Exploring Spatiotemporal Patterns By IntegratingVisual Analytics With a Moving Objects Database System

Mahmoud A. Sakr, Thomas Behr, Ralf Hartmut Güting,
Gennady Andrienko, Natalia Andrienko, Christophe Hurter

Fernuniversität Hagen, Germany
Fraunhofer IAIS, Sankt Augustin, Germany
ENAC, Toulouse, France

Spatio-Temporal Pattern Predicate

Air Traffic Control Application

- Data set with 17238 flights over France and neighboring countries on one day, Feb 22, 2008.
- Radar recordings (Lon, Lat, Alt) of aircraft positions.
- Sampling interval about one to three minutes.

Some Problems:

Missed Approach: while aircraft is close to landing, pilot decides to climb again.

Stepwise Descent Landing: landing pattern where aircraft alternates between descent and cruise.

Holding Pattern: aircraft flies loops before the final approach to the airport.

Spatio-Temporal Pattern Predicate

Mahmoud Sakr & Ralf Hartmut Güting, *GeoInformatica* 2011

The goal:

- Formulate conditions on moving objects that may be fulfilled at certain time intervals.
- Specify relationships between these fulfillment time intervals.

Predicates may be, for example:

- Speed is higher than 120 km/h
- Car drives through forest
- Extent of storm area is larger than 4 square kms
- Altitude of air plane is less than 500 ms

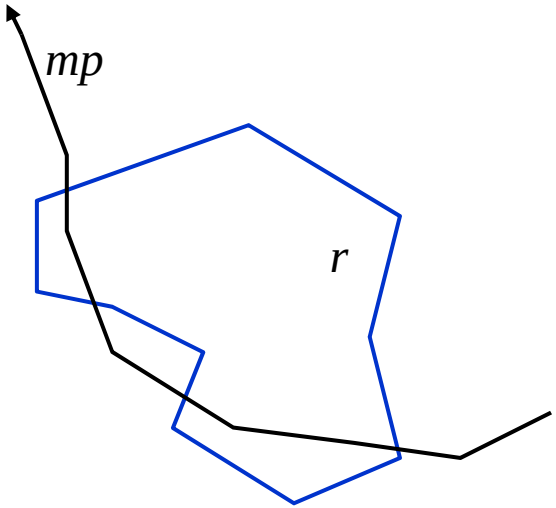
Time interval relationships:

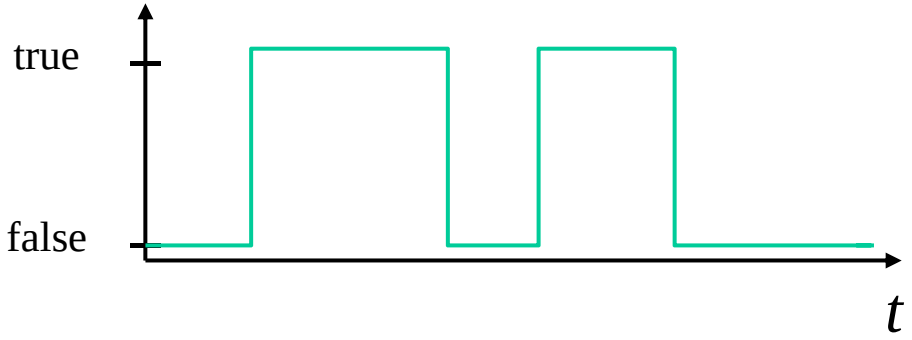
- A then B then C.
- A before B; A and B during C

Spatio-Temporal Pattern Predicate

Approach:

- Use the lifted predicates of the moving objects data model. They return time dependent boolean values (*mbool*).
- Define a notation for possible time interval relationships.
- Specify required time interval relationships between the true periods of *mbool*s.

