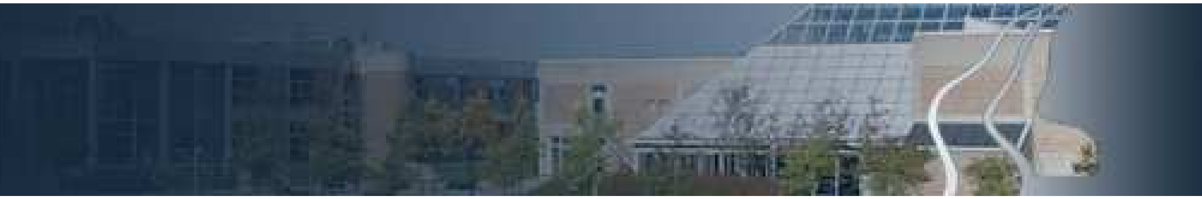


Boosting Location-Based Services with a Moving Object Database Engine

Nikos Pelekis, Yannis Theodoridis
June 25, 2006,

Dept of Informatics, Univ. of Piraeus, Hellas
{npelekis, ytheod}@unipi.gr

MobiDE'06, June 25, 2006, Chicago, Illinois, USA.

Presented by: Thomas Winterberg

Outline

1) Motivation
2) Types
3) Moving types
   a) Idea
   b) Operations on moving objects
   c) Interation with temporal and spatial domains
   d) Set Relationship

