Transforming conceptual spatiotemporal model into Object model with semantic keeping

Chamsedine Zaki, Myriam Servières and Guillaume Moreau

École Centrale Nantes
Nantes, France

Plan

- Context
- Urban database conception
- Implementation of MADS in Java (Object Environment)
- Conclusion

Context

General Context → Data Modeling

Real World

↓

Digital world

Modelisation

Conceptual scheme

- Intermediate step between the observed reality and the computer
- Formal framework mapping the content of information

(Logical scheme...) →

Physical scheme

DBMS
GIS

depends on the chosen DBMS: Oracle, SQL Server,...

Context

Objectif :

Present a transformation rules of a conceptual model into an object model that keeps (as close as possible) its semantics in the context of urban data modeling

Urban data

- Describe geolocated urban elements

- Subset of Spatiotemporal data

_Urban Database Conception: Requirements

Requirements
Spatial types hierarchy
Temporal types hierarchy
Orthogonality between concepts
Orthogonality between dimensions
Spatial and temporal constraints
Multirepresentation
Events modelisation
3D and uncertain data
Conception tools readability

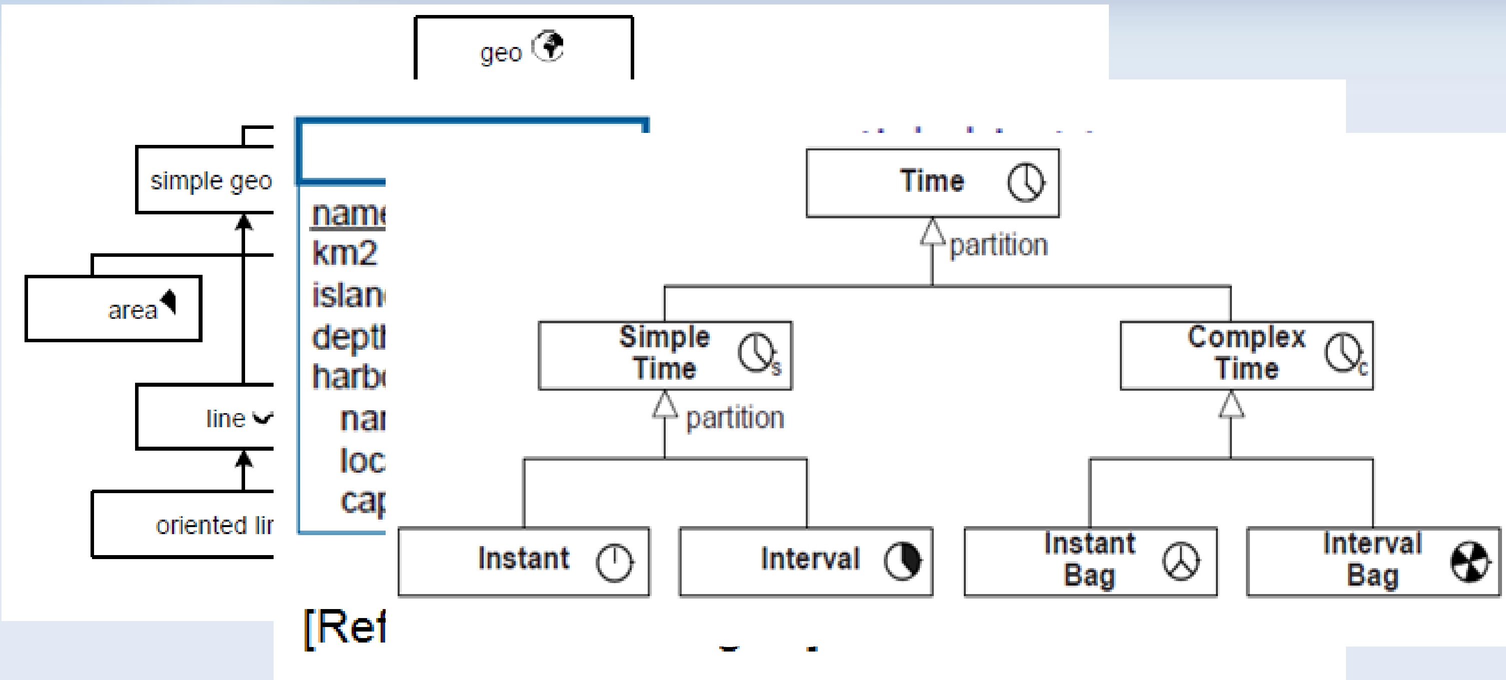
(Parent, et al., 2009) (Miralles, 2005) (Beard, 2006) (Pelekis, et al., 2004) (Pinet, 2010)

