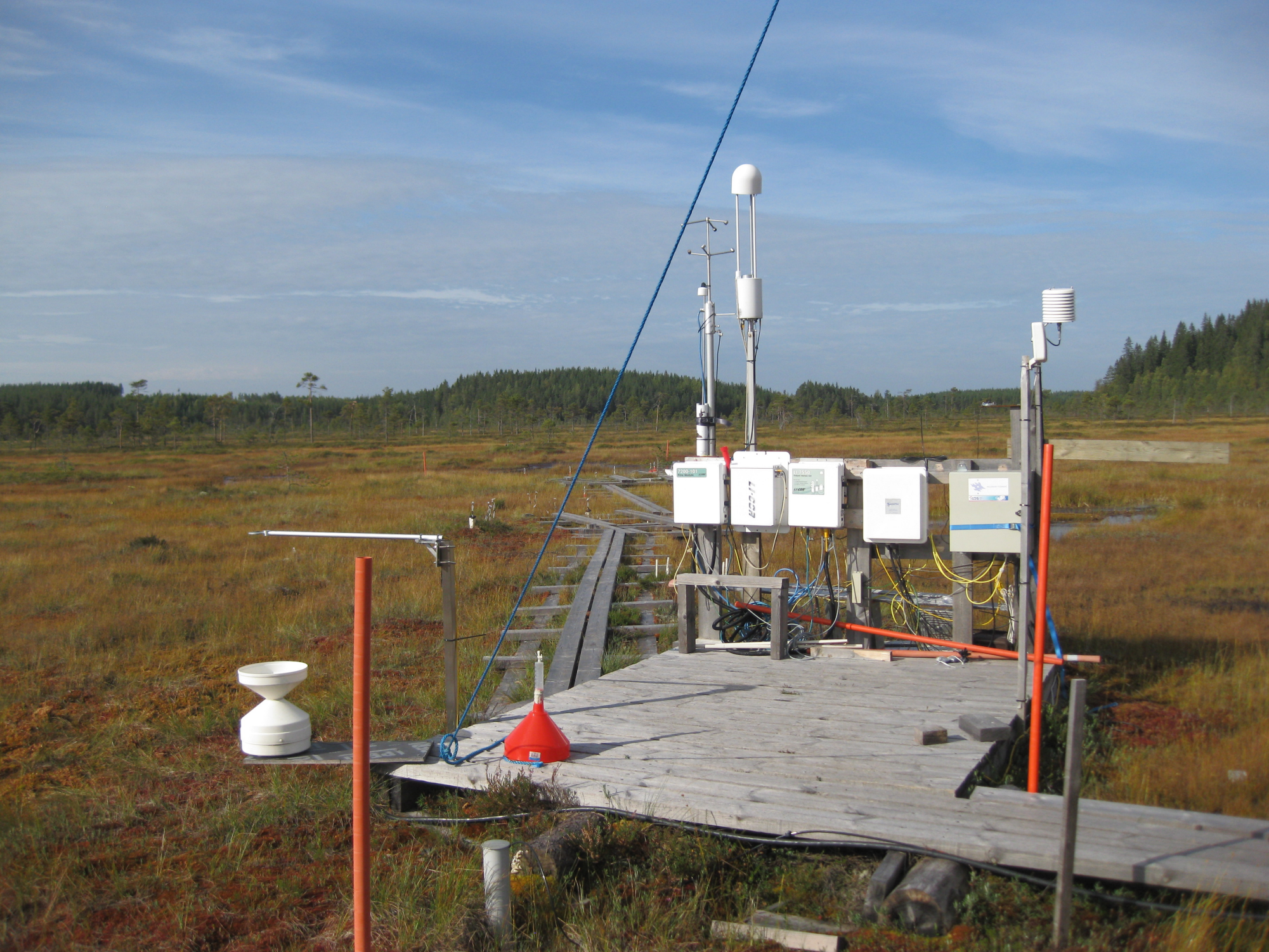
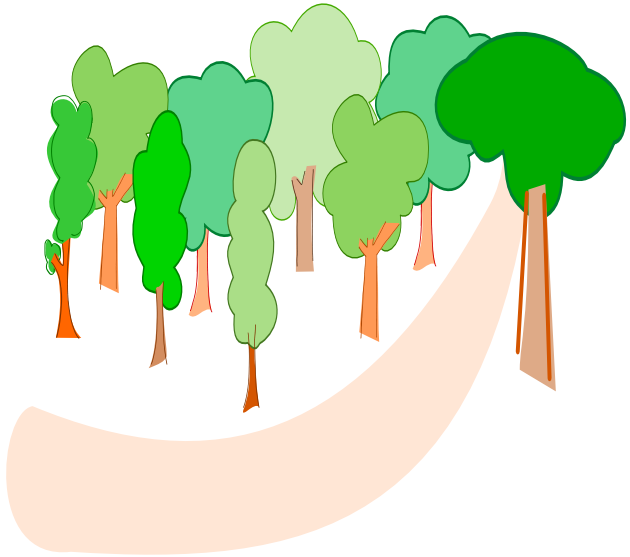
Technology Transfer Seminar
October 31, 2014, Vaisala, Helsinki

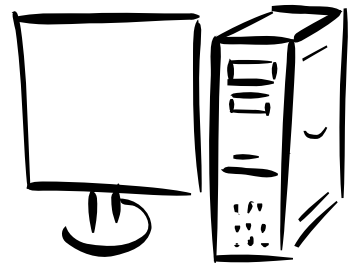
Situation awareness in environmental monitoring

