The Semantically Reflected Digital Twin

ICTAC Summer School Tutorial 2022

Einar Broch Johnsen
Silvia Lizeth Tapia Tarifa
Rudolf Schlatte
Eduard Kamburjan

University of Oslo

Today

- **Part I Digital Twins Introduction: Concepts and Engineering Perspectives**
- **Part II** Modelling Knowledge using Semantic Technologies

Tomorrow:

- **Part III** Modelling Physical Systems
- **Part IV** Semantically Reflected Digital Twins

Digital Twins — The Hype

Digital twins are an emerging, enabling technology for industry to transition to the next level of digitisation

Digital Twins — The Hype

Digital twins are an emerging, enabling technology for industry to transition to the next level of digitisation

Increasing traction

1. Digital twins: a means to **understand** and **control** assets in nature, in industry, and in society at large
2. Companies increasingly create digital twins of their physical assets

Digital Twins — The Hype

Digital twins are an emerging, enabling technology for industry to transition to the next level of digitisation

Increasing traction

1. Digital twins: a means to **understand** and **control** assets in nature, in industry, and in society at large
2. Companies increasingly create digital twins of their physical assets

Success stories

1. GE experienced 5–7% increase of energy production from digitizing wind farms
2. Johns Hopkins Hospital's centre for clinical logistics reported 80% reduction of operating theatre holds due to delays
3. For the Johan Sverdrup oil field, digital twin innovations have boosted earnings by $216 million in one year

Digital Twins: Emerging Engineering Discipline

- DTs originally conceived at NASA for the space program.
- They have emerged as an engineering discipline, based on **best practices**

NASA's definition of a DT

"an **integrated multi-physics, multi-scale**, *probabilistic simulation of a vehicle or system that uses the* **best available** *physical models, sensor updates, fleet history, etc., to* **mirror the life of its flying twin**. *It is* **ultra-realistic** *and may consider one or more important and interdependent vehicle systems"*

NASA Modeling, Simulation, Information Tech. & Processing Roadmap, 2010

DTs & Models

Is a digital twin just another word for "model"?

DTs & Models

Is a digital twin just another word for "model"?

Is a digital twin just another word for "control system"?

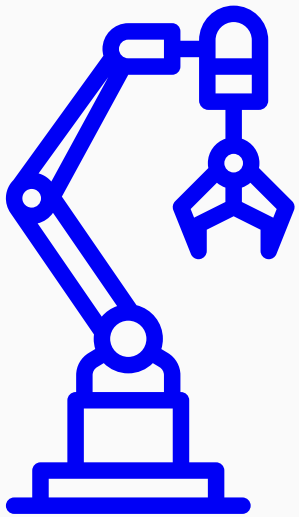
DTs & Models

Is a digital twin just another word for "model"?

Is a digital twin just another word for "control system"?

A digital twin integrates aspects of models and control systems

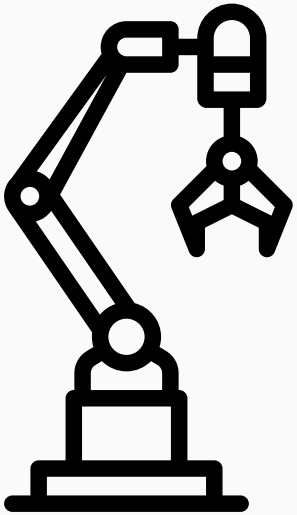
What is a Digital Twin?

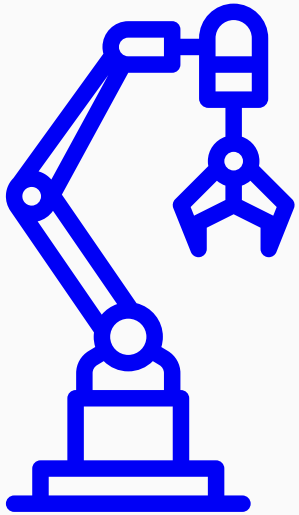


Model

What is a Digital Twin?

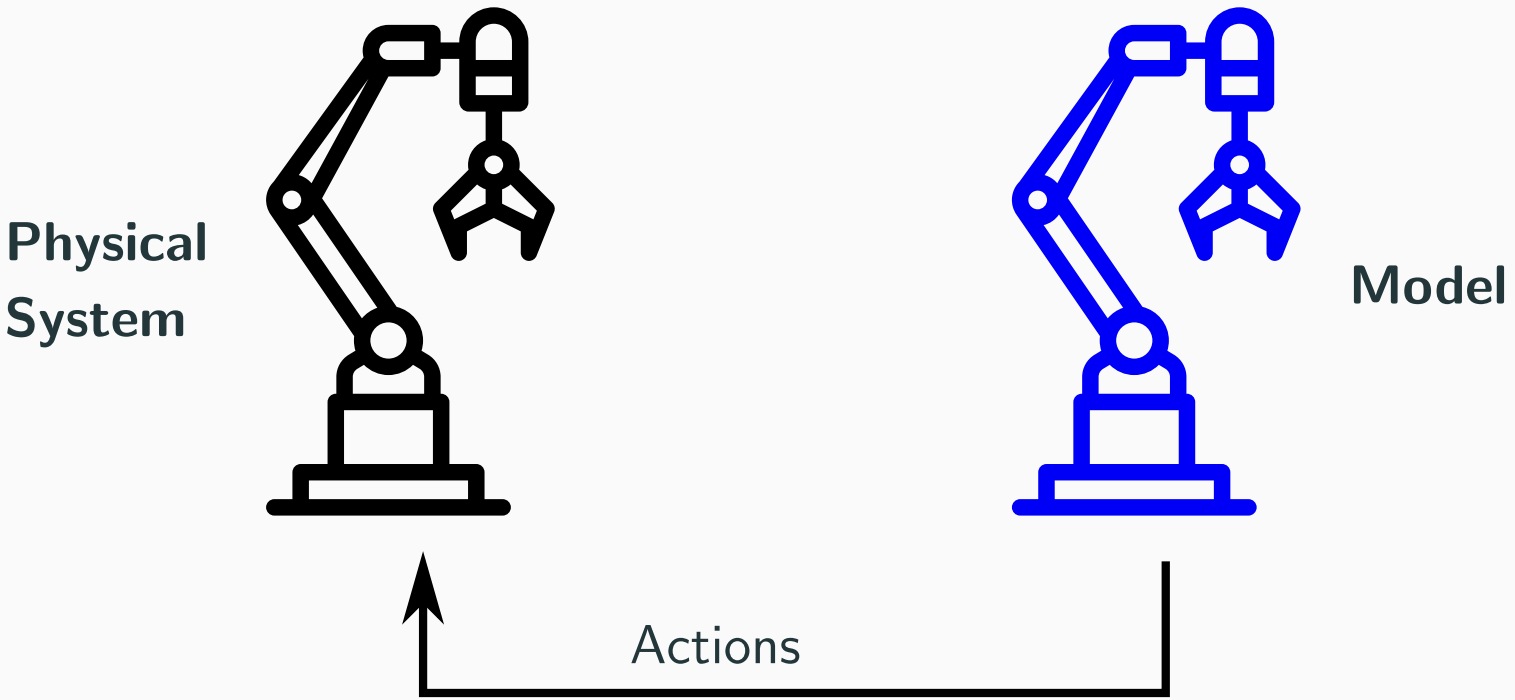
Physical System



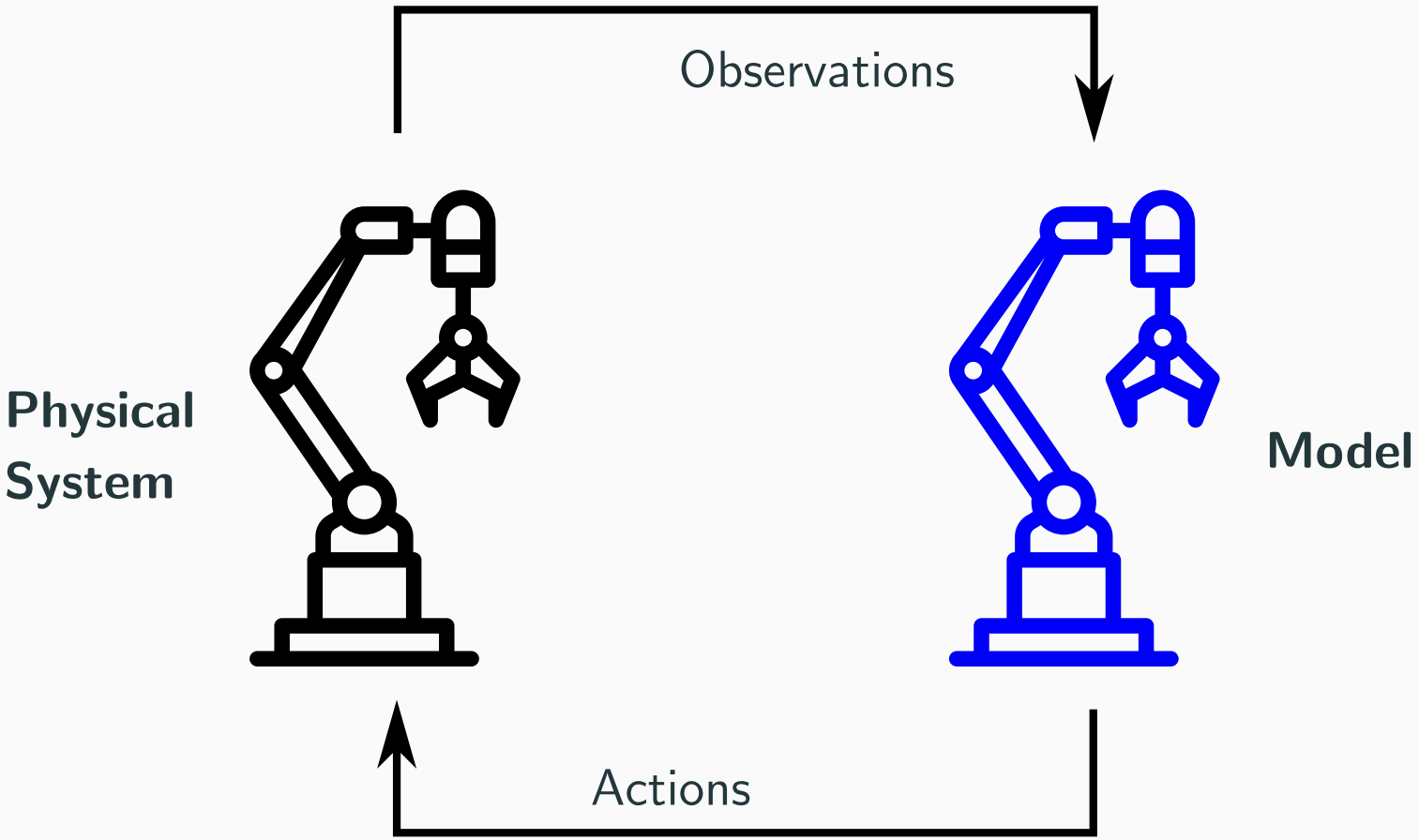


Model

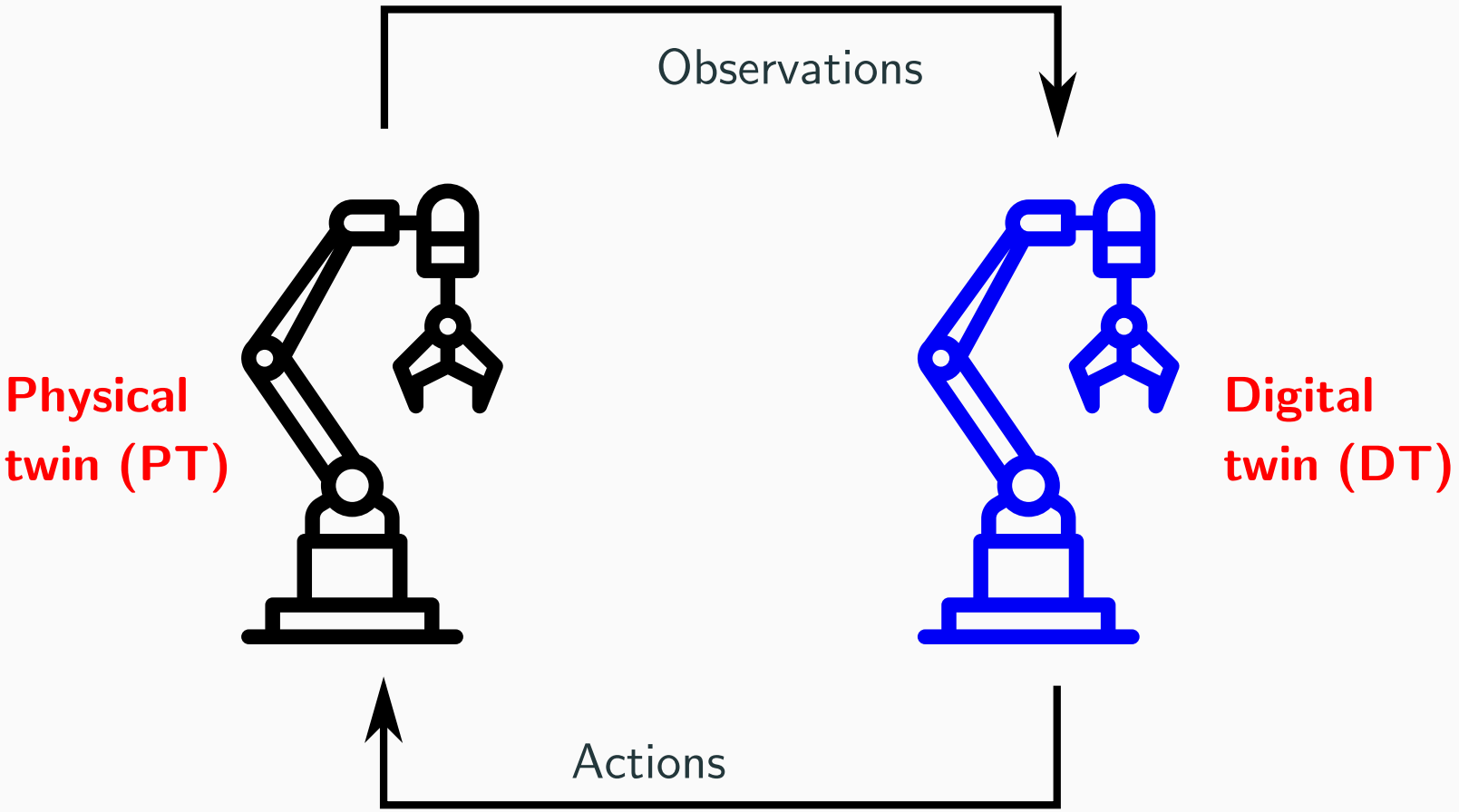
What is a Digital Twin?



