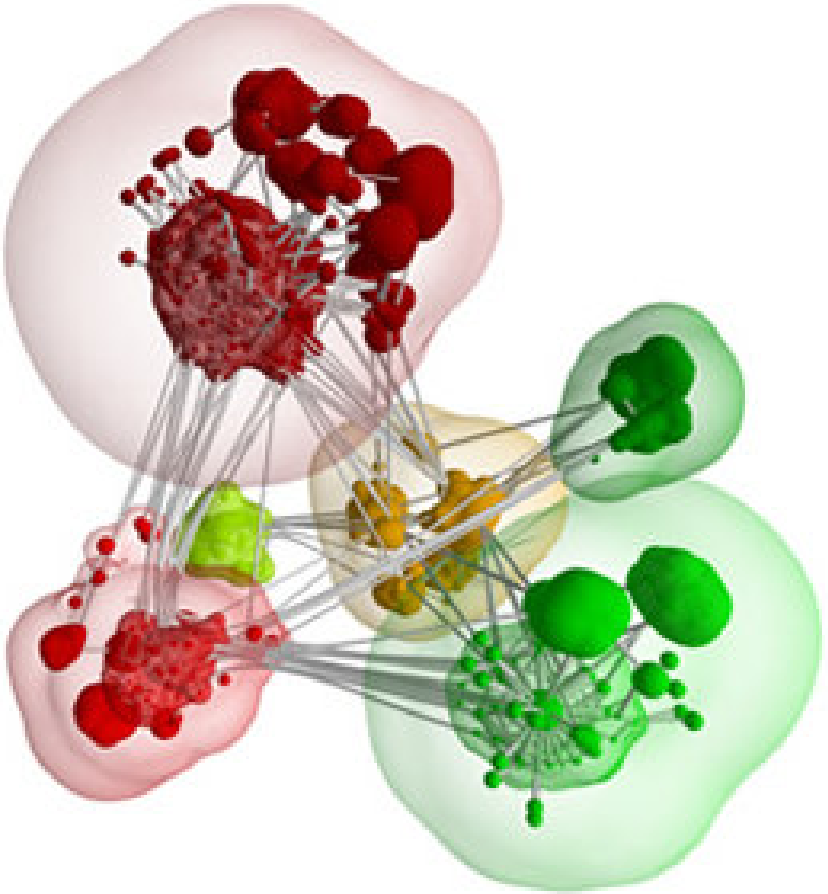
GeoSpatial with AllegroGraph



Jans Aasman, Ph.D.
CEO Franz Inc
Ja@Franz.com

Why the RDF community needs GeoTemporal reasoning capabilities

- Most of the Semantic Web projects are about
  - People and their relationships
  - Events that happen in space and time

- Reasoning about time and space needs to be an integral part of Semantic Web.

Why the geospatial community needs RDF

- RDF is about MetaData
- MetaData is usually very unstructured and sparse
- Objects that you describe are usually part of an object hierarchy
- You need ontologies to describe geospatial objects..
  - See how the national map project uses ontologies to solve the metadata problem

The National Map - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

http://nationalmap.gov/

Gmail GCAL Movies Sudoku Ttext Weather Amazon Twine This Wells Fargo FtoC nwa IMDB BB wheather Franz psf

GoToMeeting : Web conferencing, Onli... | The National Map

USGS science for a changing world

The National Map

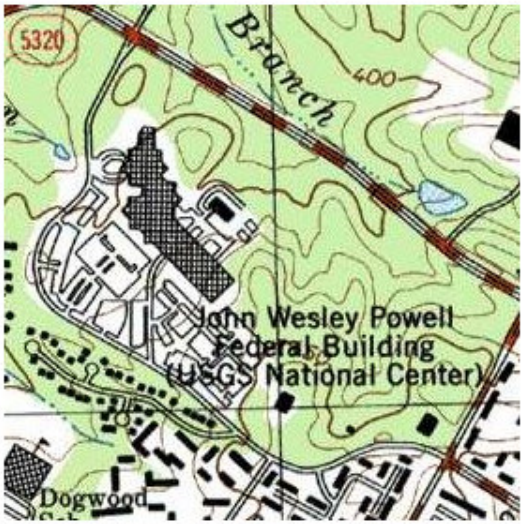
USGS Home
Contact USGS
Search USGS

The National Map

- *The National Map* Home
- Products and Services
- Partnerships
- Reading Room
- News & Events
- *The National Map* Corps
- National Geospatial Program

The National Map

As one of the cornerstones of the U.S. Geological Survey's (USGS) National Geospatial Program, *The National Map* is a collaborative effort among the USGS and other Federal, State, and local partners to improve and deliver topographic information for the Nation. It has many uses ranging from recreation to scientific analysis to emergency response. *The National Map* is easily accessible for display on the Web, as products and services, and as downloadable data. The geographic information available from *The National Map* includes orthoimagery (aerial photographs), elevation, geographic names, hydrography, boundaries, transportation, structures, and land cover. Other types of geographic information can be added within the viewer or brought in with *The National Map* data into a Geographic Information System to create specific types of maps or map views. *The National Map* is a significant contribution to the National Spatial Data Infrastructure (NSDI) and currently is being transformed to better serve the geospatial community by providing high quality, integrated geospatial data and improved products and services including new generation digital topographic maps.



In addition to being an important contribution to the NSDI, *The National Map* is foundational to implementation of the Department of the Interior (DOI) Geospatial Modernization Blueprint and meeting the DOI mission to protect America's treasures for future generations, provide access to our nation's natural and cultural heritage, offer recreation opportunities,

