A Data Integration Approach To Dynamically Fusing Geospatial Sources

Snehal Thakkar

Ph. D. Dissertation

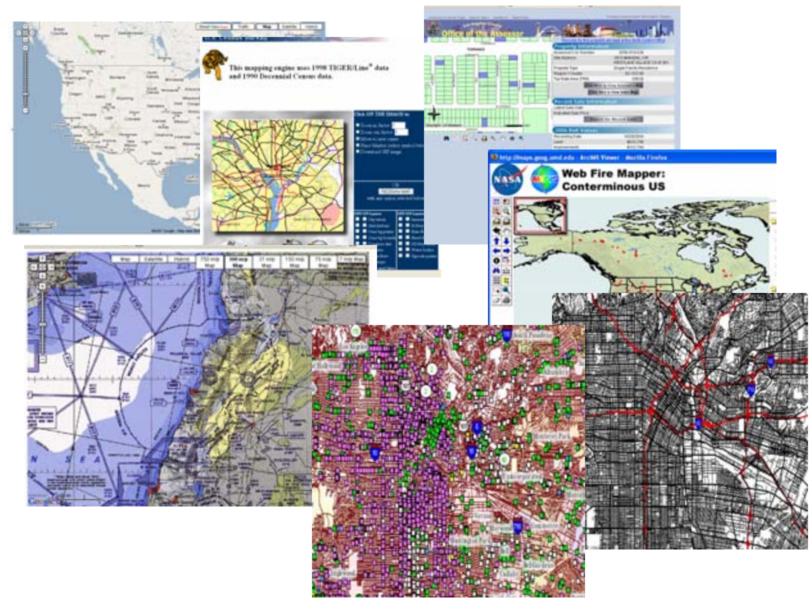
August 2007

Outline

- Introduction & motivation
- Quality-driven Geospatial Mediator (QGM)
  - Representing content and quality
  - Automatic source description generation
    - Content
    - Quality
  - Quality-driven query answering
  - Plan execution
- Related work
- Conclusions & future work

Introduction & Motivation

- Many disaster response and urban planning require integrated view of geospatial data

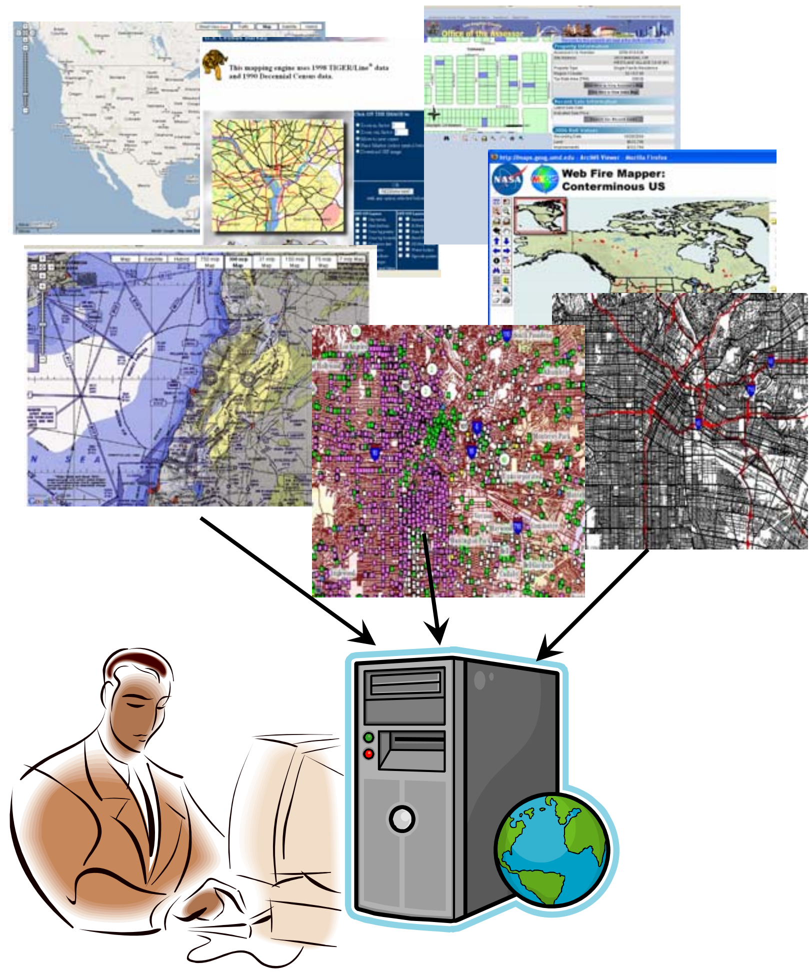


Introduction & Motivation

- Many disaster response and urban planning require integrated view of geospatial data
- Manually integrating geospatial data from a large number of sources is very hard

Introduction & Motivation

- Many disaster response and urban planning require integrated view of geospatial data
- Manually integrating geospatial data from a large number of sources is very hard
- There is a need for a geospatial data integration framework that
  - Automatically generates representations of sources
  - Dynamically provides high quality data



Thesis Statement

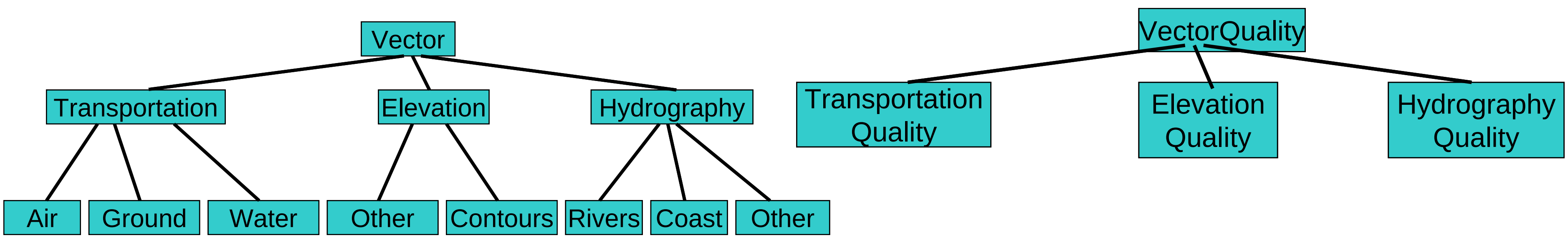
- This thesis demonstrates that by discovering geospatial sources available on the web, automatically learning the representations of both the content and the quality of data provided by the discovered sources, and exploiting the representations of the sources during query answering we can provide high quality geospatial data in response to user queries.

