural, biomass and soil volumes in relation to the Moore system which needs to be extended to a system with diagonals, i.e., for each volume, not only addressing up, below, left or right, but also diagonals (Orciuoli et al, 2017).

3.3 EIM Ontology 3 and Components of Loop 3

In this subsection, the role of EIM Ontology 3 and other components of Loop 3 (Fig. 6) of the generative computational design process is discussed in the context of the generative computational design process. This is done with respect to the conceptual, methodological, and technical approaches that are being explored and introduced. Selected examples demonstrate aspects of the components and how they operate.

3.3.1 The Conceptual Approach

EIM Ontology 3 builds on EIM Ontology 1 and EIM Ontology 2 and is developed to generate query results for Competency Questions related to the Landform Generation Task according to specific criteria established in Loop 3, as well as to enable the implementation of rules inferred from the ontology to aid the iterative generation of CAD landform geometry in the Generative Process 2.

3.3.2 The Technical Approach

The purpose of EIM Ontology 3 is to aid the generation of geometry of building volumes (and for design case 1 landscaped surfaces) (dataset *landform*). This ontology describes and represents a building or feature in an *abstract form*. The abstract representation of the

building is used for describing it and for performing different computations tasks such as reasoning. The abstract representation does not describe a 1-1 (one-to-one) relation with the actual building or feature, in other words does not represent the building in its entirety. Instead, this representation captures only the *essential features* that facilitate description reasoning. For instance, the designer can use it to infer or deduce whether a certain fact will hold or not, in our examples whether 'bufotes viridis' can climb the building. In the following we give an example of how such an ontology is structured:

```
:feature/1234567 a geonames:Feature ;
     rdfs:label "Building X" ;
     geo:lat "27.988056" ;
     geo:long "86.925278" ;
     :angle '45' .
```

For instance, a concrete requirement can be that an *ecolope* should be accessible to animals such as, for instance, 'bufotes viridis'. This requirement entails that the angle of a selected building face or surface should not be 90'. The ASP rules would flag this up and the generation algorithm would give a concrete set of solutions (*geometric variety*), which in this case are angles that satisfy the requirement.

In sum EIM Ontology 3 is an abstract representation of the building that satisfies set requirements and rules. The rules defined in ASP hence inform and constrain geometric articulation. The graph data is transferred from EIM Ontology 2 to EIM Ontology 3 so that we take as input the volume distribution generated in the previous step. Subsequently, different abstract representations of geometries of specific faces or surfaces of the building that satisfy the constraints will be the solution set returned from /asp solver endpoint. The components of GraphDB (querying, reasoning, constraint checking, etc.) can be used similar to EIM Ontology 2 if the designer has further queries.

3.3.3 Open Questions

Open questions related to Ontologies in Loop 3 include:

- To what extent do we need to abstract a building so that we can reason effectively with ASP?
- Which ontology can we reuse to describe a building such as BOT, bricks?

4. Interfaces between EIM Ontologies, Rule-Based Systems, and CAD Models

In this section, we describe the interfaces between the EIM Ontologies, rule-based systems, and CAD models. In the following, we discuss the interfaces from the perspective of each Loop.

The interfaces in Loop 1 have been integrated except for integrating the Ecological Model. The designer-related integrations need to be integrated in the workflow and the interaction with other components needs to be tested. The interfaces in Loop 2 are conceptual except for the algorithmic part which is developed and need to be tested and integrated. Loop 3 is

conceptual, for the time being, and will be further developed and implemented (see *D5.3 ECOLOPES Computational Model*).

4.1 Interfaces of Loop 1

In this section we describe the interfaces that are part of Loop 1, namely the interface between databases and EIM Ontology 1, ecological model and voxel model, ecological model. Furthermore, we describe the EIM Ontologies, interaction between EIM Ontology 1 and the algorithms, EIM Ontology 1 and CAD, and designer and EIM Ontology 1. We go into more details for each one of them in the subsections below.

4.1.1 Databases and EIM Ontology 1: DB→O

This section describes the interface between Database and ECOLOPES Voxel Model in Loop 1 (Fig. 2, Fig. 4, Fig. 11, Table 2). In this context, a database refers to a data store that holds contextual data (i.e., regional plants) coming from different sources (i.e., open government data). Depending on the source, this can be GEO-Data, CAD-Data, Simulation-Data, Survey-Data, or spatial or spatio-temporal data or datalists. At this stage, the non-spatial data can feed directly into Ontology 1, whereas spatial data is first combined and structured in what can be conceptualised as a Voxel Data Model, using RDB (*D5.2 ECOLOPES Voxel Model*), according to the needs from the ontology perspective (i.e., computational reasoning, interoperability). From this perspective, the database(s) and voxel model (purposefully structured data) provide instance data that are used to instantiate the ontology. In this way data coming from different sources that is collected or generated with different methods in different scales (i.e., geological, architectural), and which is based on different referencing systems (i.e., Geographic and Cartesian Coordinates, or Moore Neighbourhood), can be fused and made operable with EIM Ontology 1 and serve as input to an ASP program. Hence, all aspects of Loop 1 will be covered.

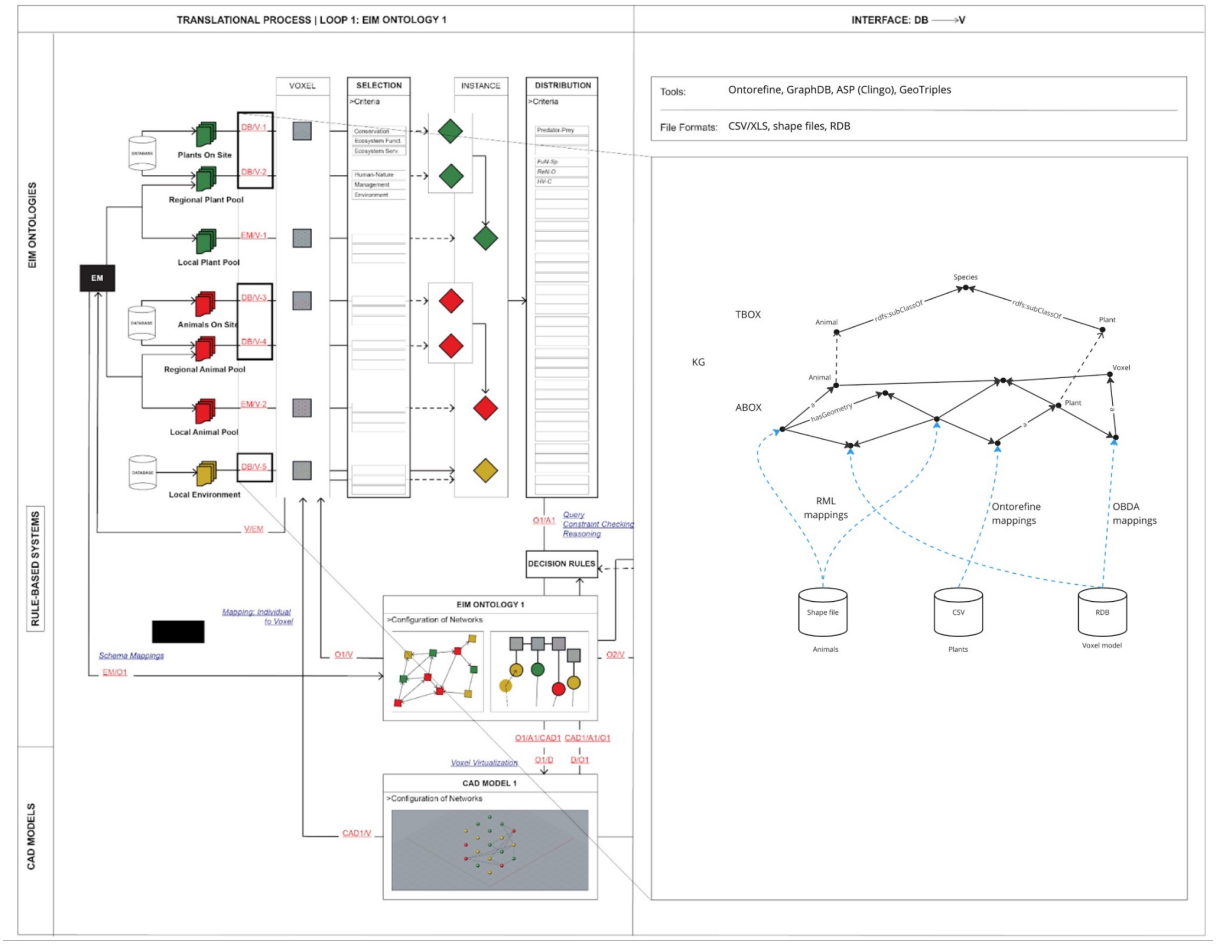

Fig. 11: Loop 1 and the interface DB → V with information about the computational tools used and data exchange formats, as well as an illustration exemplifying the interface.

Figure 12 shows how different datasets can be mapped and transformed to the KG. Some of the datasets are virtualized (e.g., Voxel model), while others are materialised, i.e., stored explicitly (Animals and Plants) in RDF. For each dataset there exist appropriate mappings, namely for shape files RML mappings are generated that convert data to RDF by preserving Geometry. For CSV OntoRefine mappings are created by using GUI that map each column of CSV to a property in the ontology. The Voxel model uses OBDA mappings that are used to virtualize the data into RDF form. Therefore, the KG integrates heterogeneous data from different sources, providing unified access to the data by using the SPARQL query language. In the case of virtualized data, SPARQL queries are translated to SQL on the fly.

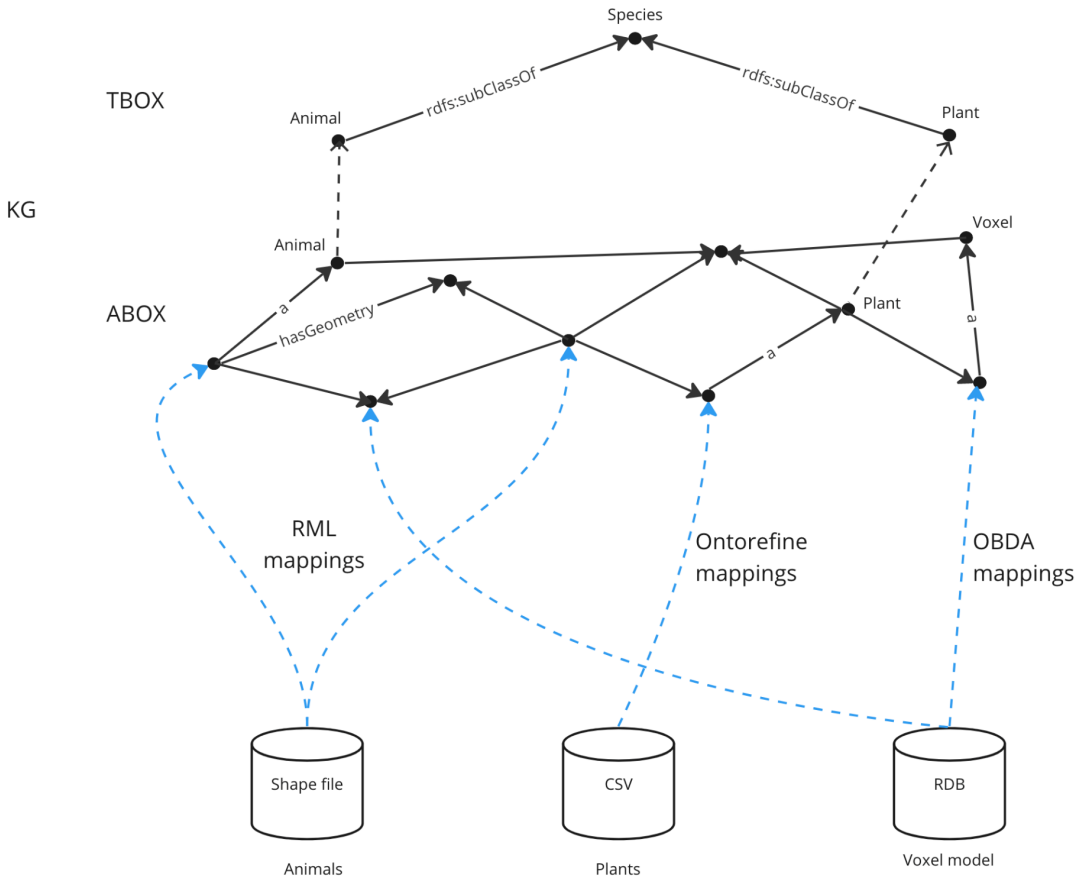

Fig. 12: Heterogeneous data is mapped and converted into a graph form in KG (ABox) by using different mapping languages such as RML, Ontorefine and OBDA. Some of the data is virtualized meaning data is not copied (Voxel model), while some data is copied/materialised. The user can query the KG as a unified interface that encompasses and integrates different datasets combining the results, and, if required, reasoning on top using TBox statements.

The databases, which store unstructured and structured data (Voxel Model Data), provide input for EIM Ontology 1 in the *translational* process. The datasets that are contained include, but are not limited to the following:

(1.0) Plants on Site (DB/V-1): This geospatial dataset may specify which plant species exist and where they are in an *ecolope* site. This data might come from ecological surveys carried out for design and construction in support of a planning application or environmental impact assessment and help determine the ecological constraints at an early stage. It might form a part of the design brief.

(2.0) Regional Plant Pool (DB/V-2): This is a geospatial dataset (i.e., shape file), which specifies and potentially locates plant species that exist in a region and can potentially occur and be observed in an *ecolope* site. This might come from, i.e., open government data and local planning authority, might form a part of the design brief. Whether these plants can reach the *ecolopes* site is calculated with the regional model (see Part B).

(3.0) Animals on Site (DB/V-3): This geospatial dataset may specify which animal species exists and where they might be located (i.e., based on observations) in an *ecolope* site. This data

might come from ecological surveys carried out for design and construction in support of a planning application or environmental impact assessment and help determine the ecological constraints at an early stage. It can also be derived from the regional species pool using the regional model that calculates whether the species can reach the *ecolopes* site (see Part B). Such species may become a part of the design brief.

(4.0) Regional Animal Pool (DB/V-4): This is a geospatial dataset (i.e., shape file), which specifies and locates animal species that exist in a region and can potentially occur and be observed in an *ecolope* site. This might come from, i.e., open government data and local planning authority, and might form a part of the design brief. It serves as input for the regional ecological model.

(5.0) Local Environment (DB/V-5): This dataset may include environmental factors at the local (*ecolope*) scale (borders expandable) that are important for design decision-making and may include biotic and abiotic factors that may influence selection and distribution choices and outcomes in the configuration of networks.

4.1.2 Ecological Model and Voxel Model: EM→V

The following elaborates the future interface between Ecological Model and Voxel Model in Loop 1 (Fig. 2). From the ontology perspective, it is important that the Ecological Model (EM) (see *D4.1 Preliminary EIM Ontology*) can simulate species distribution and abundance at species and community level interactions, because this can provide essential expert analysis-derived data that is otherwise currently missing and difficult to obtain for each site. Therefore, it is useful if the EM can produce precise contextual information and thereby complement and enhance the more general and less accurate ecological datasets. The EM can essentially generate a new dataset on Local Species Pool, thereby extending the number of possible datasets that can feed into the ontology, but also can be expanded and used to verify Regional Species Pool as input in the translational process:

(2.1) Regional Plant Pool (EM/V): This non-spatial (i.e., CSV file) or spatial dataset (i.e., spatial Data in R) results from statistical modelling, which computes which plant functional group that exist in a region can potentially reach and colonise an *ecolope* site under constraints at the regional/city scale (100x100m/voxel/2.5D). If this is a list, it can be directly used to instantiate the ontology and expand the list of individuals for further processing in Loop1. If this is spatial data, then it will be first structured in the voxel model to be prepared as an input for the ontology. Regional Plant Pool data is a one-time input into the process at the initial stage.

