

Generative Ontology - BetterBIM workshop

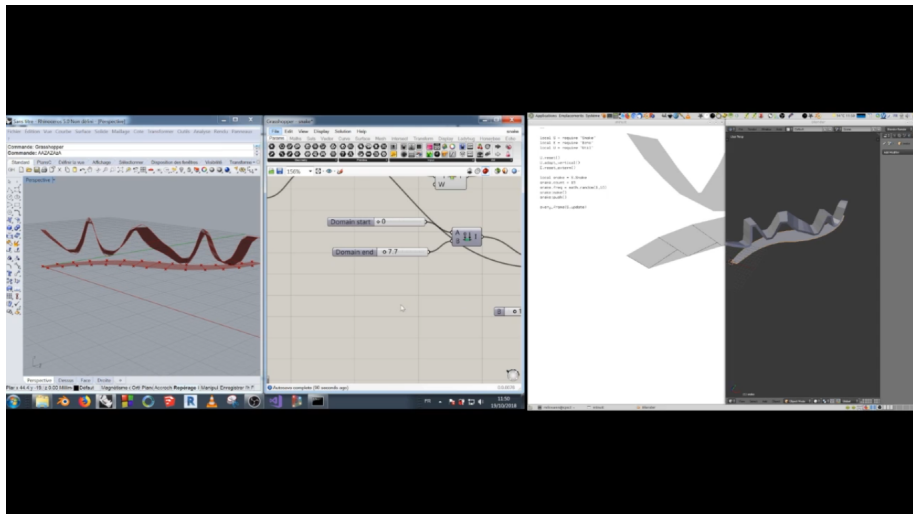
Milovann Yanatchkov

July 1, 2022

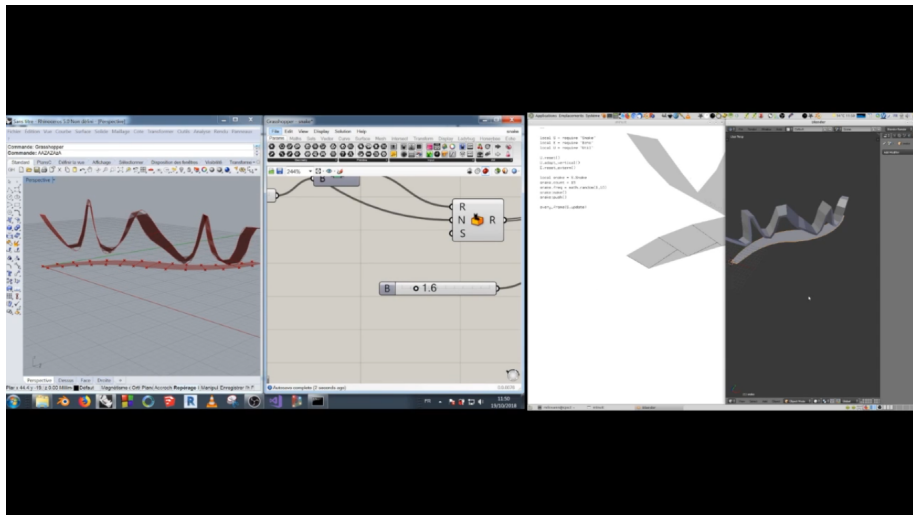
Generative Ontology

- KGs : **speculative ideas** on graphs and **generative modeling**
- From flow-based modeling
- A « linked-over-software » data model
- github.com/rvba/echo (video)
- To knowledge graphs
- Can ontology apply to « distributed modeling » ?

