The Decision Table Template For Geospatial Business Rules

Alex Karman
CTO and Co-Founder
Revolutionary Machines, Inc.
www.rev-mac.com

San Jose, Oct 13-15, 2014

OpenRules Now Supports Spatial Rules

- Leverages the popular JTS Topology Suite (“JTS”)
- Supports the Egenhofer Relationships (“DE9-IM”) for 2D points, polygons and line strings
  - Contains, touches, crosses, overlaps, disjoint, etc.
- Supports distance and area calculations; and ranking by distance or area
- Supports aggregates (max/min) of spatial rules
- Supports non-spatial mereological rules
  - Part of/comprises
- Loads Geographic Markup Language (GML) from text files with a GeometryDatabaseBuilder utility

Motivation

- Last year, we used OpenRules to handle business rules related to security constraints and service level agreements in a data center management project.
- This year, the customer asked us if OpenRules could manage fraud detection and privacy rules in a healthcare project in the same data center.
- We looked at the problem domain and saw a large number of spatial rules.

Spatial Business Rules Are Everywhere

- Healthcare
  - Hospital Referral Region, Hospital Service Area, Hospital, Patient, Emergency Routes
- Sales
  - Supplier and buyer territories, census block demographics
- Utilities
  - Markets are usually defined geographically
- Local government
  - Cadasters, zones, counties, municipalities

Most Spatial Business Rules Only Require a Simple Vocabulary

- Describe how simple points, polygons and lines interact
- Describe distances between them
- Describe “at least” or “no more than” rules (aggregate spatial rules)

Most Spatial Business Rules Never Use Most GIS Features

- Continuous field data
  - Weather, climate, netCDF, raster
- Slope and aspect
  - Digital elevation model, bathymetry, viewshed
- Topology
  - The shoreline borders the shore
- Spatial statistics
  - Autocorrelation, Moran's I, Geary's C, etc.
- Spatial Reference System, Projections

Therefore
Most Business Rule Projects

- Should never need a GIS system
- Should never need a GIS person
- Should have a single unbroken phase of rule development performed by the same people
- Should have a single rule repository
- Should have a single rule execution environment
- **Spatial business rules should not cause project separation, duplication or delay**

Goals

- Create a simple spatial business rule spreadsheet template
- Express spatial business rules in plain English
- Allow spatial rule spreadsheets to be used in combination with non-spatial business rule spreadsheets
  - Decision Model: the “conclusion” of a spatial business rule family can serve as the “condition” of another business rule family
- Make it simple to load the spatial data from GML files
  - GeoDatabaseBuilder class

JTS Topology Suite

- Java spatial engine
- Started in 2000
- Stable, peer reviewed, and widely used
- Used in GeoServer, OpenJUMP, and uDIG
- Converts GML into spatial objects in memory, and evaluates the spatial relationships between them
- **Consistent** results

GML

- **Geography Markup Language, an Open Geospatial Consortium ("OGC") standard**
- **It is an XML file that conforms to the GML schema**