(4.1) Regional Animal Pool (EM/V): This non-spatial (i.e., CSV file) or spatial dataset (i.e., spatial data in R) results from statistical modelling, which computes which animal species that exist in a region can potentially reach and colonise an *ecolope* site under constraints at the regional/city scale (100x100m/voxel/2.5D). If this is a list, it can directly feed into the ontology to instantiate it and expand the list of individuals for further processing in Loop1. If this is spatial data, then it will be first structured in the voxel model to be prepared as an input for the ontology. Regional Animal Pool data is a one-time input into the process at the initial stage.

(6.0) Local Plant Pool (EM/V): The Ecological Model is being developed to generate spatio-temporal data derived from the simulation of ecosystem dynamics between plants, animals, soil, and architecture to compute and estimate plant distribution and abundance over time in

3D CAD space at the local/architectural scale (1x1m/voxel/3D). Thereby what it produces is a dynamic dataset that changes over the course of a building life cycle and design iterations under constraints. It generates detailed information about plant occurrence and community assembly in an *ecolope*. This spatio-temporal data is added into and structured in the voxel model and prepared for input to instantiate the ontology at each design iteration.

(7.0) Local Animal Pool (EM/V): The Ecological Model is being developed to generate spatio-temporal data derived from the simulation of ecosystem dynamics between plants, animals, soil, and architecture to compute and estimate animal distribution and abundance over time in 3D CAD space at the local/architectural scale (1x1m/voxel/3D). Thereby what it produces is a dynamic dataset that changes over the course of a building life cycle and design iterations under constraints. It generates detailed information about animal occurrence and community assembly in an *ecolope*. This spatio-temporal data is added into and structured in the voxel model and prepared for input to instantiate the ontology at each design iteration.

4.1.3 Ecological Model and EIM Ontology 1: EM→O1

This section describes the not yet implemented interface between Ecological Model (EM) and EIM Ontology 1 (O1) in Loop 1 (Fig. 2, Fig. 7), focusing on questions of interface and interoperability in terms of semantic alignment, controlled vocabularies and taxonomies, existing data standards, automatic classification, etc. Furthermore, it explains how EM standards apply to the representation of Ecological Networks (ENs) and how data that is generated by the EM at each design iteration feeds into Ontology 1 / Knowledge Graph for instantiation. This direct interface allows data that is not spatial and thus does not need to be structured in the voxel model to populate the ontology.

The Ecological Model consists of different types of data, ranging from local and regional pools including both animals and plants. Regional data has been collected from the Vienna government, in this way shaping the ontology with the respective details coming from the data sources that are needed to model them for one specific model case. EIM Ontology 1 was therefore constructed based on a bottom-up approach to KG construction. Furthermore, we reused existing ontologies from Ecological Networks in a top-down approach based on literature research. The data has been provided in different forms, ranging from CSVs to shape files. The shape files have also been converted to graphs. This allows us to make queries that are relevant in the geographical sense, such as “closeness”, “adjacent”, etc, in combination with other data that are semantically interlinked. Moreover, local data will be instantiated from the existing site.

The translation of the various data models and formats into the graph representation in RDF is done by using a set of mappings (Fig. 8). The mappings specify how a column, or a query is mapped to an ontology predicate. Mappings are a declarative way to specify how a column (in general form: a query) in a tabular data is converted to a property in the ontology. This means that the data cell in that column would be used as an object (note that each triple consists of <subject> <predicate> <object>) by using the property in which the column has been mapped to. For instance, the column [scientific name] has been mapped to the property https://schema.dap.tuwien.ac.at#scientificName, which we can denote it with the symbol “->”:

[scientific name] -> https://schema.dap.tuwien.ac.at#scientificName

In the case of the plant dataset, after the mapping process is executed, there will be a triple created for the specie “Ballota nigra”:

<https://resource.dap.tuwien.ac.at/Ballota%20nigra>
<https://schema.dap.tuwien.ac.at#scientificName> “Ballota nigra”.

In the case of tabular data such as CSV, we have applied the mappings created directly by using the OntotextRefine tool, whereas for shape files we used GeoTriples application that creates a RML mapping for creating RDF data. The resulting data in RDF is stored in our GraphDB triple store.

The EM uses functional groups as basic units for simulations. The FGs do not directly map to species taxonomies, but a small selection of key species can be manually assigned to FGs for the Vienna Case Study. We aim to map EM to O1 after the Vienna Case Study if time permits.

4.1.4 EIM Ontology 1 and Algorithm 1: O1→A1

This section describes the interface between Ontology 1 and Algorithm 1 in Loop 1 (Fig. 2). EIM Ontology 1 is developed for the representation of Ecological Networks (ENs) as a Knowledge Graph. It integrates User Networks (UNs) defined by the designer according to the context, project, and user specific determinations and input. Finally, it reasons over the KG for inference of decision rules using ASP, which are implemented by Algorithm 1 under predefined and emergent constraints. The tasks of O1 and A1 is to facilitate the (1) selection process and aid filtering those individuals (ontology instances/KG nodes) that will be included in the KG (i.e. selection from the regional plant pool according to conservation objectives), and spatial (2) distribution of individuals with non-spatial, spatial, or spatio-temporal EN relations (i.e., prey-predator relationships, ecological niche relationships (e.g., according to the fundamental niche of a selected species)), and guide the configuration of 3D Networks (Ns) in CAD.

Algorithm 1 consists of reasoning techniques including querying, constraint checking, and reasoning with ASP rules. This rule-based algorithm ensures sound and complete answers to the queries and constraints. As an example, this ensures that prey-predator relationships are put into the design context with requirements that are established in a graph, instead of a generic ecological query. The (2.5D/3D data) spatial or (2.5D/3D and time-stamped data) spatio-temporal instances in the graph are then registered as nodes/voxels in CAD. This results, for example, in the assignment of voxels to the local species pool, which may include selected and derived (inferred) instances. Such an assignment can be seen in Figure 2, under the CAD Model 1 in Loop 1, in which a plant, animal or local environment is associated with a voxel (or, alternatively, a multidimensional array).

EIM Ontology 1 is used in Algorithm 1 as there are defined relationships such as “preysOn”, “threatened species”, “invasive species” etc. and together with the data coming from databases are fed to Algorithm 1. The algorithm helps with both the “selection” and “distribution” process (see Fig. 2).

4.1.5 EIM Ontology 1 and Algorithm 1 and CAD Model 1: O1→A1→CAD1

This section describes the interface between EIM Ontology 1 and Algorithm 1 and CAD Model 1 in Loop 1 (Fig. 2). At this stage of the *translational* process the Ontology is used for representation and reasoning of KGs, which integrates ENs and UNs, and a rule-based algorithm that can enable human controlled and automated reasoning to infer design instructions or decision rules for the iterative configuration of 3D Networks in the CAD environment (Ns of CAD Model 1) under predefined and emergent constraints. These constraints might be coming from the ontology (axioms, assertions, etc.), from designer defined inputs (see sections 4.1.6 and 4.1.7), and general / meta-level design rules that apply to all design cases that are integrated in Algorithm 1 (see *D5.3 ECOLOPES Computational Model*). Ontology 1 is configured based on a set of competency questions (CQs), which will need to be revised and expanded over time (see section 3.1) related to the selection and distribution processes and the configuration of 3D Networks in CAD (CAD 1). This rule-based generative process using ASP generates single or multiple solutions, which then can be evaluated by the designer manually and/or given as an input for graph optimization using machine learning to guide the selection of "best" 3D network alternatives in each iterative step in the translational process. We use a voxel-based system that allows retrieving and representing spatial nodes and turning instances of the ontology into voxel data points in 3D space in CAD.

In the context of the Vienna use case, we will illustrate the concepts we have discussed with concrete examples. The use of EIM Ontologies plays a significant role in guiding the generative computational design process for designing ecological building envelopes. EIM Ontology 1 contains information about urban design principles, regulations, and constraints in Vienna, such as building heights, setback requirements, green spaces, and transportation infrastructure. Designers can access EIM Ontology 1 to retrieve relevant data and ensure compliance with the specific urban design guidelines of Vienna. Additionally, the ECOLOPES Voxel Model integrates various datasets that are required or desired for the design process. By incorporating geospatial data such as topography, land use patterns, and existing infrastructure, the voxel model enables designers to visualise and manipulate the physical and environmental context of the project site. This empowers the designers to make informed decisions regarding the optimal placement and configuration of buildings, open spaces, and transportation networks within the city of Vienna.

The CAD environment employed in the Vienna use case takes advantage of algorithmic processes guided by the EIM Ontologies. For instance, designers can utilise Answer Set Programming (ASP) rules to automate the analysis and generation of design variations. By encoding design constraints and objectives into ASP rules, the CAD environment can generate alternative solutions that meet the specified criteria. The validation includes assessing the performance and effectiveness of the process in generating sustainable and contextually responsive designs in the urban environment of Vienna.

By utilising this voxel-based representation, the design process can effectively incorporate spatial considerations, allowing for more comprehensive and informed decision-making. Additionally, the voxel-based system enables the capturing and preservation of spatial relationships between nodes, enhancing the accuracy and fidelity of the design

representation. This representation facilitates a more holistic understanding of the design space, allowing designers to visualise and manipulate the elements in three-dimensional space. By converting instances of the ontology into voxel data points in 3D space within CAD, designers gain the capability to interact with the design at a granular level. They can manipulate and modify the voxel-based representation, exploring different design alternatives and evaluating their feasibility in real-time. Moreover, the integration of machine learning techniques in the graph optimization process further enhances the design selection process. By leveraging data and patterns, machine learning algorithms can provide valuable insights and recommendations to guide the designer in selecting network alternatives. This iterative feedback loop between the designer, generative process, and machine learning optimization ensures a continuous improvement and refinement of the design solution.

The integration of this interface is done using the Hops component, where we query and reason with the data from Ontology 1 stored in GDB. Typical queries include for instance:

- “What is the context of a particular node type?”
- “What kind of relations exist for this node type?”
- “Which kind of relations exist between node types?”

Answers are provided by the KG. In addition, the designer can ask questions that are specified in the design brief and which have been formalised in KG and can be queried.
Regarding Algorithm 1 this is used for the next step of reasoning with constraints and rules provided as input by the designer in the form of networks. Such constraints and rules are fed to the ASP program, which is the core of Algorithm 1.

4.1.6 EIM Ontology 1 and Designer: O1→D

This subsection focuses on the interface between EIM Ontology 1 and Designer in Loop 1 (Fig. 2). It shows how the information coming from User Networks (UNs) and 3D voxel space (i.e., map datasets), which is predefined and preconfigured by the user based on the design brief, local building and planning regulations, design intentions, and site analysis (i.e., solar data) provide input into the ontology as constraints in the initialization of the algorithmic process. Furthermore, this section shows how the designer interacts with the ontology in the iterative generation of 3D Networks in CAD based on instructions derived from EIM Ontology 1. Finally, this section describes how UNs are validated with the help of Ontology 1. This interface is established via GraphDB and the Hops component of Rhino.

4.1.7 Designer and EIM Ontology 1: D→O1

This subsection describes the interface between Designer and EIM Ontology 1 in Loop 1 (Fig. 2). First we describe (1) how User Networks (UNs) and 3D voxel space are established in CAD by the designer based on the design brief, local building and planning regulations, design intentions, and site analysis to provide input into the ontology as constraints in the initialization of the algorithmic process. Secondly, we describe how the designer queries the ontology according to the selection and distribution criteria in the instantiation and spatialization of nodes and edges that make up the 3D Networks. Thirdly, we describe how

CAD Networks, non-spatial data and relations are visualised in CAD. Finally, we describe (4) how the designer implements rule-based algorithm (ASP) and other methods in the selection of the best network alternative in the iterative process.

The concept of interaction between a designer and ontology 1 is described according to different possible use cases, which have an influence on the type of query and user feedback.

1. Initiation: the interaction between designer and EIM Ontology 1 starts with the initiation use case, in which the designer gives input to EIM Ontology 1 by specifying design brief and site-specific data. The role of the user network is to provide input to generate a project specific knowledge graph. From the user network, the existing site conditions such as buildings, vegetation and environmental conditions which are specified in the design brief can be provided as an input. Beside the user network, other input is provided through Rhino (e.g., design space constraints, bounding box) and from the SQL database (environmental maps) to initiate the generation of a project specific knowledge graph in GraphDB.

For the user input, three categories of nodes were defined, namely 1) “EcoNodes” which capture ecological related functions, 2) “ArchiNodes” which capture architectural related functions, and 3) “EnviNodes” which capture additional information related to environmental conditions. The “EnviNodes” provide additional information for “EcoNodes” and “ArchiNodes” and therefore are added to one of those nodes in the user network.

The tables 5-7 show an overview of specified features for “EcoNodes”, “ArchiNodes” and “EnviNodes”. In the table, “Input Type” indicates in which form input can be provided. The “Input Options” specify predefined values, which must be selected by the user for certain features. The specified options, for the “Input Type = check list”, serve as a starting point and can be extended. “Input Provider” determines, if the input is provided only by the user or if a query of the Ontology must be conducted in combination with the user input.

Table 5: Overview of ArchiNode features

ArchiNode Features	Description	Input Type	Input Options	Input Provider
Label	Name Tag given by the user	Text Panel	-	User
Location	Location to spatially reference node, could be center point/voxel	Point (X,Y,Z)	-	User
Area	Area / Radius of the spatial expansion of a function	Number / Slider	-	User
Exposition	Determination about the level of exposition of a function	Drop Down List	• open space • transitional space • enclosed space	User
Existence	Determination, if function is already existing or if it is part of a new planning	Drop Down List	• existing • new	User

Table 6: Overview of EcoNode features

ArchiNode Features	Description	Input Type	Input Options	Input Provider
Label	Name Tag given by the user	Text Panel	-	User
Location	Location to spatially reference node, could be center point/voxel	Point (X,Y,Z)	-	User
Area	Area / Radius of the spatial expansion of a function	Number / Slider	-	User
Exposition	Determination about the level of exposition of a function	Drop Down List	• open space • transitional space • enclosed space	User
Existence	Determination, if function is already existing or if it is part of a new planning	Drop Down List	• existing • new	User
Stakeholder	Determination, which stakeholder is targeted by the function	Drop Down List	• animal • vegetation • human	User
Resources	Determination, which resources and structures are provided by the function	Check List	• Water • Meadow • Hedge • Woods • Gravel • Flowers • Berries	User

Table 7: Overview of EnviNode features

ArchiNode Features	Description	Input Type	Input Options	Input Provider
Label	*Name Tag given by the user*	*Text Panel*	*applied from Eco/ArchiNode*	*User*
Location	*Location to spatially reference node, could be center point/voxel*	*Point (X,Y,Z)*	*applied from Eco/ArchiNode*	*User*
Area	*Area / Radius of the spatial expansion of a function*	*Number / Slider*	*applied from Eco/ArchiNode*	*User*
Causation	Description of environmental conditions which are caused by the function	Check List	• Sun • Shade • Artificial Light • Wind Protection • Noise • Open Water Source • Sealed Surface • Unsealed Surface	User
Requirement	Description of environmental conditions which are required by the function	Check List	• Sun • Shade • Artificial Light • Wind Protection • Noise • Open Water Source • Sealed Surface • Unsealed Surface	User
Avoidance	Determination, which stakeholder is targeted by the function	Check List	• Sun • Shade • Artificial Light • Wind Protection • Noise • Open Water Source • Sealed Surface • Unsealed Surface	User
Flexibility	Determination of the spatial flexibility of a function	Drop Down List	• fix location • dynamic location • no location	User
Status	Determination of the environmental protection status of a function	Drop Down List	• protected • endangered • highly endangered • endangered by extinction	User

Fig. 13-15 show example inputs in the GH interface for the node types “ArchiNode” (Fig.13), “EcoNode” (Fig. 14) and the combination of an “EcoNode” and an “EnviNode” (Fig. 15).

