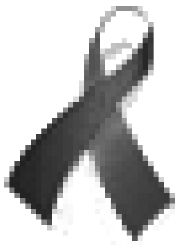


Studying an Approach to Query Spatial XML

Jose E. Córcoles

Pascual González

Main aim

- Semantic Geospatial Web
- We are defining a system to integrate non-spatial resources with spatial resources
- “Web pages about all restaurants located to a distance a less than 20 meters of the *Wörthersee* lake from *Klangerfurt*”

Aim of this Paper

- We study a prototype for querying spatial XML resources.
- The main task of this approach is to provide users with a unique interface for querying spatial XML resources with different schemas, independently of their actual organization and location

Aim of this Paper

- The resources are integrated using RDF.
- The most novel and critical feature of this approach is the querying of spatial XML resources, because it uses a different way from that of querying and relating non-spatial resources.

Overview

- The main task of an integration mediator is to provide the users with a unique interface for querying the data, independently of its actual organization and location.
- This interface, or global schema, is described as an *ontology* expressed with RDF(S).
- As used here, an *ontology* denotes a light-weight conceptual model and not a hierarchy of terms or a hierarchy of concepts

Overview

- Users pose queries in terms of global schema (RDFS).
- The mediator system must contain a module (*Solve mapping*) that uses the resource descriptions in order to translate a user query into a query that refers directly to the schemas of the GML resources.
- We establish a correspondence between each resource and the global schema using RDF instances.

Restrictions

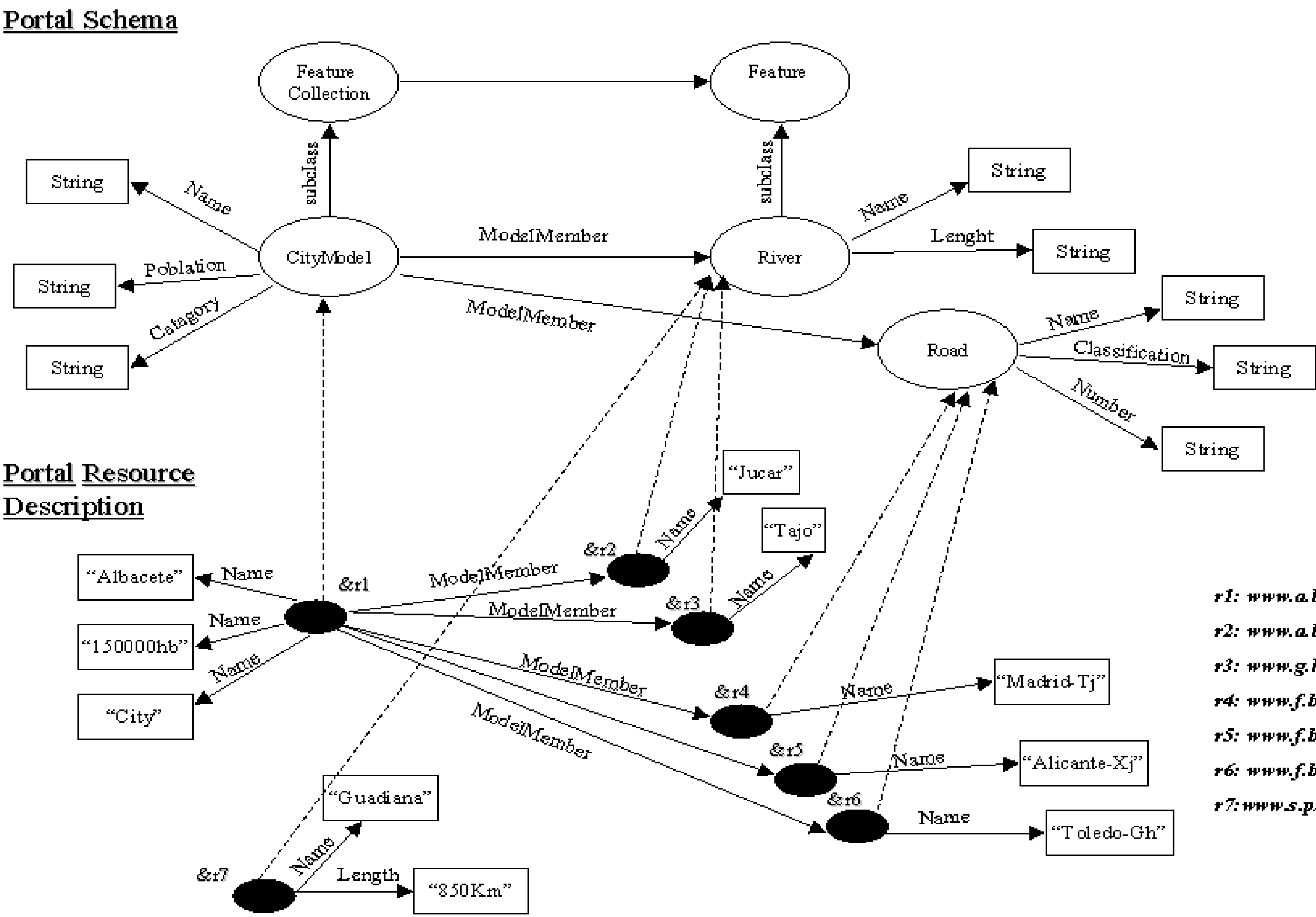
- The spatial query language used for the users should have the same semantic as the query language used in the sources, so the translation between both languages is easier.
- Due to this an efficient method for storing GML documents is necessary.
- Our approach uses *description* to obtain *a priori* a set of candidate resources.

