University of Antwerp
Faculty of Science

Towards a Modular Multi-Conformance (Meta-)Modelling Framework

By Andrei Bondarenko

Master's thesis defence, September 2021

About Me

2nd year master's student at the UA

specialising in Data Science and Artificial Intelligence

Outline

- **Why?**
- **What?**
- **How?**
  - (Modular) Meta-Modelling
  - Framework
  - Implementation/Bootstrapping
- **Summary of Contributions**
- **Future Work**

Why?

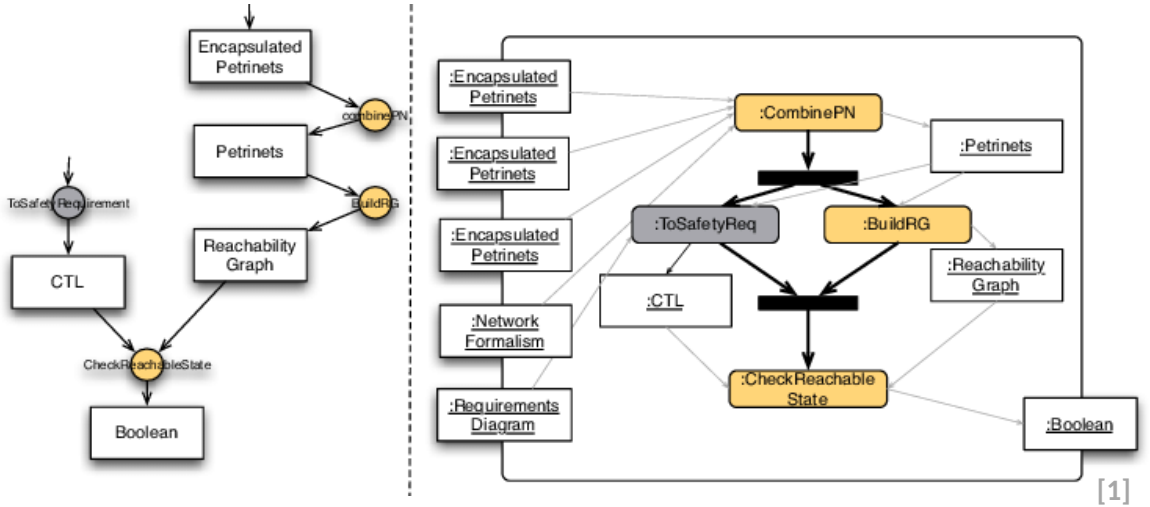
What is the problem?

[1]

Modelling complex systems

- Systems are growing in:
  - Size
  - Complexity
- Context diversity is increasing

[1] Lúcio, Levi & Amrani, Moussa & Dingel, Juergen & Lambers, Leen & Salay, Rick & Selim, Gehan & Syriani, Eugene & Wimmer, Manuel. (2014). Model Transformation Intents and Their Properties. Software & Systems Modeling. 15. 10.1007/s10270-014-0429-x.



[1]

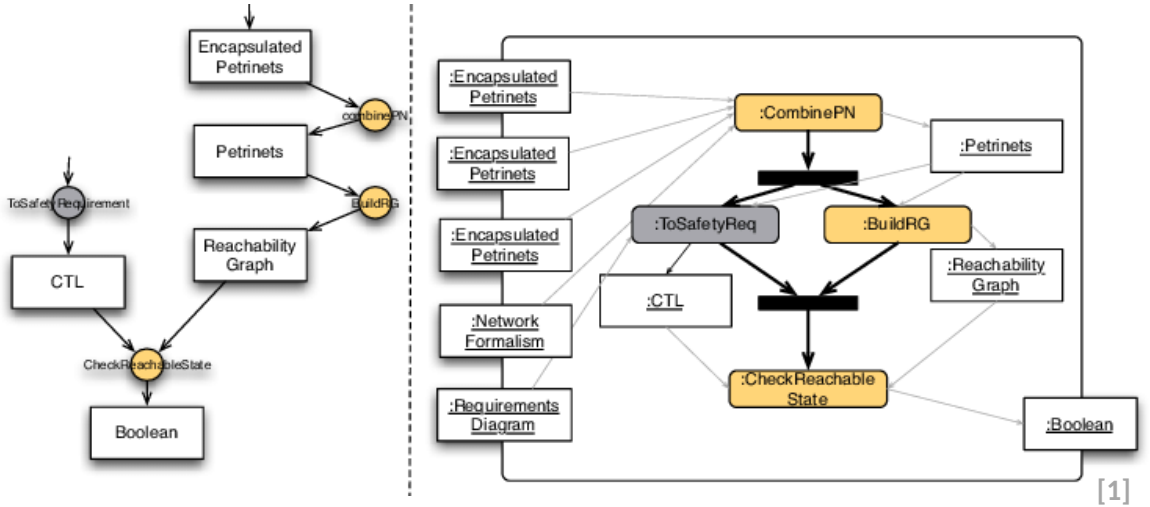
Modelling complex systems

- Systems are growing in:
  - Size
  - Complexity
- Context diversity is increasing

Models

- Number of models increases
- Size of models increases
- **Heterogeneity** of models increases
  - Storage
  - Algorithms
  - Domain-specific implementations

[1] Lúcio, Levi & Amrani, Moussa & Dingel, Juergen & Lambers, Leen & Salay, Rick & Selim, Gehan & Syriani, Eugene & Wimmer, Manuel. (2014). Model Transformation Intents and Their Properties. Software & Systems Modeling. 15. 10.1007/s10270-014-0429-x.



[1]

Modelling complex systems

- Systems are growing in:
  - Size
  - Complexity
- Context diversity is increasing

Models

- Number of models increases
- Size of models increases
- **Heterogeneity** of models increases
  - Storage
  - Algorithms
  - Domain-specific implementations

Framework

- Uniform and efficient access/interaction
  - Does not exist
- Reimplementation of domain-specific storage and algorithms
  - Inefficient storage
  - Hit in performance
  - Impossible if proprietary
- Communication with external services not transparent

[1] Lúcio, Levi & Amrani, Moussa & Dingel, Juergen & Lambers, Leen & Salay, Rick & Selim, Gehan & Syriani, Eugene & Wimmer, Manuel. (2014). Model Transformation Intents and Their Properties. Software & Systems Modeling. 15. 10.1007/s10270-014-0429-x.

What?

What do we need to solve the problem?

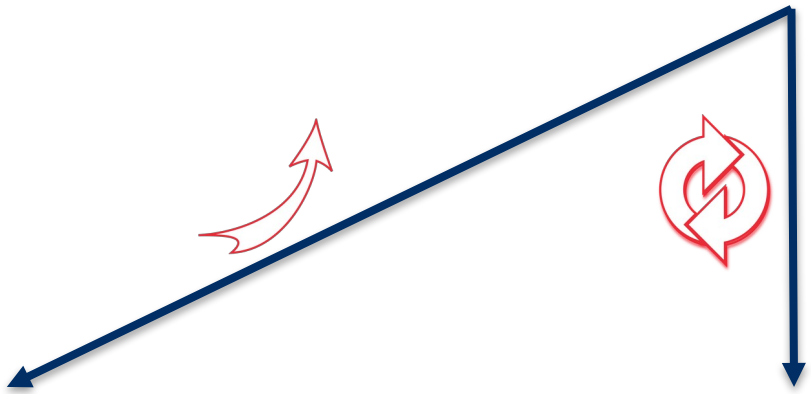
FRAMEWORK

FRAMEWORK

Scalability

- Harmonise heterogeneous models
  - Management
  - Interaction
- Easy extension with new model types
  - Efficient development
- Transparent to users

FRAMEWORK



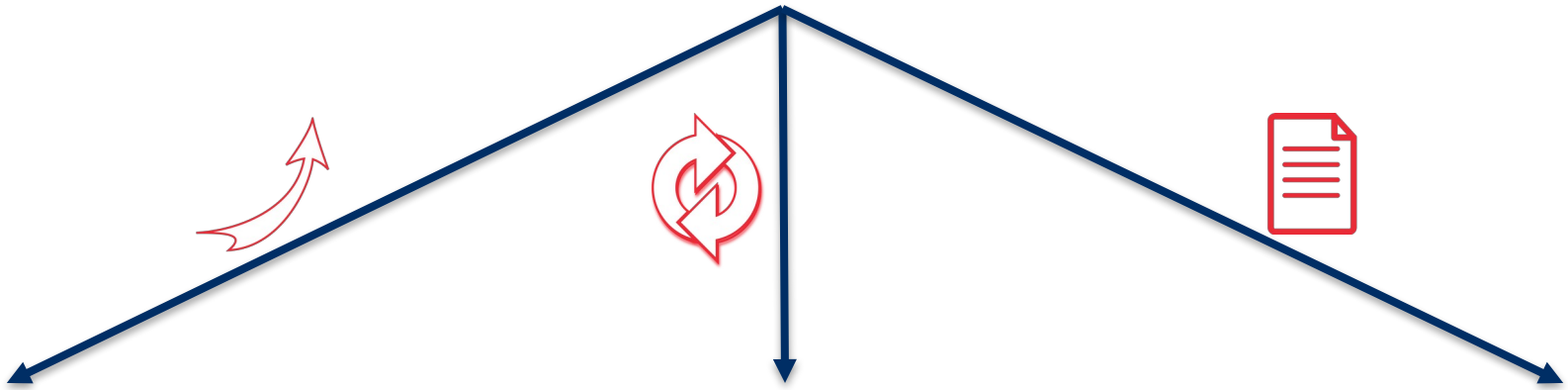
Scalability

- Harmonise heterogeneous models
  - Management
  - Interaction
- Easy extension with new model types
  - Efficient development
- Transparent to users

Reuse

- Allow for integration of existing implementations/services
  - Efficient storage
  - Optimised performance
  - Proprietary implementations
  - Decreased development time

FRAMEWORK



Scalability

- Harmonise heterogeneous models
  - Management
  - Interaction
- Easy extension with new model types
  - Efficient development
- Transparent to users

Reuse

- Allow for integration of existing implementations/services
  - Efficient storage
  - Optimised performance
  - Proprietary implementations
  - Decreased development time

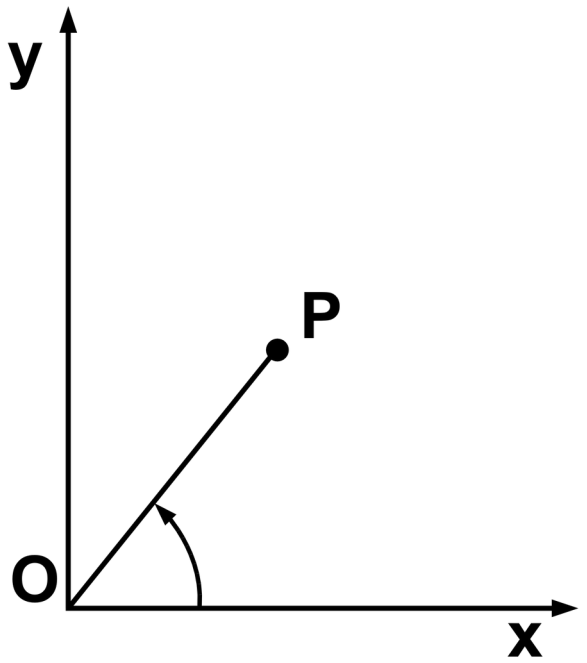
Explicit development process

- Explicitly documented development process
  - Less time spent reverse engineering
  - Interested readers are up to speed quickly
  - Starting point for future contributions

How?