4) Hermes LBS(Location Based Service) tool

5) Related work

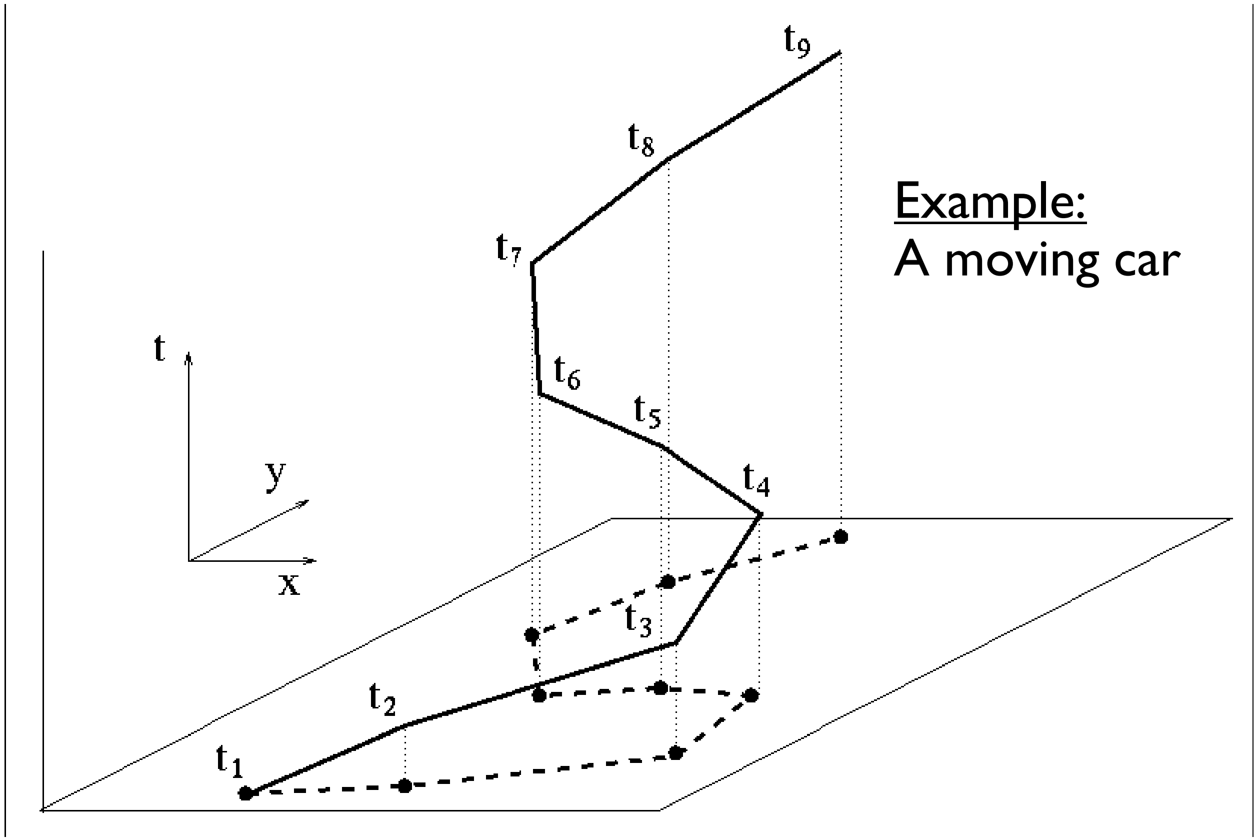
6) Evaluation

1) Motivation

- Develop a plugin to an Object Relational DBMBS(ORDBMS).
- The plugin named Hermes-MDC(Moving Data Catridge) should provide temporal functionality to the database (spatio-temporal functionality).
- Hermes-MDC supports modelling and querying of moving objects(Types). Objects that change location, shape and size either discretly or continously in time.

2) Types

The movement an object in time and the corresponding trajectory



Example:
A moving car

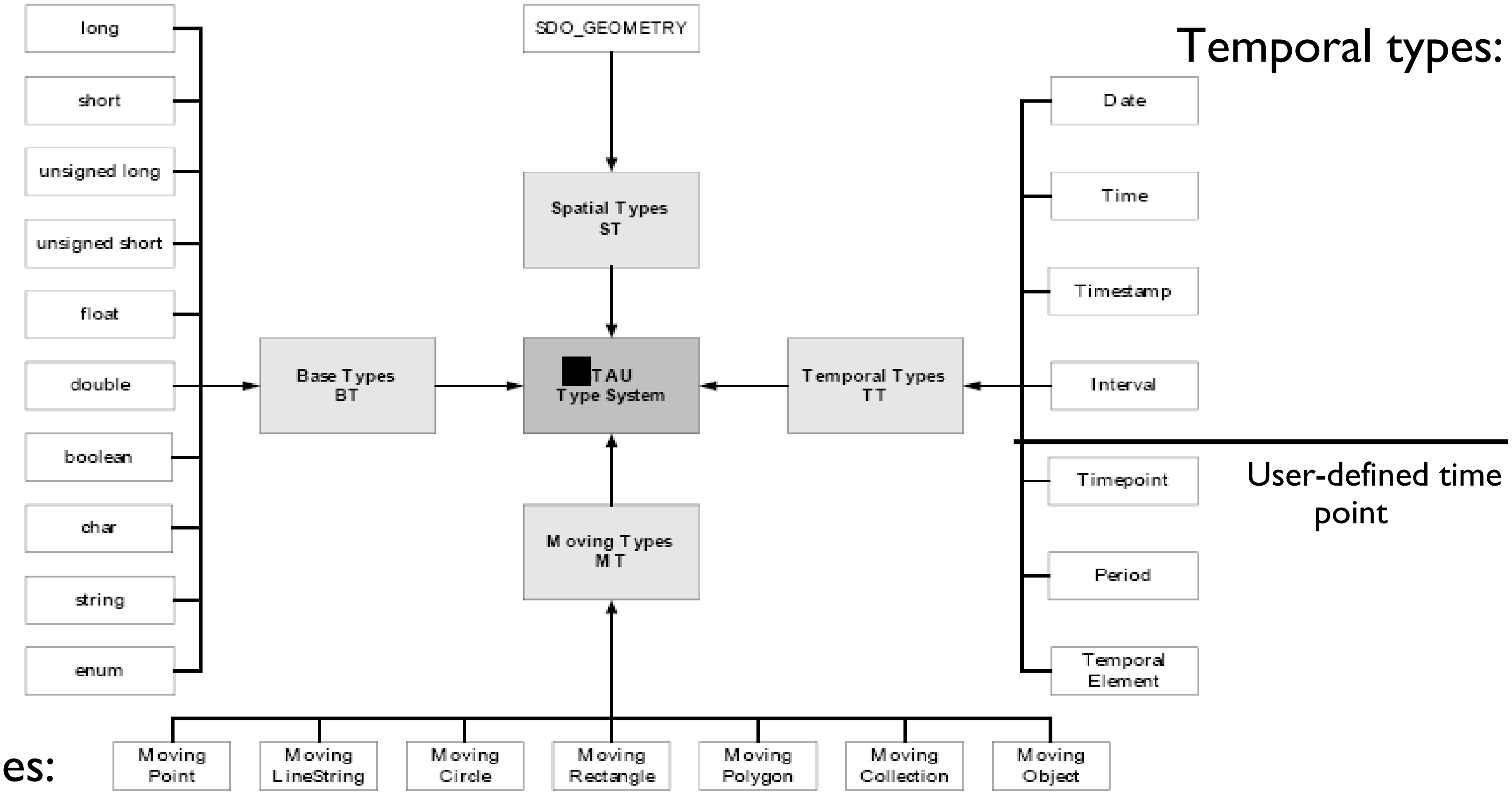
2) Types

System overview

Spatial types:

Base types:

Temporal types:



long
short
unsigned long
unsigned short
float
double
boolean
char
string
enum

SDO_GEOMETRY

Spatial Types
ST

Base Types
BT

TAU
Type System

Temporal Types
TT

Moving Types
MT

Date
Time
Timestamp
Interval
Timepoint
Period
Temporal Element

User-defined time point

Moving types:

Moving Point
Moving LineString
Moving Circle
Moving Rectangle
Moving Polygon
Moving Collection
Moving Object

3.a) Moving types

Basic idea of moving types

- Decompose temoporal development of a moving type into slides.
- Continuously → Discrete. Reduce number of points to be saved
- More easy to accomodate discrete changing objects in databases

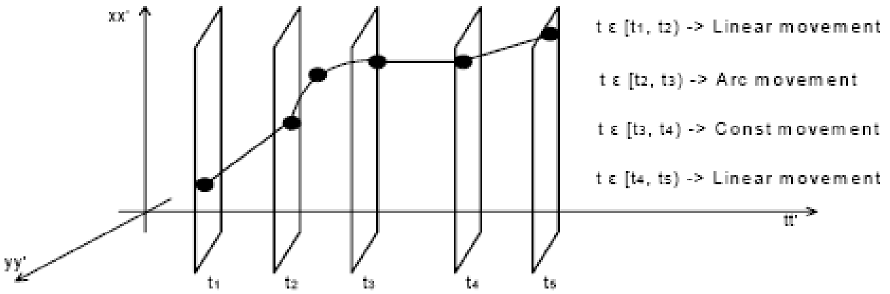
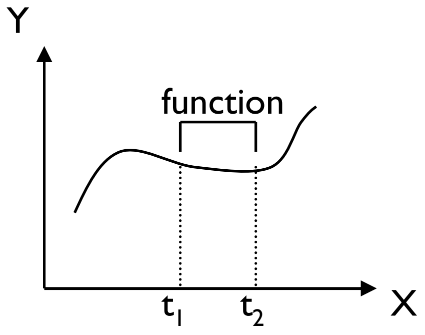

xx'
yy'
tt'
t1
t2
t3
t4
t5
t ε [t1, t2) -> Linear movement
t ε [t2, t3) -> Arc movement
t ε [t3, t4) -> Const movement
t ε [t4, t5) -> Linear movement

Figure 2 Moving Point with various types of movement

- Decompose definition of each moving type into sub-definition(=unit moving type) and compose these definitions as a collection to define the moving type.

3.a) Moving types

Basic idea of moving types



Definition 1: *Unit_Function =*
$_d\langle$ *x_i:double, y_i:double, x_e:double, y_e:double, x_c:double, y_c:double, v:double, a:double, flag:TypeOfFunction*$\rangle$, where
Π TypeOfFunction T={ CONST, PLNML_1, ARC_<1..8> }

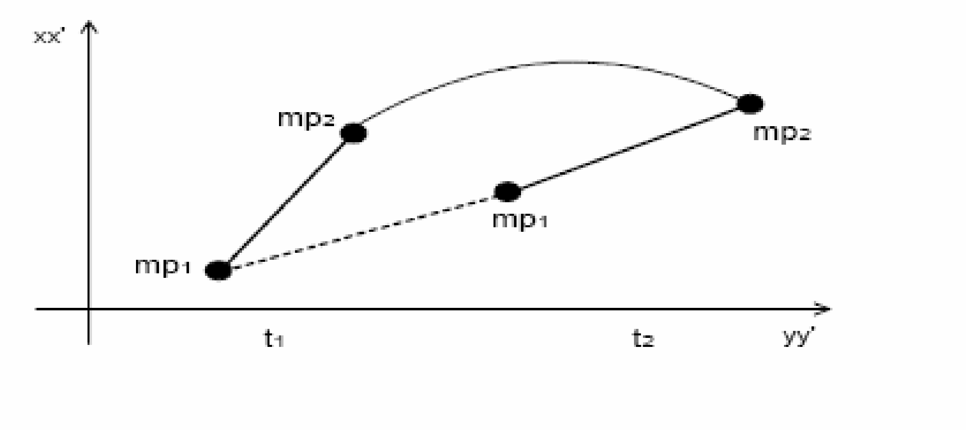
(x_i,y_i) → (x_e,y_e)
(X_c,Y_c) = centre of circle
v = velocity, a = acceleration

