Standards-Based Data Models and Web Services for Coastal Ocean Observations

Surya Durbha, Roger King, Nicolas Younan, Santhosh Rajender, and Shruthi Bheemireddy

Geosystems Research Institute (GRI)
Dept. of Electrical and Computer Engineering
Mississippi State University

Outline

- Background
- Sensor Web for the Ocean Observing System
- Semi-Automated Ontologies Alignment tool
- Semantic Matchmaking and Mobile Computing

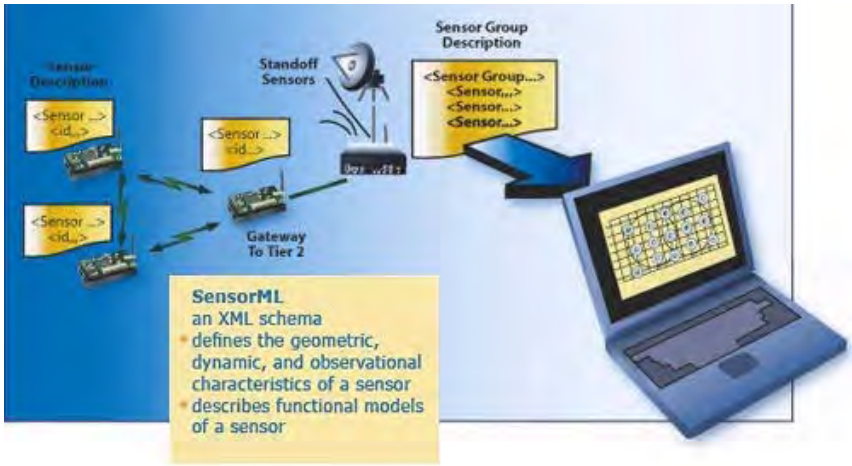
Acknowledgement:
This work is currently funded by the Northern Gulf Institute (NGI)

Sensor Networks

- The Integrated Ocean Observing System (IOOS) (The U.S. contribution to the GOOS and GEOSS)
  - Buoys and other ocean platforms
- Presently national networks
  - NDBC, GoMOOS, TAO, etc.
- Data discovery and conversion problems due to
  - Syntactic, structural, and semantic heterogeneity in the datasets.
- Ongoing implementation of a Services-driven Sensor Web Enablement framework for resolving these heterogeneity problems.

Sensor Model Language (SensorML)

- Means by which sensors and processes make themselves and their capabilities known.
- Geolocation of observed values (measured data)
- Performance characteristics
  - e.g. accuracy, threshold, etc.
- Explicit description of the process by which an observation was obtained (i.e. it's lineage)
- Executable process chain for deriving new data products on demand (i.e. derivable observation) (e.g. Geolocation)
- Archive fundamental properties and assumptions regarding sensor
- Supports data discovery



http://www.mitre.org/news/the_edge/spring_06/los.html)

Coastal Sensor Web Enablement

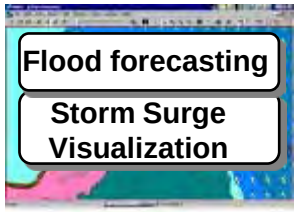
Heterogeneous Network Sources
(Various monitoring sensors)

Sensor Web Enablement (SWE)

