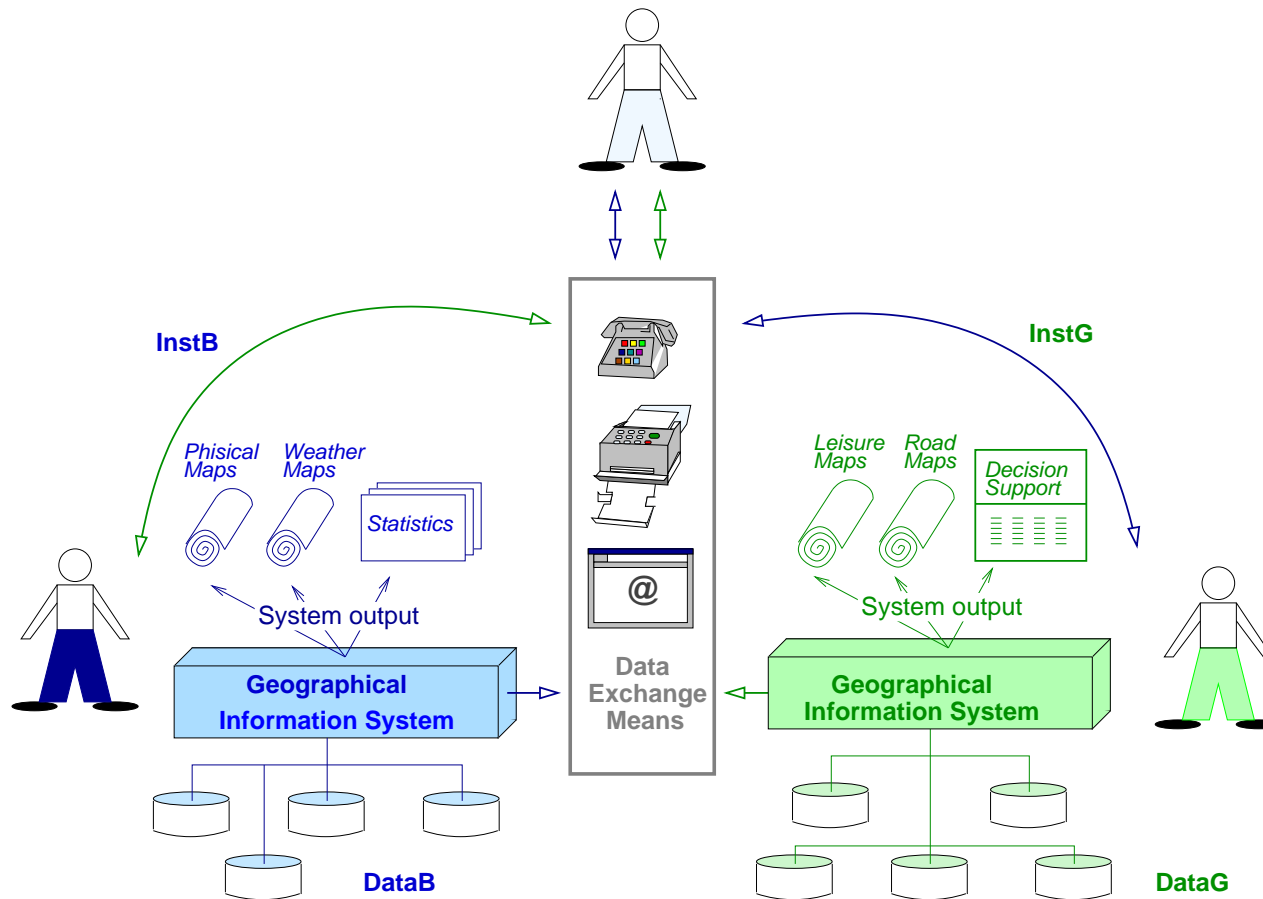


Spatio-temporal Schema Integration with Validation: A Practical Approach.

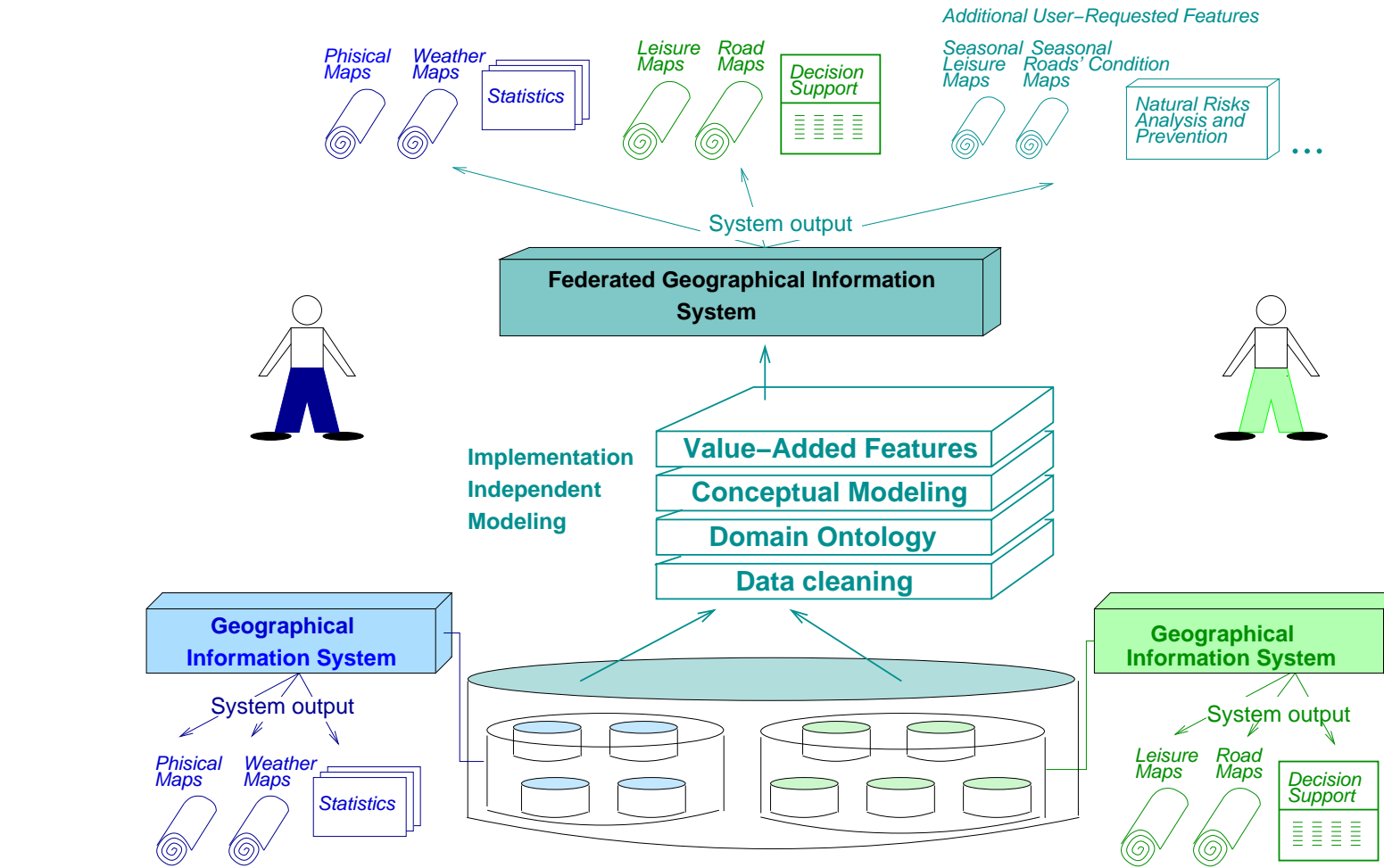
*Anastasiya Sotnykova, Nadine Cullot and Christelle Vangenot
EPFL & Université de Bourgogne*