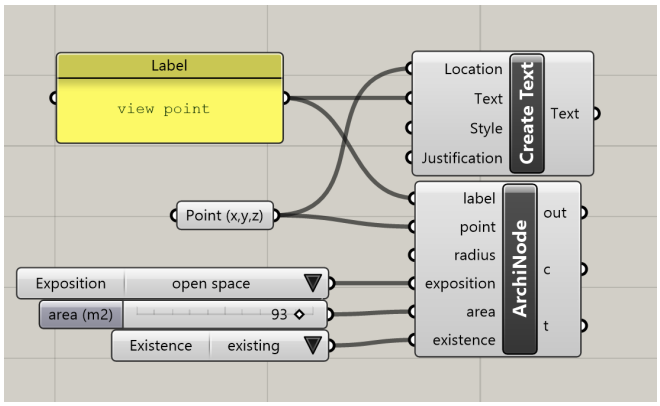

Fig. 13: Use Case "Initiation": Example Input of an ArchiNode

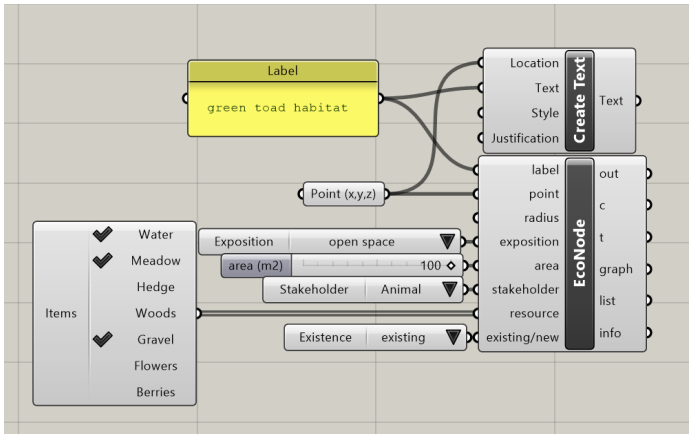

Fig. 14: Use Case "Initiation": Example Input of an EcoNode

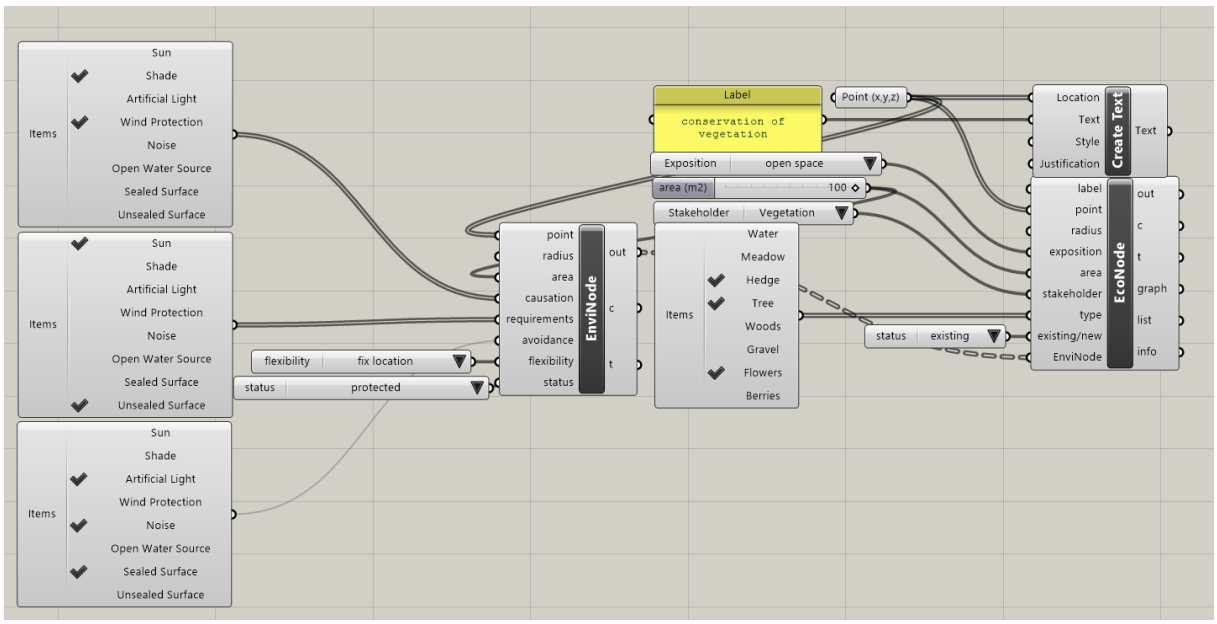

Fig. 15: Use Case "Initiation": Example Input of an EcoNode with an associated EnviNode, which serves as an "Add-On" to provide additional information about the environmental conditions which are required or caused by the functions. An "EnviNode" can be added to an

"EcoNode" as well as an "ArchiNode" and adopts their user input (such as location, area, label,etc.).

In Fig. 16 the user input provided in the GH interface is shown in the Rhino view. To demonstrate the site context, the landscape plan of the Vienna Case Study site "Freie Mitte Nordbahnhof" is used as the background layer.

Fig. 16: Use Case Initiation: Screenshot of user defined existing nodes in CAD, example Vienna Case Study

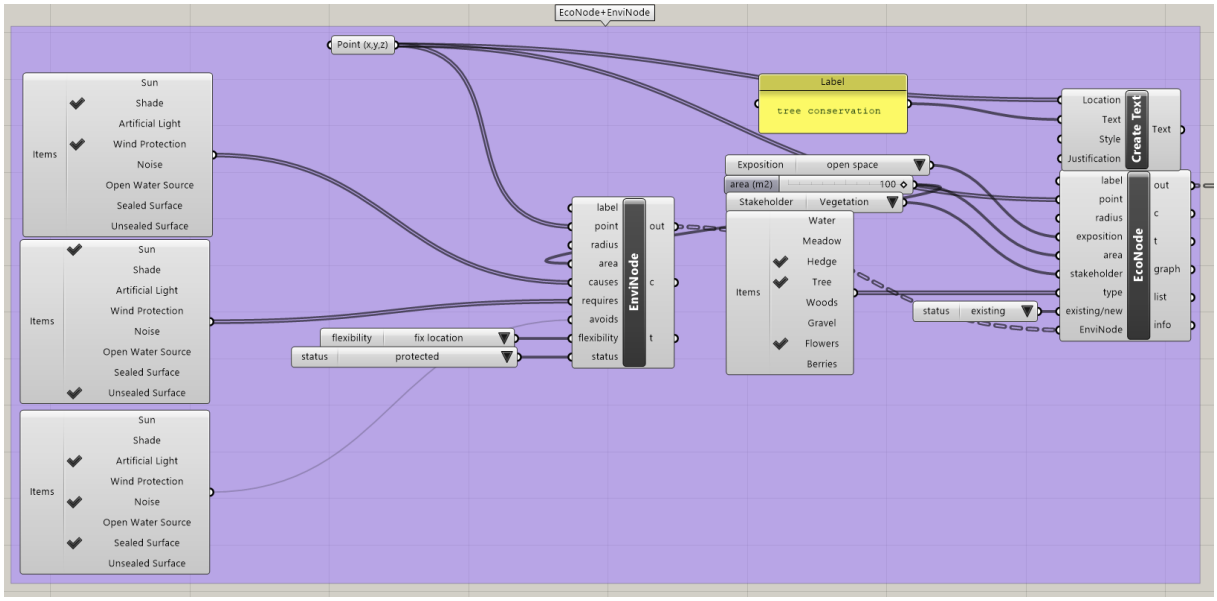

Fig. 17: Use Case "Initiation": Sketch of GH interface for defining existing nodes as a combination of "EnviNodes" and "EcoNodes"

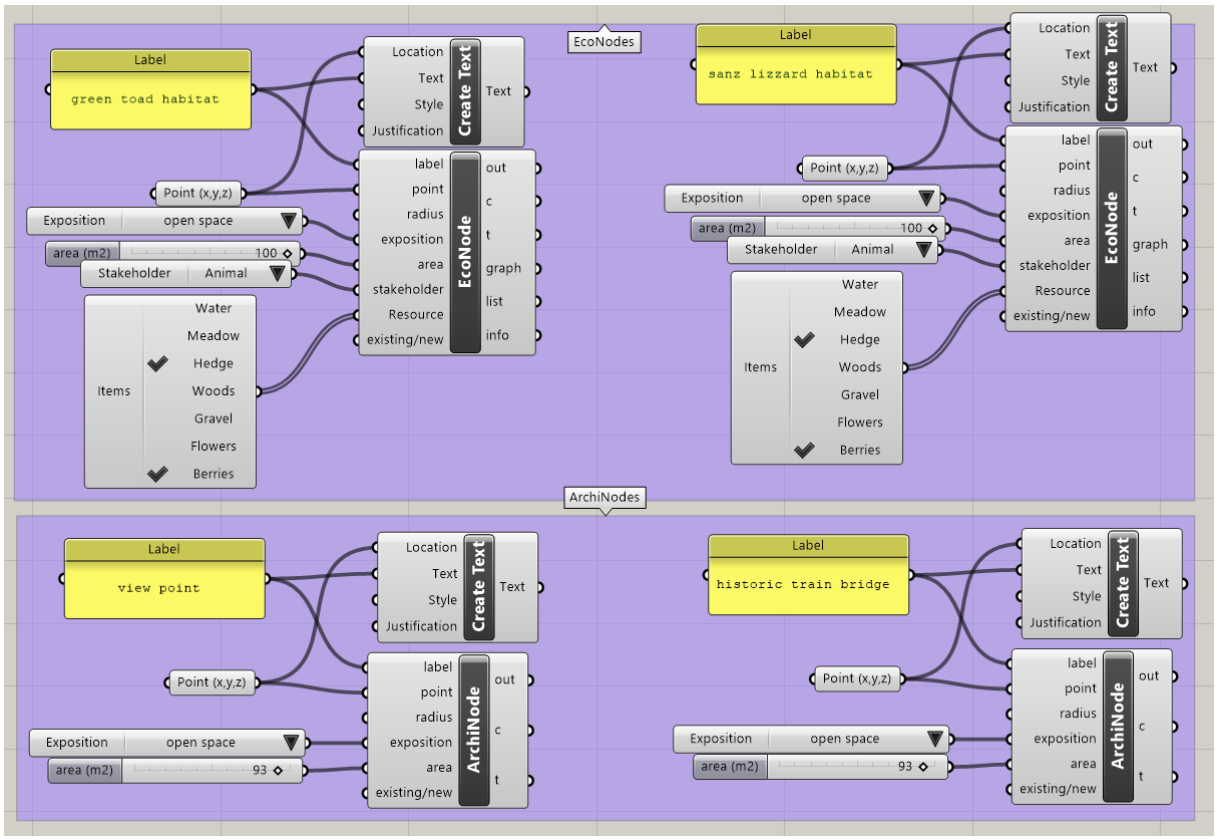

Fig. 18: Use Case "Initiation": Sketch of GH interface for defining existing nodes as "ArchiNodes" and "EcoNodes"

2. Show Knowledge Graph (Fig. 19): in this use case the designer wants to see general relations between items of interest, which are not spatialized on a site. The designer can query EIM Ontology 1 by selecting and filtering items of interest from a predefined list. The user feedback can be visualised through a non-spatial knowledge graph with a textual explanation in addition. Given a set of URIs that represent items of interest, a query is generated and posed against GraphDB's SPARQL endpoint, returning all the relations that connect the items of interest. In this case, only /sparql endpoint service is used. The verbalization to text is also returned by taking the labels of each URI in subject and object positions, and the property's label in the predicate position.

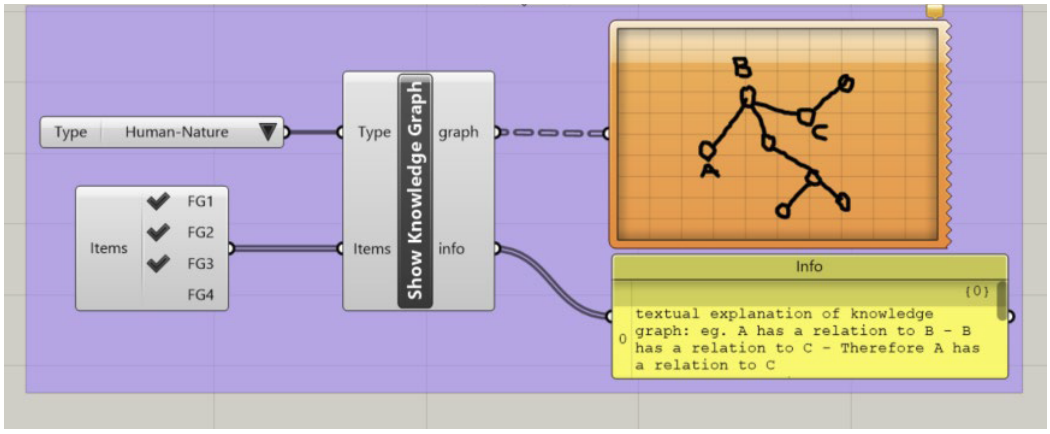

Fig. 19: Use Case "Show Knowledge Graph": Sketch of GH interface for giving user input to receive a knowledge graph

3. Complete Network (Fig. 20): in this use case the designer wants to receive feedback on missing items in the user network to understand if the required resources are available. By comparing the items in the user network with items in the knowledge graph, the missing items can be visualised graphically, shown as a list, and in addition explained as a text for the user. After receiving feedback, the user can add the missing items to the network. In this case, each item of interest is checked against all the relations that it contains and that are returned to the designer. Also, in this case only the /sparql endpoint service is used.