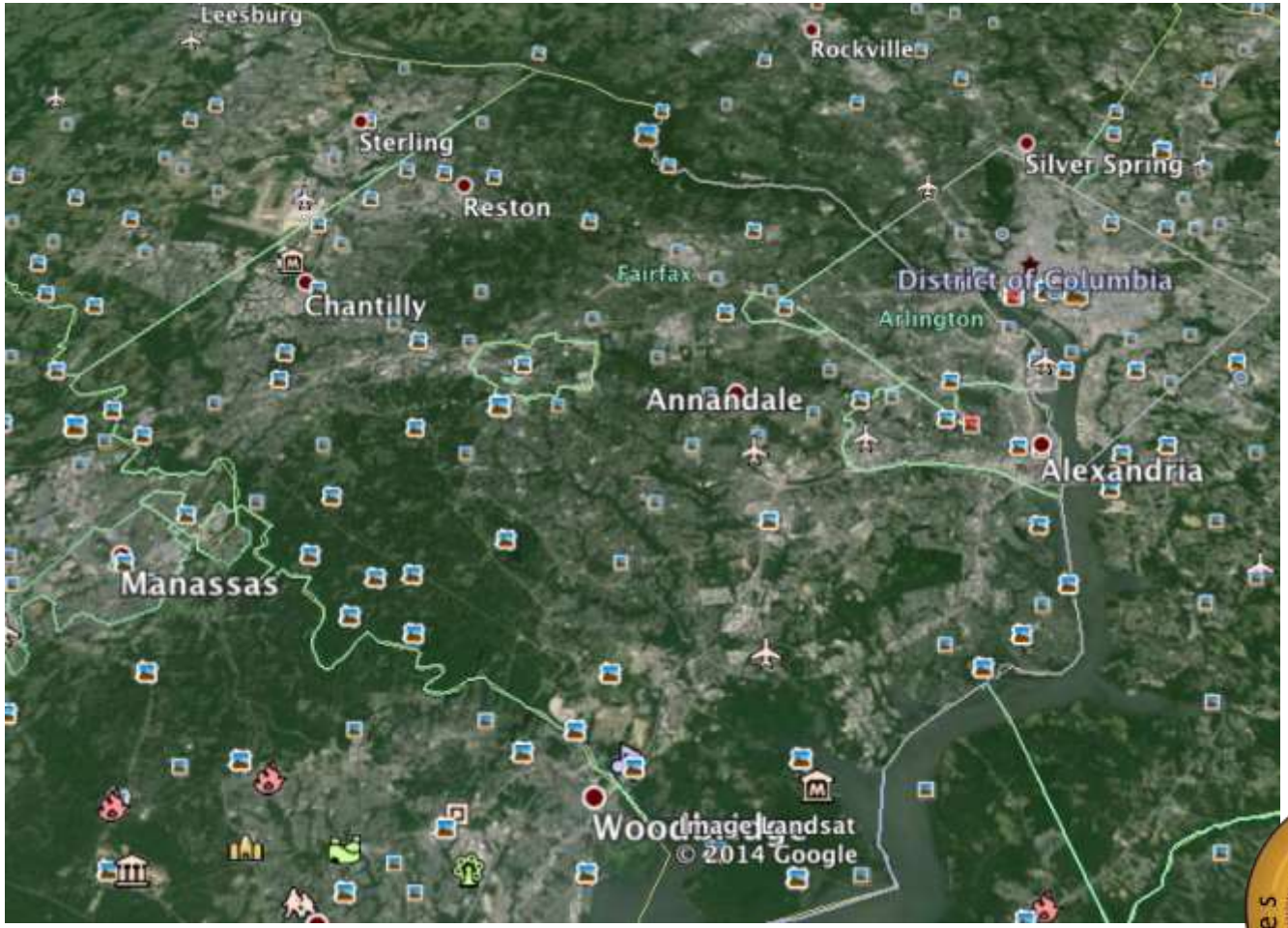
```
<gml:featureMember>
  <gml2:Placemark>
    <gml:pointProperty>
      <gml:Point srsName="LL84">
        <gml:coordinates>-112.025980391177,33.4539329678913,0
        </gml:coordinates>
      </gml:Point>
    </gml:pointProperty>
  </gml2:Placemark>
</gml:featureMember>
```