How did we solve the problem?

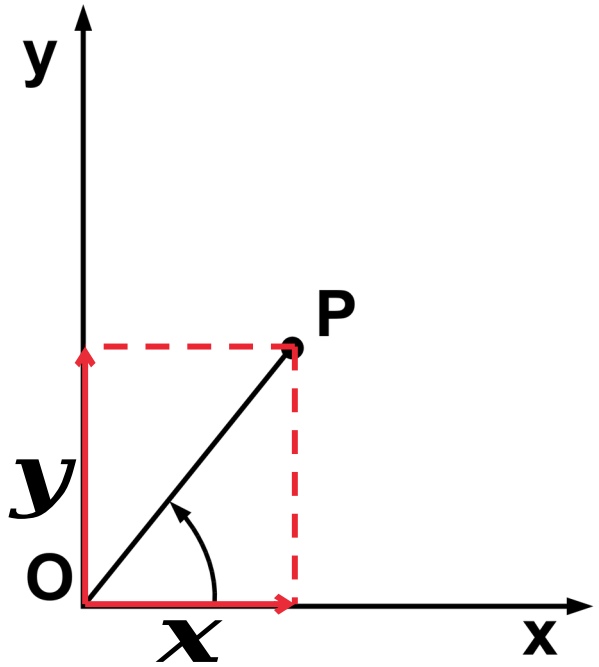
Running Example: Coordinate Systems



Running Example: Coordinate Systems

Cartesian coordinate system

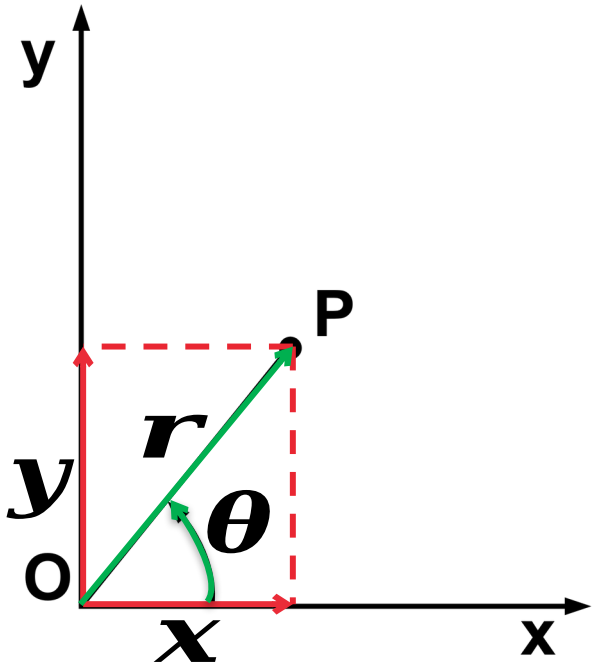
- 2-dimensional
  - -coordinate: distance from -axis
  - -coordinate: distance from -axis
- Efficient computation of translation



Running Example: Coordinate Systems

Cartesian coordinate system

- 2-dimensional
  - -coordinate: distance from -axis
  - -coordinate: distance from -axis
- Efficient computation of translation



Polar coordinate system

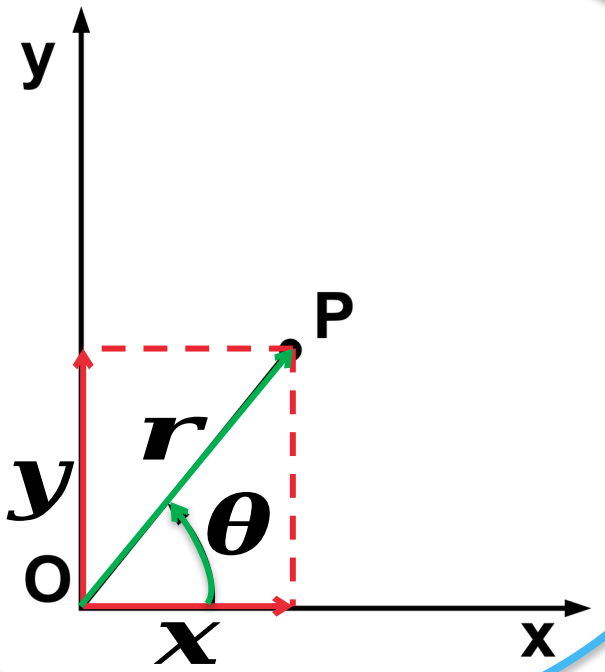
- 2-dimensional
  - -coordinate: distance from pole/origin
  - -coordinate: angle between ray and polar axis
- Efficient computation of rotation

Running Example: Coordinate Systems

$$\begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = \arctan\left(\frac{y}{x}\right) \end{cases}$$

Cartesian coordinate system

- 2-dimensional
  - -coordinate: distance from -axis
  - -coordinate: distance from -axis
- Efficient computation of translation



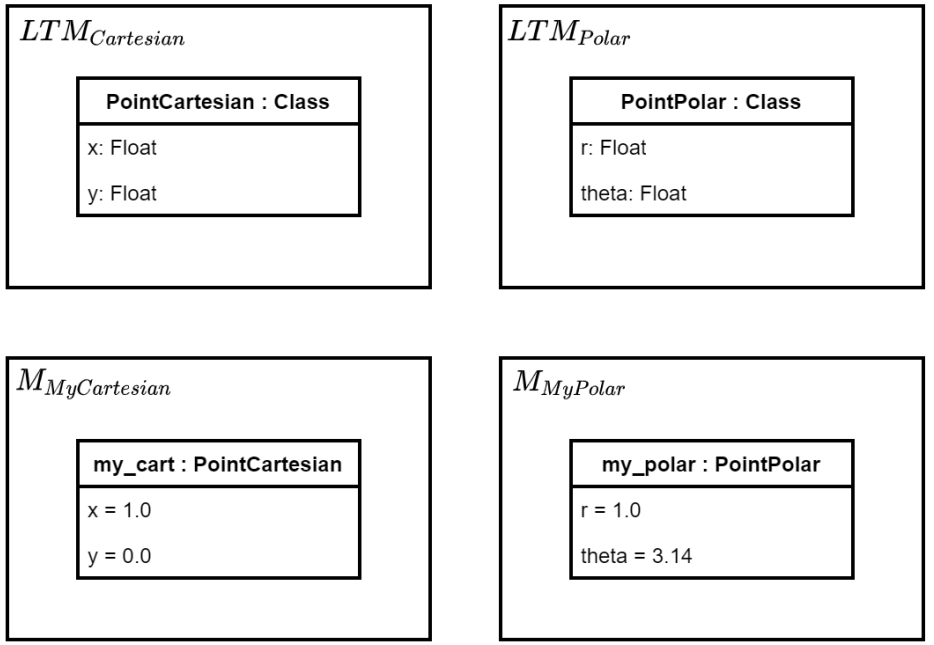
Polar coordinate system

- 2-dimensional
  - -coordinate: distance from pole/origin
  - -coordinate: angle between ray and polar axis
- Efficient computation of rotation

$$\begin{cases} x = r \cdot \cos(\theta) \\ y = r \cdot \sin(\theta) \end{cases}$$

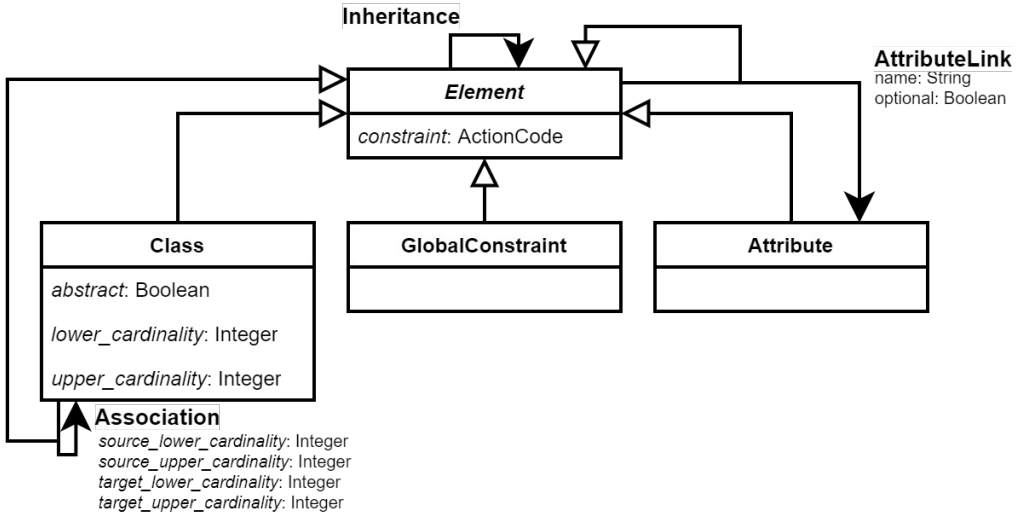
Meta-Modelling (Language Engineering)