- Discovery
- Access
- Tasking
- Alerts

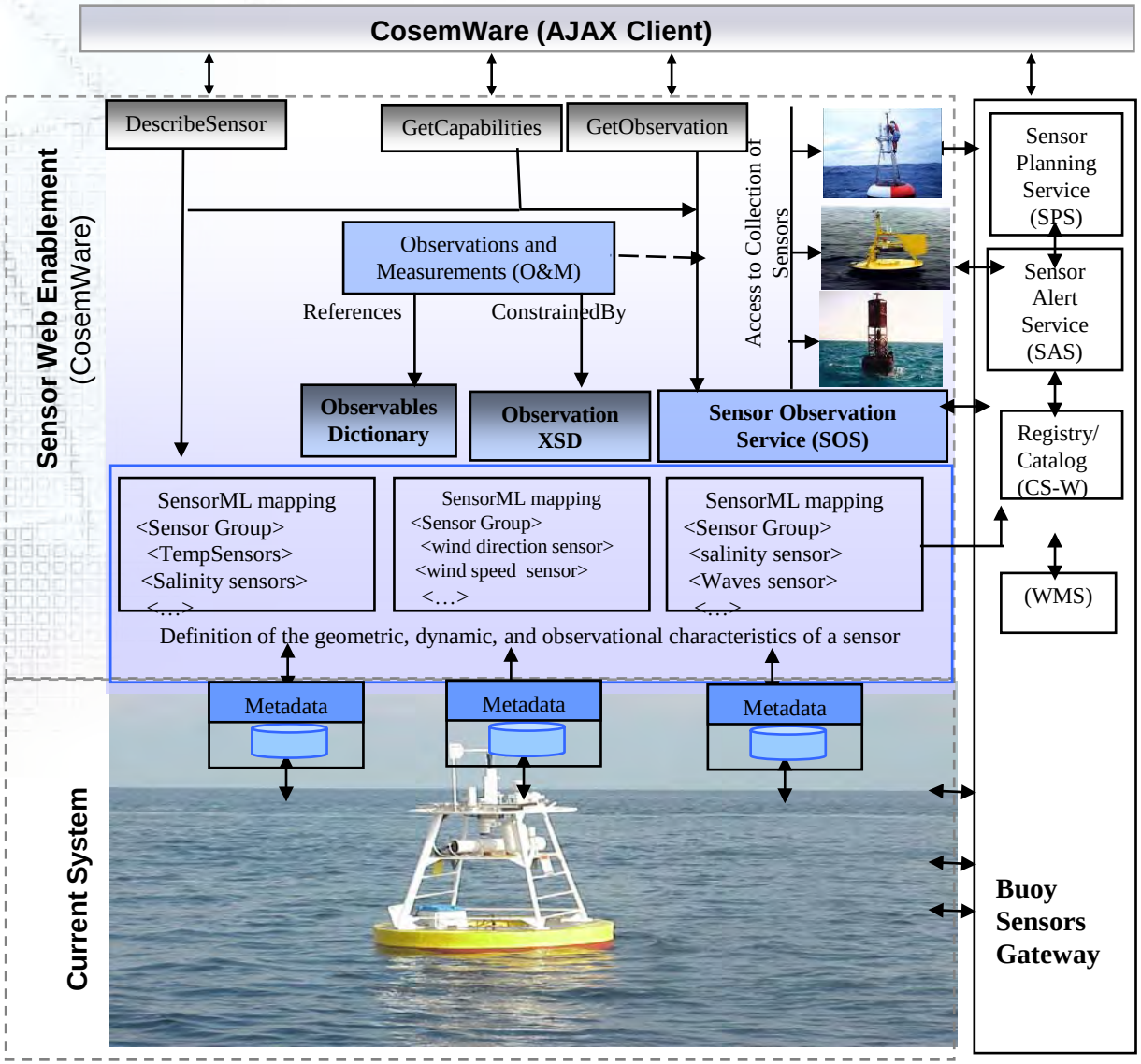
Web services (Sensor observation Service, O&M, etc.)
Encodings based on open standards

Decision Support Tools (monitoring, control, emergency response)