New Glossary Type: GeoEntity

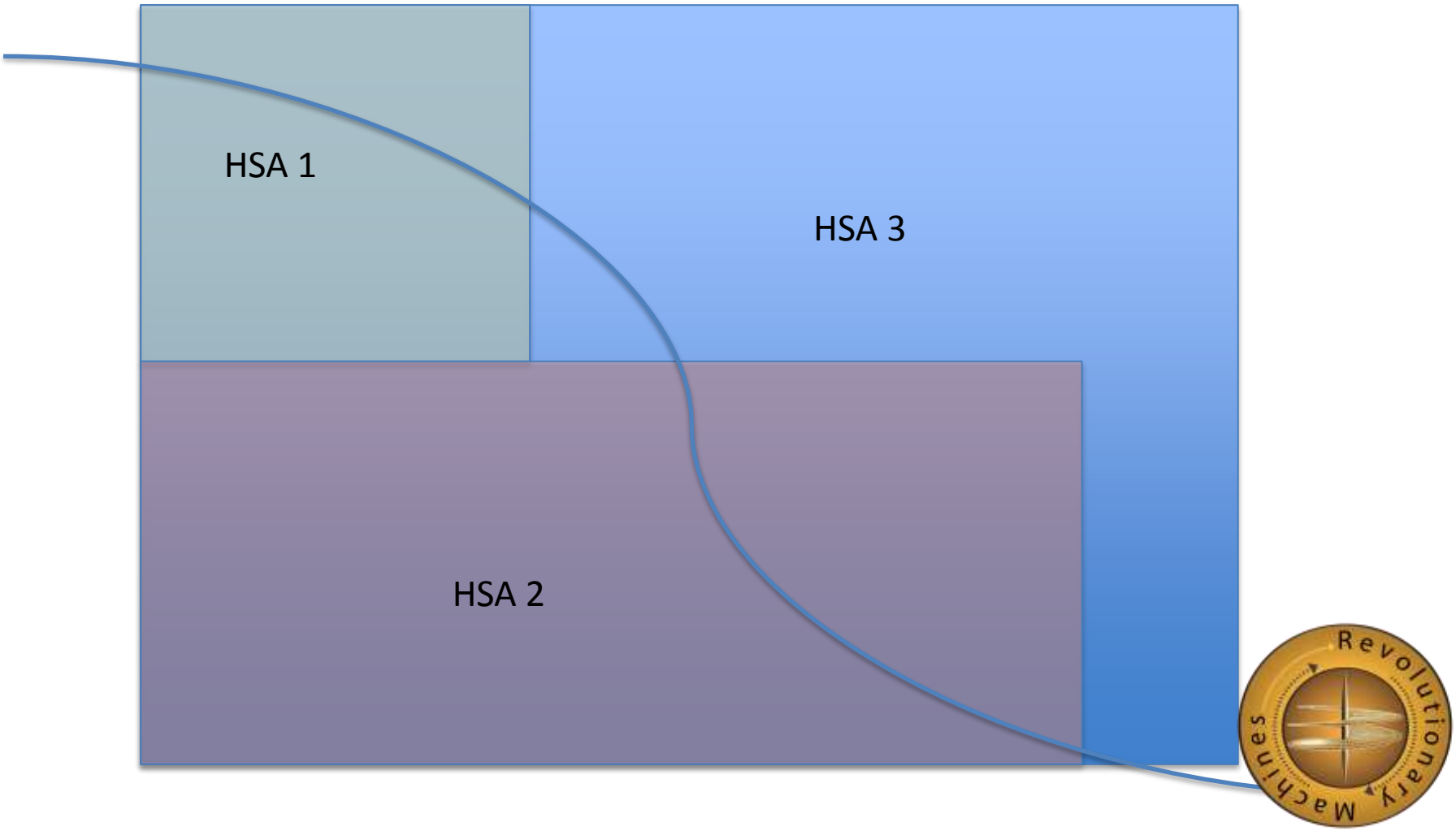
Type	GeoEntity	type	
Id		id	
Name		name	
Geometry		geometry	