Outline

- Introduction & motivation
- Quality-driven Geospatial Mediator (QGM)
  - Representing content and quality
  - Automatic source description generation
    - Content
    - Quality
  - Quality-driven query answering
  - Plan execution
- Related work
- Conclusions & future work

Representation: Domain Concepts

- Content
  - Set of domain concepts by merging FGDC, NGA, and NationalMap concepts
- Quality
  - Similar hierarchy for quality of data for each domain concept
    - E.g. Road → RoadQuality



Representation: Domain Concept Attributes

- Vector
  - source, type, format, cs, bbox, vectorobj
- Raster
  - source, type, format, cs, bbox, size, resolution, rasterobj
- VectorQuality
  - source, type, date, completeness, resolution, horizontalaccuracy, verticalaccuracy, vectorswithinaccuracybounds, attributecompleteness
- RasterQuality
  - source, type, date, completeness, originalresolution, multispectral

Representation: Source Descriptions

- Source represented using two relations
  - Content
  - Quality
- Datalog descriptions
  - Content
    - Type of data: domain relation in the body
    - Coverage specified using constraints with spatial operations
  - Quality
    - Facts specifying the quality
    - Rule defining the relationship with corresponding quality relation

```
NavteqRoads(bbox, vectorobj):-
   Roads(type, format, cs,
       bbox, source, vectorobj)^
   bbox coveredby
       `[[33,-117],[34,-118]]'^
   source = `Navteq'^
   type = `Roads'^
   format = `Shapefile'^
   cs = `EPSG:4326'
```

Representation: Source Descriptions

- Source represented using two relations
  - Content
  - Quality
- Datalog descriptions
  - Content
    - Type of data: domain relation in the body
    - Coverage specified using constraints with spatial operations
  - Quality
    - Facts specifying the quality
    - Rule defining the relationship with corresponding quality relation

```
NavteqRoadsQuality(res, date,
   horiz-acc, vert-acc,
   vectorsin-acc-bounds,
   attr-comp, completeness):-
 RoadQuality(source, type, res, date,
   horiz-acc, vert-acc, vectorsin-acc-
   bounds, attr-comp, completeness)^
 source = `Navteq'^
 type = `Roads'

NavteqRoadsQuality(5,1/1/2005,3.6,
        3.6, 85%, 90%, 96%)
```

Representation: Queries

- Expressed by Datalog rules
- Three parts: data, quality, combination
  - Predicates allowed
    - Domain relations
    - Operations
      - Spatial selection
        - intersects, coveredby, disjoint
      - Aggregate
        - pack, unpack, sum, average,
        - min, max, skylinemin, skylinemax
      - Mathematical
        - add, subtract, multiply, divide
      - Order Constraints
        - e.g. completeness > 50

Representation: Sample Query 1

- Find road vector data covering the bounding box `[[33,-115],[34,-116]]' with completeness over 50%

```
Q1(vectorobj, completeness):-
   Q1Data(type, source, vectorobj) ^
   Q1Quality(type, source, completeness)

Q1Data(type, source, vectorobj):-
   Roads(type, format, cs, bbox, source, vectorobj) ^
   bbox coveredby `[[33,-115],[34,-116]]'

Q1Quality(type, source, completeness):-
   RoadQuality(source, type, res, date, horiz-acc,
         vert-acc, vectorsin-acc-bounds, attr-comp,
         completeness) ^
   completeness > 50
```

Representation: Sample Query 2

- Find road vector data covering the bounding box `[[33,-115],[34,-116]]' with the highest completeness

```
Q2(vectorobj, completeness):-
   Q2Data(type, source, vectorobj) ^
   Q2Quality(type, source, completeness)

Q2Data(type, source, vectorobj):-
   Roads(type, format, cs, bbox, source, vectorobj) ^
   bbox coveredby `[[33,-115],[34,-116]]'

Q2Quality(type, source, completeness):-
   RoadQuality(source, type, res, date, horiz-acc, vert-acc,
      vectorsin-acc-bounds, attr-comp, completeness) ^
   pack(completeness, packedcompleteness) ^
   max(packedcompleteness, maxcompleteness) ^
   maxcompleteness = completeness
```

Representation: Sample Queries 3

- Find satellite image and road vector data covering bounding box `[[33,-116],[34,-117]]' such that both the resolution and date differences are minimized

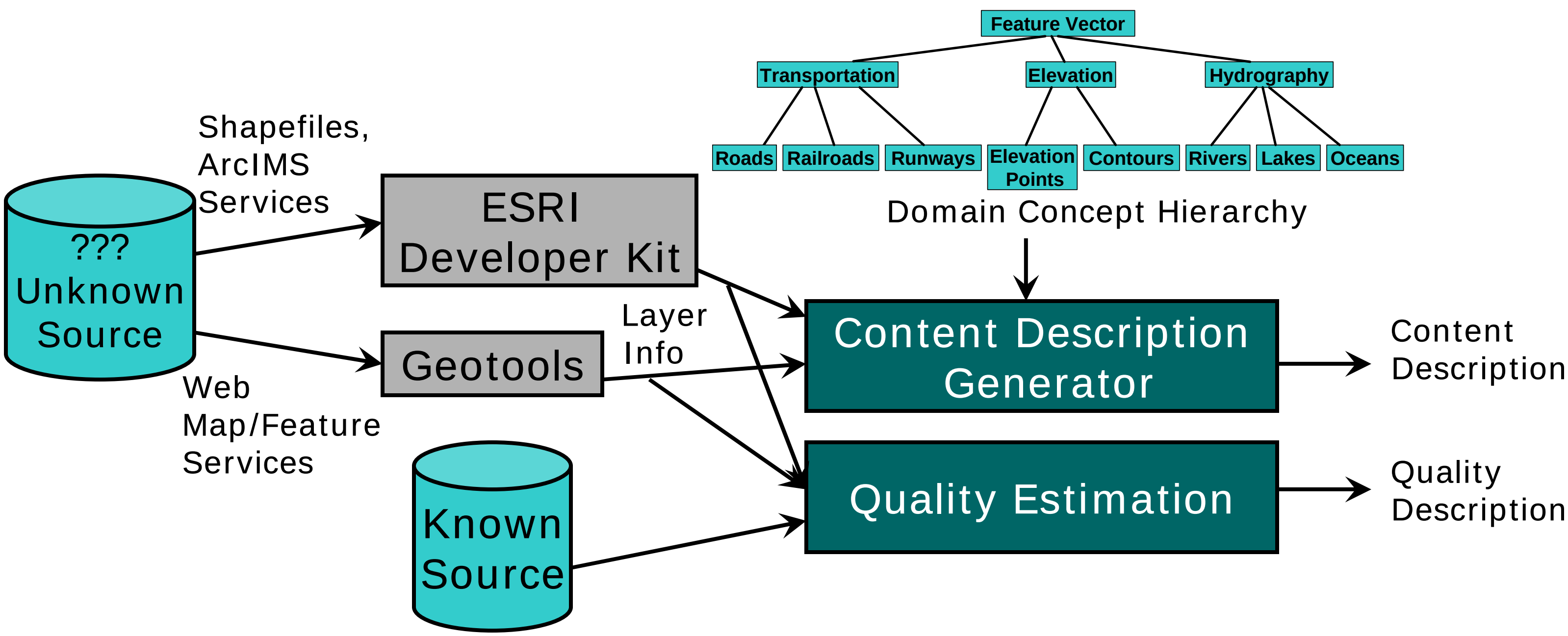
```
Q3(imageobj, vectorobj, resdiff, datediff):-
   Q3Data(itype, isource, vtype, vsource, imageobj, vectorobj) ^
   Q3Quality(itype, isource, vtype, vsource, resdiff, datediff)

Q3Data(itype, isource, vtype, vsource, imageobj, vectorobj):-
   Roads(vtype, vformat, cs, bbox, vsource, vectorobj) ^
   SatelliteImage(itype, iformat, size, resolution, cs, bbox, isource,
rasterobj) ^
   bbox coveredby `[[33,-115],[34,-116]]' ^
   size = `[400,400]' ^
   cs = `EPSG:4326'

Q3Quality(itype, isource, vtype, vsource, resdiff, datediff):-
   RoadQuality(vsource, vtype, vres, vdate, horiz-acc, vert-acc,
       vectorsin-acc-bounds, attr-comp, completeness) ^
   SatelliteImageQuality(isource, itype, idate, ires, multispectral,
       completeness) ^
    Subtract(idate, vdate, datediff) ^
    Subtract(ires, vres, resdiff) ^
    Pack(datediff, resdiff, date-res-diff) ^
    SkylineMin(date-res-diff, skylineresultrel) ^
    Unpack(skylineresultrel, smindatediff, sminresdiff) ^
    smindatediff = datediff ^
    sminresdiff = resdiff
```