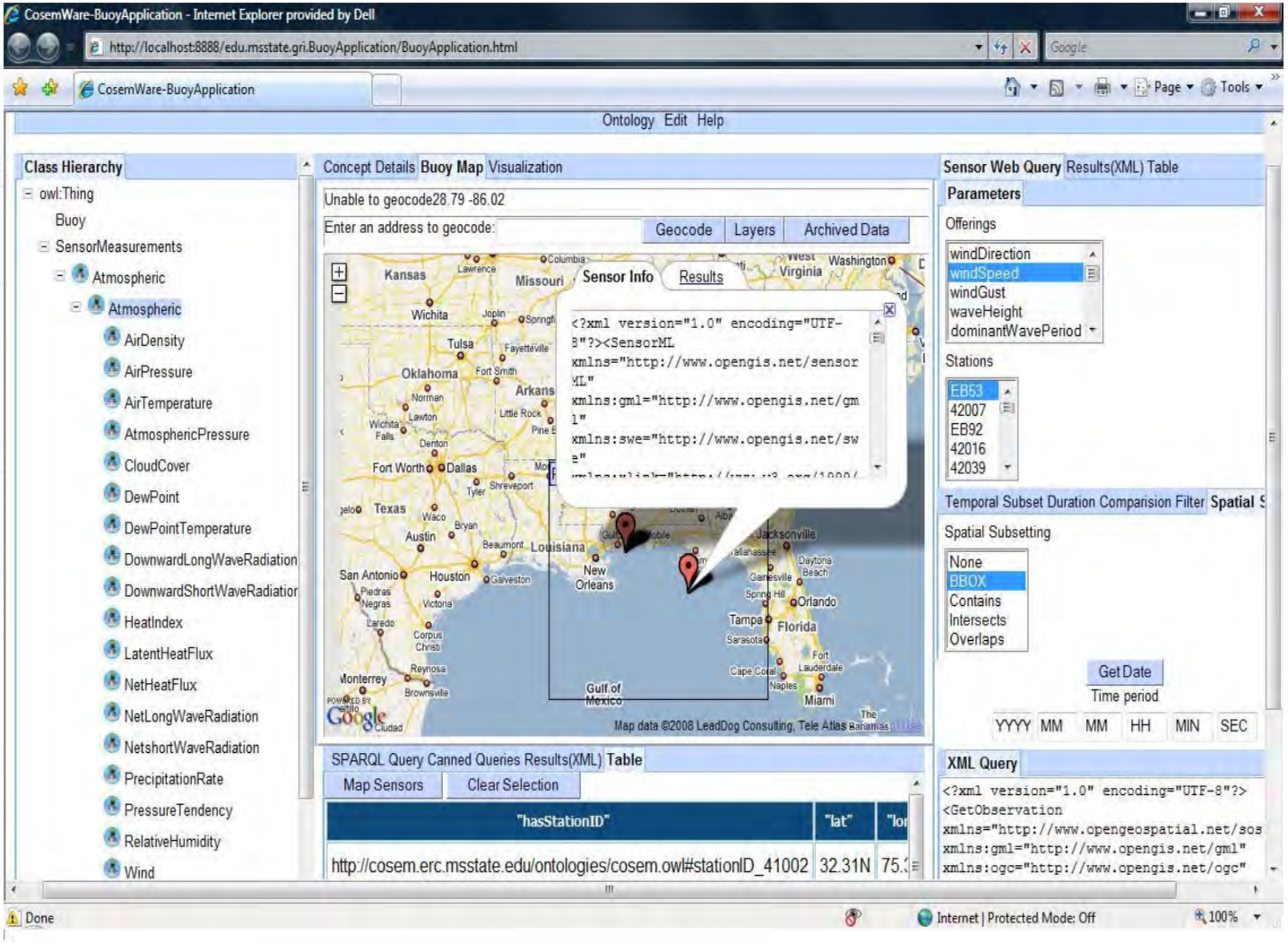
SWE Clients

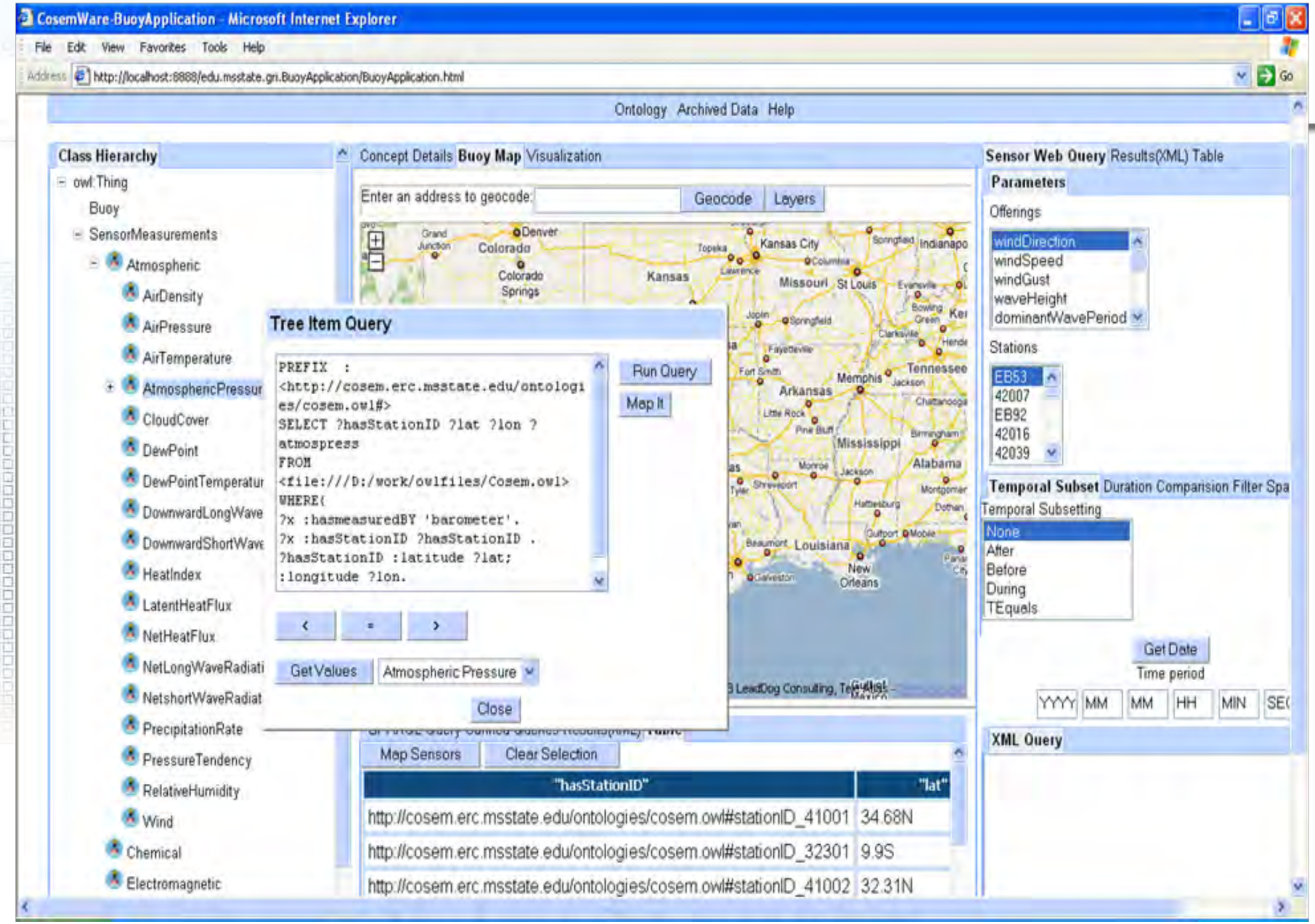
Sensor Web Architecture



Spatio- temporal data mining from the sensor web

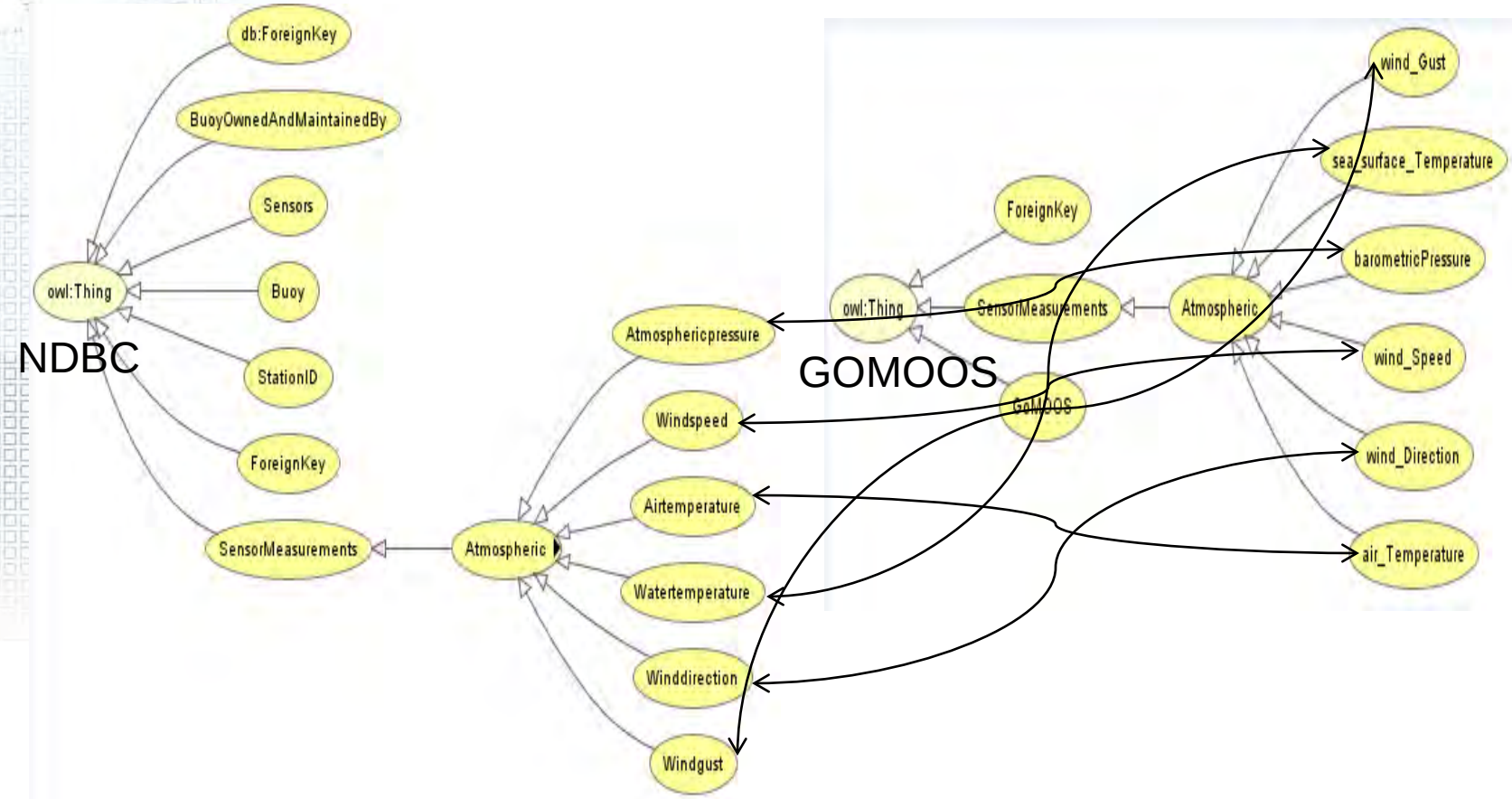
- **Spatial Subsetting**
  - bounding box Overlap, containing, intersection
- **Temporal sub-setting**
  - after a time instant, before time instant, during time instant, TEquals, past N sec/min/hrs/days
- **Filtering**
  - comparison operators such as Between, EqualTo, NotEqualTo, LessThan, GreaterThanEqualTo etc.





Example SPARQL Query (Scenario: "Find devices that can produce certain output variables")

Ontology Mapping



Source Ontology

Target Ontology

Semi-automated tool for Ontologies Alignment

- An instance-based approach combined with string distance metrics to achieve semantic interoperability among ontologies developed for different coastal sensor systems.
- Application of this methodology to determine the similarity between concepts of two ontologies by finding similarities between instances contained in each of the concepts.

- Instances
- Feature Extraction-kpca
- SVM-based Matching Algorithm
- String Distance Metrics

GRI

Mississippi State UNIVERSITY

Instance- based Matching

- In most of the ontologies the semantic relations between concepts is defined through the set of their instances.

Training Data:

- Given a source ontology, a set of data instances(30 days data) associated with each concept is collected and used as the training data for each concept in the ontology.
- The training set of data represented as feature vectors used to train the svm classifier to deal with the ontology mapping problem.

