Information Modeling and Processing (IMP)

Ernest Teniente

teniente@essi.upc.edu

imp
information matters

INFORMATION
MODELING
& PROCESSING

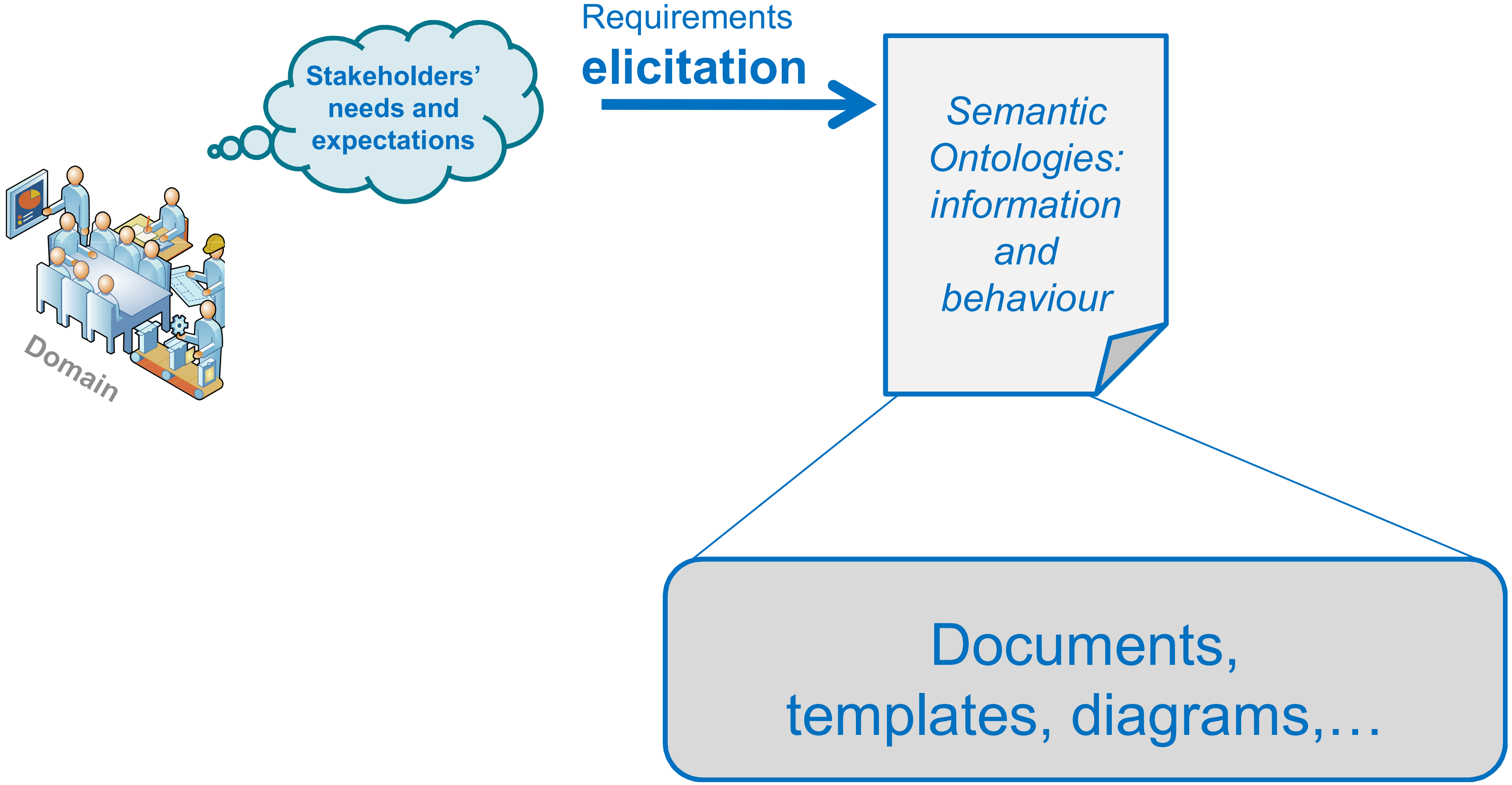
Some facts about IMP (see also *imp.upc.edu*):