Approach based on *RDF*

- A *Community Web Portal (C-Web)* integrates semantically various information resources (sites, documents, data).
- *Focused on Catalog*
- *Catalog = RDF(S)*
- *Catalog is queried by RQL*

Approach based on *RDF*

- GML documents (or part of) are a resource. (Spatial operators)
- Extension of RQL to support spatial operators
- Extension of C-Web architecture to support spatial operators.



r1: www.a.b/cities.gml

r2: www.a.b/riversJucar.gml

r3: www.g.h/rivers.gml#Tajo

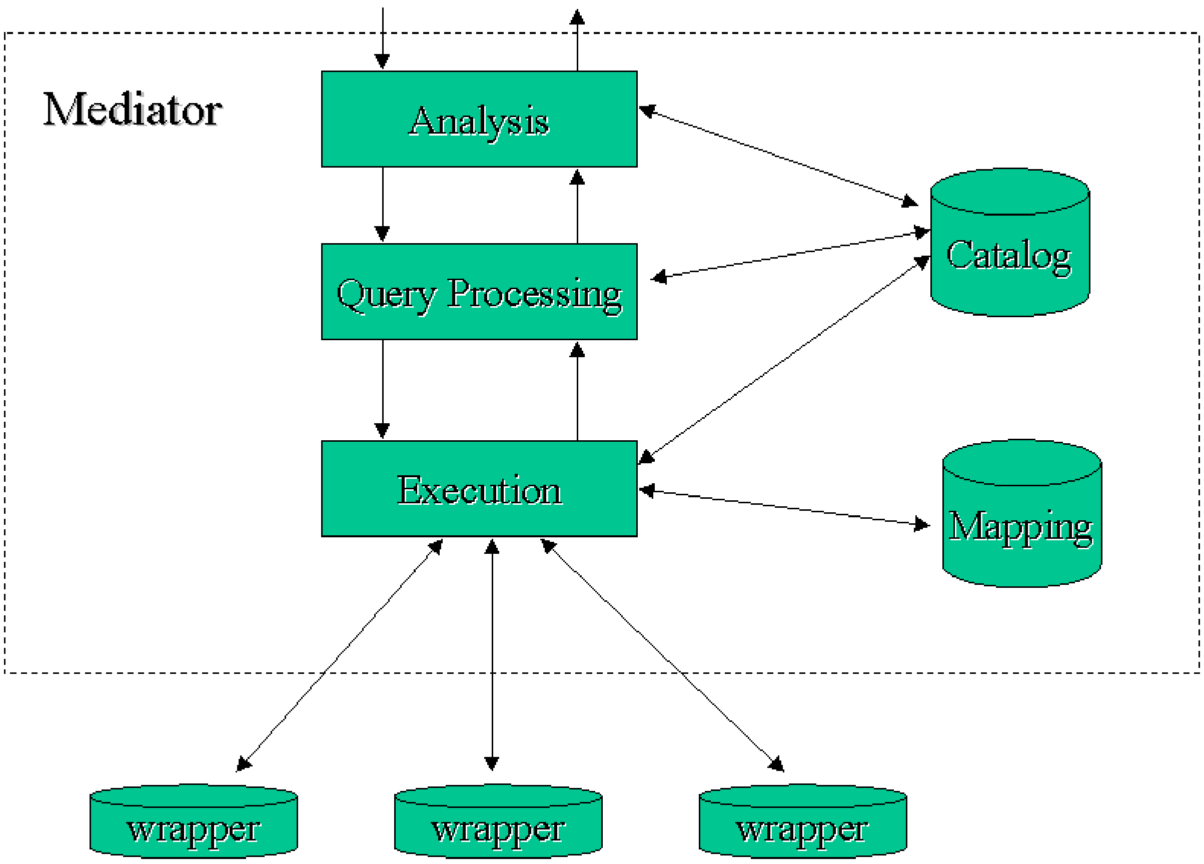
r4: www.f.b/roads.gml#madrid

r5: www.f.b/roads.gml#alicante

r6: www.f.b/roads.gml#toledo

r7: www.s.p/guadiana.gml

RDF Portal Architecture



A Performance Study

We focus on the effectiveness of this approach in terms of …

- translation of queries over an ontology into queries supported by the wrappers,
- translation of these queries into spatial SQL queries,
- query processing and
- generation of GML documents from the results.

A Performance Study

- All the experiments were conducted on an 2300Mhz PC with 512Mb RAM, 40Gb hard disk with operating system Windows XP professional.
- The RDBMS used was *Oracle9i Spatial 9.0.1*, which we selected because it allows the storage of spatial objects and the application of spatial operators over them. Oracle's Spatial object-relational model was used (SDO_GEOMETRY object)

A Performance Study

- This data set represents a *City model* where a city has several blocks, each block has several parcels and each parcel has an owner.
- We used a data set with 7.1 Mb approx. and 5000 rectangular parcels.

Elapsed time

We have focused our attention on the following aspects involved in assessing performance:

1. Translating queries over an ontology into queries supported by the wrappers has an elapsed time that should be studied.

