Supporting Complex Thematic, Spatial and Temporal Queries over Semantic Web Data

Matthew Perry[1], Amit P. Sheth[1], Farshad Hakimpour[2], Prateek Jain[1]

2nd International Conference on Geospatial Semantics
Mexico City, MX, November 29 - 30, 2007

1. *Kno.e.sis Center, CSE Dept., Wright State University, Dayton, OH, USA*
2. *LSDIS Lab, CS Dept., The University of Georgia, Athens, GA, USA*

Outline

- Background and Motivation
- Contributions
- Modeling Approach
- Querying Approach
- Evaluation
- Conclusions

Background

Semantic Discovery*

From …..

Finding things

To …..

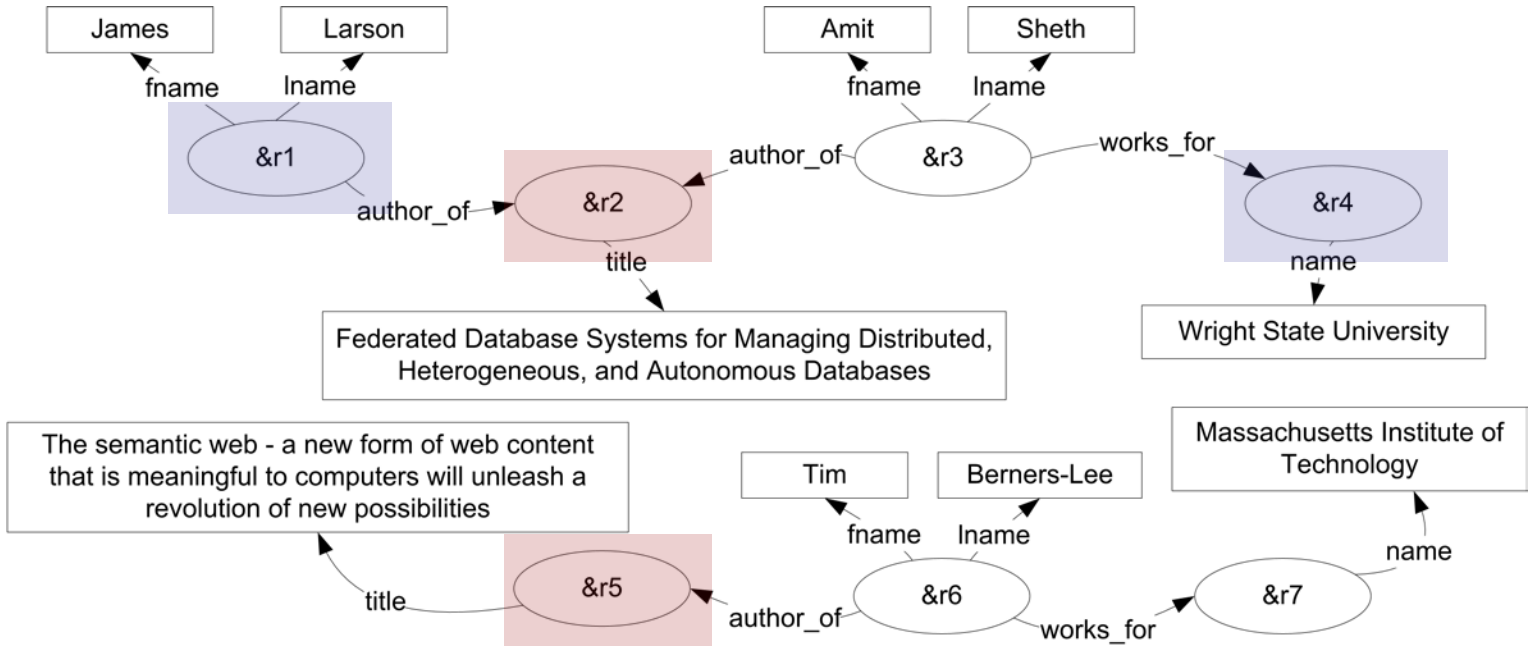
Finding out about things

Relationships!