Maps to Java bean com.openrules.spatial.GeoEntity

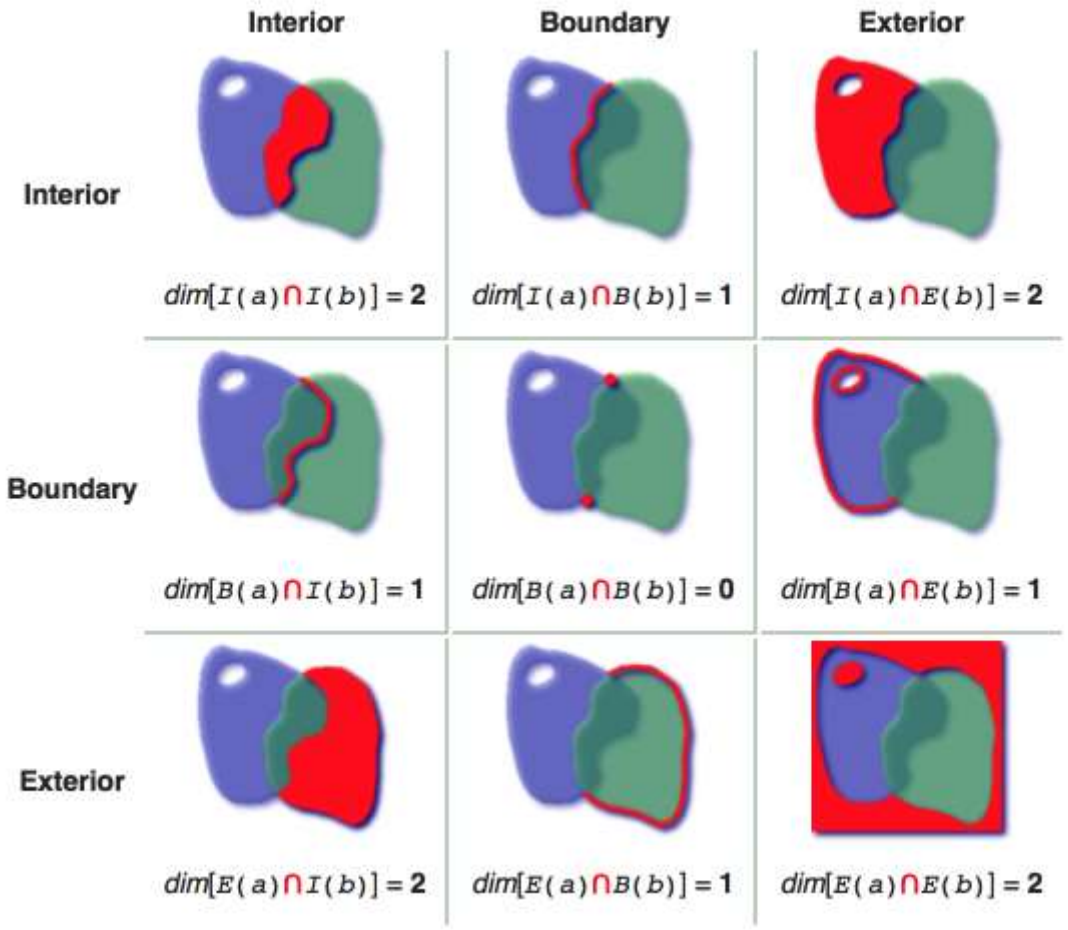
Egenhofer Relationships



Several Hospital Service Areas Form One Hospital Referral Region



Linear Algebra (DE-9IM)



Usually Expressed As Spatial Predicates

1. Equals
2. Disjoint
3. Touches
4. Contains
5. Covers
6. Intersects
7. Within
8. Covered By
9. Crosses
10. Overlaps

See http://en.wikipedia.org/wiki/DE-9IM

New Decision Table Type: DecisionTableSpatial

DecisionTableSpatial EntityToEntityRules							
C#	ConditionEntityToEntity					Conclusion	
#	Relationship Between Two Entities					Spatial Significance Score	
	Main Entity Type	Relationship	Related Entity Type	Oper	Value		
0						=	0
1	HRR	Contains	HSA	Is	TRUE	+=	2
2	HRR	Touches	HSA	Is	TRUE	+=	1
3	HRR	Is Disjoint From	HSA	Is	FALSE	+=	3
4	HRR+5	Contains	HSA	Is	TRUE	+=	1
5	Route	Crosses	HSA	Is	TRUE	+=	2
6	HRR	Overlaps	County	Is	TRUE	-=	2
7	Airport	Distance	Hospital	<	250	+=	1
8	HRR	Area		<	25	+=	1
9	County	Area		<	25	+=	1
10	HRR	Is Among 25 Closest To	Hospital	Is	TRUE	+=	1
11	Facility	Is Part Of	University	Is	TRUE	+=	1
12	University	Comprises	Facility	Is	TRUE	+=	4