3.2. Translating queries from our data model to the relational model (spatial SQL) needs a mapping process.

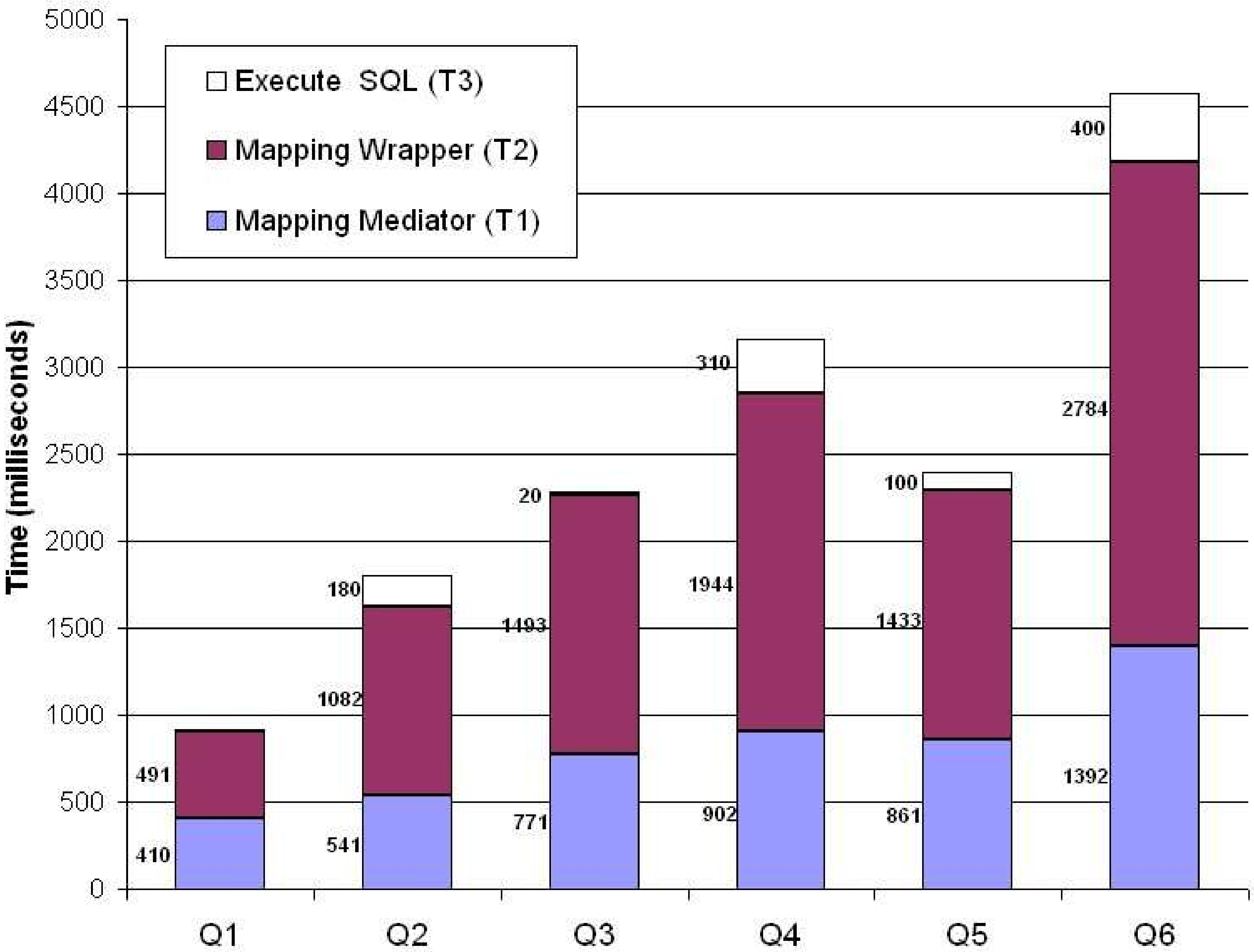
4.3. Since we are dealing with an XML query language, a very large number of joins is necessary.