- 31 researchers: 28 PhDs + 3 PhD Students
- People from three different universities: UPC, UAB and UOC
- Multidisciplinary group: ESSI, EIO, AC, TSC
- More than 30 years of experience in research
- Publications in the most relevant journals and conferences
- Active collaborations with Spanish and International research groups
- Large experience on I+D+R Projects

IMP areas of expertise:

- Smart Cities
- **Data Science and Big Data**
- **Service and Business Process Engineering**
- Modeling, Simulation and Optimization
- **Ontologies and Information Modeling**
- Cybersecurity
- Learning Analytics
- eHealth Information

Ontology: Semantic Model of the Domain



Stakeholders' needs and expectations

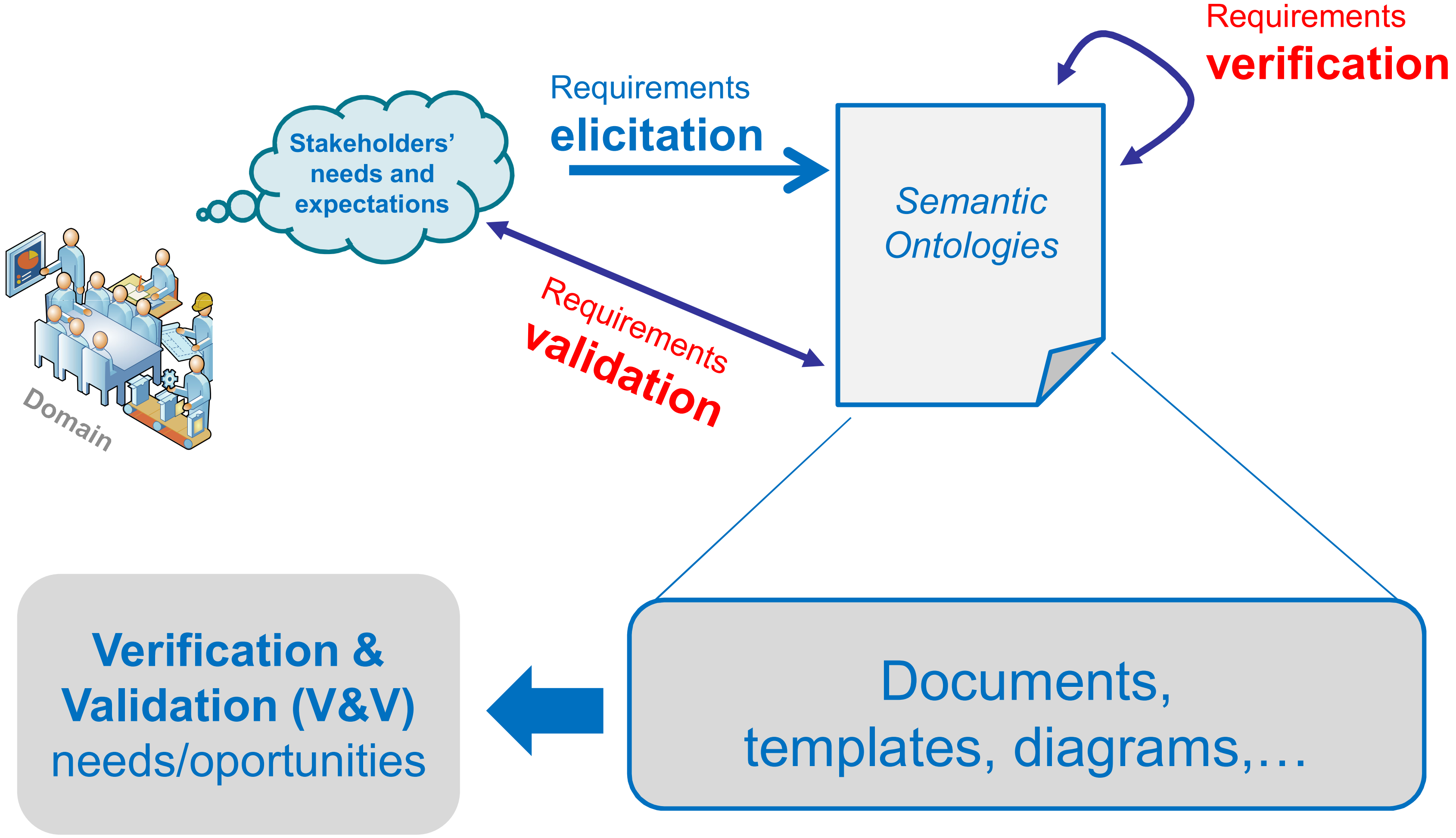
Domain

Requirements **elicitation**

Semantic Ontologies: information and behaviour

Documents, templates, diagrams,…