03 November 2005, OTM Workshop - Semantic-based GIS

The context ... before

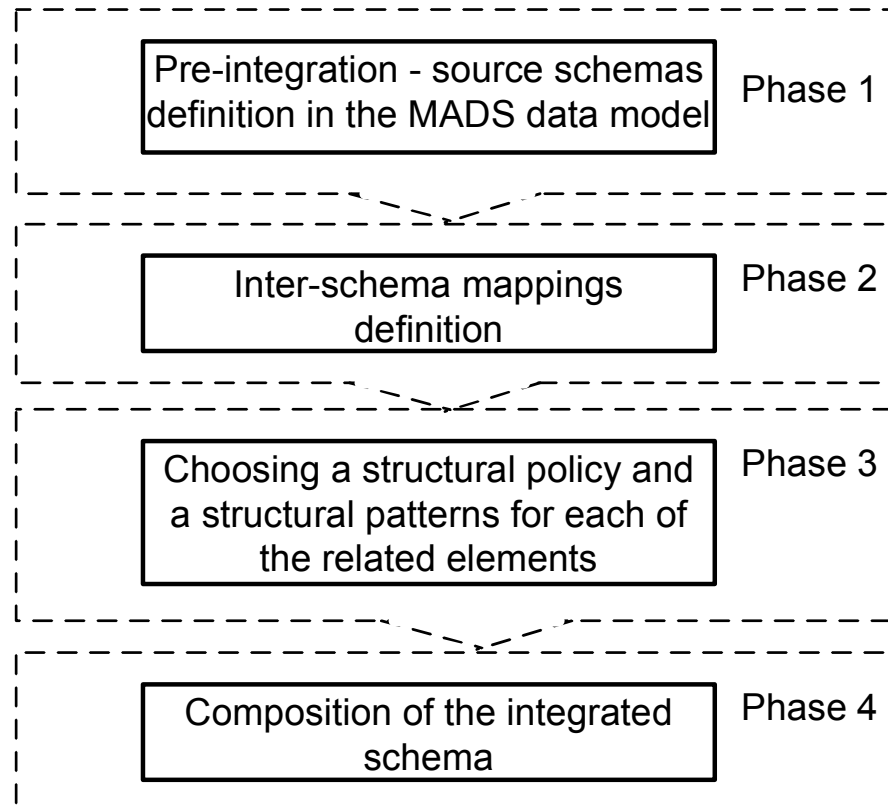


The context ... after



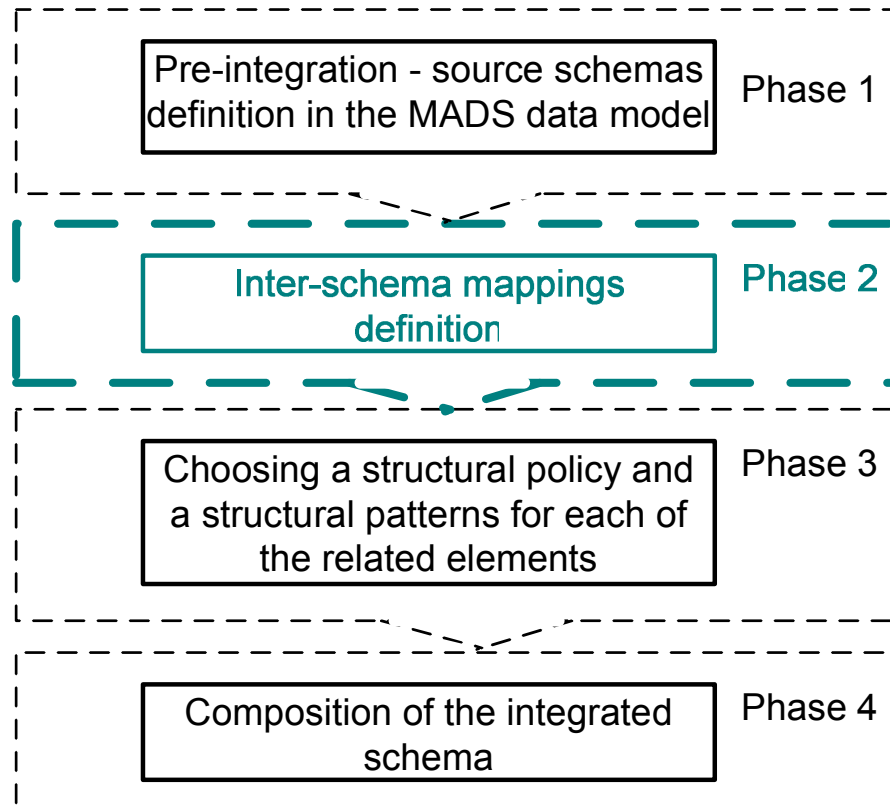
The MADS-based method

In the MADS data model



The MADS-based method

In the MADS data model



The context . . . semantics

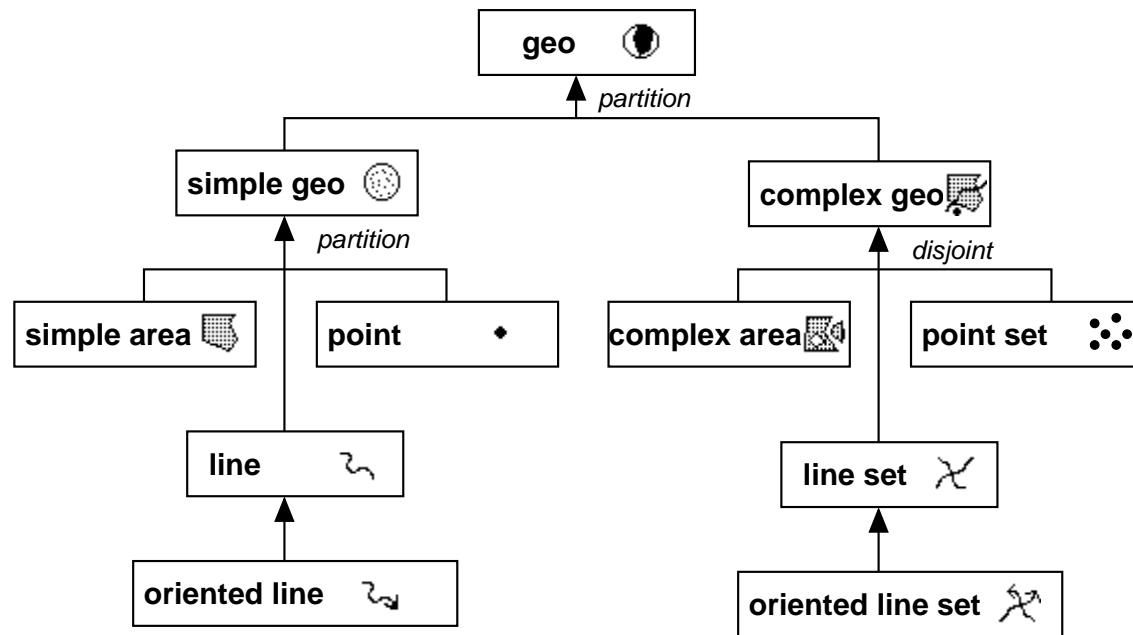
Table 1: How a 'building' object can be represented.