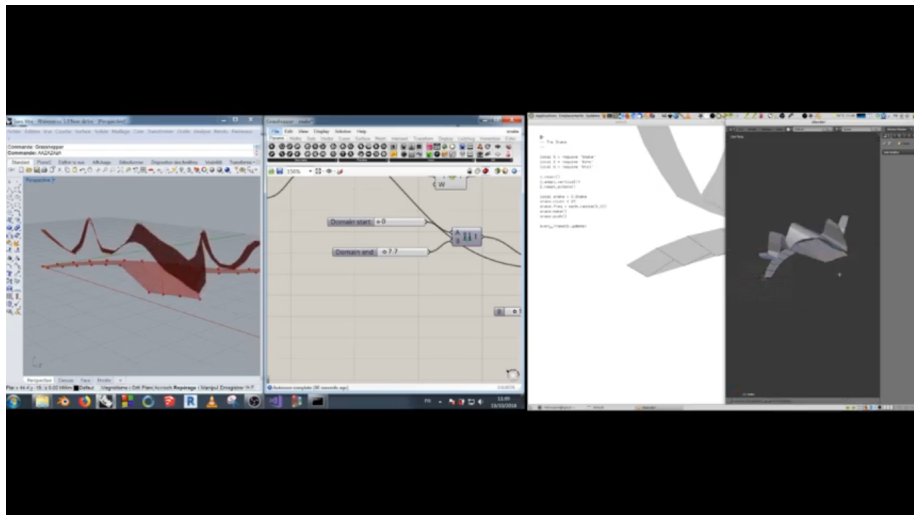
Generative Ontology



Generative Ontology



Generative Ontology



Generative Ontology

- AEC digital tools has grown to the size of a « global **ecosystem** »
- New **workflows** imply a switch from the « software » paradigm
- To the « **platform** » paradigm
- There is now a need for an « extended interoperability »
- With **Federated Linked Building Data**

Generative Ontology

- In the context of the platform paradigm
- The current BIM paradigm has a « **monolithic data** » problem
- And a « **static data** » problem :
- BIM models are not designed for « **change** »
- This is acceptable for the late design phase
- But not for the **early design phase**

Generative Ontology

- **Knowledge graphs** are used for « static knowledge » (facts)
- When project data is generated in the late design phase
- Can they be usefull for « **dynamic** knowledge » ?
- When project data is subject to constant change ?
- Can knowledge graphs apply to **generative design** ?

Generative Ontology

- *RDF_Prototypes* has shown BHoM's
- « open » **object-oriented** approach
- Closer to a **graph-based** ontology
- Rather than the « classical » OOP paradigm
- Data-flow seeks **flow-based programming** paradigm
- A paradigm that focuses on *edges* rather than *nodes*

Generative Ontology

- **Flow-based programming** defines
 - « *applications as networks of black box processes* »
- **Adapters** are BHoM's core « translation flow »
- With *FromX* and *ToX* inputs and outputs
- **Engines** are BHoM's core « generative flow »
- With « *generative* » and/or « *transformative* » functions
- Complex design workflows imply
- Flows of *translation* and *generation* processes
- Is there an **ontology** for this kind of flow-based networks ?

Generative Ontology

- Let's name *Generative Ontology* an ontology that
- Can represent the « **flows of transformations** » of the Data
- An ontology that can describe :
- How the data is **generated**
- How the data is **translated**
- Enabling the sharing and composition of « generative networks »

Generative Ontology

- Generative ontology should describe networks made of :
- Edges describing the « generative logic »
- Nodes as **operators** (functions)
- **Abstract** « black-boxes » with :
- Types : Inputs and Outputs *object types*
- Verbs : the *abstract function* of the node

Generative Ontology

- Is RDF Subject-Predicate-Object the right approach ?
- With the **predicate** describing generative operators
- Object *Generates* \Rightarrow Object
- Object *IsGeneratedBy* \Rightarrow Object
- Object *IsTranslateIn* \Rightarrow Object
- Object *IsTranslatedFrom* \Rightarrow Object

Generative Ontology

- A (not-so) related ontology could be found in Autodesk
- *Ontology for Generative Design of Mechanical Assemblies*
- But it focuses mainly on
- **Configuration design** of mechanical assemblies with
- « **Assembly Ontology** developed to specify
- Connection, parthood, shape, and boundary constraints
- For a generative design software tool. »

Generative Ontology

- The **Function Ontology** maybe more related in
- « *how to semantically declare and describe functions, their input parameters, and possible outputs.* »
- A clear **flow-based** approach is found in the *problem statment*:
- « functions are *described* independent of the technology that implements them »