* http://knoesis.wright.edu/projects/semdis/

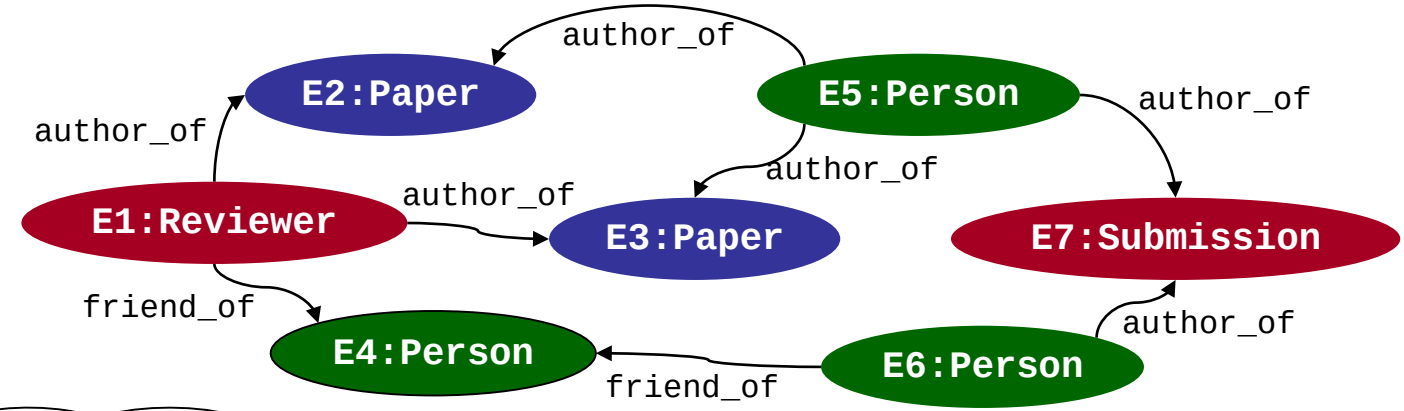
Semantic Associations

ρ-path association



ρ-iso association

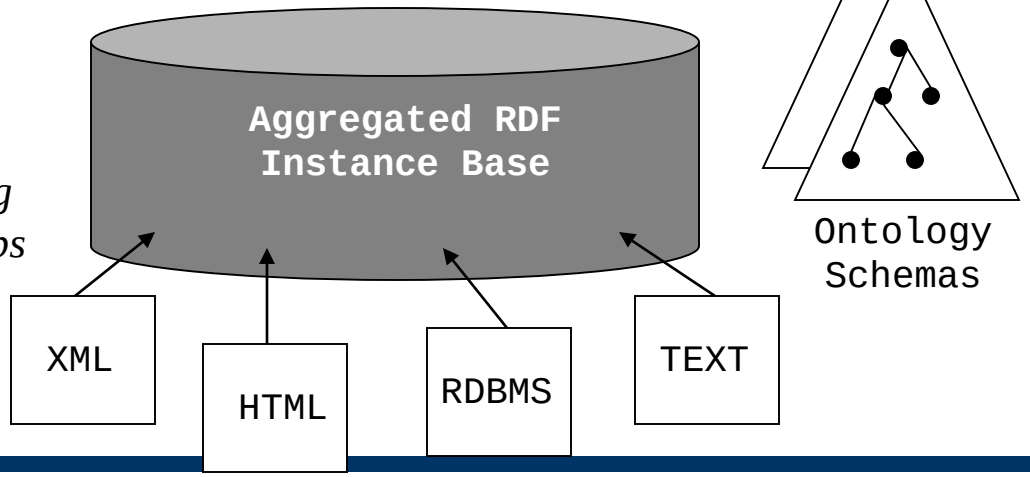
SemDis Project



How is entity1 (Reviewer) related to entity7 (Submission) ?

Semantic Analytics

Searching, Exploring and Visualizing semantically meaningful relationships



Motivation

Motivation

- In many analytical applications spatial and temporal data is critical
  - national security, regulatory compliance, etc.
- Current semantic analytics research has focused on thematic relationships
- GIS and Spatial Databases
  - Traditionally model thematic aspects as directly attached attributes of geospatial objects
  - Thematic entities/events and relationships should be represented as first-class objects to allow semantic analytics

Motivating Example

Scenario (Biochemical Threat Detection): Analysts must examine soldiers' symptoms to detect possible biochemical attack

Find all soldiers with symptoms indicative of exposure to chemical X that fought in battles within 2 miles of sightings of members of enemy group Y

```
select a from table (spatial_eval ('(?a has_symptom ?b)
  (Chemical_X induces ?b)(?a fought_in ?c)', ?c,
  '(?d member_of Enemy_Group_Y)(?d spotted_at ?e)', ?e,
  'geo_distance(distance=2 units=mile)'));
```

Query specifies

(1) a relationship between a soldier, a chemical agent and a battle location

(2) a relationship between members of an enemy organization and their known locations

(3) a spatial filtering condition based on the proximity of the soldier and the enemy group in this context

Contributions

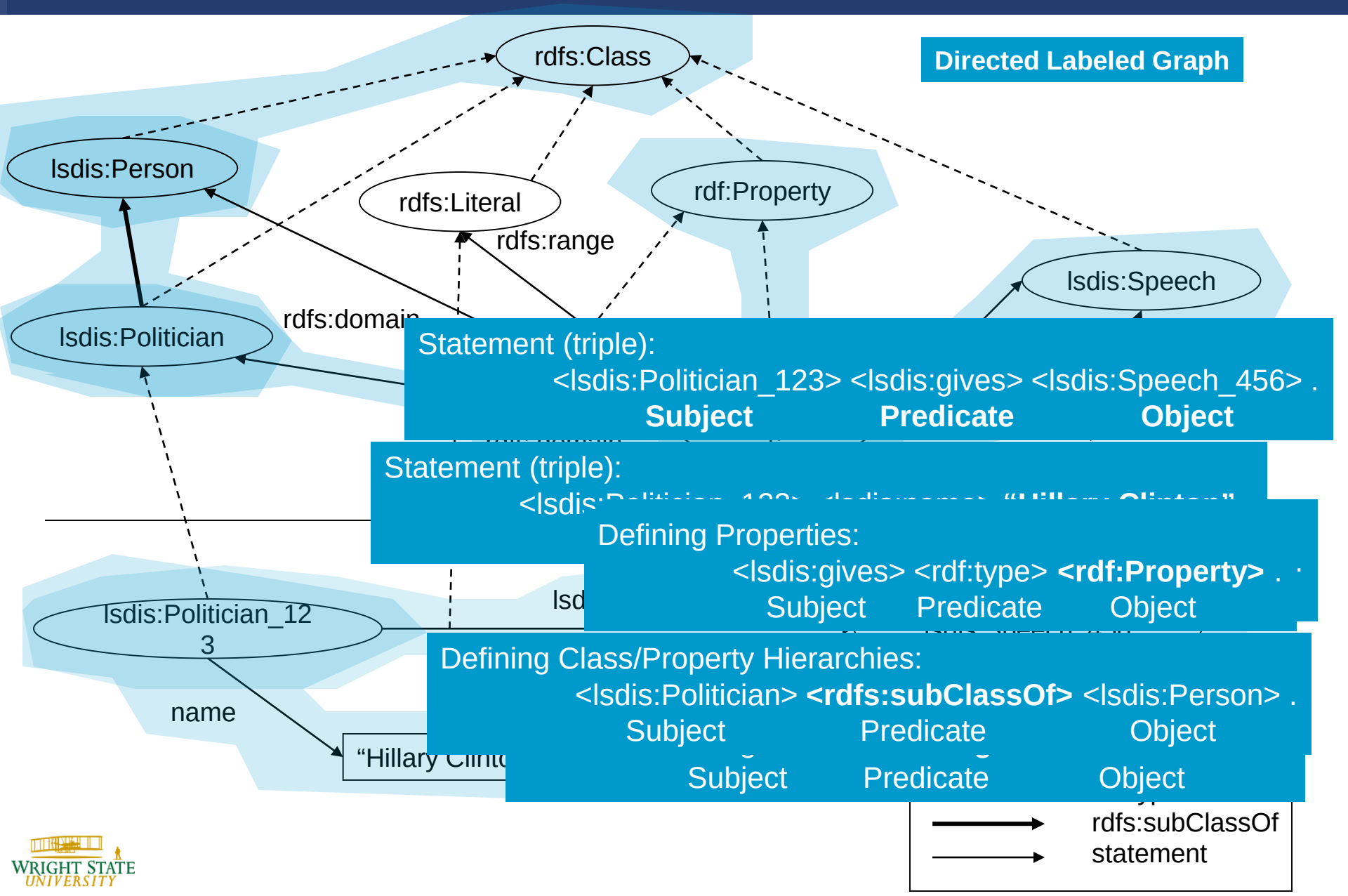
- Storage and indexing scheme for spatial and temporal SW data (i.e. RDF)
- Efficient treatment of temporal inferencing
- Definition and implementation of 4 spatial and temporal query operators
- Performance study using large dataset