Purpose of representation	User
Architectural style and fitting in the neighborhood environment	Architect department of a city administration
Robustness of the construction of the building and the materials it is built of	Rescue crew of the city
Condition of the building and suitability for living in it	Renovation construction company
Location and dimensions of the building	Cadastral department of the city administration

Common Data Model (CMD) : MADS Data Model

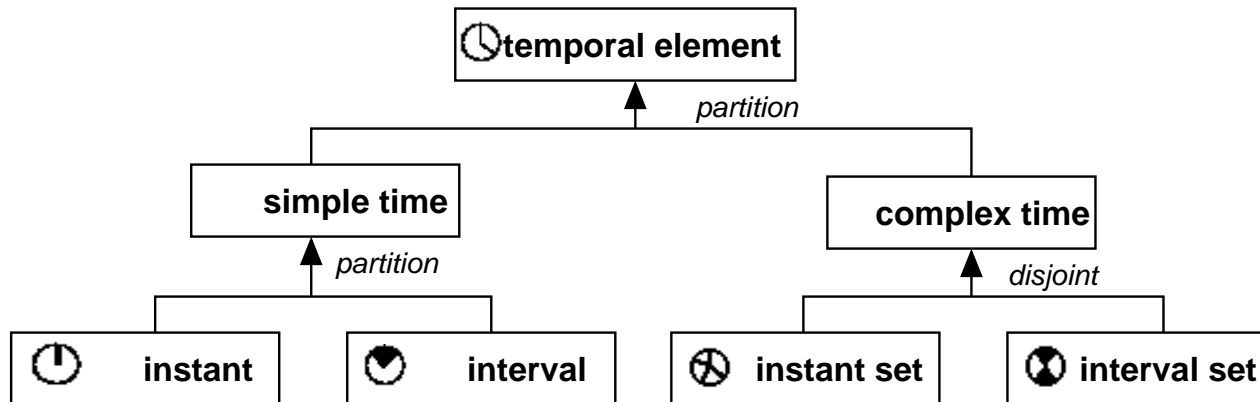
- thematic dimension - ER extended, graphical
- spatial dimension - has a predefined spatial primitives
- temporal dimension - time can be modeled with inbuilt concepts
- topological relationships - relationships between spatial object types
- synchronisation relationships - relationships between temporal object types

MADS spatial



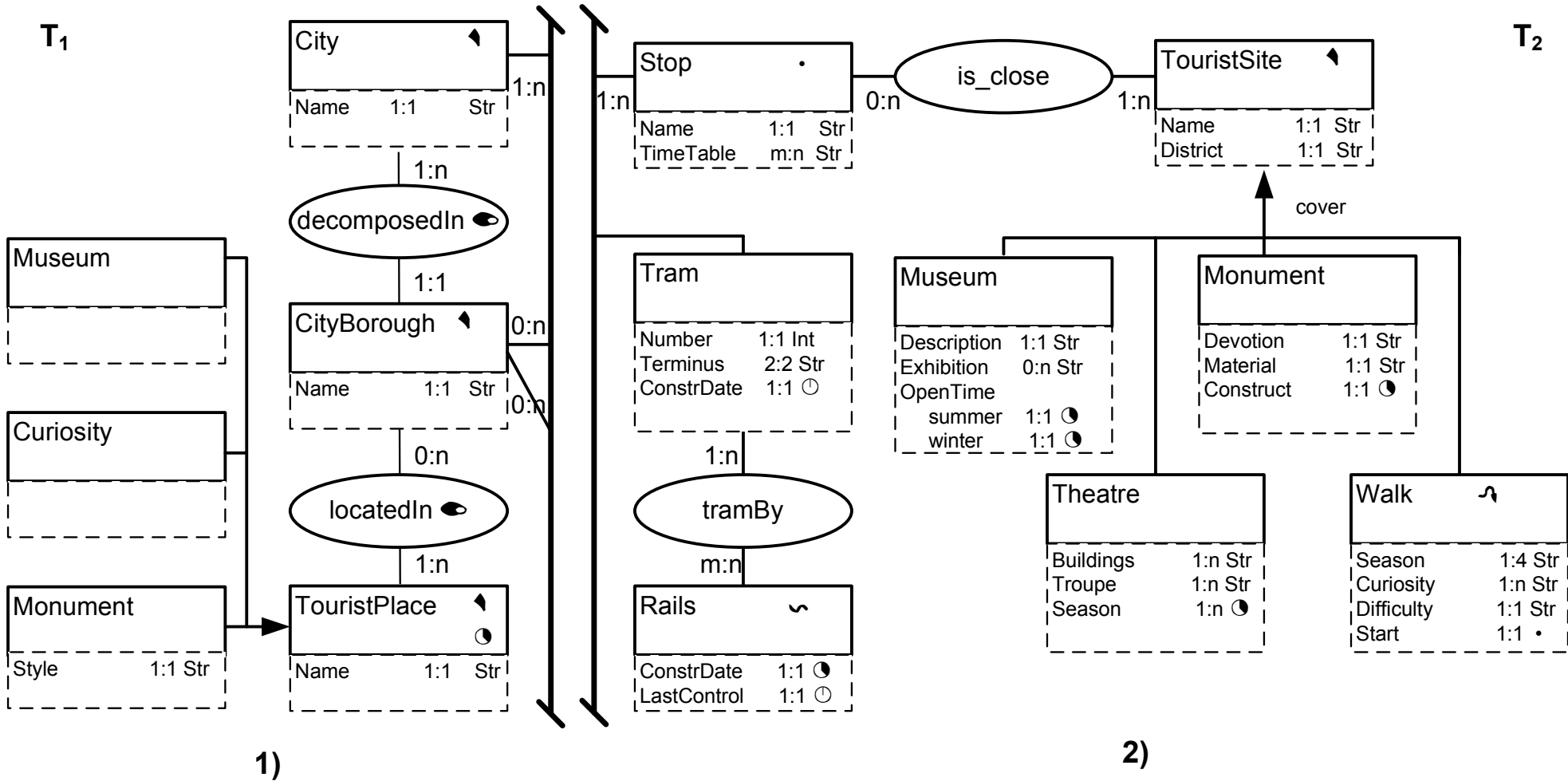
<i>Topological</i>	<i>Icon</i>	<i>Topological</i>	<i>Icon</i>
disjunction	○●	overlapping	◐
adjacency	○●	inclusion	◑
crossing	⋈	equality	●

MADS temporal



<i>Synchronization</i>	<i>Icon</i>	<i>Synchronization</i>	<i>Icon</i>
equal	≡	during	⊃
meets	⊢	starts	⊢
overlaps	⊃	finishes	⊃
before	⊢		

Schemas T_1 and T_2



Inter-schema mappings

- (1) $\text{TouristPlace}_{T_1} \subseteq \text{TouristSite}_{T_2}$;
- (2) $\text{Museum}_{T_2} \subseteq \text{Museum}_{T_1}$;
- (3) $\text{Monument}_{T_2} \subseteq \text{Monument}_{T_1}$;
- (4) $\text{Museum}_{T_1} \bullet \text{Museum}_{T_2}$;
- (5) $\text{Monument}_{T_2} \bullet \text{Monument}_{T_1}$;
- (6) $\text{CityBorough}_{T_1}.\text{name} = \text{TouristSite}_{T_2}.\text{district}$;
- (7) $\text{TouristPlace}_{T_1} \bullet \text{TouristSite}_{T_2}$
- (8) $\text{TouristPlace}_{T_1}.\text{name} = \text{TouristSite}_{T_2}.\text{name}$;