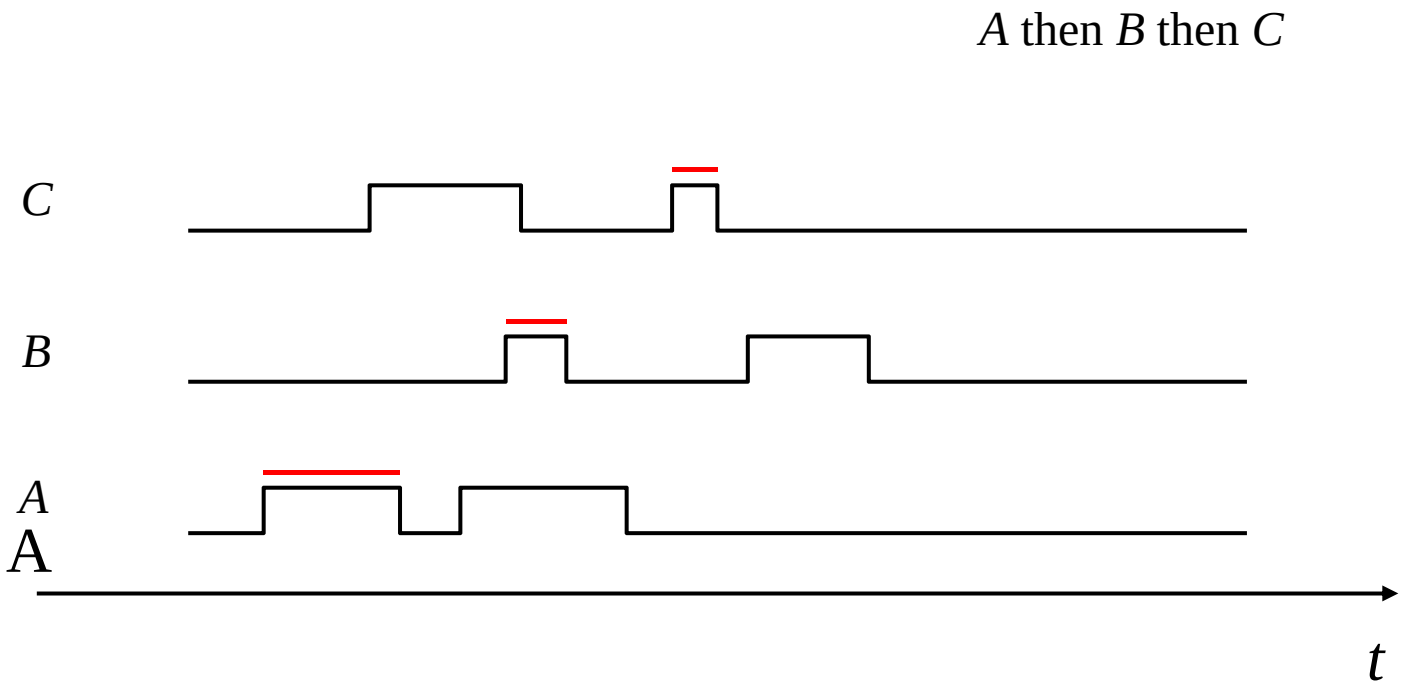




mp **inside** *r*

Spatio-Temporal Pattern Predicate

Several *mbool* values:

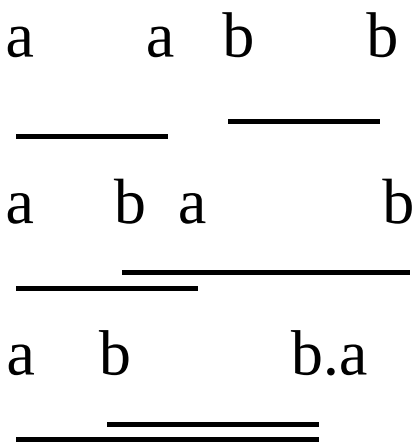


What is the precise meaning of *then*?