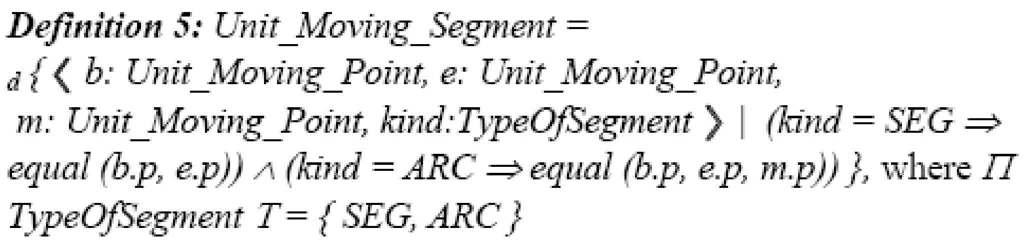
Definition 2: *Unit_Moving_Point =*
$_d\langle$*p: Period(SEC), m: Unit_Function*$\rangle$

Unit_Moving_Point + associate period of time
e.g [b,e) where b is beginning and e is ending point

3.a) Moving types

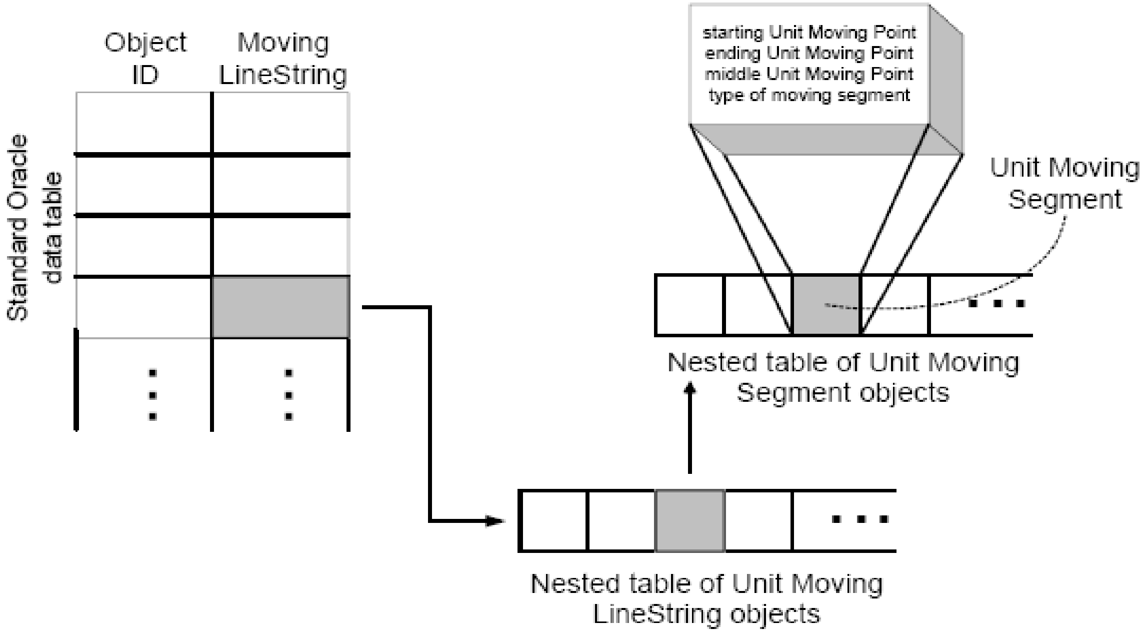
Unit_Moving_Segment (change shape/form)




Definition 5: *Unit_Moving_Segment =*
d { ⟨ b: Unit_Moving_Point, e: Unit_Moving_Point,
m: Unit_Moving_Point, kind:TypeOfSegment ⟩ | (kind = SEG ⇒
equal (b.p, e.p)) ∧ (kind = ARC ⇒ equal (b.p, e.p, m.p)) }, where *Π*
TypeOfSegment T = { SEG, ARC }