Aggregate Rules

DecisionTableSpatial EntityToCountsRules							
C#	ConditionEntityToCountOfRelatedEntities					Conclusion	
#	Count of Related Entities					Spatial Significance Score	
	Main Entity Type	Relationship	Type of Related Entities	Oper	Value		
	HRR	Contains	HSA	>=	5	+=	2
	HRR	Contains	HSA	<	2	-=	1
	HRR	Overlaps	County	>=	2	+=	1
	Airport	Distance <=5	Hospital	>=	1	+=	5
	Airport	Distance >25	Hospital	In	5..15	+=	1
	Airport	Distance <10	Hospital	<	1	+=	1
	Residence	Distance <20	Hospital	>	2	+=	1
	Hospital	Distance < 15	Residence	>	1200	+=	1
	HRR	Distance <=5	Hospital	>	5	+=	3
	Hospital	Distance <=5	Hospital	>	5	+=	3

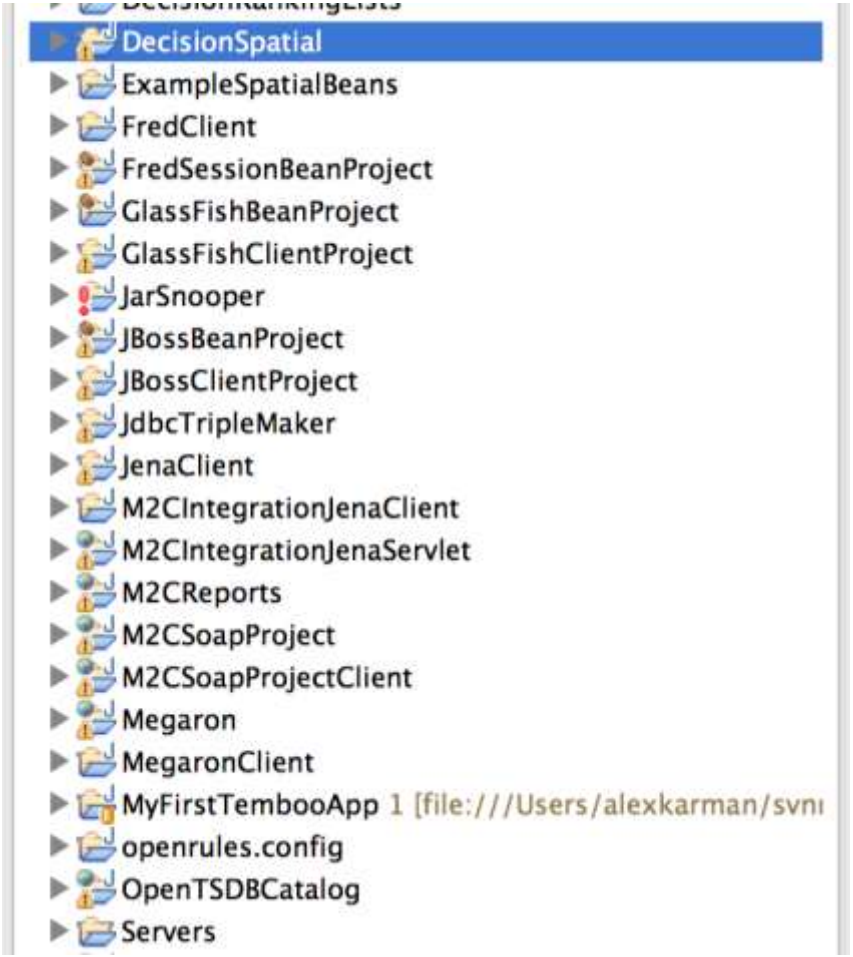
Aggregate Rules

HRR has at least 5 HSAs in it
HRR has fewer than 5 HSAs in it
HRR overlaps at least two Counties
At least one hospital is within 5 km from the Airport
Between 5 and 15 Hospitals are > 25 km from the Airport
No Hospital is within 10 km from the airport
More than 2 hospitals within 20 km from the Residence
More than 1200 residences within 20 km from the Hospital