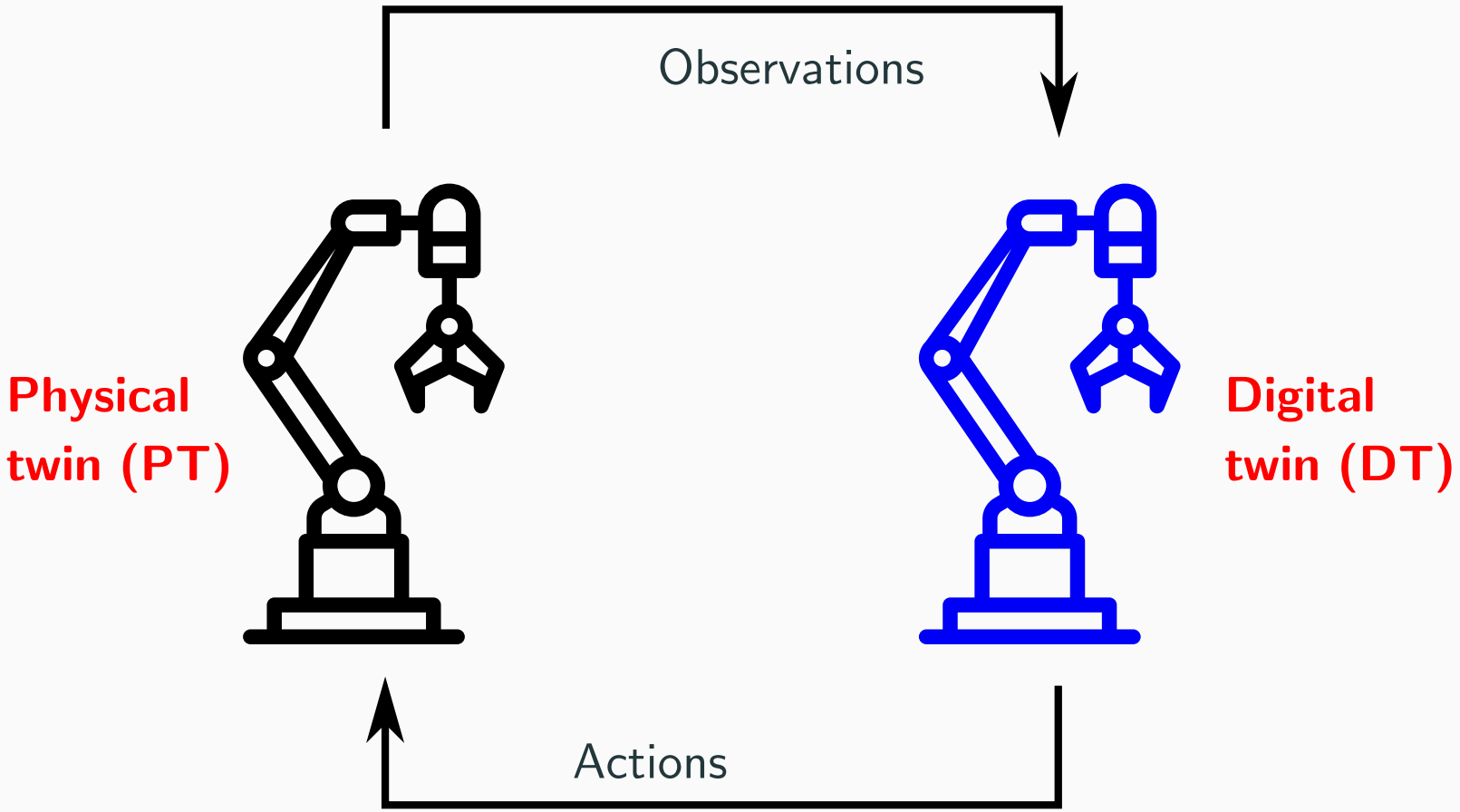
What is a Digital Twin?



What is a Digital Twin?



What is a Digital Twin?



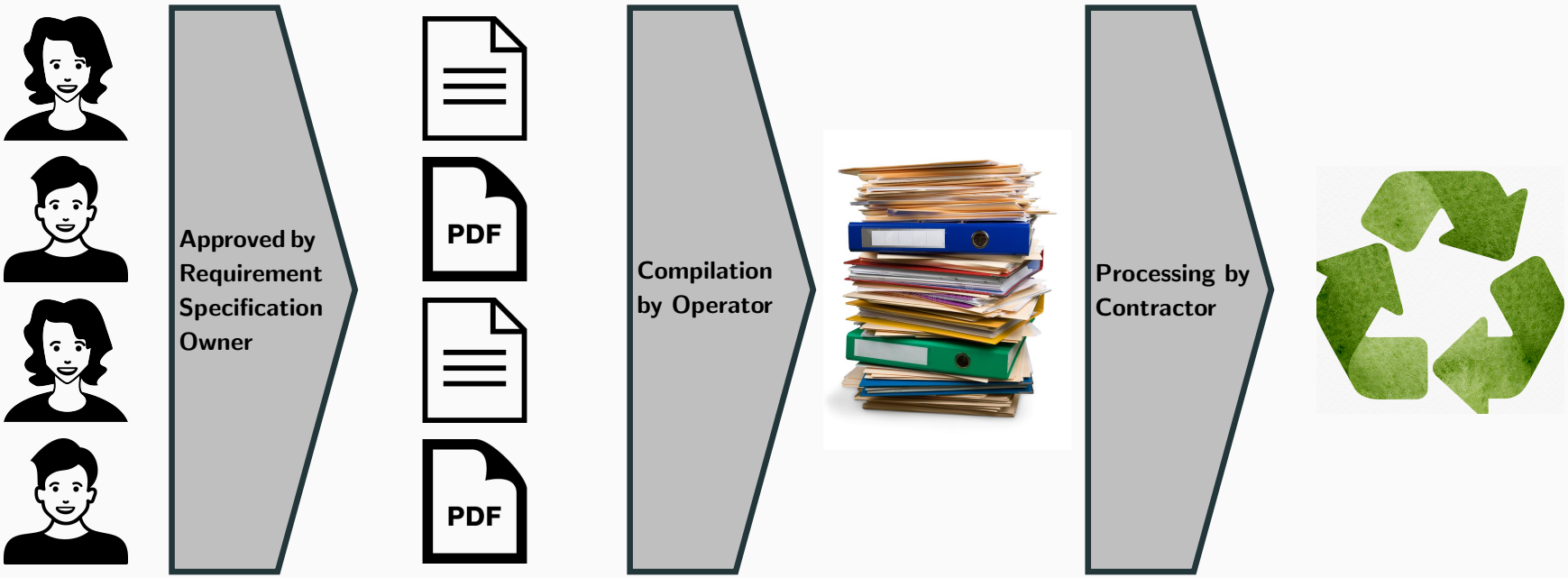
Digital Twin

1. DT is a live replica of a physical system (PT)
2. DT is connected to PT in near real-time via data streams

Lifecycle Management

Digital Thread: The Digital Twin Evolves in Tandem with the Asset

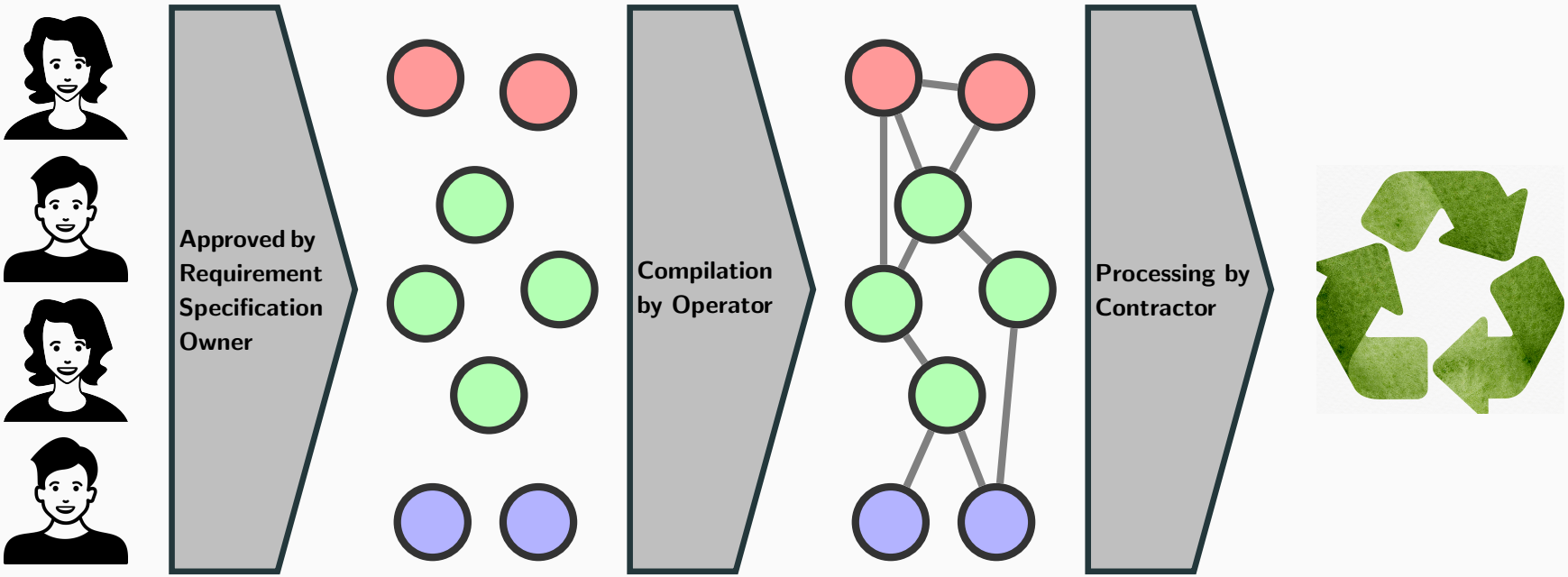
1. Connects the designs, requirements and software that go into the system represented by the DT
2. Connects the different phases of the system to the DT: design, development, operation, decommissioning, ...