▪ Using ontologies: Quality Assurance



Stakeholders' needs and expectations

Domain

Requirements **elicitation**

Requirements **validation**

Semantic Ontologies

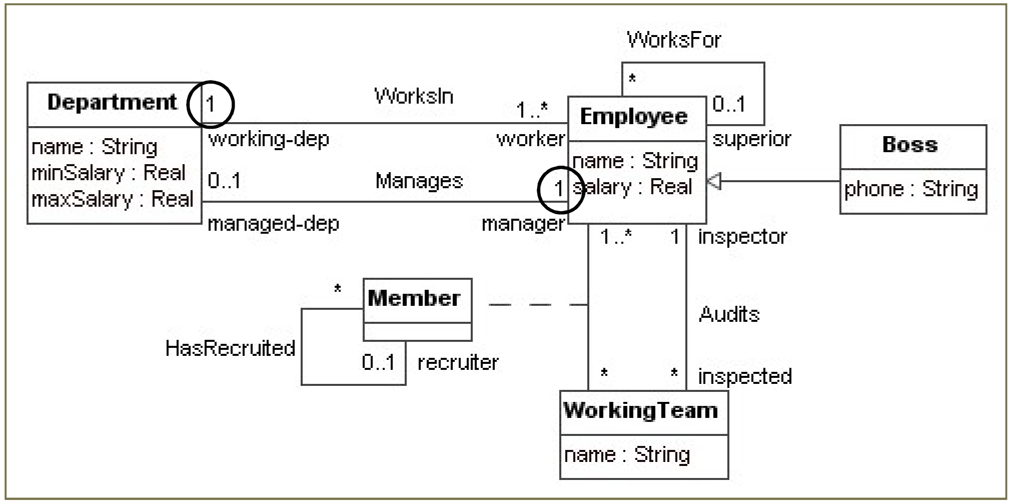
Requirements **verification**

Verification & Validation (V&V) needs/oportunities

Documents, templates, diagrams,…

QA: Conceptual Schema Verification

Is it possible to have at least one instance of *Employee* (liveliness)?



```
Integrity constraints
…
6.context Department inv ManagerIsWorker:
self.worker->includes(self.manager)
…
8.context Boss inv BossIsManager:
self.managed-dep->notEmpty()
9.context Boss inv BossHasNoSuperior:
self.superior->isEmpty()
10.context Boss inv SuperiorOfAllWorkers:
self.employee->includesAll(self.managed-
dep.worker)
```

Employee	employee(#e1,john)
WorksIn	worksIn(#e1,#s1)
Department	department(#s1,sales)
Manages	manages(#e1,#s1)