Mereological Rules (Non-Spatial "Part-Of" Rules)

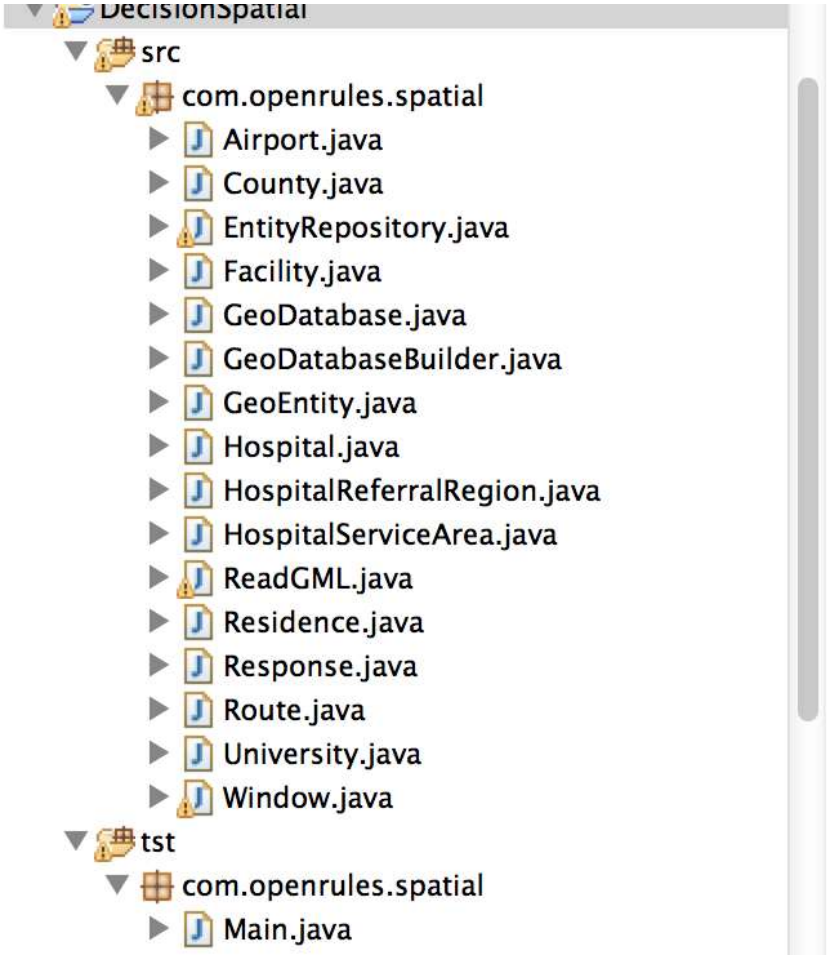
- Sometimes, we **think** we have discovered a spatial rule
- Then we realize that we don't care about the geography **at all**
- Example: Harvard has a Facility in Mumbai, India
- We only care that the Facility "Is Part Of" the University, and that the University "Comprises" the Facility. We don't care about the distance.
- Mereological rules work with any two kinds of Java Beans, as long as one has a property reference to the other