Done

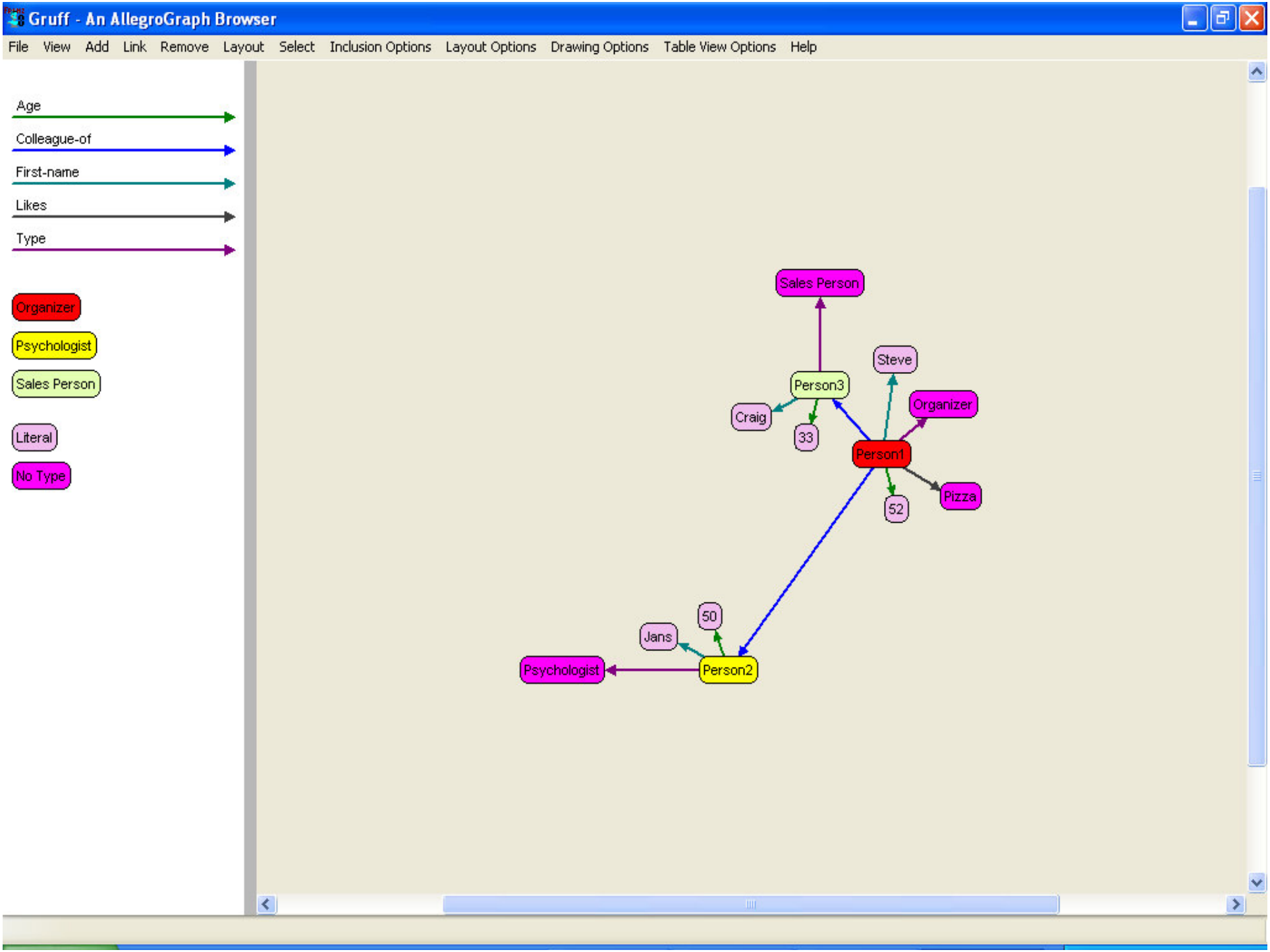
start 5:10 PM

This talk

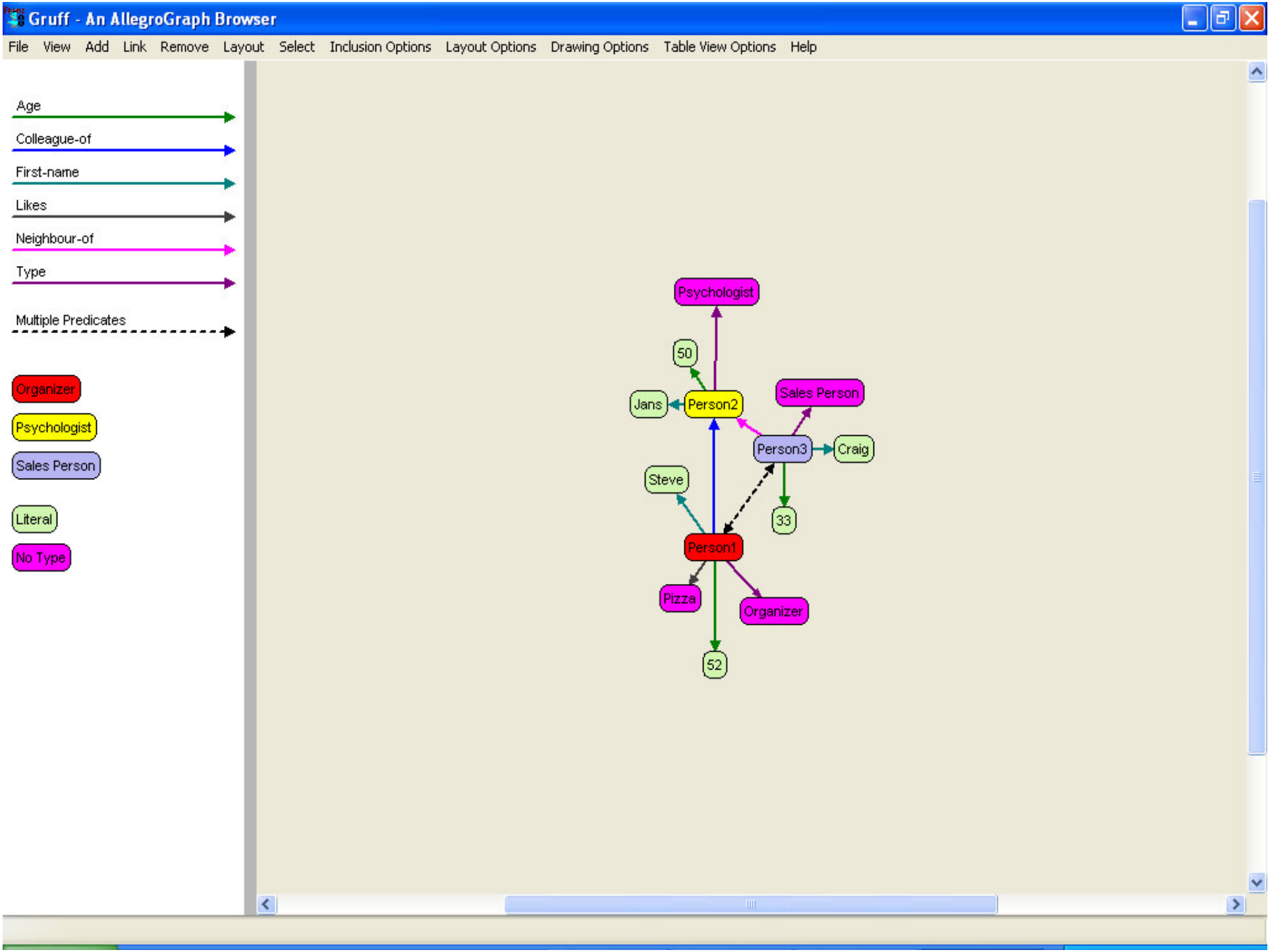
- The basics of a triple store
- Events everywhere
- One query language to combine
  - Geospatial, Temporal, Social Network Analysis
- Geospatial tutorial
- Example: a river network in GeoBC
- A future direction: moving objects

Graphs, triples, triple-store?

```
createTripleStore("seminar.db" )

addTriple (Person1 first-name Steve)
addTriple (Person1 isa  Organizer)
addTriple (Person1 age 52)
addTriple (Person2 first-name Jans)
addTriple (Person2 isa Psychologist)
addTriple (Person2 age 50)
addTriple (Person3 first-name Craig)
addTriple (Person3 isa SalesPerson)
addTriple (Person3 age 32)

addTriple (Person1 colleague-of Person2)
addTriple (Person1 colleague-of Person3)

addTriple (Person1 likes Pizza)
```


Gruff - An AllegroGraph Browser
File View Add Link Remove Layout Select Inclusion Options Layout Options Drawing Options Table View Options Help
Age
Colleague-of
First-name
Likes
Type
Organizer
Psychologist
Sales Person
Literal
No Type
Sales Person
Steve
Person3
Organizer
Craig
33
Person1
Pizza
52
50
Jans
Psychologist
Person2


```
addTriple ( Person3 neighbour-of Person1)
addTriple ( Person3 neighbour-of Person2)
```


Gruff - An AllegroGraph Browser
File View Add Link Remove Layout Select Inclusion Options Layout Options Drawing Options Table View Options Help
Age
Colleague-of
First-name
Likes
Neighbour-of
Type
Multiple Predicates
Organizer
Psychologist
Sales Person
Literal
No Type
Psychologist
50
Jans
Person2
Sales Person
Person3
Craig
Steve
33
Person1
Pizza
Organizer
52

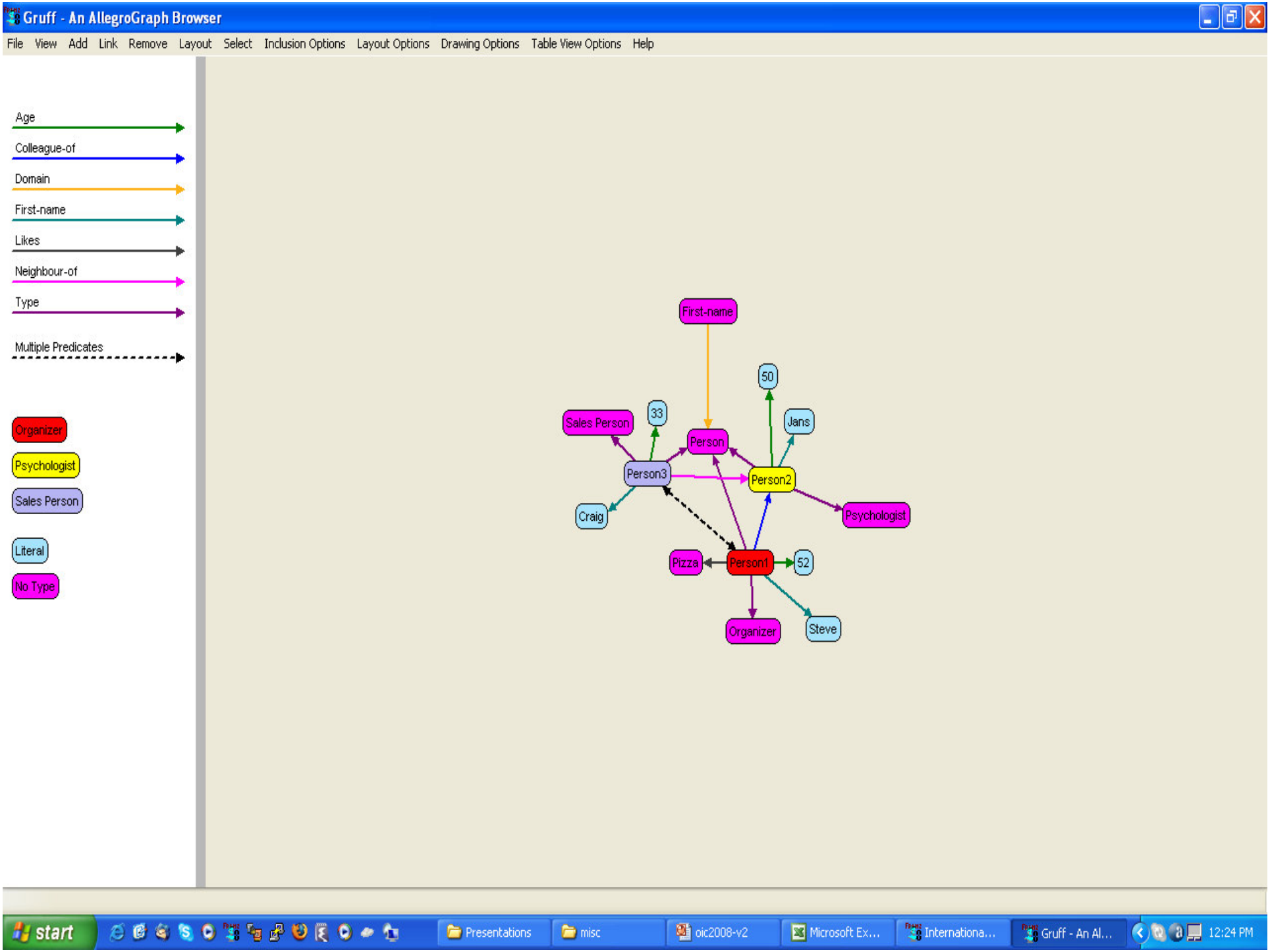
And now you can query

```
(select (?xname ?yname)
  (?x colleague-of ?y)
  (not (?y neighbour-of ?x))
  (?x first-name ?xname)
  (?y first-name ?yname))
```