Modeling Approach

Representing ontologies and instance data

- W3C standards
  - Resource Description Framework (RDF)
    - Language for representing information about resources
    - Resources are identified by Uniform Resource Identifiers (URIs) – globally-unique
    - Common framework for expressing information allows exchange and reuse without loss of meaning
    - Graph-based data model
    - Relationships are first class objects

RDF Model

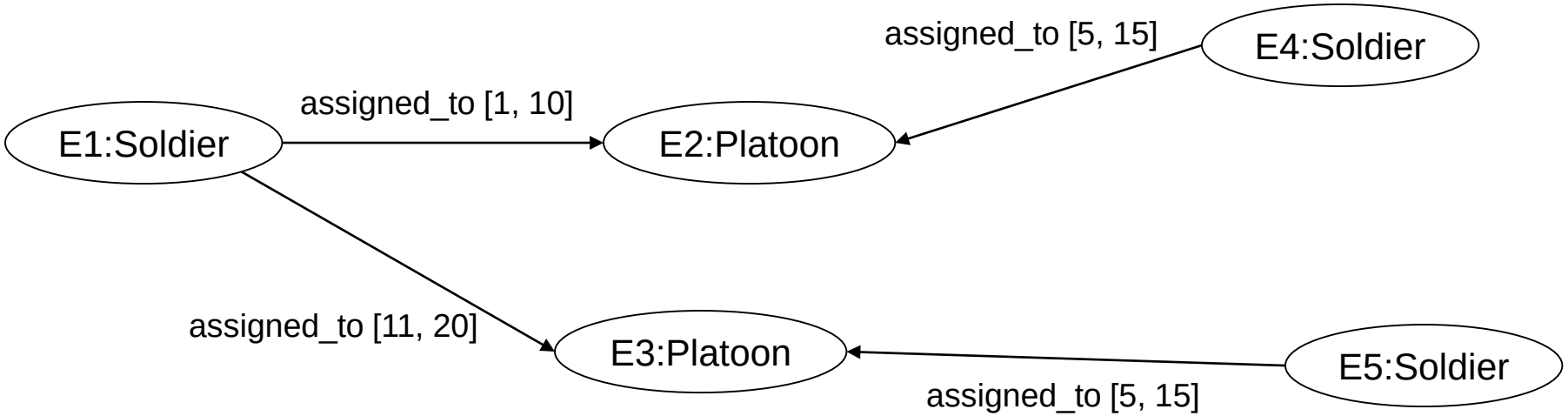

Directed Labeled Graph
rdfs:Class
lsdis:Person
rdfs:Literal
rdf:Property
rdfs:range
lsdis:Speech
lsdis:Politician
rdfs:domain
Statement (triple):
<lsdis:Politician_123> <lsdis:gives> <lsdis:Speech_456> .
Subject Predicate Object
Statement (triple):
Defining Properties:
<lsdis:gives> <rdf:type> <rdf:Property> .
Subject Predicate Object
lsdis:Politician_123
Defining Class/Property Hierarchies:
<lsdis:Politician> <rdfs:subClassOf> <lsdis:Person> .
Subject Predicate Object
name
Subject Predicate Object
rdfs:subClassOf
statement

Incorporation of Temporal Information

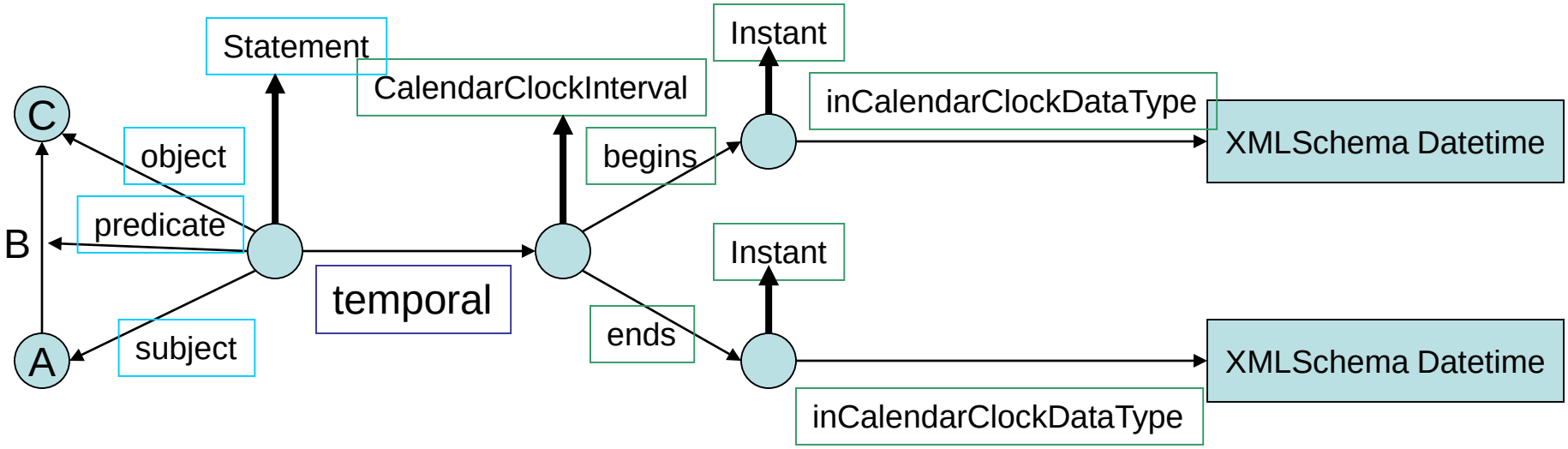
- Use Temporal RDF Graphs defined by Gutiérrez, et al[1]
- Considers time as a discrete, linearly-ordered domain
- Associate time intervals with statements which represent the valid-time of the statement
  - Essentially a quad instead of a triple