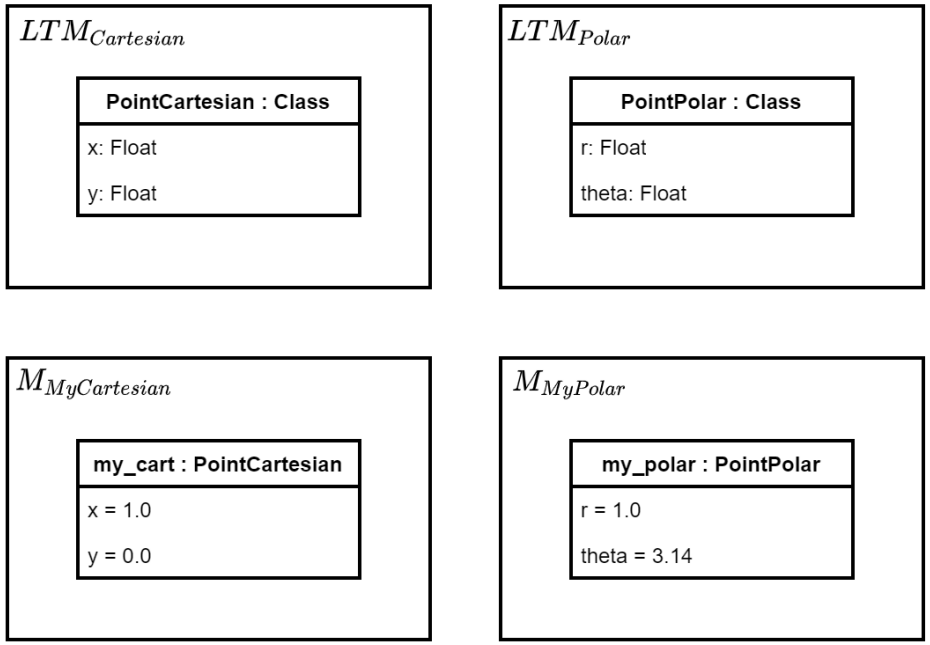
- **Define linguistic type model (LTM)**
  - Concepts
  - Concept structures
  - Interrelationships



Meta-Modelling (Language Engineering)

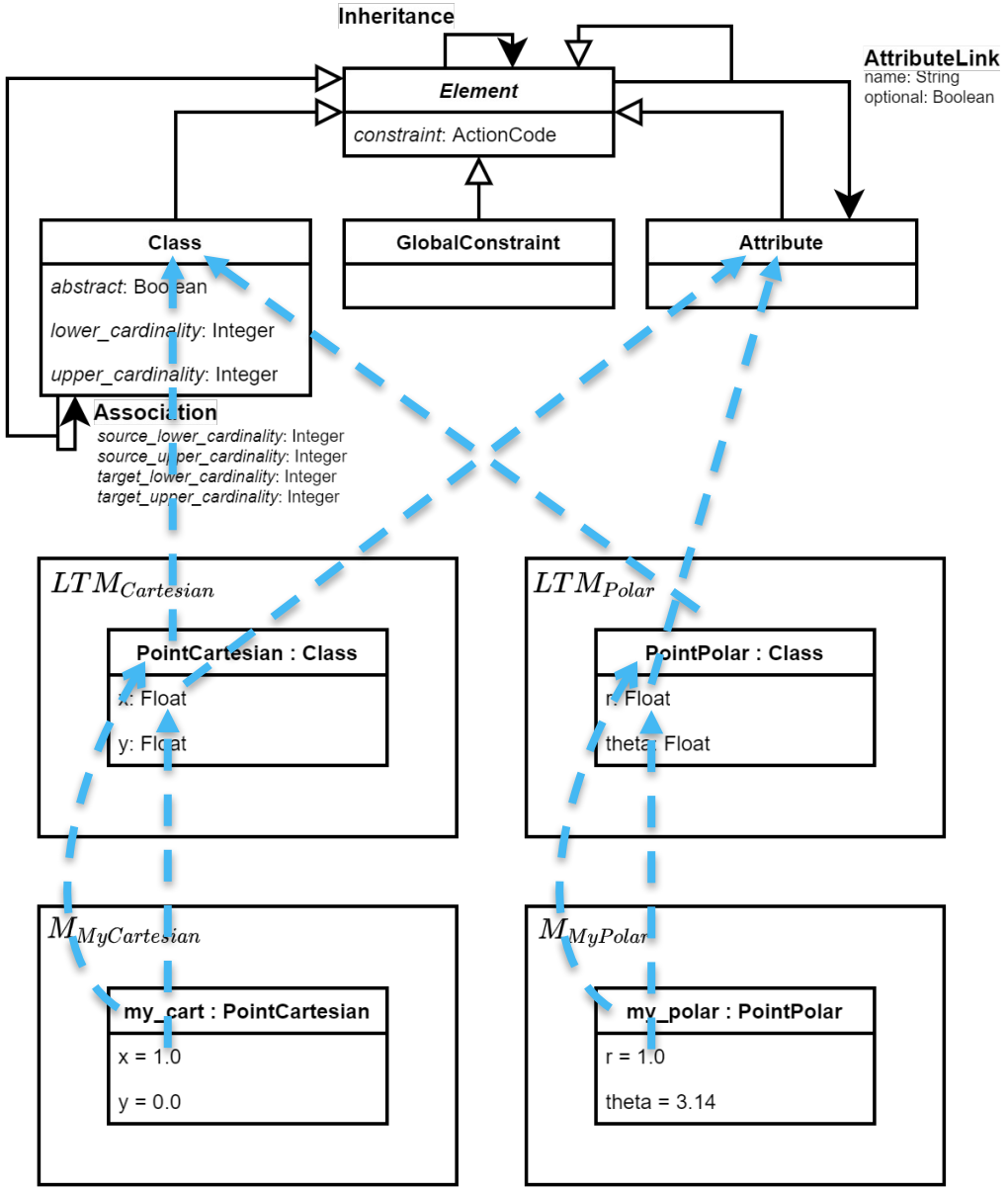
- **Define linguistic type model (LTM)**
  - Concepts
  - Concept structures
  - Interrelationships
- **Described using a meta-language**
  - Simplified form of UML class diagrams





Meta-Modelling (Language Engineering)

- **Define linguistic type model (LTM)**
  - Concepts
  - Concept structures
  - Interrelationships
- **Described using a meta-language**
  - Simplified form of UML class diagrams
- **Type/instance relationship**
  - By construction (instantiation)
  - *A posteriori* (conformance check)