Lifecycle Management

Digital Thread: The Digital Twin Evolves in Tandem with the Asset

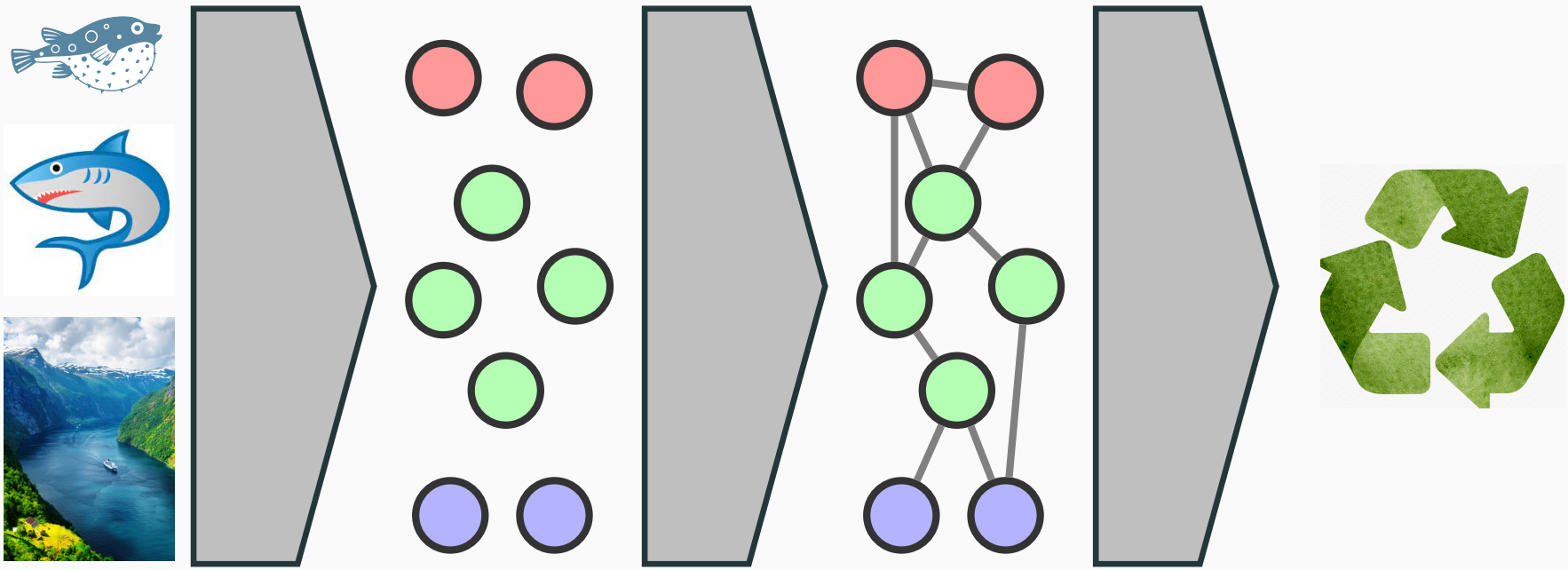
1. Connects the designs, requirements and software that go into the system represented by the DT
2. Connects the different phases of the system to the DT: design, development, operation, decommissioning, ...



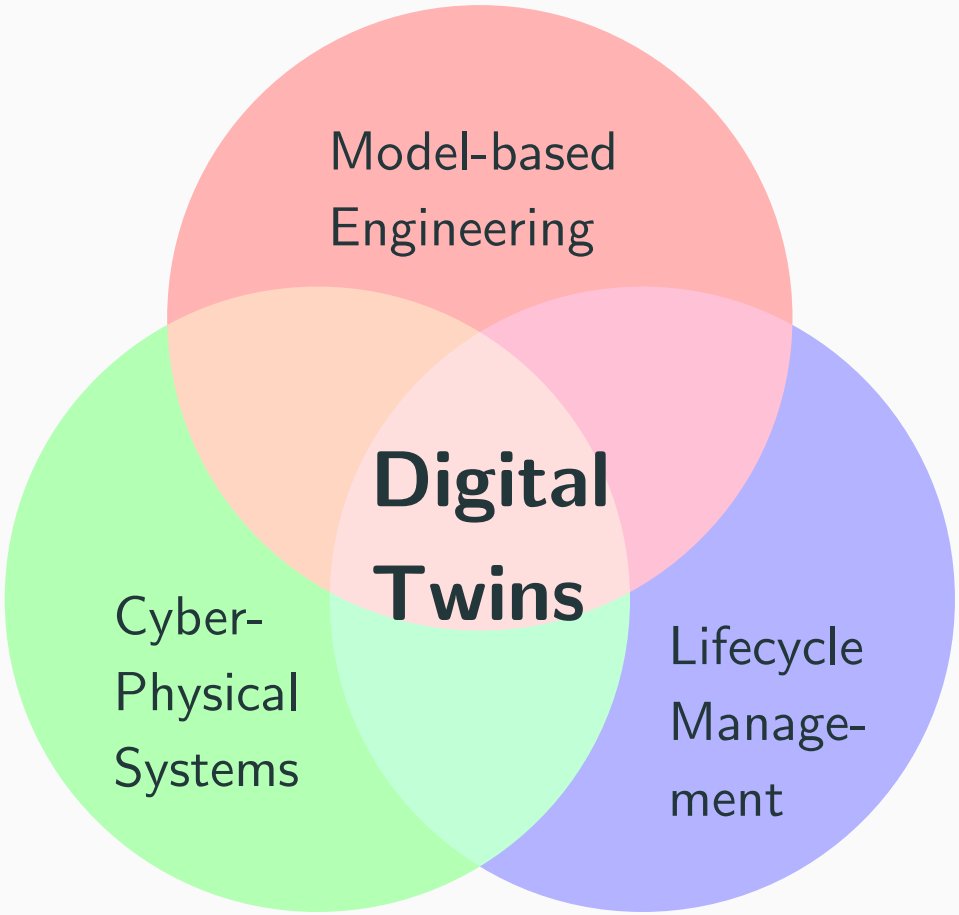
Lifecycle Management

Digital Thread: The Digital Twin Evolves in Tandem with the Asset

1. Connects the designs, requirements and software that go into the system represented by the DT
2. **What are the lifecycle events for operational systems?**



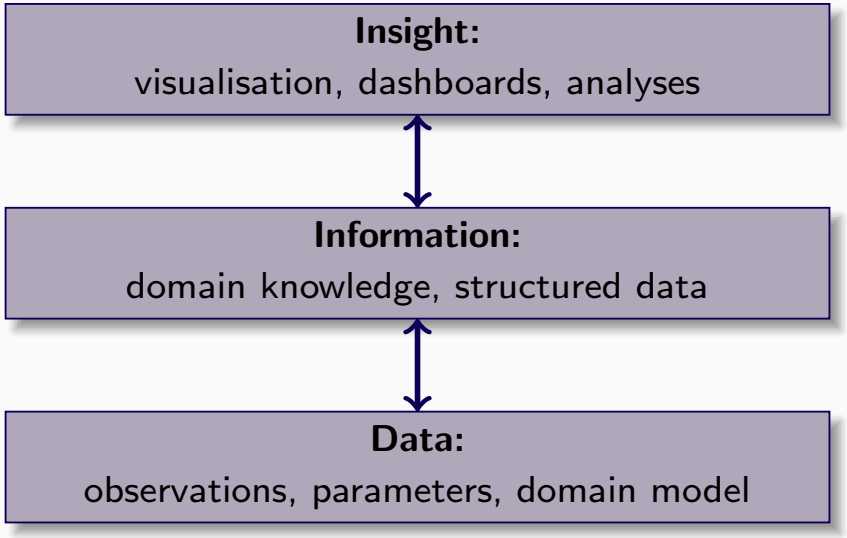
Digital Twins: A New Paradigm in SE?



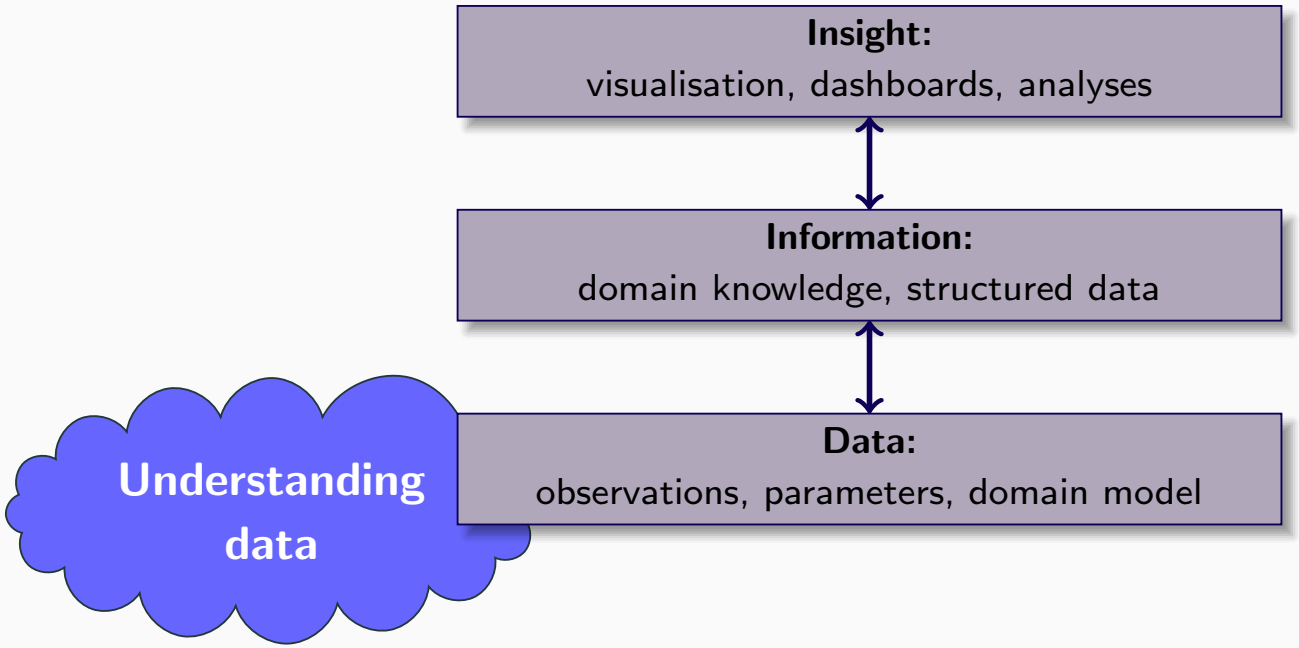
DTs: a new paradigm in SE

- Models go beyond the system design phase
- **Model-centric systems** the purpose is not models to build software, but software to maintain models
- **Model evolution:** reflect changes to the asset (automatically) throughout its lifetime
- **CPS in-the-large:** distributed, heterogeneous

