SensorML / OGC Sensor Web Enablement Notes

Sensor Standard Harmonization
March 7 2006

Dr. Mike Botts
Earth System Science Center, NSSTC
University of Alabama in Huntsville
Huntsville, AL 35899

mike.botts@uah.edu

SensorML

What is SensorML?

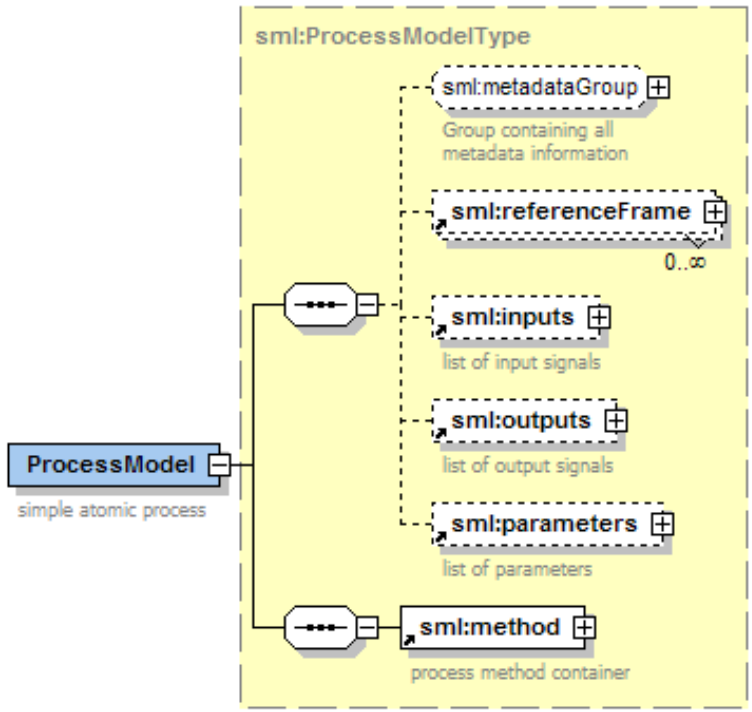
- **Models and XML encodings for describing processes**
  - Detectors, actuators, etc. are modeled as processes
  - Can be used to describe “left-side” processes (“how were these observations obtained?”) and “right-side” processes (“what can I derive from these observations?”)
- **Vision:**

  - **Discovery of transducers and processes / plug-n-play sensors** – SensorML is the means by which sensors and processes make themselves and their capabilities known
  - **Observation lineage** – SensorML provides sensing and processing history of observations; supports quality knowledge of observations
  - **On-demand processing** – SensorML supports on-demand derivation of higher-level information (e.g. geolocation or products) without a priori knowledge of the sensor system
  - **Intelligent, autonomous sensor network** – SensorML enables the development of taskable, adaptable sensor networks
  - **Network and software friendly** – SensorML is ideal for distributing processing demands throughout the network/command hierarchy
  - **Extensibility** – SensorML provides easy means for meeting various community needs

SensorML Concepts - ProcessModel

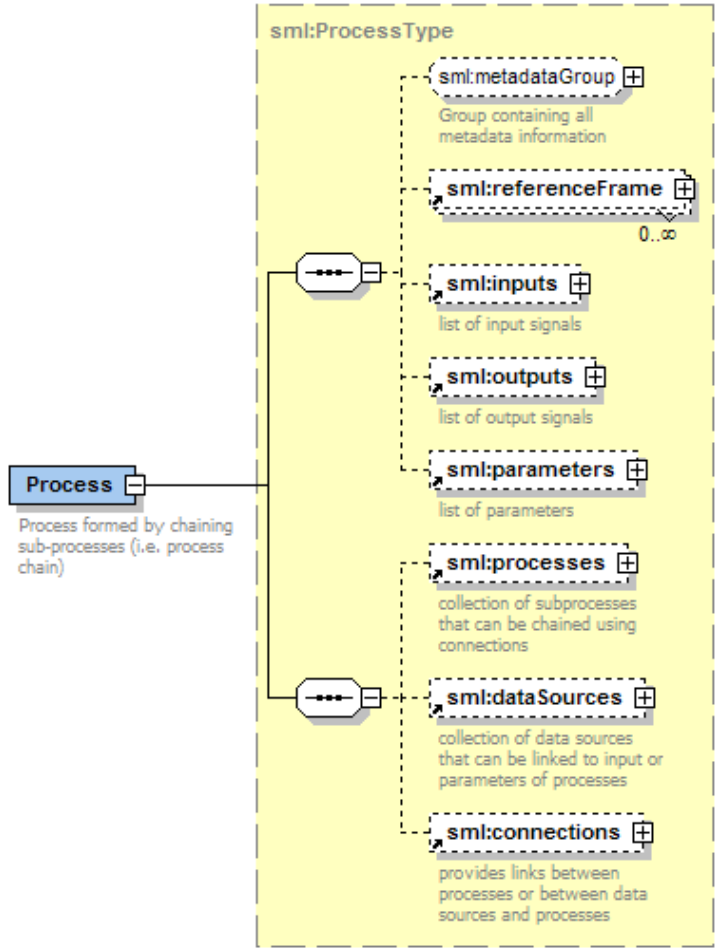
In SensorML, everything is modeled as a Process