Importing Sample Application in Eclipse



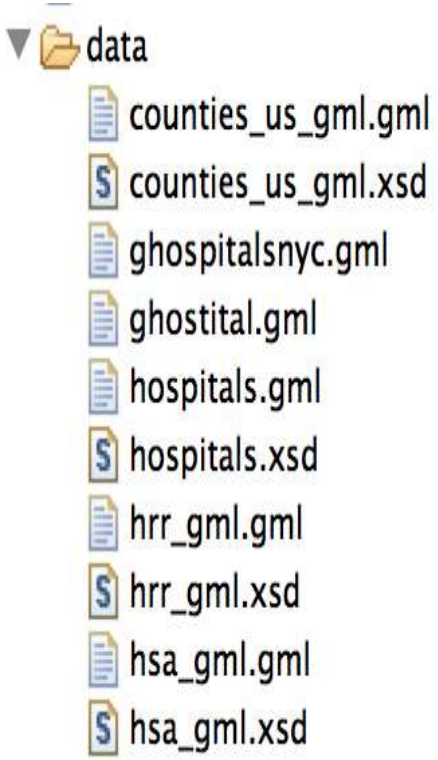
- Import DecisionSpatial
- Import openrules.config
- Link the projects
- If not Windows, change the file separators in GeoDatabaseBuilder.java from \\ to /
- Main class: com.openrules.spatial.Main

Java Code



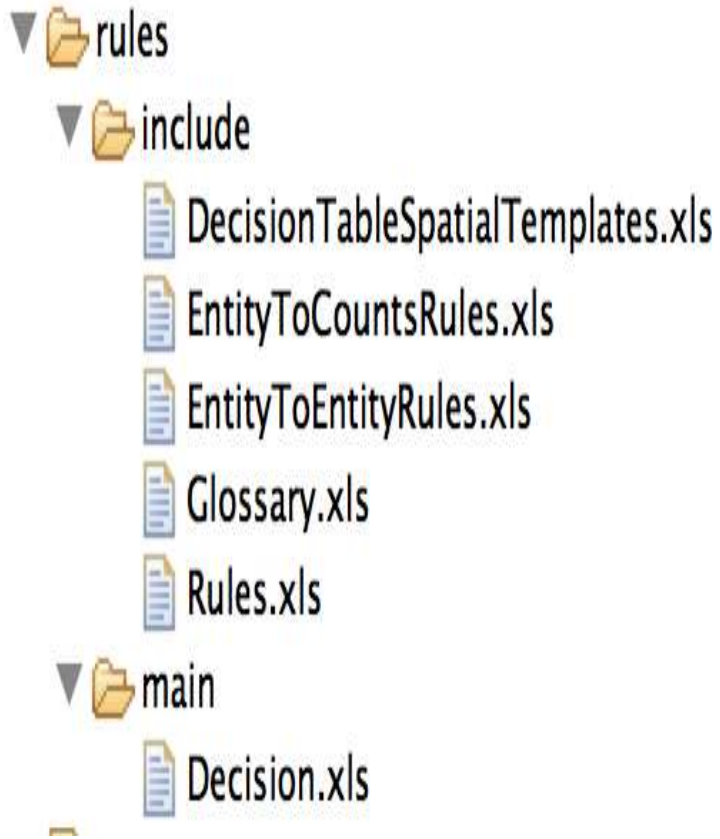
- Spatial Beans
  - Polygons: County, Hospital Service Area, Hospital Referral Region
  - Points: Airport, Hospital
  - Lines: Route
- Non-Spatial Beans
  - University, Facility, Window

GML Data Files



- Counties, HSAs, HRRs
- xsd files are supplied but not needed
- Point data is both ingested and created programmatically

Rules



- Template
- Egenhofer rules
  - EntitytoEntityRules
- Aggregate rules
  - EntityToCountsRules
- Mereological Rules
  - EntityToEntityRules
- Glossary
- Higher-level rule family
  - Rules.xls

Modifying the Sample Project

- Easiest way: just modify the decision tables
  - EntityToEntityRules.xls
  - EntityToCountsRules.xls
- Otherwise :
  - Add GML
  - Modify GeoDatabaseBuilder.java
- Like any other sample project:
  - Add new JavaBean classes
  - Modify Main.java, Glossary, Decision.xls