Training Data Set

Classes	#1	#2	#3	#4	#5	#6
Samples	300	300	300	300	300	300

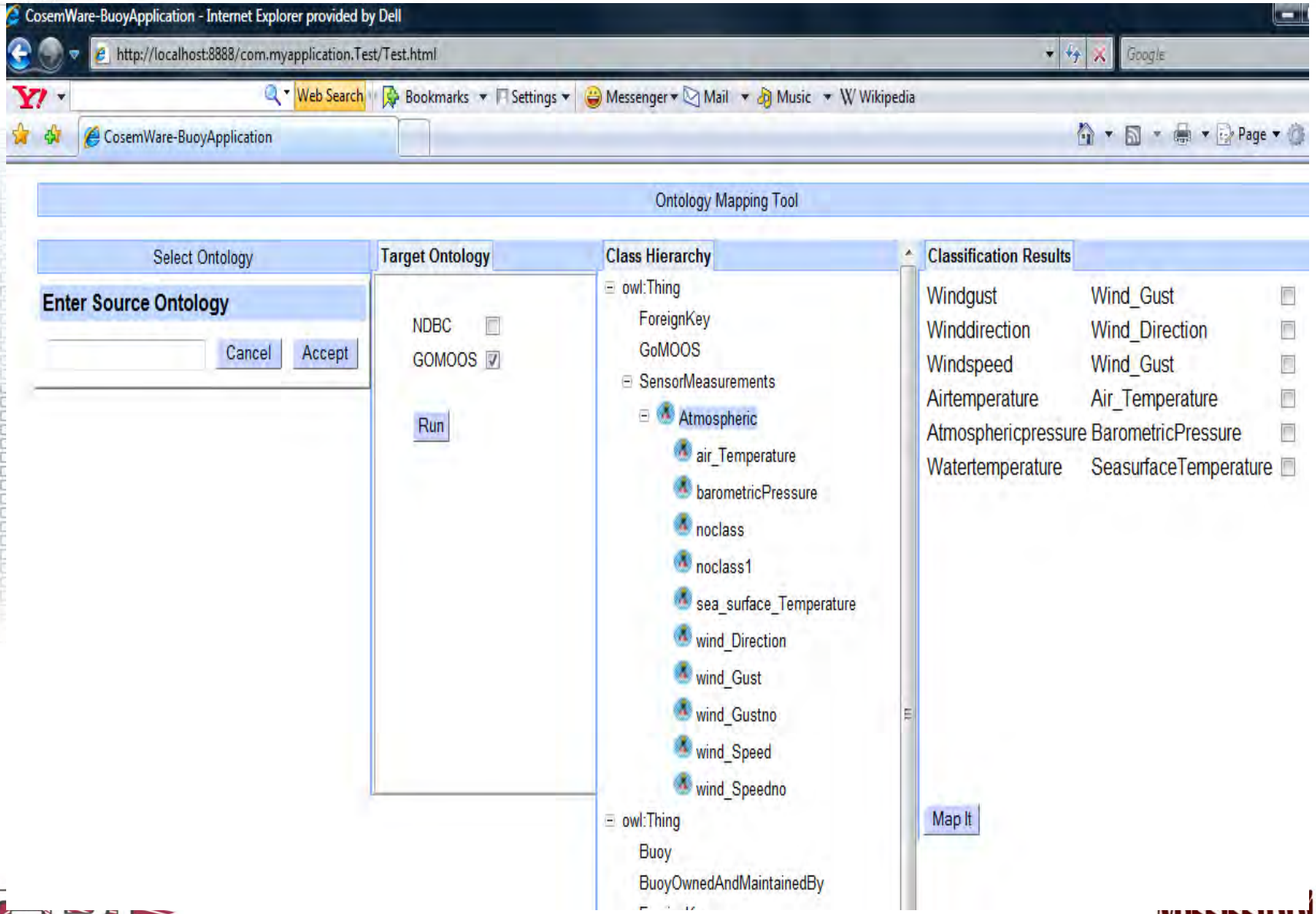
Support Vector Machines

- The major step of the methodology is to employ machine learning techniques to find ontology mappings.
- Support vector machine (SVM) is a popular and powerful technique for classifying any kind of data.
- The kernel principal components are then used as input to a SVM classifier for classification of each concept contained in the source ontology.
- SVM classifier trained with the kernel PCA components generates the model for source ontology.
- Given a target ontology, we then apply the trained SVM model to classify concepts in the target ontology.