1. Claudio Gutiérrez, Carlos A. Hurtado, Alejandro A. Vaisman. **"Temporal RDF"**. ESWC 2005: 93-107

Example Temporal Graph: Platoon Membership



Temporal RDF Serialization

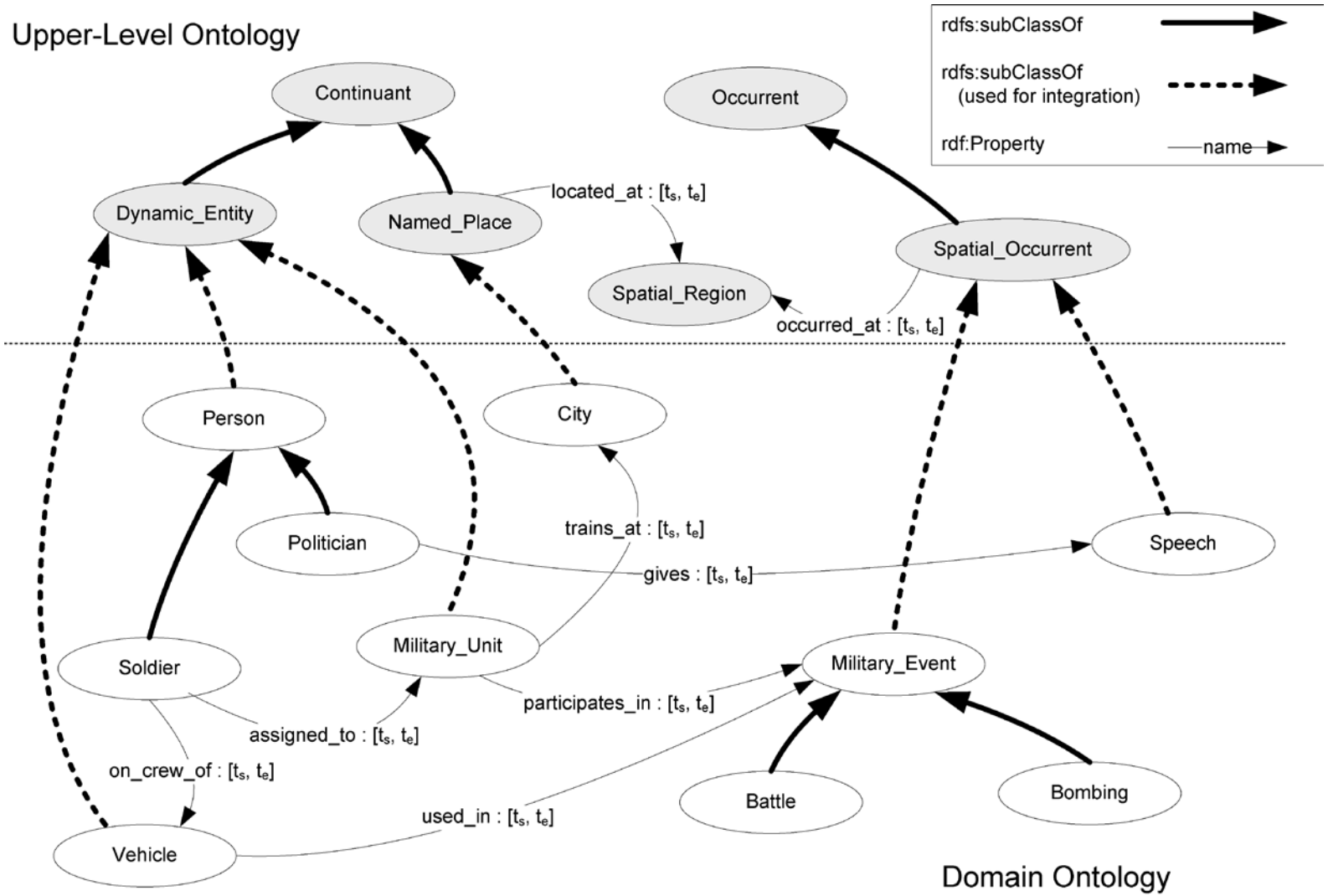


RDF Reification

OWLTime

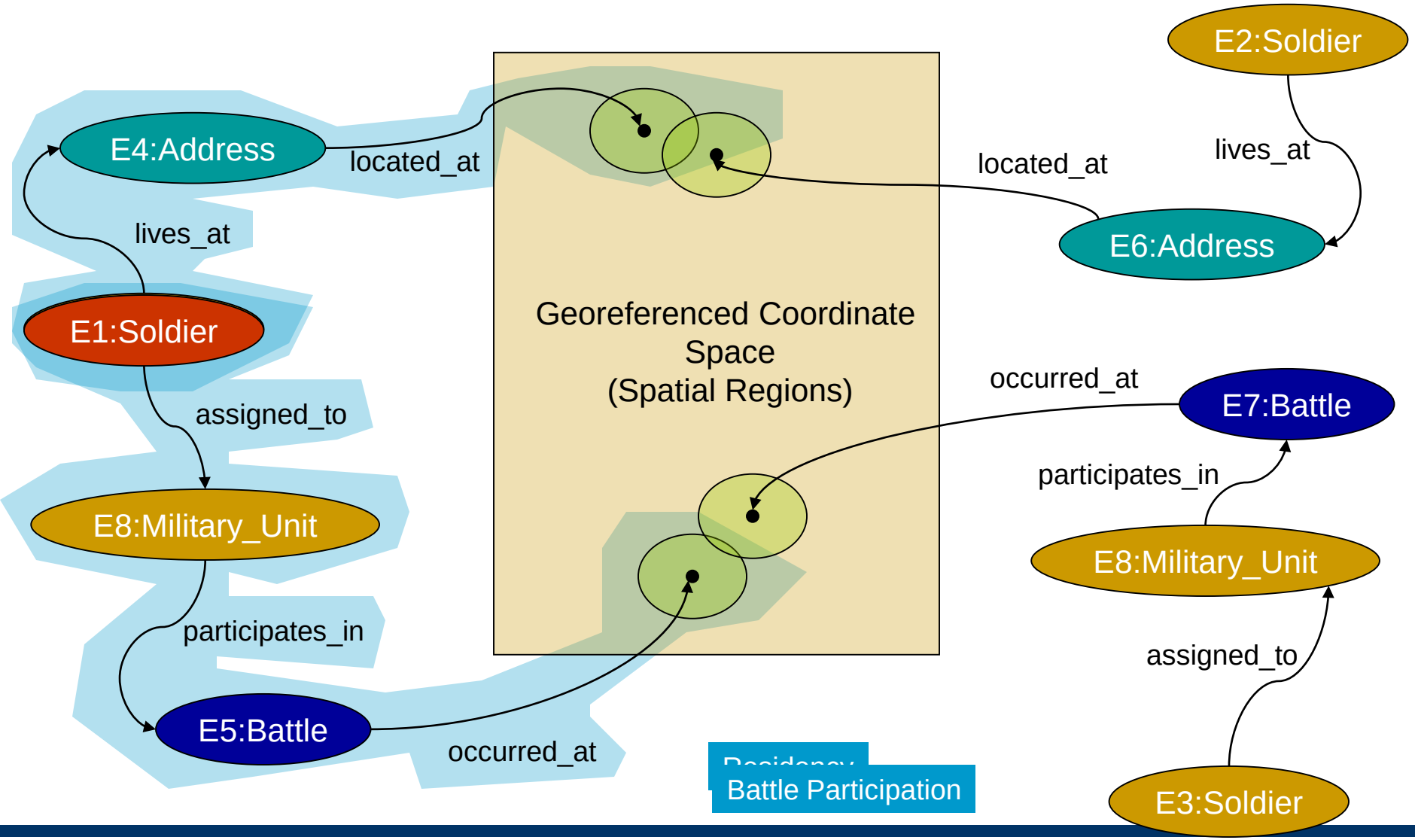
NEW

Example Domain Ontology



1. Matthew Perry, Farshad Hakimpour, Amit Sheth. **"Analyzing Theme, Space and Time: An Ontology-based Approach"**, 14th Intl Symp. on Advances in Geographic Information Systems (ACM-GIS '06), Arlington, VA, November 10 - 11, 2006

Thematic Contexts Linking Non-Spatial Entities to Spatial Entities

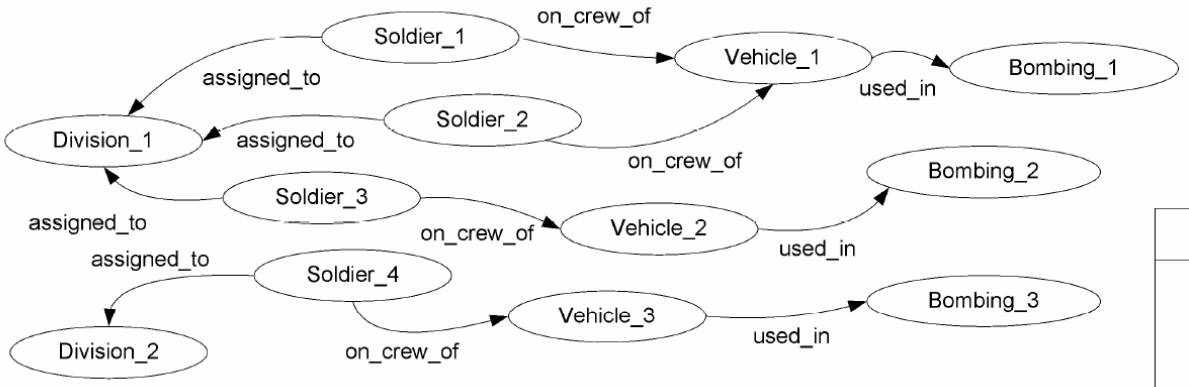


Named Places | Spatial Occurrents | Dynamic Entities

Querying Approach

Graph Patterns

RDF Data



Graph Pattern

```
(?x assigned_to Division_1)