Outline

- Introduction & motivation
- Quality-driven Geospatial Mediator (QGM)
  - Representing content and quality
  - Automatic source description generation
    - Content
    - Quality
  - Quality-driven query answering
  - Plan execution
- Related work
- Conclusions & future work

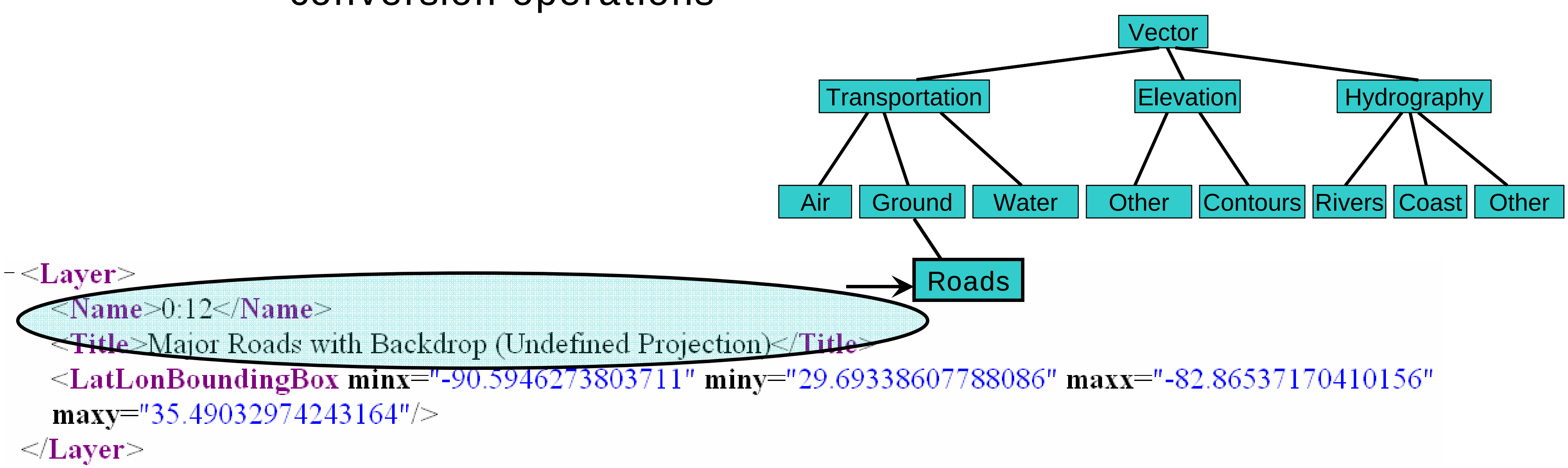
Automatic Source Description Generation

- Idea: Utilize well-known formats, existing standards, and information from existing sources to automatically generate description of new source



Content Description Generator

- Match domain concepts with source layers
  - Create tokens from names of layers and titles/descriptions
  - Use Dice similarity [Rijsbergen79]
- Coverage
  - Use the coverage information from the capabilities file
  - Address different coordinate systems by using coordinate conversion operations



Experimental Results: Content Description Generator

- Tested on 1248 real-world sources
- Used QGM to find matching domain concepts
- Ground truth by manually matching domain concepts with sources
  - Using name, title, and actual data returned by sources

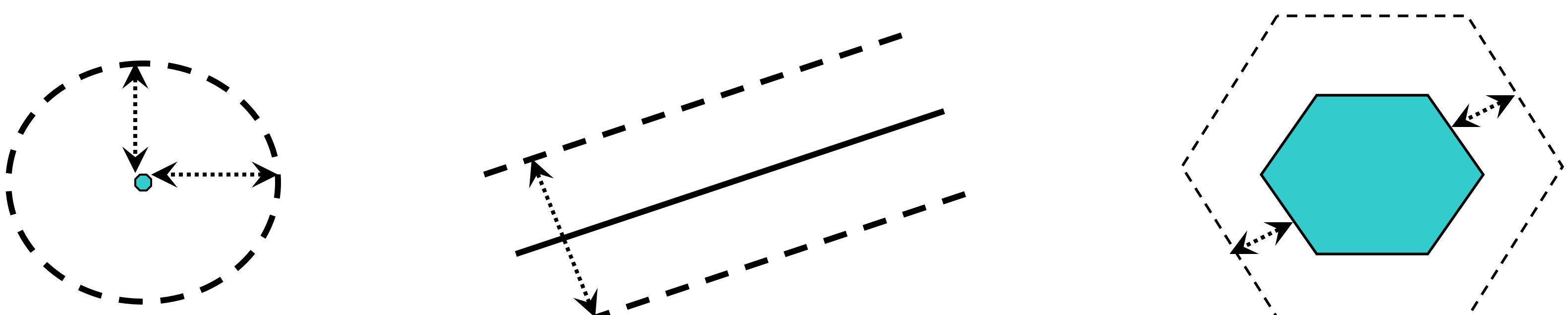
Layer	Precision	Recall	F-measure
Roads	98.02	95.84	96.92
Orthophoto	82.44	64.67	72.48
Raster	70.63	84.87	77.10
Vector	71.62	92.98	80.92
Rivers	94.55	98.11	96.30
Cart Tracks	100.00	97.83	98.90
Bridge/Overpasses	100.00	91.49	95.56
Ramp Lines	95.45	91.30	93.33
Topographic Maps	95.35	95.35	95.35
Airports	97.30	100.00	98.63
Schools	70.73	93.55	80.56
Administrative Areas	71.05	100.00	83.08
Counties	87.50	95.45	91.30
Flood Zones	100.00	100.00	100.00
Census Blocks	84.21	88.89	86.49
Totals	**88.21**	**89.66**	**88.93**