Or reason

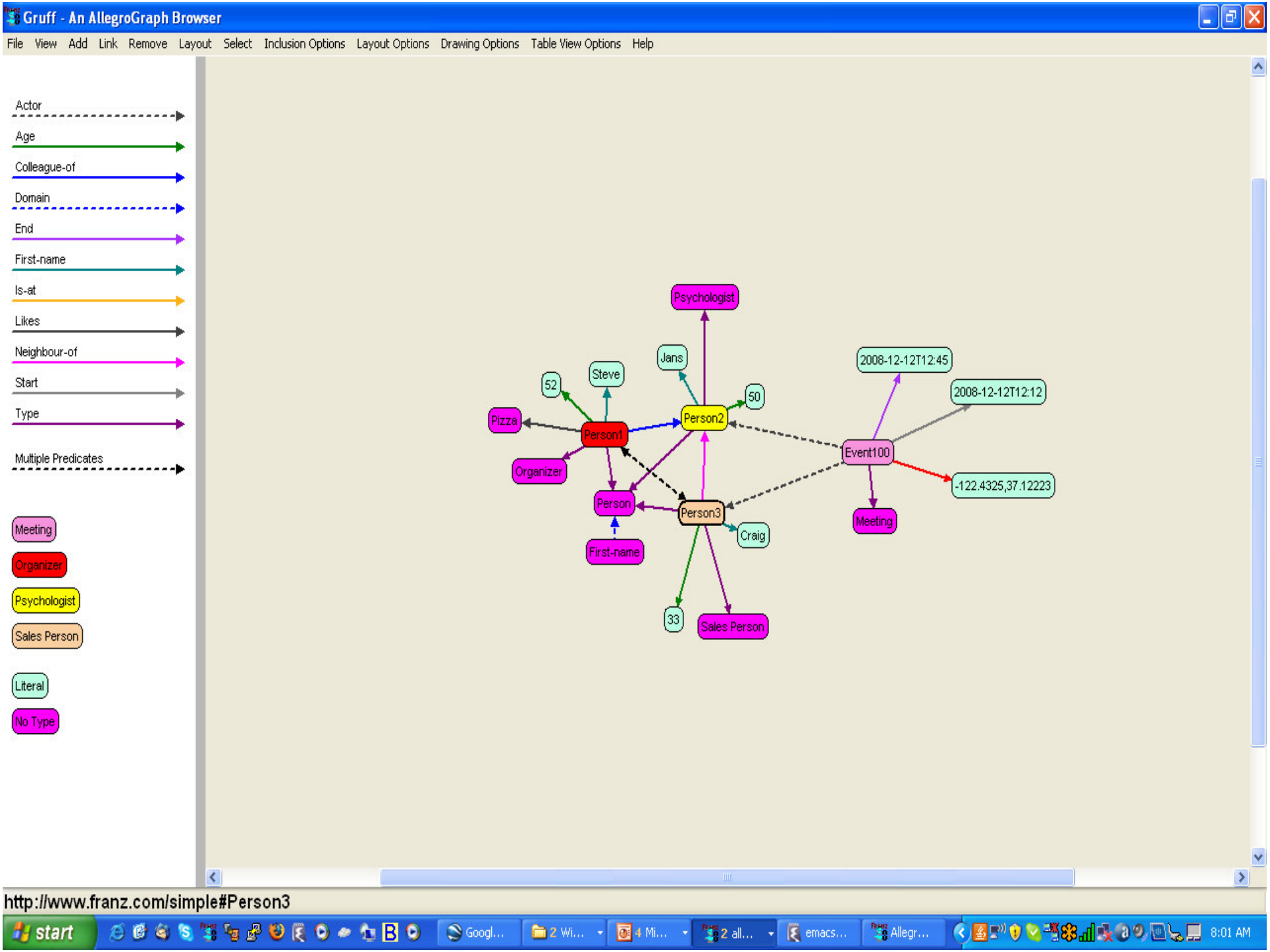
```
addTriple ( first-name domain Person)
```

Every subject that has a predicate 'first-name' must be of type Person.


Gruff - An AllegroGraph Browser
File View Add Link Remove Layout Select Inclusion Options Layout Options Drawing Options Table View Options Help
Age
Colleague-of
Domain
First-name
Likes
Neighbour-of
Type
Multiple Predicates
Organizer
Psychologist
Sales Person
Literal
No Type
First-name
50
Sales Person
33
Jans
Person
Person3
Person2
Craig
Psychologist
Pizza
Person1
52
Organizer
Steve
start
Presentations
misc
oic2008-v2
Microsoft Ex...
Internationa...
Gruff - An Al...
12:24 PM

And deal with events

```
addTriple (Event100 type Meeting)

addTriple (Event100 actor Person3)
addTriple (Event100 actor Person2)

addTriple (Event100 start 2008-12-12T12:12)
addTriple (Event100 end 2008-12-12T12:45)

addTriple (Event100 is-at -122.4325,37.12223)
```

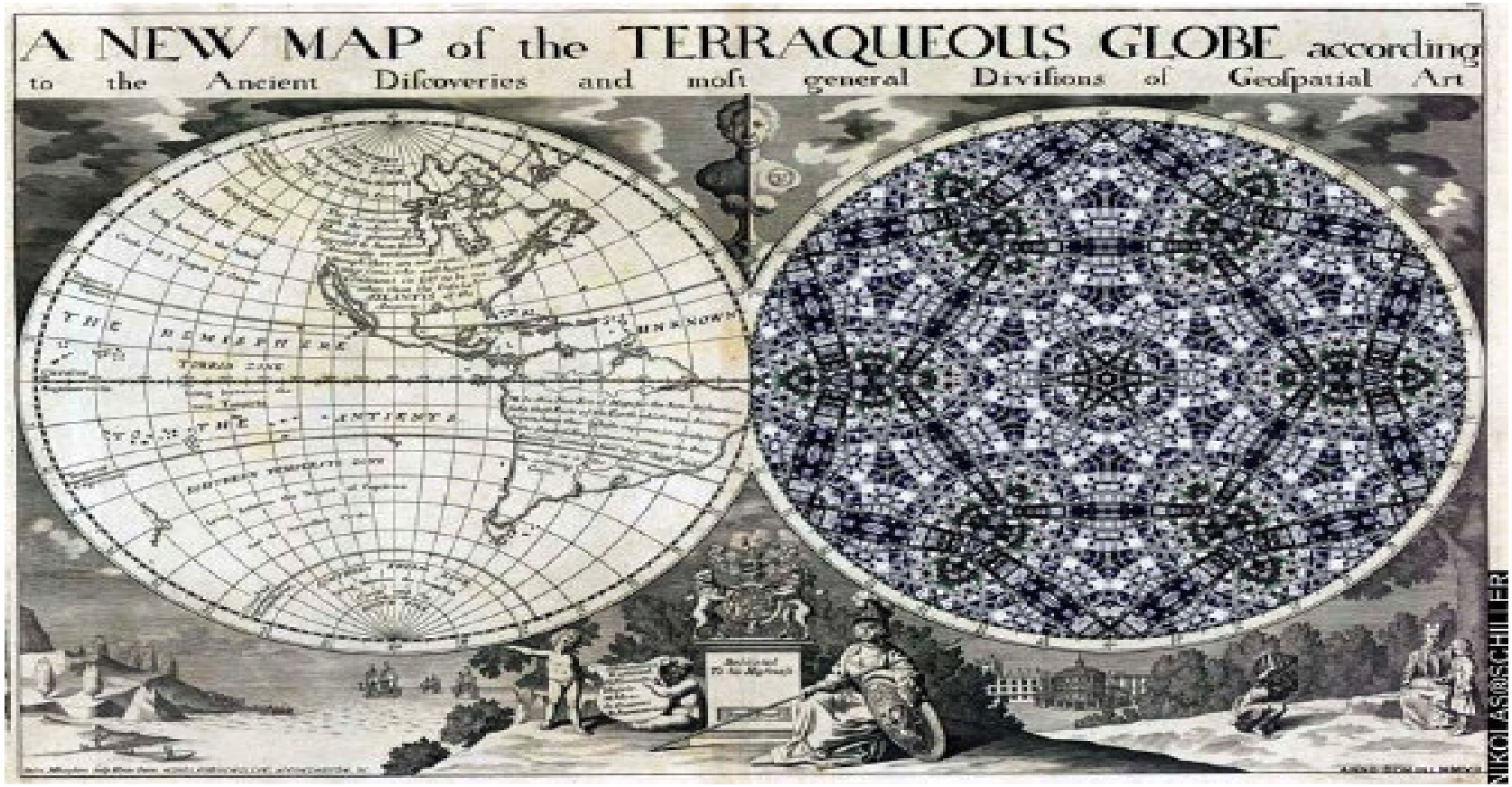

Gruff - An AllegroGraph Browser
File View Add Link Remove Layout Select Inclusion Options Layout Options Drawing Options Table View Options Help
Actor
Age
Colleague-of
Domain
End
First-name
Is-at
Likes
Neighbour-of
Start
Type
Multiple Predicates
Meeting
Organizer
Psychologist
Sales Person
Literal
No Type
Psychologist
Jans
Steve
52
50
Pizza
Person1
Person2
Organizer
Person
Person3
Craig
First-name
33
Sales Person
2008-12-12T12:45
2008-12-12T12:12
Event100
-122.4325,37.12223
Meeting
http://www.franz.com/simple#Person3
start
Googl...
2 Wi...
4 Mi...
2 all...
emacs...
Allegr...
8:01 AM

Integrated in select language.

