Ontology-aided Generative Computational Design Approach to Ecological Building Envelopes

Author: Albin Ahmeti, TU WIEN

ECOLOPES

Technical Summary #2

Ontology-aided Generative Computational Design Approach to Ecological Building Envelopes

Authors: Albin Ahmeti, TU WIEN

Short Summary

A team of researchers from the Department of Digital Architecture and Planning at Institute of Architectural Science from Vienna University of Technology chose Ontotext's technology to bring a new approach to an ontology-aided early stage computational design for ecological building envelopes and urban sites

The Challenge

The main challenge is to provide the most complete view of relevant available multi-disciplinary data to enable architects and ecologists to co-create designs capable of addressing architectural and ecological requirements of their projects.

Existing data covers different sub-domains of architecture and ecology, as well as diverse subject matter and formats. It usually comprised separate datasets on buildings, sites, species and their requirements, ecological networks, correlations between geometry and species, geometric articulation of buildings and the implication on species reachability, etc. On top of that, the data was often in different states of completeness and needed to be normalized and enriched.

The Solution: Ontology-based data access for ECOLOPES in GraphDB

The team at Vienna University of Technology collaborated with the other consortium partners to establish a growing set of competency questions, which are required to design ecological building envelopes, and applied Ontology-based Data Access (OBDA) to integrate all the data silos within the ECOLOPES Knowledge Graph. OBDA is a data integration framework that can be implemented off-the-shelf by the Ontotext GraphDB RDF database since it supports both persistence (materialization) and virtualization of data via Ontop.

Integrating data in ECOLOPES Knowledge Graph

When creating ecological solutions for building design, in some cases the data needs to be ingested from original sources, transformed, and stored. In other scenarios, it doesn't have to be moved but instead virtualized using an existing ontology.

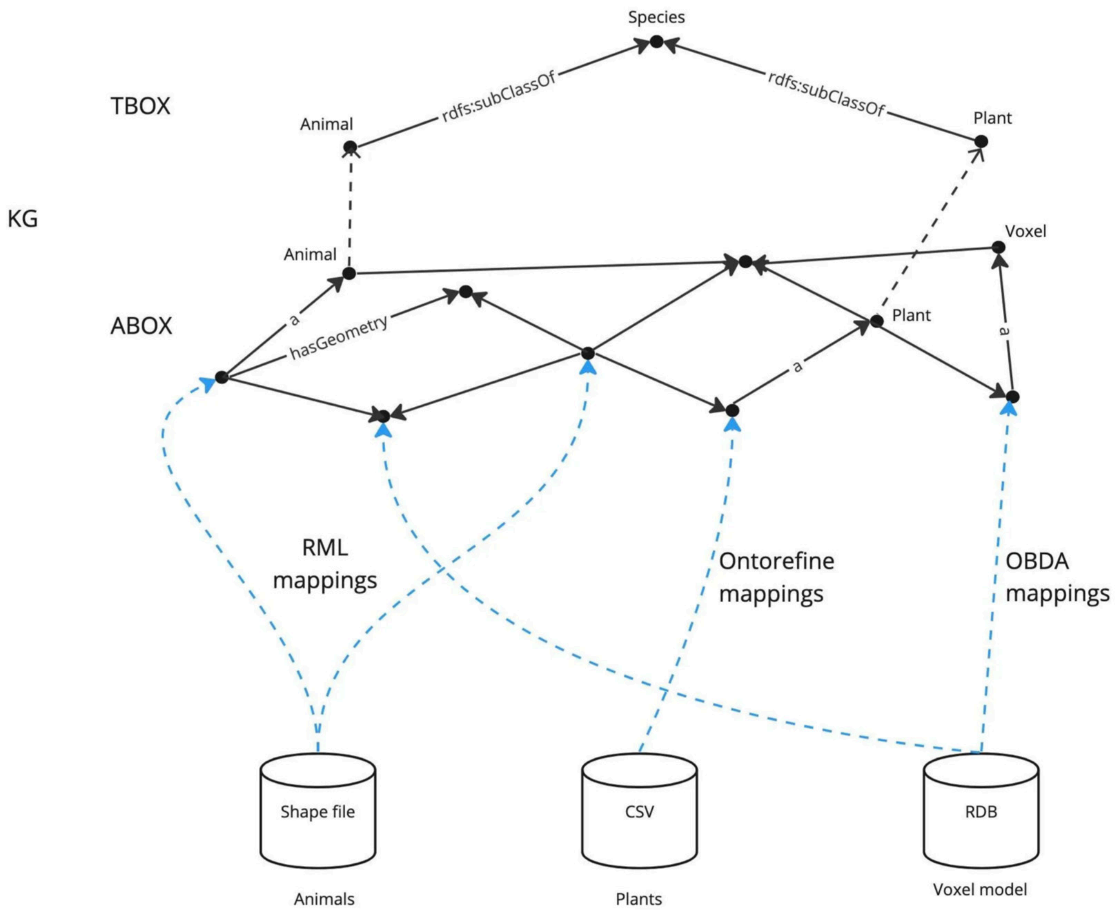

Figure 1: Ontology-based data access in GraphDB. Voxel model is virtualized using OBDA