Validation objectives

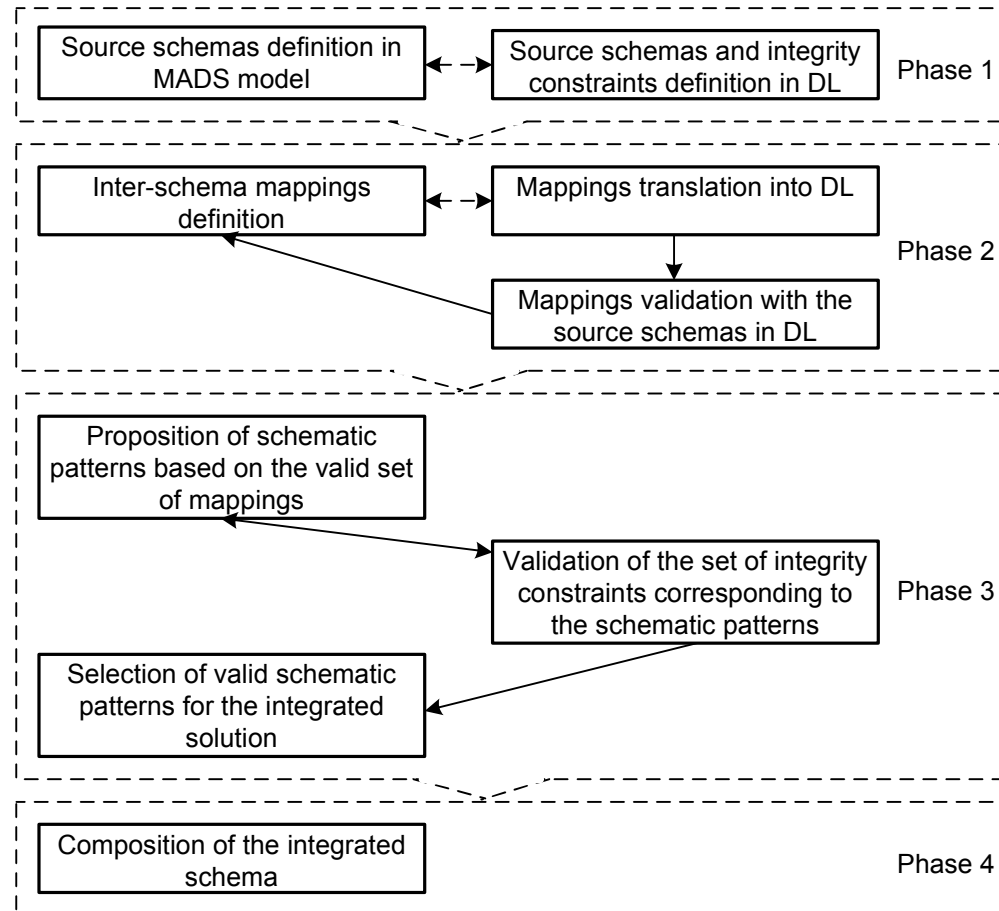
Invalid Correspondences

- TouristPlace_{T₁} (Time : Interval) \nrightarrow (no Time) Theatre_{T₂}
- MetroLine_{T₁} (Geometry: Line) $\circ\bullet$ (Geometry : Point) Stop_{T₂}
- TouristPlace_{T₁} (Geometry : Area) \bullet (Geometry : Point) Stop_{T₂}
- Museum_{T₁} (Geometry: Area) $\circ\bullet$ (Geometry : Area) Museum_{T₂}
- Museum_{T₁} (Geometry: Area) \bullet (Geometry : Area) Museum_{T₂}

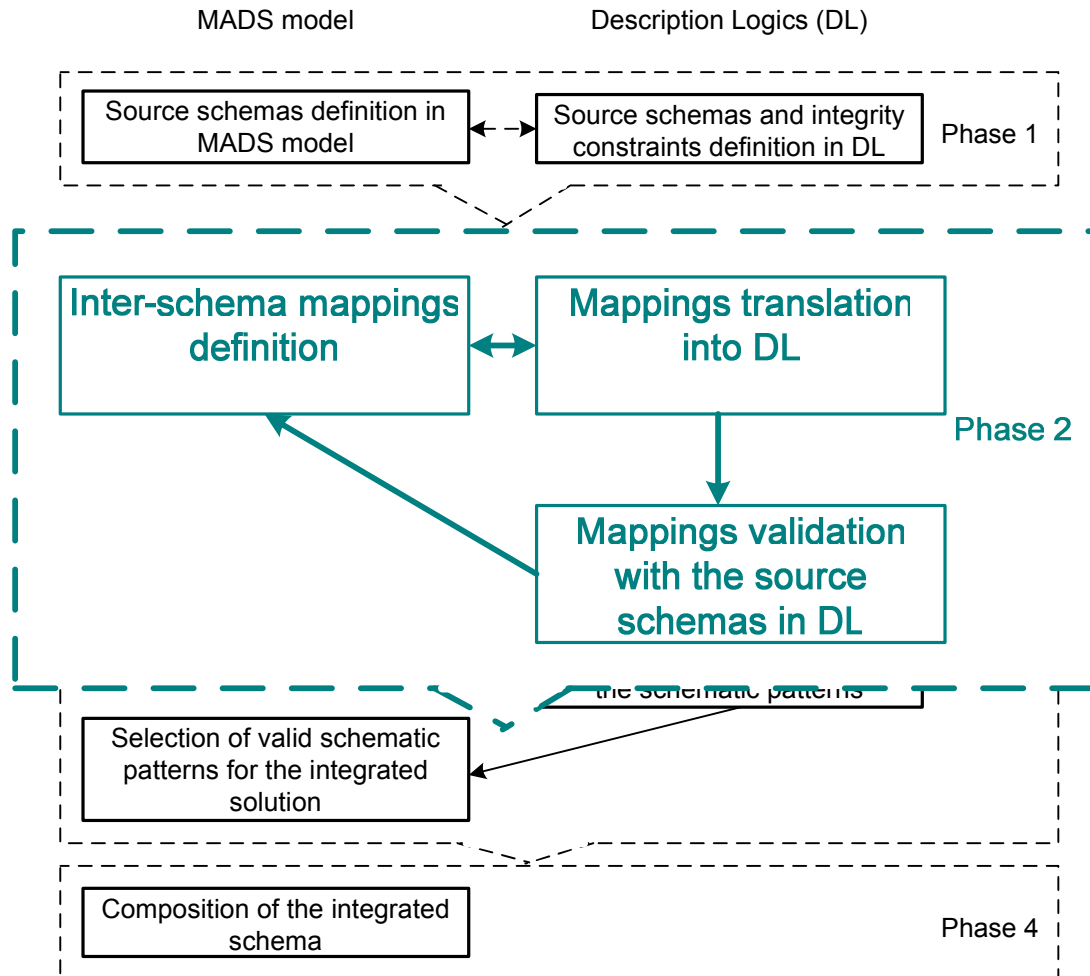
The MADS + DL method

MADS model

Description Logics (DL)