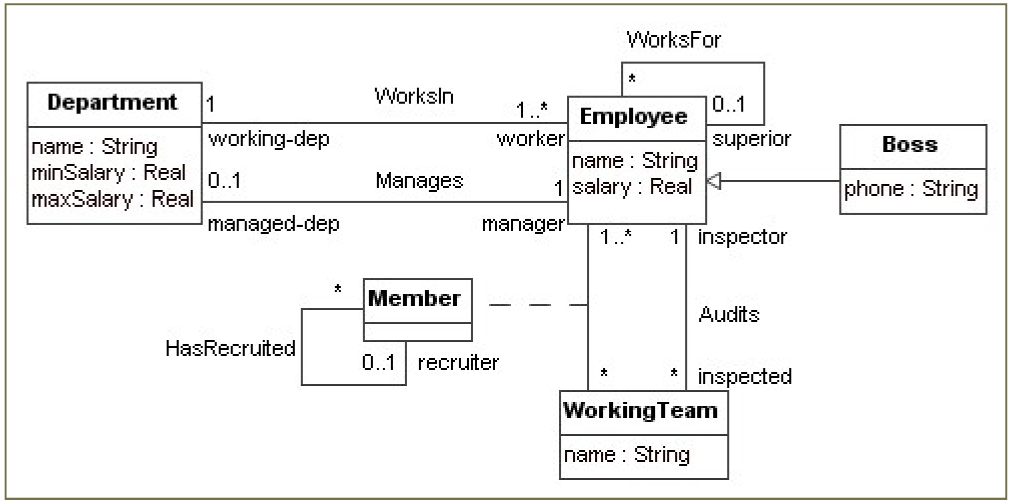
Employee **is lively**

The class is well-defined

Verification tests:
- **automatically generated**
- their **result** determines the (un)correctness of the schema

Using ontologies: Generating Test Data

