OWFgraph

A graph database for the off-shore wind farm domain

Erik Quaeghebeur Sebastian Sanchez Michiel Zaaijer

Wind Energy Group — Delft University of Technology — The Netherlands

EUROS Programme – Wind Farm Design Optimization – Uncertainty Model of Wind Farms

WESC 2017
27 June 2017

2.1 Wake description

The wake behind a turbine is assumed to have a start diameter equal to the turbine diameter, and to spread linearly as a function of downwind distance. This simplification means that the wake velocity cannot be found very accurately at all downwind positions, but by adjusting the spread angle to fit data at distances larger than about four diameters, only the calculation of the near-wake zone will involve large errors. As wind turbines are seldom put closer together than this distance, it is not necessary to make accurate calculations here.

Inside the wake the velocity is considered constant, instead of using the commonly seen Gaussian distribution. This simplification is made because the aim of the model is to give an estimate of the energy content in the wind field seen by the downwind turbines, rather than to describe the velocity field accurately.

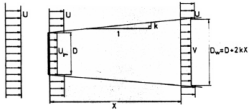


Fig. 2. Schematic view of wake description

With symbols defined in Fig. 2, a balance of momentum gives:

$$D^2U_r + (D_w^2 - D^2)U = D_w^2 V$$

The wake velocity is found by the expression

$$V/U = 1 - 2a/(1 + 2kX/D)^2 \quad ,$$

a is defined as the initial velocity deficit $1-U_r/U$ but can also be expressed as

$$a = (1 - \sqrt{1 - C_t})/2$$

where Ct is the thrust coefficient of the turbine. Hence, the velocity deficit of the wake at a given position X is:

$$1 - V/U = (1 - \sqrt{1 - C_t})/(1 + 2kX/D)^2 \ .$$

The problem of interacting wakes is solved by assuming the kinetic energy deficit of a mixed wake to be equal to the sum of the energy deficits for each wake at the calculated downwind position.

TUDelft

2.1 Wake description

The wake behind a turbine is assumed to have a start diameter equal to the turbine diameter, and to spread linearly as a function of downwind distance. This simplification means that the wake velocity cannot be found very accurately at all downwind positions, but by adjusting the spread angle to fit data at distances larger than about four diameters, only the calculation of the near-wake zone will involve large errors. As wind turbines are seldom put closer together than this distance, it is not necessary to make accurate calculations here.

Inside the wake the velocity is considered constant, instead of using the commonly seen Gaussian distribution. This simplification is made because the aim of the model is to give an estimate of the energy content in the wind field seen by the downwind turbines, rather than to describe the velocity field accurately.

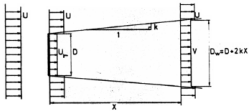


Fig. 2. Schematic view of wake description

With symbols defined in Fig. 2, a balance of momentum gives:

$$D^2U_r + (D_w^2 - D^2)U = D_w^2\,V$$

The wake velocity is found by the expression

$$V/U = 1 - 2a/(1 + 2kX/D)^2 \quad ,$$

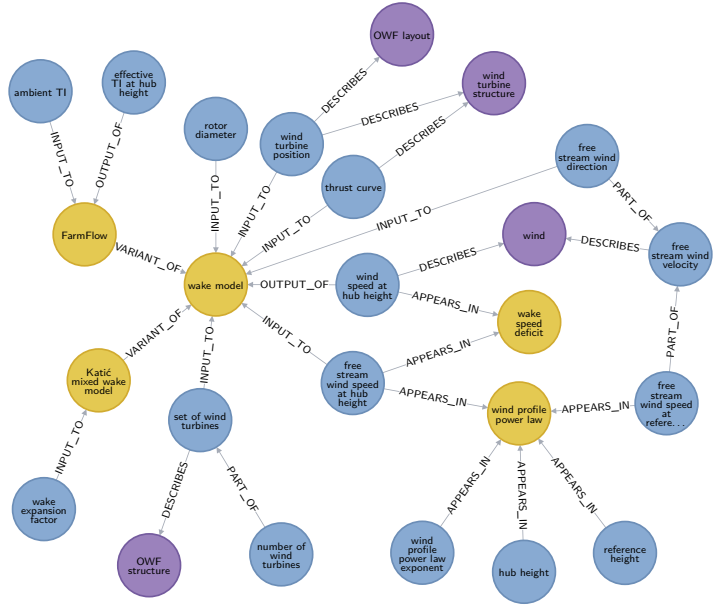
a is defined as the initial velocity deficit $1-U_r/U$ but can also be expressed as

$$a = (1 - \sqrt{1 - C_t})/2$$

where Ct is the thrust coefficient of the turbine. Hence, the velocity deficit of the wake at a given position X is:

$$1 - V/U = (1 - \sqrt{1 - C_t})/(1 + 2kX/D)^2 \quad .$$

The problem of interacting wakes is solved by assuming the kinetic energy deficit of a mixed wake to be equal to the sum of the energy deficits for each wake at the calculated downwind position.



Overview

Goal Why build a graph database for the offshore wind farm domain?

Overview

Goal Why build a graph database for the offshore wind farm domain?

Content What do we put into the database?
How do we structure the content in the database?
Does everything fit easily in this structure?

Overview

Goal Why build a graph database for the offshore wind farm domain?

Content What do we put into the database?
How do we structure the content in the database?
Does everything fit easily in this structure?

Practical setup What software and hardware do we use?

Overview

Goal Why build a graph database for the offshore wind farm domain?

Content What do we put into the database?
How do we structure the content in the database?
Does everything fit easily in this structure?

Practical setup What software and hardware do we use?

Querying How do we add and extract data?

