A Framework for Collaborative Decision Support Modeling using OGC Specifications

Raj Singh
MIT Dept. of Urban Studies & Planning
rajsingh@mit.edu

Introduction

- Land management combines public discourse and expert analysis
- There is little connection between the two
- The result is decisions made by negotiation OR analysis--no give and take between the two

Decision Making in City Planning

- Decisions affect people's daily lives
- Decisions are always contentious
- Supporting dialog between experts and lay people is critical

Related Efforts

- Collaborative Decision Support
  - provides real time feedback
  - mostly weighting and rating
  - the model is local, small data requirements
- Large-scale modeling
  - e.g. UrbanSim, CUF, transportation models
  - too computationally or data intensive to bring to a meeting
  - important decisions buried in code

Planning Modeling Language: Goals

- Define a universal encoding for the modeling process
- Create a graphical environment for viewing and editing the model
- Specify a framework for collaboration
- Allow one model to be the input to another.

Planning Modeling Language: System Requirements

- Network-centric
- Portable
- Robust feature set
  - spatial operations
  - raster operations
  - table joins
  - aggregation, reclassification
- Fine-grained security
- Chainability

Test Case: Wetlands protection

- Task
  - Define a policy to exclude development around sensitive wetlands
- Points of possible contention
  - what types of wetlands are particularly sensitive?
  - what types of land uses have the greatest impact?
  - what is the impact on growth and revenue?

Wetlands protection model

- Gather data
  - land cover: roads, wetlands, zoning, agriculture
  - land use: traffic, building permits, TRI
- Construct a hypothesis
  - compare types of wetlands and their proximity to types of land uses

Wetlands data access

```
<pml:DataSource typeName="myFedNWI">
    <pml:DataBinding>
        <pml:port>http://a.b.com/wfs</pml:port>
        <pml:version>1.0.0</pml:version>
    </pml:DataBinding>
        <wfs:GetFeature>
            <wfs:Query typeName="FedNWI">
                <ogc:PropertyName>id</ogc:PropertyName>
                <ogc:PropertyName>nwiCode</ogc:PropertyName>
                <ogc:PropertyName>geom</ogc:PropertyName>
            </wfs:Query>
        </wfs:GetFeature>
</pml:DataSource>
```

Wetlands data processing

- Reclassification and aggregation

id	nwiCode	geom
1	PL2	x
2	LL1	x
3	LL2	x
4	AR1	x
5	LL1	x

nwiCode	sensitivityCode
PL1	1
PL2	1
LL1	2
LL2	2
AR1	3
AR2	3

id	sensitivityCode	geom
1	1	x
2	2	x
3	2	x
4	3	x
5	2	x

PML for wetlands data processing

```
<wfs:GetFeature>
    <wfs:Query typeName="FedNWI">
        <ogc:PropertyName>id</ogc:PropertyName>
        <ogc:PropertyName>nwiCode</ogc:PropertyName>
    </wfs:Query>
</wfs:GetFeature>
```


```
<wfs:GetFeature>
    <wfs:Query typeName="MyLookupTable">
        <ogc:PropertyName>nwiCode</ogc:PropertyName>
        <ogc:PropertyName>sensitivityCode</ogc:PropertyName>
    </wfs:Query>
</wfs:GetFeature>
```