Spatio-Temporal Pattern Predicate

A notation for time interval relationships

- A definition by Allen (1983) of 13 relationships exists. But needs to be learned and remembered. Also we have 26 relationships as intervals may degenerate into points.
- Graphical notation: aabb, abab, abb.a



- vec(abba, abb.a) means disjunction of relationships.
- May define names such as

```
let later = vec(aabb)
Let during = vec(abba, a.bba, abb.a)
```

Spatio-Temporal Pattern Predicate

Implementation in Secondo

- **stpattern** operator. Takes a tuple, a list of labeled predicates, and a list of constraints and returns a *bool*
- Example: vehicle passes first through region A and later through point B

```
.. stpattern[
InA: .Pos inside A,
AtB: .Pos = B;
stconstraint("InA", "AtB", vec("aabb")) ]
```

- **stpatternex** operator. A variant that takes a further predicate that allows one to formulate further conditions on lengths of time intervals etc.
- Example: vehicle passes first through region A and *at most one hour later* through B

```
.. stpattern[
InA: .Pos inside A,
AtB: .Pos = B;
stconstraint("InA", "AtB", vec("aabb"));
start("AtB") - end("InA") < onehour ]
```

Spatio-Temporal Pattern Predicate

Further Variants

- **stpatternextend**
  - Process a tuple stream. For each tuple fulfilling the STP predicate, add the time intervals of the first supported assignment to the result stream.
- **stpatternextendstream**
  - Same as before, but report all supported assignments, making copies of the input tuple.
- **stpatternexextend, stpatternexextendstream**
  - As above, but for **stpatternex** operator

Spatio-Temporal Pattern Predicate

```
let MissedApproach = Flights feed

# reduce flights to the last twenty minutes
  extend[Last20:
    theRange(inst(final(.Pos)) - TwentyMin, inst(final(.Pos)), TRUE, TRUE)]
  projectextend[Id
  ; Pos: .Pos atperiods .Last20,
    Alt: .Alt atperiods .Last20]

# add destination (final position) and altitude derivative
  extend[
    Destination: val(final(.Pos)),
    Der: derivative(.Alt)]
  ...
```

```
...
# st pattern predicate
  stpatternexextend[
    Close: distance(gk(.Pos), gk(.Destination)) < 15000.0,
    Down1: ((.Der < 0.0) and (.Alt < 600.0)),
    Up: .Der > 0.0,
    Down2: .Der < 0.0

  ; stconstraint("Close", "Down1", vec("abba", "a.bba", "baba")),
    stconstraint("Close", "Up", vec("abba", "aba.b", "abab")),
    stconstraint("Down1", "Up", vec("aabb", "aa.bb")),
    stconstraint("Up", "Down2", vec("aabb", "aa.bb"))

  ; ((start("Up") - end("Down1")) < create_duration(0, 120000)) and
    is_within_total_turn_range(.Pos atperiods theRange(start("Close"),
      inst(final(.Pos)), TRUE, FALSE), 0.0, 400.0) and
    is_within_total_turn_range(.Pos atperiods theRange(end("Up"),
      inst(final(.Pos)), TRUE, FALSE), 0.0, 270.0)]

  filter[isdefined(.Close)]
...
```

Spatio-Temporal Pattern Predicate

Add information to interpret the result

```
...
extend[
    PosClose: .Pos atperiods .Close,
    PosDown1: .Pos atperiods .Down1,
    PosUp: .Pos atperiods .Up,
    PosDown2: .Pos atperiods .Down2]
extend[
    TrajClose: trajectory(.Pos atperiods .Close),
    TrajDown1: trajectory(.Pos atperiods .Down1),
    TrajUp: trajectory(.Pos atperiods .Up),
    TrajDown2: trajectory(.Pos atperiods .Down2)]
  extend[
    AltClose: .Alt atperiods .Close,
    AltDown1: .Alt atperiods .Down1,
    AltUp: .Alt atperiods .Up,
    AltDown2: .Alt atperiods .Down2]
  extend[
    DerClose: .Der atperiods .Close,
    DerDown1: .Der atperiods .Down1,
    DerUp: .Der atperiods .Up,
    DerDown2: .Der atperiods .Down2]
  consume

9.1 secs
result size 201
```