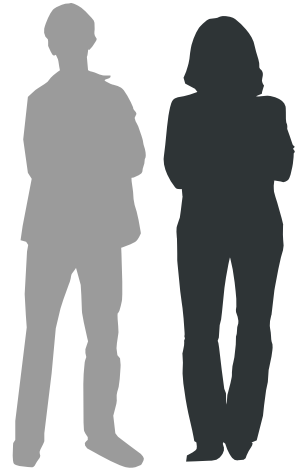
Markus Stocker

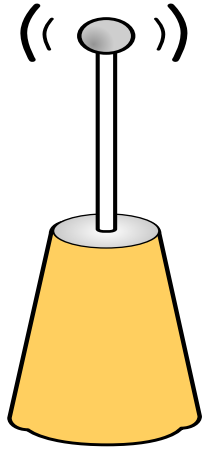
UNIVERSITY OF
EASTERN FINLAND



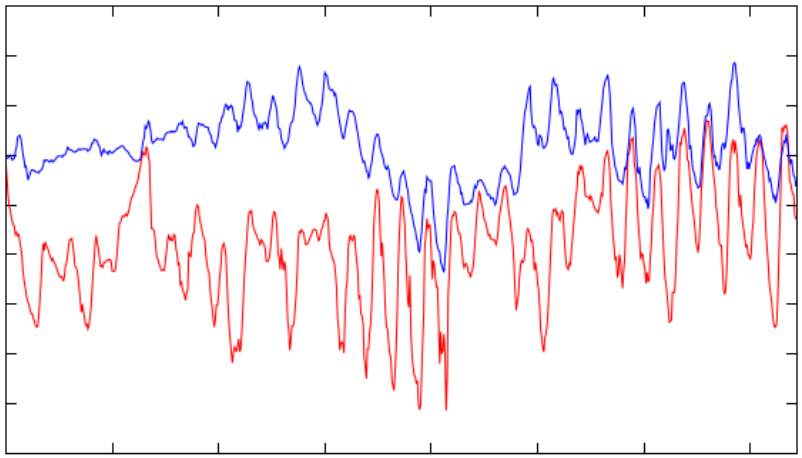


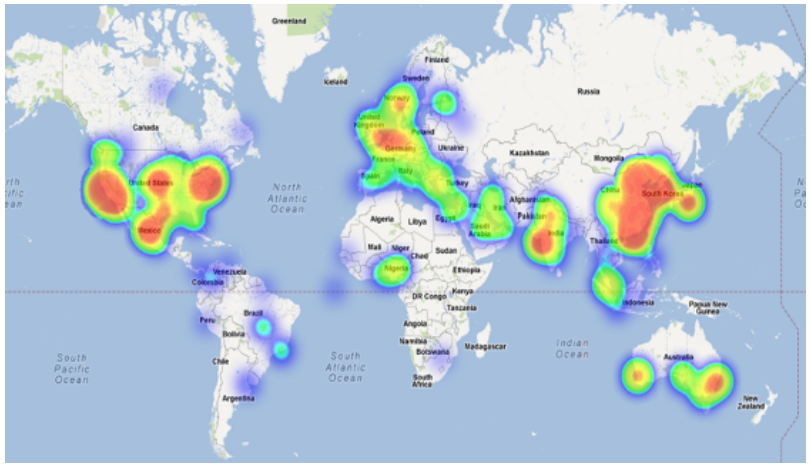


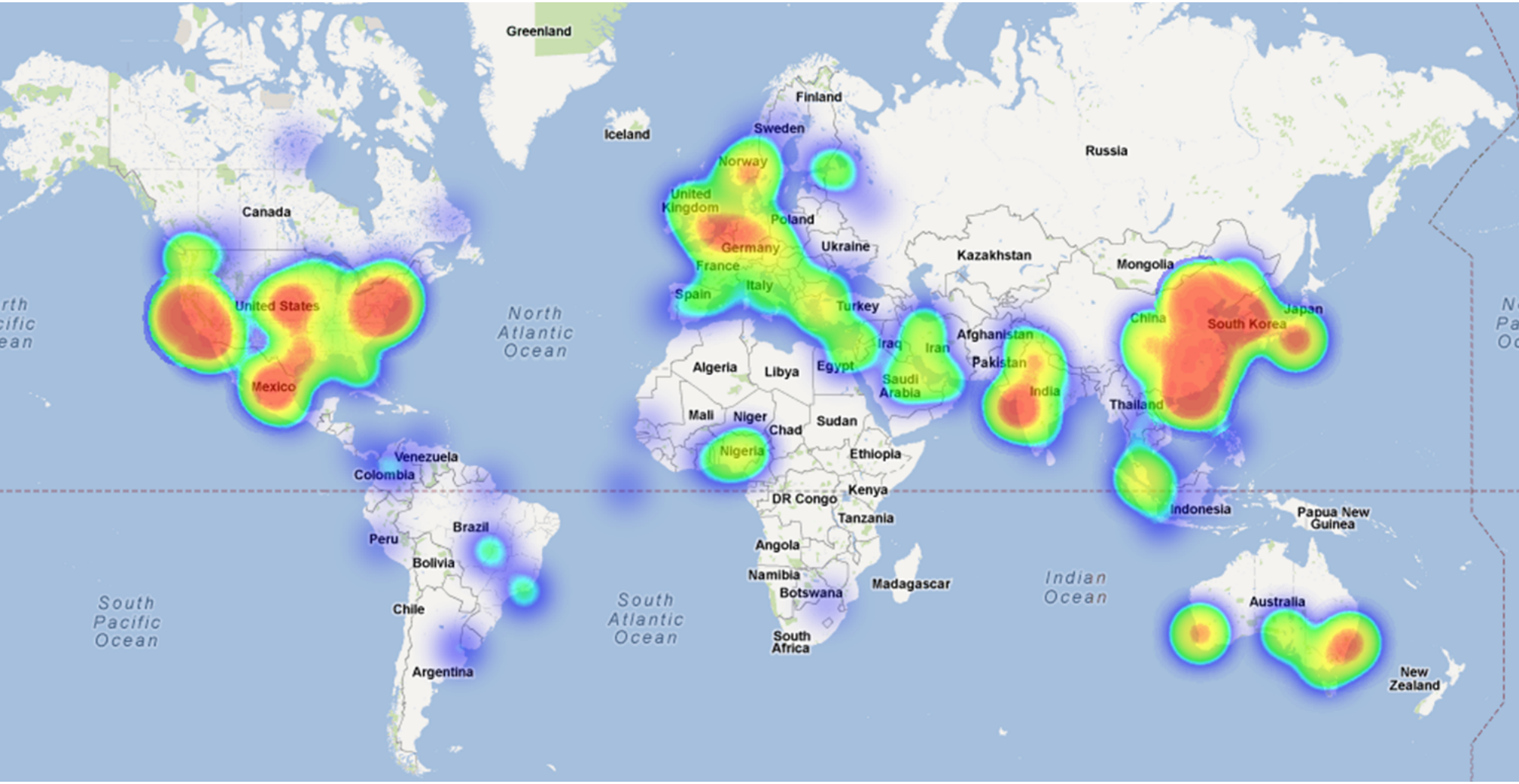


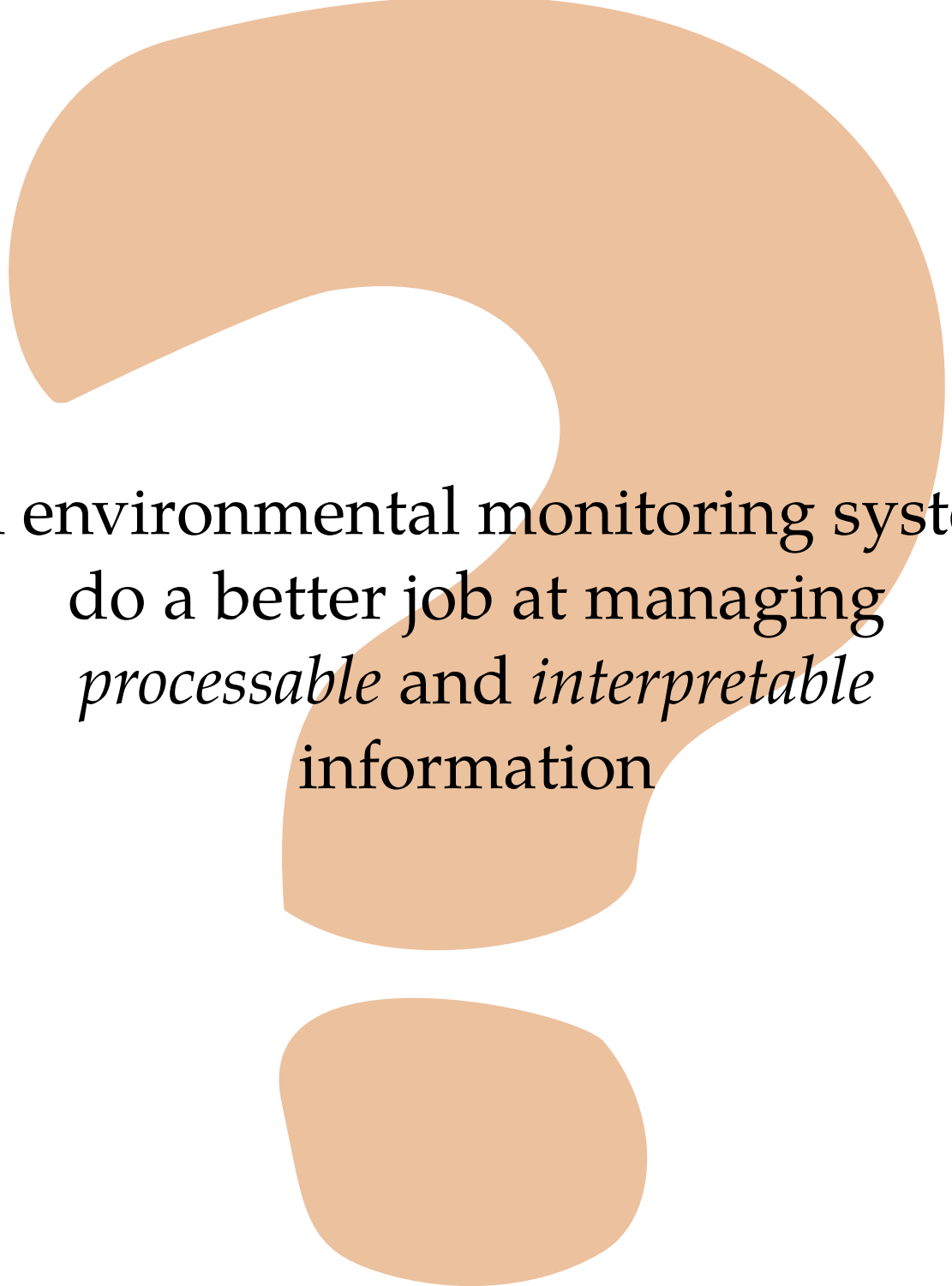


```
  7.3316302167824074e+005  6.7295417637024639e+002  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
1.1248345678061637e+003  7.6451655783892170e+002  6.6489916883398178e+002  5.0996188060151417e+002  5.3037337771849309e+002
2.5990152166875021e+002  2.6354791501170808e+002  2.6316883341227492e+002  3.3696093295963192e+002  3.2708190669952091e+002
3.9391899945110441e+002  3.2426846166900322e+002  2.0141200942601449e+002  1.4700101128227615e+002  4.0928587811142940e+001
1.2831620686413419e+001  0.0000000000000000e+000
  7.3316303024305555e+005  5.9954073981185900e+002  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
8.1548383245909952e+002  6.4755047038178782e+002  4.7342333956288417e+002  5.0497438804107304e+002  4.8678378833469156e+002
2.1558831592426984e+002  2.4469781600550951e+002  2.4637216889127018e+002  3.9338495014282574e+002  2.8801709503956096e+002
4.1110643983031497e+002  2.8396322218576239e+002  2.1375040618051182e+002  1.4957123471247169e+002  3.7975823102376701e+001
2.0382122539321763e+001  1.6943213617983506e+001
  7.3316303880787035e+005  6.4135476612374202e+002  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
7.9725650443997608e+002  4.0630047999881816e+002  3.5247269226184869e+002  3.7902085695621594e+002  2.8805461752290518e+002
1.8132905435068977e+002  2.6028214884093450e+002  1.6268304363454374e+002  3.3463583010398571e+002  2.0169500653114130e+002
4.6380154662366067e+002  2.6412454349776425e+002  2.0127541006127689e+002  1.3503244293786514e+002  2.3154902150570045e+001
7.6393422067327084e+000  1.7201363075124494e+001
  7.3316304687500000e+005  6.1036574617458143e+002  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
7.2736971215268670e+002  3.9684227323502876e+002  3.4903661050199901e+002  3.4531808153726166e+002  3.0686816846877008e+002
1.5193881810750614e+002  2.4133238139702240e+002  2.4228704440975204e+002  2.8825745672488563e+002  2.0412591958392611e+002
4.0962054312850012e+002  2.8308360749792962e+002  2.2063151348349507e+002  1.2405735268505646e+002  1.0751097756079318e+001
1.1881689141194544e+001  1.3547412000218912e+001
  7.3316305520833330e+005  6.4313854345743607e+002  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
8.0539618874151654e+002  3.8626777986640963e+002  2.8741638600588004e+002  2.5488895887853258e+002  2.6289312410222209e+002
1.6973869269222212e+002  1.7064856680396792e+002  2.6528938061547774e+002  2.7436008754172127e+002  2.3416327840132379e+002
4.5037513571727550e+002  3.4357911665879971e+002  2.1968465764114876e+002  1.0707033953281511e+002  4.2106918027684735e+001
0.0000000000000000e+000  2.5075705754016887e+001
  7.3316306354166672e+005  6.9018149795028319e+002  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000  0.0000000000000000e+000
1.0122740400440272e+003  4.7588170401009631e+002  3.6163701941444793e+002  3.8492799229831303e+002  3.1633325134373962e+002
1.8305150227202395e+002  2.2529123956614126e+002  2.0836586101380522e+002  3.1634735061210313e+002  2.5690547494848136e+002
3.6567505629777969e+002  3.0461779733996269e+002  1.8490080930780519e+002  1.1544775603278390e+002  4.1043905610232025e+000
1.3740708174213207e+001  2.9403296138810457e+001
```