Find a meetings that happened in July within 5 miles of Berkeley that was attended by the most important person in Jans' friends and friends of friends.

```
(select (?x)
   (ego-group !person:jans knows ?group 2)
   (actor-centrality-members ?group knows ?x ?num)
   (q ?event !fr:actor ?x)
   (qs ?event !rdf:type !fr:Meeting)
   (interval-during ?event "2009-07-01" "2009-07-25")
   (geo-box-around !geoname:Berkeley ?event 5 miles)
   !)
```

Geospatial Reasoning

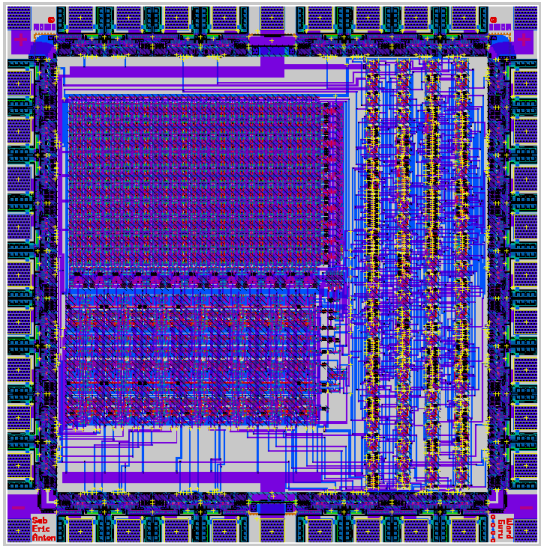


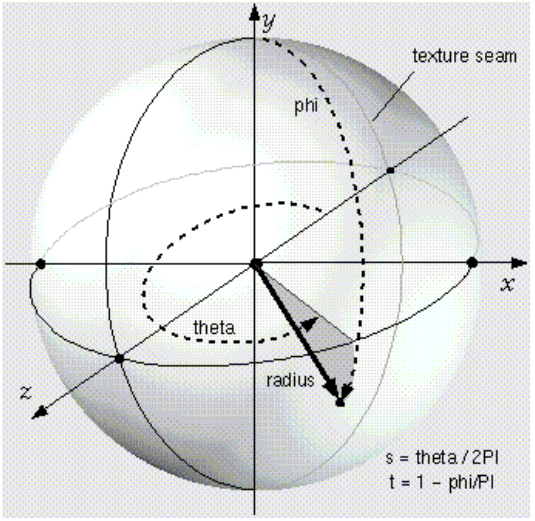
Geospatial Challenge

- Make the following super efficient
  - Where did something happen?
  - How far was event1 from event2?
  - Find all the events that occurred in a bounding box or radius of M miles?
  - Do these two shapes overlap?
  - Find all the objects in the intersection of two shapes
- On a very large scale
  - when things don't fit in memory
  - millions of events and polygons

Scale

- We work both in a flat plane (x, y) and on a sphere (lat, lon)
- We work on different scales : From ångström to miles





The law of haversines [edit]

Given a unit sphere, a "triangle" on the surface of the sphere is defined by the great circles connecting three points **u**, **v**, and **w** on the sphere. If the lengths of these three sides are *a* (from **u** to **v**), *b* (from **u** to **w**), and *c* (from **v** to **w**), and the angle of the corner opposite *c* is *C*, then the law of haversines states:

(the law of haversines)

$$\operatorname{haversin}(c) = \operatorname{haversin}(a - b) + \sin(a)\sin(b)\operatorname{haversin}(C)$$

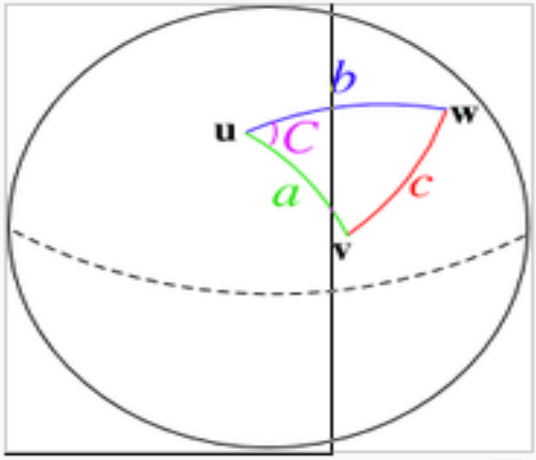
Since this is a unit sphere, the lengths *a*, *b*, and *c* are simply equal to the angles (in radians) subtended by those sides from the center of the sphere (for a non-unit sphere, each of these arc lengths is equal to its central angle multiplied by the radius of the sphere).

In order to obtain the haversine formula of the previous section from this law, one simply considers the special case where **u** is the ~~north pole~~, while **v** and **w** are the two points whose separation *d* is to be determined. In that case, *a* and *b* are $\pi/2 - \varphi_{1,2}$ (i.e., 90° − latitude), *C* is the longitude separation $\Delta\lambda$, and *c* is the desired *d*/*R*. Noting that $\sin(\pi/2 - \varphi) = \cos(\varphi)$, the haversine formula immediately follows.

To derive the law of haversines, one starts with the spherical law of cosines:

$$\cos(c) = \cos(a)\cos(b) + \sin(a)\sin(b)\cos(C)$$

As mentioned above, this formula is an ill-conditioned way of solving for *c* when *c* is small. Instead, we substitute the identity that $\cos(\theta) = 1 - 2\operatorname{haversin}(\theta)$, and also employ the addition identity $\cos(a - b) = \cos(a)\cos(b) + \sin(a)\sin(b)$, to obtain the law of haversines, above.



Spherical triangle solved by the law of haversines.

(spherical law of cosines)

R-tree

From Wikipedia, the free encyclopedia

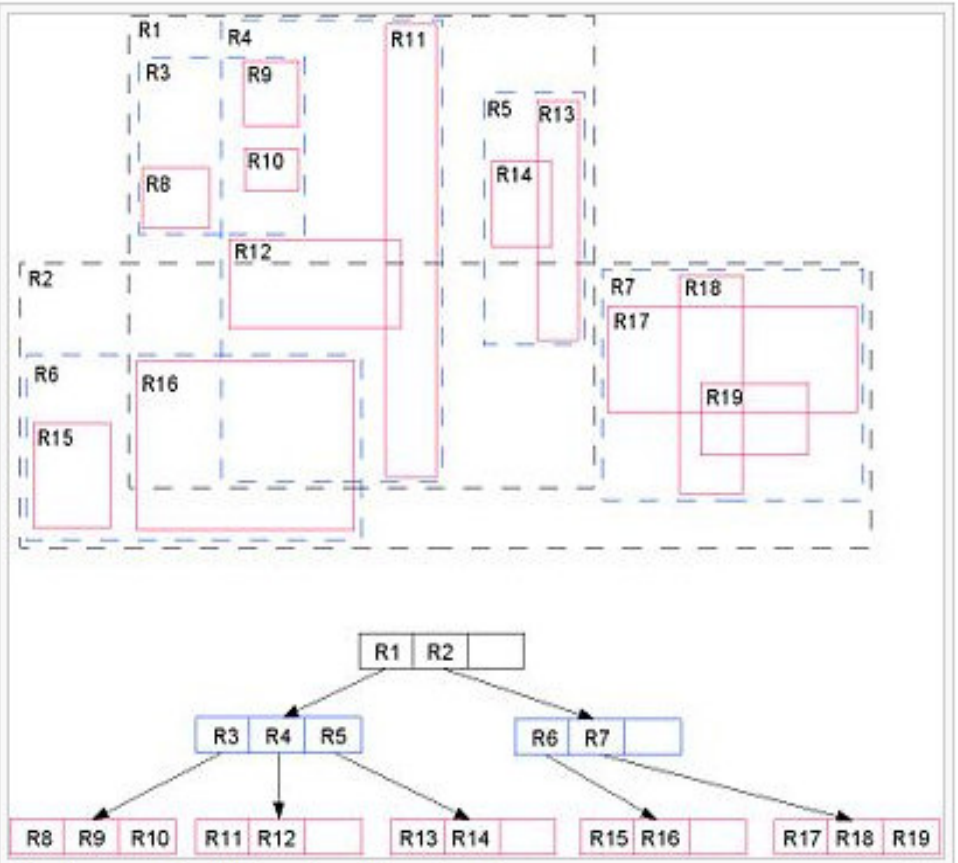
This article is about the data structure. For the type of metric space, see Real tree.

R-trees are tree data structures that are similar to B-trees, but are used for spatial access methods i.e., for indexing multi-dimensional information; for example, the (X, Y) coordinates of geographical data. A common real-world usage for an R-tree might be: "Find all museums within 2 miles of my current location".

The data structure splits space with hierarchically nested, and possibly overlapping, minimum bounding rectangles (otherwise known as bounding boxes).

Each node of an R-tree has a variable number of entries (up to some pre-defined maximum). Each entry within a non-leaf node stores two pieces of data: a way of identifying a child node, and the bounding box of all entries within this child node.

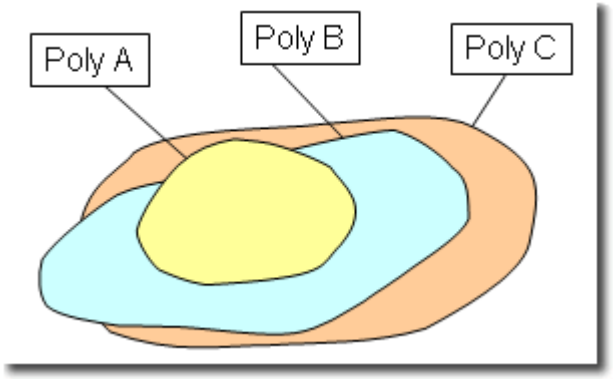
The insertion and deletion algorithms use the bounding boxes from the nodes to ensure that "nearby" elements are placed in the same leaf node (in particular, a new element will go into the leaf node that requires the least enlargement in its bounding box). Each entry within a leaf node stores two pieces of information; a way of identifying the actual data element (which, alternatively, may be placed directly in the node), and the bounding box of the data element.