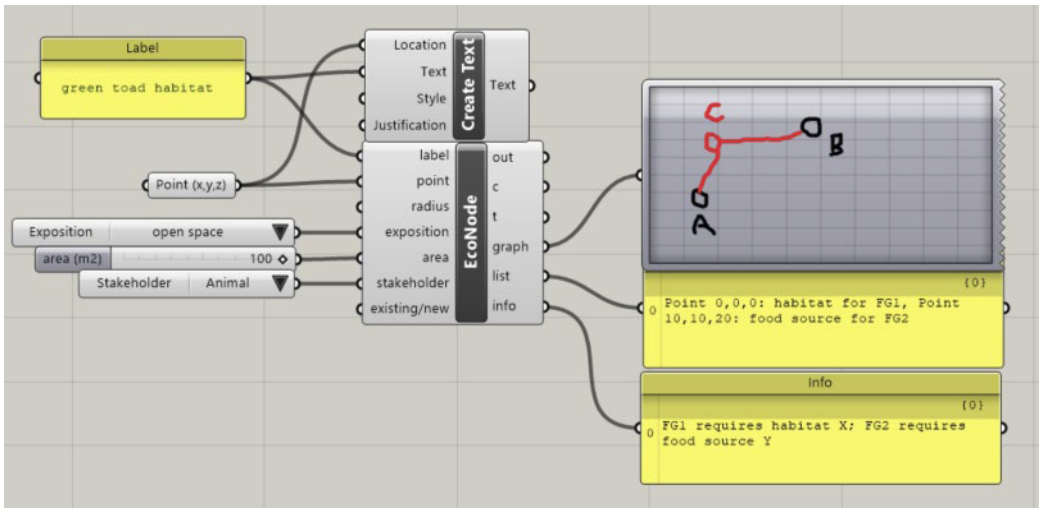

Fig. 20: Use Case "Complete Network": Sketch of GH interface for receiving feedback on missing items in the user network.

4. Conflict Check (Fig. 21): in this use case the designer wants to know if there is a conflict between the defined nodes and check the suitability of the location with microclimatic conditions. Like use case 2, the user can receive graphical and textual feedback. After receiving feedback, the user can modify, move, or remove conflict items. In this case, for given items of interest the query to check for conflicts in the GraphDB's endpoint might not be sufficient, given that different stakeholders' constraints might be involved. Hence /asp endpoint is used to check for conflicts and report back. In ASP priorities can be encoded such that one option can have higher 'weight' and override other options. This option is used when the decision must be made in an automated way, whereas the option without the conflict resolution leaves it open to the designer to resolve the conflict.

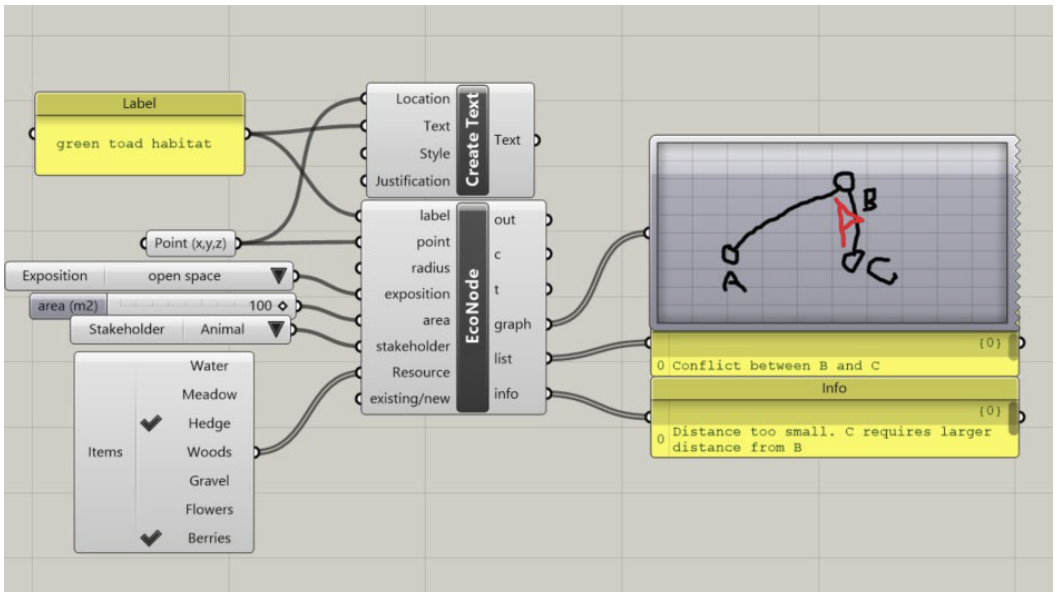

Fig. 21: Use Case "Conflict Check": Sketch of GH interface for highlighting conflicts between defined nodes in the network.

The use cases can be understood as a step-by-step process to provide user input and receive feedback. Use Case 1 "Initiation" starts with a basic user input, which can be extended and specified for the following cases. Depending on the use case, the interaction between the different components might change. Figure 22 shows the same user network example as used to describe the components above in a diagrammatic way. In Use Case 1 the user classifies specific functions and their relations. (It is necessary to point out that the conceptual development of relations in the user network has not been completed yet. For illustration purposes, however, we distinguish between two categories of relations: "spatial connection" and "sight line".) The input provided in case is used by EIM Ontology 1 to generate project specific knowledge graphs in Use Case 2. It might be necessary to provide additional user input, for instance, in Use Case 1 the provided input for EcoNode "vegetation conservation" describes the function in general terms. Therefore, in Use Case 2 it is necessary to provide additional information such as an indication of the species that are part of the "vegetation conservation". A knowledge graph can only be generated for nodes in the user network which have a corresponding entry in Ontology 1. In the example shown in figure 20 there is no corresponding entry for the ArchiNodes "viewpoint" and "historic bridge" and therefore they are greyed out. As in Case 3 and 4 the relation between ArchiNodes and EcoNodes is important, missing corresponding entries can be bypassed by providing additional information in a generalised way. This can be done by specifying the resources that can be provided by an ArchiNode. The example in figure 18 for Use Case 3 shows that the ArchiNode "viewpoint" can provide a sunlit surface and a wooden deck, if there is a spatial connection to the EcoNode "green toad biotope".

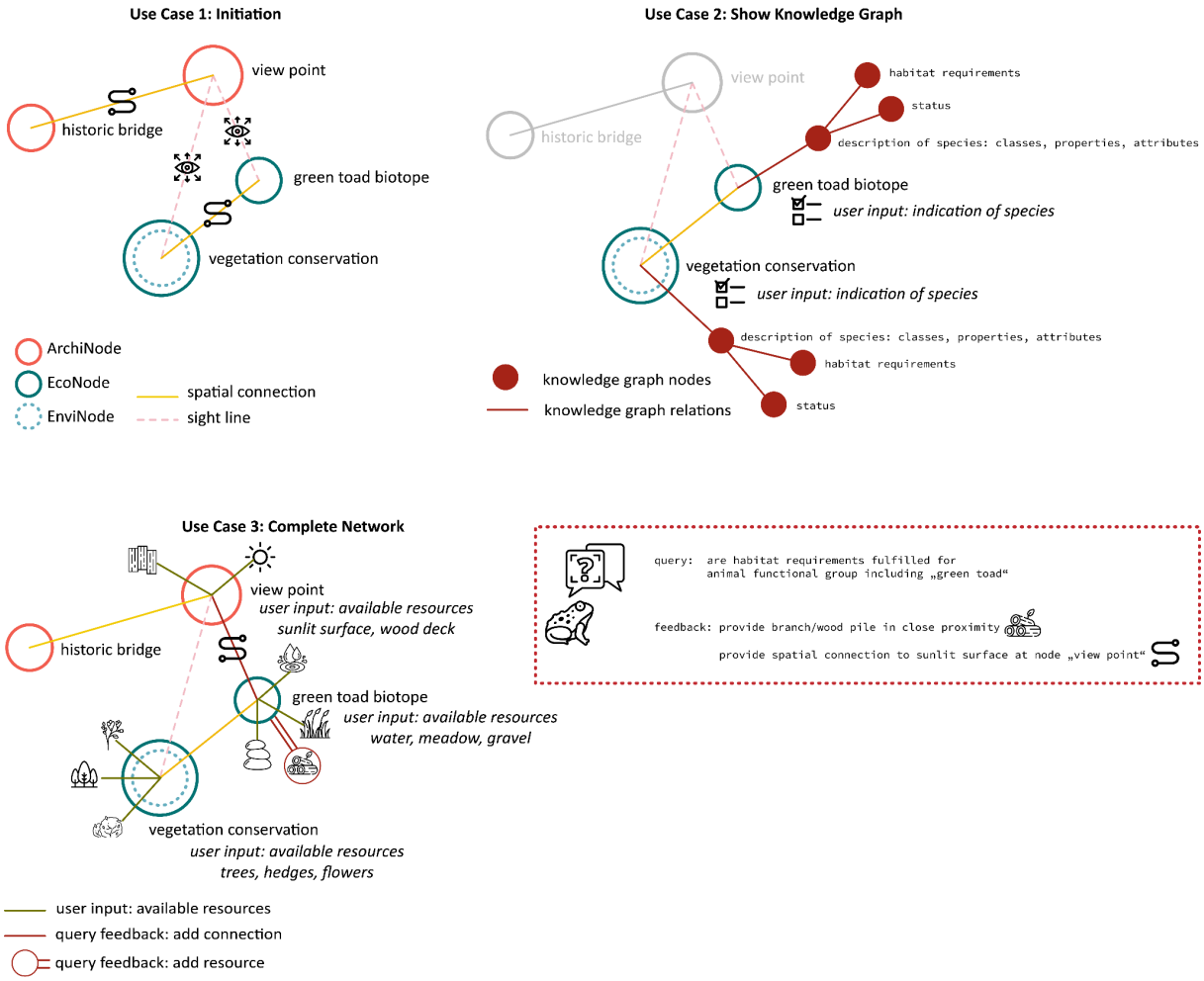

Fig 22: Diagrammatic illustration of Use Case 1, Use Case 2 and Use Case 3

4.1.8 CAD Model 1 and Algorithm 1 and EIM Ontology 1: CAD1→A1→O1

This section provides an overview of the interface between CAD Model 1, Algorithm 1, and EIM Ontology 1 in Loop 1 (Fig. 2). The interface serves two main purposes: (1) it facilitates CAD Model 1 input, including voxel data used to initialise the translational process and reasoning over the Knowledge Graph of ENs, and (2) it enables designer feedback in each design iteration based on the 3D configuring Networks solutions, which integrate ENs and UNs and are derived from EIM Ontology 1-driven rule-based algorithmic procedures.

We will be using a sequence diagram (Fig. 20 Designer (D) and EIM Ontology 1) to illustrate the interactions and flow of messages between the designer, Knowledge Graph (KG) component, and the algorithm component. The diagram begins with the designer initiating the process by requesting information or performing an action related to the algorithm. The KG component, responsible for managing the ontology data, receives the request and processes it by querying the ontology and retrieving relevant data.
Algorithm 1 is a computational algorithm employed in the generative design process. ASP represents complex constraints and provides a systematic approach to finding feasible design solutions within the specified design space. The following lines of code declare three constants representing the required number of networks for each class of material in an *ecolope*. The

integration of CAD Model 1 - Algorithm 1 (ASP) and EIM Ontology 1 empowers the generative design process with computational capabilities from answer set programming and semantic reasoning provided by the EIM ontology. This integration allows for the exploration and generation of design solutions that meet constraints and requirements while leveraging the knowledge base and reasoning capabilities of EIM Ontology 1.

4.1.9 EIM Ontology 1 and Voxel Model: O1→V

EIM Ontology 1 (O1) supports the designer in configuring the user network to initiate the ontology-aided design process. In this context the role of the ECOLOPES Voxel Model (V) is to provide data describing the local environment (dataset maps) to supplement the data input by the designer in the process of configuring the user network (dataset networks). For this reason, ECOLOPES Voxel Model contains a collection of multi-temporal datasets describing local environmental conditions, generated with geospatial analysis and simulation. Design of the ECLOPES Voxel Model component is open-ended and the decision on the inclusion of a particular dataset can be made by the designer, informed by the availability of a certain data and its perceived role in the design process. Currently, datasets describing solar exposure (sunlight hours), wind exposure (exposure towards wind flux) and tendence of the terrain geometry to accumulate water (Topographic Wetness Index) has been inserted into the ECOLOPES Voxel Model (V) to provide representative datasets that could be utilised in the ontology-aided, generative design process.

The databases that are part of Voxel Model (V) are made part of the KG. This is done by translating the data in RDB data model to Knowledge Graph data model, that is from relational model to graph model. For this, we leverage a set of mappings that are used to convert the relational model to a graph model. These mappings are used to overcome the “impedance mismatch” when translating from one data model to another. In these mapping we specify how each column of RDB is mapped to a property that is specified in EIM Ontology 1 of our KG. In addition, we map a table name to a class in EIM Ontology 1.

The mappings are run in the virtualized mode. In result, the data is not copied in graph data, but instead it is being “virtualized.” This method always provides the up-to-date answers to queries, as the data does not need to be synced when it is out of sync (underlying data has changed) as it is always being relied upon the data that exists on RDB that is queried on-the-fly using EIM Ontology 1. The queries (in the language of SPARQL) created by using classes and properties of EIM Ontology 1, are translated on-the-fly to SQL that is used to access the data and return answers.

The integration of the ECOLOPES Voxel Model and the EIM Ontology is facilitated through the Ontop Virtualization technology, which is integrated in the GraphDB software solution. This technique is widely applied in commercial applications of GraphDB technology where the graph reasoning capabilities need to be utilised on pre-existing data stored in an RDB environment. In such cases, either it is too expensive to move the data, data will be inherently duplicated and out of sync, additional storage is required to persist the data. In the scientific literature, namely in the field of data integration, this paradigm is called Ontology-based Data Access (OBDA). This data integration framework is often used when the underlying data changes frequently.

In such cases, the effort of data migration from RDB to GDB is minimised by the application of the SQL virtualization technique. The main task in such a virtualization-based workflow is related to the description of how the tables, columns and primary keys in the RDB map onto graph structure in the GDB environment. On the technological level, such mapping is declared as a set of ODBA/R2RML mappings, written in a single file. This file (Fig. 23) is uploaded into the GraphDB instance when the RDB/GDB connection is created. This process has been tested to enable the integration between the RDB-based voxel model and EIM Ontology stored in the GraphDB environment.

```
[PrefixDeclaration]
:           http://resource.dap.tuwien.ac.at
schema:     http://schema.dap.tuwien.ac.at#
ex:         http://example.org/
owl:        http://www.w3.org/2002/07/owl#
rdf:        http://www.w3.org/1999/02/22-rdf-syntax-ns#
xml:        http://www.w3.org/XML/1998/namespace
xsd:        http://www.w3.org/2001/XMLSchema#
foaf:       http://xmlns.com/foaf/0.1/
obda:       https://w3id.org/obda/vocabulary#
rdfs:       http://www.w3.org/2000/01/rdf-schema#

[MappingDeclaration] @collection [[

mappingId   lvl30
target      :/vox/{vox_idx} a schema:Voxel ; schema:x {vox_x}^^xsd:integer  ; schema:y {vox_y}^^xsd:integer  ; schema:z {vox_z}^^xsd:integer  ;
schema:zCont {vox_z_cont}^^xsd:decimal  ; schema:isNew {is_new}^^xsd:integer  ; schema:cls {cls}^^xsd:integer  ;
schema:red {red}^^xsd:integer  ; schema:green {green}^^xsd:integer; schema:blue {blue}^^xsd:integer  ;
schema:slope {slo}^^xsd:integer  ; schema:aspect {vox_z}^^xsd:integer  .
source      SELECT * FROM vox_lvl30;

mappingId   lvl40
target      :/vox/{vox_idx} a schema:Voxel ; schema:x {vox_x}^^xsd:integer  ; schema:y {vox_y}^^xsd:integer  ; schema:z {vox_z}^^xsd:integer  ;
schema:zCont {vox_z_cont}^^xsd:decimal  ; schema:isNew {is_new}^^xsd:integer  ; schema:cls {cls}^^xsd:integer  ;
schema:red {red}^^xsd:integer  ; schema:green {green}^^xsd:integer; schema:blue {blue}^^xsd:integer  ;
schema:slope {slo}^^xsd:integer  ; schema:aspect {vox_z}^^xsd:integer  .
source      SELECT * FROM vox_lvl40;


]]
```

Fig. 23: Data saved in the RDB-based voxel model can be virtualized in GraphDB, by defining the mapping between the RDB and GDB data structure in an OBDA / R2RML file.