Greenland
Iceland
Finland
Sweden
Norway
Russia
Canada
United States
Mexico
Ukraine
Kazakhstan
Mongolia
China
South Korea
North Atlantic Ocean
Algeria
Libya
Egypt
Afghanistan
Pakistan
India
Thailand
Mali
Niger
Chad
Sudan
Ethiopia
Nigeria
Kenya
DR Congo
Tanzania
Angola
Namibia
Botswana
Madagascar
South Africa
Venezuela
Colombia
Peru
Brazil
Bolivia
Chile
Argentina
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Indonesia
Papua New Guinea
Australia
New Zealand


Greenland
Iceland
Finland
Sweden
Norway
Russia
United Kingdom
Poland
Canada
Germany
Ukraine
Kazakhstan
Mongolia
France
Spain
Italy
Turkey
China
South Korea
Japan
United States
North Atlantic Ocean
Afghanistan
Iraq
Iran
Pakistan
Algeria
Libya
Egypt
Saudi Arabia
India
Thailand
Mexico
Mali
Niger
Chad
Sudan
Nigeria
Ethiopia
Venezuela
Colombia
Kenya
DR Congo
Tanzania
Indonesia
Papua New Guinea
Brazil
Peru
Angola
Bolivia
Namibia
Madagascar
Indian Ocean
Botswana
Australia
South Pacific Ocean
South Atlantic Ocean
Chile
South Africa
Argentina
New Zealand



Can environmental monitoring systems do a better job at managing *processable* and *interpretable* information

Approach

The recipe

- Get the data from somewhere
  - Sensors, files, middleware, database, ...
- Get rid of its heterogeneity
  - Bring it into some canonical form
  - Often that's some database schema
  - Sometimes there is an XML schema (e.g. MMEA)
- Process the data
  - Generally necessary to achieve anything useful
  - Even simple things like hourly average
- Obtain information from data
  - Lot's of ways, manually and automatically
- Represent information
  - So that it is processable and interpretable

The technologies

- The obvious ones ...
  - The math, stats, and related software packages
  - At least one programming language
- Machine learning
  - Handy to extract information from data
  - But depends on the application
  - Other computational models are cool too
- Ontologies
  - Oh no, that word again
  - Let's skip it
- Situation theory
  - Situation what?

Situation theory

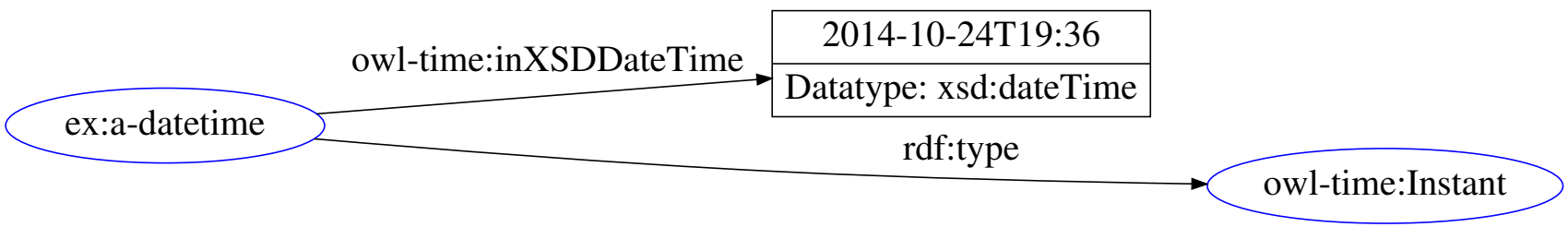
- Useful to describe what is occurring in the world
- In a monitored environment, in particular
- Situation is a structured part of the world
- What objects exist in a situation?
- When and where do they exist?
- What are their attributes?
- How do they relate to each other?

Ontologies and related technologies

- (Ask me for a definition if you want one.)
- Support you in teaching yourself what you already know
- And along the way teach some of it to computers
- Useful ontologies here
  - Semantic Sensor Network
  - RDF Data Cube Vocabulary
  - Situation Theory Ontology
  - OWL-Time, GeoSPARQL, PROV-O

Resource Description Framework (RDF)

- Data model at the base of semantic technologies
- Statement
  - Triple consisting of <subject, predicate, object>
  - Subject is a resource, entity referred to by URI
  - Predicate is a property, referred to by URI
  - Object is the value of the property, URI or literal
- Set of statements is a RDF graph



RDF serialization (RDF/XML)

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl-time "http://www.w3.org/2006/time#" >
]>

<rdf:RDF
  xmlns:owl-time="http://www.w3.org/2006/time#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<rdf:Description rdf:about="http://example.org#a-datetime">
  <owl-time:inXSDDateTime rdf:datatype="&xsd;dateTime">2014-10-24T19:36</owl-time:inXSDDateTime>
</rdf:Description>

<rdf:Description rdf:about="http://example.org#a-datetime">
  <rdf:type rdf:resource="&owl-time;Instant"/>
</rdf:Description>