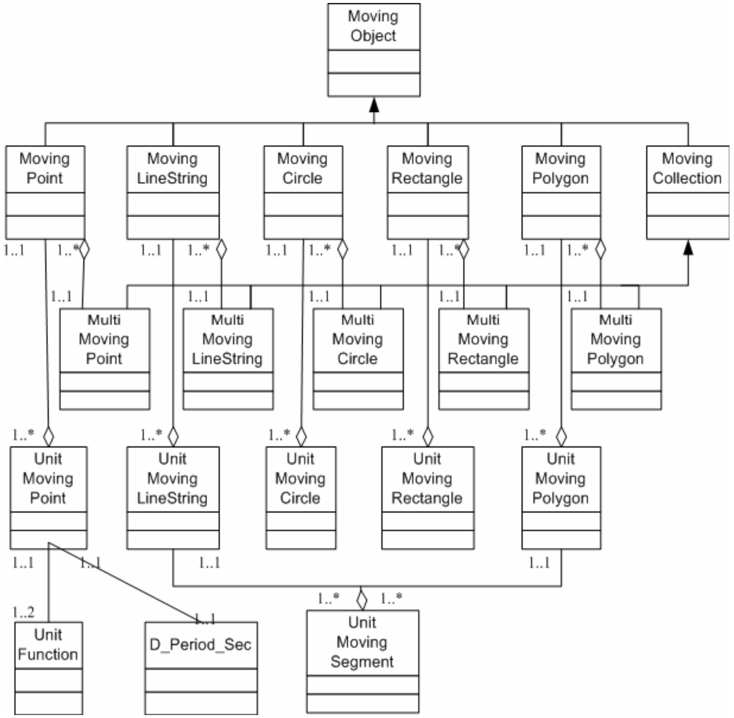
3.a) Moving types

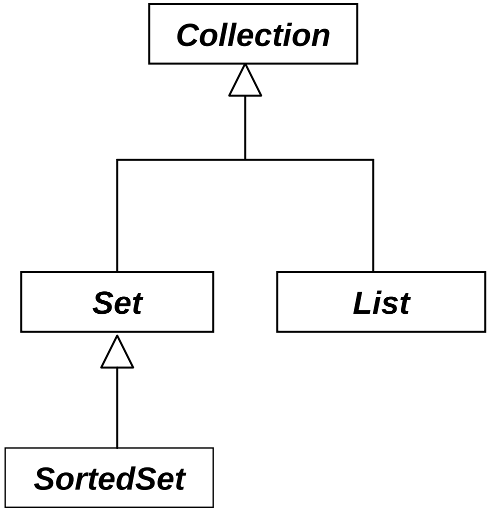
Structure of the Moving_LineString Object


Object ID
Moving LineString
Standard Oracle data table
starting Unit Moving Point
ending Unit Moving Point
middle Unit Moving Point
type of moving segment
Unit Moving Segment
Nested table of Unit Moving Segment objects
Nested table of Unit Moving LineString objects

3.a) Moving types

UML: Hermes-MDC architecture and Java Collection





3.b) Moving types

Operations on moving objects

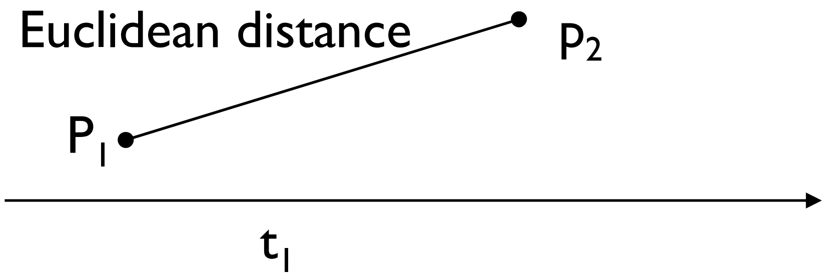
2 Types of operations:

- Time dependant: User-defined time point/at given instant.
- Time independant: % Timepoint<SEC>. Models sequence of time intervals that 2 objects are within. Return type is Moving_Object

3.b) Moving types

Operations on moving objects

<u>Examples distance:</u>



- Boolean {Moving_Object} **within_distance**(distance, Moving_Point, tolerance, Timepoint<SEC>)

- Determines if two moving objects are within some specified distance from each other at a user-defined time point.