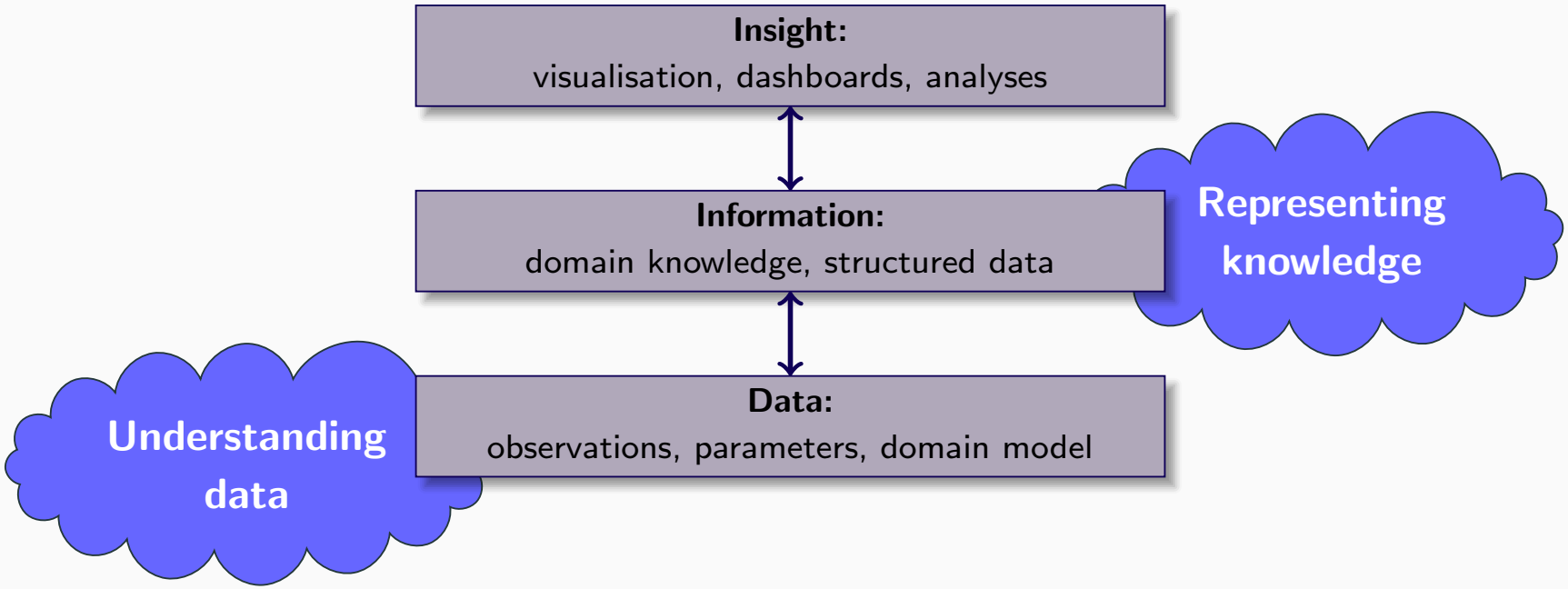
The Conceptual Layers of a Digital Twin



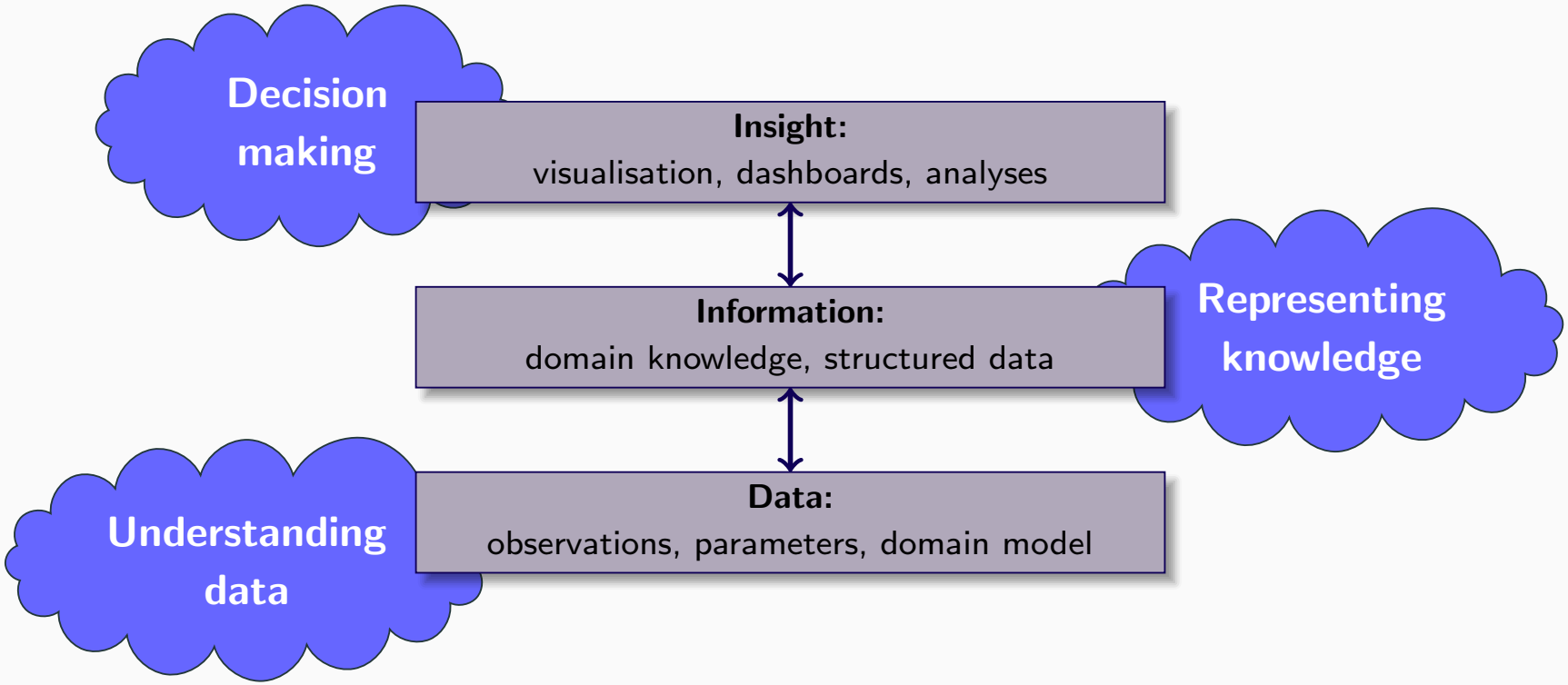
The Conceptual Layers of a Digital Twin



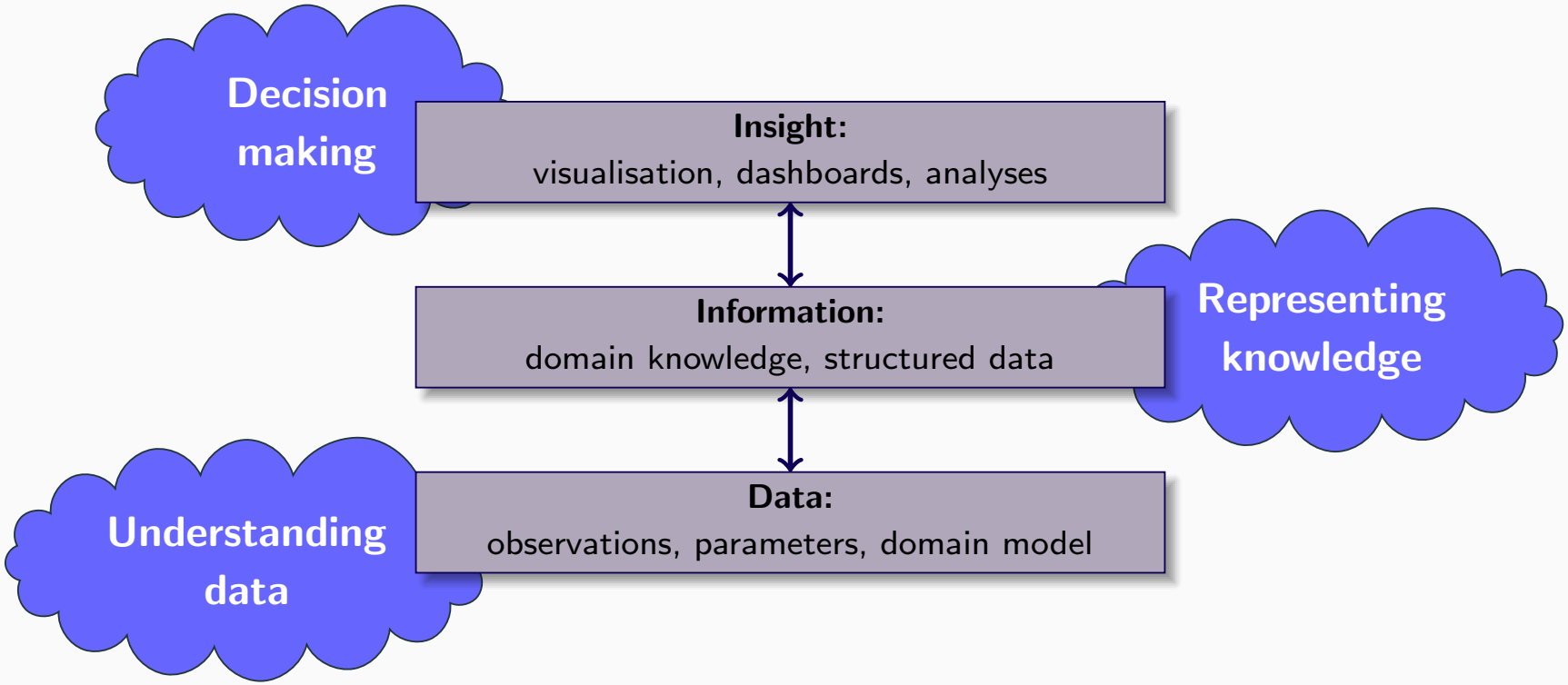
The Conceptual Layers of a Digital Twin



The Conceptual Layers of a Digital Twin

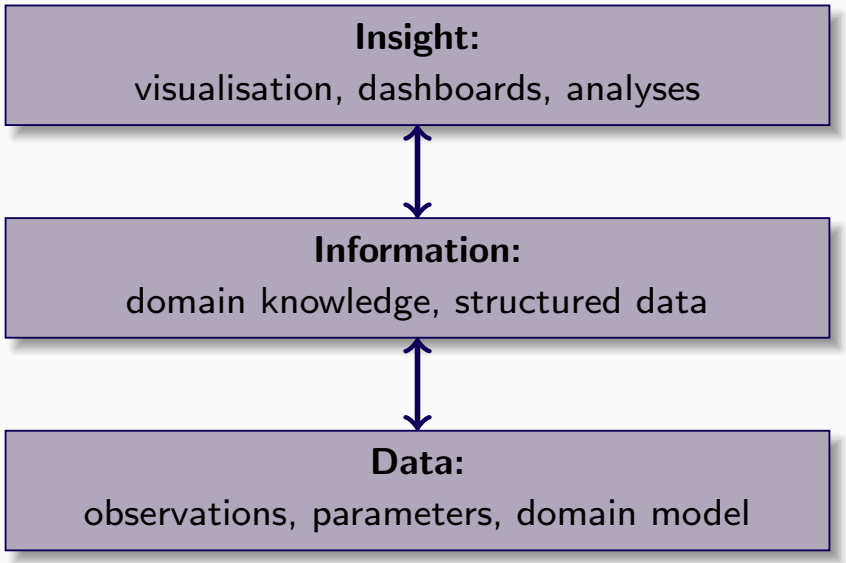


The Conceptual Layers of a Digital Twin



- **Descriptive:** Insight into the past ("what happened" scenarios)
- **Predictive:** Understanding the future ("what may happen" scenarios)
- **Prescriptive:** Advise on possible outcomes ("what if" scenarios)
- **Reactive:** Automated decision making

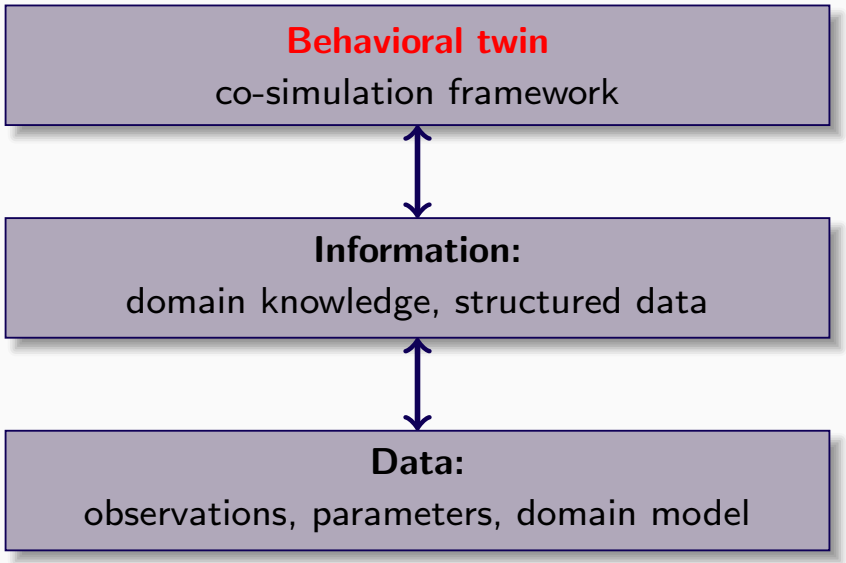
From Information to Insight (and Back Again)



Between information and insight

- Interesting to explore relations between the different layers: information and insight
- We are currently exploring connections between behavioral analyses (e.g., using simulators) and knowledge representation in the information layer

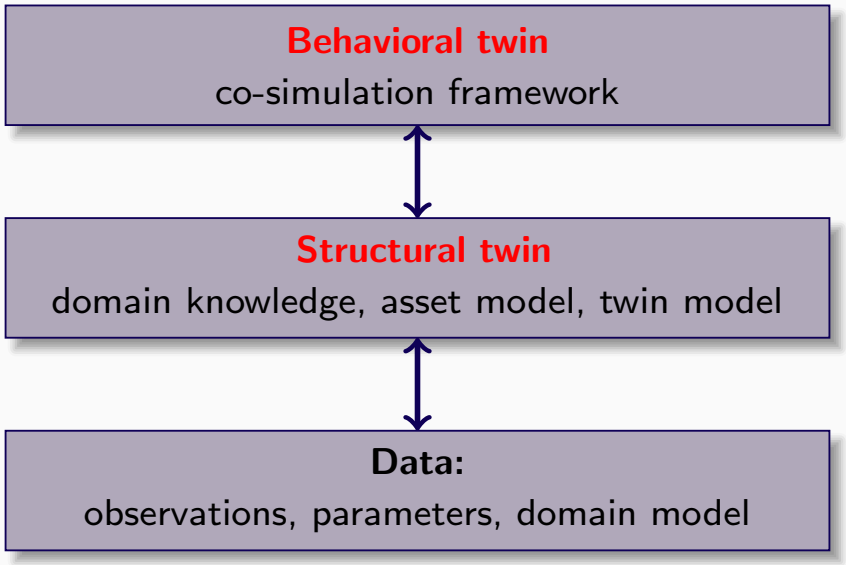
From Information to Insight (and Back Again)



Between information and insight

- Interesting to explore relations between the different layers: information and insight
- We are currently exploring connections between behavioral analyses (e.g., using simulators) and knowledge representation in the information layer

From Information to Insight (and Back Again)



Between information and insight

- Interesting to explore relations between the different layers: information and insight
- We are currently exploring connections between behavioral analyses (e.g., using simulators) and knowledge representation in the information layer

DTs & Formal Methods

What is the role of formal methods in digital twins?

DTs & Formal Methods

What is the role of formal methods in digital twins?

- Conceptual clearness, semantics, compositionality
- Correctness
- Better tool support
- Beyond simulation: worst-case, what-if scenarios, etc

DTs & Formal Methods

What is the role of formal methods in digital twins?

- Conceptual clearness, semantics, compositionality
- Correctness
- Better tool support
- Beyond simulation: worst-case, what-if scenarios, etc

What is the role of knowledge representation in digital twins?