</rdf:RDF>
```

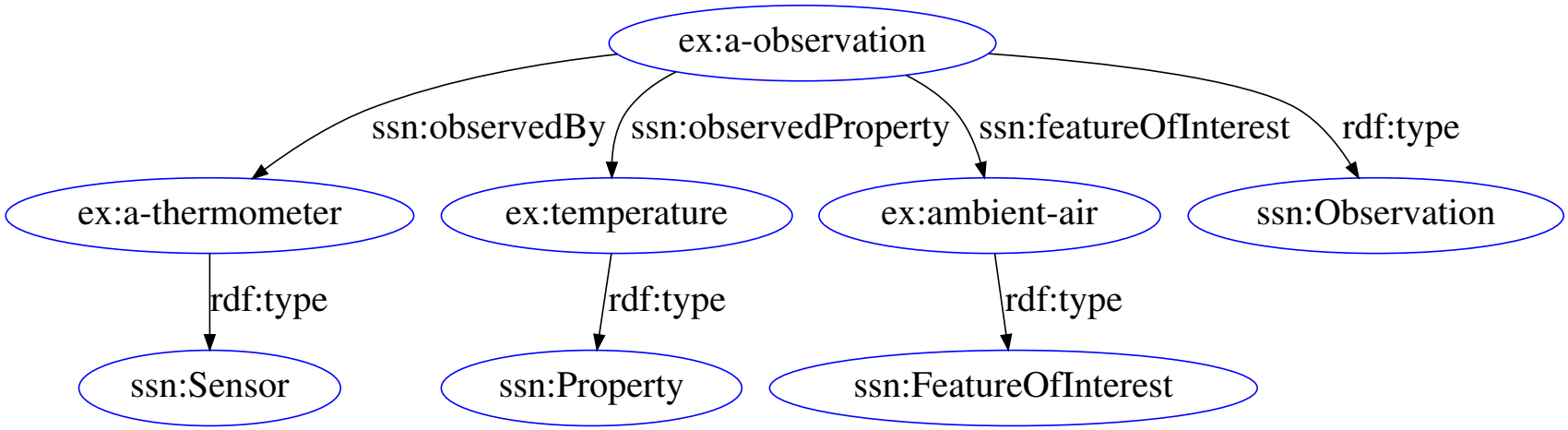
RDF serialization (N-Triples)

