OWFgraph

a graph database for the off-shore wind farm domain

Erik Quaeghebeur

joint work with Sebastian Sanchez & Michiel Zaaijer

TU Delft Wind Energy Presentation

8 November 2016

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 consideted 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.

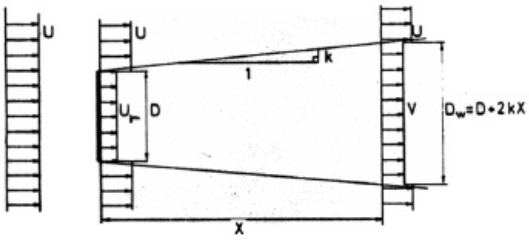


Fig. 2. Schematic view of wake description

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

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

The wake velocity is found by the expression

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

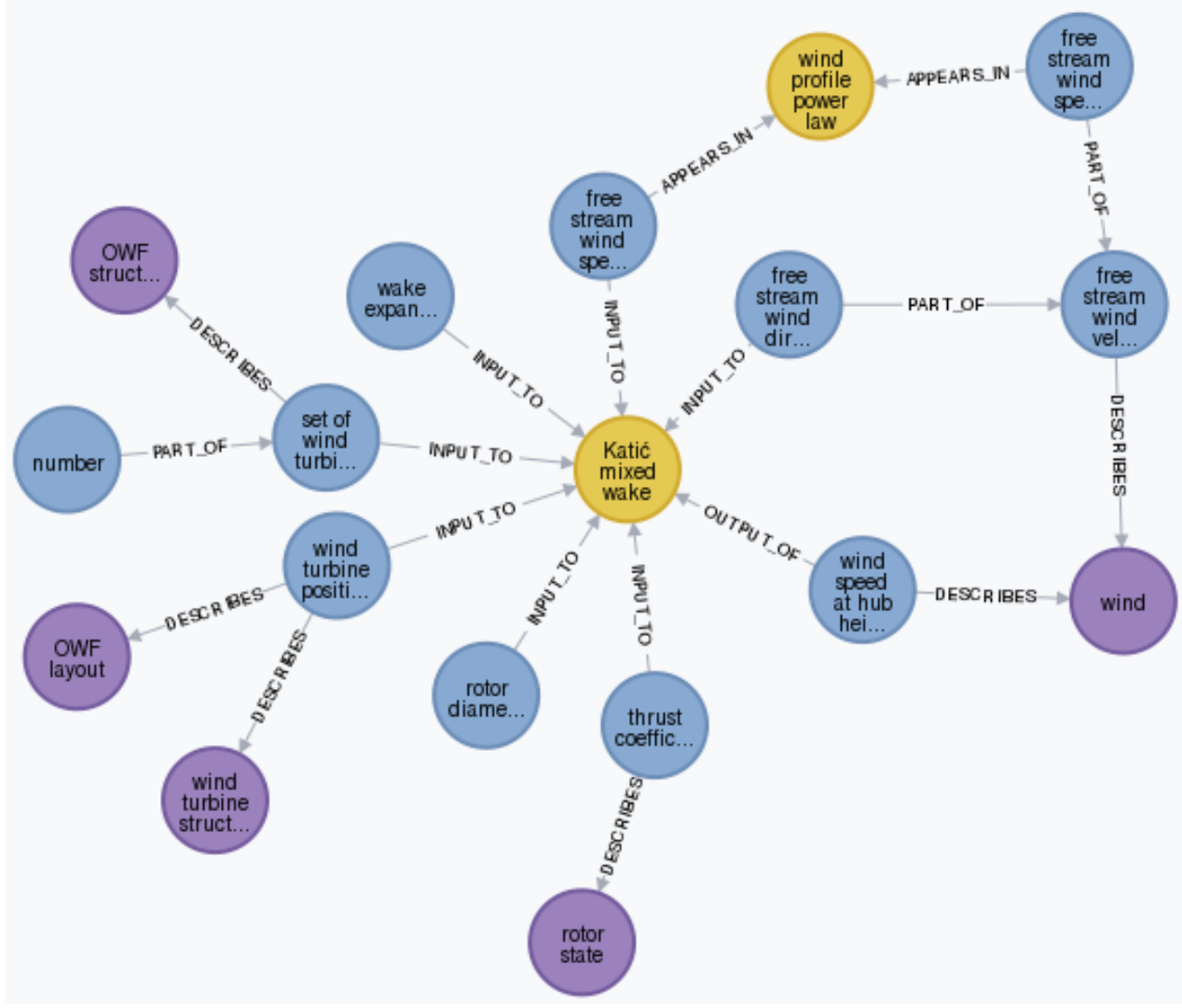
a is defined as the initial velocity deficit $1-U_r/U$ but can also be expressed as

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

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.



Overview

Goal Why do we want to build a graph database for the offshore wind farm domain?

Overview

Goal Why do we want to build a graph database for the offshore wind farm domain?

Content What do we put into the database?

Overview

Goal Why do we want to build a graph database for the offshore wind farm domain?

Content What do we put into the database?

Structure How do we structure the content in the database?

Overview

Goal Why do we want to build a graph database for the offshore wind farm domain?

Content What do we put into the database?

Structure How do we structure the content in the database?

Implementation What graph database program do we use?

Overview

Goal Why do we want to build a graph database for the offshore wind farm domain?

Content What do we put into the database?

Structure How do we structure the content in the database?

Implementation What graph database program do we use?

Installation What is our setup?