Custom Class 1
GeoDatabaseBuilder.java

Main.java | GeoDatabaseBuilder.java | Hospital.java | University.java | Window.java

```
    public static boolean build() {
        boolean result = true;
//      List<Geometry> counties = GeoDatabase.getCountyGeometries(".\\data\\counties_us_gml.gml");
//      List<Geometry> hrrs = GeoDatabase.getHRRGeometries(".\\data\\hrr_gml.gml");
//      List<Geometry> hsas = GeoDatabase.getHSAGeometries(".\\data\\hsa_gml.gml");
        List<Geometry> counties = GeoDatabase.getCountyGeometries("./data/counties_us_gml.gml");
        List<Geometry> hrrs = GeoDatabase.getHRRGeometries("./data/hrr_gml.gml");
        List<Geometry> hsas = GeoDatabase.getHSAGeometries("./data/hsa_gml.gml");

//      ReadGML.setPrint(true);
//      List<Geometry> hospitals = GeoDatabase.getHospitalGeometries(".\\data\\hospitals.gml");
        List<Geometry> hospitals = GeoDatabase.getHospitalGeometries("./data/hospitals.gml");
        if (counties != null)
            Log.info("There are " + counties.size() + " counties");
        else
            result = false;
        if (hrrs != null)
            Log.info("There are " + hrrs.size() + " HRRs");
        else
            result = false;
        if (hsas != null)
            Log.info("There are " + hsas.size() + " HSAs");
        else
            result = false;
        if (hospitals != null)
            Log.info("There are " + hospitals.size() + " hospitals");
        else
            result = false;
        return result;

    }
```

Custom Class 2: Main.java

Main.java | GeoDatabaseBuilder.java | Hospital.java | University.java | Window.java

```
public static void main(String[] args) {

    String fileName = "file:rules/main/Decision.xls";
    // Build EntityRepository from GML files
    EntityRepository.build();

    // Create Decision
    Decision decision = new Decision("DetermineSpatialSignificanceScore",fileName);
    decision.put("report", "On");
    decision.saveRunLog(true);

    GeoEntity mainEntity = EntityRepository.hrrs[0];
    GeoEntity relatedEntity = EntityRepository.hsas[0];
    run(decision,"FIRST RUN", mainEntity, relatedEntity);

    mainEntity = EntityRepository.hospitals[0];
    relatedEntity = EntityRepository.hsas[0];
    run(decision,"SECOND RUN", mainEntity, relatedEntity);

    mainEntity = EntityRepository.hrrs[1];
    relatedEntity = EntityRepository.hospitals[10];
    run(decision,"THIRD RUN", mainEntity, relatedEntity);

    Facility f1 = new Facility();
    f1.setId("Facility 1");
    Facility f2 = new Facility();
    f2.setId("Facility 2");
    Facility f3 = new Facility();
    f3.setId("Facility 3");
    University u = new University();
    u.setId("University 1");
    Facility[] facilities = { f1, f2, f3 };
    u.setFacilities(facilities);
    mainEntity = f2;
    relatedEntity = u;
    run(decision,"FOURTH RUN", mainEntity, relatedEntity);
}
```

Where Can We Go From Here?

- **Spatial Loading from spreadsheet**
  - Two column table: entity type and .gml file name
- **A "stipulation spreadsheet"**
  - So you don't have to find actual geometries to test the downstream effects of a spatial rule
- **Maven and Jenkins Integration**
- **Linear Referencing System Support**
  - Utilities, transportation networks, supply chains
  - Social networks
- **GeoSPARQL integration**
  - also uses GML and DE-9IM
  - Will provide spatial visualization
  - Will provide spatial rule validation

About Us

Revolutionary Machines, Inc. is a start-up company, based in Reston VA, that seeks to advance the development of technologies that will improve our world. We focus on the **intersection of the physical world and the information world**; where we employ **data science** techniques to help index, understand and improve enterprise, government and private domains of interest. We are preparing for the challenge of the **Internet of Things** and **Augmented Reality** through open source hardware and software initiatives, and by employing our expertise in **Semantic Web, GIS**, systems integration, **sensors**, and associated communications systems.

Thank You

- Please visit Revolutionary Machines at www.rev-mac.com
- Please email me at alex.karman@rev-mac.com
- Please download the DecisionSpatial sample project at www.openrules.com