Outline

- Introduction & motivation
- Quality-driven Geospatial Mediator (QGM)
  - Representing content and quality
  - Automatic source description generation
    - Content
    - Quality
  - Quality-driven query answering
  - Plan execution
- Related work
- Conclusions & future work

Estimating Vector Quality

- Sample data from known and new source
- Compute value for completeness and positional accuracy attributes
  - Completeness
    - #features$_{(new)}$ * completeness$_{(known)}$ / # features$_{(known)}$
  - Accuracy bounds
    - Use accuracy bounds of the known sources
  - Features within accuracy bounds
    - # of features that fall within accuracy bounds/ # features



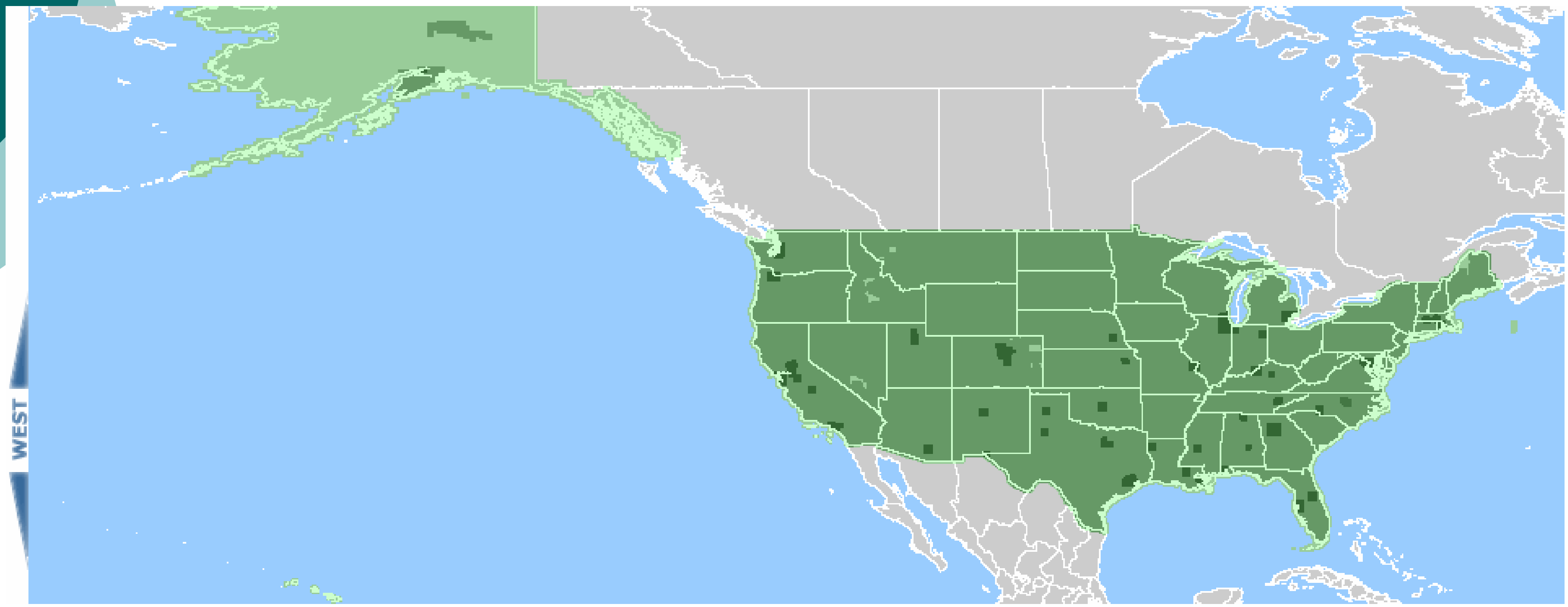
Experimental Results: Vector Quality Completeness Estimation

Type	Sample Size %	# of Layers	Avg. Completeness With 100% Sampling	% Error with Sampling		
				Diag.	Center	Column
Points	5	93	91.76	17.22	18.67	17.86
Points	10	93	91.76	17.54	21.53	15.19
Points	20	93	91.76	14.27	18.85	13.20
Points	25	93	91.76	13.68	16.94	12.04
Polylines	5	297	38.09	30.18	32.42	26.65
Polylines	10	297	38.09	24.69	29.84	24.71
Polylines	20	297	38.09	20.74	29.57	18.20
Polylines	25	297	38.09	19.68	28.58	17.94
Polygons	5	12	68.12	19.54	28.01	24.70
Polygons	10	12	68.12	25.61	28.53	24.19
Polygons	20	12	68.12	24.97	27.75	24.26
Polygons	25	12	68.12	23.68	27.47	23.14

Experimental Results: Vector Accuracy Estimation

Type	Sample Size%	# of Layers	Avg. Vec. in Bounds With 100% Sampling	% Error with Sampling		
				Diag.	Center	Column
Points	5	93	95.6	12.27	8.69	11.71
Points	10	93	95.6	9.83	8.27	8.96
Points	20	93	95.6	7.95	7.20	7.18
Points	25	93	95.6	7.71	7.14	7.23
Polylines	5	297	80.28	9.8	8.14	8.63
Polylines	10	297	80.28	8.68	7.73	6.81
Polylines	20	297	80.28	8.95	8.50	6.98
Polylines	25	297	80.28	8.67	8.26	6.84
Polygons	5	12	82.19	10.63	10.28	10.53
Polygons	10	12	82.19	9.81	11.36	9.68
Polygons	20	12	82.19	10.12	9.64	9.41
Polygons	25	12	82.19	9.97	9.83	9.43