(?x on_crew_of  ?y)
(?y used_in     ?z)
```

Result Variable Bindings

x	y	z
Soldier_1	Vehicle_1	Bombing_1
Soldier_2	Vehicle_1	Bombing_1
Soldier_3	Vehicle_2	Bombing_2

Query Operators

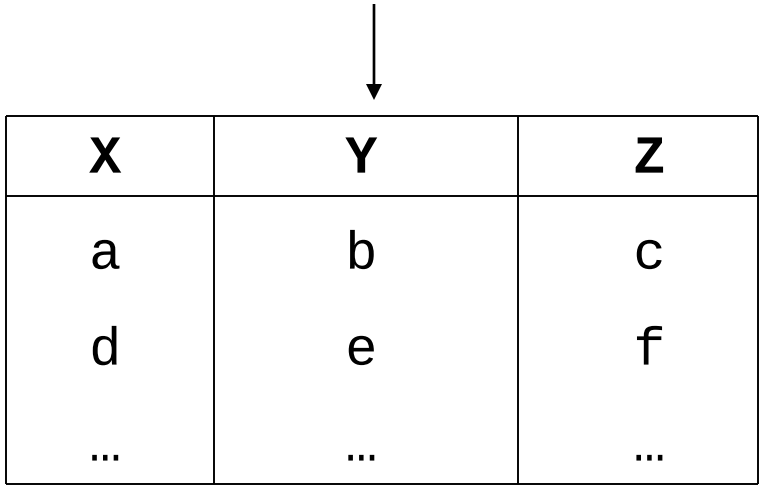
- Two spatial operators
  - spatial_extent()
    - *Find all soldiers participating in military events that take place within an input bounding box*
  - spatial_eval()
    - *Find all soldiers with symptoms indicative of exposure to chemical X that fought in battles within 2 miles of sightings of enemy group Y*
- Two temporal operators
  - temporal_extent()
    - *Return all soldiers exhibiting a given symptom during a specific time period*
  - temporal_eval()
    - *Find all soldiers who exhibited symptoms after participating in a given military event*