Initial tests showed that for performance reasons file-based SQL databases, such as SQLite, should be avoided. PostgreSQL and GraphDB server instances hosted on the same machine have shown sufficient performance to execute the operations required for the integration between the ECOLOPES Voxel Model and the EIM Ontology. Currently, each voxel model level contained in the RDB can be interactively linked with the GraphDB instance, based on the provided OBDA/R2RML mappings. As a result, the data contained in the RDB-based voxel model can be transparently queried and reasoned using techniques implemented in the GDB environment. SPARQL query is translated to SQL query on-the-fly and the results are returned that are described using the ontological properties (Fig. 24).

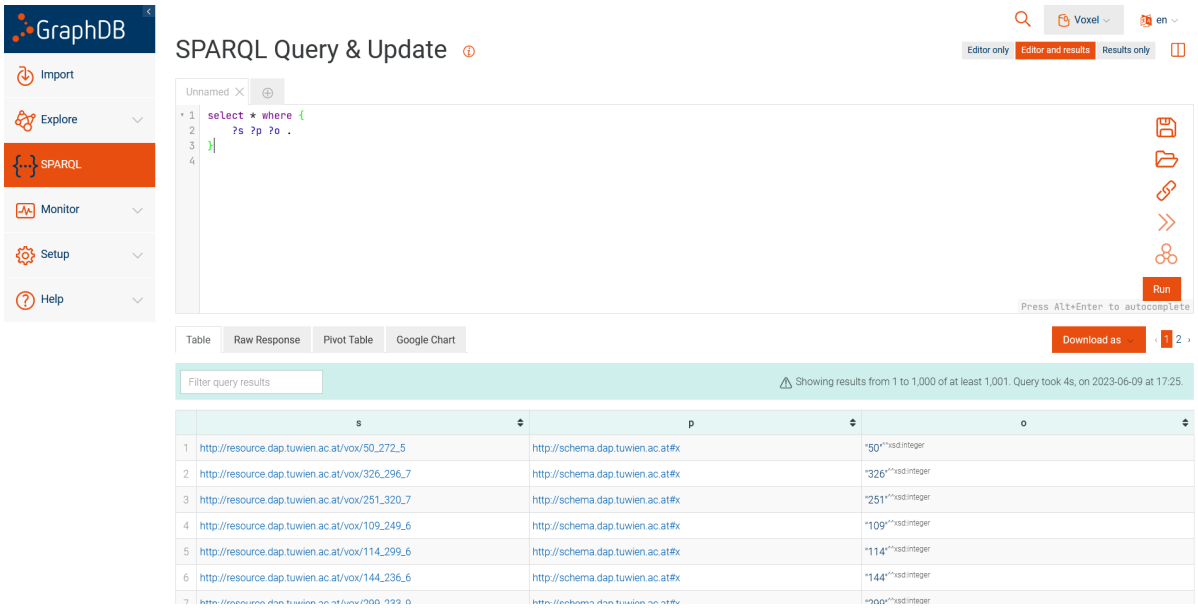

Fig. 24: Screenshot from the GraphDB interface, showcasing how data contained in the ECOLOPES Voxel Model can be queried and represented in ontology-based format (RDF triples).

An initial test-run of a combined query, utilising EIM Ontology 1 and ECOLOPES Voxel Model data was successfully executed. At that stage ECOLOPES Voxel Model data was available in the namespace of the local GraphDB instance. Dummy data describing three types of plants and their solar exposure requirements has been added in a separate repository in GraphDB. Requirements related to the minimum and maximum amount of sunlight hours have been assigned to plant types named: aSunnyPlant, aHalfShadyPlant and aShadyPlant. As a result, the user can query the plants dataset, link the ECOLOPES Voxel Model data and get a result describing the plant requirements represented in a format that is compatible with the structure of the ECOLOPES Voxel Model. Example of such a query for a plant named aHalfShadyPlant is presented in Figure 20a below. Definition of SPARQL queries requires careful consideration because the structure of SPARQL query might have significant impact on the performance and resulting user experience. First approach implemented for the initial test-run was utilising the FILTER procedure within the main GraphDB repository (*plants*). In result, the whole voxel dataset had to be pulled from the RDB (PostgreSQL) and converted into RDB triples, before it could be filtered inside GraphDB. For example, a query of the voxel model representative of 1 km² (vox_lvl40) for locations suitable for *aShadyPlant* took 7 min and 36 seconds. To address this issue, the SPARQL query has been adapted to execute the FILTER operation inside the SERVICE federation query, which filters the data before it is fully retrieved into the GraphDB instance. As a result, the same query took only 27 seconds to complete.

Fig. 25: Exemplary SPARQL query utilising GraphDB internal SPARQL federation functionality to query a dummy plant dataset and the ECOLOPES Voxel Model data, based on a chosen plant requirements related to sunlight exposure.

The property related to the structure of data being aligned with the structure of the ECOLOPES Voxel Model allows further integration with the components implemented within WP5. Interaction between the ECOLOPES Voxel Model and the McNeel Grasshopper is facilitated through the McNeel Hops technology. This technological approach was extended to enable interaction with the GraphDB environment and to implement a temporary Hops component that executes the SPARQL query presented in the upper right side of Figure 26 below. This temporary component, previewed in the upper left side of Figure 26 below, can be used to query the dummy plant data and to retrieve locations which fulfil solar exposure requirements. Both the existing ECOLOPES Voxel Model components and the SPARQL query component can be used simultaneously to produce an interactive visualisation, which shows both the preferred plant locations and the underlying 3D voxel data, represented in real-world colours. The sequence of components required to visualise the coloured 3D voxel data is shown in the lower part of Figure 26 below. 3D visualisations of the ECOLOPES Voxel Model data representing preferred plant locations for the three classes (aSunnyPlant, aHalfShadyPlant and aShadyPlant) are presented in the middle row of Figure 26b below.

The current state of this integration is limited to the initial test described above. In the following steps, a collection of SPARQL queries relevant to the work of WP5 and the EIM Ontology Task would need to be established. Currently, the SPARQL query presented in Figure 25 above is hardcoded inside the Grasshopper component. A systematic approach to store and modify the SPARQL queries based on user input in the Grasshopper environment needs to be conceptualised. For each identified functionality, relevant data in GraphDB would need to be identified. Each functionality would require a dedicated collection of SPARQL queries, which would need to be written and tested in combination with the Grasshopper components already implemented within the specific functionality. Lastly, a dedicated component that returns properly formatted data from GraphDB to Grasshopper interface would need to be

implemented, based on the functionalities of the components implemented for each functionality.

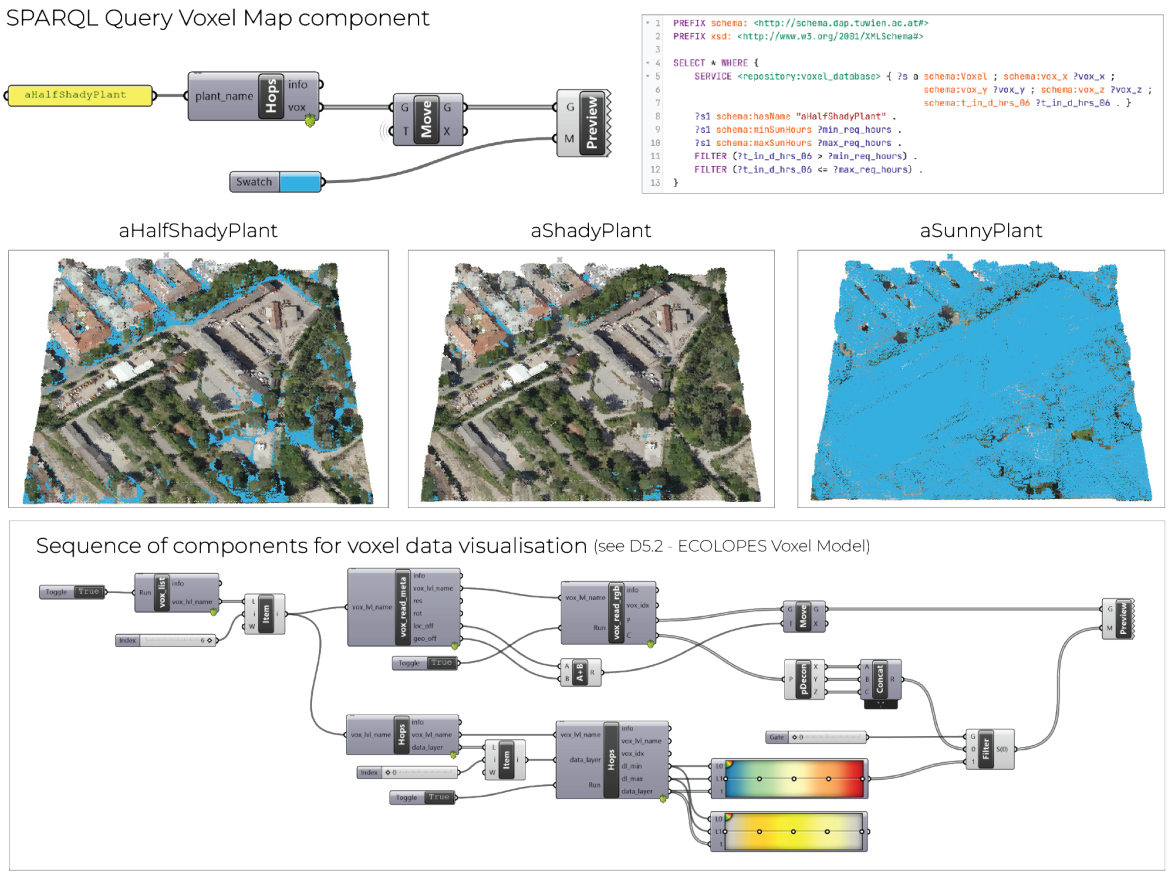

Fig. 26: Initial integration of ECOLOPES Voxel Model components and the SPARQL query functionality provided by GraphDB. The current state of this integration enables execution of SPARQL query which returns locations fulfilling solar exposure requirements for a chosen plant. Data returned by the query is aligned with the structure of the ECOLOPES Voxel Model and can be integrated with the existing components implemented for ECOLOPES Voxel Model visualisation.

4.1.10 **CAD Model 1 and Voxel Model: CAD1→V**

In the first loop of the ontology-aided design process the connection between the CAD Model 1 (CAD 1) and the ECOLOPES Voxel Model (V) is facilitated through the McNeel Rhino 3D interface. In this phase of the process, the designer configures networks in the Rhino 3D software. The outcome of this process is a spatial configuration of the network nodes and their properties, expressed as native Rhino objects. Spatial configuration and internal structure of the network is validated by Ontology 1. The ECOLOPES Voxel Model features a Rhino 3D / Grasshopper interface that allows the designer to interact with the voxel model, querying available datasets that can be interactively visualised in the 3D viewport of the Rhino 3D software (see *D5.2 ECOLOPES Voxel Model*.)

4.1.11 Voxel Model and Ecological Model: V→EM

The development of the ECOLOPES Ecological Model can be found in Part B, D3.3. *Interim ECOLOPES platform architecture,* D1.5 (*Report after 2nd year*) and *D4.1 Preliminary EIM Ontology*. A possible future integration between the ECOLOPES Voxel Model and ECOLOPES Ecological Model is described in *D5.2 ECOLOPES Voxel Model*. According to the descriptions in D4.1, D3.2 and D3.3, the ECOLOPES Ecological Model is implemented as a standalone executable file (though callable from GH in the computational workflow) and its execution parameters are defined in a separate configuration file. The required data inputs consist of (1) a definition of PFGs and AFGs, (2) soil classification data and (3) description of the site geometry, including shading and solid depth properties. All inputs are contained in JSON files, stored in dedicated directories. The geometric description of the site is defined in a voxel-like structure, indexed with a unique key, and computed based on the nodal point coordinates. The procedure of extracting geometric data required for the initiation of the ECOLOPES Ecological Model from the ECOLOPES Voxel Model could be implemented, at the time when the ECOLOPES Ecological Model has reached a higher level of advancement. Such a procedure can consist of an SQL or SPARQL query followed by a purpose-made utility script that maps the data returned by the query with the structure of the ECOLOPES Ecological Model input. Based on the detailed description of the procedure required to calculate the shading parameter, it could be evaluated if the shading parameter can be derived from solar exposure data contained in the ECOLOPES Voxel Model. In the context of the KGF, the soil depth parameter and soil classification data are defined by the designer for the new geometry and a reasonable default value was assumed for the surrounding area. Detailed soil depth and classification data at the required resolution does not exist for the real-world locations that were selected to evaluate the ECOLOPES design approach. For this reason, reasonable defaults would need to be identified and a spatial distribution of soil depths and classes would need to be generated before such data could be integrated with the ECOLOPES Voxel Model. ECOLOPES Ecological Model inputs related to the PFGs and AFGs are currently created based on the functional trait-based classification (Part B, *D4.1 Preliminary EIM Ontology*). Those inputs are lists of PFGs and AFGs, including names and their traits, such as Maturation Time or Light Tolerance (example from PFG data). Since this data does not have a spatial representation, it will not be integrated in the ECOLOPES Voxel Model.

4.2 Interfaces of Loop 2

In this subsection we are going to describe the interface and interaction between EIM Ontology 2 and Algorithm 2, as well as their interaction with the CAD model separately. Furthermore, we also elaborate on the designer input with respect to EIM Ontology 2.

4.2.1 EIM Ontology 2 and Algorithm 2: O2→A2

This interface is the mediator between EIM Ontology 2 and Algorithm 2. In EIM Ontology 2 we have properties such as “volume” that describe each volume regarding whether it has occupied one of the values: air, architecture, soil, or biomass.

The data from EIM Ontology 2 that is derived from its SPARQL endpoint is fed to Algorithm 2 that generates volume distribution. In case the constraints are not satisfied by ASP then it is

necessary to either repair the distribution of volumes, or to re-generate the volumes from scratch. Repairing the distribution of volumes in a minimal way is not always possible or can lead to ambiguous solutions, e.g., remove/add volumeX or remove/add volumeY, where a designer has to choose between the two while only knowing the impact of this decision further down the line.