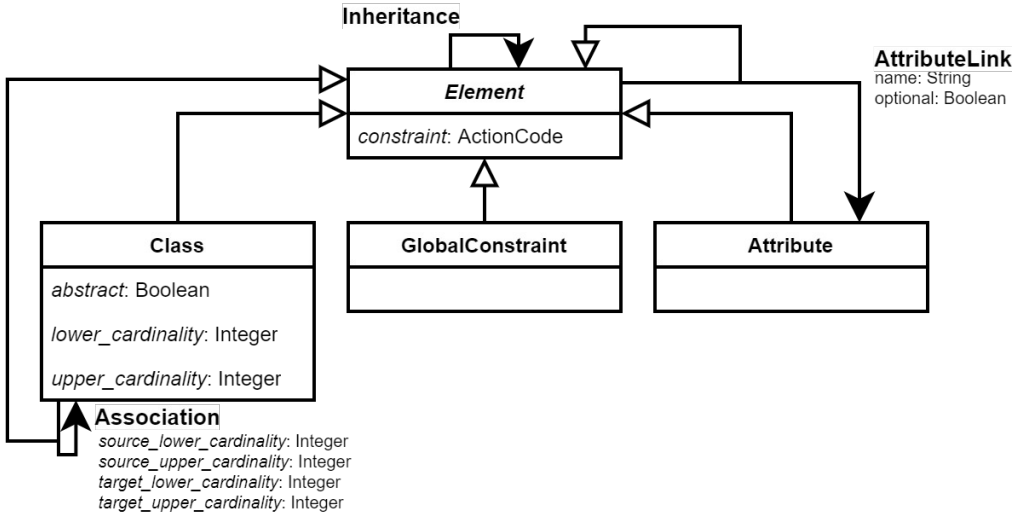


Modular Meta-Modelling

Contribution 1

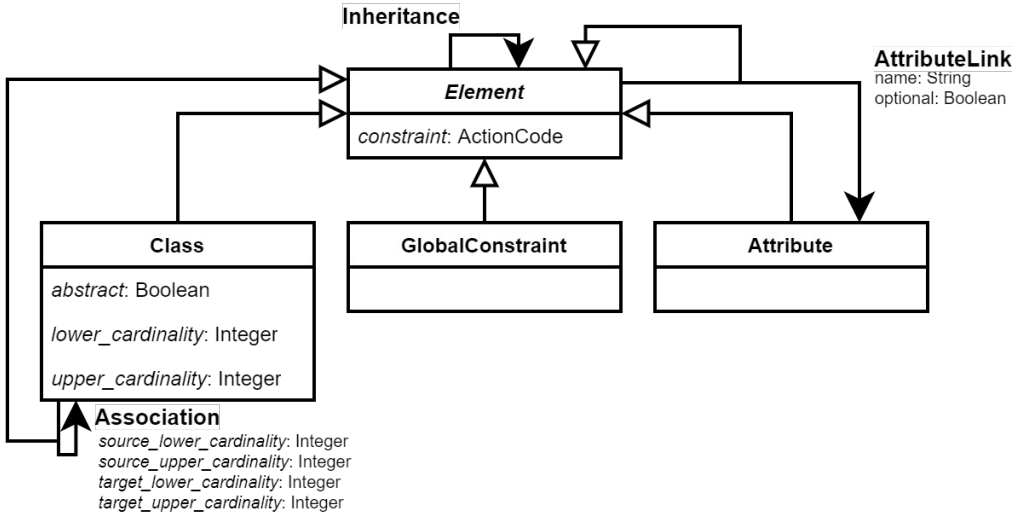
Modular Meta-Modelling

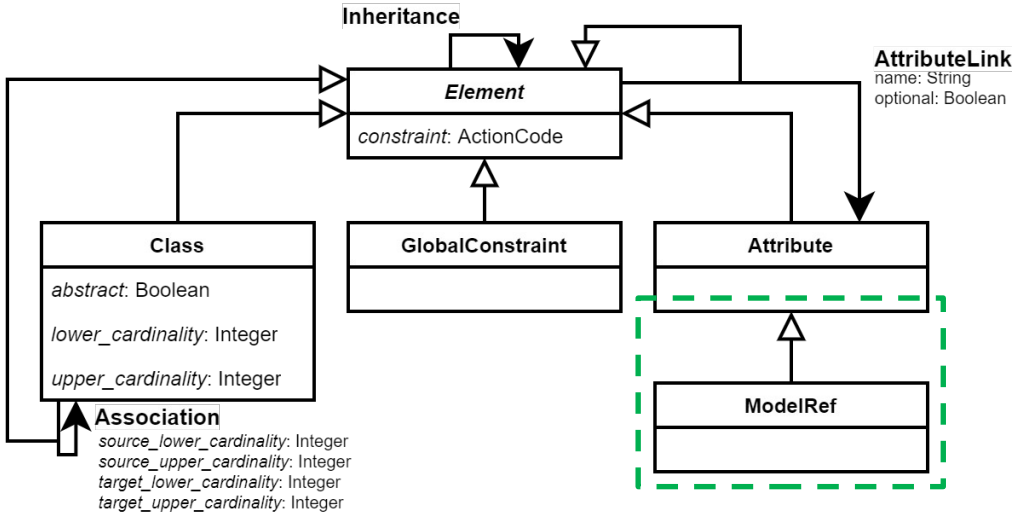
- **Modelverse's meta-language ()**
  - Redefinition of (primitive) types in each LTM



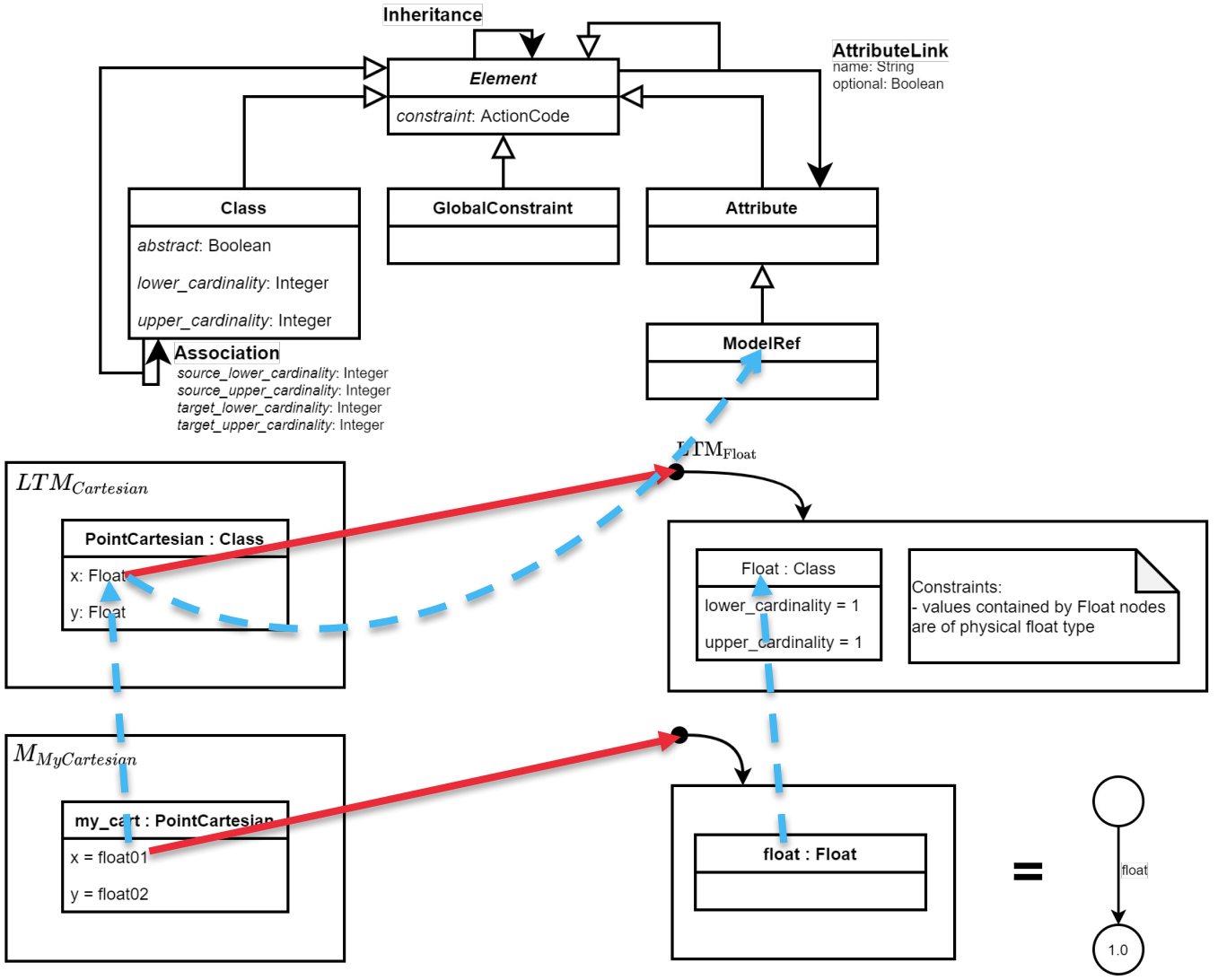
Modular Meta-Modelling

- **Modelverse's meta-language ()**
  - Redefinition of (primitive) types in each LTM
- **Proposed meta-language**
  - Allow references to other models
  - Enables modular composition





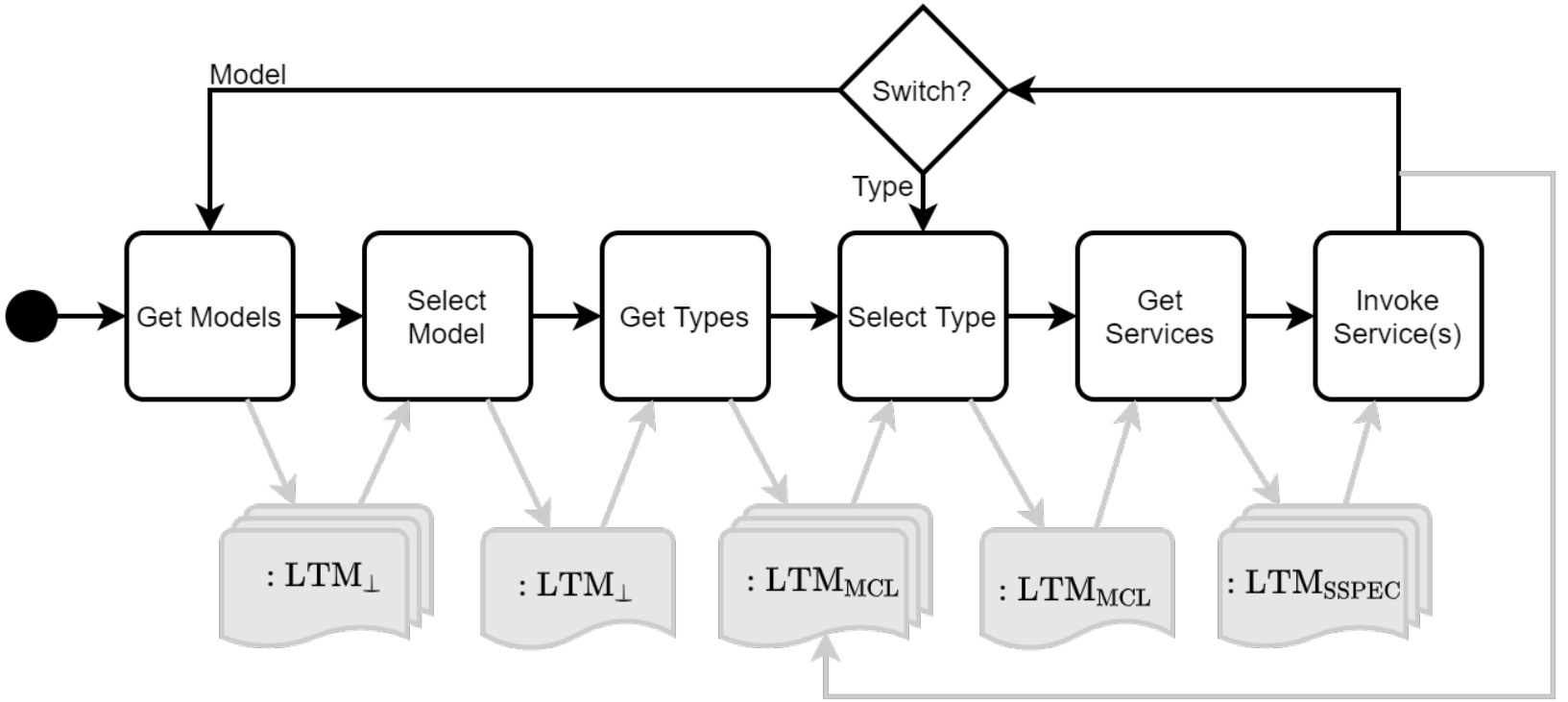
Modular Meta-Modelling: Example



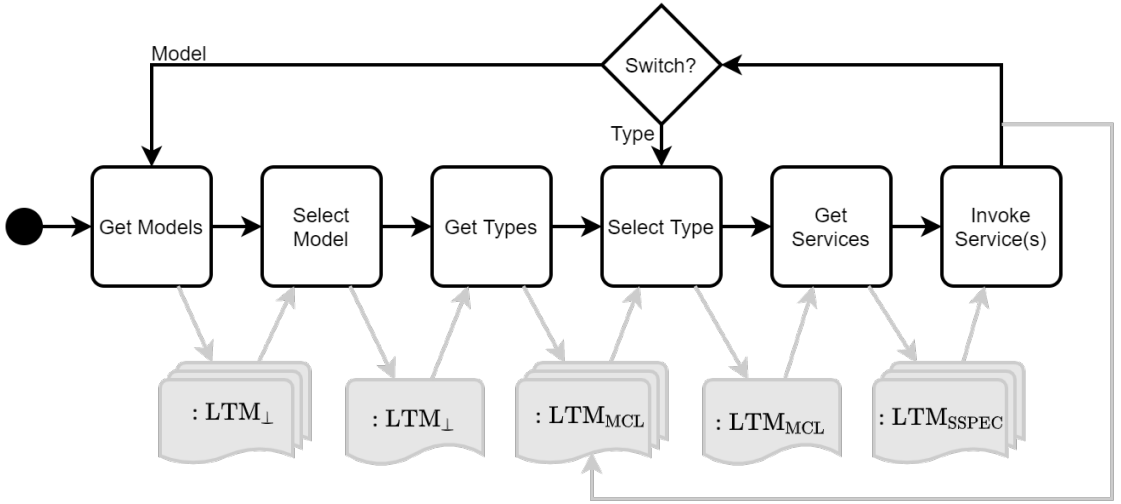
Multi-Conformance Meta-Modelling Framework