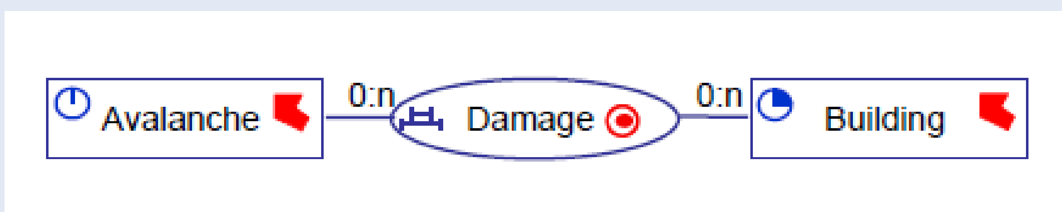
_Urban Database Conception : Existing models

Requirements	STER	STUML	PERCEPTORY	MADS
Spatial types hierarchy	**No**	**No**	**No**	**Yes**
Temporal types hierarchy	**No**	**No**	**No**	**Yes**
Orthogonality between concepts	**No**	**Yes**	**No**	**Yes**
Orthogonality between dimensions	**Yes**	**Yes**	**Yes**	**Yes**
Spatial and temporal constraints	**No**	**No**	**No**	**Yes**
Multirepresentation	**No**	**No**	**No**	**Yes**
Events modelisation	**No**	**No**	**No**	**No**
3D and uncertain data	**No**	**Yes**	**Yes**	**No**
Conception tools readability	+	++	+++	++++
	(Tryfona, et al., 1999)	(Price, et al., 2000)	(Bedard, 1999)	(Parent, et al., 1997)

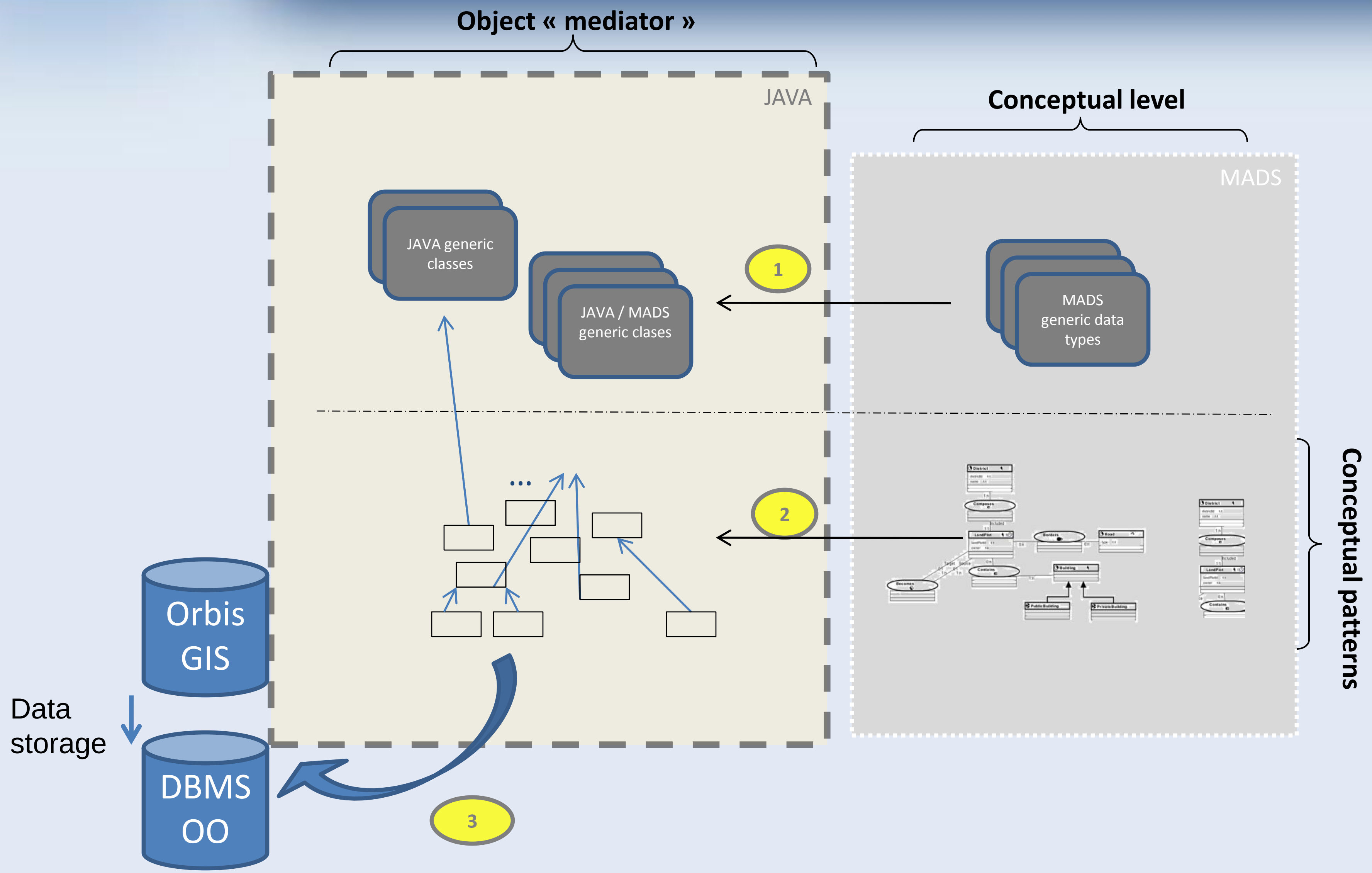
MADS (Modeling of Application Data with Spatio-temporal features)