Implementation

- SQL-based Approach
  - Created user-defined functions in Oracle DBMS
  - Create procedures for spatial and temporal indexing
- Existing Oracle Technologies
  - Semantic Technology Component
    - Storage Structures for RDF(S) Data
    - RDFS Inference Procedures
    - SQL-based Querying
  - Spatial Component
    - Spatial Types – SDO_GEOMETRY
      - Implementation of *Spatial_Region*
    - Spatial Indexing (R-Tree)
    - Spatial Operators

SQL Table Functions

```
SELECT X, Y
FROM TABLE (Table_Func(…));
```



X	Y	Z
a	b	c
d	e	f
…	…	…

Spatial Operators (spatial_extent)

```
spatial_extent (graphPattern VARCHAR, spatialVar VARCHAR,
                ontology RDFModels, <geom SDO_GEOMETRY>,
                <spatialRelation VARCHAR>)
returns AnyDataSet;
```

Select all soldiers on the crew of a vehicle used in a military event that occurred within 45 miles of a given point

```
SELECT x
FROM TABLE (spatial_extent(
     '(?x <knoesis:on_crew_of> ?y)(?y <knoesis:used_in> ?z)
      (?z <knoesis:occurred_at> ?l)',
     'l',
     SDO_RDF_Models('military'),
     SDO_GEOMETRY(2001, 8265, SDO_POINT_TYPE(
         -71.796531, 44.304772, NULL), NULL, NULL),
     'GEO_DISTANCE(distance=45 unit=mile)')));
```

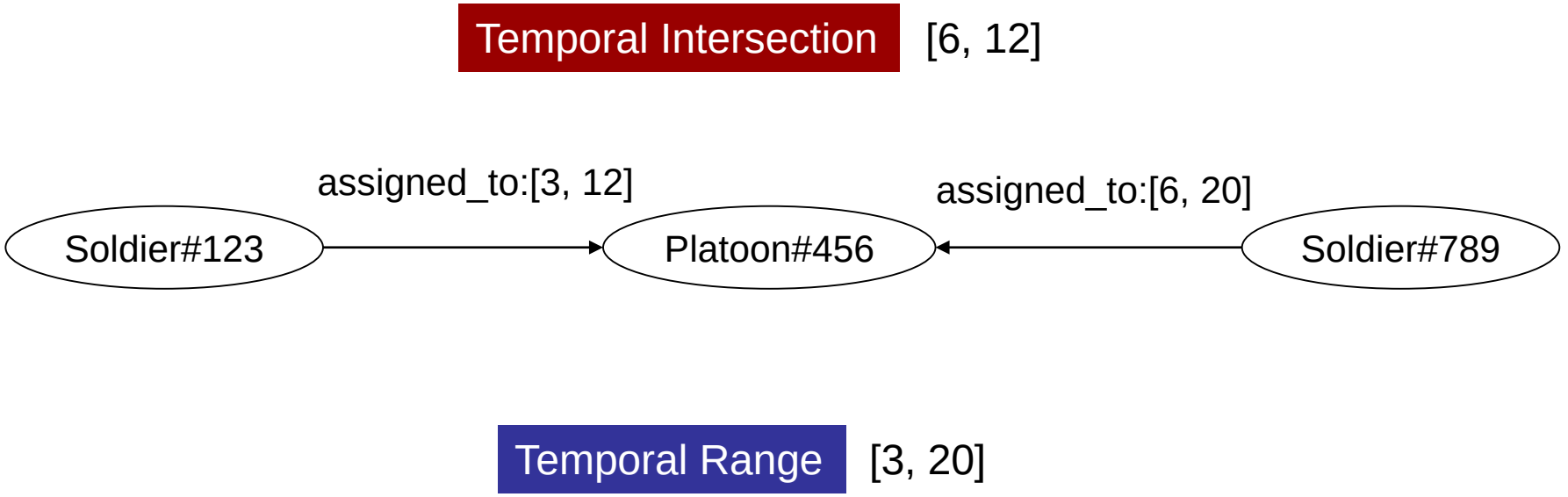
Spatial Operators (spatial_eval)

```
spatial_eval (graphPattern VARCHAR, spatialVar VARCHAR,
              graphPattern2 VARCHAR, spatialVar2 VARCHAR,
              spatialRelation VARCHAR, ontology RDFModels)
return AnyDataSet;
```

Select all platoons with a training area that overlaps the training area of Platoon_12996

```
SELECT b
FROM TABLE (spatial_eval(
  '(<knoesis:Platoon_12996> <knoesis:trains_at> ?z)
   (?z <knoesis:located_at> ?l)',
  'l',
  '(?b <knoesis:trains_at> ?c) (?c <knoesis:located_at> ?d)',
  'd',
  'GEO_RELATE(mask=overlap)',
  SDO_RDF_Models('military')));
```

Temporal Properties of Context Instances



Temporal Operators (temporal_extent)

```
temporal_extent (graphPattern VARCHAR, intervalType VARCHAR,
                 ontology RDFModels, <start DATE>, <end DATE>,
                 <temporalRel VARCHAR>)
return AnyDataSet;
```