Overview

Goal	Why do we want to build a graph database for the offshore wind farm domain?
Content	What do we put into the database?
Structure	How do we structure the content in the database?
Implementation	What graph database program do we use?
Installation	What is our setup?
Querying	How do we add and extract data?

Overview

Goal	Why do we want to build a graph database for the offshore wind farm domain?
Content	What do we put into the database?
Structure	How do we structure the content in the database?
Implementation	What graph database program do we use?
Installation	What is our setup?
Querying	How do we add and extract data?
Lessons learned	What is different from what we expected?

Goal – context within EUROS program

Project 3 Wind Farm Design Optimization

Work Package 3.2 Uncertainty Model of Wind Farms

Goal – context within EUROS program

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 – context within EUROS program

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 – context within EUROS program

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 – context within EUROS program

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

Content – concept types

The physical world Objects, Procedures, Attributes, and Phenomena

```
$ match (o:Object) return o, rand() as r order by r limit 3
```

*(3) **Object**(3)


```
$ match (o:Procedure) return o, rand() as r order by r limit 3
```

*(3) **Procedure**(3)


```
$ match (a:Attribute) return a, rand() as r order by r limit 3
```

*(3) **Attribute**(3)


```
$ match (p:Phenomenon) return p
```

(no rows)

Content – concept types

The physical world Objects, Procedures, Attributes, and Phenomena

```
$ match (o:Object) return o, rand() as r order by r limit 3
```


```
$ match (o:Procedure) return o, rand() as r order by r limit 3
```


```
$ match (a:Attribute) return a, rand() as r order by r limit 3
```

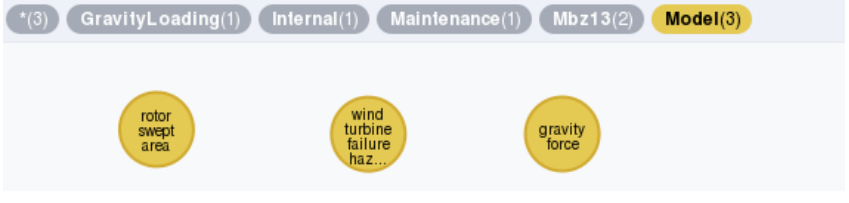

```
$ match (p:Phenomenon) return p
```

(no rows)

The mathematical world Variables and Models

```
$ match (v:Variable) return v, rand() as r order by r limit 3
```


```
$ match (m:Model) where exists(m.name) return m, rand() as r order by r limit 3
```



Structure – Graph representation

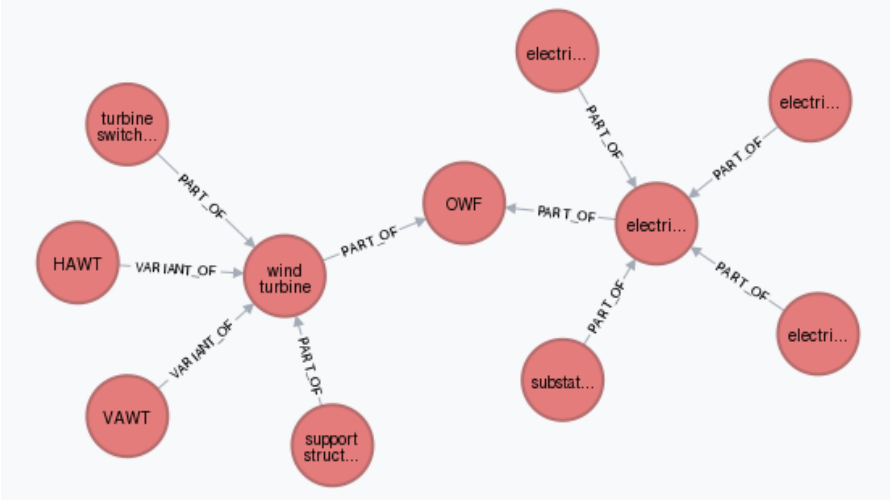
Graph representation as structured domain knowledge representation:

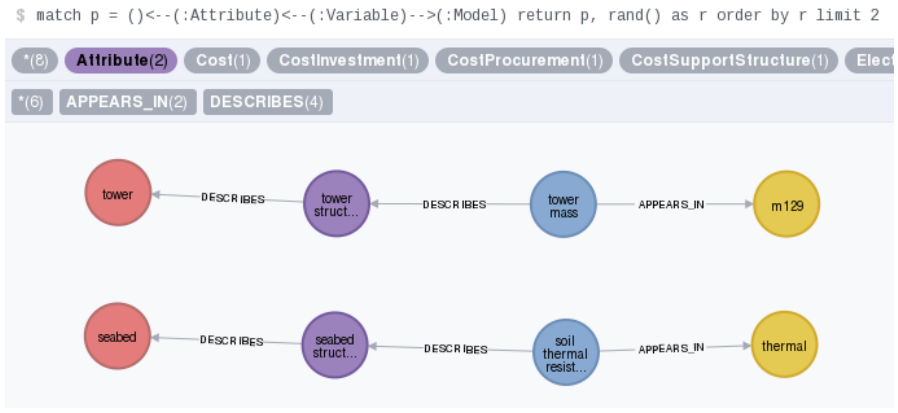
- *concepts* as nodes;
- *interrelations* as edges.

Structure – Graph representation

Graph representation as structured domain knowledge representation:

- *concepts* as nodes;
- *interrelations* as edges.