DTs & Formal Methods

What is the role of formal methods in digital twins?

- Conceptual clearness, semantics, compositionality
- Correctness
- Better tool support
- Beyond simulation: worst-case, what-if scenarios, etc

What is the role of knowledge representation in digital twins?

- Structural twin: uniformly represent knowledge about PT and DT
- Reasoning support that can exploit this knowledge
- Allows correctness properties to be expressed as relations between DT and PT

Outline: The Semantically Reflected Digital Twin

In the following, we describe a digital twin architecture using formal methods based on three technologies/techniques.

SWT

Semantic Web Technologies for uniform knowledge representation and integration of domain knowledge (part II).

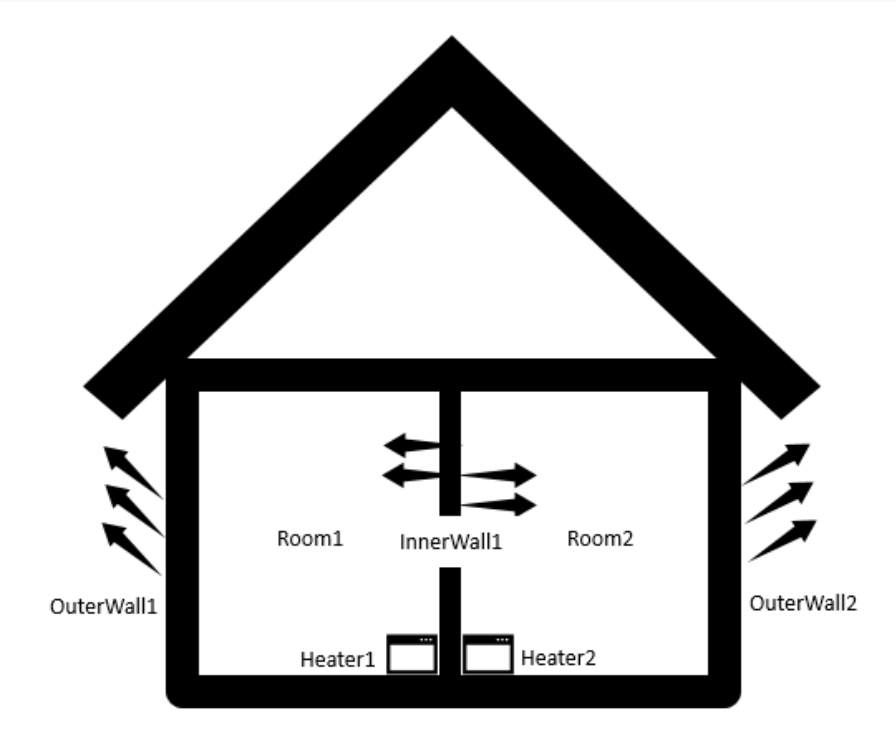
FMI

The Functional Mock-Up Interface standard for interfaces between PT and DT, as well as simulations (part III).

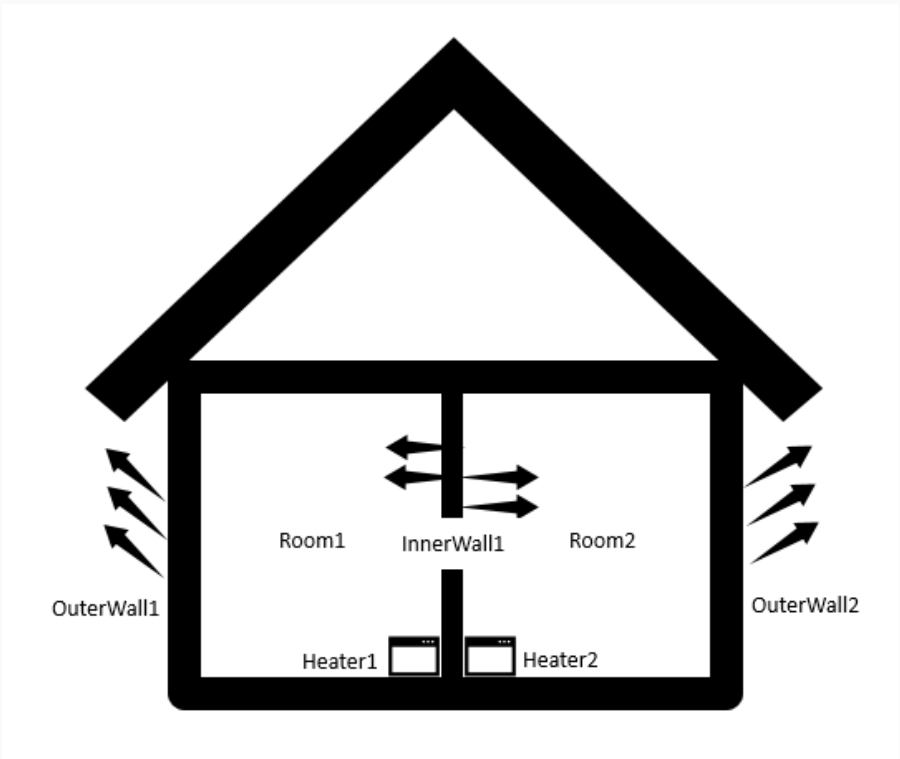
SMOL

Semantic Reflection to reason about PT and DT through the integration of SWT and FMI into a programming language (part IV).
The system is implemented in **SMOL**, a unique language designed specifically for integration of SWT and programming.

Example: House heating



Example: House heating



Structural twin

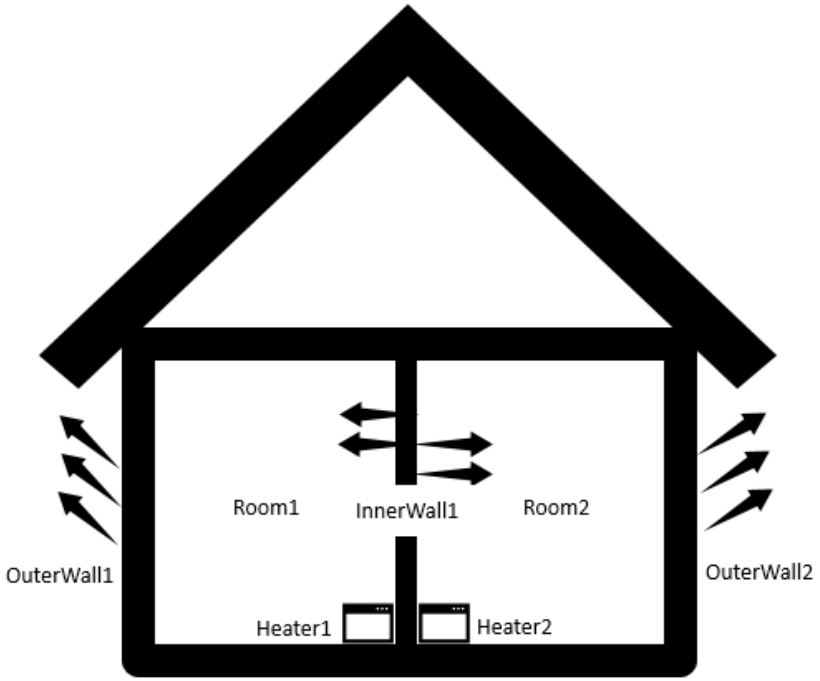
- **Asset model: Domain knowledge**

connects the rooms, heaters, walls into a "house", with corresponding simulators, etc

- **Asset model: Instance**

instance of the domain knowledge for a particular house

Example: House heating



Structural twin

- **Asset model: Domain knowledge**

connects the rooms, heaters, walls into a "house", with corresponding simulators, etc

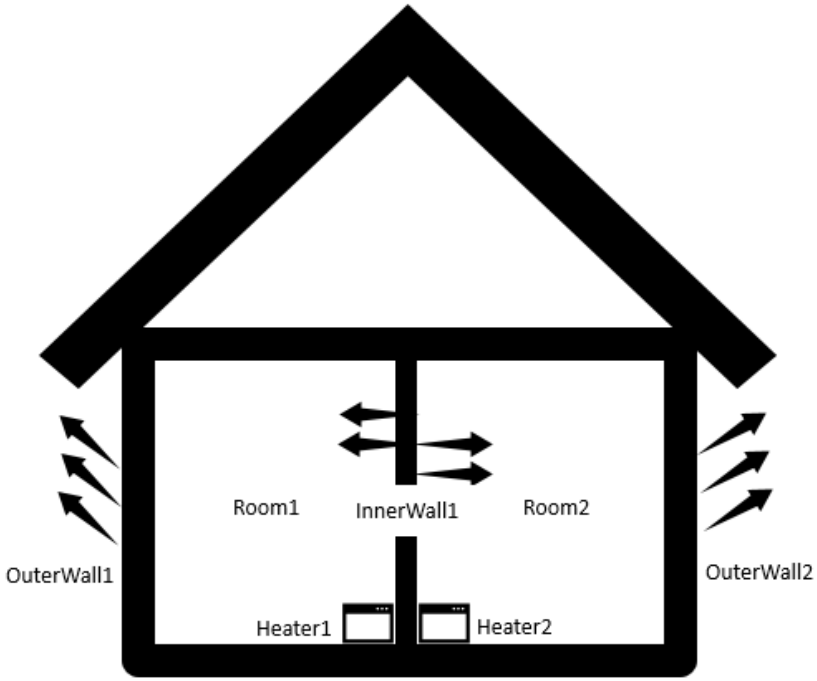
- **Asset model: Instance**

instance of the domain knowledge for a particular house

Behavioral twin

1. **Digital twin infrastructure:** coordinates simulation units
2. **Twin configuration:** coupled simulation units

Example: House heating



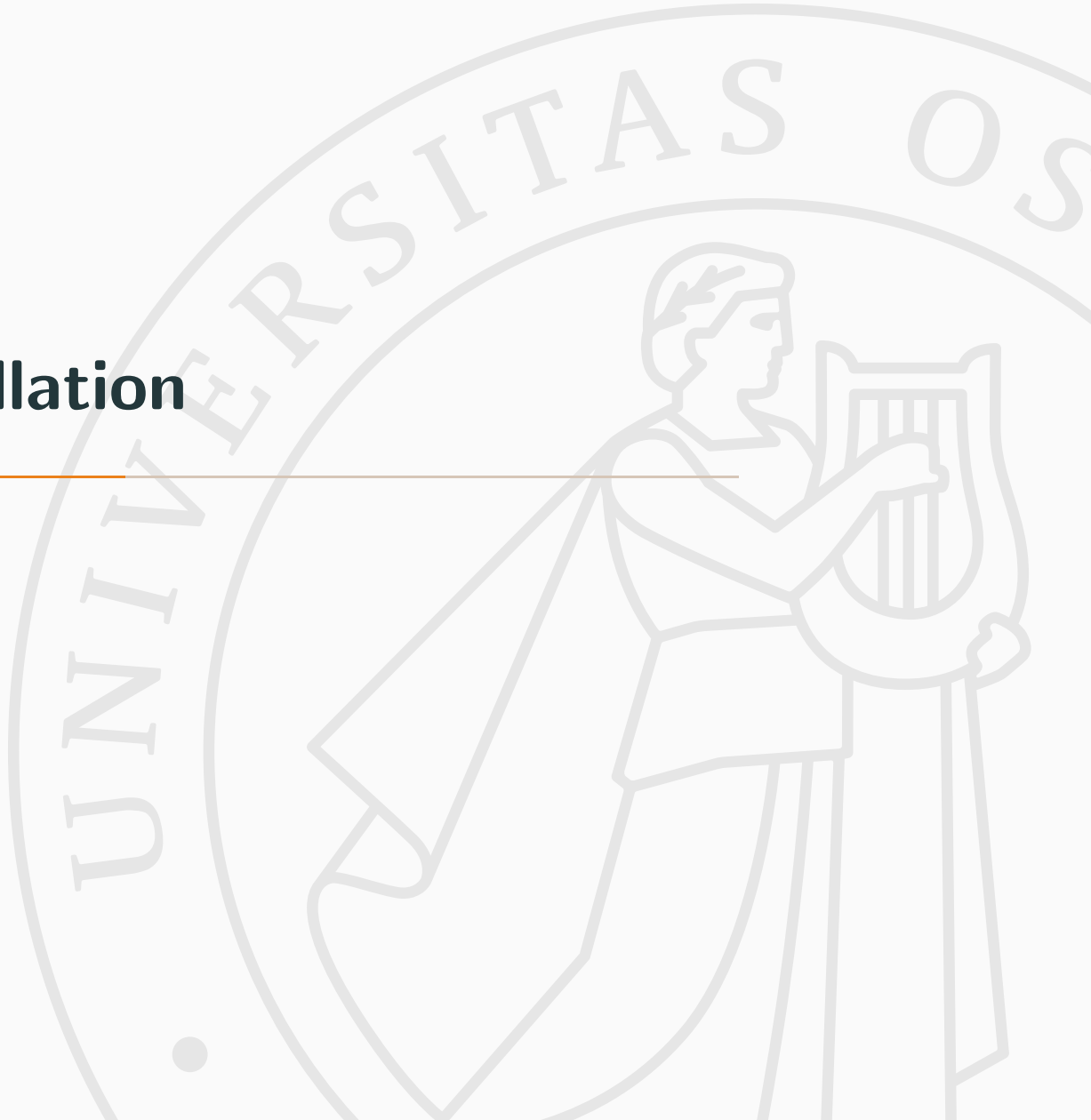
Structural twin

- **Asset model: Domain knowledge**
connects the rooms, heaters, walls into a "house", with corresponding simulators, etc
- **Asset model: Instance**
instance of the domain knowledge for a particular house
- **Twin model: Domain & Instance**
instance of the domain knowledge for the behavioral twin