_Urban Database Conception : MADS





Implementation of MADS in Java



Implementation of MADS in Java : Rules for ST transformation

Structural, spatial and temporal dimension transformation			
Class	Binary association: role1(X,1), role2 (X,1) / X= 0 ou X=1	Spatial classes	Temporal classes
Attribute(1,1)	Binary association: role1 (0,n), role2(0, n)	Spatial associations	temporal associations
Attribute(0,n)	Binary association: role1 (ind ,X), role2 (ind, ind)	Topological Constraints of relations	Timing constraints relation
Attribute(0,1)	n-ary associations	Spatial attribute	“role” cardinality temporal constraint
Attribute(1,n) Attribute(1,n) and explicit « n » Attribute(0,n) and explicit « n »	Reflexive association	Variable attribute in space	temporal attribute
Complex Attribute (1,1)	Multi-Association		Variable concepts in time
Complex Attribute (1,n)	Generalization		Multi-representation
Complex Attribute (0,1)	Aggregation		
Complex Attribute (0,n)	Generation, Transition		
Attribute: enumeration	Occurrent - Occurrent		
Key	Occurrent - Continuant		
Method	Method		

Implementation of MADS in Java : Rules for ST transformation

Structural, spatial and temporal dimension transformation			
Class	Binary association: role1(X,1), role2 (X,1) / X= 0 ou X=1	Spatial classes	Temporal classes
Attribute(1,1)	Binary association: role1 (0,n), role2(0, n)	Spatial associations	temporal associations
Attribute(0,n)	Binary association: role1 (ind ,X), role2 (ind, ind)	Topological Constraints of relations	Timing constraints relation
Attribute(0,1)	n-ary associations	Spatial attribute	“role” cardinality temporal constraint
Attribute(1,n) Attribute(1,n) and explicit « n » Attribute(0,n) and explicit « n »	Reflexive association	Variable attribute in space	temporal attribute
Complex Attribute (1,1)	Multi-Association		Variable concepts in time
Complex Attribute (1,n)	Generalization		Multi-representation
Complex Attribute (0,1)	Aggregation		
Complex Attribute (0,n)	Generation, Transition		
Attribute: enumeration	Occurrent - Occurrent		
Key	Occurrent - Continuant		
Method	Method		

Implementation of MADS in Java

Transformation of general concepts : classes - attributes

Building		
number	Integer	1:1
fireplace	Fireplace	1:1
roofing		1:1
form	String	1:1
surface	Integer	1:1
tent	Integer	1:1
owner	String	1:n
otherName	String	0:1

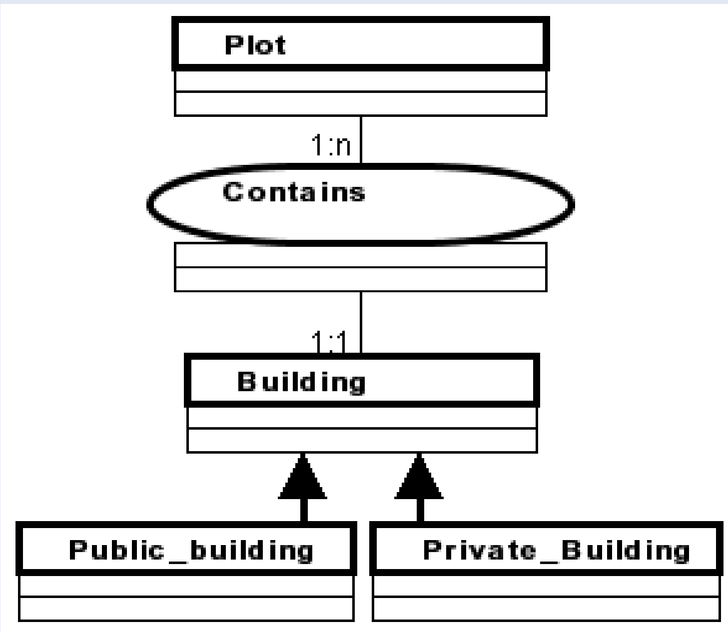
```
class Building { // constructor...
    Integer number;
    Fireplace fireplace; // method set (), get ()...

    Roofing roofing;
    class Roofing {
            String form ;
            Integer surface ;
            Integer tent;   }

List<String> owner = new ArrayList <String>();
...}
```