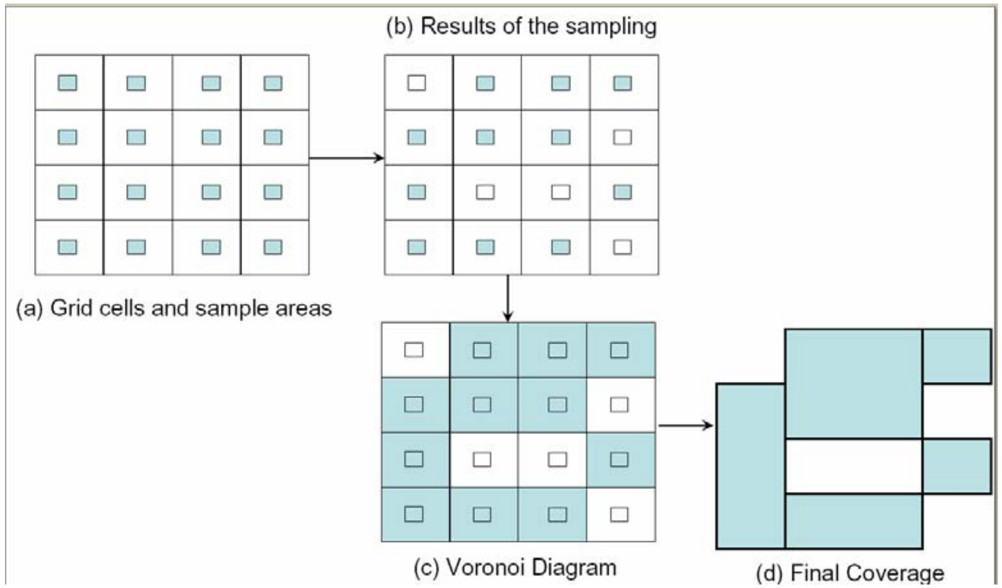
Raster Quality Estimation: Overstated Coverage



- Water no coverage
- Land no coverage
- B/W Satellite Image Only
- Topo maps & B/W Satellite Image
- Multi-spectral Satellite Image

Estimating Raster Coverage & Completeness

- Address the problem of sources overstating coverage
- Sample data from a source
- Use the sampling results and Voronoi diagram
- Estimate accurate coverage and completeness



Experimental Results: Raster Coverage Estimation

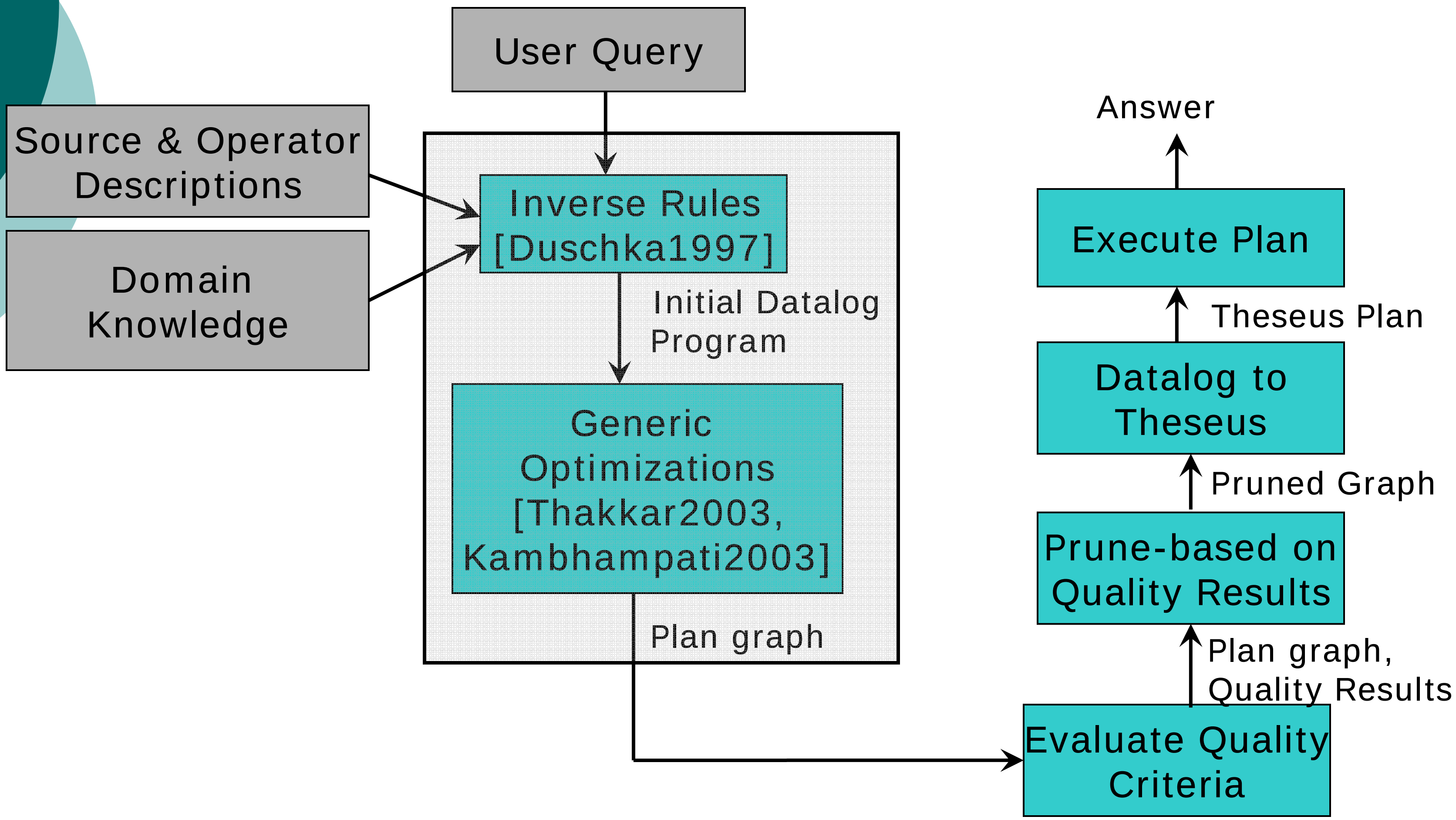
- Automatic estimation of Raster Quality
  - 60 queries with resolutions 1,5,10,50 m/p
  - Compare reported coverage with estimated coverage by sampling
  - Estimated coverage
    - loses some images (lower recall)
    - returns fewer empty images (higher precision)

Resolution	Reported Coverage			Estimated Coverage		
1 meter/pixel	Precision	Recall	F-measure	Precision	Recall	F-measure
1	72.15	100.00	83.82	94.12	84.21	88.89
5	82.43	100.00	90.37	91.38	86.89	89.08
10	81.82	100.00	90.00	92.86	86.67	89.66
15	81.58	100.00	89.86	90.00	87.10	88.52
Total	79.23	100.00	88.41	91.94	86.22	88.99