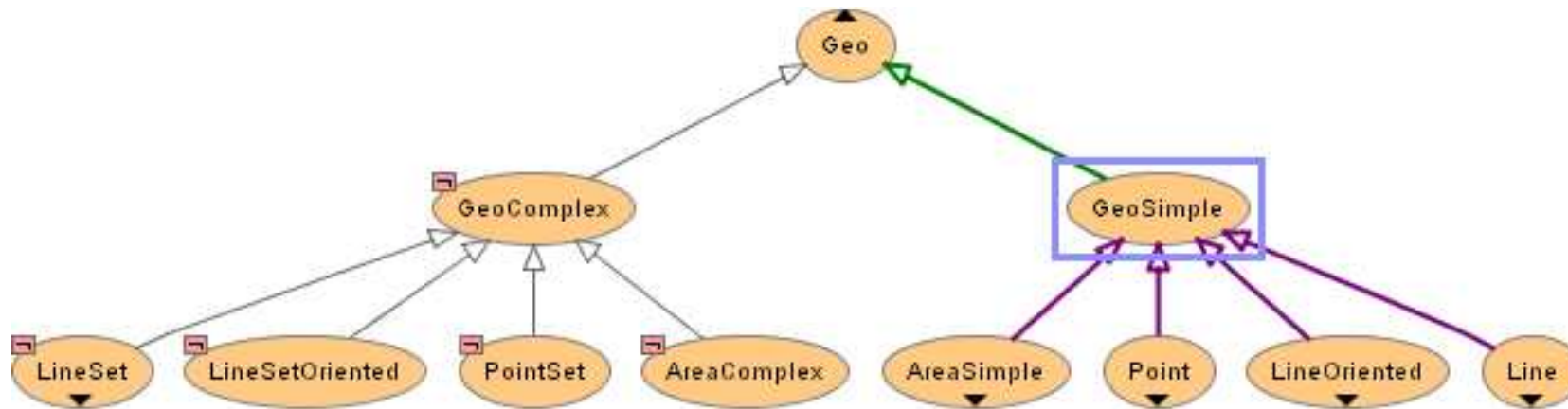
The MADS + DL method



MADS spatial dimension

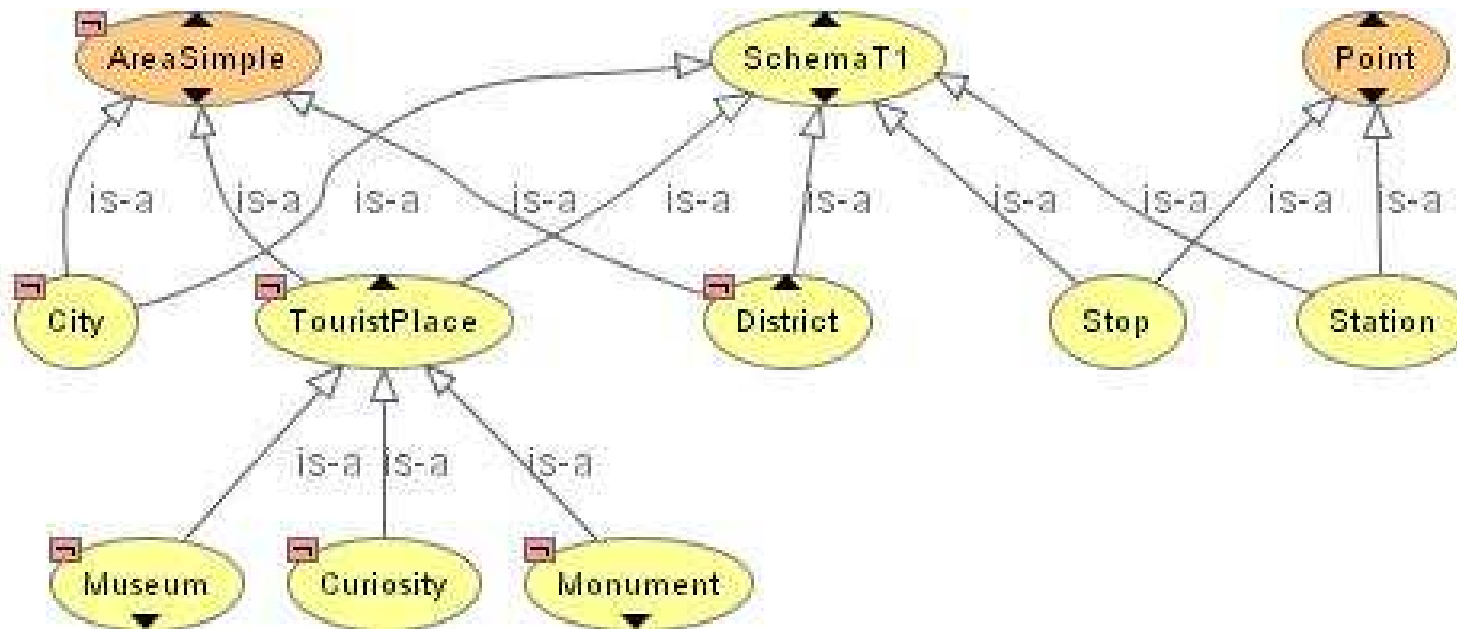
MADS spatial dimension	OWL spatial dimension
Spatial ADT hierarchy	Spatial abstract types hierarchy
Hierarchy in spatial subclasses	not possible in OWL-DL
Mandatory Geometry attribute that defines spatial features	Intrinsic hasGeometry property with predefined values