```
<http://example.org#a-datetime>
  <http://www.w3.org/2006/time#inXSDDateTime>
    "2014-10-24T19:36"^^<http://www.w3.org/2001/XMLSchema#dateTime> .
<http://example.org#a-datetime>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://www.w3.org/2006/time#Instant> .
```

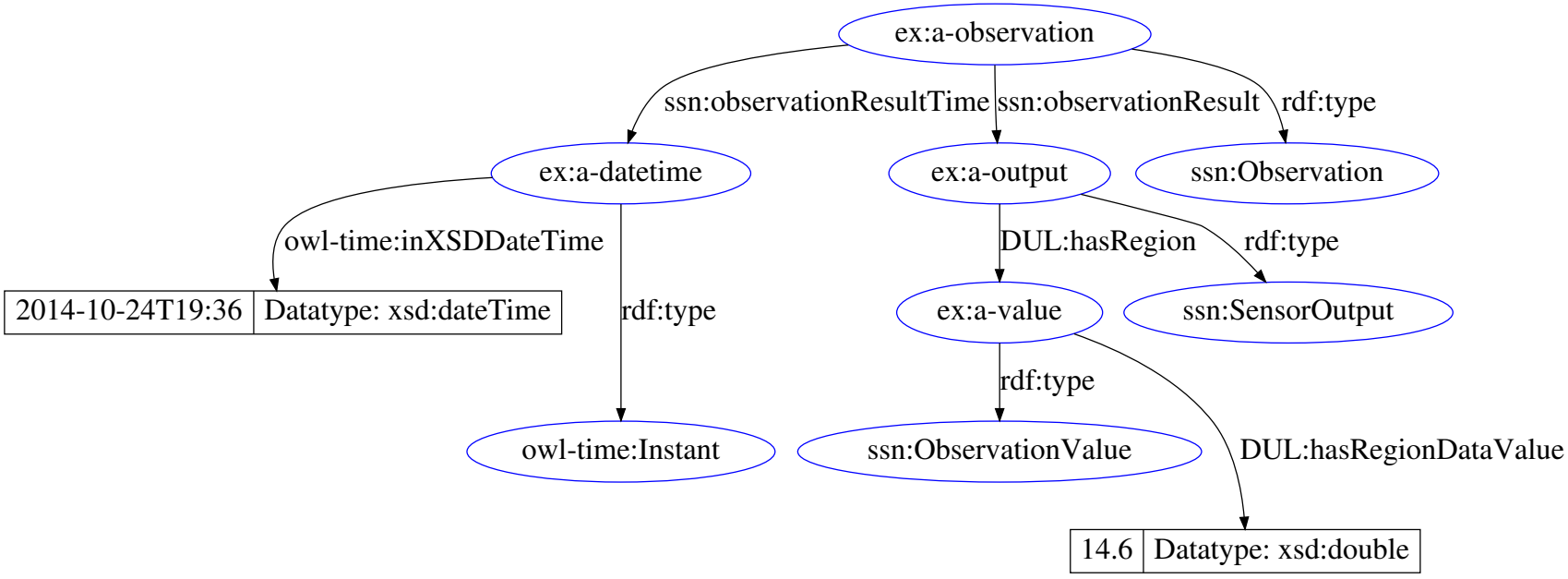
Semantic Sensor Network: Features

- Sensors
  - The sensing method they implement
  - Their capabilities, e.g. range of operation
  - On what platform they are installed
- Observations
  - Property of feature observed
  - Sensor that made the observation
  - When (and where) the observation was made
  - The observation value obtained in measurement
  - The quality of observation

Semantic Sensor Network: Example



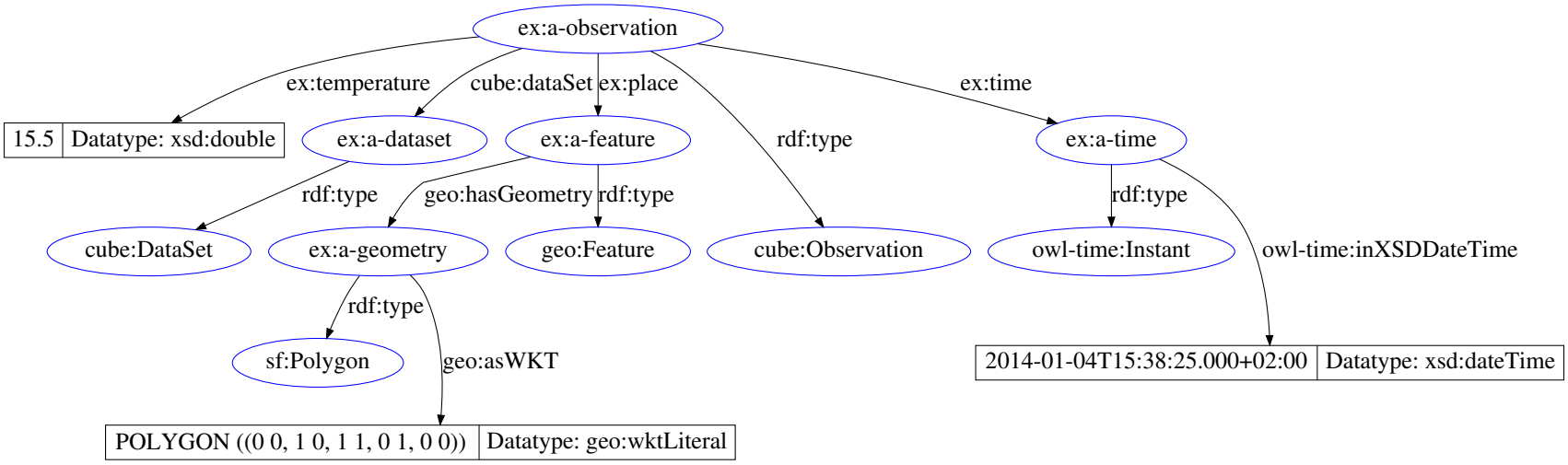
Semantic Sensor Network: Example (cont'd)



RDF Data Cube Vocabulary: Features

- Datasets
  - Define the structure of a dataset
  - Specify the components (the “columns”)
  - Including the property and metadata (e.g. order)
- Observations
  - The “rows” of a dataset
  - Relate to a dataset
  - And to component property values (the “cells”)
  - According to dataset structure definition

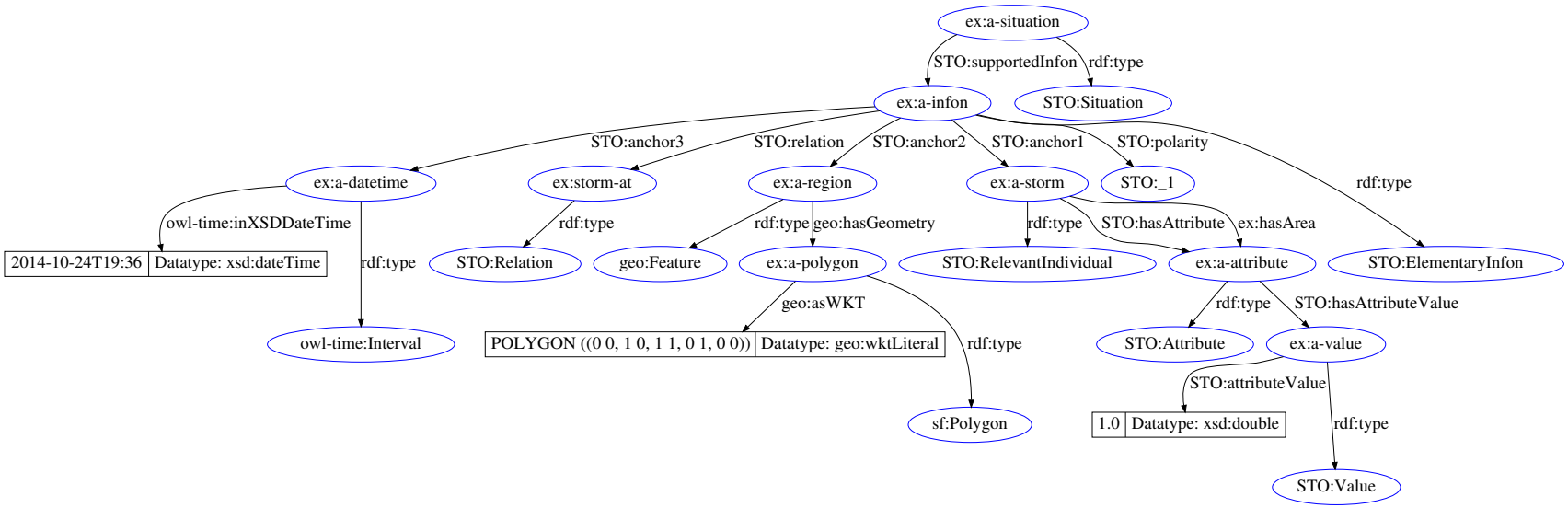
RDF Data Cube Vocabulary: Example

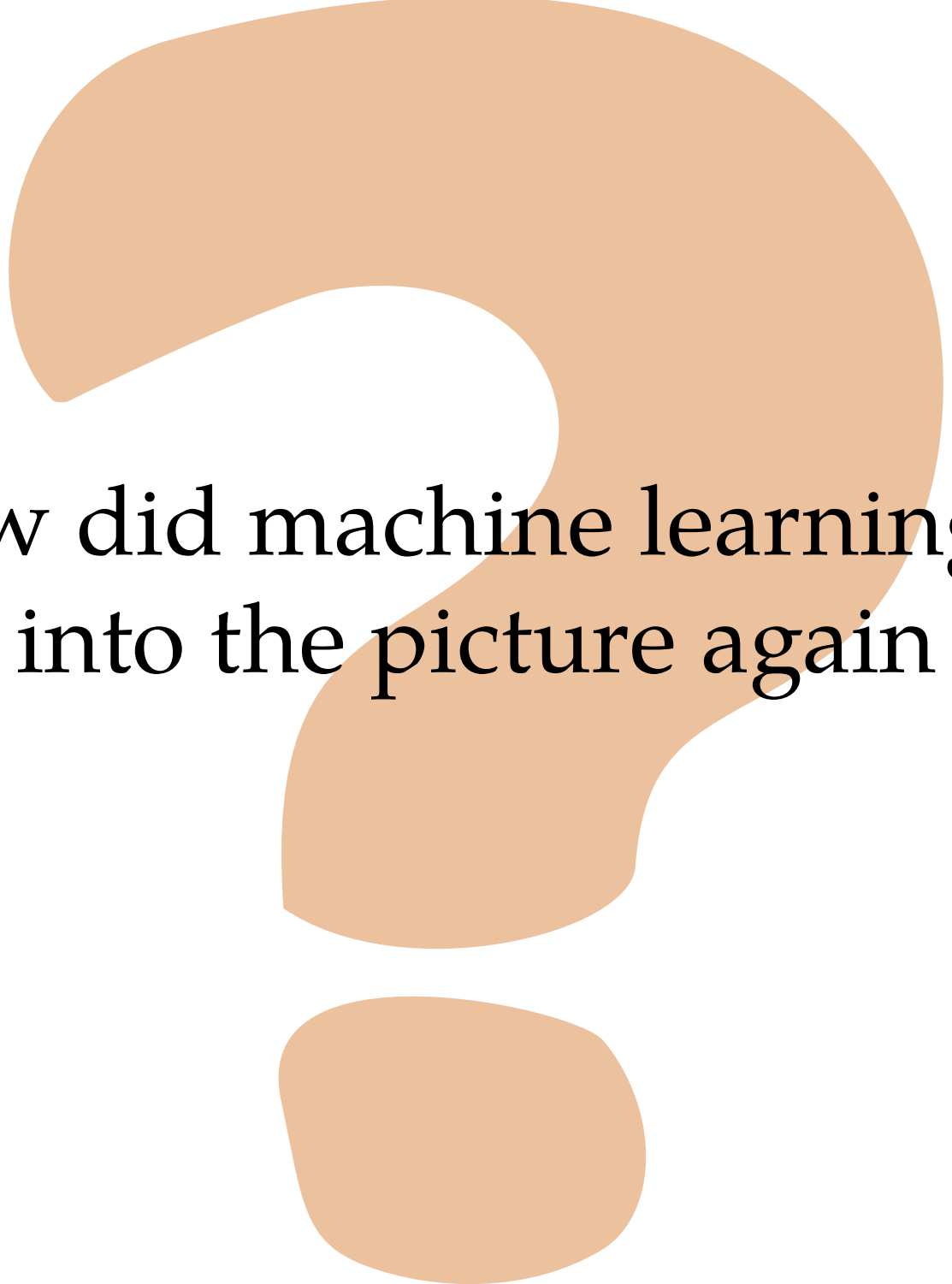