String Distance Metrics

- String Distance Metrics are used to measure distances between all pairs of concepts contained in both the source and target ontologies.
- From the different number of available distance metrics, we use the best-performed distance metrics:Jaro-Winkler,Monge-Elkan,TFIDF, SFS and SoftTFIDF.
- For better matching results, a decision on matching concepts is based on the weights obtained from both SVM classifier and string based techniques.

Semi-Automated Tool



Semantic Matchmaking and Mobile Interface

- Mobile device technology with the influence of the Internet is enabling Web-based services so that people can have easy, 24-hour access from any location.
- The goal is to:
  - Enable flexible mobile access to distributed web resources for advanced personalization and localization features.
  - Automatic discovery and invocation of web services

- **Android** is a new and promising mobile platform based on the Linux operating system provided by Google.

- Android is not just another Java-based mobile platform but actually the only platform that adopts the results of the mobile middleware research.

Web Services

•Web services are modular, self-describing, and self-contained applications that are accessible over the internet.

• The core components of the Web services infrastructure are XML based standards like SOAP , WSDL and UDDI.

• SOAP and WSDL are designed to provide descriptions of message transport mechanisms, and for describing the interface used by each service.

•UDDI provides a registry of businesses and web service to describe services profiles with human-readable way.

• XML guarantees syntactic interoperability allowing Web Services to parse each others' messages but it is not enough for semantic understanding of the message content, as a result they do not know how to interpret them.

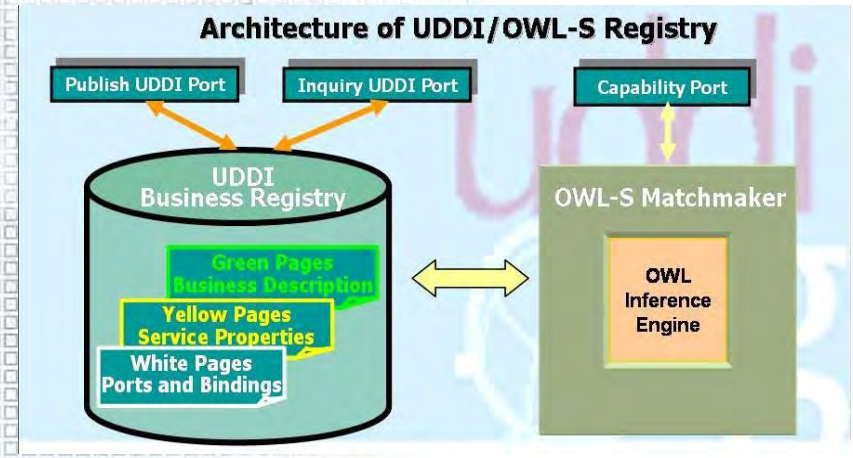
Semantic Web Services

- Semantic interoperability is crucial for Web services providing additional features like knowledge-based, location or context aware information.

- Integration of semantic metadata and the Web services infrastructure results in a service named Semantic Web Services (SWS) that has well-defined semantics.

- OWL-S provides a language to describe actual Web services that can be discovered and then invoked using standards such as WSDL and SOAP in such a way that the descriptions can be interpreted by a computer system in an automated manner.

Architecture of OWL-S / UDDI Matchmaker



- Discovery mechanism with OWL-S eliminates the problem caused with UDDI discovery mechanism which lacks an explicit capability representation and syntax based search producing results that are coarse in nature.

- Like UDDI, OWL-S allows a web service to describe using the classification schemes but with a capability-based description mechanism.
- OWL-S is characterized by three modules - Service Profile, Process Model and Grounding.

- In order to combine OWL-S and UDDI, we embed an OWL-S profile description in a UDDI data structure, and we augment the UDDI registry with an OWL-S Matchmaking component, for processing OWL-S profile information.

- N.Srinivasan, M. Paolucci, K.Sycara, Robotics Institute, Carnegia Mellon University, Adding OWL-S to UDDI, implementation and throughput.