Give me a sample database where an Employee works for himself



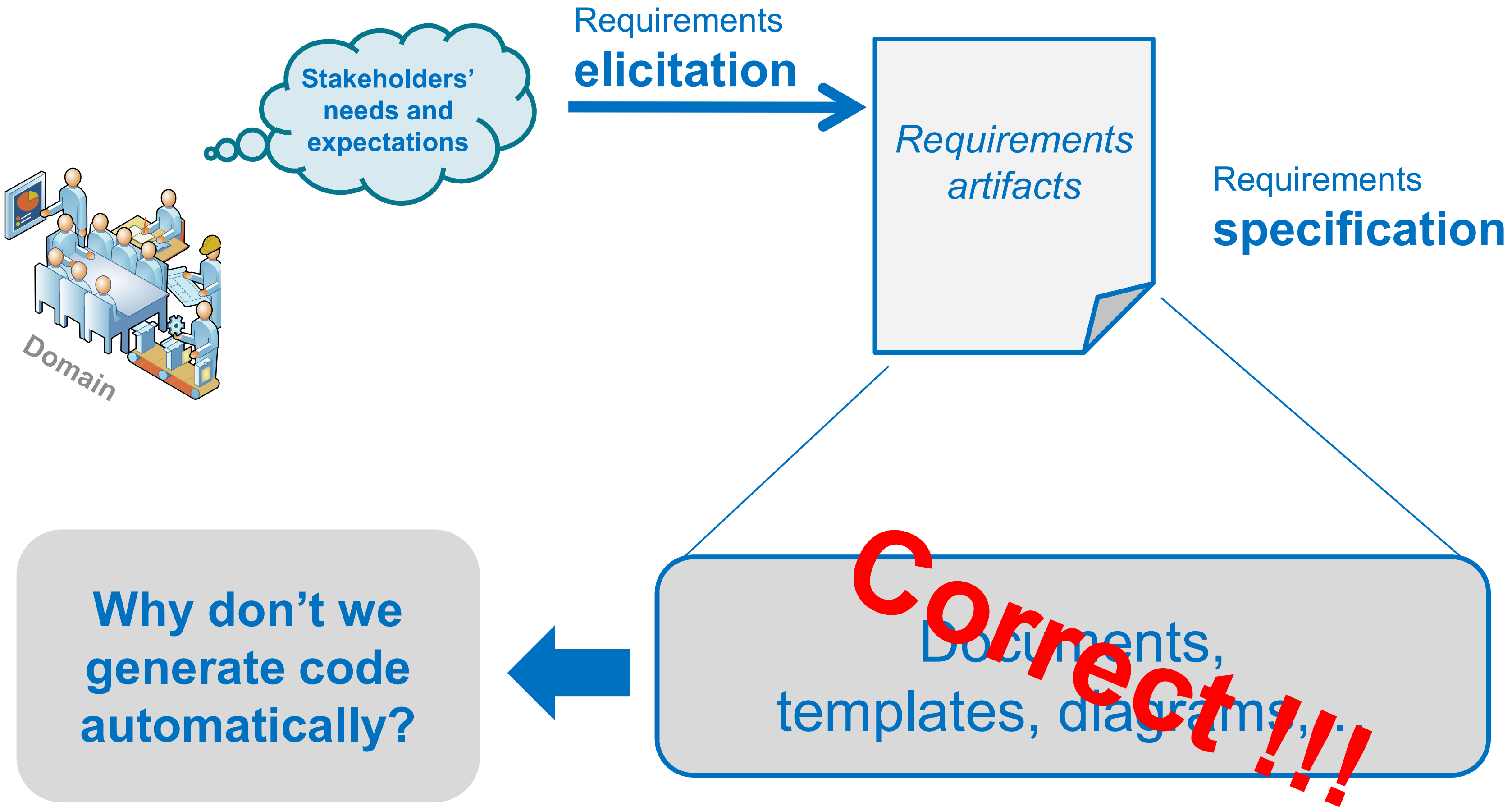
```
Integrity constraints


…
6.context Department inv ManagerIsWorker:
self.worker->includes(self.manager)
7.context Department inv MgrHasNoSuperior:
self.manager.superior->isEmpty()
…
```