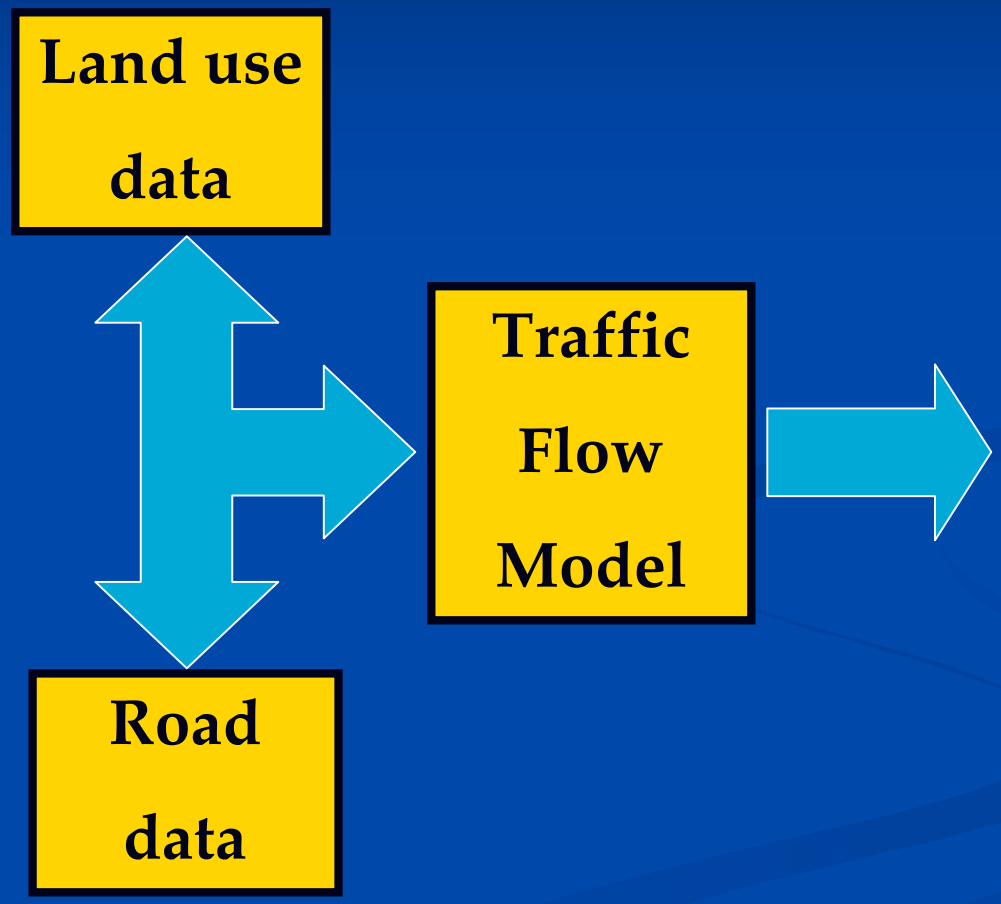
PML for wetlands data processing

```
<pml:Model name="ReclassedNWI">
    <pml:Join>
        ?????
    </pml:Join>
</pml:Model>
```

```
<wfs:GetFeature>
    <wfs:Query typeName="FedNWI">
        <ogc:PropertyName>id</ogc:PropertyName>
        <ogc:PropertyName>nwiCode</ogc:PropertyName>
    </wfs:Query>
</wfs:GetFeature>
```

```
<wfs:GetFeature>
    <wfs:Query typeName="MyLookupTable">
        <ogc:PropertyName>nwiCode</ogc:PropertyName>
        <ogc:PropertyName>sensitivityCode</ogc:PropertyName>
    </wfs:Query>
</wfs:GetFeature>
```