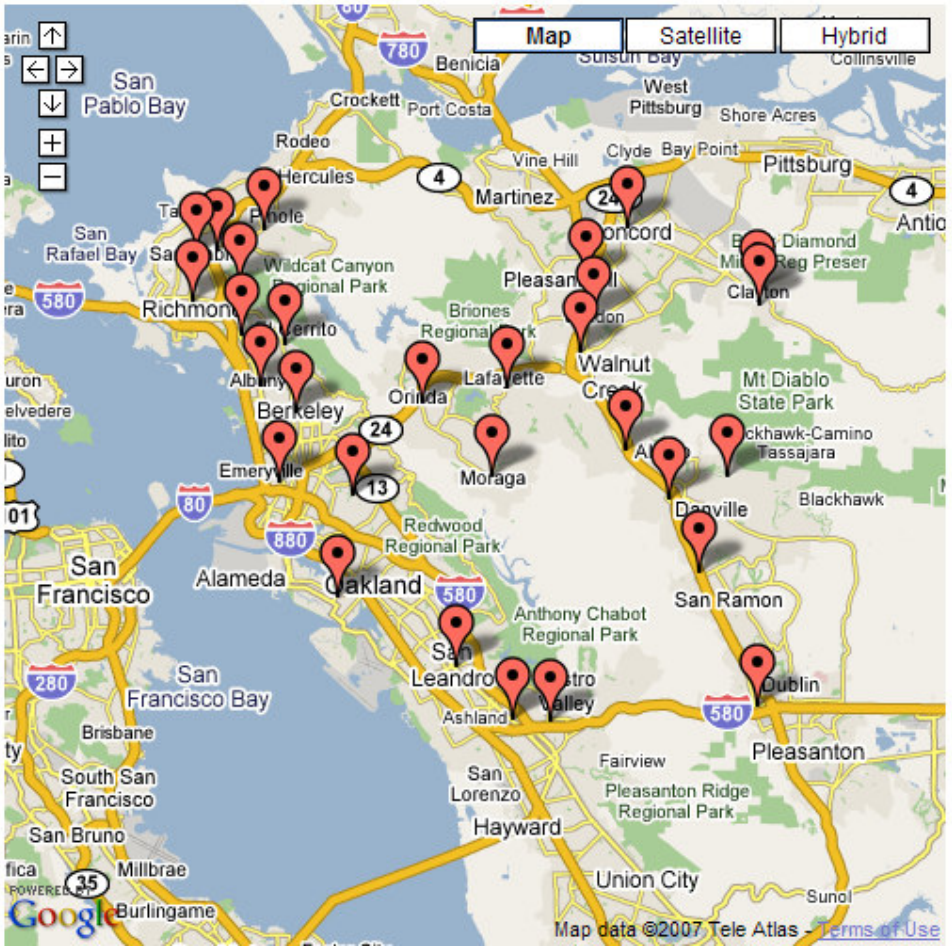


Simple example of R-tree for a 2D rectangles

Sample Geospatial Primitives

```
(geo-bounding-box ?x +minlat +maxlat +minlon +maxlon)
(geo-box-around +x ?y +miles)
(geo-distance +x +y ?dist)
(geo-radius-around +x ?y +miles)
(polygon-in ?p1 ?p2)
(polygon-touch ?p1 ?p2)
(polygon-overlap ?p1 ?p2)
Etc.
```





Now with SPARQL Support [1]

```
PREFIX fr:
        <http://franz.com/ns/allegrograph/3.0/geospatial/>
  PREFIX geo:          <http://www.geonames.org/ontology#>
  PREFIX country:      <http://www.geonames.org/Countries#>

  SELECT ?placename ?population WHERE {
    GEO OBJECT
    HAVERSINE ( ?londonpos, 50 MILES ) {
      ?place fr:pos ?pos ;
             geo:name ?placename ;
             geo:population ?population ;
             geo:countryCode ?cc
    }
    WHERE {
      # Select London, UK.
      ?london geo:name 'London' ;
              geo:countryCode 'GB' ;
              fr:pos ?londonpos .
    }
    FILTER (?population > 25000)
```

Benchmarking on GeoNames.rdf

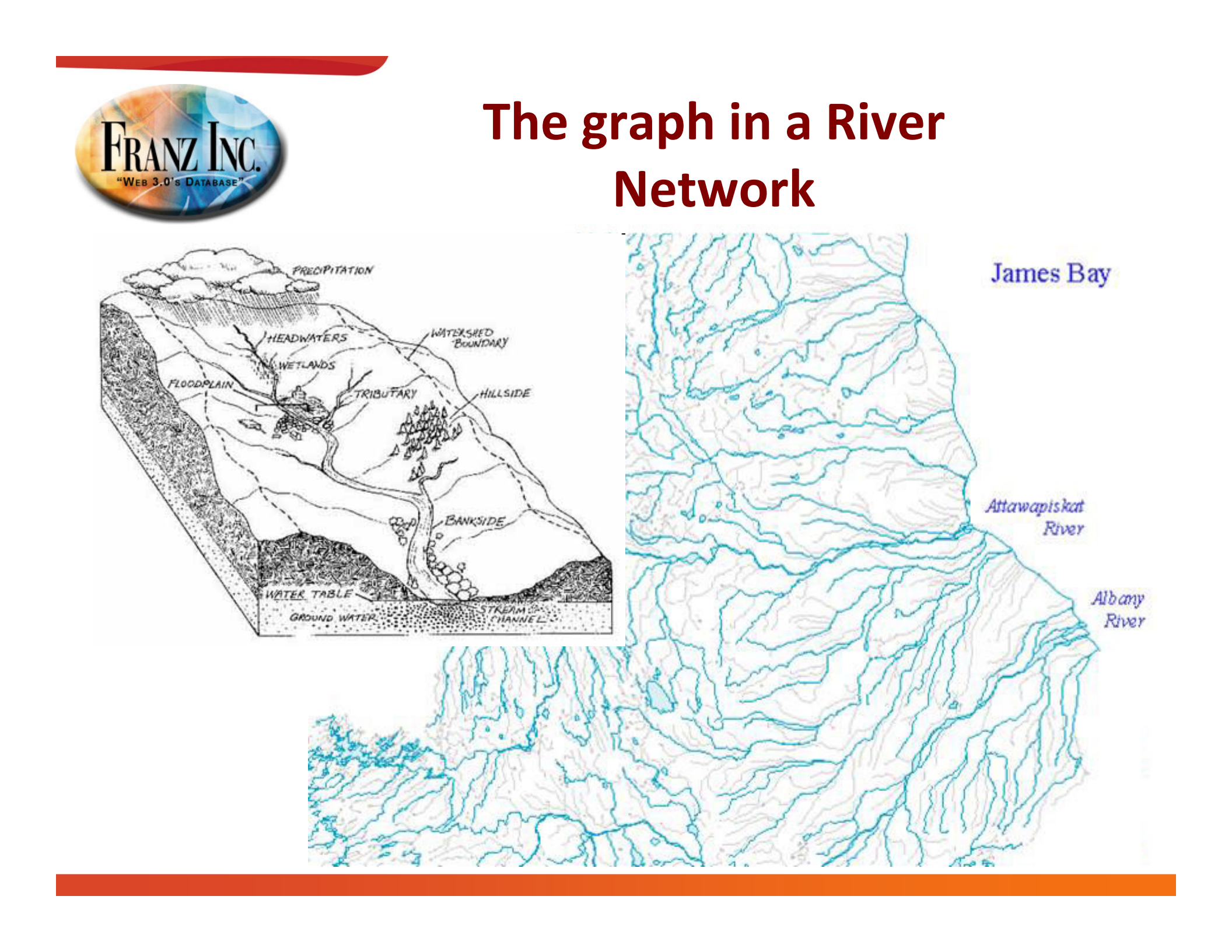
Using GeoNames

- coordinates for 6,445,201 places
- 109,568,417 RDF triples

AllegroGraph requires only 95 msec real time to return the 502 entries within a 3 mile radius of the Franz Inc offices.

Example 1: dealing with a river network using KML and Google Earth

- Working with data from GeoBC


FRANZ INC.
"WEB 3.0's DATABASE"
The graph in a River Network
PRECIPITATION
HEADWATERS
WATERSHED BOUNDARY
WETLANDS
FLOODPLAIN
TRIBUTARY
HILLSIDE
BANKSIDE
WATER TABLE
GROUND WATER
STREAM CHANNEL
James Bay
Attawapiskat River
Albany River