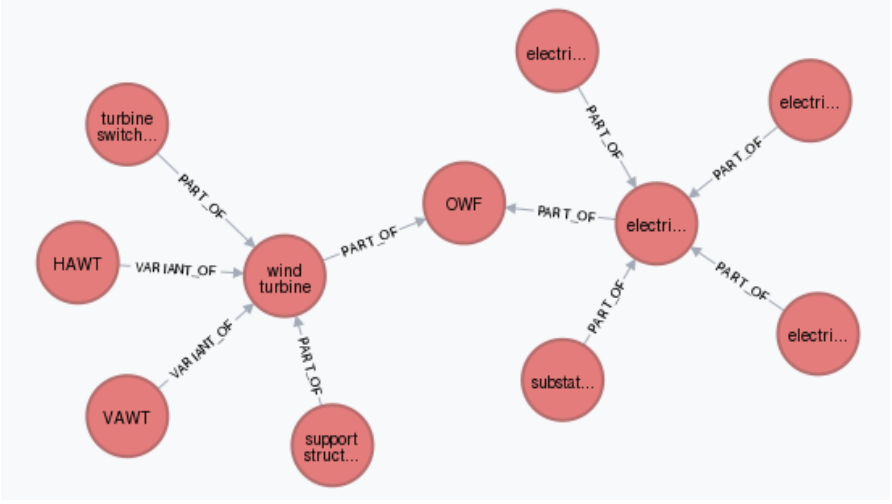


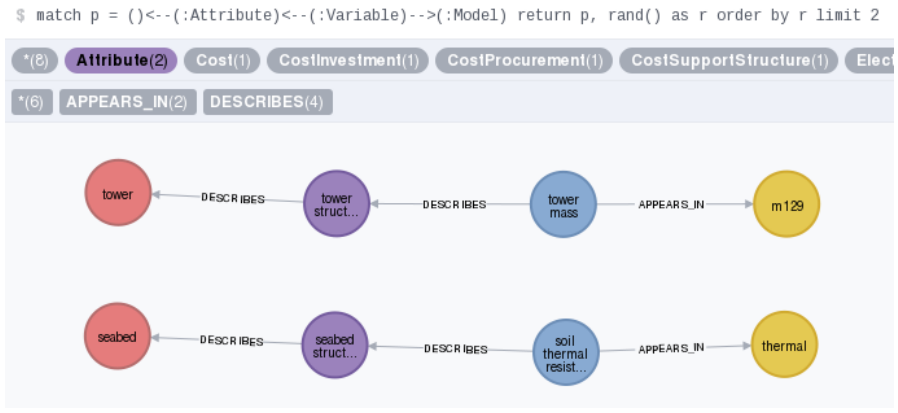


Structure – Graph representation

Graph representation as structured domain knowledge representation:

- *concepts* as nodes;
- *interrelations* as edges.