Overview

Goal	Why build a graph database for the offshore wind farm domain?
Content	What do we put into the database? How do we structure the content in the database? Does everything fit easily in this structure?
Practical setup	What software and hardware do we use?
Querying	How do we add and extract data?
Conclusions	What is different from what we expected? What are our plans for the future?

Goal – EUROS programme context

Project 3 Wind Farm Design Optimization

Work Package 3.2 Uncertainty Model of Wind Farms

Goal – EUROS programme context

Project 3 Wind Farm Design Optimization

Work Package 3.2 Uncertainty Model of Wind Farms

Challenge Develop a model for the accumulation of uncertainty from multiple sources in performance and cost of an entire OWF.

Goal – EUROS programme context

Project 3 Wind Farm Design Optimization

Work Package 3.2 Uncertainty Model of Wind Farms

Challenge Develop a model for the accumulation of uncertainty from multiple sources in performance and cost of an entire OWF.

Activities
- make inventory of sources of uncertainty;
- create causal map of uncertainty propagation;
- assessment of uncertainty contributions to OWF CoE;
- select uncertainty propagation approach.

Goal – EUROS programme context

Project 3 Wind Farm Design Optimization

Work Package 3.2 Uncertainty Model of Wind Farms

Challenge Develop a model for the accumulation of uncertainty from multiple sources in performance and cost of an entire OWF.

Activities
- make inventory of sources of uncertainty;
- create causal map of uncertainty propagation;
- assessment of uncertainty contributions to OWF CoE;
- select uncertainty propagation approach.

Goal – EUROS programme context

Project 3 Wind Farm Design Optimization

Work Package 3.2 Uncertainty Model of Wind Farms

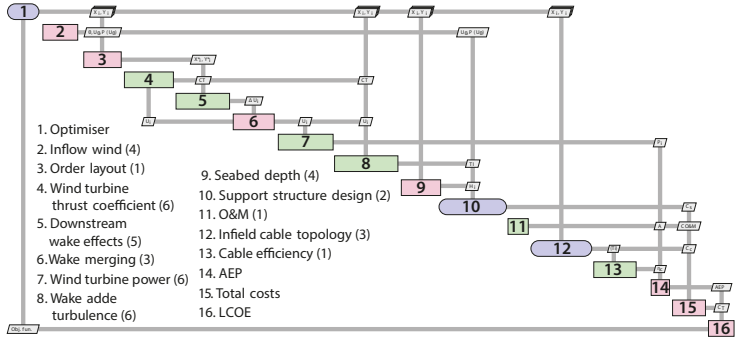
Challenge Develop a model for the accumulation of uncertainty from multiple sources in performance and cost of an entire OWF.

Activities
- make inventory of sources of uncertainty;
- create causal map of uncertainty propagation;
- assessment of uncertainty contributions to OWF CoE;
- select uncertainty propagation approach.

⇒ Conclusion Create a structured description of the domain

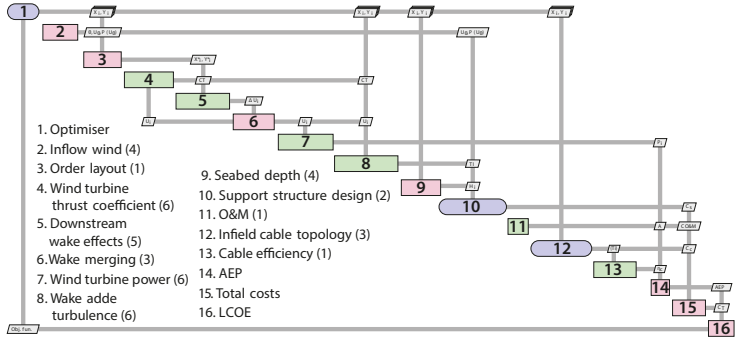
Goal – other contexts

- Evaluation of multidisciplinary design analysis and optimization workflows (Sebastian's project)



Goal – other contexts

- Evaluation of multidisciplinary design analysis and optimization workflows (Sebastian's project)



- Teaching:
  - domain exploration;
  - analysis of how different disciplines are coupled.
- . . .

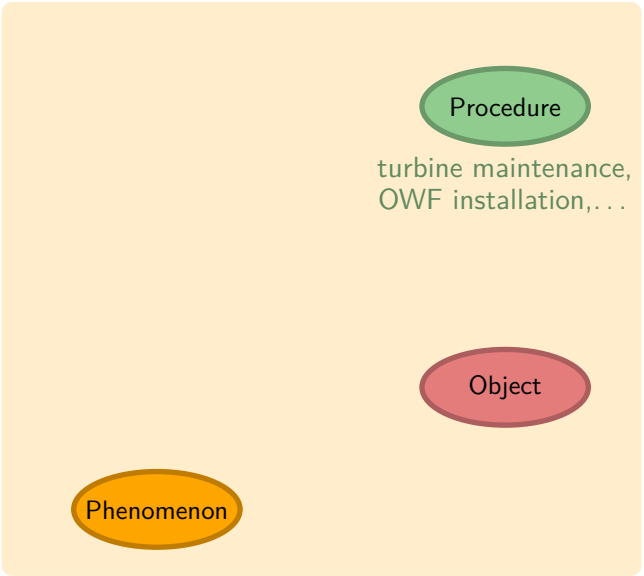
Content

Content as node labels

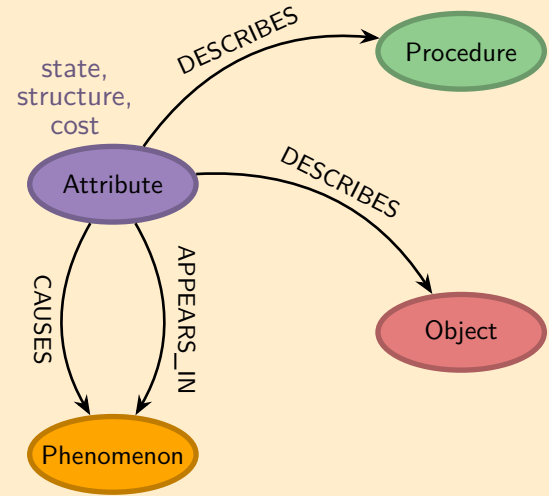
support structure,
active yaw system,
rotor bearing,...