Mapping between OWL-S Profile and UDDI

- The OWL-S/UDDI registry requires the embedding of OWL-S profile information inside UDDI advertisements.

- The mechanism uses a one-to-one mapping if an OWL-S profile element has a corresponding UDDI element, such as, for example, the contact information in the OWL-S Profile.

- For OWL-S profile elements with no corresponding UDDI elements, it uses a T-Model based mapping which is loosely based on the WSDL-to-UDDI mapping.

- It defines specialized UDDI T-Models for each unmapped elements in the OWL-S Profile like OWL-S Input, Output, Service Parameter and so on.

- OWL-S Profile is given as input from the mobile device for registering, this profile is register in the UDDI using the above mechanism.

Sensor Observation Service

LatLon
DateTime
SenosrType
Parameters

Search Sensor Network List

SensorNetworkList_Out

SensorNetworkList_In

Select Sensor

SensorSpecification
NearbySensorObs
SensorDataFormat_Out

SensorID

SensorDataFormat_In

Retrieve Data

DataAccess
ReservationID

Mapping OWL-S to UDDI

OWL-S Profile

service name=SOS-Service
textDescription=Sensor
Observation Service

Inputs
Outputs

Business Service

```
<businessServices>
<businessService serviceKey="">
<name>SOS-Service</name>
<description>Sensor Observation
Service</description>
<categoryBag>
Input_t-model
Output_t-model
</categoryBag>

</businessService>
</businessServices>
</businessEntity>
</save_business>
```

Mapping OWL-S Inputs to UDDI

OWL-S Profile

Inputs:
- LatLon
- Parameters
- DateTime
- SensorType
- SensorNetworkList
- SensorId
- SensorDataFormat

CatagoryBag:

Input_t-model

```
<keyedReference
keyName="http://localhost:8080/owlfiles/SOS-
                    Profile.owl#Parameters"
keyValue="http://localhost:8080/owlfiles/SensorConcepts.
                    owl#SensorParameters"
tModelKey="UUID:63194360-DA5E-11D7-A0CA-
                    000629DC0A53"/>
……..
<keyedReference
keyName="http://localhost:8080/owlfiles/SOS-
                    Profile.owl#SensorNetworkList_In"
keyValue="http://localhost:8080/owlfiles/SensorConcepts.
                    owl#SensorList"
tModelKey="UUID:63194360-DA5E-11D7-A0CA-
                    000629DC0A53"/>
```

Mapping OWL-S Outputs to UDDI

OWL-S Profile

Outputs:
- SensorNetworkList
- SensorSpecification
- NearbySensorObs
- SensorDataFormat
- DataAcess
- ReservationId

CatagoryBag:

Output_t-model

```
<keyedReference
keyName="http://localhost:8080/owlfiles/SOS-
                    Profile.owl#SensorSpecification"
keyValue="http://localhost:8080/owlfiles/SensorConcepts.
                    owl#SensorID"
tModelKey=" UUID:6E926D70-DA5E-11D7-A0CA-
                    000629DC0A53 "/>
……..
<keyedReference
keyName="http://localhost:8080/owlfiles/SOS-
                    Profile.owl#SensorDataFormat_Out "
keyValue="http://localhost:8080/owlfiles/SensorConcepts.
                    owl#SensorFormat"
tModelKey=" UUID:6E926D70-DA5E-11D7-A0CA-
                    000629DC0A53 "/>
```

12:19 PM
Google Map
Minnesota
Minneapolis
Wisconsin
South Dakota
Michigan
Toronto
Milwaukee
Detroit
Buffalo
Chicago
Cleveland
Iowa
Toledo
Nebraska
Omaha
Lincoln
Kansas City
Illinois
Indiana
Ohio
St. Louis
Cincinnati
Kansas
Missouri
Wichita
Kentucky
Virginia
Tulsa
Oklahoma
Arkansas
Tennessee
North Carolina
Atlanta
Mississippi
Alabama
South Carolina
Dallas
Texas
Georgia
Houston
Louisiana
San Antonio
Orlando
Florida
Monterrey
Miami
Gulf of Mexico
La Habana
Cuba
WMS Layer
MENU

suryad@gri.msstate.edu