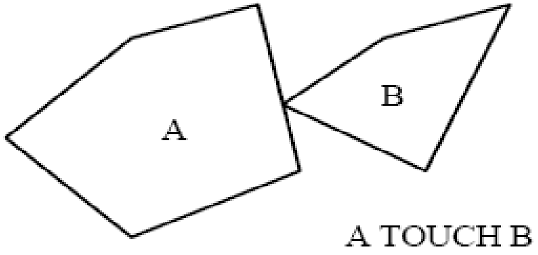
- WHERE kind =’test’ AND truck245.within_distance(50000,location,now)

3.b) Moving types

Operations on moving objects

Examples topological relationships:

mask:*ANYINTERACT* **-**
Returns *TRUE* if
the objects are not disjoint.

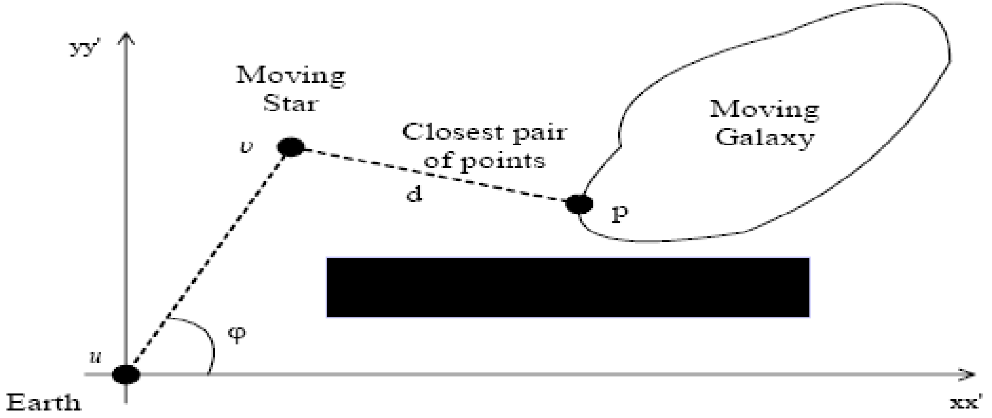


- Varchar2{Moving_Object}**relate** (mask, Moving_Polygon, tolerance, Timepoint<SEC>

3.b) Moving types

Operations on moving objects

<u>Examples distance:</u>



- Number {Moving_Object} **distance**(Moving_Point, tolerance, Timepoint<SEC>)

3.c) Interation with temporal and spatial domains

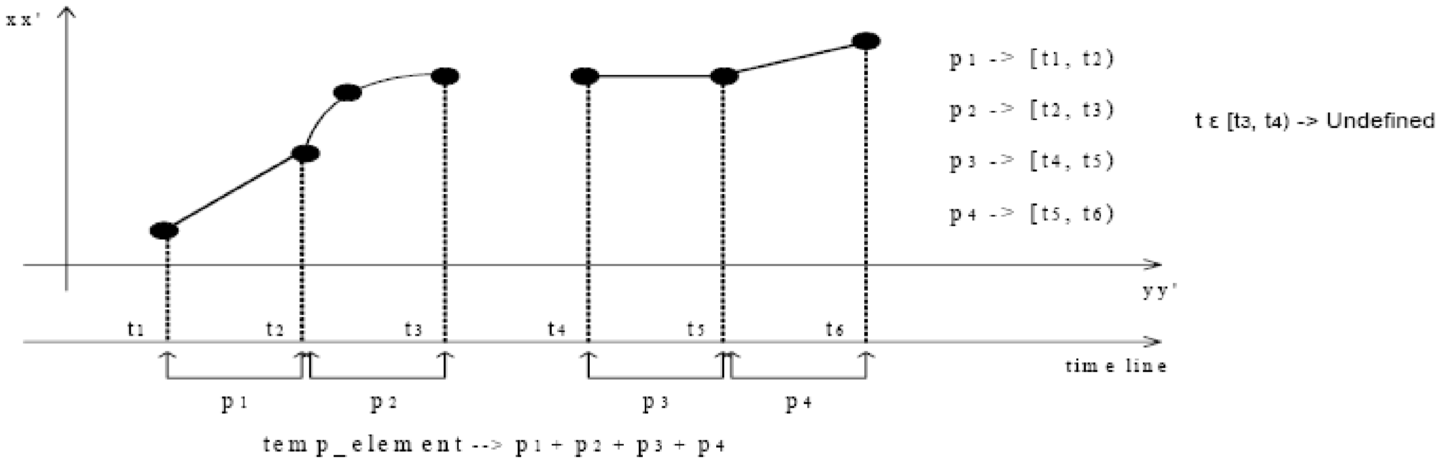
<u>Restriction:</u>

- Moving_Point at_periode(Period<SEC>)
  Delimit time period that is meaningful to ask the projetion of the moving object.