Regular data with a graph

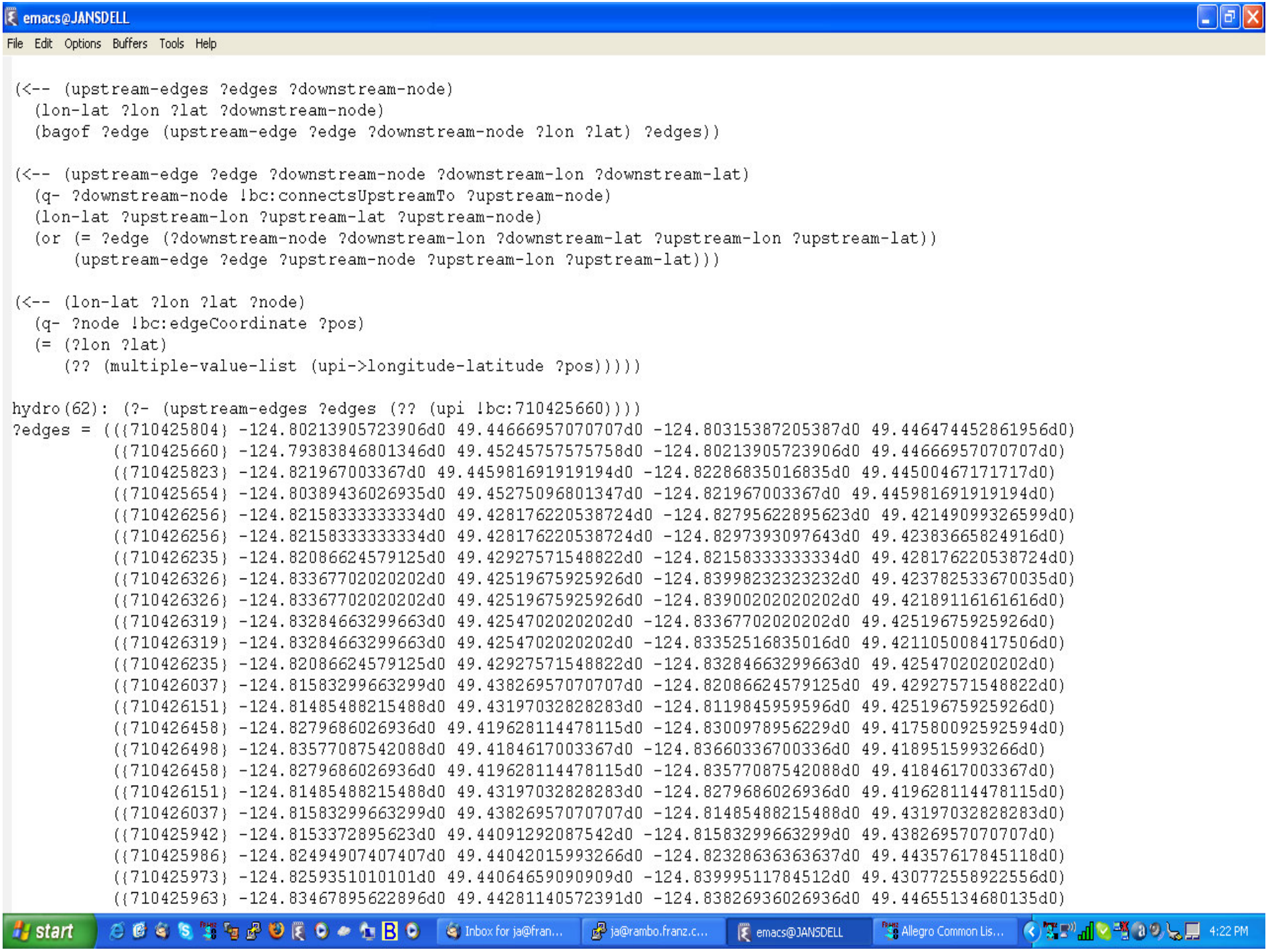
```
S1 type stream-segment
S1 upstream S2
S1 upstream S3
S1 left-drainage D1
S1 right-drainage D2
S1 isAt (-121.2, 12.1)
```

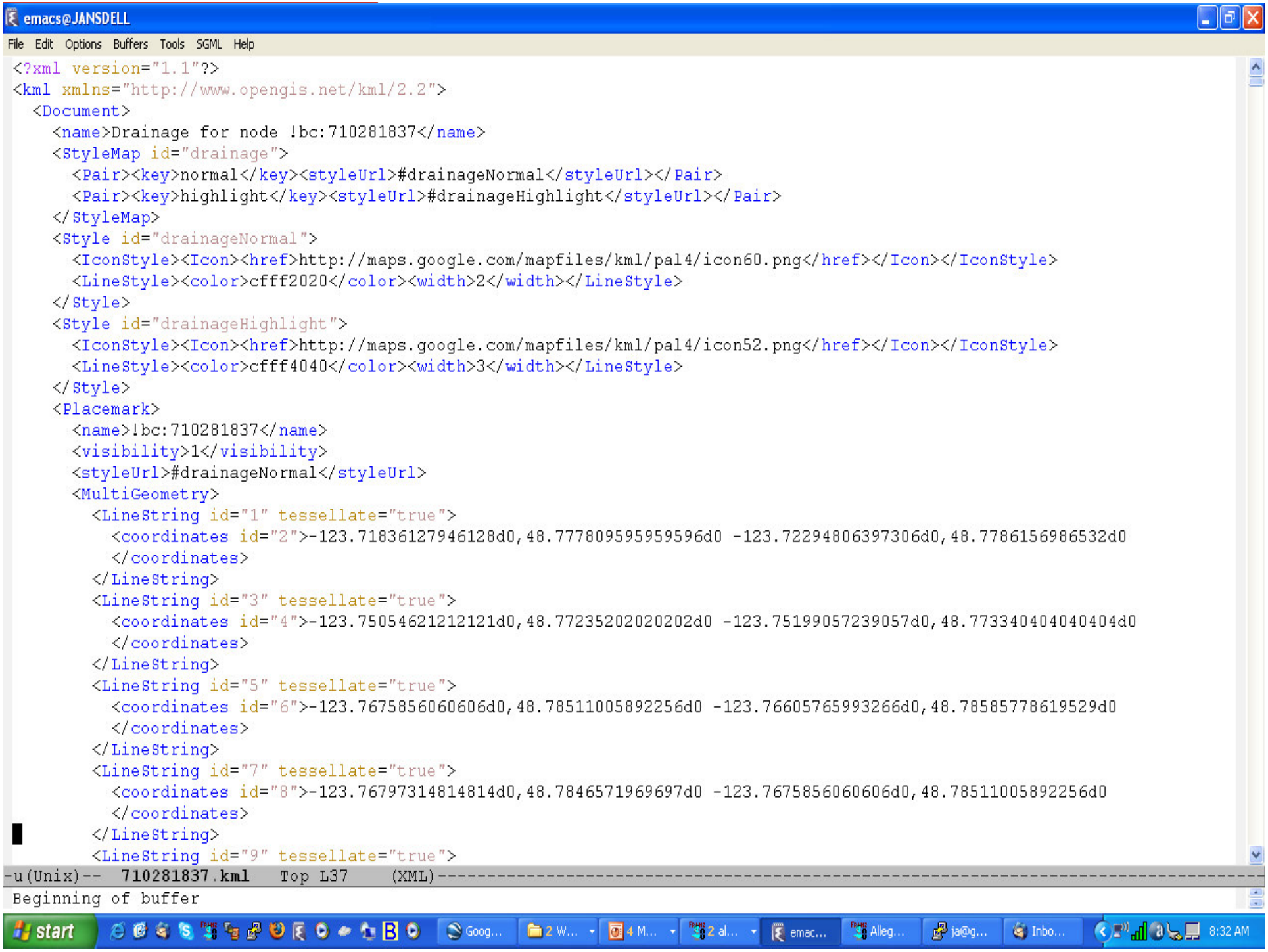
Given the polluted segment S1 find all the upstream segments within 50 miles of City1200

Given the polluted drainage D1 find all the schools in the rectangle <x1, y1, x2, y2> that might be influenced

Queries

- Graph search
  - find all upstreams or downstreams of a segment
- Graph search + bounding box
  - Given the polluted drainage D1 find all the schools in the rectangle <x1, y1, x2, y2> that might be influenced


emacs@JANSDELL
File Edit Options Buffers Tools Help
(<-- (upstream-edges ?edges ?downstream-node)
(lon-lat ?lon ?lat ?downstream-node)
(bagof ?edge (upstream-edge ?edge ?downstream-node ?lon ?lat) ?edges))
(<-- (upstream-edge ?edge ?downstream-node ?downstream-lon ?downstream-lat)
(q- ?downstream-node !bc:connectsUpstreamTo ?upstream-node)
(lon-lat ?upstream-lon ?upstream-lat ?upstream-node)
(or (= ?edge (?downstream-node ?downstream-lon ?downstream-lat ?upstream-lon ?upstream-lat))
(upstream-edge ?edge ?upstream-node ?upstream-lon ?upstream-lat)))
(<-- (lon-lat ?lon ?lat ?node)
(q- ?node !bc:edgeCoordinate ?pos)
(= (?lon ?lat)
(?? (multiple-value-list (upi->longitude-latitude ?pos)))))
hydro(62): (?- (upstream-edges ?edges (?? (upi !bc:710425660))))
?edges = (((710425804) -124.80213905723906d0 49.446669570707707d0 -124.80315387205387d0 49.4464744528619560)
((710425660) -124.793838468013460 49.45245757575758d0 -124.80213905723906d0 49.446669570707070)
((710425823) -124.821967003367d0 49.4459816919191940 -124.82286835016835d0 49.445004671717170)
((710425654) -124.80389436026935d0 49.45275096801347d0 -124.821967003367d0 49.4459816919191940)
((710426256) -124.82158333333334d0 49.428176220538724d0 -124.82795622895623d0 49.421490993265990)
((710426256) -124.82158333333334d0 49.428176220538724d0 -124.8297393097643d0 49.42383665824916d0)
((710426235) -124.82086624579125d0 49.42927571548822d0 -124.82158333333334d0 49.428176220538724d0)
((710426326) -124.83367702020202d0 49.4251967592592 6d0 -124.83998232323232d0 49.423782533670035d0)
((710426326) -124.83367702020202d0 49.4251967592592 6d0 -124.8390020202020 2d0 49.42189116161616d0)
((710426319) -124.83284663299663d0 49.4254702020202d0 -124.83367702020202d0 49.42519675925926d0)
((710426319) -124.83284663299663d0 49.4254702020202d0 -124.83352516835016d0 49.421105008417506d0)
((710426235) -124.82086624579125d0 49.42927571548822d0 -124.83284663299663d0 49.4254702020202d0)
((710426037) -124.81583299663299d0 49.43826957070707d0 -124.82086624579125d0 49.42927571548822d0)
((710426151) -124.81485488215488d0 49.43197032828283d0 -124.8119845959596d0 49.42519675925926d0)
((710426458) -124.8279686026936d0 49.4196281144781150 -124.8300978956229d0 49.417580092592594d0)
((710426498) -124.83577087542088d0 49.4184617003367d0 -124.83660336700336d0 49.4189515993266d0)
((710426458) -124.8279686026936d0 49.4196281144781150 -124.83577087542088d0 49.4184617003367d0)
((710426151) -124.81485488215488d0 49.43197032828283d0 -124.8279686026936d0 49.4196281144781150)
((710426037) -124.81583299663299d0 49.43826957070707d0 -124.81485488215488d0 49.43197032828283d0)
((710425942) -124.8153372895623d0 49.44091292087542d0 -124.81583299663299d0 49.43826957070707d0)
((710425986) -124.82494907407407d0 49.440420159932660 -124.82328636363637d0 49.443576178451180)
((710425973) -124.8259351010101d0 49.44064659090909d0 -124.83999511784512d0 49.430772558922560)
((710425963) -124.83467895622896d0 49.442811405723910 -124.83826936026936d0 49.44655134680135d0)
start
Inbox for ja@fran...
ja@rambo.franz.c...
emacs@JANSDELL
Allegro Common Lis...
4:22 PM