Content as node labels

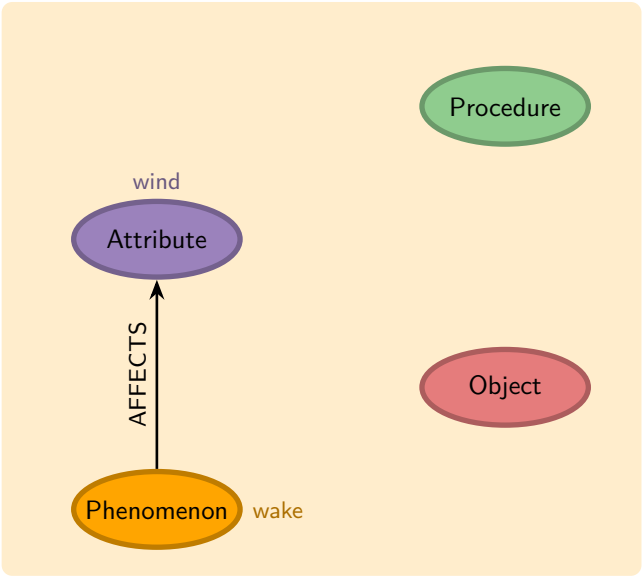
Content as node labels



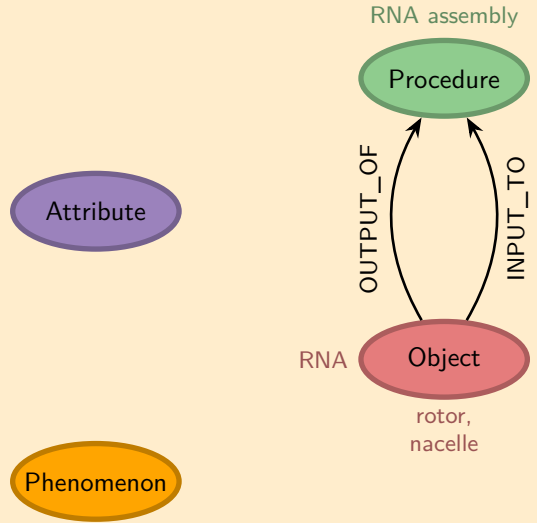
Content & its structure as edges and their type



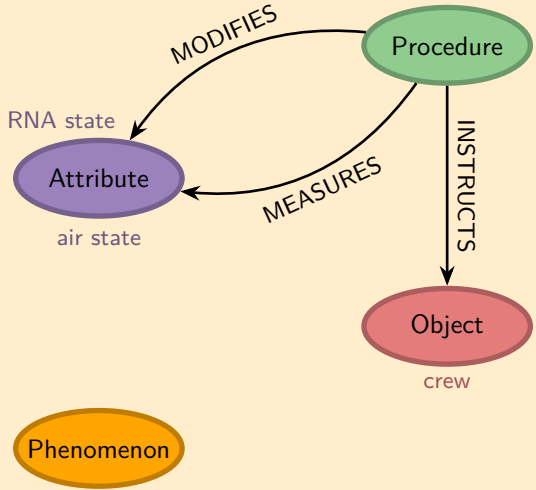
Content & its structure as edges and their type



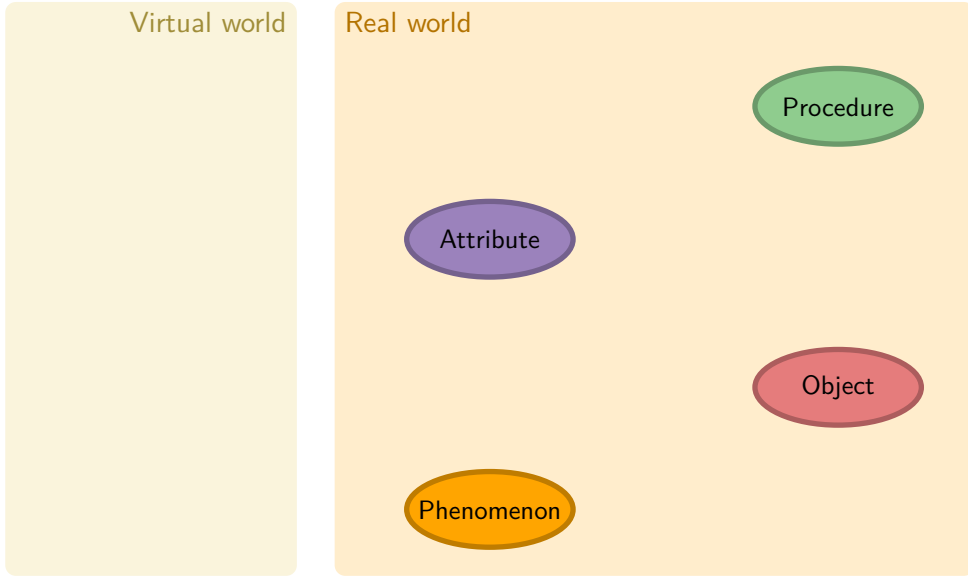
Content & its structure as edges and their type