Outline

- Introduction & motivation
- Quality-driven Geospatial Mediator (QGM)
  - Representing content and quality
  - Automatic source description generation
    - Content
    - Quality
  - Quality-driven query answering
  - Plan execution
- Related work
- Conclusions & future work

QGM's Query Answering



Sample Query 3

- Find satellite image and road vector data covering `[[33,-116],[34,-117]]' such that both the resolution and date differences are minimized
  - Q3(imageobj, vectorobj, resdiff, datediff):-
    Q3Data(itype, isource, vtype, vsource, imageobj, vectorobj) ^
    Q3Quality(itype, isource, vtype, vsource, resdiff, datediff)

    Q3Data(itype, isource, vtype, vsource, imageobj, vectorobj):-
    Roads(vtype, vformat, cs, bbox, vsource, vectorobj) ^
    SatelliteImage(itype, iformat, size, resolution, cs, bbox, isource, rasterobj) ^
    bbox coveredby `[[33,-115],[34,-116]]' ^
    size = `[400,400]' ^
    cs = `EPSG:4326'

    Q3Quality(itype, isource, vtype, vsource, resdiff, datediff):-
    RoadQuality(vsource, vtype, vres, vdate, horiz-acc, vert-acc,
    vectorsin-acc-bounds, attr-comp, completeness) ^
    SatelliteImageQuality(isource, itype, idate, ires, multispectral,
    completeness) ^
    Subtract(idate, vdate, datediff) ^
    Subtract(ires, vres, resdiff) ^
    Pack(datediff, resdiff, date-res-diff) ^
    SkylineMin(date-res-diff, skylineresultrel) ^
    Unpack(skylineresultrel, smindatediff, sminresdiff) ^
    smindatediff = datediff ^
    sminresdiff = resdiff

Inverse Rules [Duschka 1997]

- Determine how to query domain relations
- Invert the source descriptions
- In the example query
  - Definition of Roads & SatelliteImage as views over sources
  - Definition of RoadQuality and SatelliteImageQuality as views over source quality

```
NavteqRoads(bbox, vectorobj):-
   Roads(type, format, cs,
       bbox, source, vectorobj)^
   bbox coveredby
        `[[33,-116],[34,-117]]'^
   source = `Navteq'^
   type = `Roads'^
   format = `Shapefile'^
   cs = `EPSG:4326'
```

```
Roads(`Roads', `Shapefile',
   `EPSG:4326', bbox, `Navteq',
   vectorobj):-
  NavteqRoads(bbox, vectorobj)^
  bbox coveredby
        `[[33,-116],[34,-117]]'
```

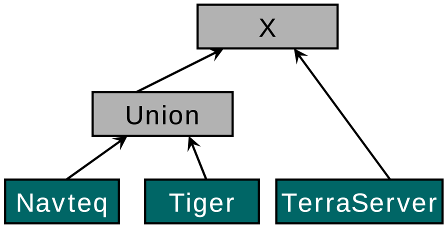
Datalog Program Generation

- Identify Relevant Rules
  - Extension: Check geospatial constraints
  - Find sources that
    - Appear in definition of relevant domain concepts
    - Do not have conflicting coverage constraints
  - In the example query
    - Find sources that appear in definition of Roads or SatelliteImage
    - Have coverage intersecting with Query's bounding box

```
Roads(`Roads', `Shapefile',
   `EPSG:4326', bbox, `Navteq',
   vectorobj):-
 NavteqRoads(bbox, vectorobj)^
 bbox coveredby
      `[[33,-116],[34,-117]]'

Roads(`Roads', `Shapefile',
   `EPSG:4326', bbox, `Navteq',
   vectorobj):-
 TigerRoads(bbox, vectorobj)^
 bbox coveredby
   `[[33,-116],[34,-117]]'

Parks(`Roads', `Shapefile',
   `EPSG:4326', bbox, `Navteq',
   vectorobj):-
 NGAParks(bbox, vectorobj)^
 bbox coveredby
   `[[33,-116],[34,-117]]'
```

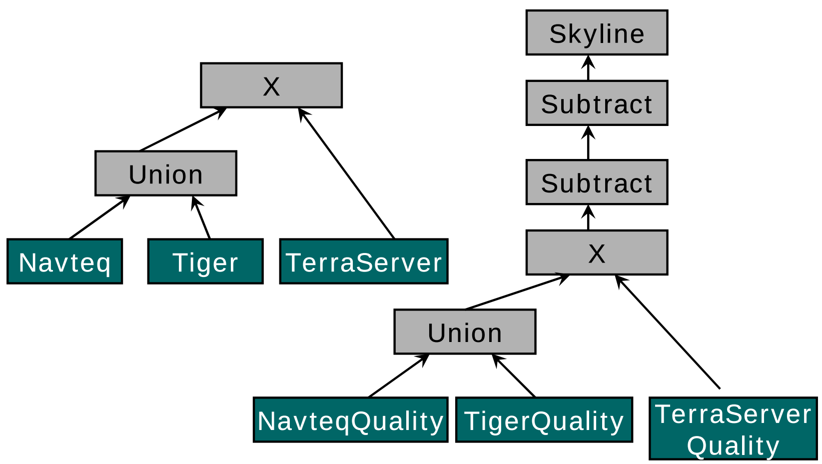
Generated Plan

- Two branches
  - Content
    - Has requests to data sources
    - Select operations to apply constraints
  - Quality
    - Has requests to obtain facts about quality of data for sources that appear in the content plan
    - May have requests to mathematical, aggregate, or skyline operations
  - In our example query
    - Assume two relevant vector sources
    - Assume one image source



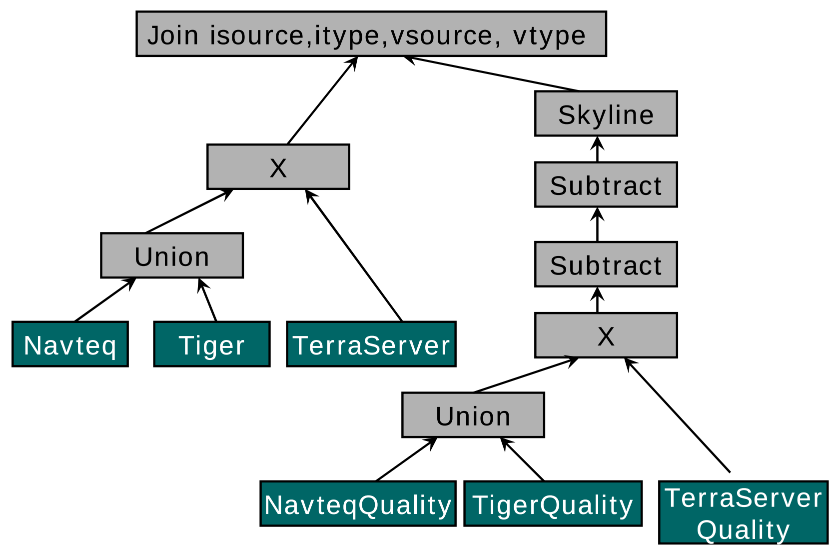
Generated Plan

- Two branches
  - Content
    - Has requests to data sources
    - Select operations to apply constraints
  - Quality
    - Has requests to obtain facts about quality of data for sources that appear in the content plan
    - May have requests to mathematical, aggregate, or skyline operations
  - In our example query
    - Assume two relevant vector sources
    - Assume one image source