4.2.2 EIM Ontology 2 and Algorithm 2 and CAD Model 2: O2→A2→CAD2

The O2/A2/CAD2 interface uses the results obtained from O2/A2 and renders the volumes in the CAD interface. Once the ASP algorithm generates a set of instantiations of volumes that satisfy the given constraints, these results are passed on to the CAD interface for visualisation and further design refinement. To achieve this, an intermediate component called the Hops component is employed. The Hops component is responsible for reading the results in JSON format, which represent the instantiated volumes, and then rendering them within the CAD interface. By utilising the Hops component to bridge the gap between the output of the ASP algorithm and the CAD interface, designers can visualise the generated volumes and assess their feasibility. This interface enables designers to interact with the volumes, make modifications, and refine the design. The O2/A2/CAD2 interface enhances the design workflow by seamlessly integrating the computational results from O2/A2 into the CAD environment.

4.2.3 EIM Ontology 2 and Designer: O2→D, D→O2

The designer interacts with EIM Ontology 2 in Loop 2 by providing geometric and quantitative constraints for the volume distribution process. This process is initiated by the definition of (1) a 3D bounding box, (2) resolution (edge length of a single volume), (3) exclusion zones where no volumes can be placed, and (4) the total amount of volumes assigned to four types (air, architecture, soil, and biomass). The values assigned to the constraints are informed by the design brief and existing site constraints. For example, the total count of volumes is informed by the expected total usable area defined by the masterplan for a given site. Data inputs derived from the constraints are created in the Rhinoceros 3D and Grasshopper environment. Currently, they are written as a collection of JSON files. In the next steps, this data can be linked with GraphDB in a more direct way.

4.2.4 CAD Model 2 and Algorithm 2 and EIM Ontology 2: CAD2→A2→O2

The CAD2/A2/O2 interface implies incorporating EIM Ontology 2 and Algorithm2 to facilitate the first stage of the design process in the CAD environment. CAD Model 2, Algorithm 2, and EIM Ontology 2 work together to enable computational design through the application of genetic algorithms and ontological reasoning. CAD Model 2 is an advanced version that digitally captures the design solution within a CAD environment, encompassing both geometric and parametric aspects, such as architectural, biomass, and soil volumes. Designers can create and manipulate this model using CAD software like Rhino 3D Grasshopper.

EIM Ontology 2 extends the ECOLOPES Information Model (EIM) ontology and serves as a knowledge representation framework. It captures semantic relationships in the design domain and incorporates domain-specific knowledge and rules for advanced reasoning during

the generative design process. The provided Turtle code snippet defines ontology classes, properties, and rules for EIM Ontology 2, creating a structure to describe relationships and concepts within the knowledge domain.

By integrating CAD Model 2, Algorithm 2, and EIM Ontology 2, the generative design process becomes a computational iterative workflow. It harnesses ontological reasoning and optimization techniques to generate and evaluate diverse design solutions based on ecological and architectural criteria. This integrated approach empowers designers to explore a wide range of design possibilities and make informed decisions during the design iteration process.

4.2.5 EIM Ontology 2 and Voxel Model: O2→V

Data can be accessed from the GraphDB's SPARQL endpoint to be queried and fetched to Algorithm 2 in order to start the process of reasoning using ASP rules. To facilitate the volume distribution step of the GCD process, a site-specific coordinate space is introduced in the voxel model (see *D5.2 ECOLOPES Voxel Model*). The resolution of this 3D space is matched with the geometric constraints of the volume distribution process (size of a single volume, rotation, and dimensions of the site, where the volumes need to be placed). Data contained in the ECOLOPES voxel model is reprojected from the urban-scale coordinate space into the site-specific coordinate system. The reprojection procedure is implemented in the RDB (PostgreSQL) and consists of a composite transformation followed by the data aggregation step (see *D5.2 ECOLOPES Voxel Model*). In result, ECOLOPES Voxel Model data originally expressed in the urban-scale coordinate space is available in the site-specific coordinate system and can be queried by EIM Ontology 2 and further converted to a representation compatible with the ASP-based volume distribution algorithm (Algorithm 2). Spatial datasets, such as environmental analysis and simulation data, are contained in the ECOLOPES Voxel Model based on the urban-scale coordinate space definition. The reprojection process integrated within the RDB system can map any dataset from the urban scale to the site-specific coordinate space and later into EIM Ontology 2 (O2). This process requires manual definition of the mapping constraints inside the RDB system (`PostgreSQL`) and in the Ontop virtualization interface (ODBC file). It is worth noting that the number of datasets chosen impacts the computational performance of this process. Datasets can be independently reprojected to multiple resolutions, which exposes multi-scalar data to EIM Ontology 2. In the example of the case study, resolutions of 3, 6, 9 and 12 meters have been chosen to prototype the volume distribution algorithm (see *D5.2 ECOLOPES Voxel Model*). Currently, a multi-temporal dataset describing the urban-scale solar performance for a representative year has been mapped onto the site-specific coordinate space. In result, solar exposure (sunlight hours) for each month of the year can be queried for each location in the site-scale coordinate space by EIM Ontology 2.

Different types of volumes (architectural, biomass, and soil volumes) are distributed in this process. Subsequently, these volumes are checked to ascertain that they satisfy the criteria encoded in ASP. For this purpose, we extend Moore's coordinate system to reason with a focus on the 4 directly neighbouring directions (north, south, east, west) and the 4 diagonals (northeast, northwest, southwest, southeast). This approach enables making determinations about permissible and non-permissible proximities of volumes according to specified criteria and rules.

4.2.6 CAD Model 2 and Voxel Model: CAD2→V

The connection between the CAD model within the Loop 2 is indirectly facilitated through EIM Ontology 2, which aids the process of volume distribution. In this process, the ASP algorithm and EIM Ontology 2 have direct access to data stored in the voxel model by the utilisation of SQL virtualisation method. Moreover, voxel data available in the levels introduced for the operation of EIM Ontology 2 (vox_lvl30_3 etc.) can be utilised in the CAD environment. The interface developed for the visualisation and interaction with the voxel model as a part of the task T5.1 (see *D5.2 ECOLOPES Voxel Model*) enables designers to inspect the information utilised by EIM Ontology 2 and Algorithm 2 in Loop 2. This data can be visualised in the Rhinoceros interface and the results of the volume distribution can be graphically compared with the underlying voxel data that informs the distribution process.

4.3 Interfaces of Loop 3

Loop 3 facilitates the generation of geometric articulation (dataset *landform*) for each case-specific design output. First the design objectives, constraints, and criteria that guide the generative process are defined. This step also includes specifying performance metrics related to ecological and architectural criteria. EIM Ontology 3 is developed to generate query results for Competency Questions related to the geometry articulation task according to specific criteria established in Loop 3, and to enable the implementation of rules inferred from the ontology to aid the iterative generation of CAD geometry in Loop 3. The selected type of algorithm then needs to generate a set of design solutions that satisfy the set criteria. This involves manipulating the defined parameters and constraints to generate a range of design alternatives. (*D5.3 ECOLOPES Computational Model*)

One interface in the larger workflow is the one between the KGF and EIM Ontology 3 (Fig. 2: KGF→O3). This interface requires further discussion across the involved WPs, and subsequently further development.

In the ontology-aided generative computational design workflow there are the linked interfaces between EIM Ontology 3, and the algorithm employed in Loop 3 (Fig. 2: O3→A3), and the two-way interface between EIM Ontology and Designer (Fig. 2: O3→D, D→O3).

Designer inputs from Loop 1 and Loop 2 indirectly inform the processes aided by EIM Ontology 3. Volume distribution resulting from the algorithmic procedure implemented in Loop 2 is used as an input into this process. Geometric constraints, such as the 3D bounding box and exclusion zones are provided by the designer as inputs for EIM Ontology 3. The components and algorithmic approaches implemented in Loop 1 and in Loop 2 are utilised to support interactions between the designer and EIM Ontology 3.

In Loop 3 an Answer Set Programming (ASP) algorithm is implemented with which the EIM Ontology interfaces. The output of Loop 3 are variants of geometric articulation in CAD, variant specific ECOLOPES Voxel Model data, and ontological output.

The computational process for the landform generation results are geometric representations that are based on designers input and supported by ontological reasoning and data introduced

into the GCD process. The outcomes of this process are materialised as static geometry in the Rhinoceros 3D CAD interface and converted into a geometric representation compatible with the ECOLOPES Voxel Model. This 3D CAD geometry constitutes a new starting condition for the design process executed by the components positioned later in the ECOLOPES Computational Design Workflow. At the same time, a matching voxel-based representation containing domain-specific, environmental, and ecological data needs to be provided. Voxel-based representation of the geometry resulting from the GCD process will be merged with the spatial data describing the immediate surroundings of the site (D5.2 ECOLOPPES Voxel Model, vox_lvl40 data used in the Loop 1). Information content of this merged dataset will be extended by the application of the diverse analysis tools implemented within the ECOLOPES workflow. Finally, designers will be able to visualise and evaluate the voxel data that results from this process by utilising the components developed as a part of the task T5.1 (D5.2 ECOLOPES Voxel Model).

5. Ontological Output for subsequent Computational Processes

5.1 Connection to the nested hierarchy, objectives and KPIs

Within the scope of the ECOLOPES research, some of these KPIs are computed through the KGF and additional KPIs may also be derived from the outputs of the Ecological model such as plant functional group biomass or the number of animal functional groups (as described in D4.2). This will provide temporal information about species composition of the ecological community modelled on an *ecolope*. Essentially, KPIs can be computed and suggested as a priori additions depending on the defined objectives. For the nested hierarchy / nested sets see *D6.1 Draft KPI Description*.

There are two common KPIs that have been computed in the KGF and have been integrated into the nested set strategy within the optimization component. These KPIs are soil depth and shading percentage. Based on these determinations it is possible to run the computed KPIs in the ontology-aided generative computational process. The form generated by the ontology-aided generative computational design process can potentially become the initial form for the optimization based on the computed KPIs. (see *D6.1 Draft KPI Description Section 4.3.1*).

5.2 Decision Rules

Decision rules play a crucial role in the ECOLOPES framework, particularly in the context of Answer Set Programming (ASP) implementation. ASP-based decision rules are used to guide the generation and evaluation of design solutions for ecological building envelopes. In the ECOLOPES framework, decision rules are defined using logical programming constructs. These rules encode design requirements, constraints, and preferences that need to be satisfied by the generated design solutions. By formulating these rules, designers can explicitly specify the desired characteristics and behaviours of the ecological building envelopes. The decision rules in ASP-based design generation capture various aspects of the design process, such as spatial organisation, and environmental impact. These rules help define the search space of potential design solutions and guide the optimization process towards solutions that satisfy the

specified criteria. The use of decision rules in the ECOLOPES framework allows designers to explore a wide range of design possibilities and systematically evaluate the trade-offs between different design criteria. By iteratively refining and adapting these rules, designers can progressively develop design solutions and identify suitable options for ecological building envelopes. Decision rules, particularly in the context of ASP implementation, provide a powerful mechanism for encoding design requirements, constraints, and preferences in the ECOLOPES framework. They enable designers to generate innovative and sustainable design solutions while considering multiple criteria and ensuring compliance with ecological principles.

5.3 **Algorithmic Implementation in CAD**

For Loop 2 and Loop 3 we develop an ASP algorithm that will be developed to the required TRL. ASP is useful for knowledge representation and reasoning tasks, enabling designers to make well-informed choices, and facilitating the conversion of design requirements and constraints into computationally interpretable data.

Furthermore, we develop on a conceptual level an extension of the ASP algorithm approach, with GA and ML algorithms to pave the way for enhancing in future the capacity to generate design variations in Loop 2 (spatial organisation) and Loop 3 (geometry articulation). GA will be used to provide variety to the solution space, effectively creating a set of solutions that are different from the initial solution space. ML (K-means) will be used to group similar solutions, to make it easier for the designer to recognise the patterns in the solutions, ultimately achieving better informed decisions by reducing the search space into a set of groups.

Algorithmic implementation in CAD incorporates Graph Databases and SQL Databases to handle structured data efficiently. Graph Databases provide a flexible solution for managing interconnected design elements and relationships, allowing designers to analyse complex design networks. SQL Databases offer a structured approach for managing design-related information, ensuring data integrity, and facilitating efficient data management and retrieval. Combining ASP, GA, and K-means algorithms with Graph Databases and SQL Databases, CAD systems enables designers to explore diverse design alternatives and to make informed decisions based on defined criteria. Additionally, the integration of database technologies enhances data accessibility and facilitates efficient data retrieval and analysis, contributing to improved design outcomes.

5.4 **CAD Model Output**

As stated above, the output of the ontology-aided generative computational design process consists for each generated design variant of CAD Model output, related data in the ECOLOPES Voxel Model, as well as ontological output. The CAD Model output comprises (1) variations of spatial organisation, and (2) variations of geometric articulation. Spatial organisation entails the distribution of different types of geometrically generic volumes, while geometric articulation entails defining the geometry of volumes and surfaces according to specified requirements contained in the project brief and those made by the designer during the ontology-aided *translational* process.

Regarding spatial organisation we established at the stage of delivering report *D5.1 Development Process for ECOLOPES Algorithm* three distinct types of volumes: (1) architectural volumes, (2) biomass volumes, and (3) soil volumes that together form stationary spatial features of an *ecolope*. During further development steps it became apparent that these three types of volumes are not enough. At this stage we consider distinguishing between different types of green volumes, e.g., dense biomass and sparse biomass (for instance as corridors for movement). More recently we found it useful to distinguish in further development steps different types of architectural volumes, e.g., fully enclosed space and transitional space, and to assign further attributes such as including openings in the surfaces or not.

Regarding geometric articulation we established at the stage of delivering report *D5.1 Development Process for ECOLOPES Algorithm* that working towards a system of urban landform consisting of specific terrain features is advantageous to exploit the relation between geodiversity, diverse microclimates, and biodiversity. For this purpose, we selected the *geomorphons* approach, a pattern-recognition based approach to classify and map landforms (Jasiewicz and Stepinski 2013). Geomorphons are organised as a library of terrain features (e.g., flat, valley, shoulder, ridge, etc) and are based on a 2.5D definition of the terrain surface. We seek to extend the geomorphon approach with the aim to enable full computational analysis and design of urban form and architectures as continuous terrain. For this reason, we realised that it is disadvantageous to consider terrain features as a set of components or tiles, since the edges of neighbouring geomorphons might not align and therefore not result in a continuous surface. We are currently reconceptualising our approach, based on reverse engineering the analytical process of geomorphons. The process of geometric articulation will therefore commence from a "generic" condition of horizontal and vertical surfaces that are transformed in a hierarchical manner into an urban landform.

As outlined in reports D5.1 *Development Process for ECOLOPES Algorithm* and D5.4 *ECOLOPES Computational Model Validation* we are developing the ontology-aided generative computational design process for two distinct design cases to ensure practice-relevance of the approach, methods and tools that are currently in development.

Design Case 1 entails the design of a master plan for the development of a given site. In such cases the number and distribution of building volumes, including footprint, floor area ratio, maximum volume, and height, are not yet defined. In the context of this research this entails that spatial organisation is generated through the distribution of architectural, biomass and soil volumes, which we term for case 1 *primary volumes*, as well as geometric articulation of site and buildings leading to what we term for case 1 *primary landform*. Landform can therefore be coherently designed across the entire site, with all volumes adhering closely to the landform scheme.

Design Case 2 entails the design of an individual building for which all constraints, such as footprint, floor area ratio, maximum volume, and height, etc. are already established by a municipal master plan. Since the generic maximum allowed primary volume is already given by the masterplan, the task is to partition the primary volume into *secondary* and *tertiary* architectural, biomass and soil volumes. To enable different species to inhabit the envelope it is useful to develop the building geometry as a *secondary* and *tertiary landform* (hierarchical nesting of terrain features) to enable accessibility and appropriate provisions for specified species to specified parts of the building envelope.