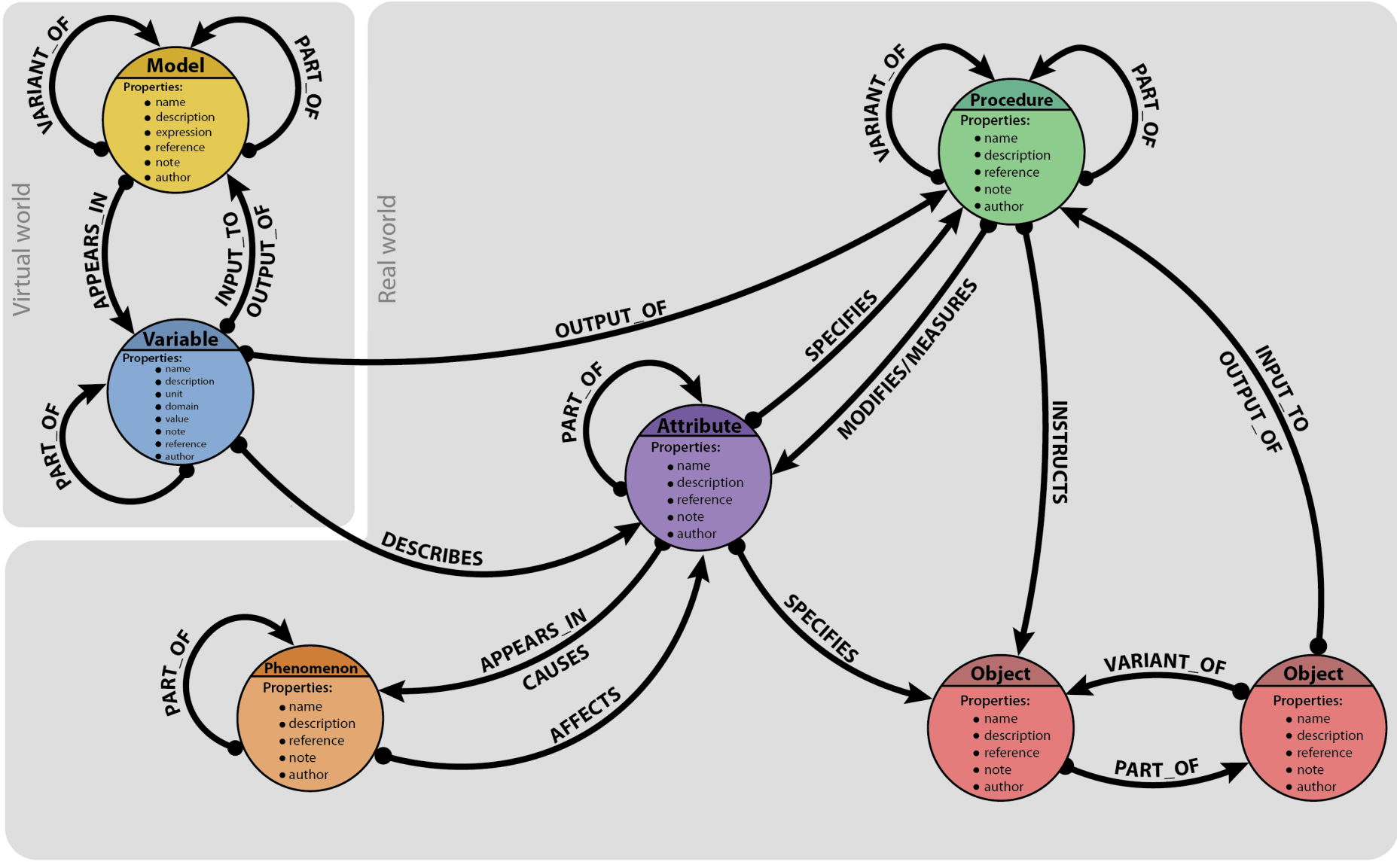




We need a 'foundational ontology' for our knowledge graph:

- classification of the domain's concepts and relationships,
- small enough to be manageable,
- large enough to be sufficiently expressive.

Structure – The foundational ontology



(Drawing courtesy of Sebastian Sanchez.)

Content – Types, Labels, and Properties

Content is added to the graph by

- giving edges a *type*,

```
$ match ()-[r]-() with type(r)...
```

t	c
APPEARS_IN	1328
DESCRIBES	542
PART_OF	408
INPUT_TO	168
VARIANT_OF	150
OUTPUT_OF	76
INSTRUCTS	8
MODIFIES	4

Content – Types, Labels, and Properties

Content is added to the graph by

- giving edges a *type*,
- giving nodes zero or more *labels*, and

```
$ match ()-[r]-() with type(r)...
```

t	c
APPEARS_IN	1328
DESCRIBES	542
PART_OF	408
INPUT_TO	168
VARIANT_OF	150
OUTPUT_OF	76
INSTRUCTS	8
MODIFIES	4

```
$ match (n) with labels(n) as l, count(n) ...
```

l	c
[Variable, Mbz13]	293
[Variable]	124
[Attribute]	102
[Object]	93
[Model, Maintenance, Mbz13, Internal]	48
[Model]	42
[Variable, Mbz13, Internal]	40
[Model, Electricity, Mbz13]	24
[Model, Mechanics, Mbz13]	15
[Model, SiteConditions, Mbz13]	8
[Model, Hydrology, Mbz13]	6
[Model, Cost, CostOperationMaintenance, CostMaintenance, Mbz13]	6
[Procedure]	6

Content – Types, Labels, and Properties

Content is added to the graph by

- giving edges a *type*,
- giving nodes zero or more *labels*, and
- attaching any number of *properties*—key-value pairs—to nodes.

```
$ match ()-[r]-() with type(r)...
```

t	c
APPEARS_IN	1328
DESCRIBES	542
PART_OF	408
INPUT_TO	168
VARIANT_OF	150
OUTPUT_OF	76
INSTRUCTS	8
MODIFIES	4

```
$ match (n) with labels(n) as l, count(n) ...
```

l	c
[Variable, Mbz13]	293
[Variable]	124
[Attribute]	102
[Object]	93
[Model, Maintenance, Mbz13, Internal]	48
[Model]	42
[Variable, Mbz13, Internal]	40
[Model, Electricity, Mbz13]	24
[Model, Mechanics, Mbz13]	15
[Model, SiteConditions, Mbz13]	8
[Model, Hydrology, Mbz13]	6
[Model, Cost, CostOperationMaintenance, CostMaintenance, Mbz13]	6
[Procedure]	6

```
$ match (m:Model {name:"Katić mixed wake model"})--(v:Variable) where v.name contains "wake"...
```

m

reference	I. Katić, J. Højstrup, and N. O.Jensen. A simple model for cluster efficiency. In Proceedings of EWEC'86, volume 1, pages 407–410. Rome, 1987. John Twidell (ed.) Offshore Wind Power, Chapter 4, Eq. 21. PhD thesis Michiel Zaaijer, p.237.
note	We have not described the model for determining the upstream wind tubines nor mentioned mirror turbines to take into account the ground effect.
author	sebastian,equaeghebeur
name	Katić mixed wake model

v

author	sebastian,equaeghebeur
domain	real
name	wake expansion factor
description	Linear coefficient with which the wake diameter increases downwind according to the Jensen model. Also called wake decay coefficient. Denoted k.

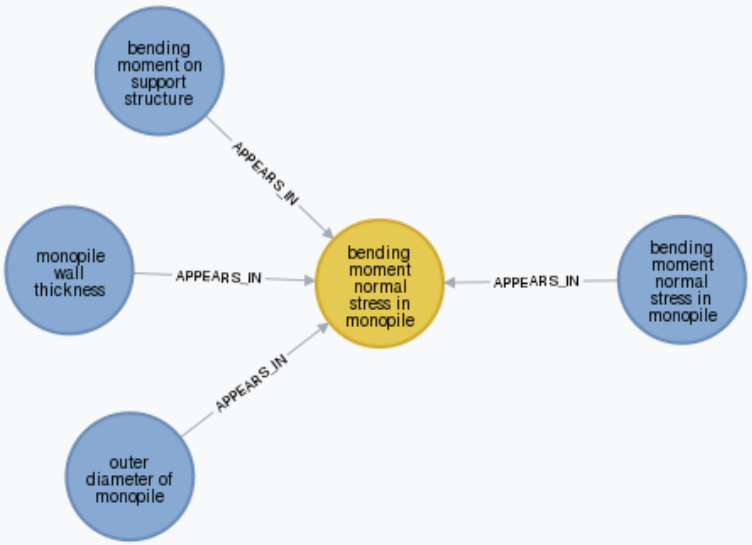
Content & Structure – Representation challenges