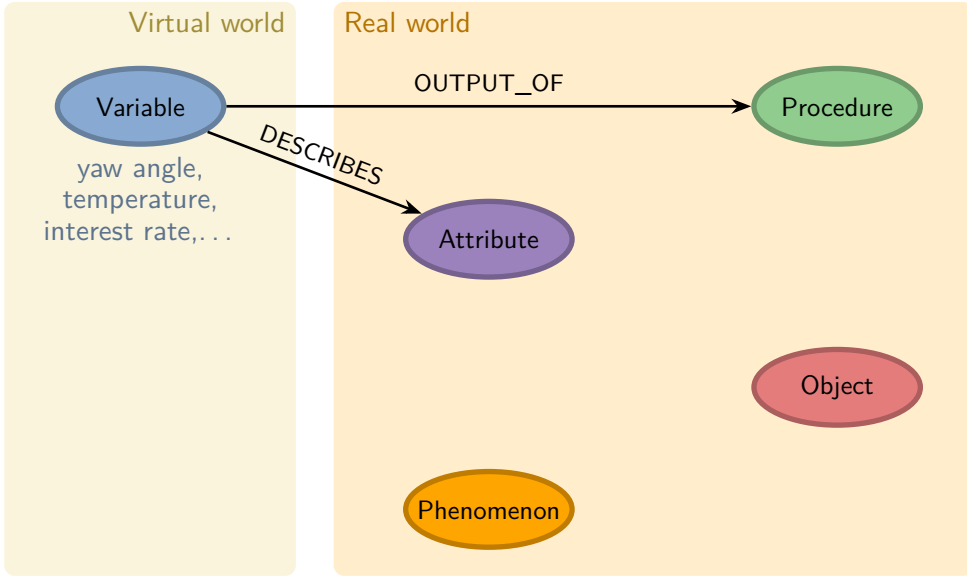
Content & its structure as edges and their type



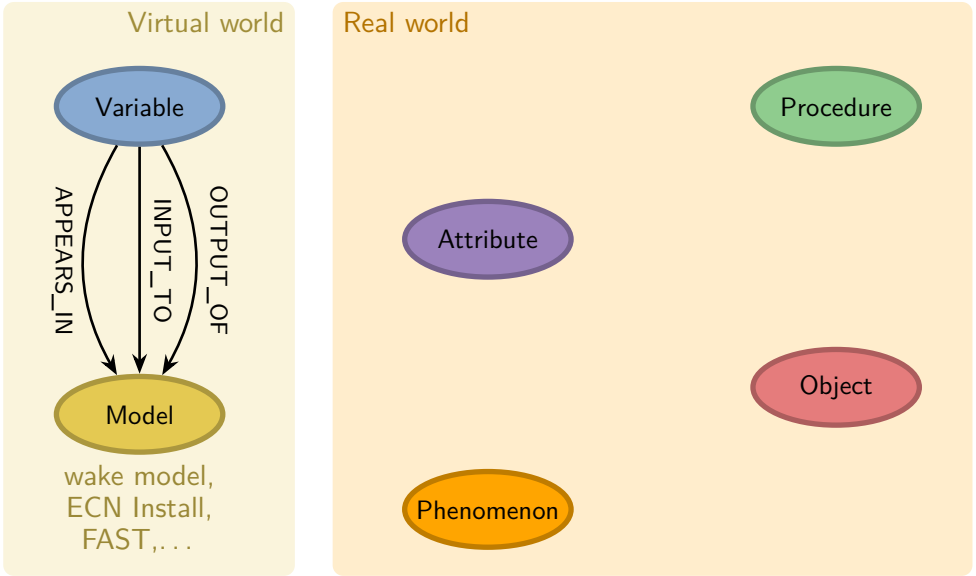
Content & its structure as edges and their type



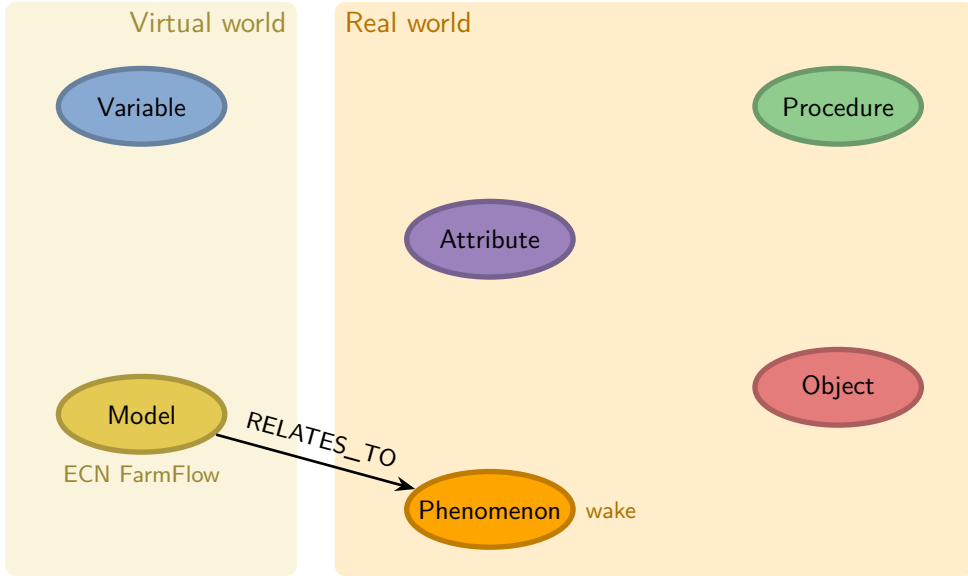
Content & its structure as edges and their type