Traffic impact

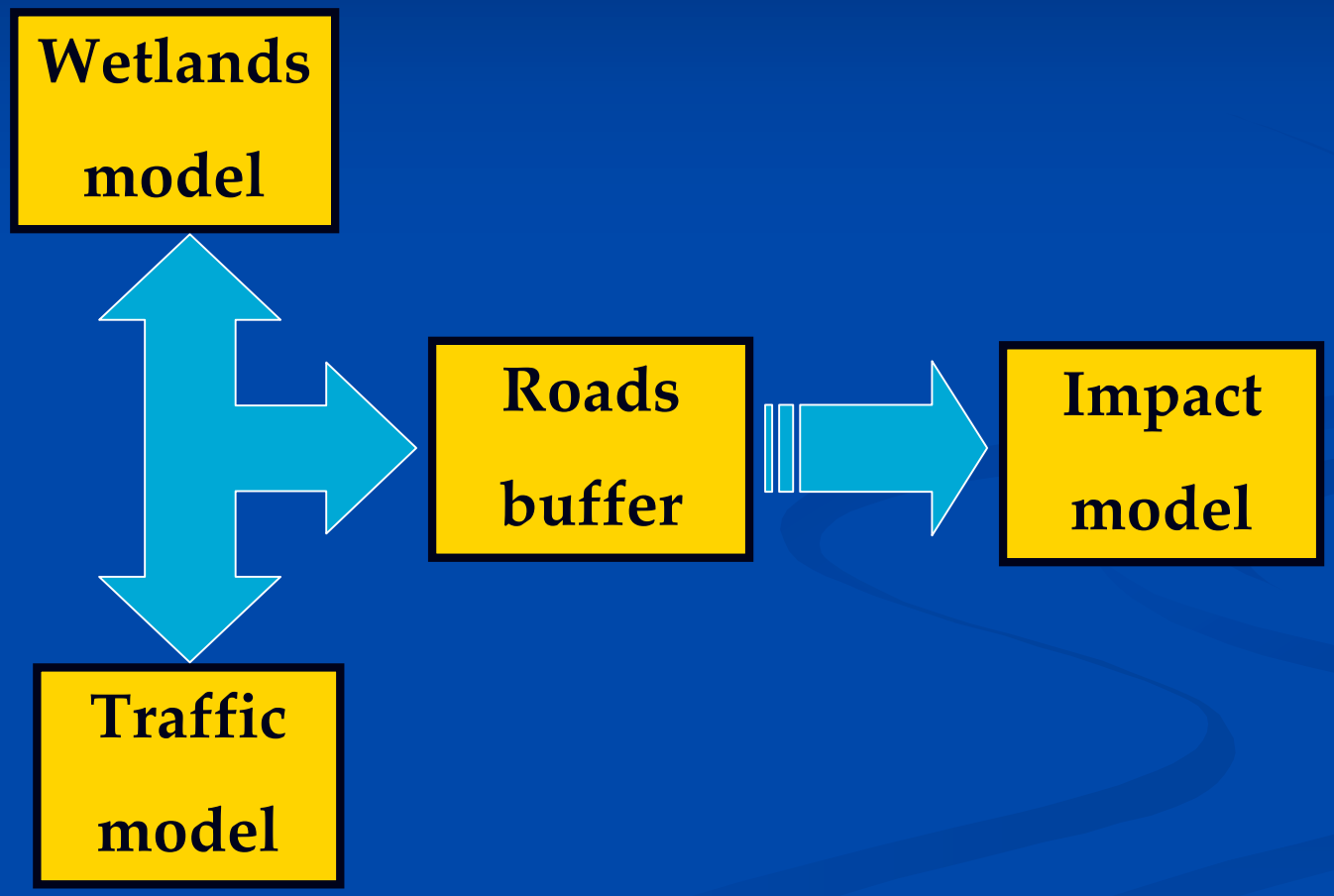


PML for roads buffer

```
<wfs:GetFeature>
    <wfs:Query typeName="StateRoads">
        <ogc:PropertyName>id</ogc:PropertyName>
        <ogc:PropertyName>numLanes</ogc:PropertyName>
    </wfs:Query>
</wfs:GetFeature>
```

```
<wfs:GetFeature>
    <wfs:Query typeName="ReclassifiedWetlands">
        <ogc:PropertyName>id</ogc:PropertyName>
        <ogc:PropertyName>sensitivityCode</ogc:PropertyName>
    </wfs:Query>
    <ogc:Filter>
      <ogc:DWithin>
        How to get the roads????
      <ogc:DWithin>
    </ogc:Filter>
</wfs:GetFeature>
```

Traffic impact



Deficiencies so far

- raster operations
  - moving window algorithms
- relates/joins
- fine-grained permissions
  - on attributes
  - on operations
- output as input

Findings

- Not sure if this is an interface or an encoding
- We're getting *close* to a language for decision support, but there's still a good bit of work to do
- Tools for building on OGC specs are limited and immature

More Information

- http://web.mit.edu/rajsingh/www/

Unused Slides

Planning Modeling Language: Tools

- Java Topology Suite for spatial operations
- OGC Filter specification for query/filter language
- Under consideration
  - GeoTools2 for map display, filter handling
  - PostGIS for data storage
  - JGraph for modeling GUI

Types of data objects

- general case: a URI that returns data
- a WFS request (which can contain a filter)
- the result of an operation on data objects
  - buffer, union, intersect
  - table join
- a non-spatial table
- and more…