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

```
$ match p = (:Model {name:"bending moment normal stress in monopile"})--(:Variable) return p
```

*(5) Mbz13(4) Mechanics(1) **Model**(1) **Variable**(4)

*(4) APPEARS_IN(4)

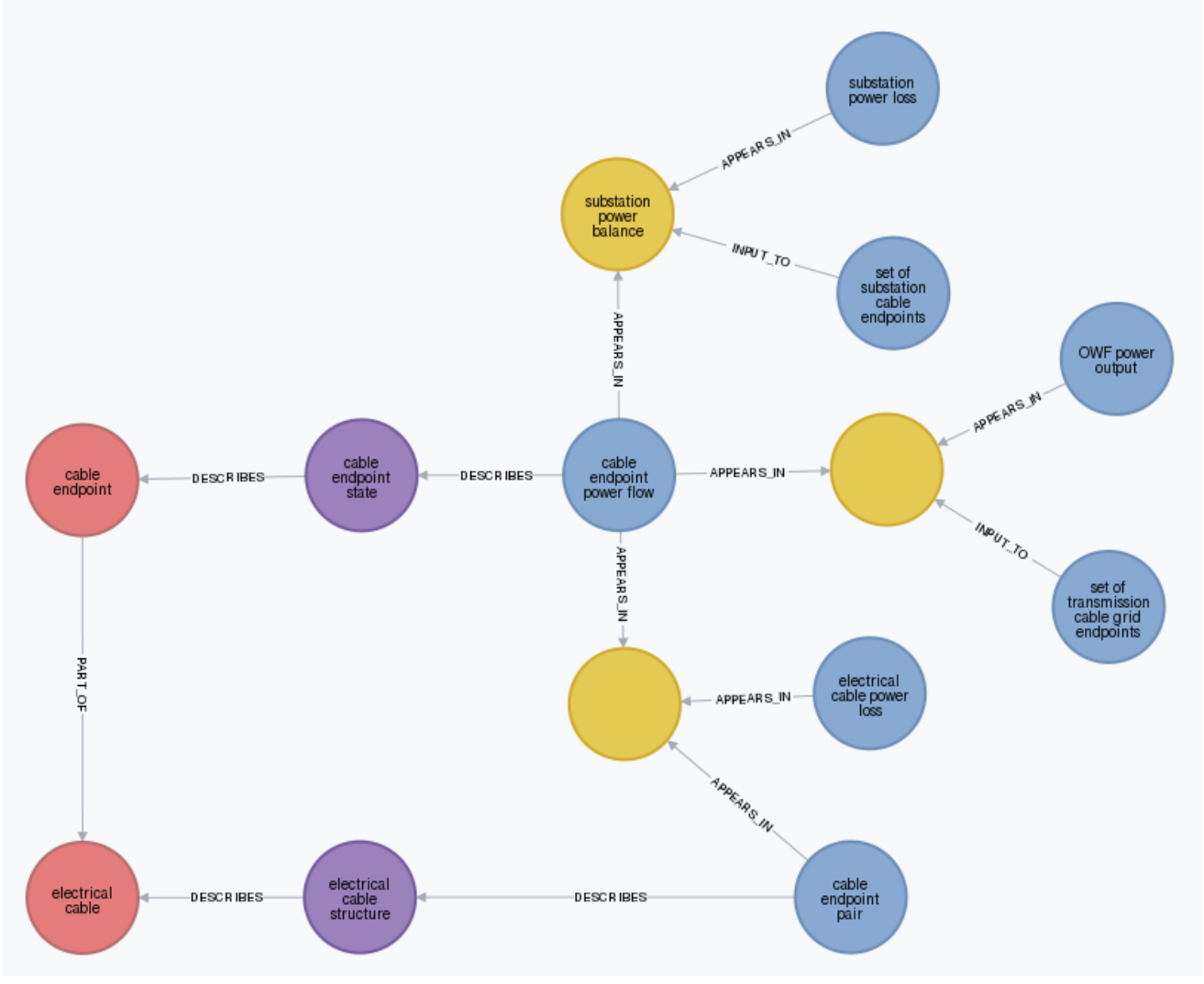



```
$ match (:Model {name:"bending moment normal...
```

v.name	r.output
bending moment normal stress in monopile	true
outer diameter of monopile	false
bending moment on support structure	false
monopile wall thickness	false