Behavioral twin

1. **Digital twin infrastructure:** coordinates simulation units
2. **Twin configuration:** coupled simulation units

Tool Installation



Software for today

- Download `https://github.com/smolang/SemanticObjects/blob/master/examples/tutorialfiles.zip`
- Download and install Protegé from `https://protege.stanford.edu/products.php`

Software for tomorrow

- Download and install docker from `https://www.docker.com/get-started/` (or from your favorite Linux distribution)
- Run `docker pull ghcr.io/smolang/smol:latest`
- Run `docker pull openmodelica/openmodelica:v1.19.2-minimal`

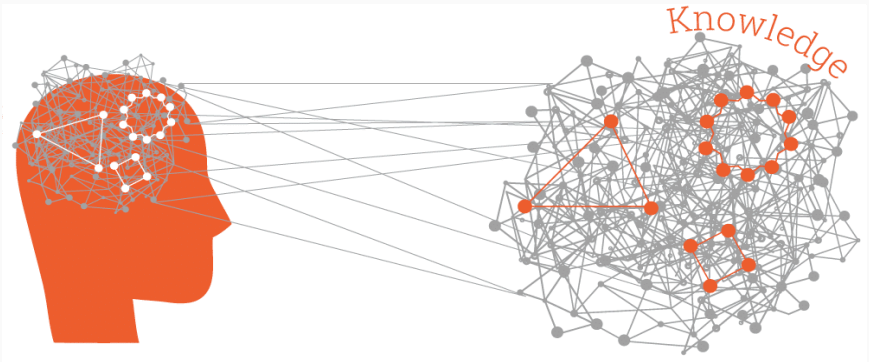
Today:

- **Part I** Digital Twins Introduction: Concepts and Engineering Perspectives
- **Part II Modelling Knowledge using Semantic Technologies**

Tomorrow:

- **Part III** Modelling Physical Systems
- **Part IV** Semantically Reflected Digital Twins

Semantic Technologies



- Knowledge can be described ad hoc or in a structural manner
- Semantic Technologies facilitate the description of structured knowledge, consistency checking and reasoning
- W3C standards and well known technologies:
  - For data: RDF (Resource description framework)
  - For knowledge: OWL (Web Ontology language)
  - For queries: SPARQL(an RDF query language)

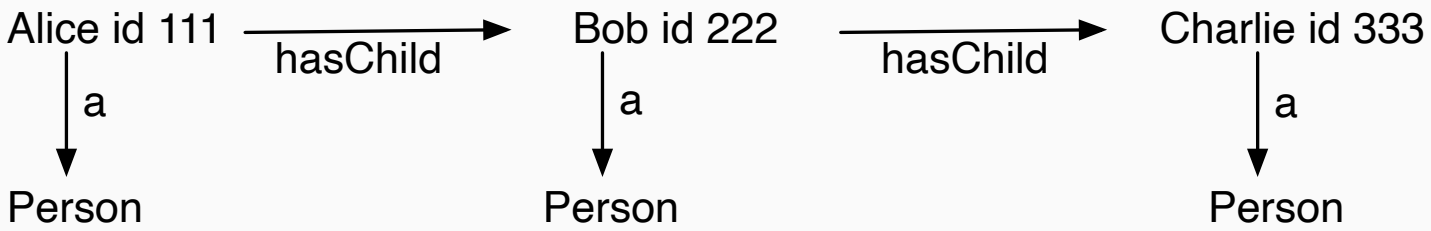
RDF (Resource description framework)

Data in **RDF** is expressed using a triple pattern, which consists of a *subject*, a *predicate*, and an *object*

RDF (Resource description framework)

Data in **RDF** is expressed using a triple pattern, which consists of a *subject*, a *predicate*, and an *object*

Example:



Here 'Alice' is subject, 'a' is predicate, 'Person' is object,
'Alice' is subject, 'id' is predicate, '111' is object, ...

OWL (Web Ontology language)

OWL: knowledge representation languages to build ontologies.

OWL (Web Ontology language)

OWL: knowledge representation languages to build ontologies.

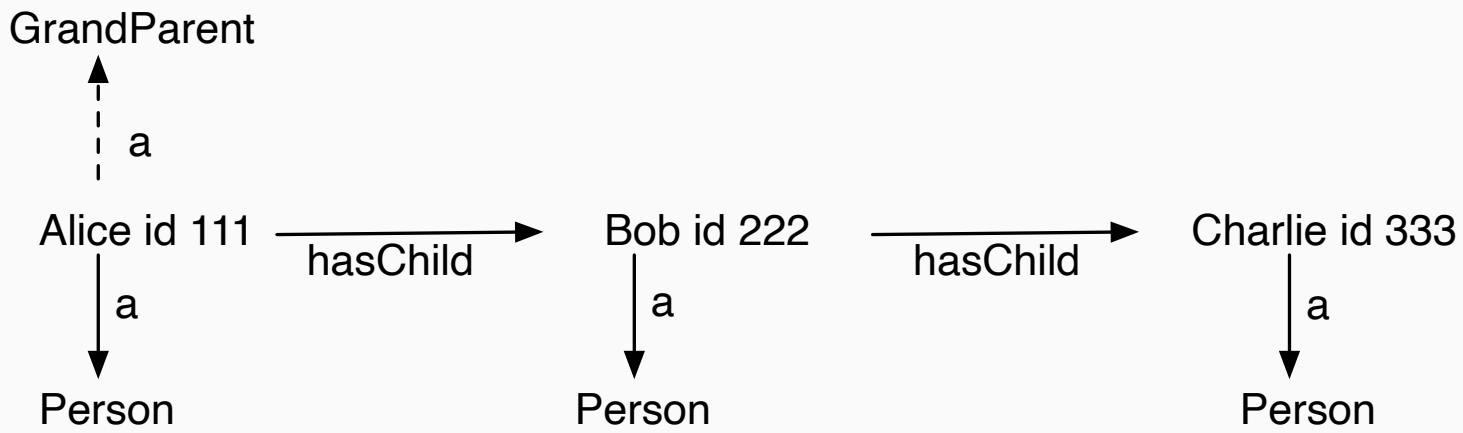
- Ontologies are logics for knowledge representation

OWL (Web Ontology language)

OWL: knowledge representation languages to build ontologies.

- Ontologies are logics for knowledge representation

Example:



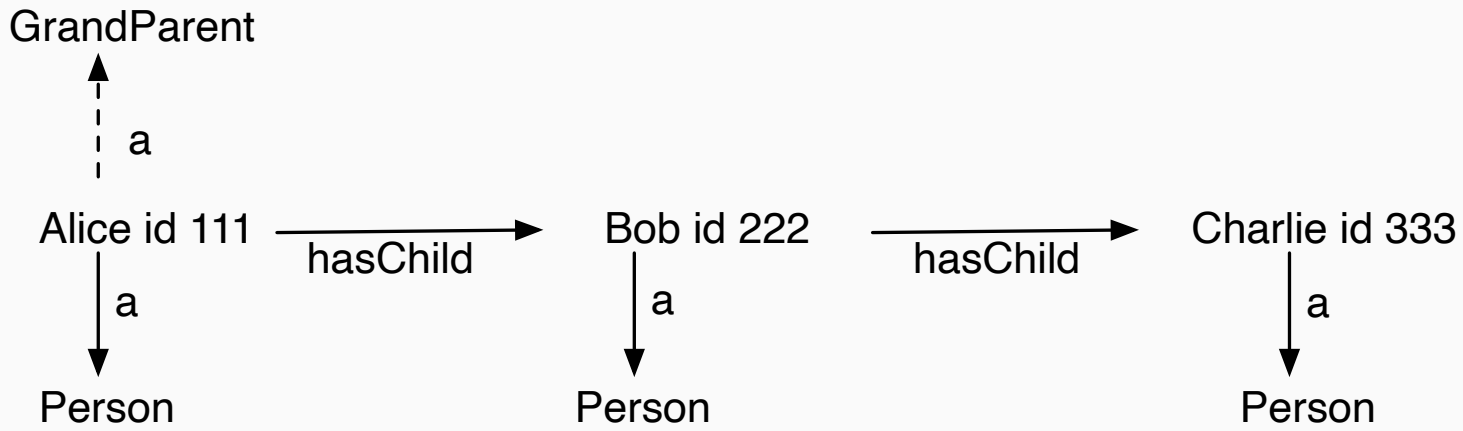
$$\forall x \exists y \exists z \cdot hasChild(x, y) \wedge hasChild(y, z) \wedge Person(z) \implies GrandParent(x)$$

OWL (Web Ontology language)

OWL: knowledge representation languages to build ontologies.

- Ontologies are logics for knowledge representation

Example:



$$\forall x \exists y \exists z \cdot hasChild(x, y) \wedge hasChild(y, z) \wedge Person(z) \implies GrandParent(x)$$

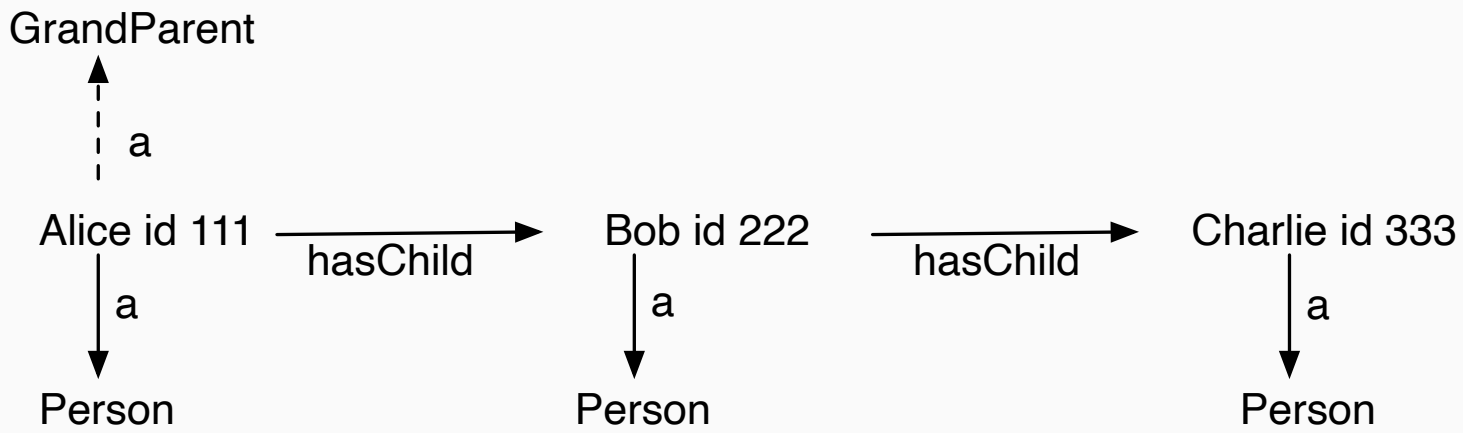
`hasChild` **some** (`hasChild` **some** `Person`) **subClassOf** `GrandParent`

OWL (Web Ontology language)

OWL: knowledge representation languages to build ontologies.