emacs@JANSDELL
File Edit Options Buffers Tools SGML Help
<?xml version="1.1"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<name>Drainage for node !bc:710281837</name>
<StyleMap id="drainage">
<Pair><key>normal</key><styleUrl>#drainageNormal</styleUrl></Pair>
<Pair><key>highlight</key><styleUrl>#drainageHighlight</styleUrl></Pair>
</StyleMap>
<Style id="drainageNormal">
<IconStyle><Icon><href>http://maps.google.com/mapfiles/kml/pal4/icon60.png</href></Icon></IconStyle>
<LineStyle><color>cfff2020</color><width>2</width></LineStyle>
</Style>
<Style id="drainageHighlight">
<IconStyle><Icon><href>http://maps.google.com/mapfiles/kml/pal4/icon52.png</href></Icon></IconStyle>
<LineStyle><color>cfff4040</color><width>3</width></LineStyle>
</Style>
<Placemark>
<name>!bc:710281837</name>
<visibility>1</visibility>
<styleUrl>#drainageNormal</styleUrl>
<MultiGeometry>
<LineString id="1" tessellate="true">
<coordinates id="2">-123.71836127946128d0,48.777809595959596d0 -123.72294806397306d0,48.7786156986532d0
</coordinates>
</LineString>
<LineString id="3" tessellate="true">
<coordinates id="4">-123.75054621212121d0,48.7723520202020 2d0 -123.75199057239057d0,48.773340404040404d0
</coordinates>
</LineString>
<LineString id="5" tessellate="true">
<coordinates id="6">-123.7675856060606d0,48.7851100589225 6d0 -123.7660576599326 6d0,48.78585778619529d0
</coordinates>
</LineString>
<LineString id="7" tessellate="true">
<coordinates id="8">-123.76797314814814d0,48.784657196969 7d0 -123.7675856060606d0,48.7851100589225 6d0
</coordinates>
</LineString>
<LineString id="9" tessellate="true">
-u(Unix)-- 710281837.kml Top L37 (XML)
Beginning of buffer
start
Goog...
2 W...
4 M...
2 al...
emac...
Alleg...
ja@g...
Inbo...
8:32 AM

Google Earth..

KML Tutorial - KML - Google Code - Mozilla Firefox

File Edit View History Delicious Bookmarks Tools Help

http://code.google.com/apis/kml/documentation/kml_tut.html

Gmail GCAL Movies Sudoku Ttext Weather Amazon Twine This Wells Fargo FtoC nwa IMDB BB wheather Franz psf todo USTA kranten router

Google Code

Search

e.g. "ajax apis" or "open source"

English ▼ | Site Directory

KML

Home **Docs** FAQ Articles Blog Group

KML in Google Maps
KML/Maps Details
Submit Your Geo Content to Google
Interactive Sampler

Documentation

Introduction
KML Tutorial
Developer's Guide
KML Reference

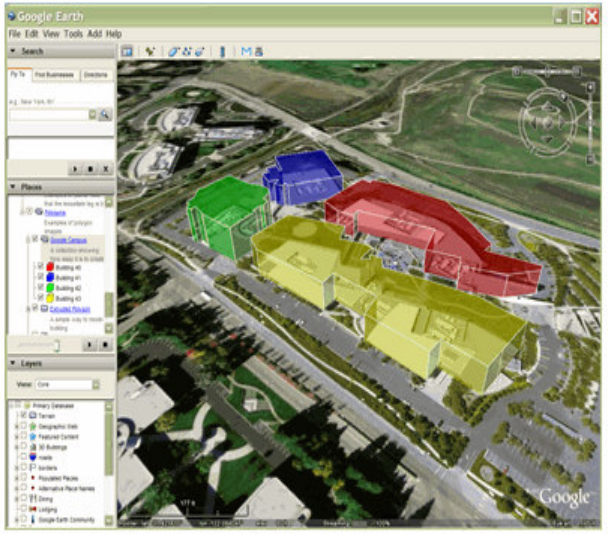
More Resources

Media Gallery

KML Tutorial

KML is a file format used to display geographic data in an Earth browser such as Google Earth, Google Maps, and Google Maps for mobile. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard. All tags are case-sensitive and must be appear exactly as they are listed in the KML Reference. The Reference indicates which tags are optional. Within a given element, tags must appear in the order shown in the Reference.

If you're new to KML, explore this document and the accompanying samples files (*SamplesInEarth* and *SamplesInMaps*) to begin learning about the basic structure of a KML file and the most commonly used tags. The first section describes features that can be created with the Google Earth user interface. These features include placemarks, descriptions, ground overlays, paths, and polygons. The second section describes features that require authoring KML with a text editor. When a text file is saved with a *.kml* or *.kmz* extension, Earth browsers know how to display it.



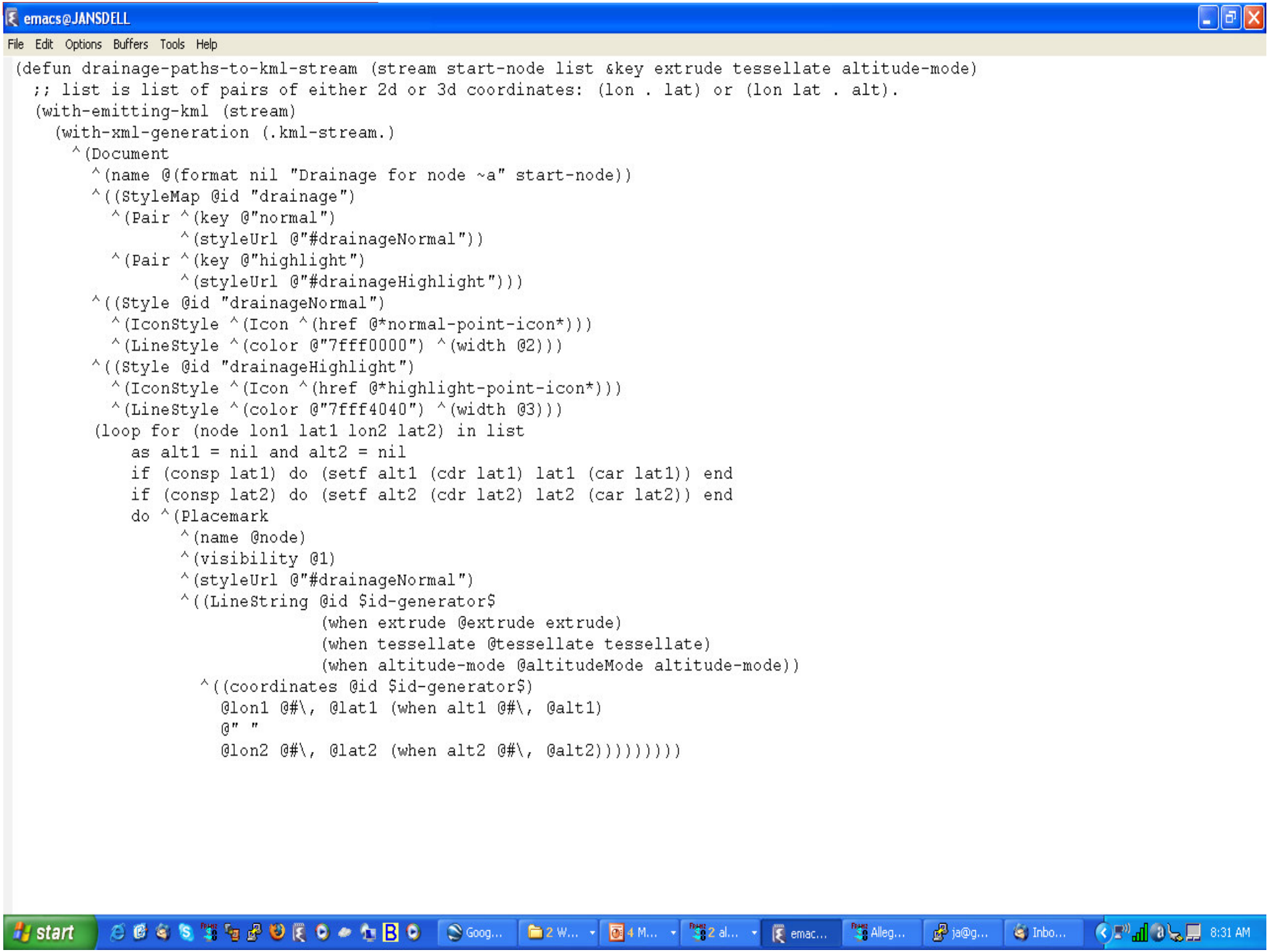
Tip: To see the KML "code" for a feature in Google Earth, you can simply right-click the feature in the 3D Viewer of Google Earth and select Copy. Then Paste the contents of the clipboard into any text editor. The visual feature displayed in Google Earth is converted into its KML text equivalent. Be sure to experiment with this feature.