Situation Theory Ontology: Features

- A situation is said to support infons
- Infons are tuples consisting of
  - Relation
  - Set of objects
  - Polarity [0/1] (“truth value”)
- Object are relevant to the relation
- Objects can have attributes
- Infons can also have attributes
- A situation can be an object

Situation Theory Ontology: Example

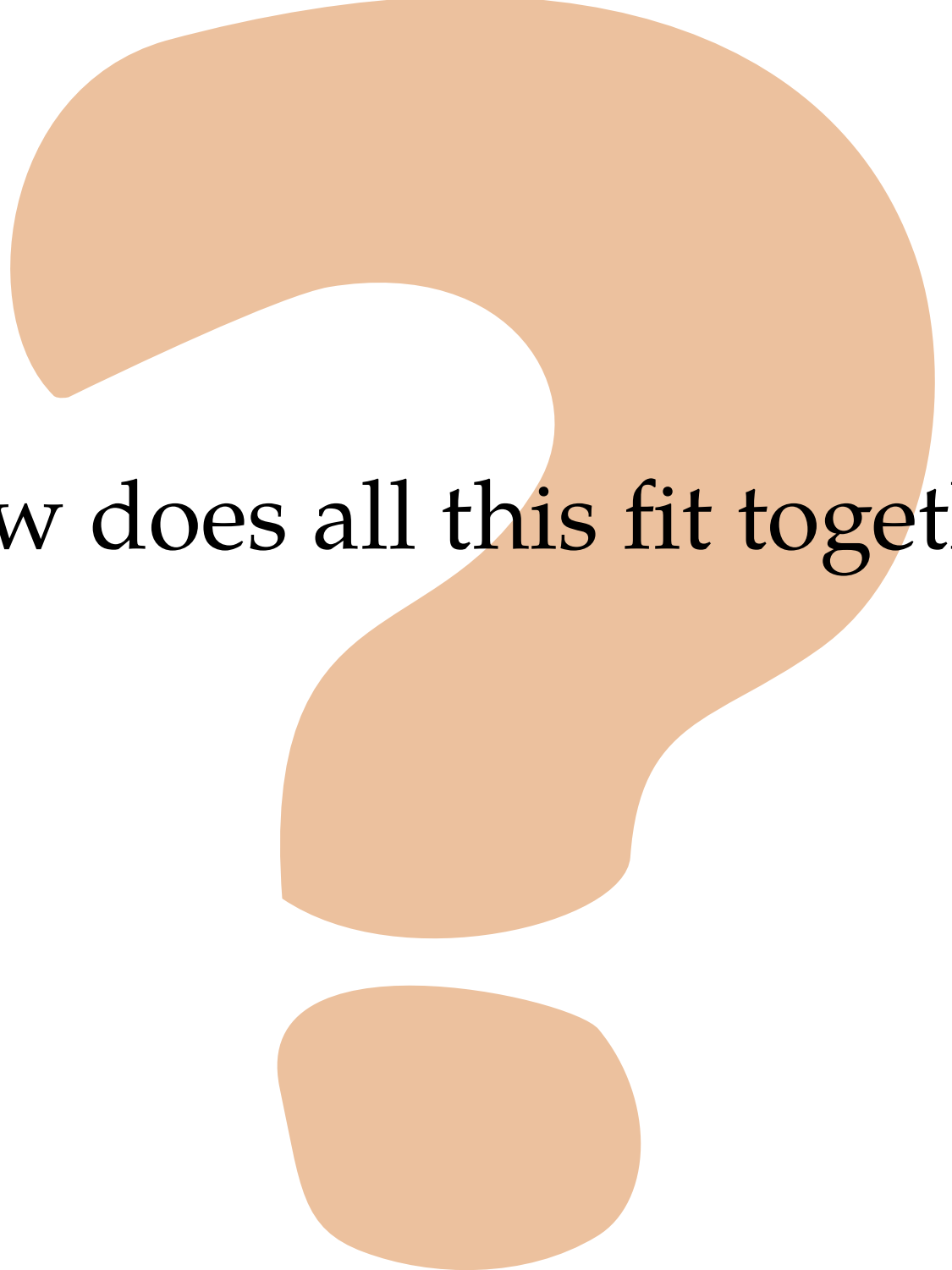




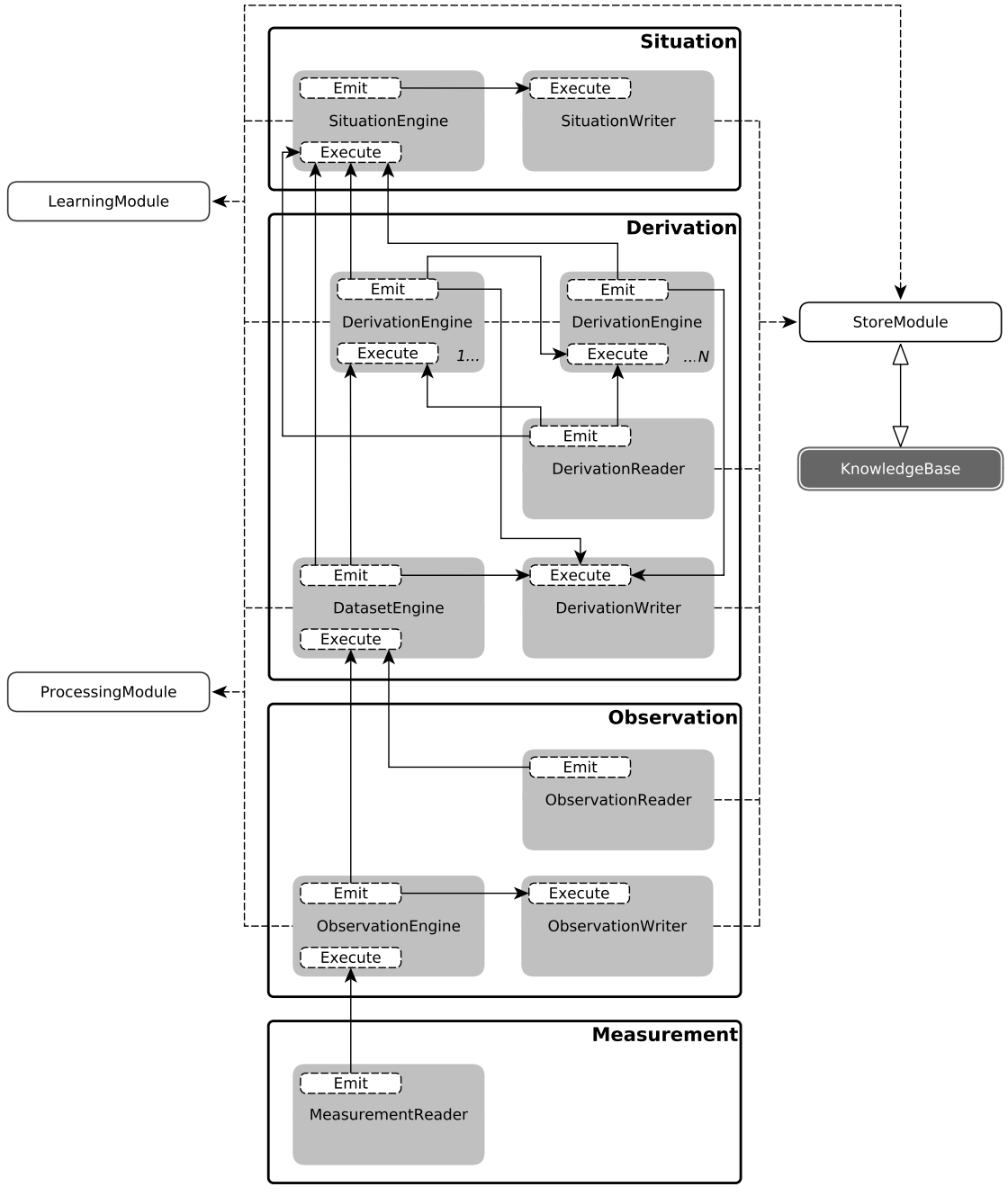
How did machine learning fit into the picture again

Computational models

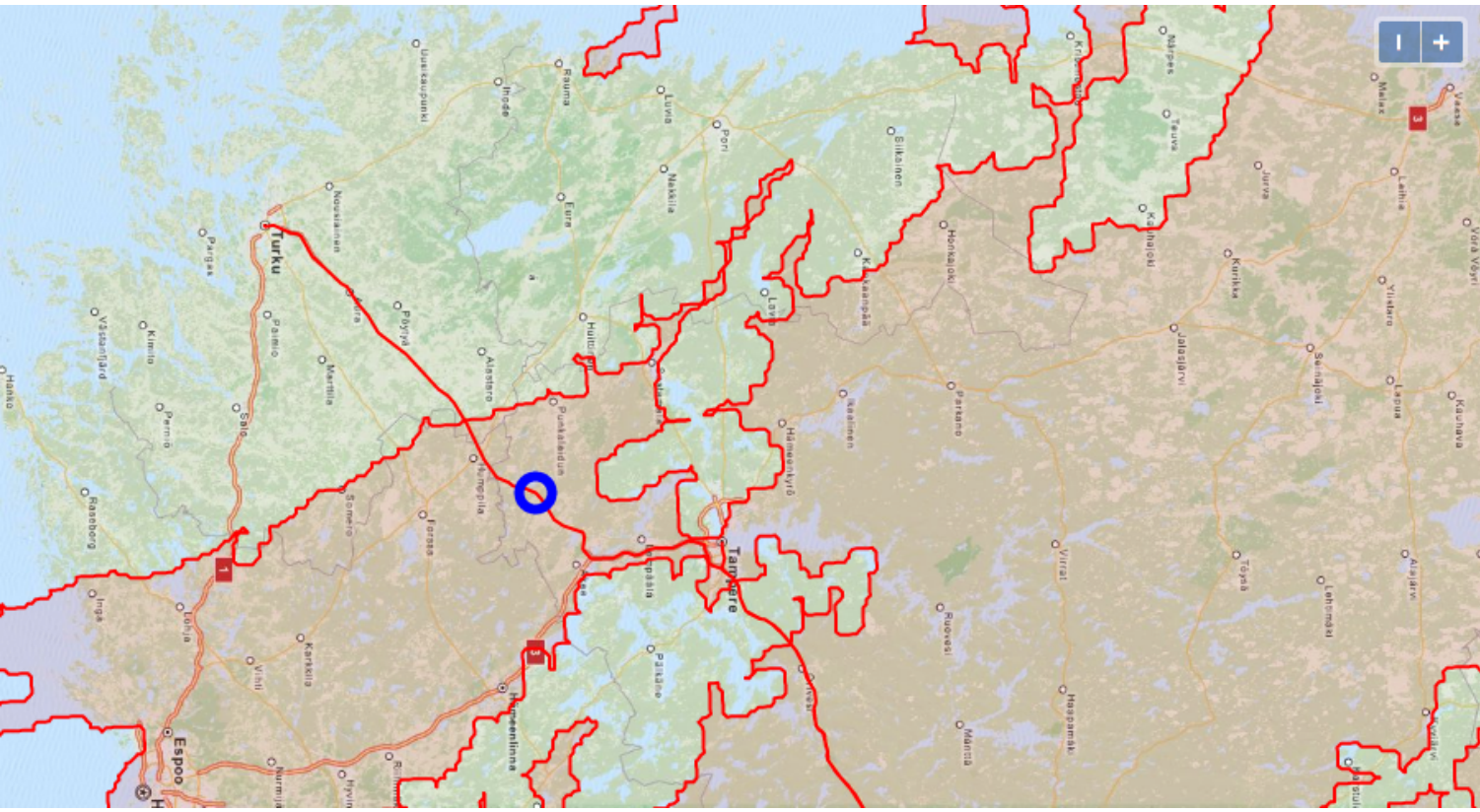
- Sit between processed data and represented information
  - Feed on data objects and return information objects
  - Objects in information about situations
- Useful for information *extraction*
  - Let software automatically do the work
  - Virtually necessary in some applications
  - Hardly possible in others
- Large family
  - Machine learning
  - Complex event processing
  - Mechanistic models



How does all this fit together


Situation
Emit
SituationEngine
Execute
Execute
SituationWriter
LearningModule
Derivation
Emit
DerivationEngine
Execute
1...
Emit
DerivationEngine
Execute
...N
StoreModule
Emit
DerivationReader
KnowledgeBase
Emit
DatasetEngine
Execute
Execute
DerivationWriter