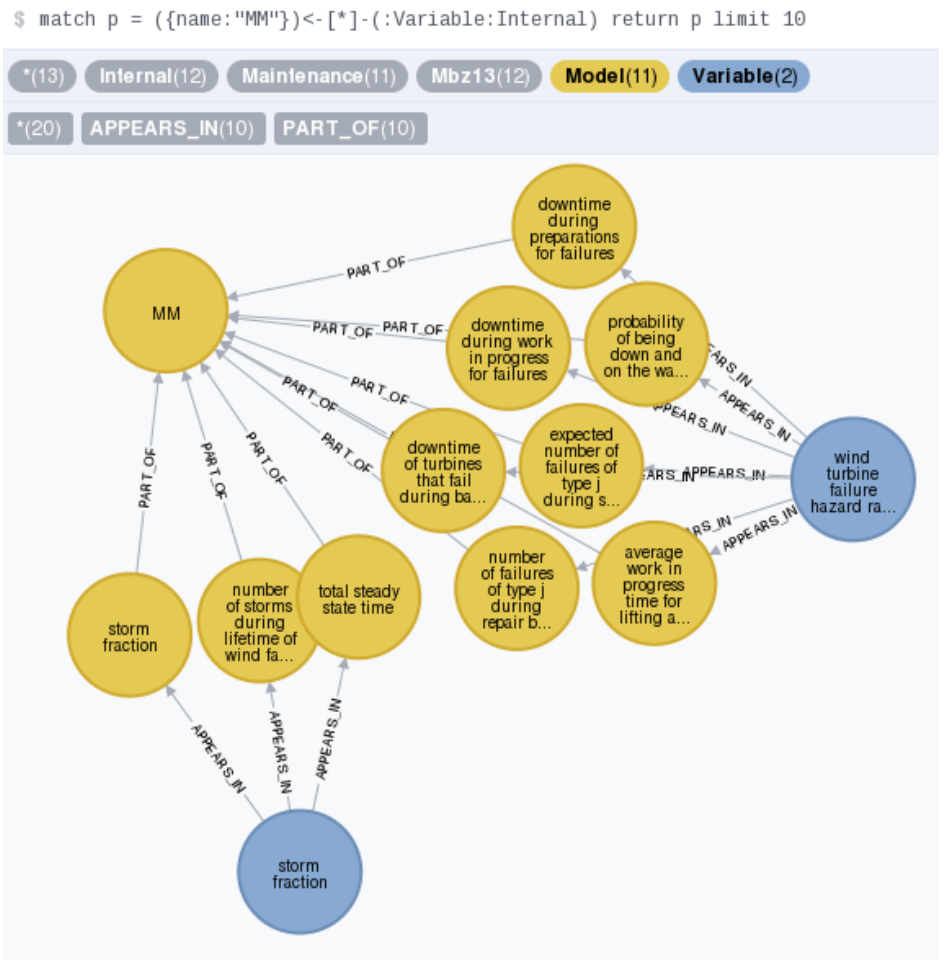
Content & Structure – Representation challenges

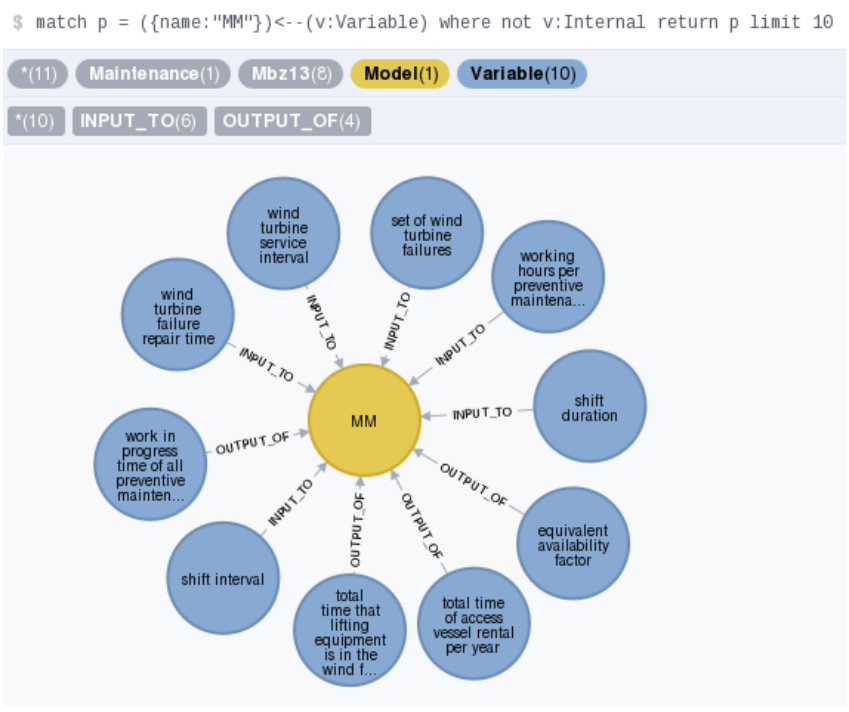
A concept should only be represented once; what about models that deal with multiple instances of a concept?



Content & Structure – Representation challenges

How to isolate sub-models of a larger model and deal with the variables involved?