<u>Find object:</u>

- Moving_Point at_Linestring(SDO Geometry)
  An object moves on a linestring geometry during a route. Find the position of the object.

3.c) Interation with temporal and spatial domains

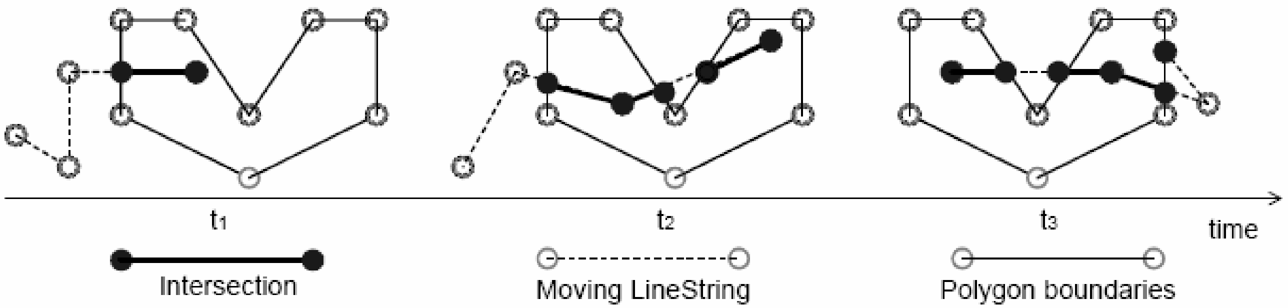


$$temporalElement\langle g\rangle =_d \{ te\!: set\langle period\langle g\rangle\rangle \mid \forall i, j \cdot i \neq j \Rightarrow te_i \cap te_j = \emptyset \}$$

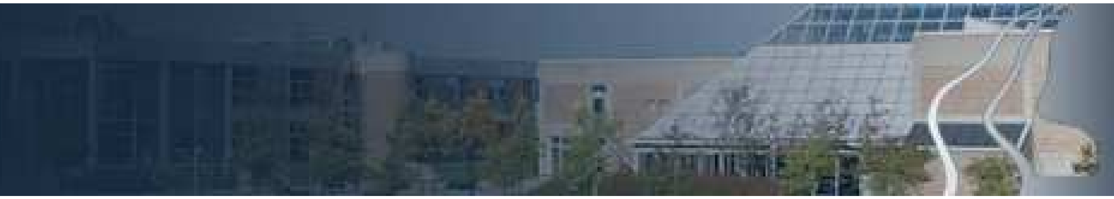
Temp Element:

- Temp_Element<SEC>temp_element()
- Project time periods that form unit moving objects. Concatenate these periods to a temp element.

3.d) Set Relationship



- Geometry{Moving_Point} **intersection(**Geometry, tolerance, Timepont<SEC>).
- Possible to define entering/leaving locations.
- Can be used to check whether a car (moving point) is intersecting with an area containing heavy rain (geometry).