In the case of materialization, the data about plant functional groups, plants and animals in Vienna, biotic interactions between species, and so on, have been mapped using Ontotext Refine mappings and stored in a GraphDB repository. In the case of virtualization, the voxel model storing the solar radiation of a site (among other attributes such as x, y, z coordinates, aspect, slope, and so on) and maintained in a Postgres relational database has been "virtualized" by providing OBDA mappings.

Another data source is what architects and designers generate when translating requirements from a design brief. This includes data initiated by CAD or Grasshopper workflows. For example, a designer might place Architectural or Ecological Nodes together with the relations and constraints between them. After such placement, all the architectural and ecological objects get transformed and linked to the knowledge graph using Grasshopper workflows via JSON-LD format. Then they are pushed to GraphDB via its API.

Fig 2 shows a snapshot of the knowledge graph connecting various silos of data.

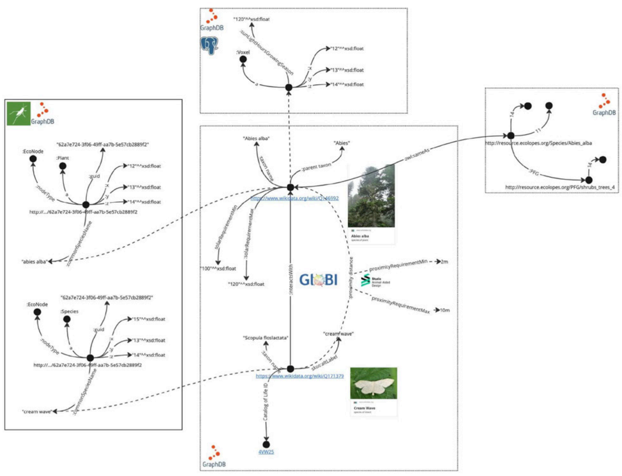

Figure 2: A snapshot of ECOLOPES Knowledge Graph connecting data sources originated from Rhino/Grasshopper (left), GLoBI (Global Biotic Interactions) and Animal Aided Design (center), voxel model (top), and plant functional groups (right)

Interacting with ECOLOPES Knowledge Graph

With the help of the ECOLOPES Knowledge Graph, architects and planners can ask more complex competency questions. For example, "Give me all tree species from the genus Abies that interact with a cream wave?", or "How do we accommodate the cream wave from the architecture perspective?", or the more intricate "Which tree species are similar to platanus x hispanica?".

These competency questions, along with the ones mentioned earlier about solar radiation and proximity distance, are implemented respectively by SPARQL CONSTRUCT queries (and exploiting GraphDB's math functions), which can mimic a rule-based system run within a Grasshopper workflow.

In Fig. 3, we can see the result within the yellow highlighted area representing the solar radiation, where the green circles represent nodes (e.g., plants) that satisfy the solar requirements and other requirements (e.g. proximity distances), whereas the red crosses are nodes that don't satisfy all of the requirements. Architects and designers can freely move those nodes on a given site or within a building, and get feedback from the knowledge graph.

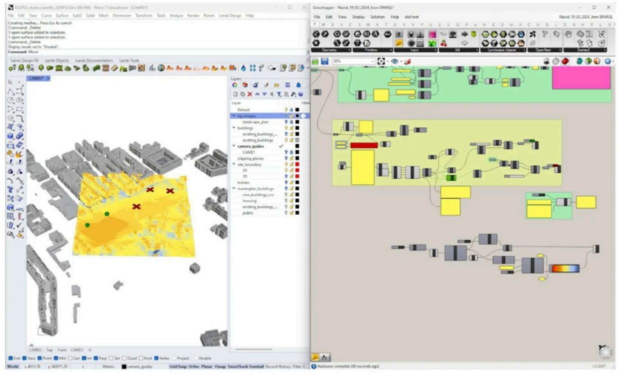

Figure 3: Designer interaction with the knowledge graph when placing nodes in CAD environment (left) and Grasshopper workflow (right) that does the computation by querying and interacting with the knowledge graph.

The visualization of the data model in the context of Ecological Node satisfying the Solar and Proximity Constraints is shown in Fig. 4, using the open source version of Ontodia.

To conclude, the ECOLOPES knowledge graph integrates multi-disciplinary data (including the spatial-voxel model) and correlation data concerning architecture and ecology (the ecological model). It comprises three linked ontologies aiming to assist the design process in computational architecture and multi-species architectural design.