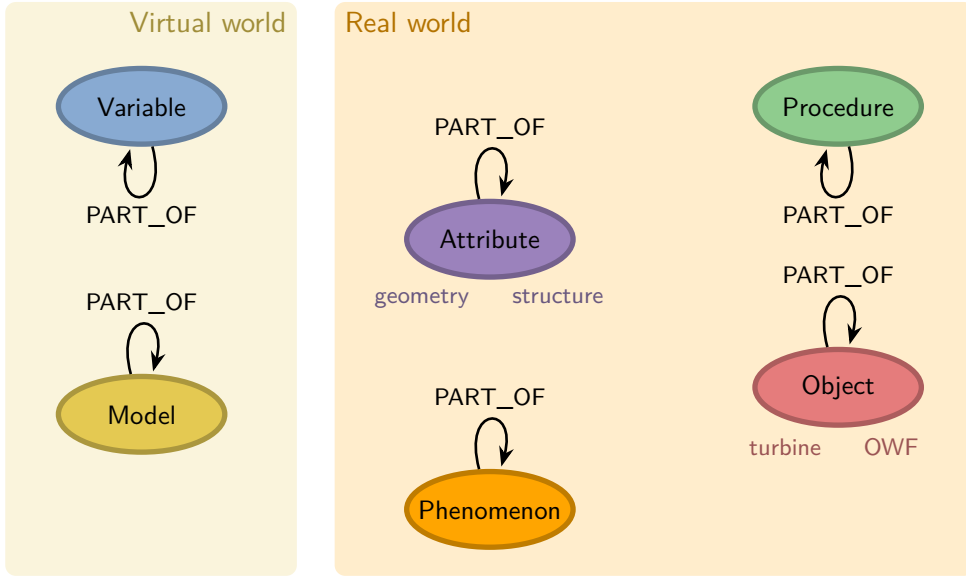
Content & its structure as edges and their type



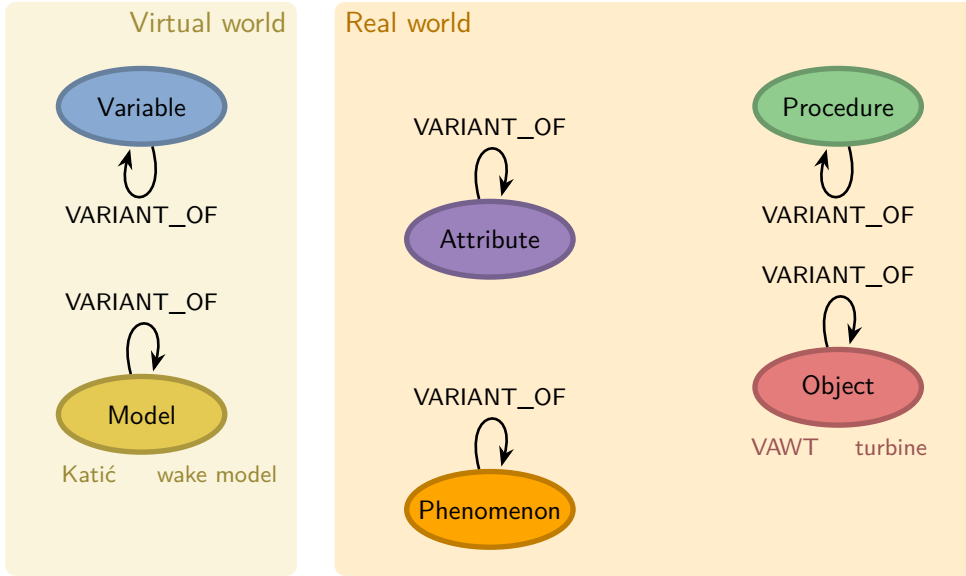
Content & its structure as edges and their type



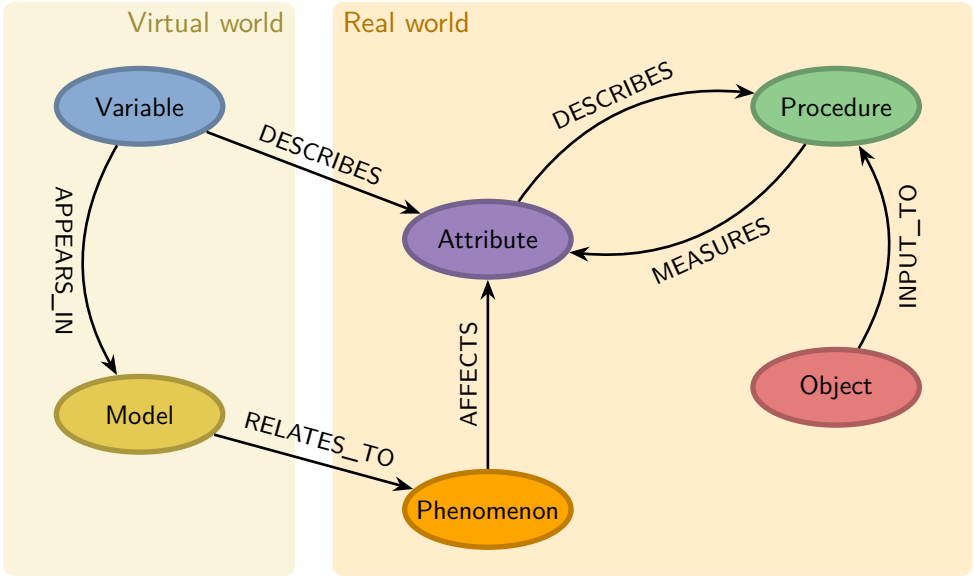
Content & its structure as edges and their type