WorksFor	worksFor(#e1, #e1)
Employee	employee(#e1,mary)
WorksIn	worksIn(#e1, #s1)
Department	department(#s1,sales)
Manages	manages(#e1, #s1)

We can define a set of conditions over the data we want to obtain to be able to test a software application

(and obtain this data automatically)

Using ontologies: Automatic Code Generation



Stakeholders' needs and expectations

Requirements **elicitation**

Requirements artifacts

Requirements **specification**

Documents, templates, diagrams,...

Correct !!!

Why don't we generate code automatically?

Using ontologies: Automatic Code Generation

- TINTIN: a tool for incremental integrity checking of SQL assertions

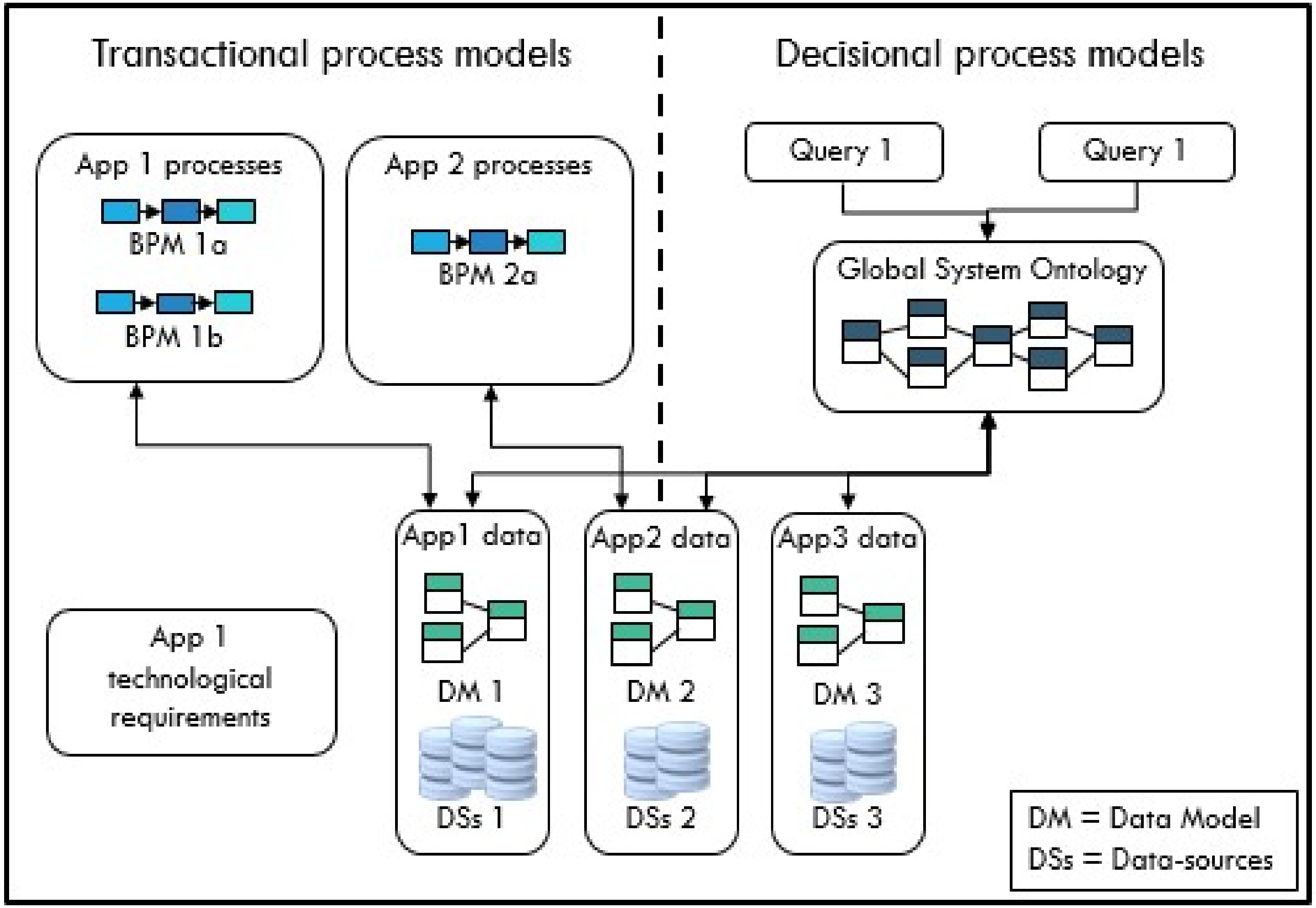
- How can we check the following constraint?

```
Select * from FamousDirector as FD
where not exists (Select * from Directs as D
                                              join Wins as W on (D.movie_id = W.movie_id)
                              where D.person_id = FD.id)
```