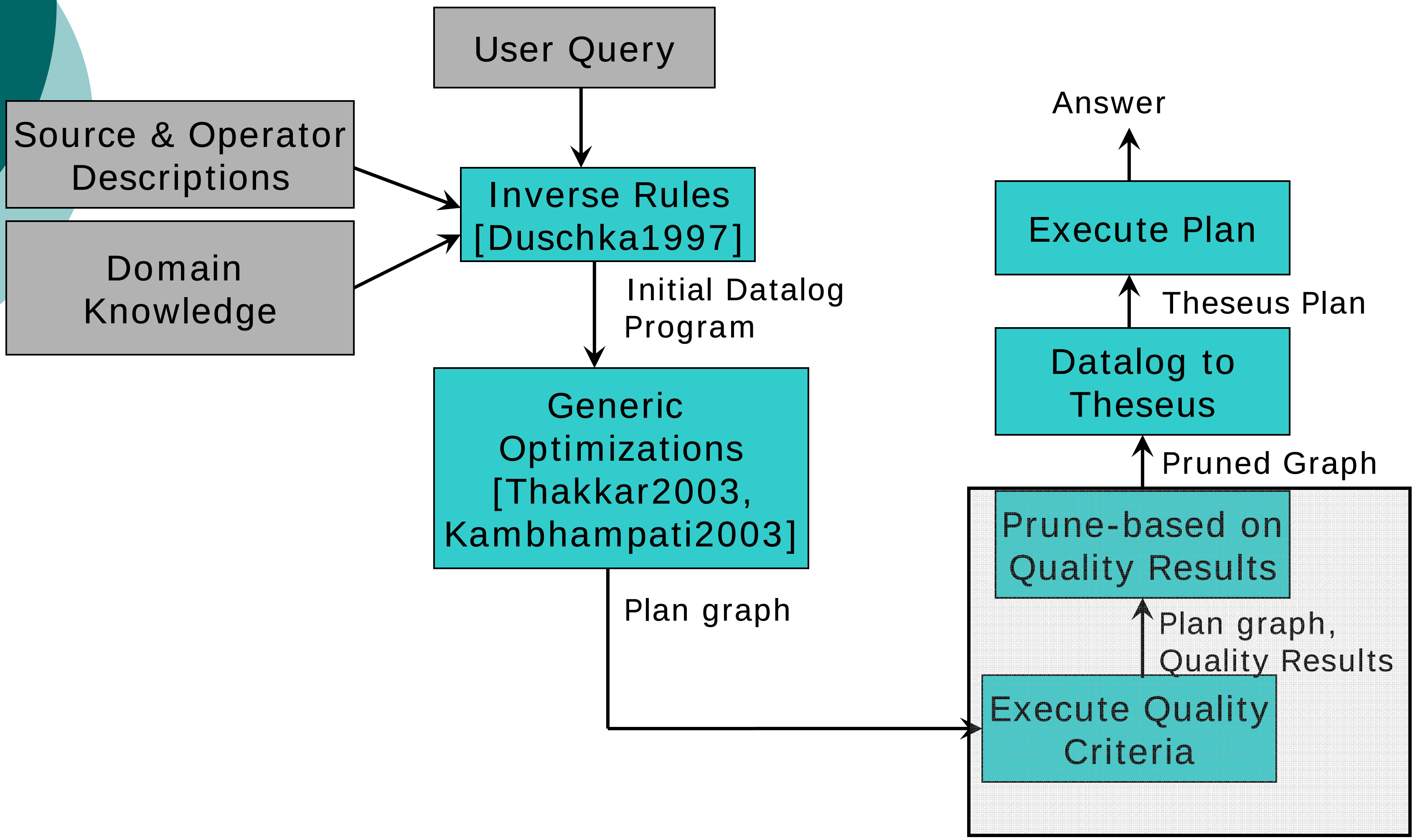


Generated Plan

- Two branches
  - Content
    - Has requests to data sources
    - Select operations to apply constraints
  - Quality
    - Has requests to obtain facts about quality of data for sources that appear in the content plan
    - May have requests to mathematical, aggregate, or skyline operations
  - In our example query
    - Assume two relevant vector sources
    - Assume one image source

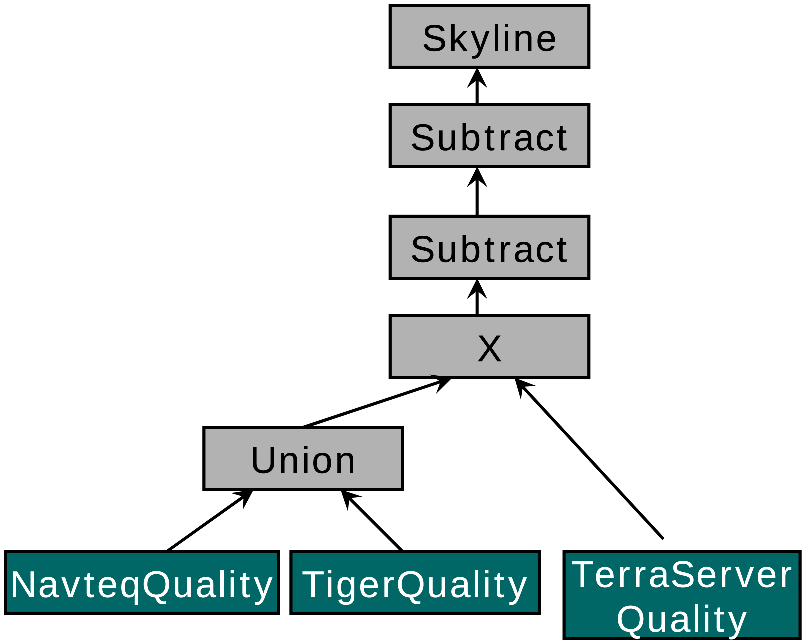


QGM's Query Answering



Executing Quality Criteria

- Obtain Quality facts
- Apply necessary relational, mathematical, or aggregate operations
- Apply constraints and/or skyline operations
- Resulting tuples include source name and type for each type of data and any other attributes requested in quality query