Select all soldiers (and their platoons) on the crew of a vehicle actively used during the time period [10/04/1942, 9/21/1944]

```
SELECT x, a
FROM TABLE (temporal_extent(
   '(?x <knoesis:on_crew_of> ?y) (?y <knoesis:used_in> ?z)
    (?x <knoesis:assigned_to> ?a)',
   'INTERSECT'
   SDO_RDF_Models('military'),
   to_date('1942-10-04 00:26:01', 'yyyy-mm-dd hh24:mi:ss'),
   to_date('1944-09-21 08:22:00', 'yyyy-mm-dd hh24:mi:ss'),
   'DURING'));
```

Temporal Operators (temporal_eval)

```
temporal_eval (graphPattern VARCHAR, intervalType VARCHAR,
               graphPattern2 VARCHAR, intervalType2 VARCHAR,
               temporalRel VARCHAR, ontology RDFModels)
return AnyDataSet;
```

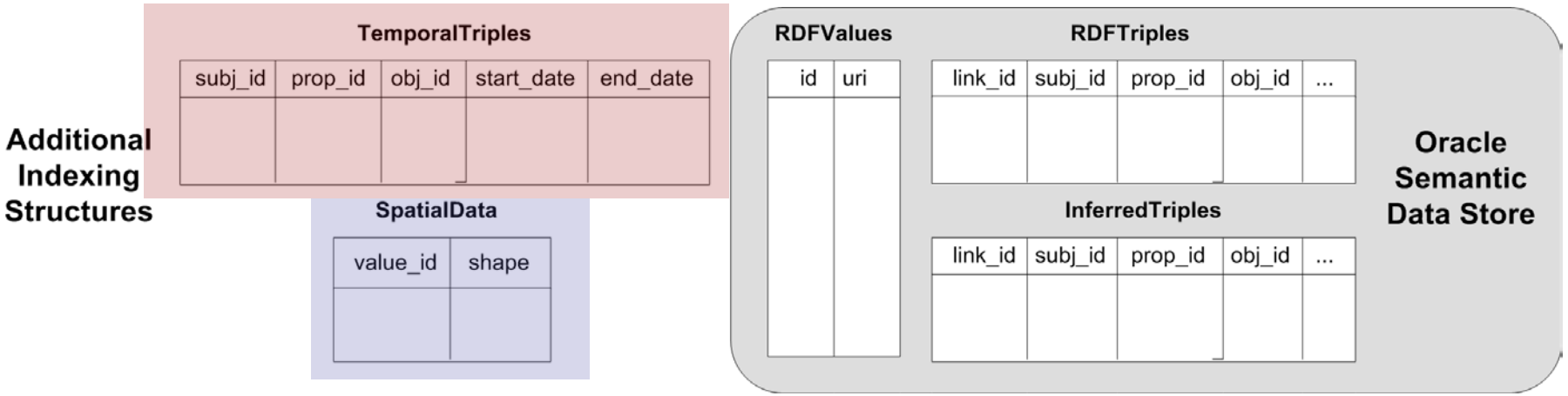
Find all pairs of soldiers such that one soldier was leader of a platoon in Division_2186 and the other soldier was leader of a platoon in Division_2191 during the same time period

```
SELECT x, a
FROM TABLE (temporal_eval(
   '(?x <knoesis:leader_of> ?y) (?y <knoesis:platoon_of> ?z)
    (?z <knoesis:battalion_of> <knoesis:Division_2186>)',
   'INTERSECT',
   '(?a <knoesis:leader_of> ?b) (?b <knoesis:platoon_of> ?c)
    (?c <knoesis:battalion_of> <knoesis:Division_2191>)',
   'INTERSECT',
   'ANYINTERACT',
   SDO_RDF_Models('military')));
```

Storage Scheme

Temporal Indexing Procedure

Load RDF Data with Oracle



Spatial Indexing Procedure

Temporal Inferencing

RDFS Inferencing Rules

1. (*x*, *rdf:type*, *y*) AND (*y*, *rdfs:subClassOf*, *z*) → (*x*, *rdf:type*, *z*)
2. (*x*, *p*, *y*) AND (*p*, *rdfs:domain*, *a*), (*p*, *rdfs:range*, *b*) → (*x*, *rdf:type*, *a*), (*y*, *rdf:type*, *b*)
3. (*x*, *p*, *y*) AND (*p*, *rdfs:subPropertyOf*, *z*) → (*x*, *z*, *y*)

Example:
(*x*, *participates_in*, *e*):[2, 5]
(*y*, *participates_in*, *e*):[3, 7]
(*z*, *participates_in*, *e*):[6, 9]

By rule 2: (*e*, *rdf:type*, *event*):[2, 9]

Temporal Inferencing Algorithm

1. create table *asserted_temporal_triples* (*subj*, *prop*, *obj*, *start_date*, *end_date*)
2. perform schema-level inferencing
3. perform instance-level inferencing
4. sort *redundant_triples* by (*subj_id*, *prop_id*, *obj_id*, *start*)
5. make a single pass and merge overlapping intervals for same statement
6. insert updated triples and intervals into final *temporal_triples* table

asserted_temporal_triples
(*x*, *participates_in*, *e*):[2, 5]
(*y*, *participates_in*, *e*):[3, 7]
(*z*, *participates_in*, *e*):[6, 9]

redundant_triples
(e, rdf:type, event):[2, 5]
(e, rdf:type, event):[3, 7]
(e, rdf:type, event):[6, 9]

temporal_triples
(e, rdf:type, event):[2, 9]

Function Implementation

- Oracle Extensibility Framework
- ODCITable Interface
  - Start()
  - Fetch()
  - Close()

Spatial Function Implementation

1. Start()
    1. Translate graph pattern into multi-way join over temporal_triples table
    2. Add sdo_relate or sdo_within_distance predicate to reflect spatial relation parameter
2. Fetch()
    1. Retrieve rows until all are returned
3. Close()
    1. Close cursor for query

For spatial_eval:

Nested Loop Join Strategy

outer spatial_extent()

inner filtered spatial_extent()