ProcessingModule
Observation
Emit
ObservationReader
Emit
ObservationEngine
Execute
Execute
ObservationWriter
Measurement
Emit
MeasurementReader

Example

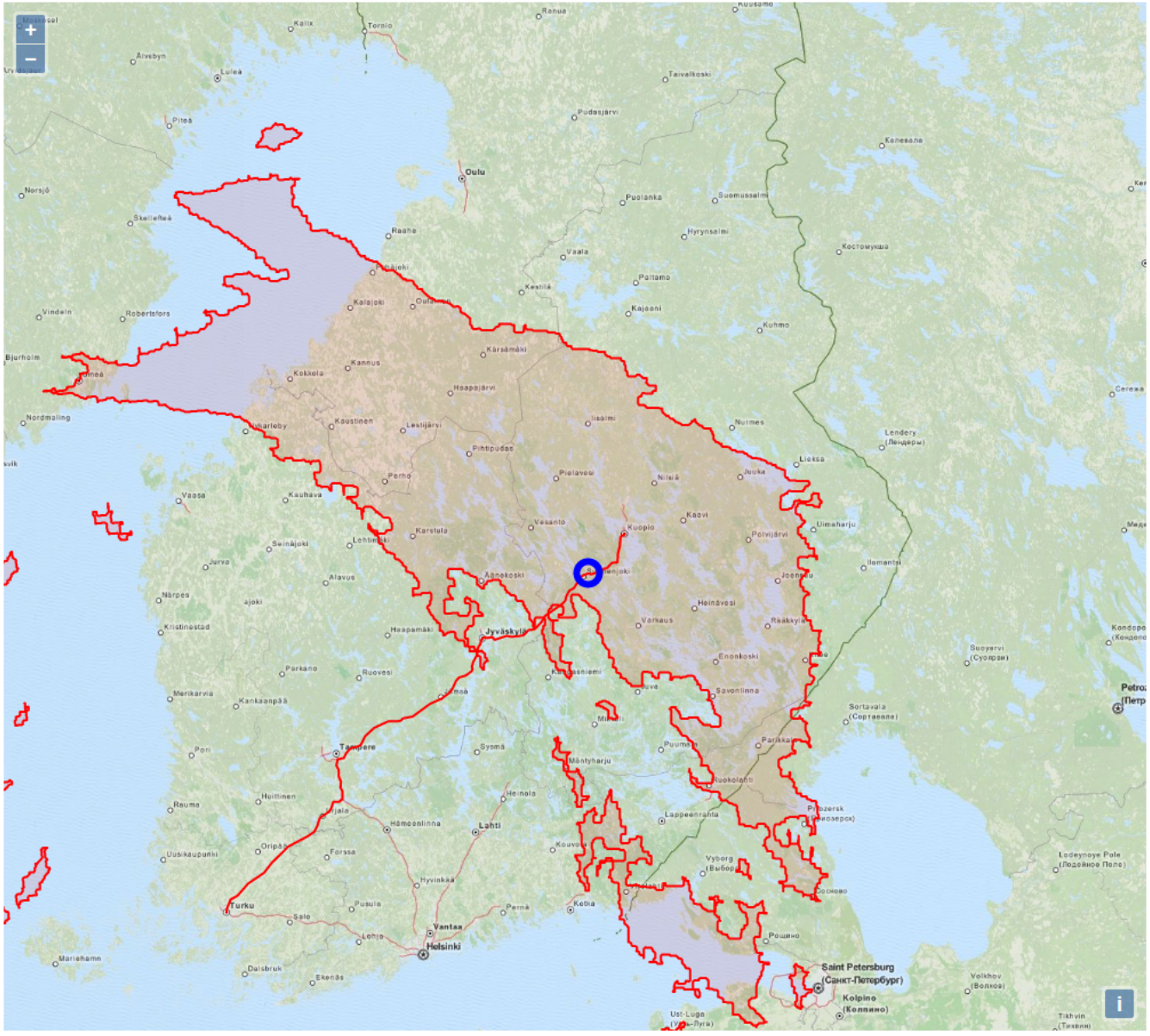


Processing

- Radar data obtained from FMI Open Data
- Octave script processes data
- Identify storm polygon contours
- Represent polygon as WKT
- Create MMEA observation message
- Includes timestamp and all polygons at that time
- Messages are processed to situations
- Situations are stored to Profium Sense

Directions

- Users can add directions
- Origin, destination, departure time
- Upon registration of a direction
  - Fetch route from Google Directions
  - Gives also estimated time of arrival
  - Fetch situation time steps in interval
  - Compute situations for driver location



New Direction

Origin

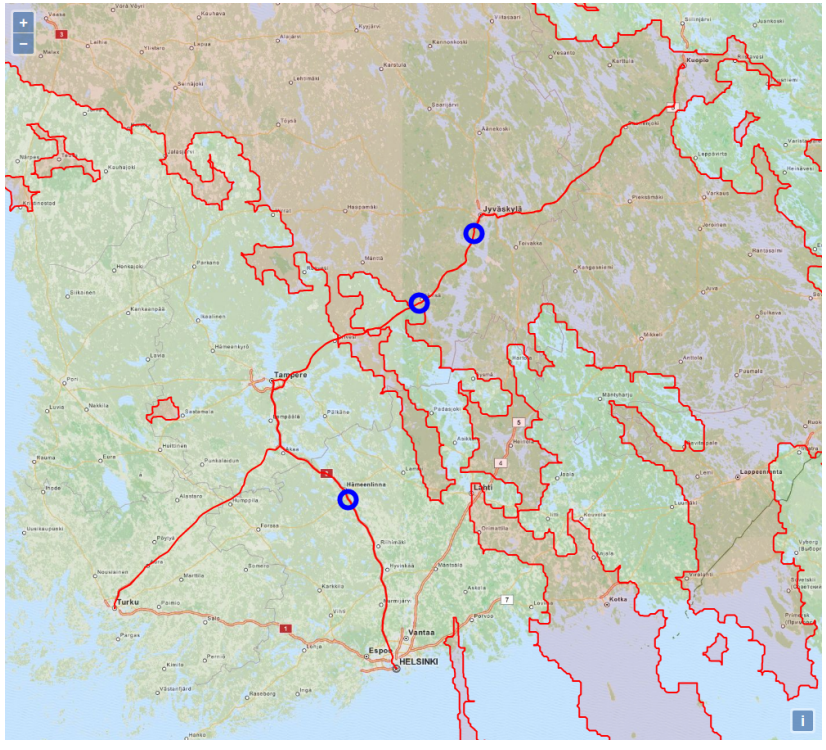
Destination

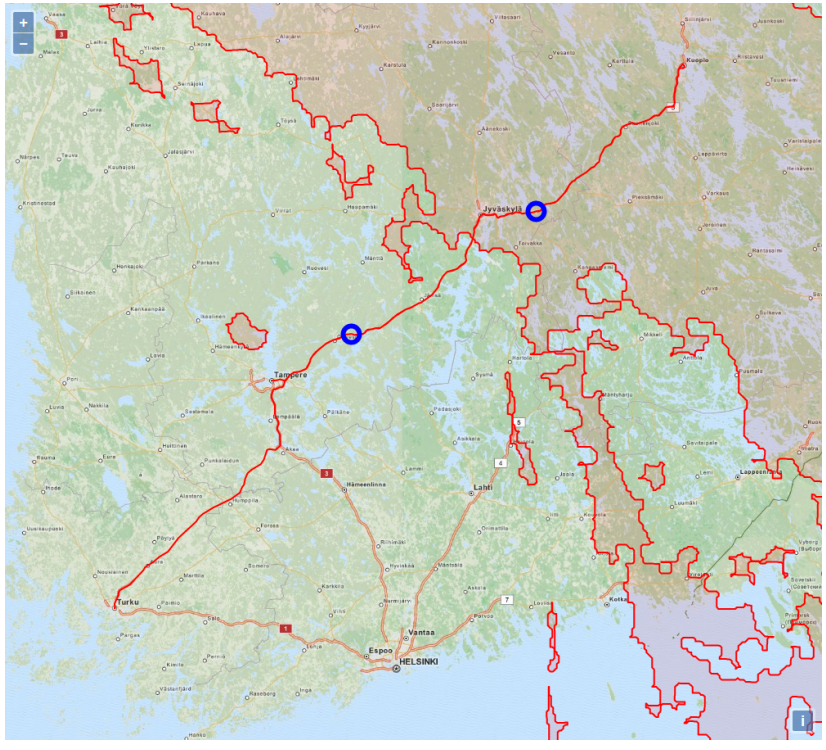
Departure time

add clear

Simulation

Current Time	Start time	End time	Overlapping		
08-10-2014 17:50	08-10-2014 13:00	08-10-2014 18:00	no	run	clear


Jyväskylä
Tampere
Hämeenlinna
Lahti
Turku
Vantaa
Espoo
HELSINKI
Kotka
Kuopio
Lappeenranta


Jyväskylä
Tampere
Lahti
Turku
Vantaa
Espoo
HELSINKI
Kotka

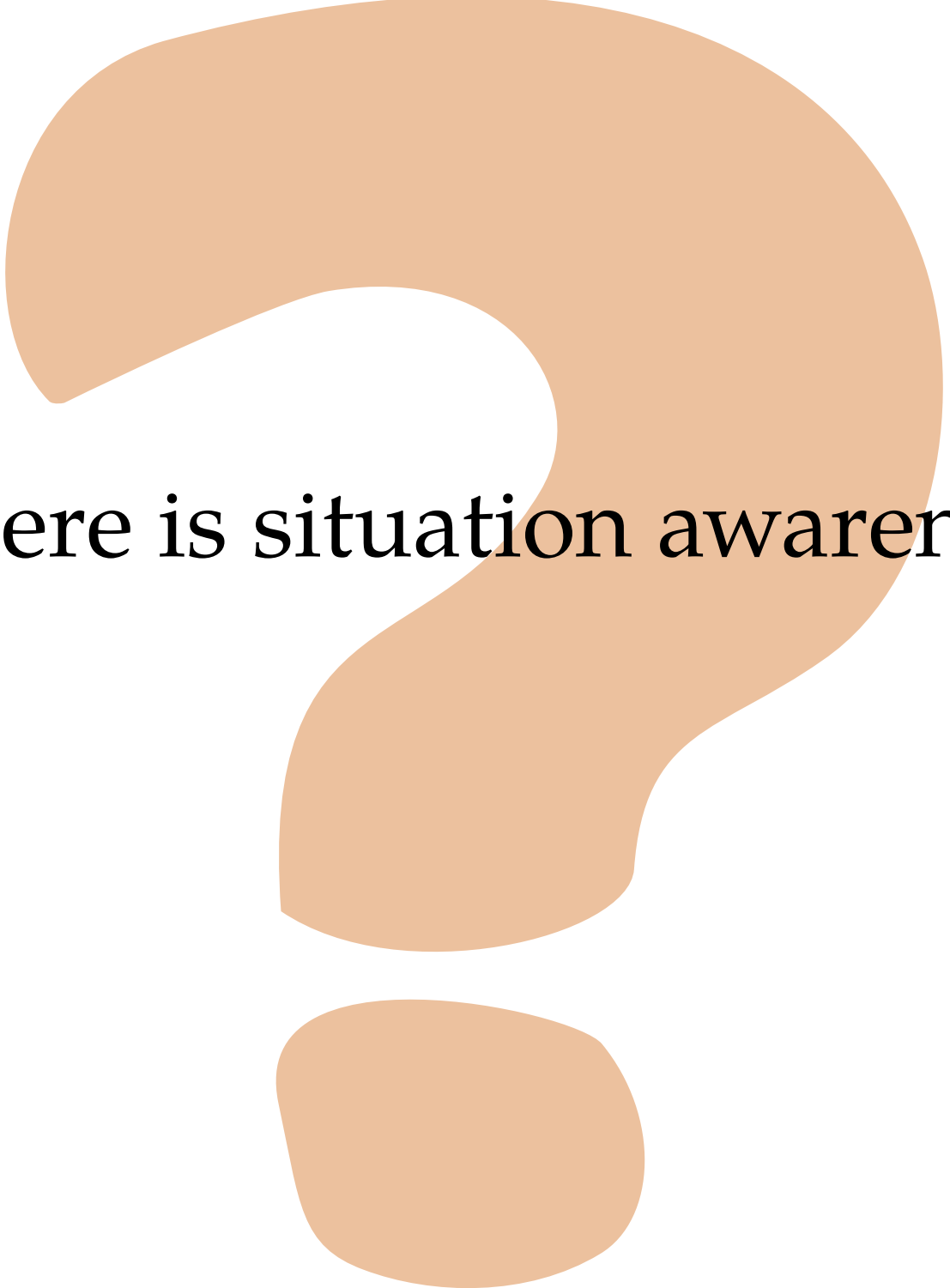
Query situations