Therefore, *primary* volumes define the location of buildings, and overall biomass and soil volumes. On a conceptual level this implies that once *primary* volumes are located it is possible to detail them further by locating *secondary* and *tertiary* volumes, which entail more specific architectural, green and soil volumes. Since the purpose of geometric articulation is to shift from generic (cuboid) geometry to landform with distinct terrain features, a matching hierarchical order is established. *Primary* landform delivers a first overall level of geometric articulation to primary volumes, especially for architectural and soil volumes. *Secondar*y and *tertiary* landforms can subsequently be generated to derive greater geodiversity across scales, which serves to enhance the possibility of meeting diverse ecological and architectural requirements. The primary volume distribution process results in the positioning of generic cuboid volumes on site in the 3D space of Rhinoceros 3D software. In the current state, the rule-based volume distribution process positions primary volumes of 4 classes (air, architecture, biomass, and soil) within the site bounding box. For the empty volumes, no Rhinoceros 3D geometry is generated. For the remaining primary volume classes, generic cuboid geometries are generated in Rhino 3D and previewed with matching colours by utilising the Grasshopper "Preview" component. Finally, the resulting volumes can be "baked" (materialised as static CAD objects) using the Grasshopper "Bake" functionality, and each class assigned to a separate layer in the Rhinoceros 3D software. Currently only the initial specification of the landform generation process has been concluded. At the current stage, it can be assumed that landform geometry will be represented as a continuous surface, created in the Grasshopper environment, and materialised as a static CAD geometry.

6. FURTHER DEVELOPMENT OF THE EIM ONTOLOGIES

To ensure the continuous improvement and effectiveness of the EIM Ontologies, a comprehensive plan for their further development has been outlined.

1. Thorough validation procedures will be conducted to ensure the accuracy, reliability, and relevance of the EIM Ontologies. Expert reviews, domain-specific evaluations, and extensive testing using real-world design scenarios will be employed to validate and refine the EIM Ontologies. This validation process aims to enhance their robustness and ensure they meet the specific requirements of sustainable design practices.

2. The coverage of the EIM Ontologies will be expanded by integrating additional data sources and knowledge domains. This includes incorporating data from environmental databases, simulation outputs, and survey data, among others. By incorporating diverse and comprehensive datasets, the EIM Ontologies will provide designers with a broader understanding of ecological factors, enabling them to generate more informed and ecologically informed design solutions.

3. Continuous improvement and updating of the EIM Ontologies will be prioritised. User feedback, advancements in technology, and emerging design practices will be considered to keep the ontologies up to date and adaptable. By staying current with the latest developments, the EIM Ontologies will remain relevant and effective in supporting designers throughout the generative design process.

4. Collaboration and engagement with experts, researchers, and practitioners in the field will play a crucial role in the further development of the EIM Ontologies. By fostering a collaborative and community-driven approach, insights can be shared, knowledge can be exchanged, and best practices can be identified. This collaboration will help establish a strong ecosystem of users and contributors, ensuring the ongoing improvement and widespread adoption of the EIM Ontologies in the field of sustainable and ecologically informed design. In conclusion, the further development of the EIM Ontologies within the ECOLOPES computational framework focuses on validation, expansion, continuous improvement, and collaboration. Knowledge will be added over time, i.e., new objectives, more detail in the ecological model, adding to the KB and to the EIM Ontology.

6.1 **Validation of the EIM Ontologies in the Generative Computational Design Process**

The validation of the EIM Ontologies within the generative computational design process is an essential step towards ensuring the accuracy and effectiveness of the knowledge representation. The validation process involves a multi-faceted approach that combines expert evaluations, empirical testing, and user feedback to assess the ontology's reliability, relevance, and usability.

Expert evaluations play a pivotal role in validating the ontology by leveraging the expertise of domain specialists and design practitioners. These experts carefully examine the EIM Ontologies´ conceptual framework, ontological relationships, and axioms to ensure they align with established design principles and ecological considerations. Their insights and critiques contribute to refining and enhancing the EIM Ontologies, ensuring its soundness, completeness, and ability to capture the domain knowledge accurately.

Empirical testing is another essential aspect of validating the EIM Ontologies. By applying the EIM Ontologies to real-world design scenarios and datasets, the effectiveness of the generative design process can be assessed. The outcomes of the generative process are evaluated against predefined criteria, benchmarks, and performance metrics to determine the EIM Ontologies' ability to generate ecologically sound and sustainable design solutions. This testing phase provides empirical evidence of the EIM Ontologies' utility and enables iterative improvements to enhance its performance.

User feedback and usability studies are integral to validating the ontology's practicality and user-friendliness. Designers and stakeholders actively engage with the EIM Ontologies and provide insights into its ease of use, clarity of concepts, and overall usability. This feedback serves to identify areas for improvement. It ensures that the EIM Ontologies are accessible to designers and facilitates their seamless integration within the design workflow.

An initial evaluation will be done by running competency questions in SPARQL against the GraphDB's SPARQL endpoint. This checks if there are results at all (ASK queries with boolean answers), number of results (COUNT queries) and the set of results returned (SELECT queries). COUNT queries are used to check completeness, whereas SELECT for the soundness of the

results. The results can then be evaluated by individual experts to see whether TOP 5 or 10 results make sense, and whether they are complete.

For the validation process, the F-measure can be used as a validation metric to assess the quality and accuracy of the generated answers obtained at the end of Loop 1. The process involves comparing the generated answers against a set of reference answers that represent the expected correct answers for a given set of questions or queries. To compute the F-measure, it is first necessary to define two sets of answers: (1) a set of generated answers (obtained from the ECOLOPES framework) and (2) a set of reference answers. These sets can be represented as lists or collections of answer instances. The F-measure is a metric that combines precision and recall providing a balanced evaluation of the system's performance. Precision measures the proportion of correct answers among the generated answers, while recall measures the proportion of correct answers among the reference answers.

To compute the F-measure, we calculate the precision and recall using the following formulas:

```
Precision = Number of Correct Answers among Generated Answers /
Total Number of Generated Answers
```

```
Recall = Number of Correct Answers among Generated Answers /
Total Number of Reference Answers
```

Once we have obtained the precision and recall values, we can compute the F-measure using the harmonic mean of precision and recall:

```
F-measure = 2 * (Precision * Recall) / (Precision + Recall)
```

The F-measure provides a single value that balances both precision and recall, giving an overall assessment of the system's performance in generating accurate answers compared to the gold standard. By employing the F-measure as a validation metric in the ECOLOPES framework, designers can quantitatively evaluate the quality of the generated answers.

The same measure can be applied for checking whether the answers obtained by volume distribution (Ontology 2) and landform generation (Ontology 3) are in the set of correct answers as outlined by the experts. For both Ontology 2 and Ontology 3 we would need corresponding datasets that are annotated to compute the F-measure.

6.2 **Validation of the EIM Ontologies in the Vienna Case Study**

A key question for the EIM Ontology concerns the selection of the types of spatial data that need to be incorporated to derive meaningful decision support and thereby meaningful design output. However, urban environments are heterogeneous. They can spatially vary greatly across short distances, have borders that are vague and hard to discern, and can be difficult to compare due to changes for instance in definitions of categories, spatial scales and methods used in data collection, data availability and classification schemes, which are critical for their description and assessment. Despite these and other challenges, there exist standards and classification schemes (Anderson et al. 1976) and methodologies (Li et al. 2019) that can facilitate objective characterisation and evaluation, which can be adapted for the purpose at

hand. Therefore, to develop a systematic approach for the design of ECOLOPES, a systematic approach for the classification of urban environments is required that is purposefully configured. This classification scheme, which is under development, will provide a means to select several case study sites, each of which will be comparable as an urban gradient or an urban analogue.

The case study sites are real locations that will be used as test beds for computational design experiments, and for testing and validating the ontology-aided generative computational design method in general and the ontology in particular. It is arbitrary (although choice must consider practicality (i.e., data availability) and competence (i.e., areas of expertise) but prerequisite to start with a model site to articulate the information modelling approach with respect to the needs of design decision making. WP5 selected a first site in Vienna Liesing for the design of an *ecolope* for a kindergarten on a site at the urban periphery that borders a residential area and landscape mosaic that includes agriculture, viticulture, and a protected forest habitat (Natura 2000 sites). An urban classification method is not necessarily needed for choosing a model site but necessary for distinguishing how case study sites are like and different from each other. Analogue sites that share important characteristics and gradients that range in terms of population density, make it possible to compare ontology-aided generative design solutions, and assess their response to local differences for calibrating model sensitivity to environmental, ecological, and user input. This can, for instance, give clues about the generative aspects of the algorithmic design process and reasoning patterns, which can serve to improve and refine the ontology. Hence, an urban habitat classification scheme not only encourages an agreement on standards and primary urban environment variables. It also enables comparison between sites from different geographic locations (in the ECOLOPES project this entails sites in Vienna, Munich, Genoa, and Haifa), evaluating design outcomes and discovering emergent solution patterns, checking inconsistencies and evaluation based on user queries and Precision and Recall metrics through analogues and gradients. Another important method for validating the ontology can be based on the comparison of results, generated by the Ecological Model for the same site, before and after design (*D4.1 Preliminary EIM Ontology*).

6.3 **Intended Development Stage at the End of the Project**

The development of Ontologies in Loops 1, 2 and 3 will reach Technology Readiness Level 4 (TLR 4) at the end of the project. As the validation (Vienna Case Study) is performed the ontologies are refined and enriched to address the feedback. In Loop 1 this will be the need to address the common designer queries as described in the brief; Loop 2 & 3 will be the data needed to fuel the Algorithm 2 & 3 respectively and the required changes.

The functionality of the ontology-aided generative computational design process depends on the interaction between its key components, namely the EIM Ontologies, the ECOLOPES Voxel Model and the ECOLOPES Computational Model.

The interaction between the EIM Ontologies with the ECOLOPES Voxel Model has been implemented through the SQL visualisation technique available in GraphDB (*D5.2 ECOLOPES Components Model*). This interface will be validated as part of the Vienna Case Study (*D5.4 Validation of ECOLOPES Computational Model*). The fully functional and validated interface will be reached by the end of the project.

The interaction between the EIM Ontologies with the ECOLOPES Computational Model relies on the development of the selected algorithms for the three Loops of the ontology-aided generative computational design process. This development is planned as two stages, of which stage 1 will reach the required TRL 4 at the end of the project, while stage 2 will develop further steps for future advancement, yet not reach TRL 4.

Stage 1 entails the development of an Answer Set Programming (ASP) approach for the *generative* process (Loop 2 and Loop 3) and will be technically implemented by the end of the project at the required Technology Readiness Level (TRL).

Stage 2 comprises outlining an approach to future development by employing additional classes of algorithms for advancing the ontology-aided generative computational design process. This involves conceptualising an ASP approach for Loop 1 and a conceptual outline for extending Loop 2 and Loop 3 with a Genetic Algorithm (GA) and a Machine Learning (ML) algorithm. Stage 2 will not reach the required TRL yet set out a clear path for future development of the ontology-aided generative computational design process.

6.4 **Technology Readiness Level 4**

The EIM Ontology will reach Technology Readiness Level 4 (TRL4) by the end of the project. To reach this maturity level, the technology must be validated in the laboratory environment. In our case, this means testing the ontology in the context of the generative design process by the ontology development team and in the context of the ECOLOPES computational design framework in the wider consortium using case-based design scenarios. To reach this developmental stage successfully by the end of the project, we are following the standards set by Semantic Interoperability Expert Group, the Ontology Landscape survey (https://ec.europa.eu/eusurvey/runner/OntologyLandscapeTemplate), and existing ontologies which has reached this TRL4. The latter includes projects such as the iCity TPSO (UofToronto) in the domain of mobility and Cities.

6.5 **Adherence to FAIR principles**

To adhere to the FAIR principle and promote research reproducibility, datasets produced during this study will be published in one of the most recognizable open access data repositories. According to the practices observed in the field, the Zenodo repository (European Organization for Nuclear Research & OpenAIRE, 2013) has been identified as a repository that promotes discoverability of datasets published in the field of architectural design. We will publish the EIM ontologies in the ECOLOPES gitlab repository which will be made public. This will enable tracking of changes between versions and feedback by the community to indicate "Issues" that need to be resolved. The effort related to ensuring data interoperability has been initiated as a part of the data exchange functionality required for the integration of the ECOLOPES Voxel Model data with the components implemented in the ECOLOPES computational workflow.

7. PUBLICATION PLAN

We have recently submitted a scientific article to Frontiers of Architectural Research journal on the "Conceptual framework for an ontology-aided generative computational design process for ecological building envelopes", in which the generative computational design process and role of the EIM Ontologies is elaborated in conceptual terms.

We are now in the process of preparing a scientific article on the specific development and utilisation of the "EIM Ontologies in the context of an ontology-aided generative computational design process for ecological building envelopes". The article on this subject from a computational architecture perspective will be submitted to the scientific journal *Automation in Construction* (Elsevier).

This article will be accompanied by a scientific article focused on technical elaboration of the EIM ontologies from a computer science perspective, targeting the *Journal of Web Semantics* (Elsevier) or *The Semantic Web Journal* (IOS Press).

Finally, we will prepare a scientific article on "Validation of the ECOLOPES ontology-aided generative computational design process for ecological building envelopes" that will report the results of the Vienna Case Study and include validation of the EIM Ontologies.

PART B - Ecological Model

8. THE ECOLOGICAL MODEL

8.1 Short summary of Ecological Model

The ECOLOPES model (Figure WP4-1) is a spatially-explicit model that models the interdependent spatial and temporal dynamics of soil, microbiota, plants, and animals, as response to the regional species pool, the local abiotic conditions, the geometry of the building, the substrate used to design the *ecolope*, and the human management. The biological units of the model are plant (PFG) and animal (AFG) functional groups and generic soil classes, which makes the model generalizable to all sets of conditions. The model is based on a multiscalar approach. The regional model determines which FGs of the species pool have a reasonable chance to colonise the *ecolope* according to its location in the city. The local model applies a second filter on these species based on the abiotic and biotic conditions delivered by the *ecolope* and simulates demographic processes to predict the outcome of inter-and intra-specific interactions. The output of the models is a temporal sequence of plant-animal-soil community development.

Deliverable 4.1. (*Preliminary EIM Ontology*) described the first prototype of the ecological model, and Deliverable 1.5 (*Report after 2nd year*) summarised major advances of the following months. A 2.5 D – version of the plant-soil model has been integrated into the CAD environment and populates the Knowledge Base. This progress regarding workflow integration is described in D3.3.(*Interim ECOLOPES platform architecture*). Here, we describe the modelling progress for months 25 - 28 (April to July 2023) where we focused on consolidating separate work branches, improving information flow between model subcomponents and further bug fixes. We describe the progress in the development of the local model in sections 8.2 and 8.3.