Implementation of MADS in Java

Transformation of general concepts : relations between classes




```
public class Contains {
     Plot plot ;
     Building Building;
...}

class Plot {
      List <Contains> contient = new ArrayList < Contains >();
 ...}

public class Building {
     Contains estContenu ;
 ...}

class Public_builging extends Building {
...}
```

Implementation of MADS in Java

- Transformation of spatial dimension

MADS Type	JTS Class
Geo	**Geometry**
SimpleGeo	
ComplexeGeo	
PointBag	
LineBag	**MultiLineString**
OrientedLineBag	***MultiLineString***


```
import com.Vividsolutions.jts.geom.Polygon;

class Building {
    Polygon  spatialFeature;

    Polygon parking;

    Lighting lighting ;
    class Lighting  {
        private Map <Polygon, Integer> value ;
        // methods set() et get ()
    ...}
 ... }
```

Implementation of MADS in Java

Transformation of temporal dimension

MADS Type	
Time	
Time	
nt	
val	
xTime	
InstantBag	
IntervalBag	IntervalBag
TimeSpan	TimeSpan


```
class Building {

    List <lifecycle> lifecycle = new ArrayList <Lifecycle> ;

    class LifeCycle{
        String status;
        Interval time;
        // verification methods
    ...    }

... }
```

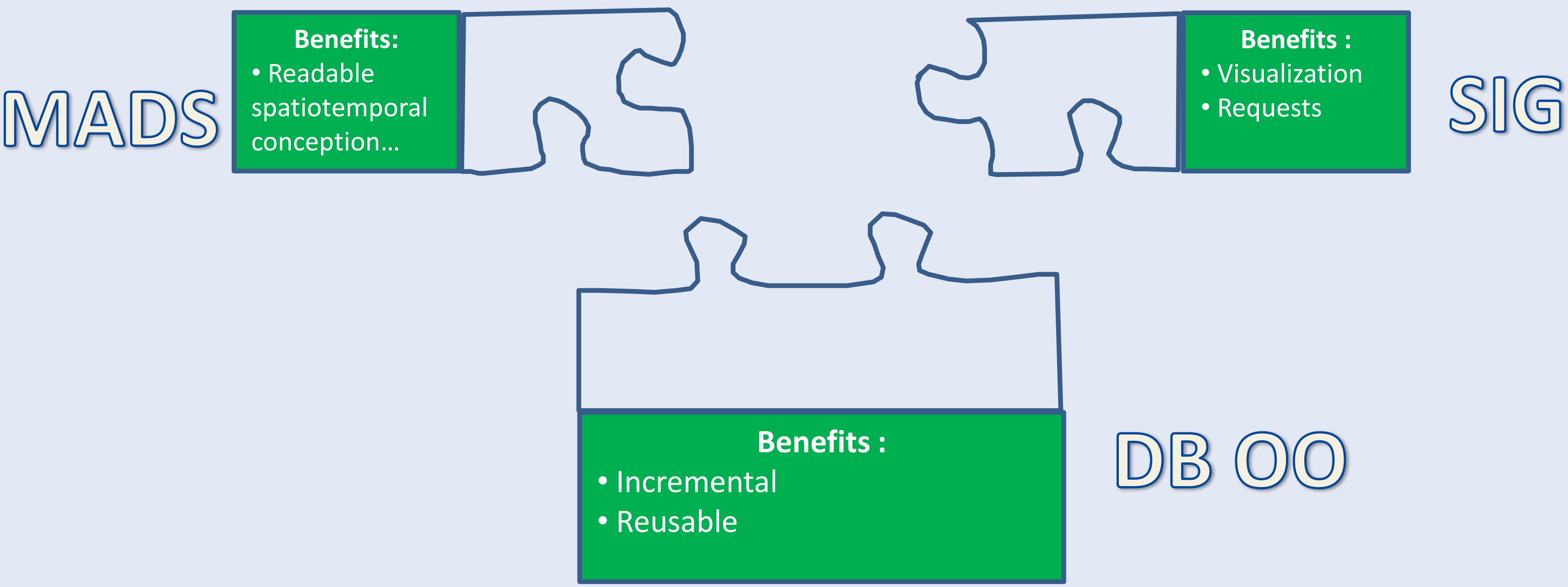
Conclusion

MADS was implemented in a flexible and incremental platform.

Translating MADS into object model permits

→ preservation of the conceptual models semantics

→ direct access and storage of any type of data



Transforming conceptual spatiotemporal model into Object model with semantic keeping

Chamsedine Zaki, Myriam Servières and Guillaume Moreau

École Centrale Nantes
Nantes, France