All of the examples described here are in the KML Samples file. Begin by downloading that file to view the examples in Google Earth.

For More Information

Done

start 8:36 AM

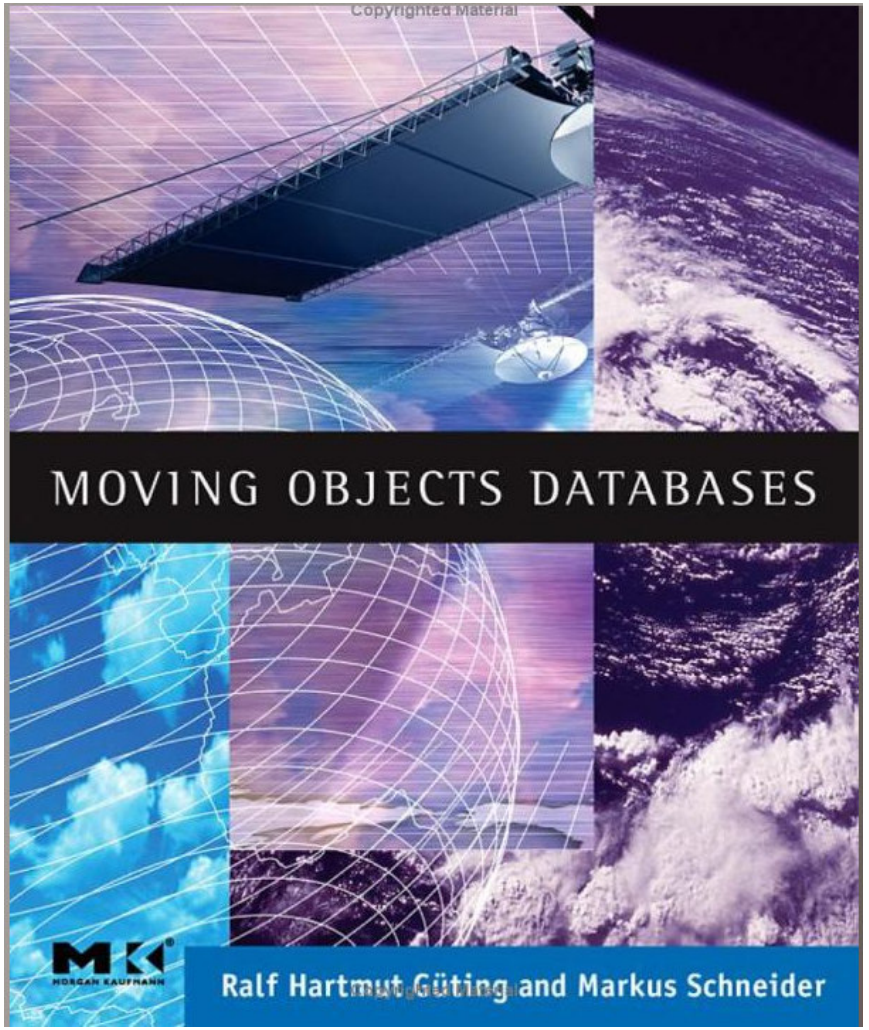

emacs@JANSDELL
File Edit Options Buffers Tools Help
(defun drainage-paths-to-kml-stream (stream start-node list &key extrude tessellate altitude-mode)
;; list is list of pairs of either 2d or 3d coordinates: (lon . lat) or (lon lat . alt).
(with-emitting-kml (stream)
(with-xml-generation (.kml-stream.)
^(Document
^(name @(format nil "Drainage for node ~a" start-node))
^((StyleMap @id "drainage")
^(Pair ^(key @"normal")
^(styleUrl @"#drainageNormal"))
^(Pair ^(key @"highlight")
^(styleUrl @"#drainageHighlight")))
^((Style @id "drainageNormal")
^(IconStyle ^(Icon ^(href @*normal-point-icon*)))
^(LineStyle ^(color @"7fff0000") ^(width @2)))
^((Style @id "drainageHighlight")
^(IconStyle ^(Icon ^(href @*highlight-point-icon*)))
^(LineStyle ^(color @"7fff4040") ^(width @3)))
(loop for (node lon1 lat1 lon2 lat2) in list
as alt1 = nil and alt2 = nil
if (consp lat1) do (setf alt1 (cdr lat1) lat1 (car lat1)) end
if (consp lat2) do (setf alt2 (cdr lat2) lat2 (car lat2)) end
do ^(Placemark
^(name @node)
^(visibility @1)
^(styleUrl @"#drainageNormal")
^((LineString @id $id-generator$
(when extrude @extrude extrude)
(when tessellate @tessellate tessellate)
(when altitude-mode @altitudeMode altitude-mode))
^((coordinates @id $id-generator$)
@lon1 @#\, @lat1 (when alt1 @#\, @alt1)
@" "
@lon2 @#\, @lat2 (when alt2 @#\, @alt2)))))))))
start
Goog...
2 W...
4 M...
2 al...
emac...
Alleg...
ja@g...
Inbo...
8:31 AM

Future directions with KML

- We are considering some general support for KML
- Templates
- Shapes:
  - Points
  - Linesegments
  - Polygons
- Please let us know what you want…

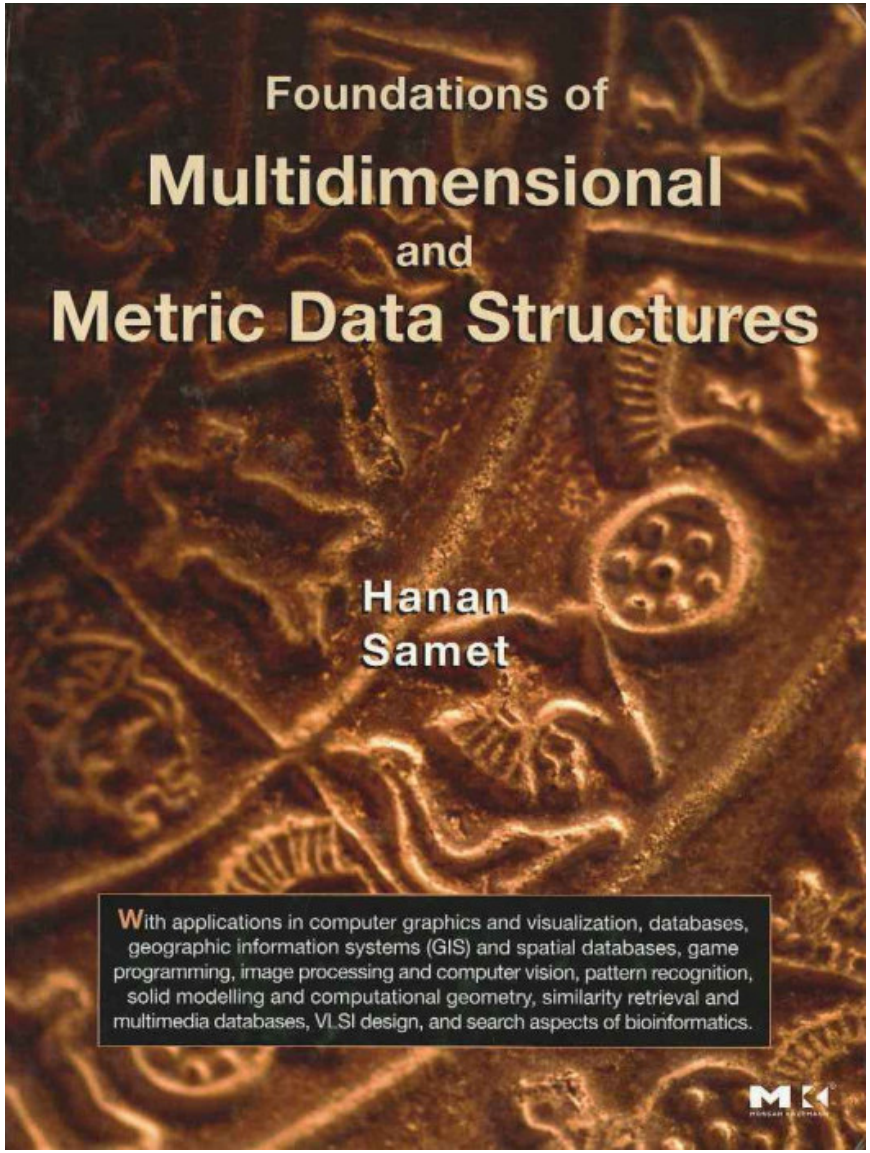
Moving Objects

- Fleets of trucks and swarms of airplanes
- Track data for soldiers
- Track data for endangered animals
- New Telco Service:
  - Loopt and friends



The challenge

- Track a million objects
- At a resolution of once per minute or a change in x meters
- Both in real time
  - Use geotemporal (3D) indexing techniques from the game industry
  - Do complex event handling for current situation
- And historically
  - Use geotemporal (3D) indexing



Questions we need to answer

- Real time:
  - Do we have 5 soldiers that have been active less than 8 with training x and equipment y within a mile distance of each other
  - Find another truck that can pick up package X at location Y so that I can pick up package A at location B so that we both will arrive at P before time T.

So far implemented – real time

- An object model for representing all active objects
- In memory 3D indexing using a grid approach
- Simple rules to trigger action based on in-memory objects
- Simple rules to aggregate real time data and store it in the triple store.

So far implemented historically

- Three dimensional indices for x, y, t and t, x, y
- (geotemporal-box ?a ?x1 ?y1 ?x2 ?y2 ?t1 ?duration)
- (geotemporal-box ?a ?gbox ?tbox)
- (how-close ?a ?b ?tbox ?dist)
- (how-close ?a ?b ?gbox ?tbox ?dist)
- (trajectories-crossed ?a ?b ?tbox ?loc ?time)

And

- Suggestions will be very welcome…

Thanks