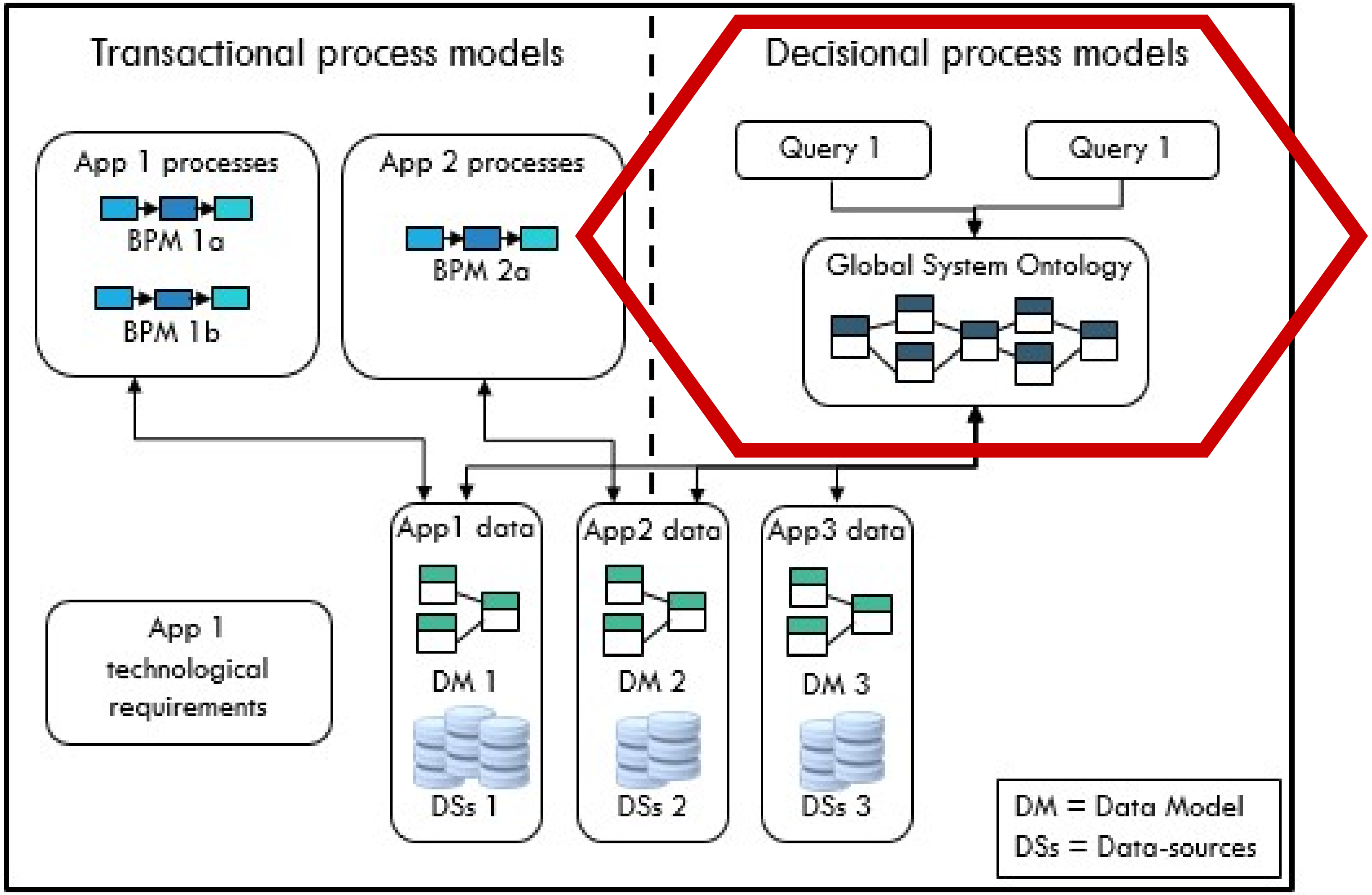
Manually programming an efficient solution is difficult:

are you sure we are taking all cases into account?

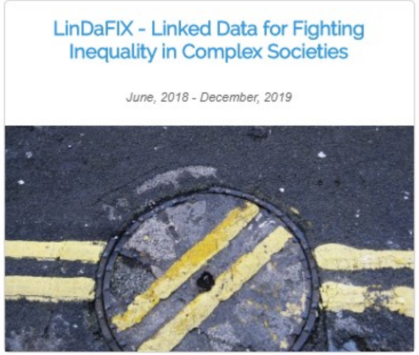
- Using ontologies: Automatic Software Execution (our vision)



- Using ontologies: Data Science and Analytics



- Some relevant projects


LinDaFIX - Linked Data for Fighting Inequality in Complex Societies

June, 2018 - December, 2019

LinDaFIX - Linked Data for Fighting Inequality in Complex Societies

Aimed at facilitating the integration, enrichment, and analysis of the data provided by the Social Rights Department of the Ajuntament de Barcelona. It builds on semantic technologies, automated reasoning and machine learning to cross information and discover hidden relationships.

https://cit.upc.edu/en/featured/projecte_lindafix

METRICS

METamodelling for Retrieving Invoked Source Code

REMEDiAL - Automatic Reasoning, Model Execution and Ontology Data Analysis.

January, 2018 - January, 2021

REMEDiAL - Automatic Reasoning, Model Execution and Ontology Data Analysis

The main goal of this project is to provide capabilities to help to automate the software development process, based on the use of domain ontologies

https://imp.upc.edu/en/projectes/remedial-automatic-reasoning-model-execution-and-ontology-data-analysis

Thank you