Spatiality : Types

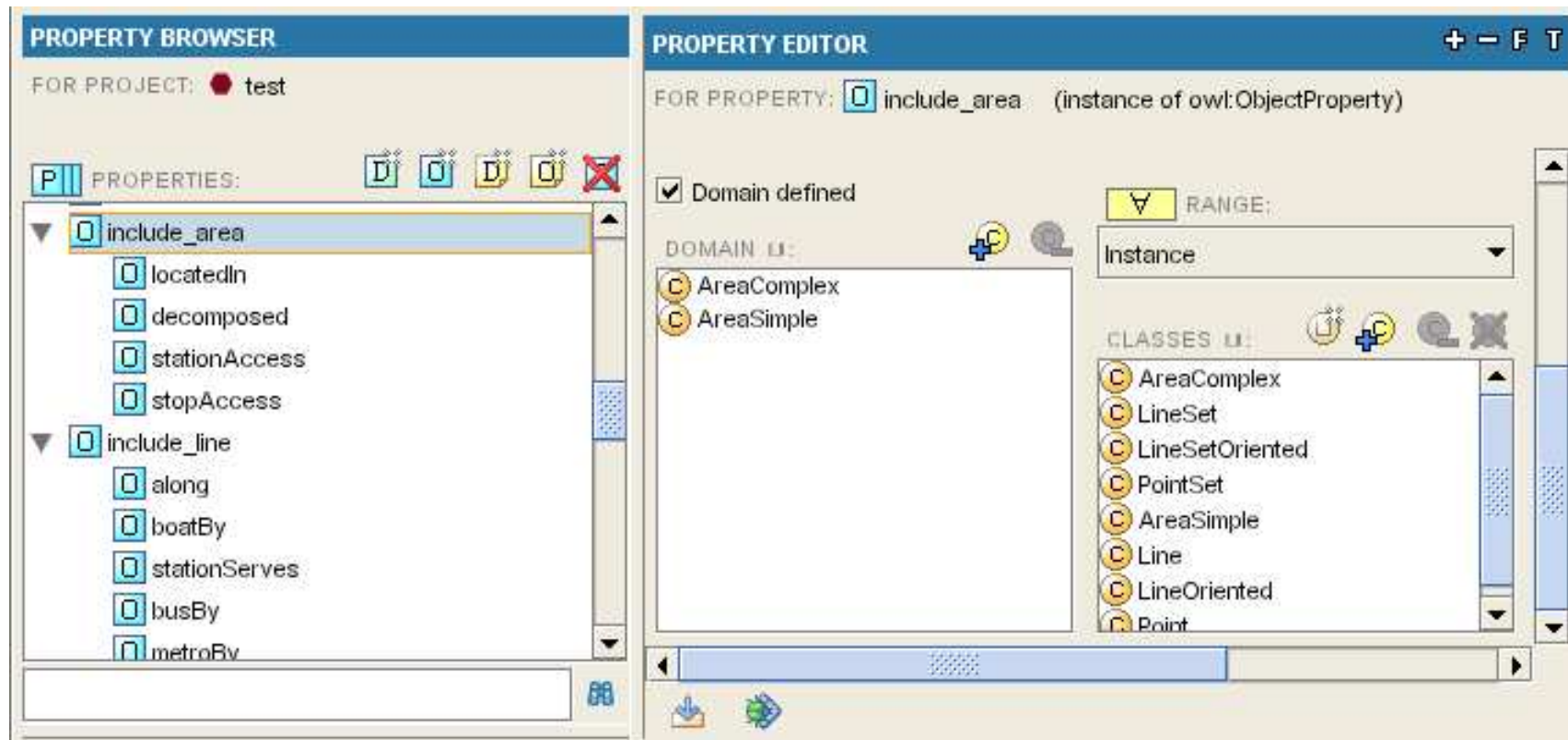


hasGeometry property defined as the necessary & sufficient

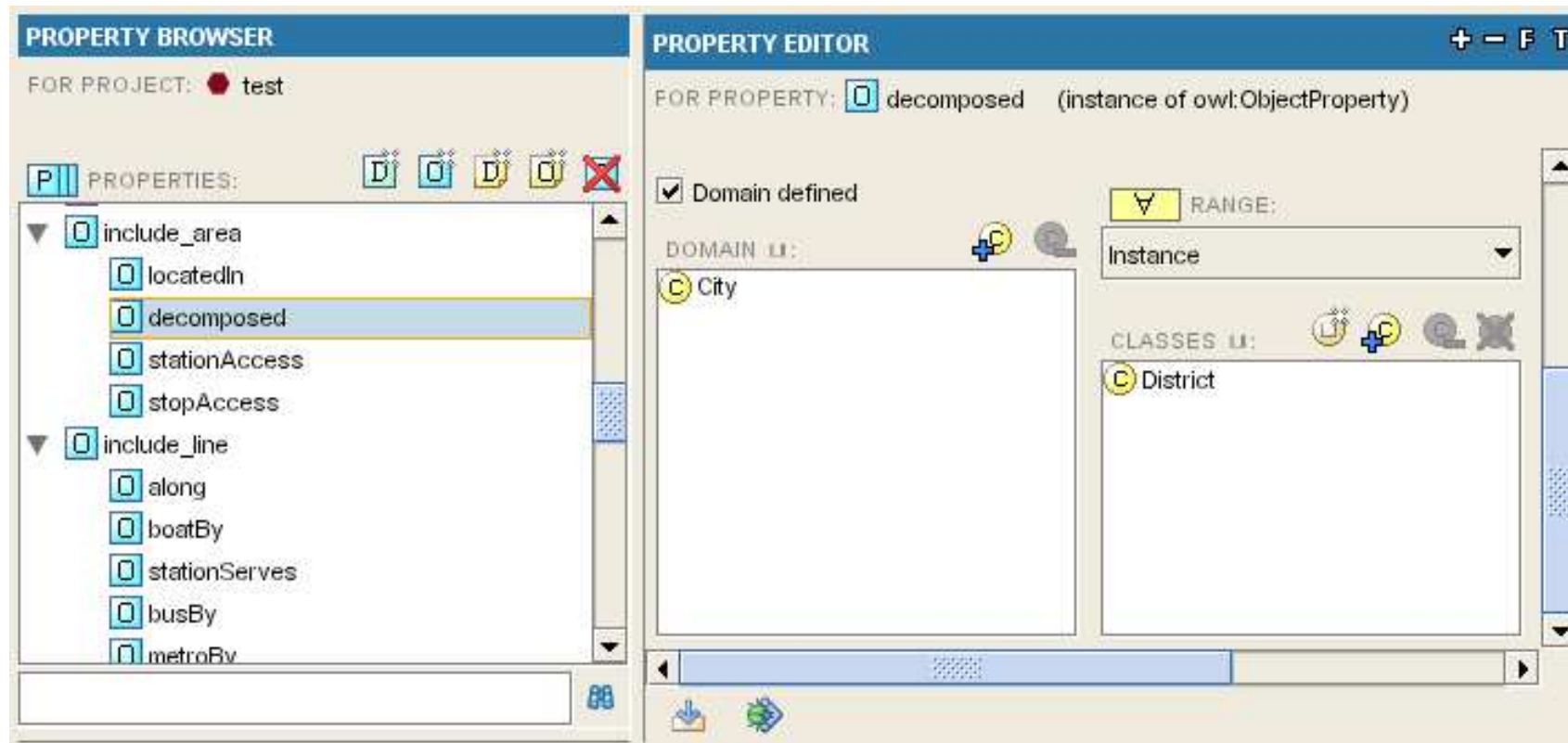
Spatiality : Types



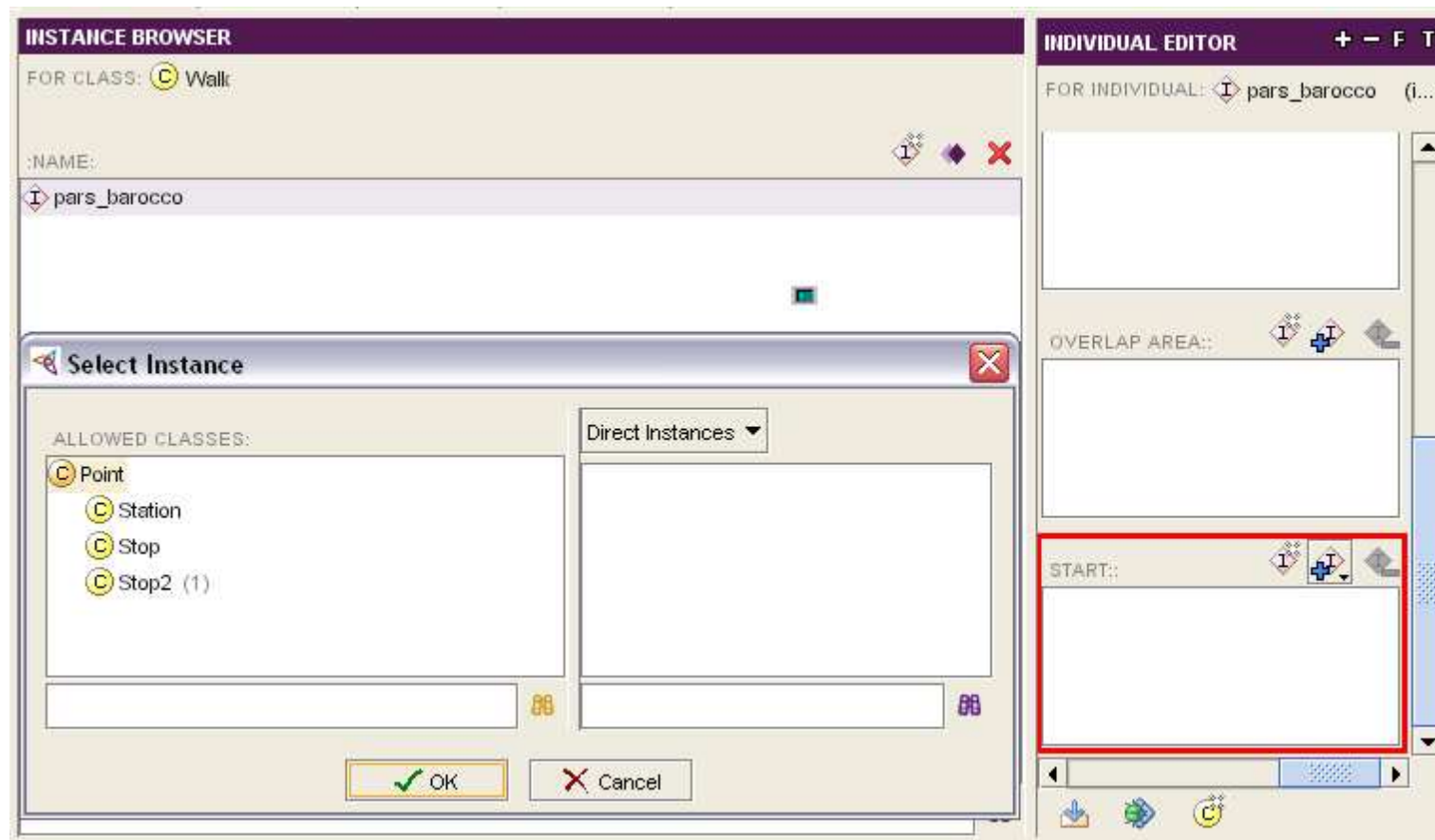
Spatiality : Relationships



Spatiality : Relationships



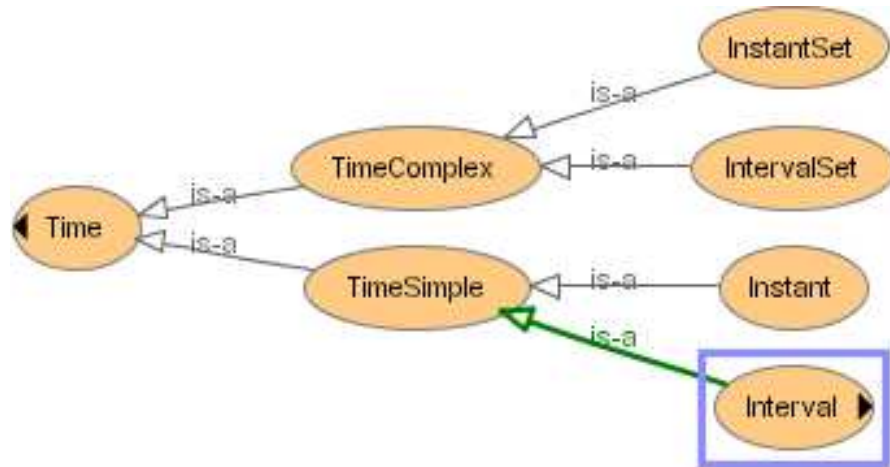
Spatiality : Attributes



Temporality : Types

MADS temporal dimension	OWL temporal dimension
Temporal ADT hierarchy	Temporal abstract types hierarchy
Mandatory LifeCycle attribute that defines spatial features	Intrinsic hasTime property with predefined values