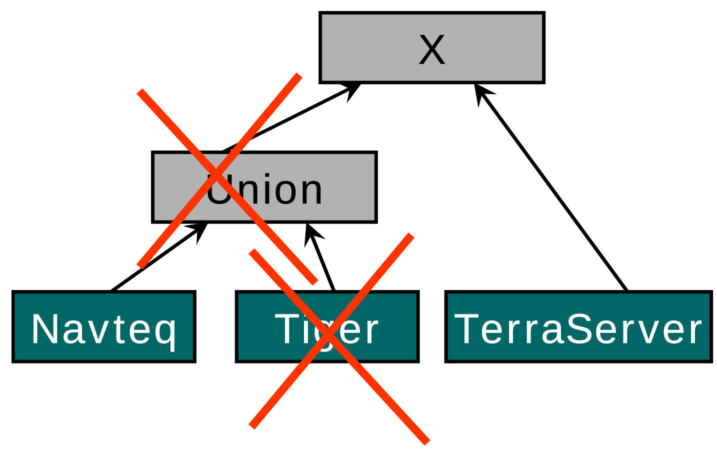


Prune based on Quality Results

- Remove all sources that did not satisfy quality criteria
  - If a source(S1) has completeness 20% and quality criteria is completeness > 50%
  - Remove source (S1) from the content subtree
- Check join constraints in the graph connected to quality subtree
  - Remove branches that do not produce tuples

Combin ation	Resdiff (m/p)	Datediff (days)
Navteq & TS	5	365
Tiger & TS	12	1825

Quality Statstics



Execute Final Plan

- QGM converts the plan to Theseus
  - Streaming, dataflow-style execution
- QGM also generates plans to access
  - Shapefiles
  - ArcIMS services
  - Web Map Servers
  - Web services
  - Databases

Experimental Evaluation

- Setup
  - Dual Xeon processor, 3 GB memory
    - Actual use: half processor, 1GB memory
  - Data sources
    - Real-world shapefiles, ArcIMS services, and Web Map Services
- Method
  - Compare with Prometheus
    - Data integration system that supports geospatial data without any quality information
  - Compare
    - Quality
    - Response time

Query Answering: Quality of Answers

- Query answering
  - Quality
    - One standard deviation better in completeness for most queries
    - Half standard deviation better for accuracy

Type	QGM		Average		Std. Deviation	
	% Comp.	% Acc.	% Comp.	% Acc.	% Comp.	% Acc.
Constraint	59.81	87.61	47.71	83.12	17.36	9.31
Aggregate	68.19	89.97	47.71	83.12	17.36	9.31
Skyline	64.03	87.90	47.71	83.12	17.36	9.31

Query Answering: Response Time

Query	# of Sources	Prometheus Gen.	Prometheus Opt.	Prometheus Exec.	Prometheus Total	Prometheus # results	QGM Gen.	QGM Opt.	QGM Exec.	QGM Total	QGM # results
Constraint	0-5	32	98	126	256	3.7	32	109	113	254	3.2
Constraint	5-10	33	119	279	431	7.9	33	116	196	345	5.8
Constraint	10-20	32	131	872	1035	16.1	32	138	524	694	11.2
Constraint	20-30	34	168	1985	2187	24.3	34	159	871	1064	17.6
Aggregate	0-5	32	98	126	256	3.7	32	113	102	247	1.3
Aggregate	5-10	33	119	279	431	7.9	33	116	115	264	2.1
Aggregate	10-20	32	131	872	1035	16.1	32	137	167	336	3.7
Aggregate	20-30	34	168	1985	2187	24.3	34	162	190	386	4.1
Skyline	0-5	32	98	126	256	3.7	32	140	134	306	2.9
Skyline	5-10	33	119	279	431	7.9	33	192	184	409	4.6
Skyline	10-20	32	131	872	1035	16.1	32	297	372	701	7.2
Skyline	20-30	34	168	1985	2187	24.3	34	421	579	1034	9.8

Outline

- Introduction & motivation
- Quality-driven Geospatial Mediator (QGM)
  - Representing content and quality
  - Automatic source description generation
    - Content
    - Quality
  - Quality-driven query answering
  - Plan execution
- Related work
- Conclusions & future work

Related Work (1/2)

- Geospatial Data Integration
  - Hermes [Adali95], MIX [Gupta99], GeonGrid [Manipura03], VirGIS [Boucelma04]
    - Focus access methods and formats
    - GeonGrid also has some quality and ontology components
  - ODGIS [Fonesca 02], GSA [Arpinar 06], SWING [Klien06]
    - Creation ontology for geospatial data and matching data layers
- Quality-driven data integration
  - Biological data [Eckman06, Mihila05]
    - Focus is on completeness
  - General-purpose [Neumann01, Bleiholder2006, Scannapieco05]
    - Assign one quality score based on user-supplied weights
- QGM
  - A Geospatial mediation framework that supports quality
  - Automatic generation of descriptions
  - More expressive quality criteria specification

Related Work (2/2)

- OpenGIS & mapping systems
  - Standards, Web-based mapping, Desktop GIS
- Source modeling [Carman06]
  - Learn source descriptions by sampling data from a source
- Geospatial data quality
  - Conflation Operation [Saalfield 1993, Chen 2003]
  - Representation [Goodchild02-03]
  - Visualization [Worboys01]
- QGM
  - QGM utilizes the existing standards and well-known formats
  - QGM provides output using OpenGIS standards
  - Quality specification in QGM is flexible and can utilize existing specifications

Conclusion: Contributions

- A declarative specification of both the content and the quality of geospatial sources
- Algorithms to automatically generate source descriptions and estimate the quality of data provided by geospatial data sources
- A quality-driven query answering algorithm
- An approach to map the generated integration plans and source requests to a program that is efficiently executed by a streaming, dataflow-style execution engine.

Conclusion: Broader Implications

- Geospatial data integration framework that supports quality-driven integration
- QGM's query answering technique can be easily applied in other domains
- Quality estimation techniques can be utilized for automated quality estimation in many domains

Conclusion: Future Work

- Source discovery
  - Use terms from gazetteer to create keywords for searching geospatial data
  - Utilize the Web catalog service (OpenGIS standards)
- Automatic source description generation
  - Utilize token weights and transformation weights
- Quality estimation
  - Density-based sampling
  - Adaptive raster sampling
    - Sample at the edges of the cells
- Query answering
  - Utilize spatial database to prune based on coverage
  - Improve response time by parallel processing on multiple machines

Questions