- Ontologies are logics for knowledge representation

  Example:



$$\forall x \exists y \exists z \cdot hasChild(x, y) \wedge hasChild(y, z) \wedge Person(z) \implies GrandParent(x)$$

`hasChild` **some** (`hasChild` **some** `Person`) **subClassOf** `GrandParent`

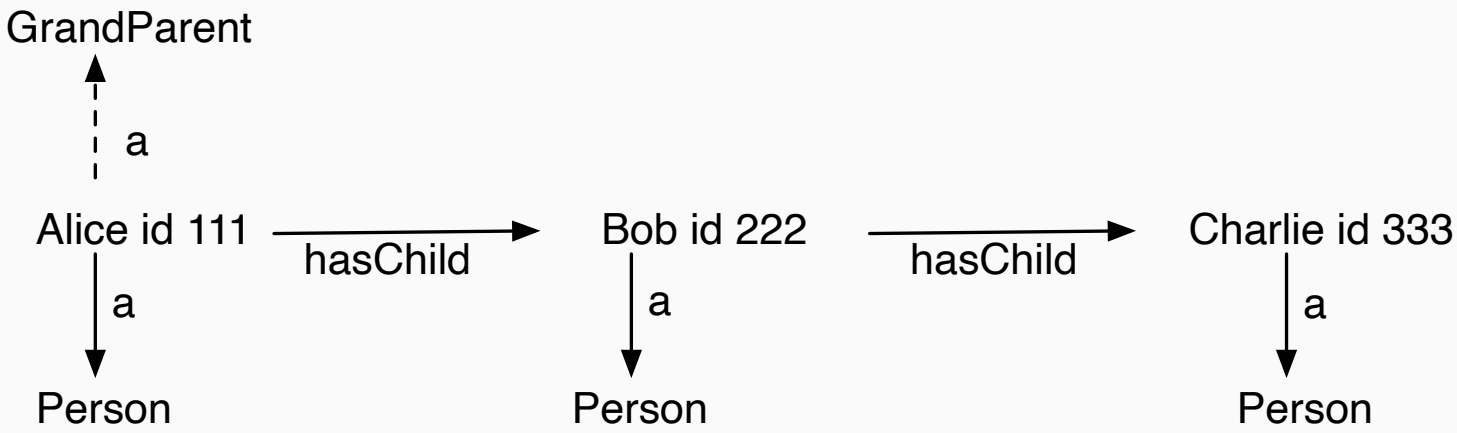
- Ontologies represent knowledge that is incremented over time

SPARQL

SPARQL is an RDF query language:
a query language for databases stored in RDF format

SPARQL

SPARQL is an RDF query language:
a query language for databases stored in RDF format

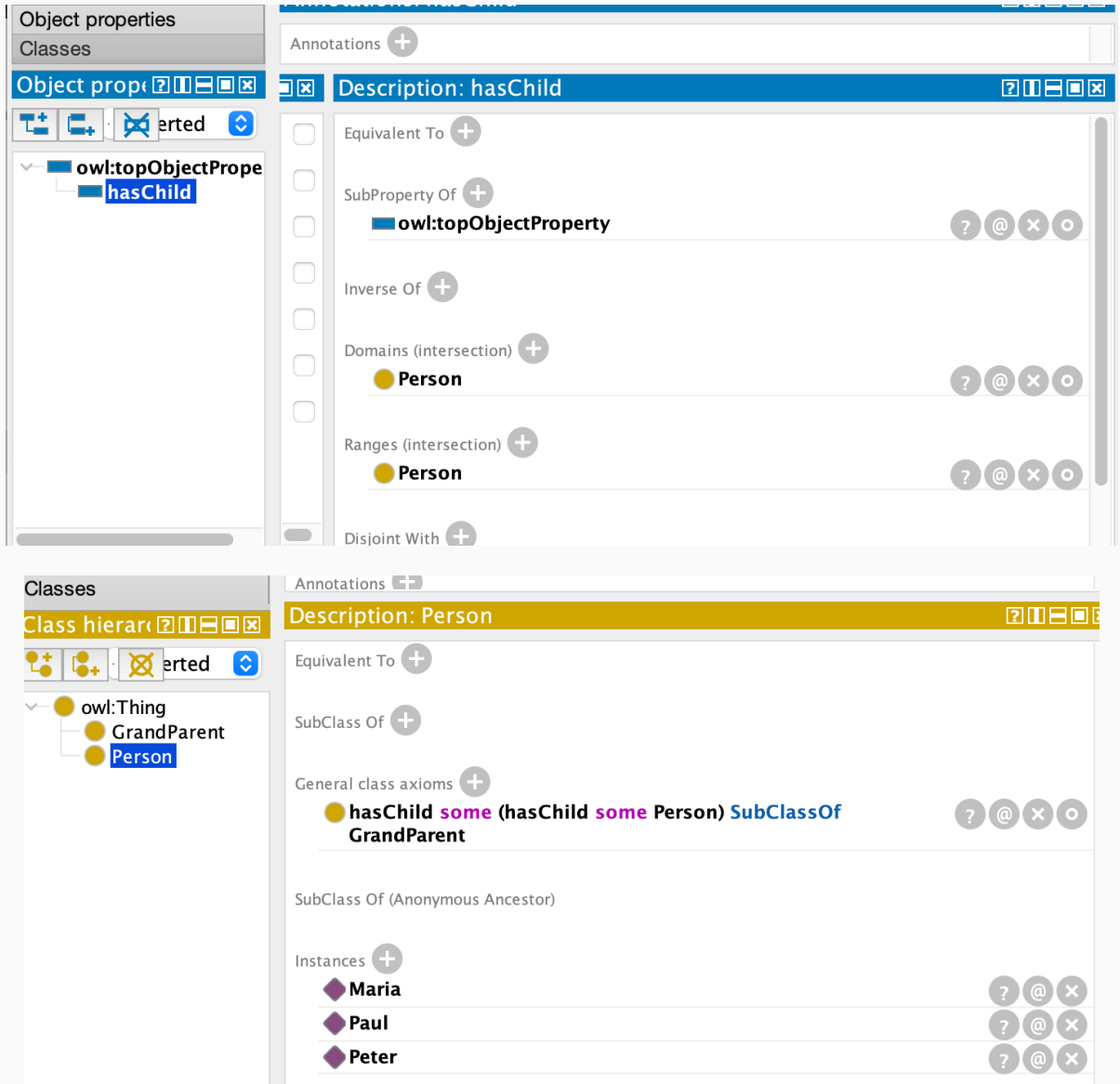



```
SELECT ?x WHERE { ?x a :Person }
```

```
SELECT ?x ?y WHERE { ?x a :Person. ?x :hasChild ?y }
```

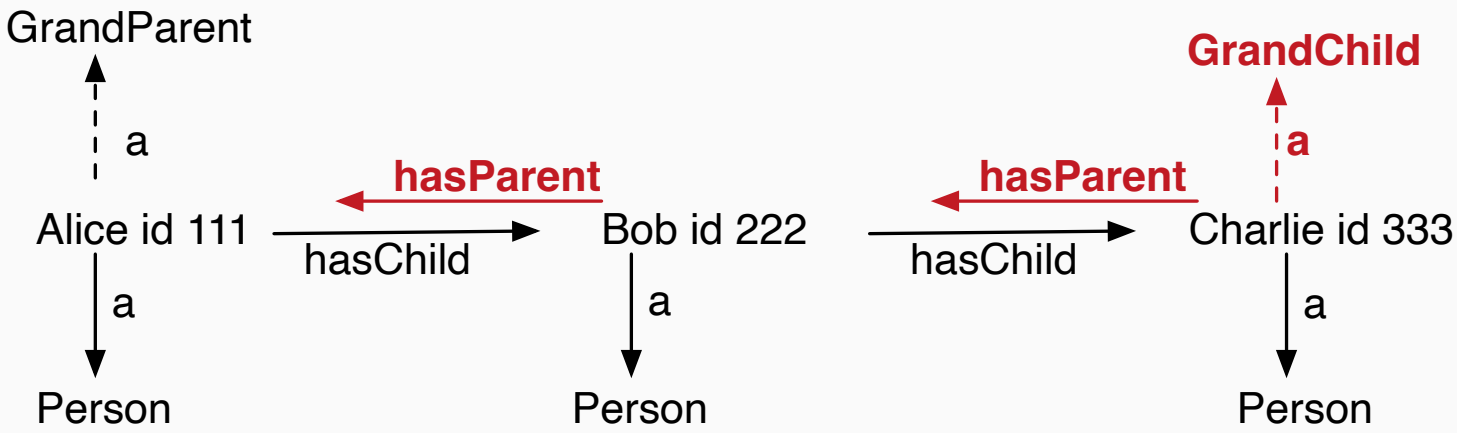
```
SELECT ?x WHERE { ?x a :GrandParent }
```

Example in Protégé



Exercise

Add the GrandChild Class and the hasParent property.



Hint:

$$\forall x \exists y \exists z \cdot \mathit{hasParent}(x, y) \wedge \mathit{hasParent}(y, z) \wedge \mathit{Person}(z) \implies \mathit{GrandChild}(x)$$

Download the example from: `https://github.com/smolang/SemanticObjects/blob/master/examples/tutorialfiles.zip`

File: example1a.ttl

Asset modelling

Asset model in the engineering domain

An asset model is an organized, digital description of the composition and properties of an asset

Asset modelling

Asset model in the engineering domain

An asset model is an organized, digital description of the composition and properties of an asset

- In the engineering domain it is common practice to build asset models to support, e.g., maintenance, operations, design etc.
- There are currently several industry initiatives that endorse the use of ontologies for asset modelling, e.g., in the Industry 4.0

Asset modelling

Asset model in the engineering domain

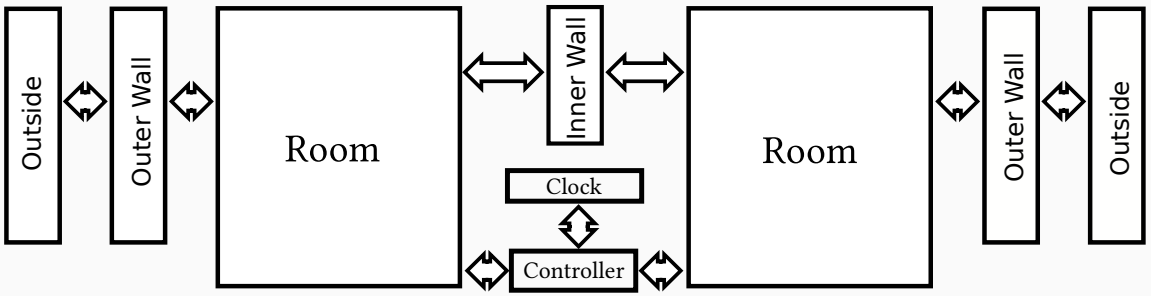
An asset model is an organized, digital description of the composition and properties of an asset

- In the engineering domain it is common practice to build asset models to support, e.g., maintenance, operations, design etc.
- There are currently several industry initiatives that endorse the use of ontologies for asset modelling, e.g., in the Industry 4.0

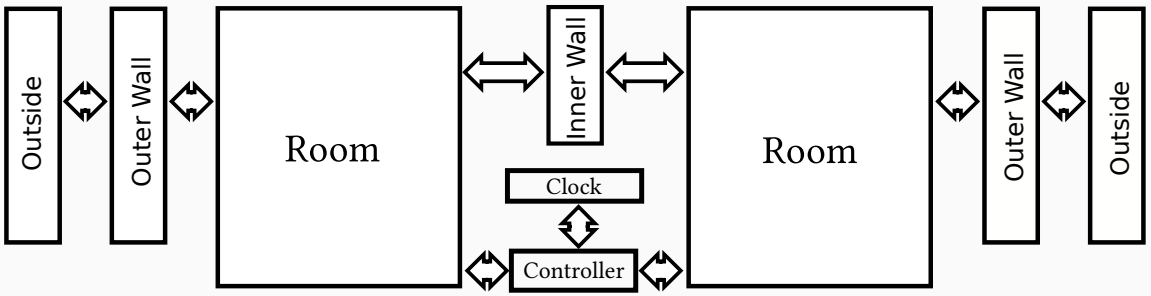
Asset models & digital twins

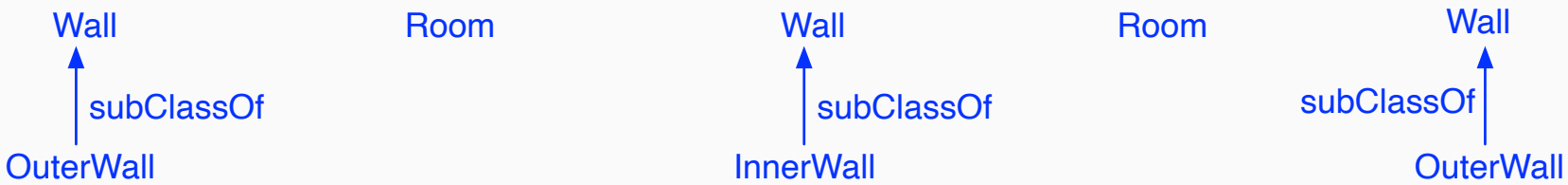
Assets models are any object of interest in a digital twin. They provide the twin with knowledge about the static structure that can be used for the twin's simulation model