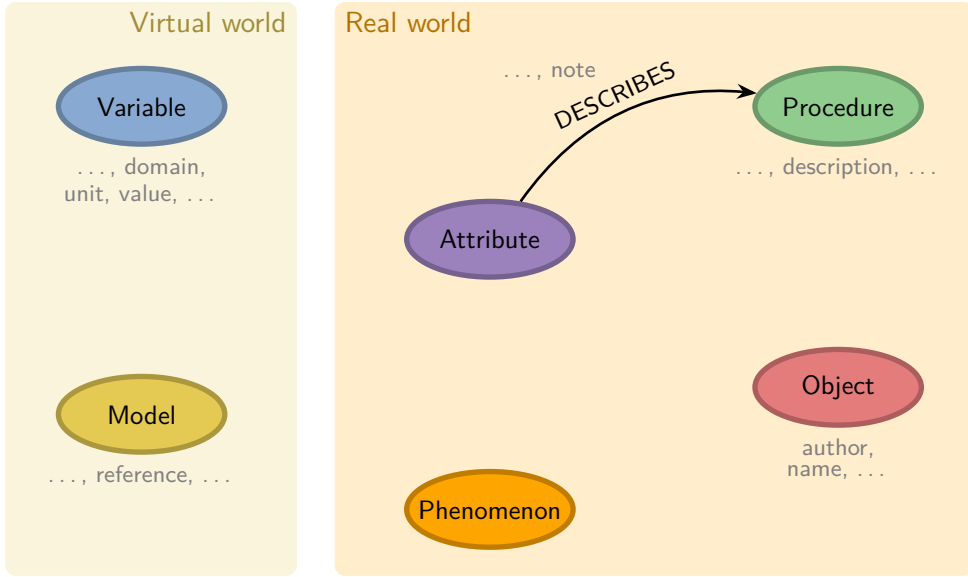
Content & its structure as edges and their type



Content constrained by the foundational ontology



Content as node and edge properties



Representation challenges

- When is variable an input to a model, an output, or both?

Representation challenges

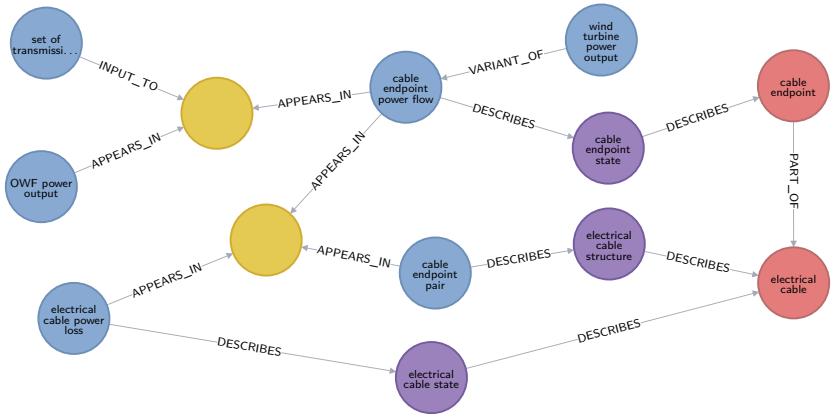
- When is variable an input to a model, an output, or both?
- How to isolate sub-models of a larger model and deal with their variables?

Representation challenges

- When is variable an input to a model, an output, or both?
- How to isolate sub-models of a larger model and deal with their variables?
- A concept should only be represented once;
  what about models that deal with multiple instances of a concept?

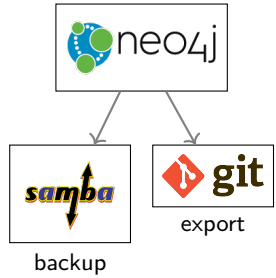
Representation challenges

- When is variable an input to a model, an output, or both?
- How to isolate sub-models of a larger model and deal with their variables?
- A concept should only be represented once;
  what about models that deal with multiple instances of a concept?