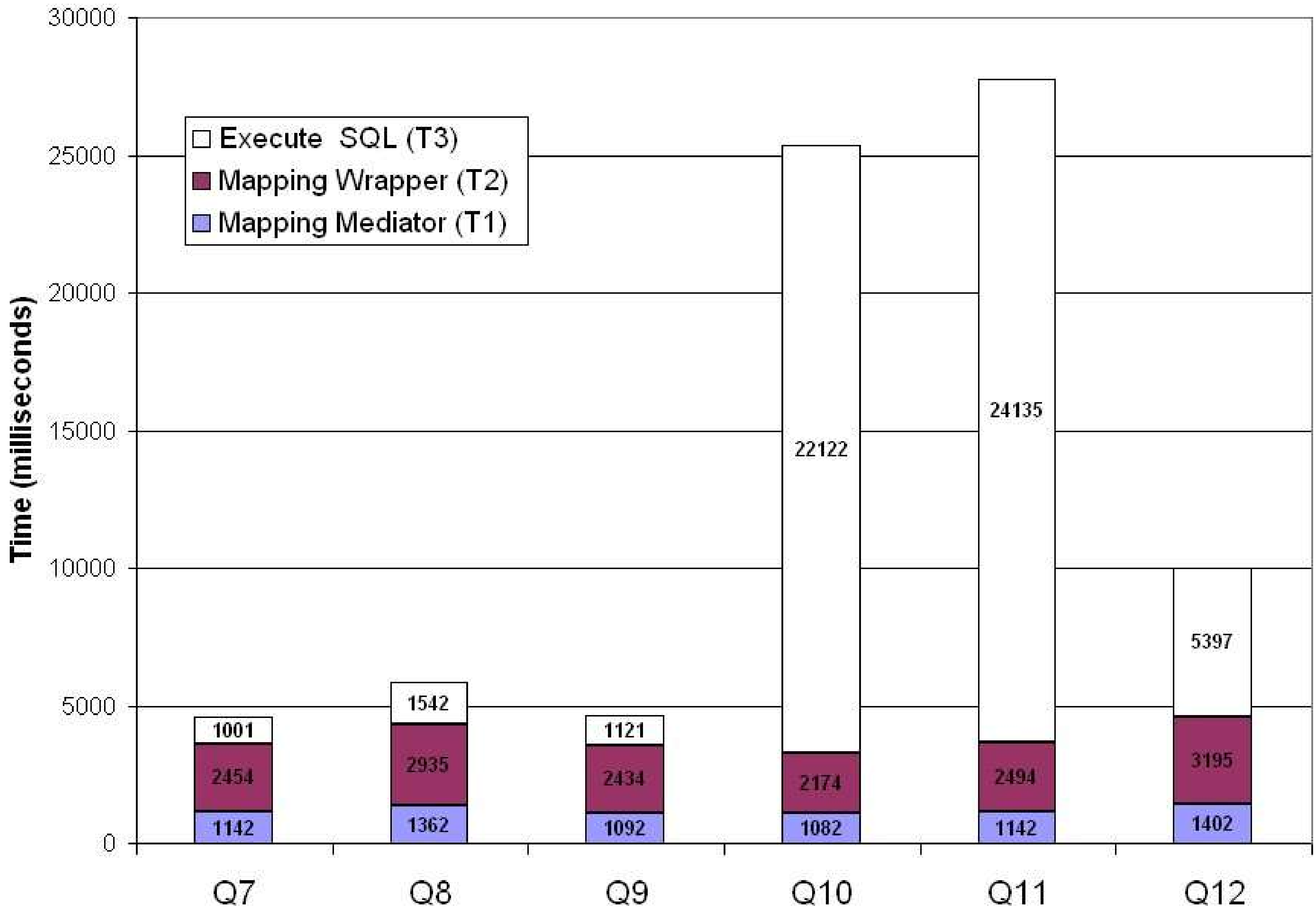
Queries Studied

Q1	*Select the Owner(firstname), Owner(lastname) where Owner(age) > 20*
Q2	*Obtain Block(Number), Parcel(extendof),State(name) where Block(name) like "bloque%"*
Q3	*Obtain Block(extendof), Parcel(extendof) State(name) where Block(name) like "bloque%"and Block(id) = Parcel(number_floors)*
Q4	*Select Owner(name), Block(extendof), Parcel(bundaryby),State(name) where Block(name) like "bloque% "and Block(id) = Parcel(number_floors) and Owner(age) > Parcel(n_flats)*
Q5	*Obtain Area[Buffer(Block(extendof),20)], Length(Parcel(extendof)),State(name) where Block(name) like "bloque%"and Block(id) = Parcel(number)*
Q6	*Obtain Owner(fisrtname), Area[Buffer(Block(extendof),20)], Convexhull(Parcel(extendof)),State(name) where Block(name) like "bloque%"and Block(id) = Parcel(number) and Owner(age) > Parcel(number_floors)*
Q7	*Obtain Owner(fisrtname), Union(Block(extendof),Parcel(extendof)),State(name) where Block(name) like "bloque%"and Block(id) = Parcel(number) and Owner(age) > Parcel(number_floors)*
Q8	*Obtain Area[Block(extendof)], Union[Block(extendof),Parcel(extendof)],State(name) where Block(name) like "bloque%"and Block(id) = Parcel(number) and Owner(age) > Parcel(number_floors) and Length [Block(extendof)]>10*
Q9	*Obtain Area[Block(extendof)], Union[Block(extendof),Parcel(extendof)] where Block(name) like "bloque%"and Block(id) = Parcel(number) and Owner(age) > Parcel(number_floors) and Length[Block(extendof)]>10*
Q10	*Obtain Owner(fisrtname), Union(Block(extendof),Parcel(extendof)), Area[Block(extendof)] where Block(name) like "start%y"and Block(id) = Parcel(number) and Block(extendof) Intersects Parcel(extendof)*
Q11	*Obtain Owner(fisrtname), Intersection (Block(extendof),Parcel(extendof)),State(name) where Block(name) like "start%y and Block(id) = Parcel(number) and Block(extendof) Intersects Parcel(extendof)*
Q12	*Obtain Area[Block(extendof)], Union(Block(extendof),Parcel(extendof)), State(name) where Block(name) like "start%y and Block(id) = Parcel(number) and Block(extendof) Intersects Parcel(extendof) and Owner(age) > Parcel(number_floors) and Area[Block(extendof)]>10*

Joins

Query	Nº Joins	Relations		Query	Nº Joins	Relations
Q1	*0*	1		**Q7**	*5+1+1*	*6*
Q2	*4*	5		**Q8**	*5+1+1*	*6*
Q3	*4+1*	5		**Q9**	*3+1+1*	*4*
Q4	*5+1+1*	6		**Q10**	*3+1+1s*	*4*
Q5	*2+1*	3		**Q11**	*5+1+1s*	*6*
Q6	*5+1+1*	6		**Q12**	*5+1+1s+1*	*6*


Execute SQL (T3)