Contribution 2

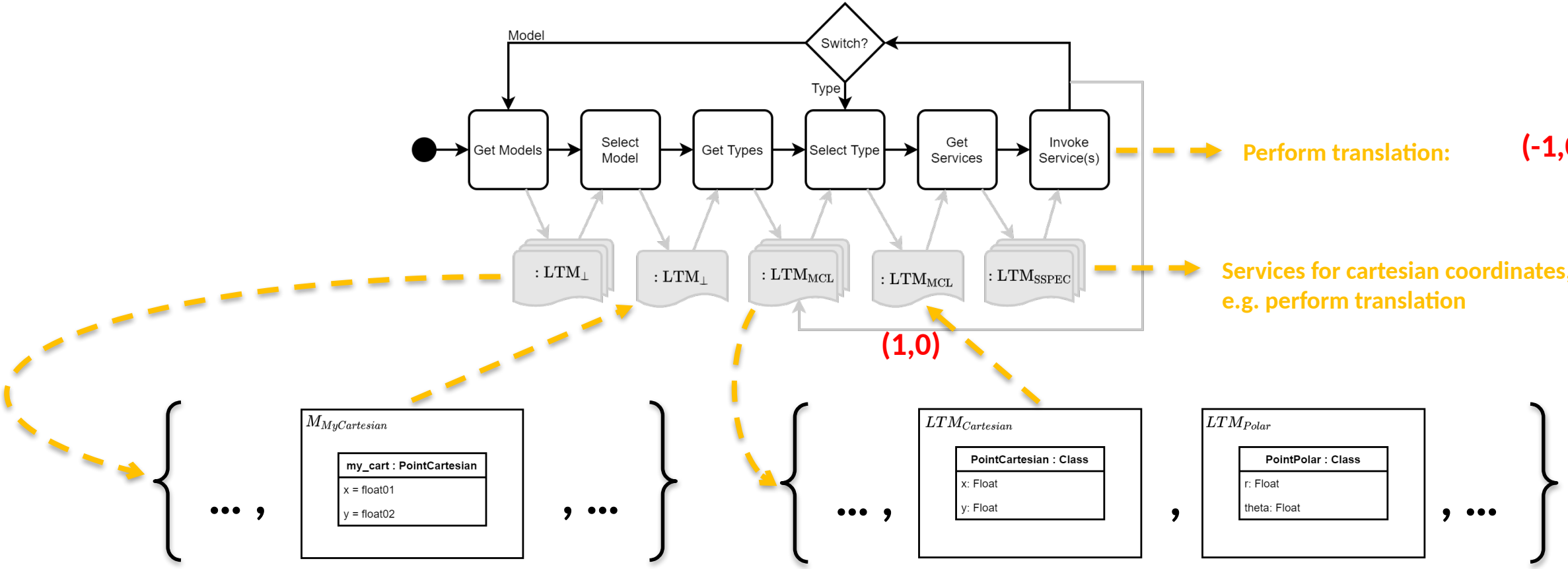
Framework: Workflow



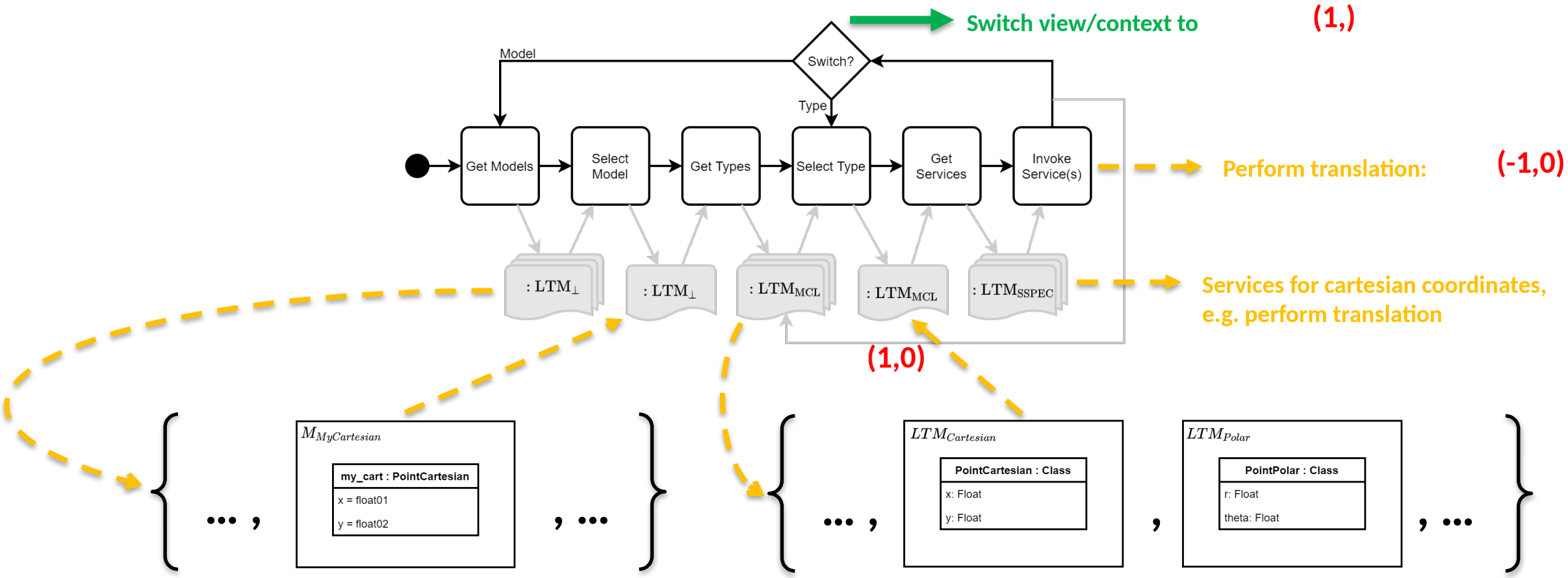
Framework: Workflow Example



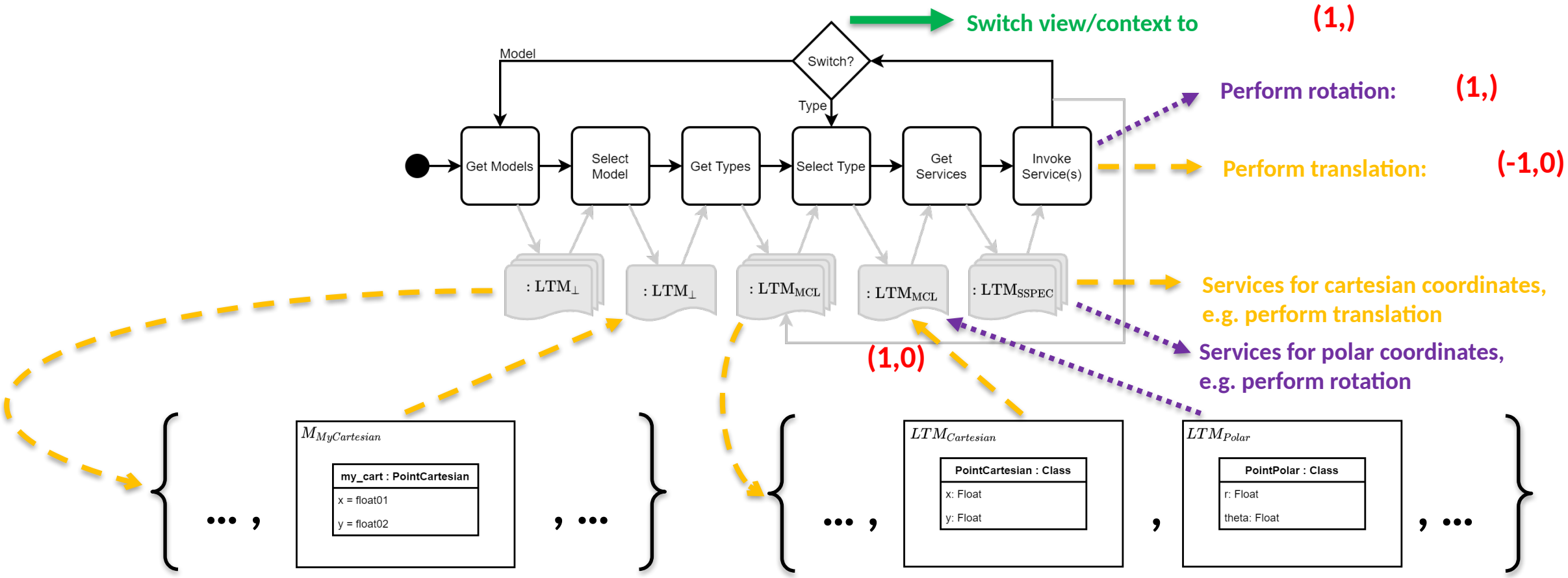
Framework: Workflow Example



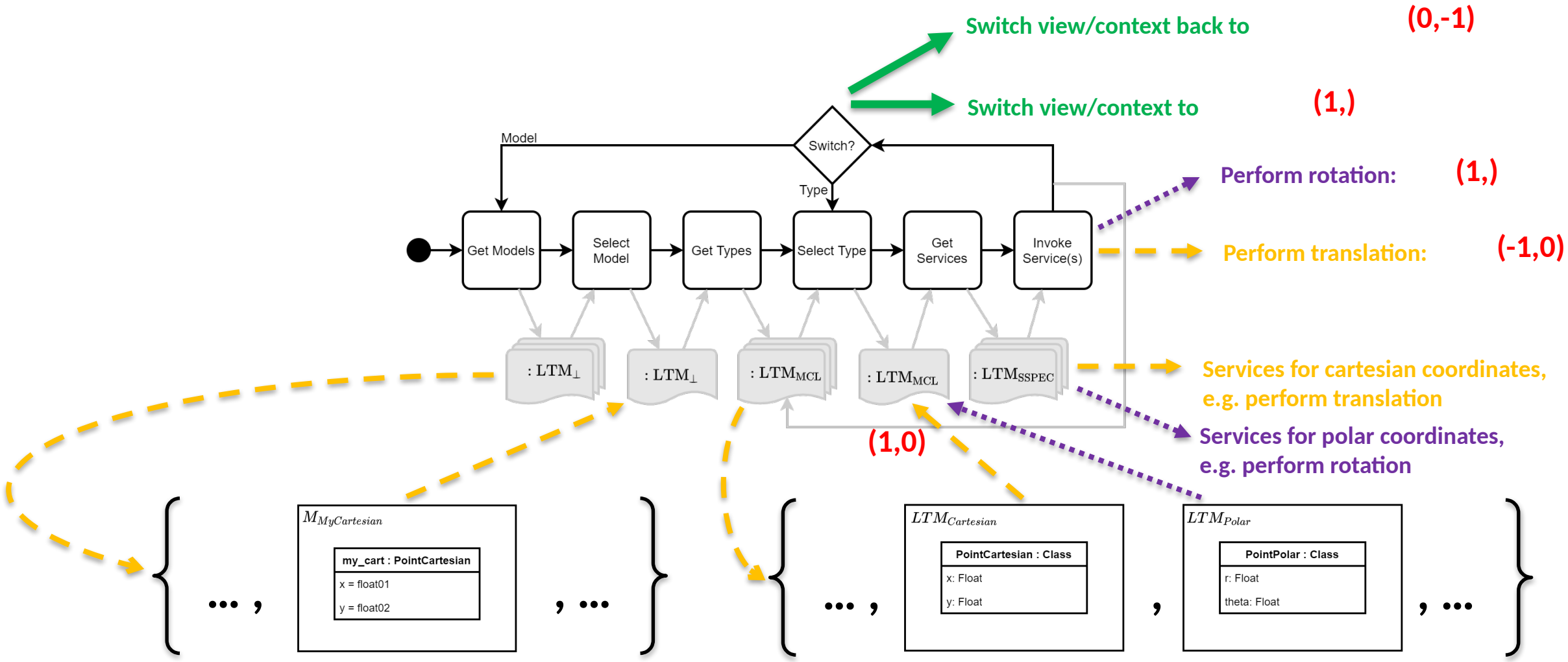
Framework: Workflow Example