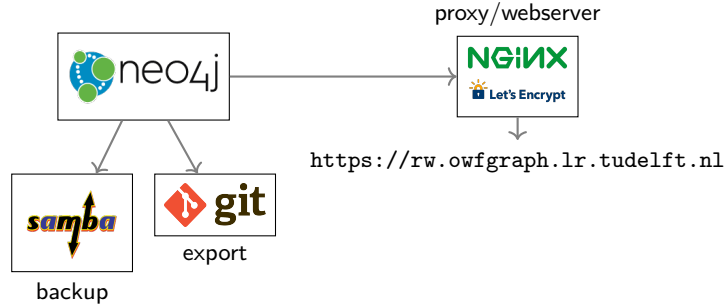


Our practical setup

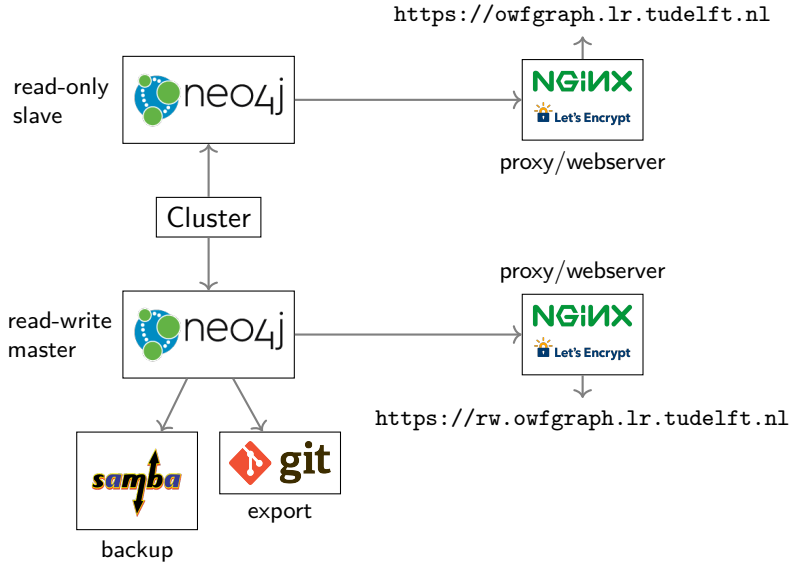
Our practical setup



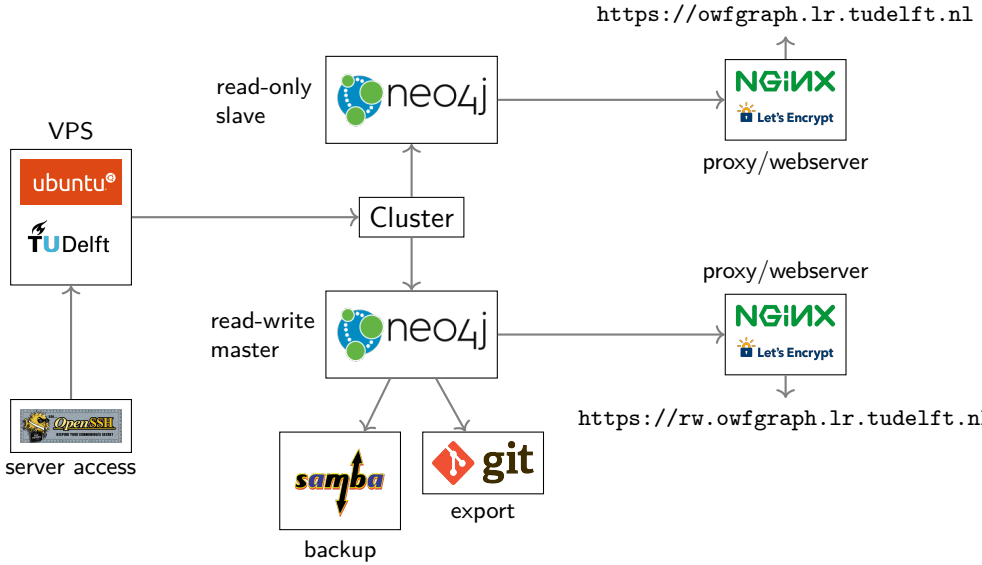
Our practical setup



Our practical setup



Our practical setup



Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.

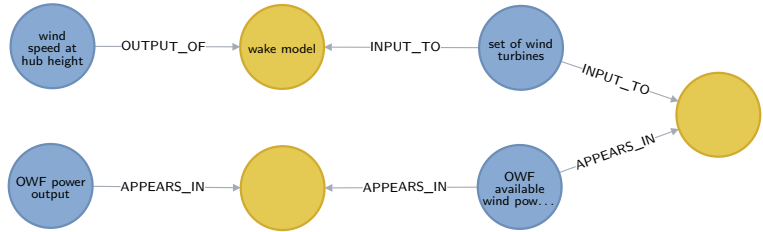
Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- My interest: query the database for possible *paths between variables of interest*.

Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- My interest: query the database for possible *paths between variables of interest*.

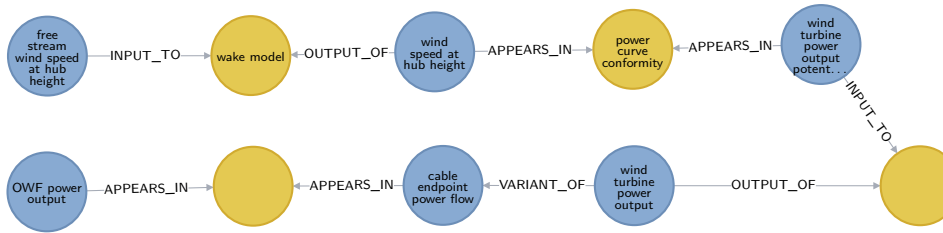
```
match p = shortestPath((w:Variable)-[*0..6]-(f:Variable))
where w.name="wind speed at hub height" and
      f.name="OWF power output" and
      all(n in nodes(p) where n:Model or
                              n:Variable)
return p
```



Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- My interest: query the database for possible *paths between variables of interest*.

```
match p = shortestPath((w:Variable)-[*0..6]-(f:Variable))
where w.name="free stream wind speed at hub height" and
      f.name="OWF power output" and
      all(n in nodes(p) where (n:Model or n:Variable) and
                              not id(n) in [278, 260, 62, 2929])
return p
```



Lessons Learned

- Designing the foundational ontology takes quite a number of iterations and requires experience from adding content.

Lessons Learned

- Designing the foundational ontology takes quite a number of iterations and requires experience from adding content.

- Even with the foundational ontology more-or-less settled, structuring content is sometimes difficult.

Lessons Learned

- Designing the foundational ontology takes quite a number of iterations and requires experience from adding content.
- Even with the foundational ontology more-or-less settled, structuring content is sometimes difficult.
- Adding well-curated content takes time.

Lessons Learned

- Designing the foundational ontology takes quite a number of iterations and requires experience from adding content.
- Even with the foundational ontology more-or-less settled, structuring content is sometimes difficult.
- Adding well-curated content takes time.
- System administration also requires a non-negligible effort.

Lessons Learned

- Designing the foundational ontology takes quite a number of iterations and requires experience from adding content.
- Even with the foundational ontology more-or-less settled, structuring content is sometimes difficult.
- Adding well-curated content takes time.
- System administration also requires a non-negligible effort.

But those are growing pains:
the knowledge base idea is feasible and our implementation works!