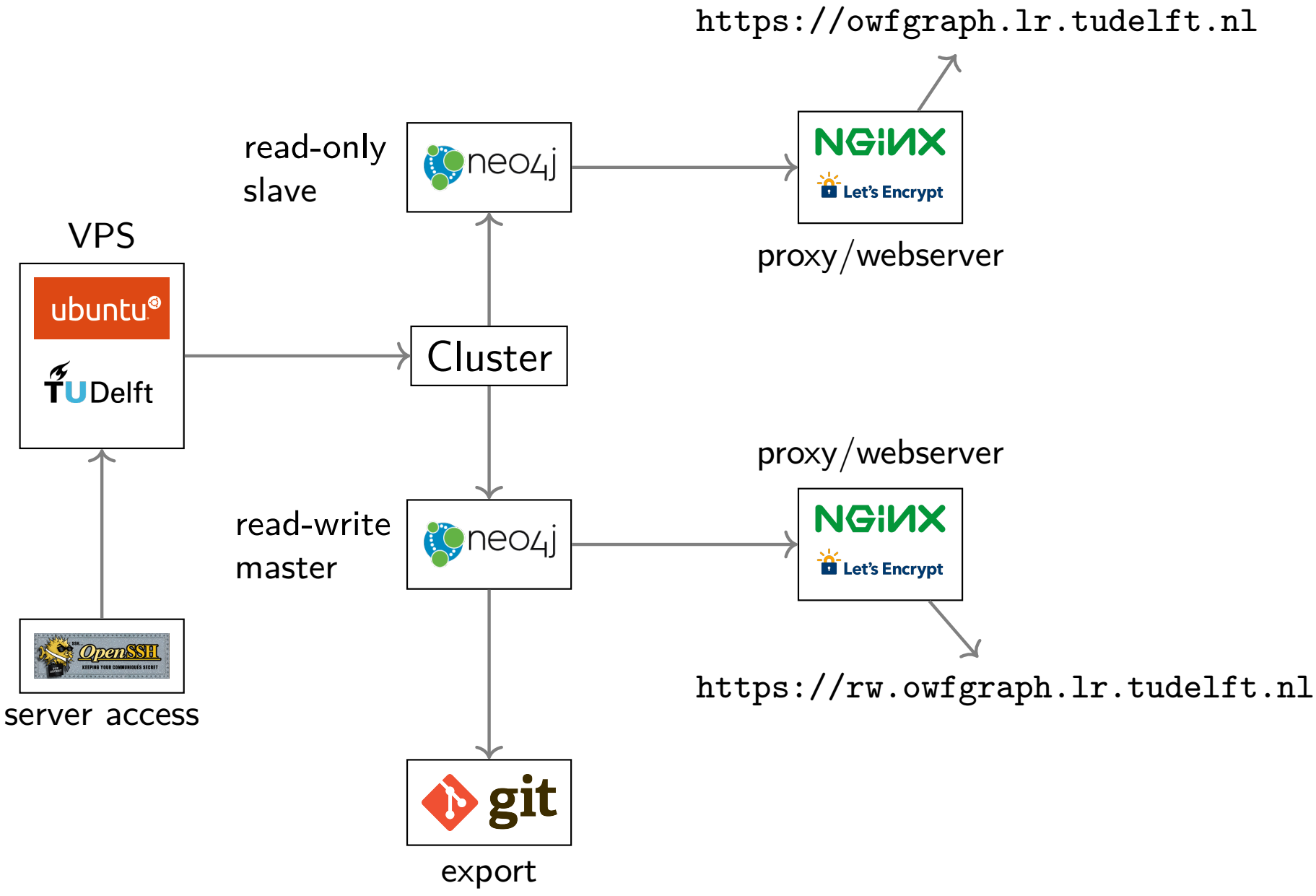




Implementation

- Native property graph database
- Java
- ‘driver’ (or wrapper) for many major languages (e.g., Python)
- Web interface for data entry and querying
- Shell access for importing and exporting data
- ‘Community edition’ (GPLv3) with limitations
- ‘Enterprise edition’ (AGLPv3) with clustering, live backups, etc.
- Mature and widely used (so free ‘forum-based’ support works)

Installation – Our setup



Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.

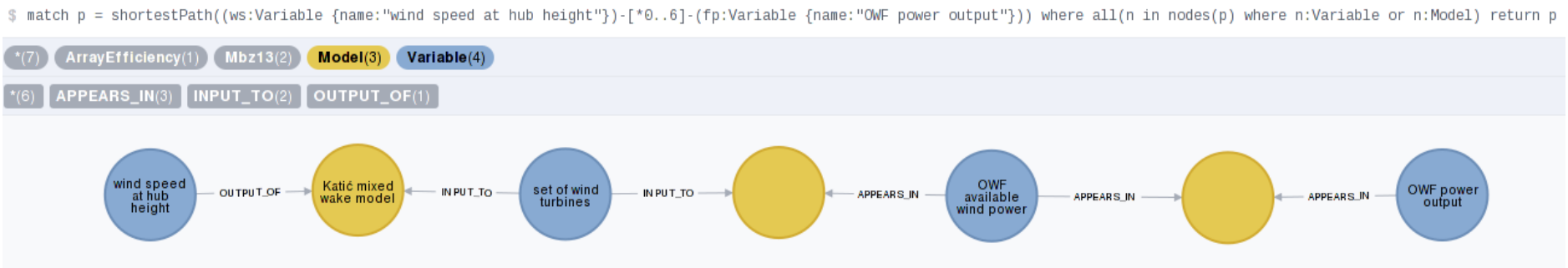
Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.
- My interest is to query the database for possible *paths between variables of interest*.
- Such questions require manual query tweaking:

Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.
- My interest is to query the database for possible *paths between variables of interest*.
- Such questions require manual query tweaking:

Example: path between ‘wind speed’ and ‘OWF power output’.