Framework: Workflow Example

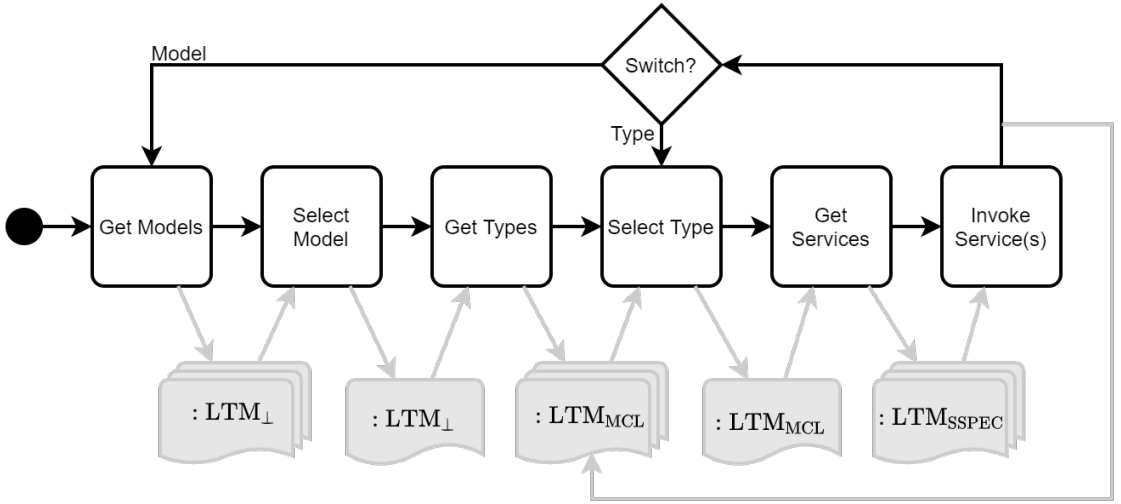


Framework: Workflow Example

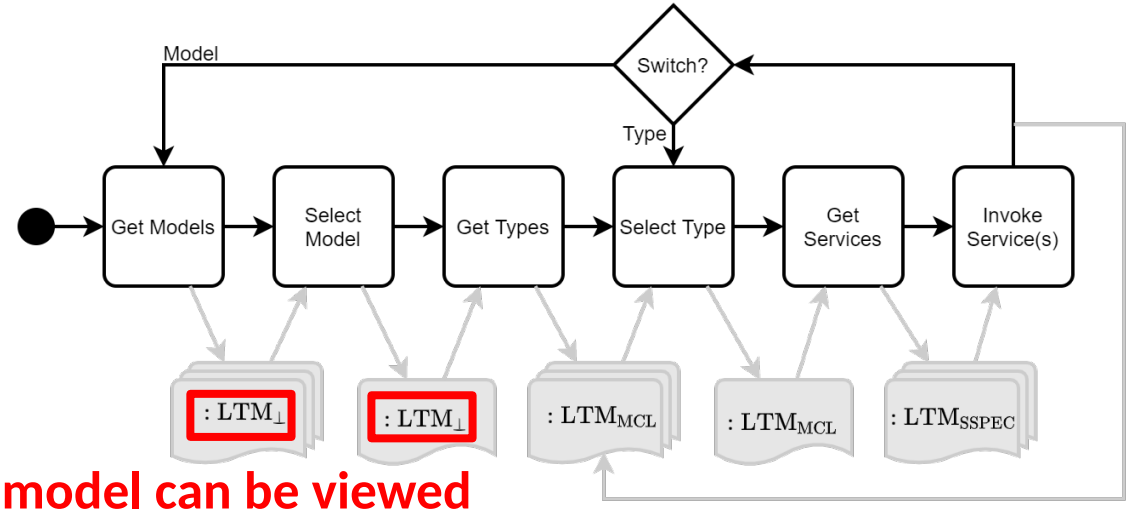


Demo

Framework: Multi-Conformance, Scalability and Reuse



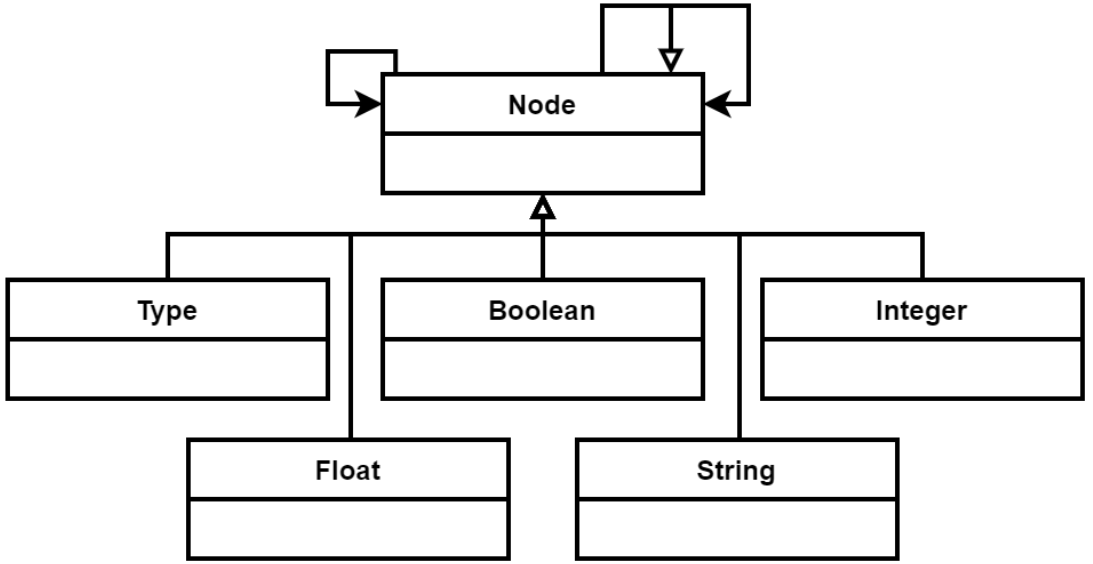
Framework: Multi-Conformance, Scalability and Reuse