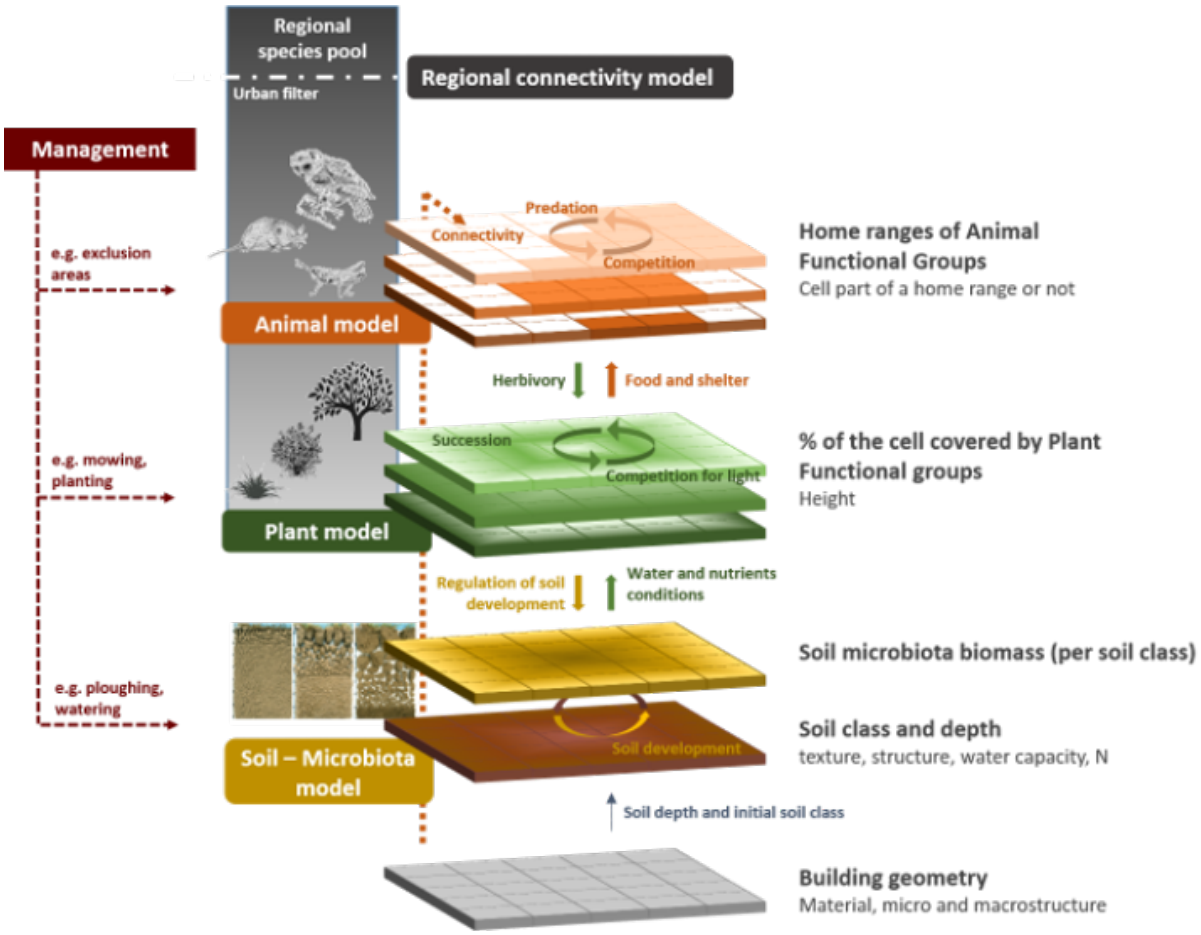

Fig. 27: Overview of the principal components of the ECOLOPES ecosystem model, of their interactions, and of the ecological processes the model accounts for.

8.2 Local Model

The ecological model is a single component in the ECOLOPES workflow that consists of four parts: animal, plant, and soil submodels, which are independent subunits, developed as stand-alone models and developed/updated asynchronously; and a central subunit that connects the subunits and handles input/output, as well as all necessary translations. The plant and animal models have been described in D4.1, and the soil model was described in the annual report of year 2. The concepts and a first prototype of the central subunit was also introduced in D4.1.

In M 25 to 28 we revised the information flow to increase speed and reliability of the code. The new information flow is as follows (see Fig. 28):

Upon startup, the model reads a configuration file that helps locating the inputs. Warnings or errors may be issued if entries are faulty, files are missing or corrupted, or information among files does not match (e.g., in spatial extent). Then the submodels are initialised in sequential order: 1) the soil model creates a soil class and passes on a copy to central storage; 2) the plant model simulates one plant model time step (using the soil class as habitat suitability information) and thus creates a plant community. It passes a summary (plant biomass) to the central storage; 3) the animal model uses the current plant biomass as resource maps and creates the animal home ranges. It also outputs the current animal biomass to the central storage. Subsequently, the model runs for t time steps (t is usually between 10 and 100 years, depending on the building's life span). Each submodel again

interprets the biomass in the information storage to build a community (1a - 1c), i.e., the plant model uses the soil class information and the animal biomass to create habitat suitability and animal disturbance maps, while the animal model uses the plant biomass to create plant resource maps. The complex community objects are stored in the submodels. The models need not run sequentially in this phase, because each model uses the biomass information of the preceding year (t-1) to update this year's (t) community. Once all three models are finished, they update the information storage with the biomass of time t (2a – 2c).

Apart from the changes to the information flow, we concentrated on testing and on fixing critical bugs that arose from consolidating separate workflows. Taken together, the bug fixes and new information workflows reduced computation time of example sites from approximately 30 minutes to 5 minutes.

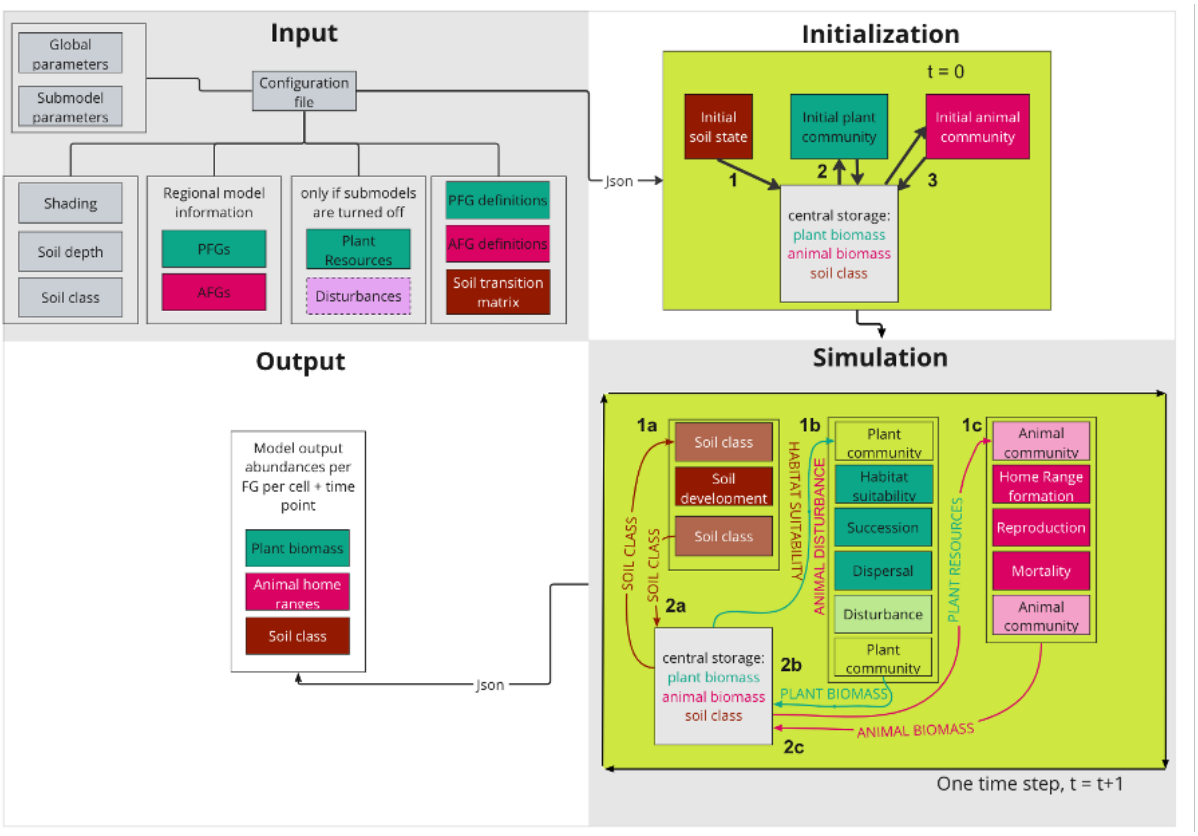

Fig. 28: The local ecological model, including soil, plant, and animal submodels, as well as a central storage container responsible for the translation and information transmission.

8.3 Model parametrization

Here we describe progress in parametrizing the plant functional groups (PGFs). The PFGs building workflow has been developed according to the methodology outlined in the previous deliverable and stored in dedicated shared R-scripts. The most time-consuming activity was data cleaning and harmonisation as the use of the TRY database for retrieving plant traits contained many different data sources and data formats that needed to be uniformed before proceeding with the analysis. The PFGs building workflow retrieved 658 groups that were based on ca. 19'000 plant species and that were then validated according to the methodology proposed by Boulangeat et al. (2012) using the open access sPlot vegetation surveys database (Sabatini et al., 2021). To this aim, we calculated taxonomic diversity indices and functional

diversity indices at both the species and PFG level and then correlated the two sets of indices. The validation retrieved high values of correlation (i.e., > |0.7|) between species and PFGs for most indices suggesting that our classification was robust and summarised adequately the main diversity trends in plant communities. Lastly, the PFGs parametrization process, that is needed to incorporate PFGs as modelling units into the plant model, was tested and carried out using the RFate R package.

REFERENCES

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., et al. (2016). Brick: Towards a unified metadata schema for buildings. In Proceedings of the ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys). (pp. 41-50). ACM. https://doi.org/10.1145/2993422.2993577

Bennett, G., & Mulongoy, K., (2006). Review of experience with ecological networks, corridors and buffer zones. *CBD Technical Series*, 23. https://www.cbd.int/doc/publications/cbd-ts-23.pdf

Boulangeat, I., Philippe, P., Abdulhak, S., Douzet, R., Garraud, L., Lavergne, S., & Thuiller, W. (2012). Improving plant functional groups for dynamic models of biodiversity: at the crossroads between functional and community ecology. *Global change biology*, 18(11), 3464-3475. https://doi.org/10.1111/j.1365-2486.2012.02783.x

Dwiartama A, Rosin C (2014) Exploring agency beyond humans: the compatibility of Actor-Network Theory (ANT) and resilience thinking. *Ecology & Society*, *19*(3), 28. http://dx.doi.org/10.5751/ES-06805-190328

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220. https://doi.org/10.1006/knac.1993.1008

Gruber, T. (2016). Ontology. In L. Liu, M. Özsu (Eds.) *Encyclopedia of Database Systems*. (pp. 1-3). Springer. https://doi.org/10.1007/978-1-4899-7993-3_1318-2

Hensel, M. (2010) Performance-oriented Architecture: Towards a Biological Paradigm for Architectural Design and the Built Environment. *FORMAkademisk* *3*(1), 36-56. https://doi.org/10.7577/formakademisk.138

Hensel, M. (2011) Performance-oriented Architecture and the Spatial and Material Organisation Complex - Rethinking the Definition, Role and Performative Capacity of the Spatial and Material Boundaries of the Built Environment. *FORMAkademisk*, *4*(1), 3-23. https://doi.org/10.7577/formakademisk.125

Hensel, M. (2012) Sustainability from a Performance-oriented Architecture Perspective - Alternative Approaches to Questions regarding the Sustainability of the Built Environment. *Sustainable Development*, *20*(3), 146-154. https://doi.org/10.1002/sd.1531

Hensel M. (2013) Performance-oriented Architecture – Rethinking Architectural Design and the Built Environment. Wiley & Sons.

Hensel, M., & Sunguroğlu Hensel, D. (2020) Performances of Architectures and Environments: En Route to a Theory and Framework. In M. Kanaani (Ed) *The Routledge Companion to Paradigms of Performativity in Design and Architecture – Using Time to Craft an Enduring, Resilient and Relevant Architecture* (pp. 27-43). Routledge.

Jasiewicz, J., & Stepinski, T.F. (2013) Geomorphons - a pattern recognition approach to classification and mapping of landforms. *Geomorphology, 182*, 147-156. https://doi.org/10.1016/j.geomorph.2012.11.005

Latour, B. (2005) Reassembling the Social: An Introduction to Actor-Network Theory. Oxford University Press.

Latour, B. (2017) Why Gaia is not a God of Totality. *Theory, Culture, & Society*, *34*(2-3), 61-81. https://doi.org/10.1177/0263276416652700

Orciuoli, F., Parente, M. (2017) An ontology-driven context-aware recommender system for indoor shopping based on cellular automata. *Journal of Ambient Intelligence and Humanized Computing, 8*, 937–955. https://doi.org/10.1007/s12652-016-0411-2

Rasmussen, M. H., Lefrançois, M., Schneider, G., & Pauwels, P. (2020). BOT: the Building Topology Ontology of the W3C Linked Building Data Group. *Semantic Web*, *12*(1), 143-161. https://doi.org/10.3233/SW-200385

Pruski, C., & Sunguroğlu Hensel, D. (2022) 'The Role of Information Modelling and Computational Ontologies to Support the Design, Planning and Management of Urban Environments: Current status and future challenges. In A. Chokhachian A, M. Hensel, & K. Perini (Eds) *Informed Urban Environments: Data-integrated Design for Human- and Ecology-centered Perspectives*. (pp. 51-70) Springer. https://doi.org/10.1007/978-3-031-03803-7_4

Sabatini, F. M., Lenoir, J., Hattab, T., Arnst, E. A., Chytrý, M., Dengler, J., De Ruffray, P., Hennekens, S.M., Jandt, U., Jansen, F., Jiménez-Alfaro, B., Kattge, J. Levesley, A., Pillar, V. D., Purschke, O., Sandel, B., Sultana, F., Aavik, T., Aćić, S., ... & Wagner, V. (2021). sPlotOpen–An environmentally balanced, open-access, global dataset of vegetation plots. Global Ecology and Biogeography, 30(9), 1740-1764. https://doi.org/10.1111/geb.13346

Sunguroğlu Hensel, D. (2017) *Convergence – Materials Adaptation and Informatics in Architecture* (Doctoral Thesis). The Oslo School of Architecture and Design.

Sunguroğlu Hensel, D., Tyc, J., & Hensel, M. (2022) Data-driven Design for Architecture and Environment Integration: Convergence of data-integrated workflows for understanding and designing environments. *Spool*, *9*(1), 19-34. https://doi.org/10.47982/spool.2022.1.02

Torta, G., Ardissono L., La Riccia L., Savoca, A., & Voghera, A. (2017). Representing Ecological Network Specifications with Semantic Web Techniques. In Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD, 86-97. https://doi.org/10.5220/0006573500860097

Weisser, W., Hensel, M., Barath, S., Grobman, Y.J., Hauck, T.E., Joschinski, J., Ludwig, F., Mimet, A., Perini, K., Roccotiello, E., Schloter, M., Shwartz, A., Sunguroğlu Hensel, D., Vogler, V. (2022) Creating ecologically sound buildings by integrating ecology, architecture, and computational design. *People & Nature*, 5(1), 4-20. https://doi.org/10.1002/pan3.10411