4) Hermes LBS tool / experiental results

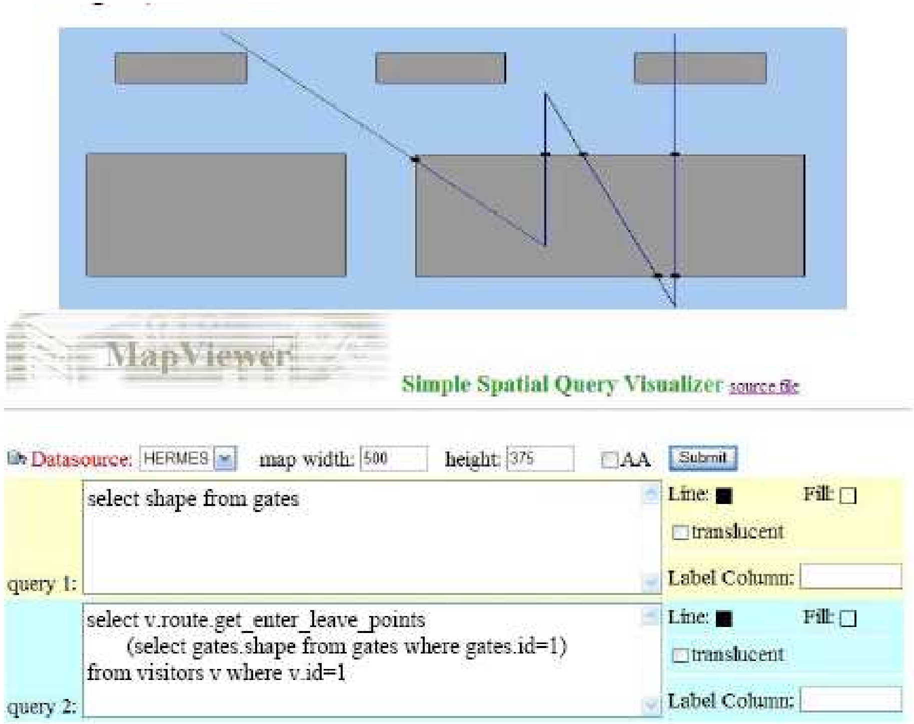
- System extension that provides spatio-temporal functionality to Oracle10g ORDBMS.

- Extends PL/SQL DML and DDL of Oracle 10g → result → query language.

- Developed a prototype application for travelers entering the area of an airport.
  spatial = ground plan of airport, random trajectories of travelers moving around the area.

4) Hermes LBS tool / experiental results

- The LBS tool



Datasource: HERMES map width: 500 height: 375 □AA

query 1:

```
select shape from gates
```

Line: ■ Fill: □
□translucent
Label Column:

query 2:

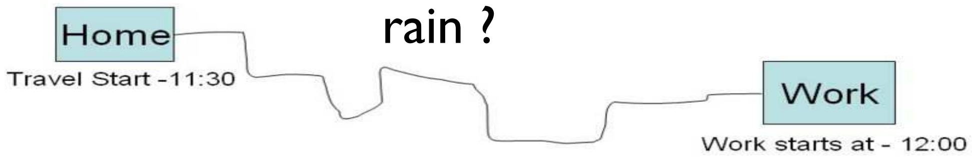
```
select v.route.get_enter_leave_points
      (select gates.shape from gates where gates.id=1)
from visitors v where v.id=1
```

Line: ■ Fill: □
□translucent
Label Column:

Figure 7 Visualization of enter/leave points in an area of interest

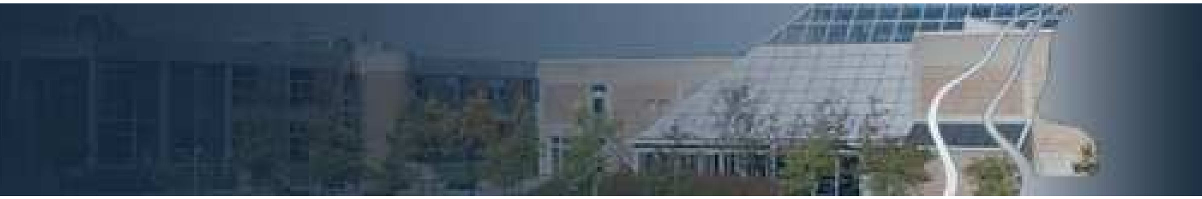
5) Related work – Dat5 project

Weather forecasting on a route :



- Weather forecasting on a route
- We receive a weather forecast every 5 minutes
- A user wants to know whether there is a risk of rain on her path

- The user is a moving object
- The rainy weather is a geometry
- When the 2 objects intersect there is rain on the path



6) Evaluation

Good points

- Can be implemented in the real world.
- Data catrigde can be a system extension that provides spatio-temporal functionality to e.g. Oracle10g.
- Clear idea of the paper

6) Evaluation

Could be improved

- The term string on page 4 paragraph 1 could be more clarified e.g. by an illustration.
- The definitions 8 and 9 on page 5 in section 2.2 could have a more obvious description (e.g what is a ulong ?).
- The mask relationships on page 7 line 24 could be exemplifed e.g. by *ANYINTERACT*.
- The figure 5 on page 7 could have an explanation for why there is a gap
- Summarize the article is short. It could be longer with more clarified definitions.