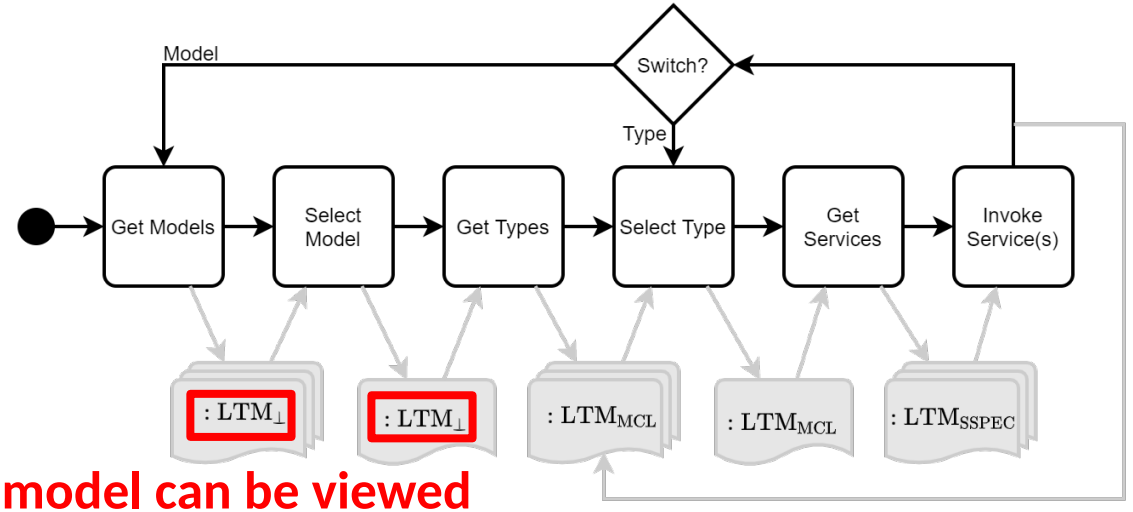
Every model can be viewed as a graph

Linguistic Type Model ()

- **Explicit representation of the physical type model**
  - Directed graph
  - Nodes and edges connected via edges
  - Nodes are empty or contain primitive values
- **<u>Axiom:</u> every model conforms to**
  - Canonical representation
  - Bootstrapping starting point
    - Decouple implementation/algorithms from internal data structure
  - Support for mega-modelling

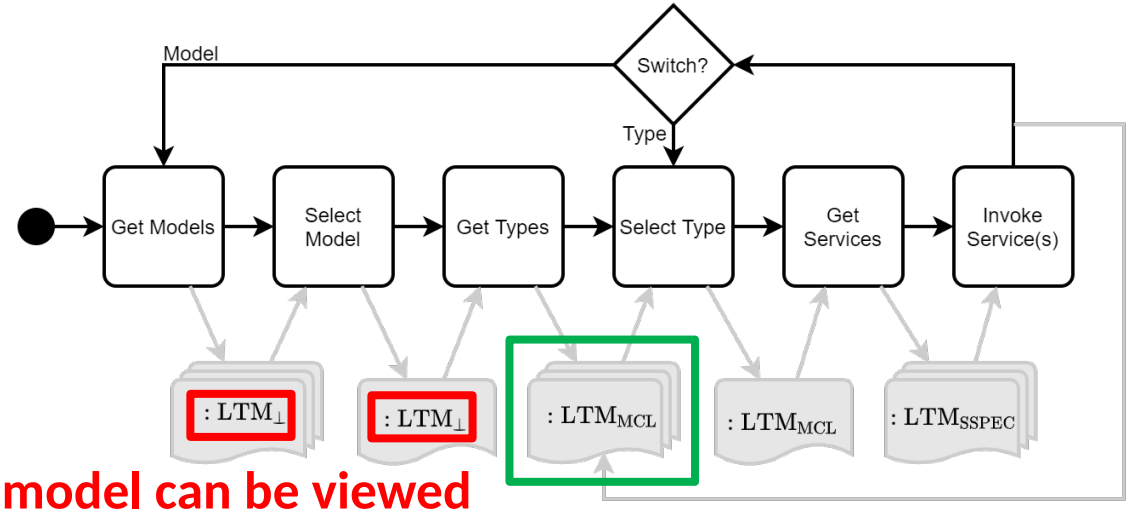


Framework: Multi-Conformance, Scalability and Reuse



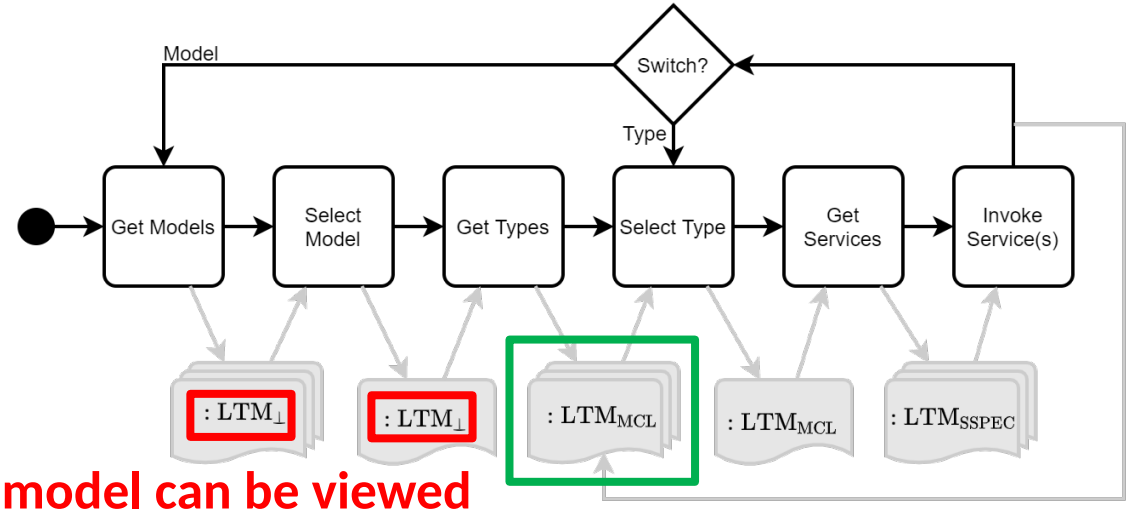
Every model can be viewed as a graph

Framework: Multi-Conformance, Scalability and Reuse



Every model can be viewed as a graph ⇨ is always part of the conformance set

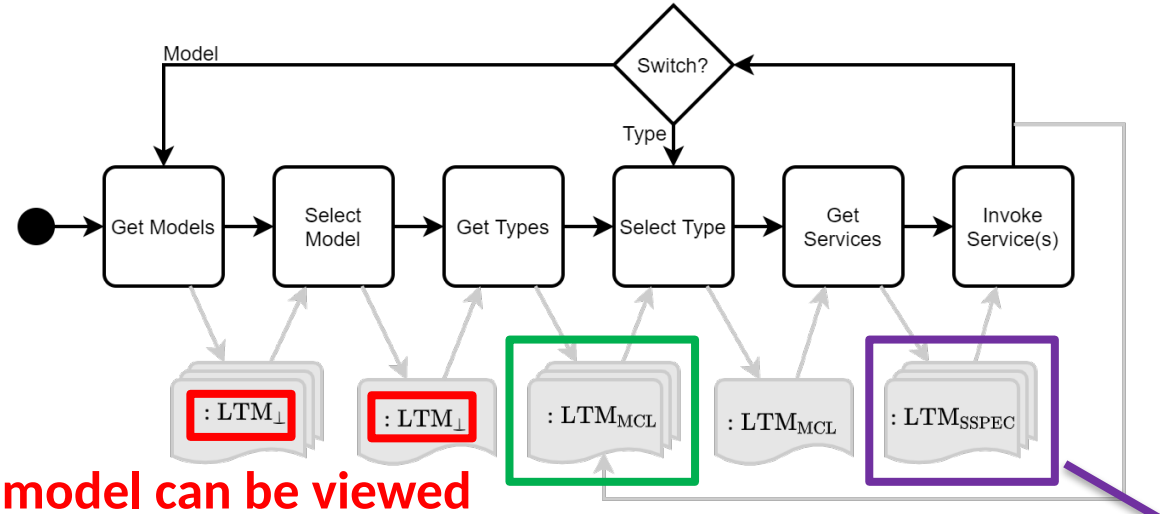
Framework: Multi-Conformance, Scalability and Reuse



Every model can be viewed as a graph ⇨ **is always part of the conformance set**

Conformance finding: structural **Conformance checking: nominal**

Framework: Multi-Conformance, Scalability and Reuse



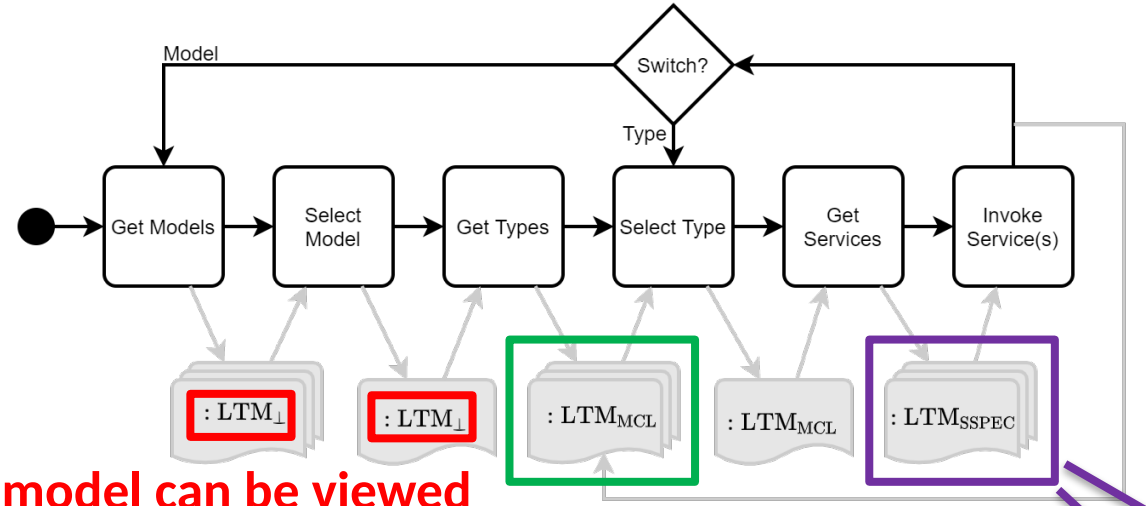
Every model can be viewed as a graph ⇨ is always part of the conformance set