Temporality : Types



Interval (instance of owl:Class)

CLASS EDITOR

For Class: **Interval (instance of owl:Class)**

Asserted Inferred

Asserted Conditions

- hasTime ∃ interval (NECESSARY & SUFFICIENT)
- TimeSimple (NECESSARY)
- ∀ hasTime TSimple [from TimeSimple] (INHERITED)
- ∃ hasTime (TComplex ⊔ TSimple) [from Time] (INHERITED)

Logic View Properties View

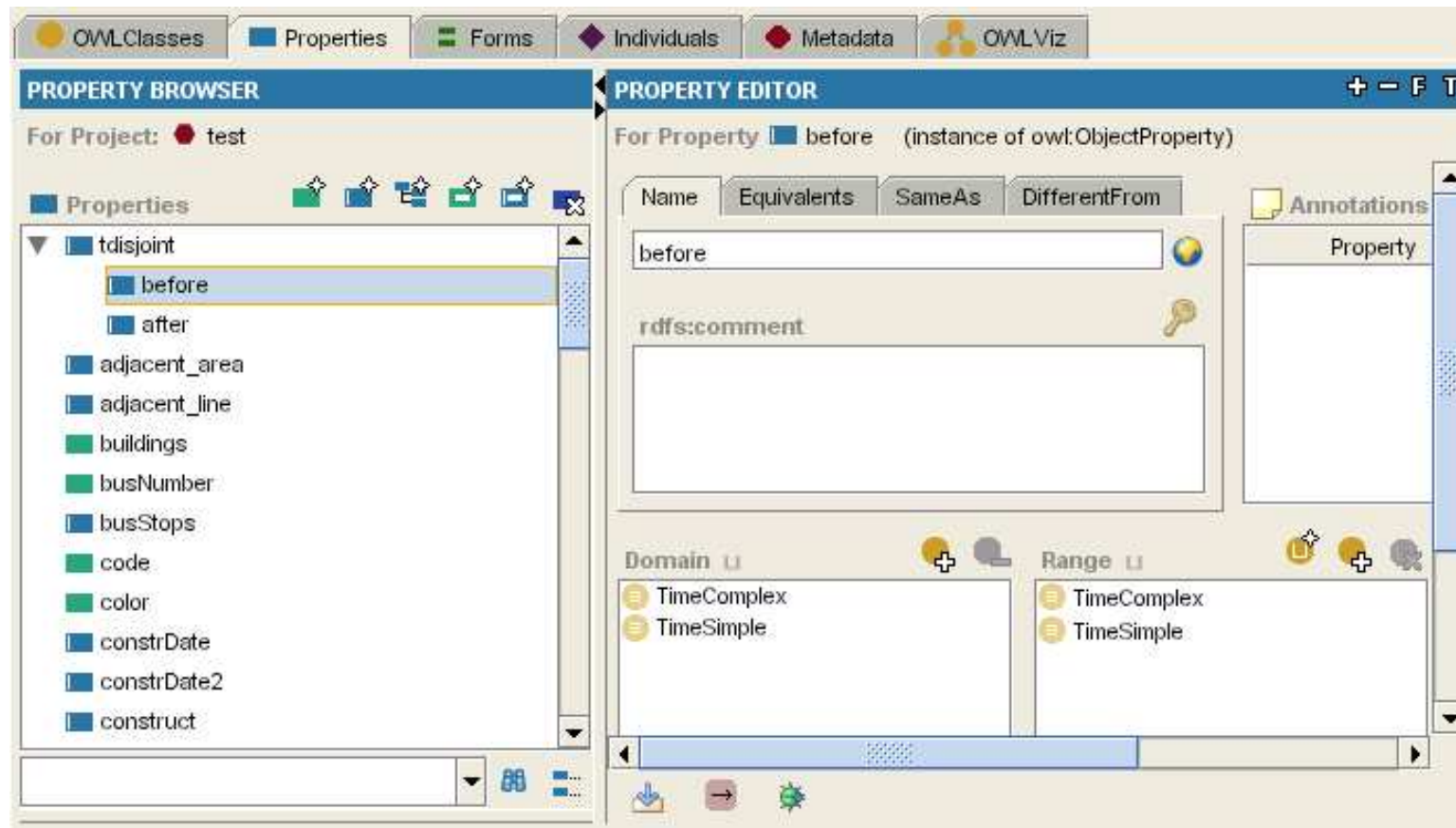
Temporality : Attributes

The screenshot displays two panels from a software application:

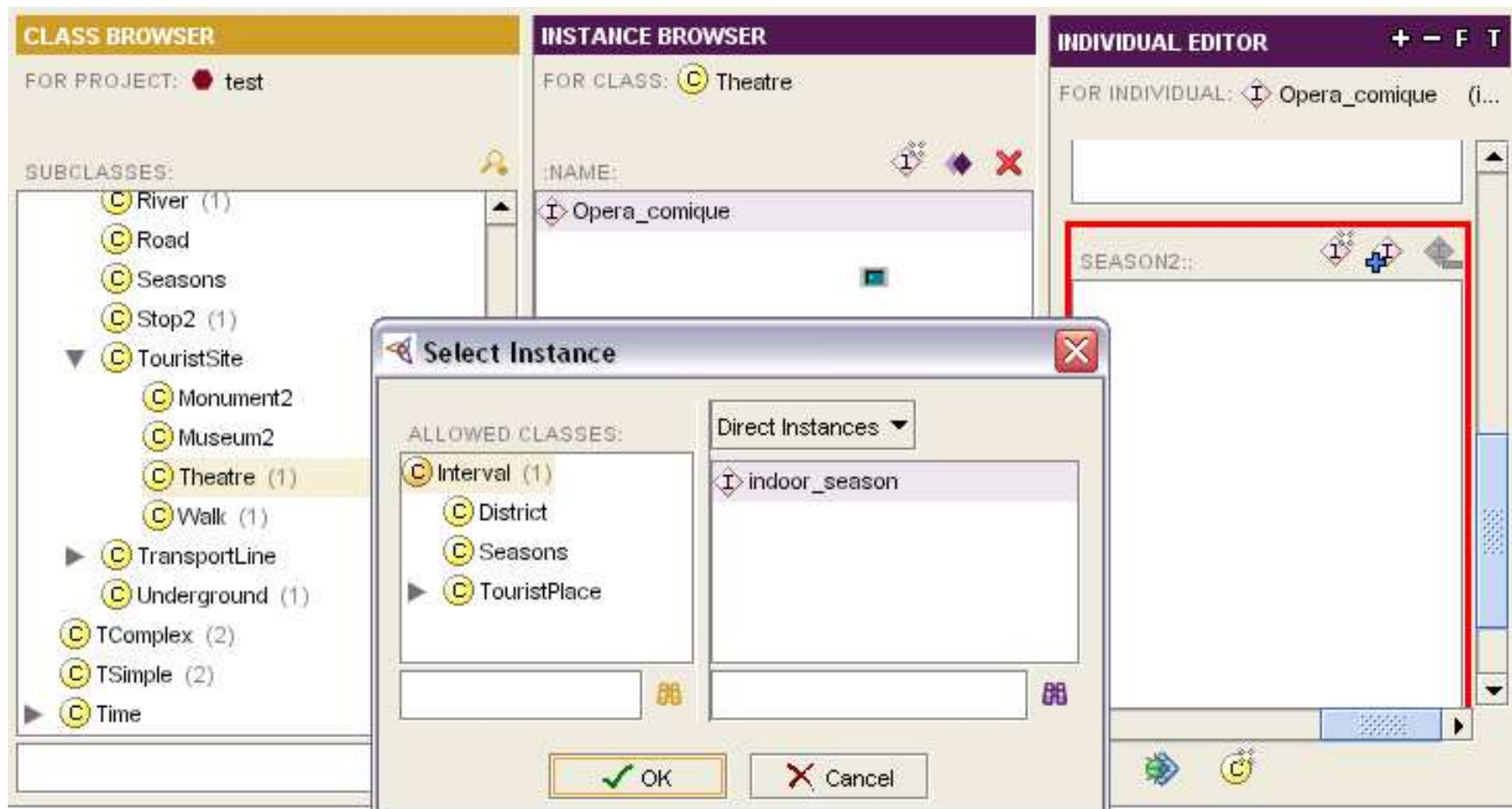
- SUBCLASS RELATIONSHIP:** Shows an asserted hierarchy for project 'test'. The hierarchy is:
 - TouristSite
 - Monument2
 - Museum2 (highlighted)
 - Theatre
 - Walk
 - TransportLine
 - Underground
 - TComplex
 - TSimple
 - Time
- CLASS EDITOR:** Shows the properties for the class 'Museum2' (instance of owl:Class). The properties listed are:
 - description (single String)
 - exhibition (multiple String)
 - openTime (multiple Interval) (minCardinality 1) (highlighted)
 - close (multiple Stop2) (minCardinality 1)
 - disjoint (multiple AreaComplex, LineSet, LineSetOriented, PointSet)
 - equal (someValuesFrom Monument, allValuesFrom AreaSimple, ...)
 - hasGeometry (hasValue areaSimple, allValuesFrom GSimple, ...)

At the bottom right of the CLASS EDITOR, there are radio buttons for 'Logic View' and 'Properties View', with 'Properties View' selected.

Temporality : Relation



Temporality : Instance



Validated

- TouristPlace_{T₁} (Time : Interval) \dashv (no Time) Theatre_{T₂}
- MetroLine_{T₁} (Geometry: Line) $\circ\bullet$ (Geometry : Point) Stop_{T₂}
- TouristPlace_{T₁} (Geometry : Area) \bullet (Geometry : Point) Stop_{T₂}
- Museum_{T₁} (Geometry: Area) $\circ\bullet$ (Geometry : Area) Museum_{T₂}
- Museum_{T₁} (Geometry: Area) \odot (Geometry : Area) Museum_{T₂}

Topological Constraints

The screenshot shows a software interface with several tabs: Metadata, Facet Constraints, OWLViz, PAL Constraints, and Queries. Below these are sub-tabs for OWL Classes, Properties, Forms, and Individuals. A 'Choose Constraints' table is visible, showing two constraints: 'spatial_rule_01' (status: failed) and 'temporal_rule_01' (status: passed). A 'Query Responses' table shows results for '?geo2' (Palais_Royal) and '?geo1' (Gare_de_J_Est).

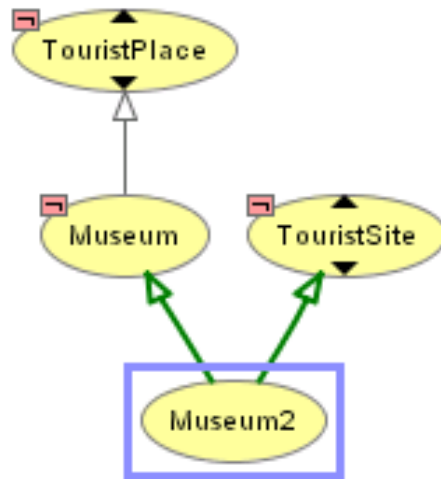
The main focus is the 'INDIVIDUAL EDITOR' for 'spatial_rule_01'. It contains the following information:

- Name:** spatial_rule_01
- Description:** For all the instances of the class Geo, if an instance has a value (of class Geo) for the topological property "s_disjoint", then this instance must not have any other topological property (adjacent_area, adjacent_line, cross_area, cross_line, cross_point, include_area, include_line, overlap_area, overlap_line, overlap_point, s_equal) with the same value.
- Statement:**

```
(forall ?geo1 (forall ?geo2
  (=> (s_disjoint ?geo1 ?geo2)
    (and (not (s_equal ?geo1 ?geo2))
      (not (adjacent_area ?geo1 ?geo2))
      (not (adjacent_line ?geo1 ?geo2))
      (not (cross_area ?geo1 ?geo2))
      (not (cross_line ?geo1 ?geo2))
      (not (cross_point ?geo1 ?geo2))
      (not (include_area ?geo1 ?geo2))
      (not (include_line ?geo1 ?geo2))
      (not (overlap_area ?geo1 ?geo2))
      (not (overlap_line ?geo1 ?geo2))
      (not (overlap_point ?geo1 ?geo2))
      (not (s_equal ?geo1 ?geo2))
```
- Range:**

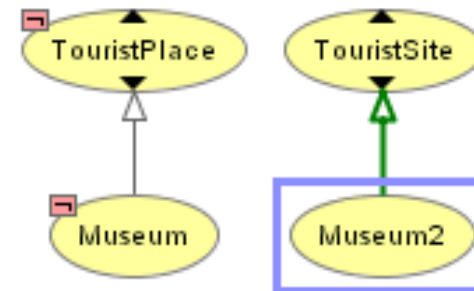
```
(defrange ?geo1 :FRAME Geo)
(defrange ?geo2 :FRAME Geo s_disjoint)
```

Structure Reasoning



Reasoner log

- Checking consistency of Museum
- Time to query reasoner = 0.42 seconds
- **✗ Museum is inconsistent**
- Total time: 0.49 seconds



Conclusions

- we adhere to a hybrid approach - DB + DL
- we model the semantics of the MADS data model required for mappings validation
- we "emulate" spatio-temporal reasoning for inter-schema mappings
- we shift the emphasis on automation from the a priori discovery to the a posteriori validation of the inter-schema mappings