Mapping Wrapper (T2)
Mapping Mediator (T1)
Time (milliseconds)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
Q1
Q2
Q3
Q4
Q5
Q6
410
491
541
1082
180
771
1493
20
902
1944
310
861
1433
100
1392
2784
400


Execute SQL (T3)
Mapping Wrapper (T2)
Mapping Mediator (T1)
Time (milliseconds)
30000
25000
20000
15000
10000
5000
0
1001
2454
1142
1542
2935
1362
1121
2434
1092
22122
2174
1082
24135
2494
1142
5397
3195
1402
Q7
Q8
Q9
Q10
Q11
Q12

A Performance Study

- In order to obtain a good performance querying spatial XML documents, it is preferable to increase the size of relations, and to reduce the number of joins between relations.
- For each source a particular heuristic must be applied, depending on the kind of information stored on them. In this way, we can reduce the number of joins in each query. This will reduce the elapsed time to execute spatial SQL and the elapsed time to get the *from* clause in the wrapper mapping.

Conclusions

- In this paper a prototype of a mediation system for querying XML spatial resources is studied.
- The main feature of this approach is the possibility of applying spatial operators over the resources that are represented in GML format.
- We studied three elapsed times: regarding the elapsed time for the mapping in the mediator, we concluded that it depends on the number of resources involved in the query.

Conclusions

- For the same number of resources, the number of properties is not determinative. The elapsed time for the mapping in the wrapper is more expensive.
- Moreover, it is more variable because it depends on the heuristic used to represent a GML data model over the relational model in each resource.
- An heuristic with several relations offers a worse elapsed time in the mapping . The same problem can be applied over the execution of the query. The elapsed time for execution obtains better results with fewer relations.

Currently

- Future work foresees we will extend this study with a scalability study. Furthermore, we will included more elapsed time (e.g. generating GML documents from the results).
- In addition, we are developing a prototype to integrate GML resources and any other kind of resource (HTML, PDF, etc). Thus, with this approach it is possible to discover spatial and non-spatial resources which are interrelated semantically on the Web. The study presented in this paper will be used to check the efficiency of this future approach