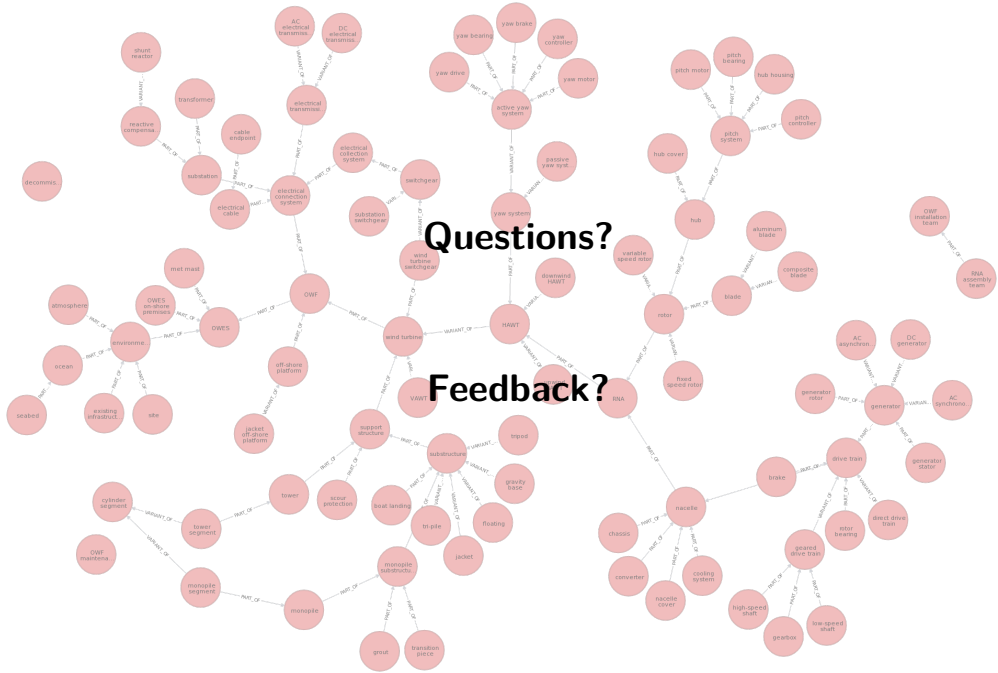
Current & Next Steps

- (Personally) shift from content entry to use.

Current & Next Steps

- (Personally) shift from content entry to use.

- Opening up the database.
  - Now: Read-only version semi-publicly available.
  - Next: get others involved in using and adding content.
    (Learning curve!)
  - Future: build a community to maintain and improve the database.



TUDelft

Live demo – Read-Only

- Surf to `https://owfgraph.lr.tudelft.nl`;
  login 'IEA37', password '. . . '.
- Interface: command line at the top, output canvas below,
  info & control pane at the left.

Live demo – Read-Only

- Surf to `https://owfgraph.lr.tudelft.nl`;
  login 'IEA37', password '. . . '.
- Interface: command line at the top, output canvas below,
  info & control pane at the left.
- Basic query:

  ```
  match (n:Object) return (n) limit 3
  ```

  Explore neighborhood interactively.

Live demo – Read-Only

- Surf to `https://owfgraph.lr.tudelft.nl`;
  login 'IEA37', password '. . . '.
- Interface: command line at the top, output canvas below,
  info & control pane at the left.
- Basic query:

```
match (n:Object) return (n) limit 3
```

  Explore neighborhood interactively.

- Table output:

```
match (n:Object) with n limit 5
return n.name, n.description, n.author
```

Live demo – Read-Only

- Surf to `https://owfgraph.lr.tudelft.nl`;
  login 'IEA37', password '. . . '.
- Interface: command line at the top, output canvas below,
  info & control pane at the left.
- Basic query:

```
match (n:Object) return (n) limit 3
```

  Explore neighborhood interactively.
- Table output:

```
match (n:Object) with n limit 5
return n.name, n.description, n.author
```

- More involved queries:

```
match p = (:Object {name:"monopile"})-[*]->()
return p
```

Live demo – Read-Write

- Surf to `https://rw.owfgraph.lr.tudelft.nl`.
  (Currently only Sebastian & I have access.)
- Same interface, but now also write—and delete—queries are enabled.

Live demo – Read-Write

- Surf to `https://rw.owfgraph.lr.tudelft.nl`.
  (Currently only Sebastian & I have access.)
- Same interface, but now also write—and delete—queries are enabled.
- Creation (merging):

```
match (a:Attribute {name:"wind"})
merge (a)<-[:PART_OF]-(b {name:"wind color"})
return a, b
```

Live demo – Read-Write

- Surf to `https://rw.owfgraph.lr.tudelft.nl`.
  (Currently only Sebastian & I have access.)
- Same interface, but now also write—and delete—queries are enabled.
- Creation (merging):

```
match (a:Attribute {name:"wind"})
merge (a)<-[:PART_OF]-(b {name:"wind color"})
return a, b
```

- Setting and removing labels and properties:

```
match (b {name:"wind color"})
set b:Attribute, b.author="killroy"
remove b.name
return b
```

Live demo – Read-Write

- Surf to `https://rw.owfgraph.lr.tudelft.nl`.
  (Currently only Sebastian & I have access.)
- Same interface, but now also write—and delete—queries are enabled.
- Creation (merging):

```
match (a:Attribute {name:"wind"})
merge (a)<-[:PART_OF]-(b {name:"wind color"})
return a, b
```

- Setting and removing labels and properties:

```
match (b {name:"wind color"})
set b:Attribute, b.author="killroy"
remove b.name
return b
```

- Deleting nodes and edges:

```
match (b {author:"killroy"}) detach delete b
```