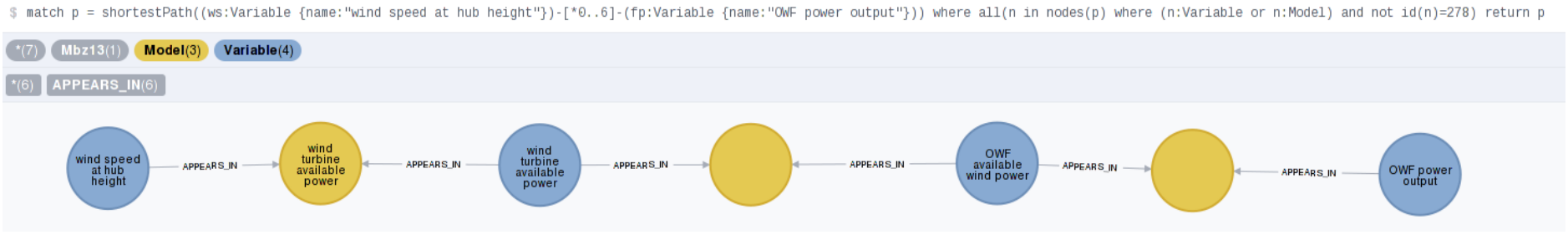


(via ‘set of wind turbines’)

Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.
- My interest is to query the database for possible *paths between variables of interest*.
- Such questions require manual query tweaking:

Example: path between ‘wind speed’ and ‘OWF power output’.

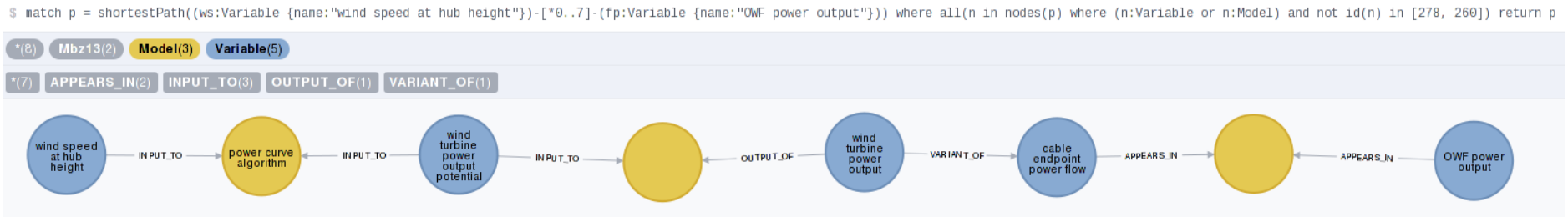


(via ‘available power’)

Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.
- My interest is to query the database for possible *paths between variables of interest*.
- Such questions require manual query tweaking:

Example: path between ‘wind speed’ and ‘OWF power output’.

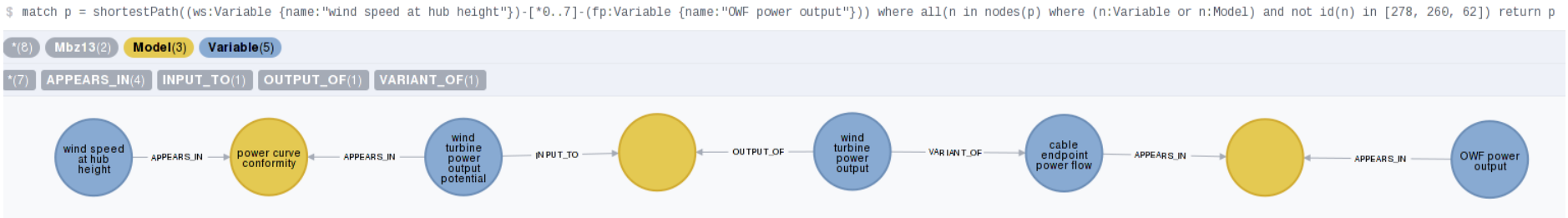


(via ‘power curve algorithm’)

Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.
- My interest is to query the database for possible *paths between variables of interest*.
- Such questions require manual query tweaking:

Example: path between ‘wind speed’ and ‘OWF power output’.

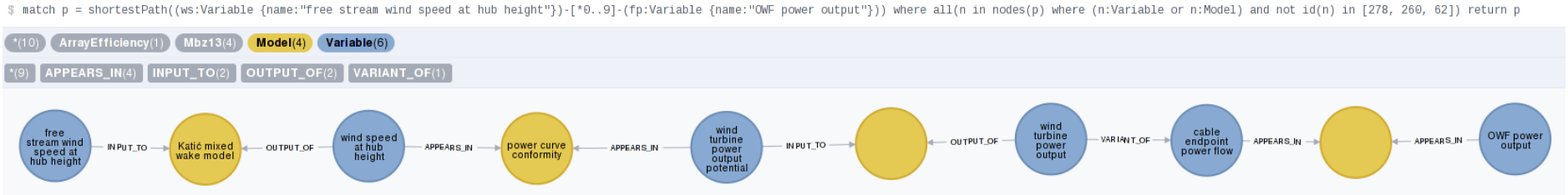


(looks good, but ‘wind speed’ is not free stream)

Querying

- Queries—questions asked or instructions given—are formulated using *Cypher*.
- All screenshots are output resulting from queries.
- My interest is to query the database for possible *paths between variables of interest*.
- Such questions require manual query tweaking:

Example: path between ‘wind speed’ and ‘OWF power output’.



(ok, a good path, now tweak further to find alternatives. . .)

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 often 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 often 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 often 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 often difficult.
- Adding well-curated content takes time.
- System administration also requires a non-negligible effort.

But overall very interesting and quite useful.

Current & Next Steps

- Focus shift from content entry to use.

Current & Next Steps

- Focus shift from content entry to use.
- Make the database semi-public. (learning curve is an issue.)
- Open up and promote for other uses as well.

Live demo – Read-Only

- Surf to `https://owfgraph.lr.tudelft.nl`;
  login ‘Euros’, 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 ‘Euros’, 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 'Euros', 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 'Euros', 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
```

Questions?

Feedback?