At least CRUD operations

Conformance finding: structural

Conformance checking: nominal

Framework: Multi-Conformance, Scalability and Reuse



Every model can be viewed as a graph ⇨ is always part of the conformance set

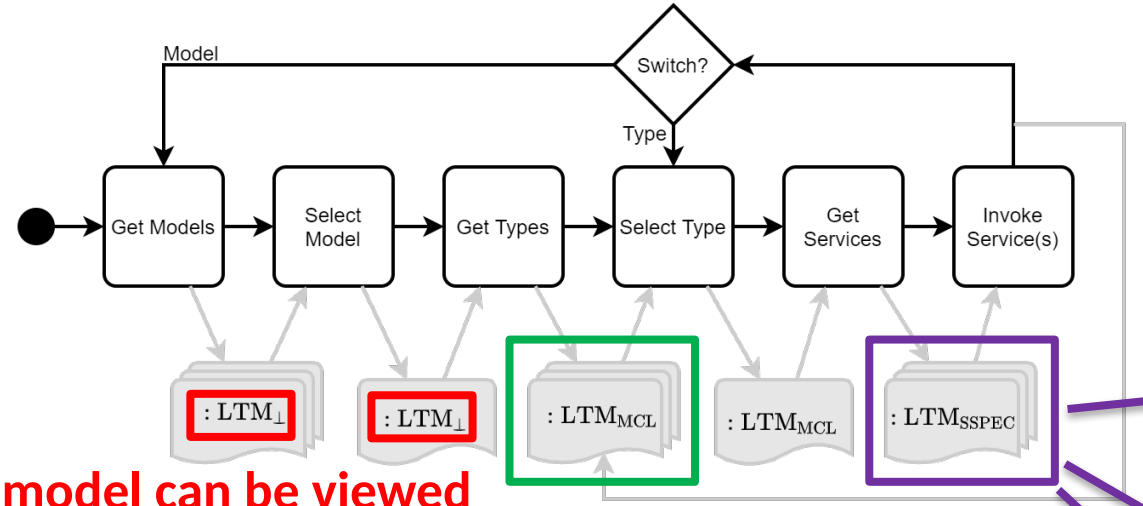
At least CRUD operations

Optionally, specialised operations, e.g. simulation...

Conformance finding: structural

Conformance checking: nominal

Framework: Multi-Conformance, Scalability and Reuse



Every model can be viewed as a graph ⇨ is always part of the conformance set

Implementation free to use any data structures/algorithms

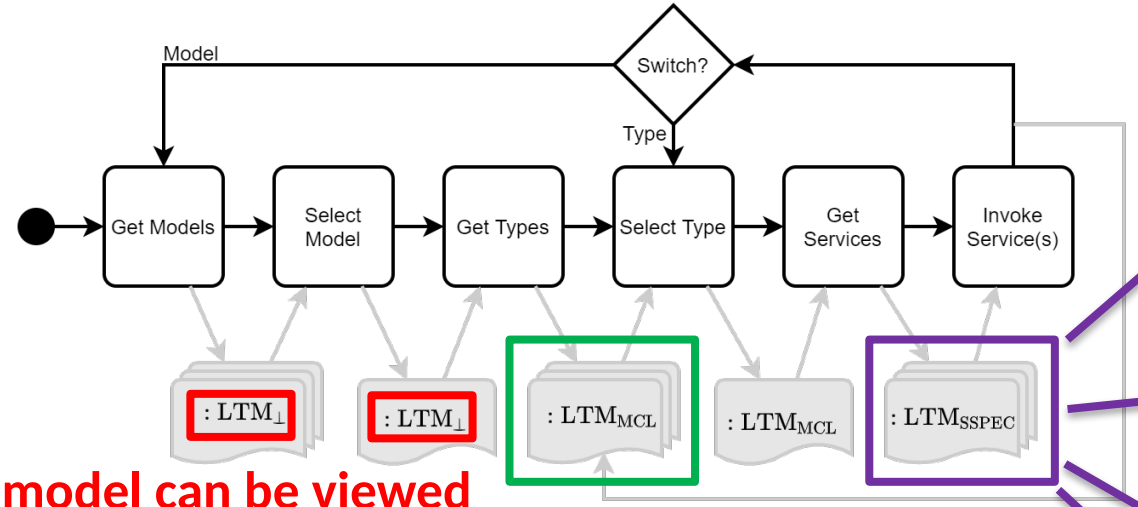
At least CRUD operations

Optionally, specialised operations, e.g. simulation...

Conformance finding: structural

Conformance checking: nominal

Framework: Multi-Conformance, Scalability and Reuse



Every model can be viewed as a graph ⇨ is always part of the conformance set

Transformation to/from

Implementation free to use any data structures/algorithms

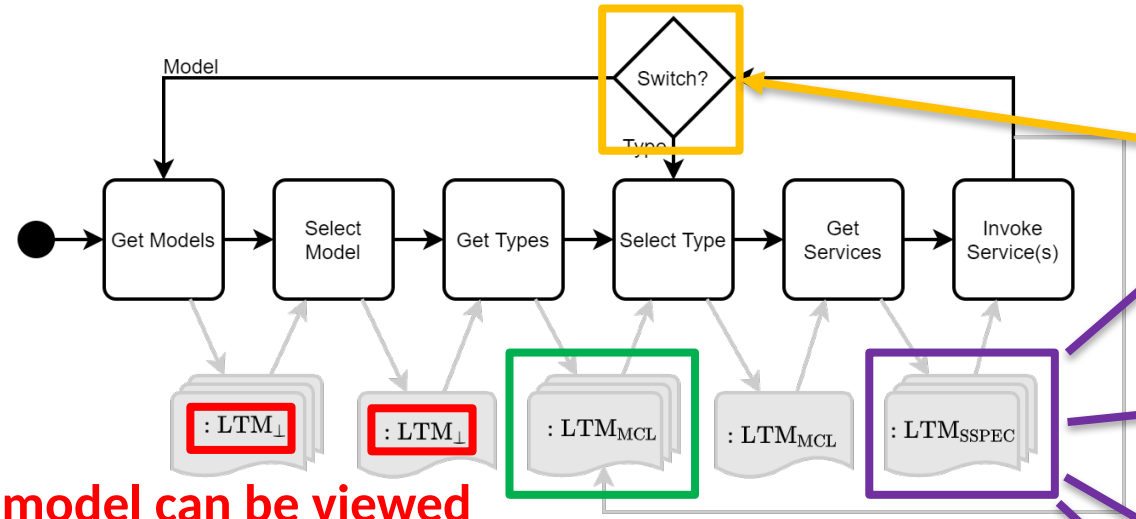
At least CRUD operations

Optionally, specialised operations, e.g. simulation...

Conformance finding: structural

Conformance checking: nominal

Framework: Multi-Conformance, Scalability and Reuse



transformations instead of transformations

Transformation to/from

Implementation free to use any data structures/algorithms

At least CRUD operations

Optionally, specialised operations, e.g. simulation...

Every model can be viewed as a graph ⇨ is always part of the conformance set

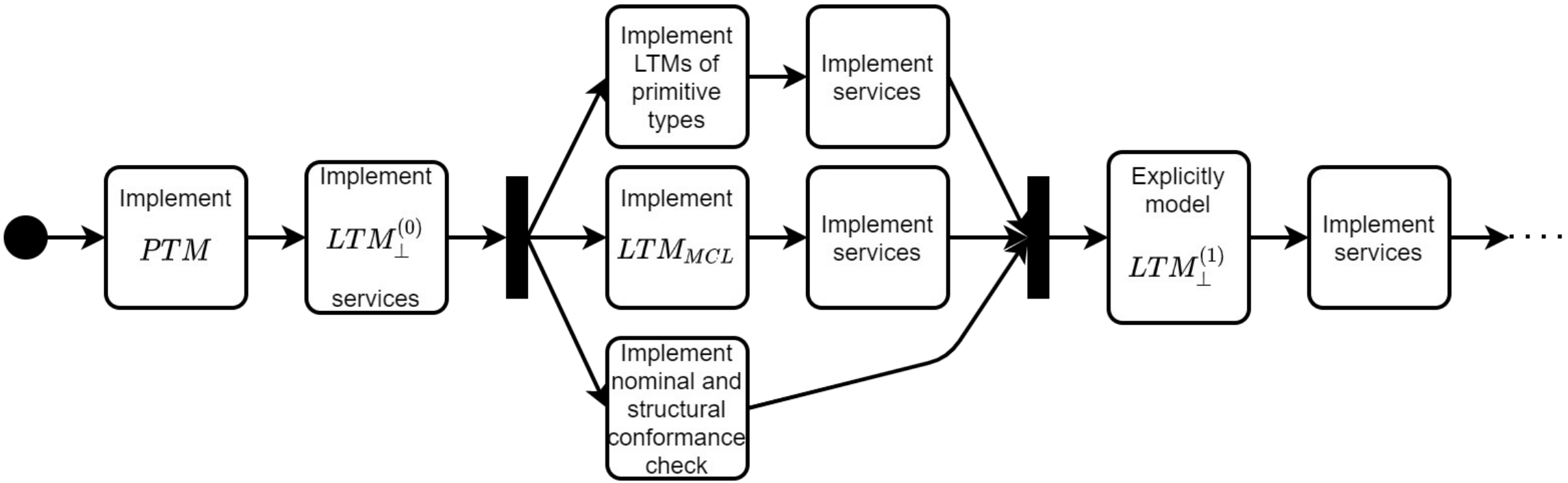
Conformance finding: structural

Conformance checking: nominal

Explicit Development Process

Contribution 3

Implementation/Bootstrapping Workflow



Summary of Contributions

- **Meta-language / with support for modular composition**
- **Meta-modelling framework**
  - Heterogeneous model storage and service implementation
  - Canonical graph representation
    - Multi-conformance
    - Context switching
- **Explicitly described implementation workflow**

Future Work

- **Addition of model transformations (reimplementation of RP2)**
  - Allows definition of translational and operational semantics
  - Full fledged language engineering environment
- **Continue the implementation/bootstrapping cycle**
  - Start bootstrapping the framework within itself
- **Efficient computation of conformance set**
  - For each model
  - Incrementally

University of Antwerp
Faculty of Science

Thank you for you attention

Questions?