Temporal Function Implementation

1. Start()
    1. Translate graph pattern into multi-way join over temporal_triples table
    2. Push temporal filtering down as much as possible
       e.g., `RANGE during [x, y]` → all statements `start >= x AND end <= y`
2. Fetch()
    1. Retrieve an intermediate result row
    2. Construct INT/RANGE interval for result row
    3. Apply temporal filtering condition and return matching rows
3. Close()
    1. Close cursor for query

For temporal_eval:

Nested Loop Join Strategy

outer temporal_extent()

inner filtered temporal_extent()

Evaluation

Evaluation

- Environment
  - Oracle 10g R2 on RedHat EL
  - Dual Xeon 3.0 GHz, 2GB RAM
  - 512 MB buffer cache
- Objective
  - Test scalability w.r.t (1) dataset size, (2) query complexity
- Dataset
  - Thematic: Synthetic RDF Graph (Military Schema)
    - Generated with TOntoGen[1]
    - 100k triples 1.6 M triples 15 M triples
  - Spatial: US Census Block Group boundary polygons
    - 873 9,352 83,236
  - Temporal: start and end times selected with uniform probability from 2 overlapping date ranges

1. Matthew Perry **"TOntoGen: A Synthetic Data Set Generator for Semantic Web Applications"**, AIS SIGSEMIS Bulletin Volume 2 Issue 2 (April - June) 2005, pp. 46 - 48

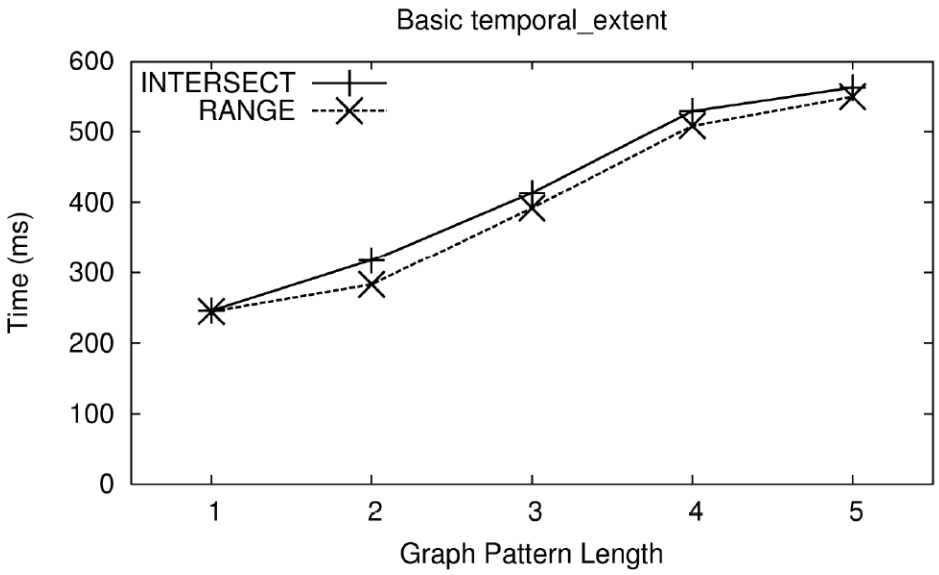
Scalability w.r.t Dataset Size

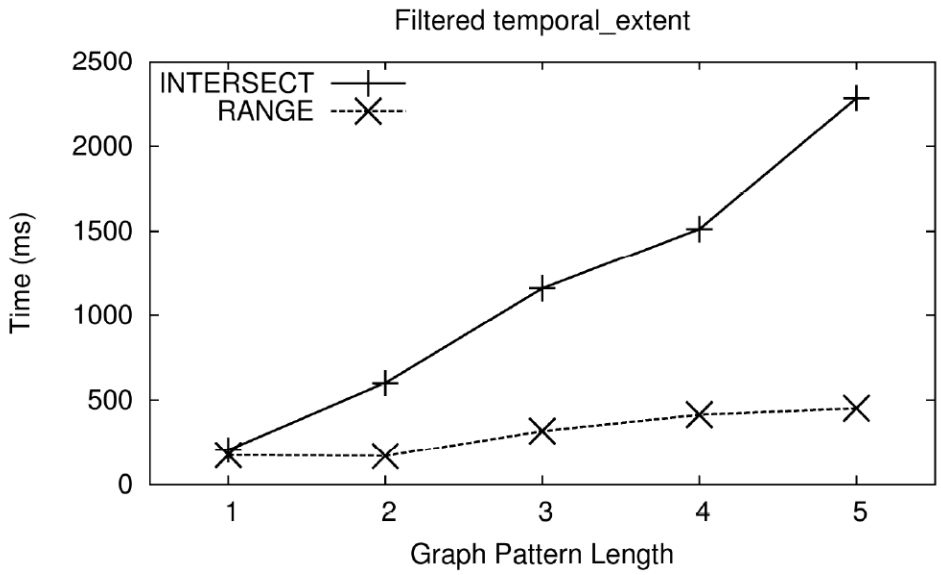
Table 1. Experimental results for query execution time with respect to ontology size

Operator (Exp. #)	Graph Pattern Type		Queries	Avg. Result Size	Avg. Execution Time for each ontology (ms)		
	# Vars	# Triples			Small	Medium	Large
T-Ext (1)	4	3	4	N/A	394	390	385
(2)	3	3	5	221	22	32	48
S-Ext (3)	4	3	3	N/A	360	350	365
(4)	3	3	3	100	22	30	67
T-Filter(5)	4	3	4	312	157	345	714
S-Filter (6)	4	3	3	331	173	192	374
T-Eval(7)	2/2	2/2	3	129	414	411	437
	2/3	3/3	3	220	306	195	268
S-Eval (8)	2/2	2/1	3	244	343	467	485
	2/2	2/3	3	209	251	385	457

All times reported are average of 15 trials using a warm cache

Scalability w.r.t. Graph Pattern Size





Times reported for 15 million triple dataset

Conclusions and Future Work

- Conclusions
  - Approach for realizing spatial and temporal queries over SW data
  - Motivated by lack of support in current semantic analytics tools
  - Good scalability for large synthetic dataset
- Future Work
  - SPARQL query language extensions