```
prefix ns: <http://cleen.mmea.com/storm#>
prefix sto: <http://vistology.com/ont/2008/STO/STO.owl#>
prefix time: <http://www.w3.org/2006/time#>

select ?relation ?time ?location
where {
  [
    rdf:type sto:Situation ;
    sto:supportedInfon [
      sto:relation ?relation ;
      sto:anchor2 [ time:inXSDDateTime ?time ] ;
      sto:anchor3 [ profium:location ?location ]
    ]
  ]
  filter (?time > "2014-10-15T08:30:00.000+03:00"^^xsd:dateTime)
  filter (?relation = ns:storm-at || ?relation = ns:driver-at)
}
```

Query situations (cont'd)

```
...
where {
[
  rdf:type sto:Situation ;
  sto:supportedInfon [
    sto:relation ns:storm-at ;
    sto:anchor1 [ ns:hasArea [ sto:hasAttributeValue [ sto:attributeValue ?area ] ] ] ;
    sto:anchor2 [ time:inXSDDateTime ?time1 ] ;
    sto:anchor3 [ profium:location ?location1 ] ;
  ]
]
[
  rdf:type sto:Situation ;
  sto:supportedInfon [
    sto:relation ns:driver-at ;
    sto:anchor1 [ ns:direction [ profium:location ?direction ] ] ;
    sto:anchor2 [ time:inXSDDateTime ?time2 ] ;
    sto:anchor3 [ profium:location ?location2 ] ;
  ]
]
  filter (?time1 = ?time2)
  filter (profium:inside(?location2, ?location1))
}
```



Where is situation awareness

Situation awareness (SA)

- Three level model useful for system architecture
  - Perception: sensor data assimilation
  - Comprehension: data processing, information extraction
  - Projection: spatio-temporal situation reasoning
- Distributed SA also useful
  - Traditionally, SA obtained and maintained by experts
  - Here SA is shared among system components
  - Technical subsystem can obtain and maintain SA
  - Social subsystem can refine SA
  - Monitoring system SA greater than sum of parts?

Take aways

- Environmental monitoring system
  - Environmental-sociotechnical system
- Technical subsystem can obtain and maintain SA
  - Using computational methods
  - Data assimilation and processing
  - Information extraction and representation
- SA shared among social and technical subsystems
  - Ontology acts as interface
- Environmental monitoring system with own SA
- Representation of information content of maps, figures, ...
  - Explicitly, i.e. accessible and processable by computers