The House Asset Use Case

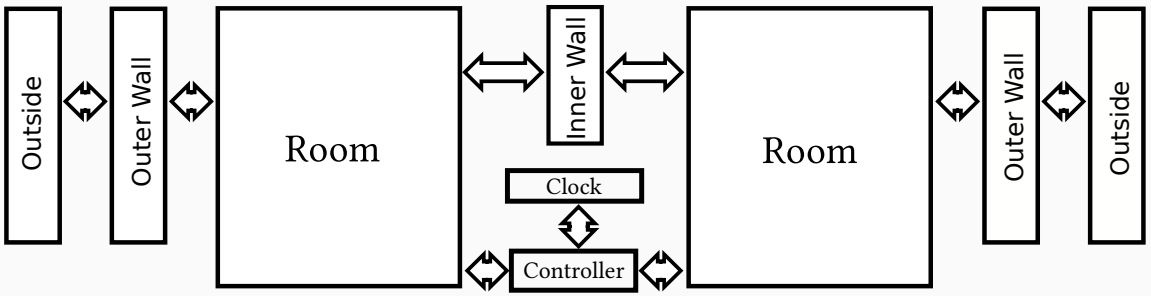


The House Asset Use Case





The House Asset Use Case



w1 id 21

Wall

subClassOf

OuterWall

r1 id 13

Room

w2 id 22

Wall

subClassOf

InnerWall

r2 id 12

Room

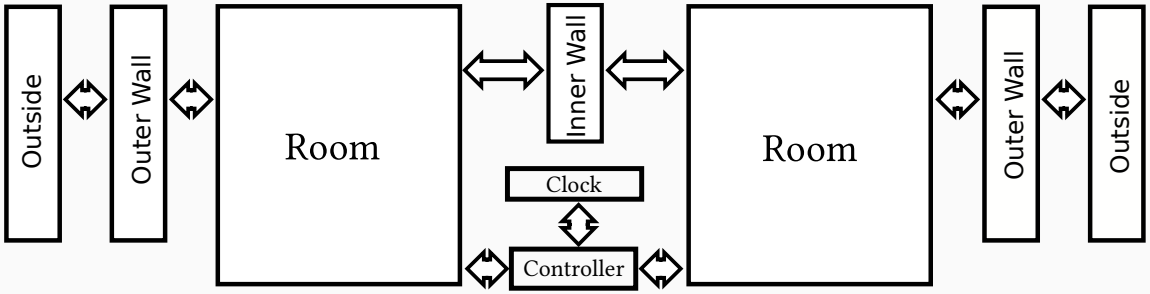
w3 id 23

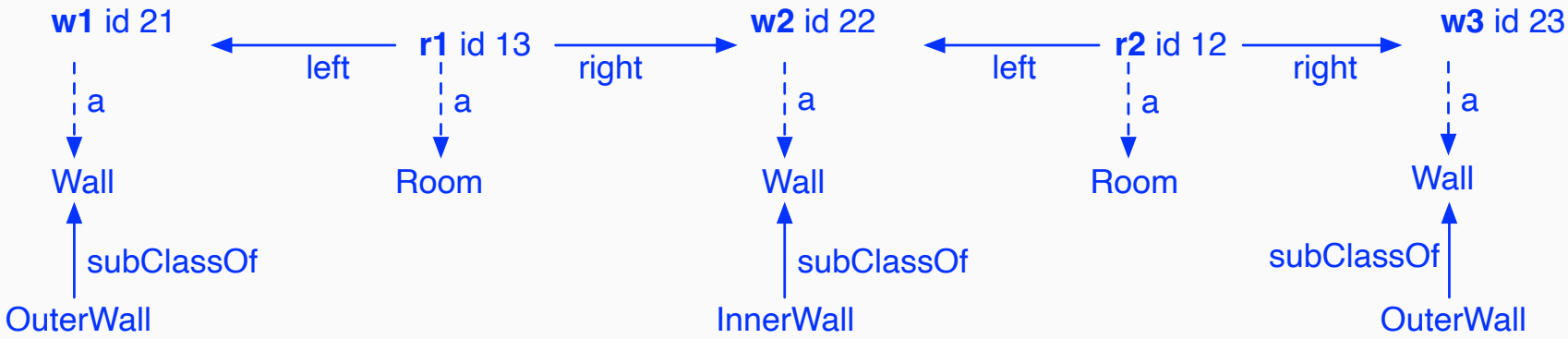
Wall

subClassOf

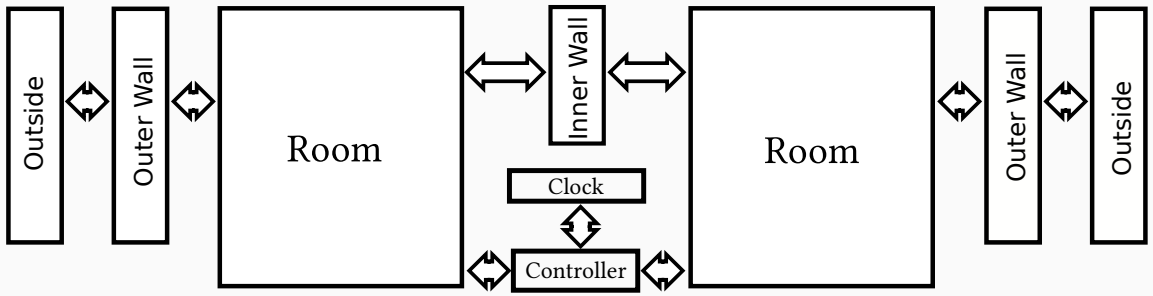
OuterWall

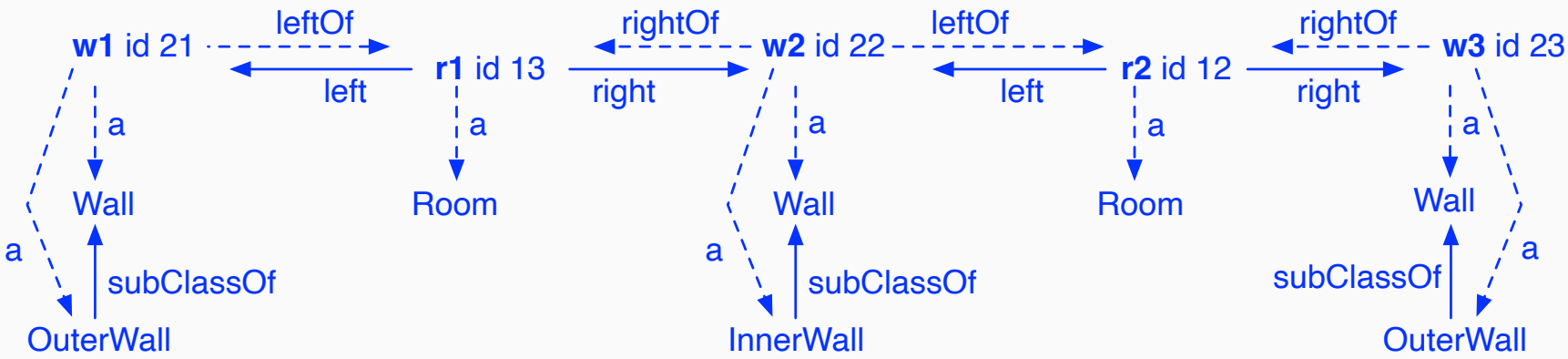
The House Asset Use Case



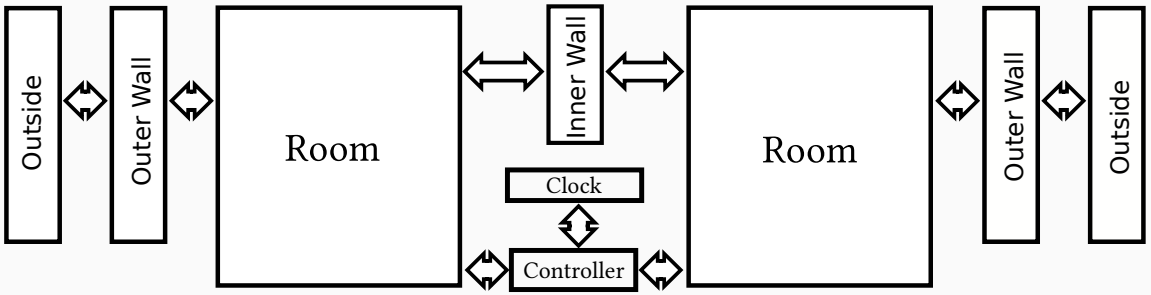


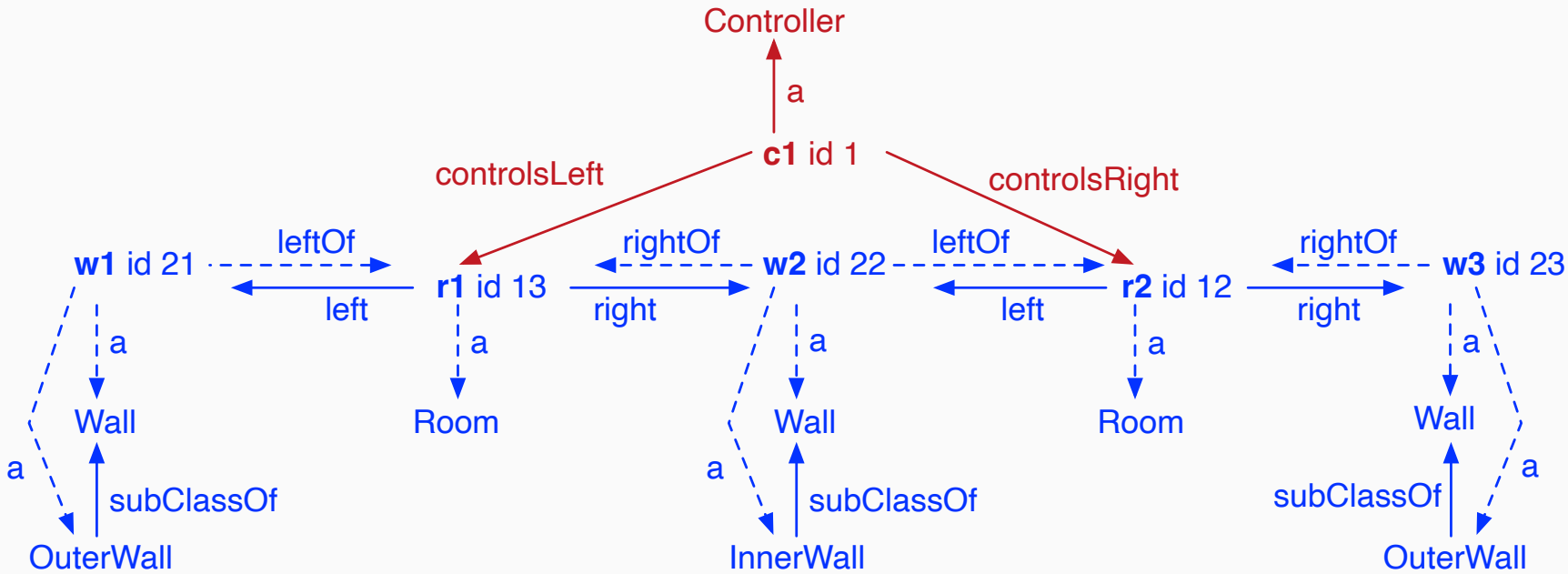
The House Asset Use Case





Exercise: The House Asset Use Case

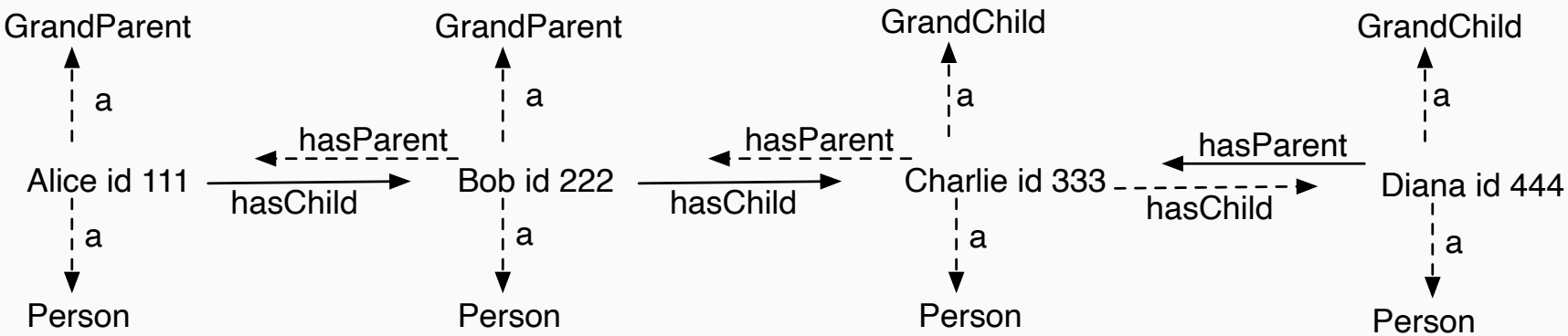




Download the house asset model from: `https://github.com/smolang/SemanticObjects/blob/master/examples/tutorialfiles.zip`

File: house.ttl

Homework: The Family Tree



SPARQL in SMOL

```
main
    List<Int> results =  access(
                    "SELECT ?obj {?a a asset:Room.
                                  ?a asset:id ?obj}");
    while results != null do
        Int current = results.content;
        results = results.next;
        print(current);
    end
end
```

Summary

Today:

- **Part I Digital Twins Introduction: Concepts and Engineering Perspectives**
- **Part II Modelling Knowledge using Semantic Technologies**

Tomorrow:

- **Part III** Modelling Physical Systems
- **Part IV** Semantically Reflected Digital Twins