- **ProcessModel**
  - defines atomic process modules (detector being one)
  - has five sections
    - metadata
    - inputs, outputs, parameters
    - method
  - Inputs, outputs, and parameters defined using SWE Common data definitions

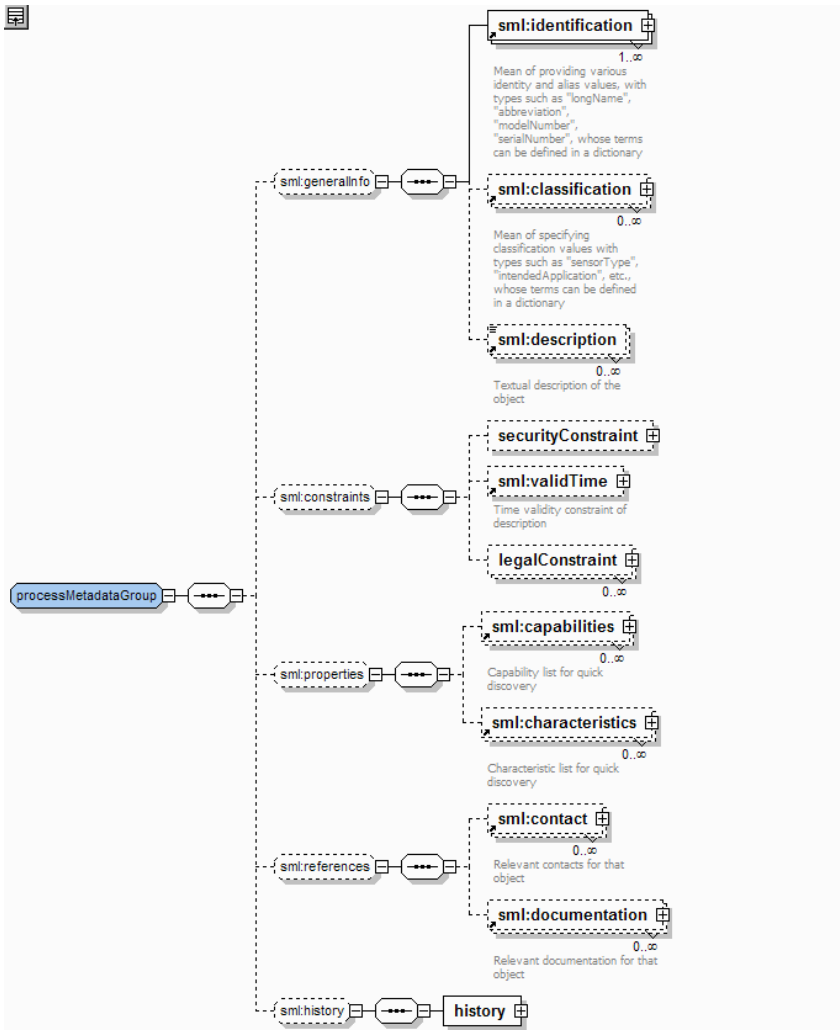


SensorML Concepts

- **Process**
  - defines a process chain
  - includes:
    - metadata
    - inputs, outputs, and parameters
    - processes (ProcessModel, Process)
    - data sources
    - connections between processes and between processes and data
- **System**
  - defines a collection of related processes along with positional information



Process/Sensor/System Metadata



Metadata is primarily for discovery and assistance, and not typically used within process execution

- **Includes**
  - Identification, classification, description
  - Security, legal, and time constraints
  - Capabilities and characteristics
  - Contacts and documentation
  - History

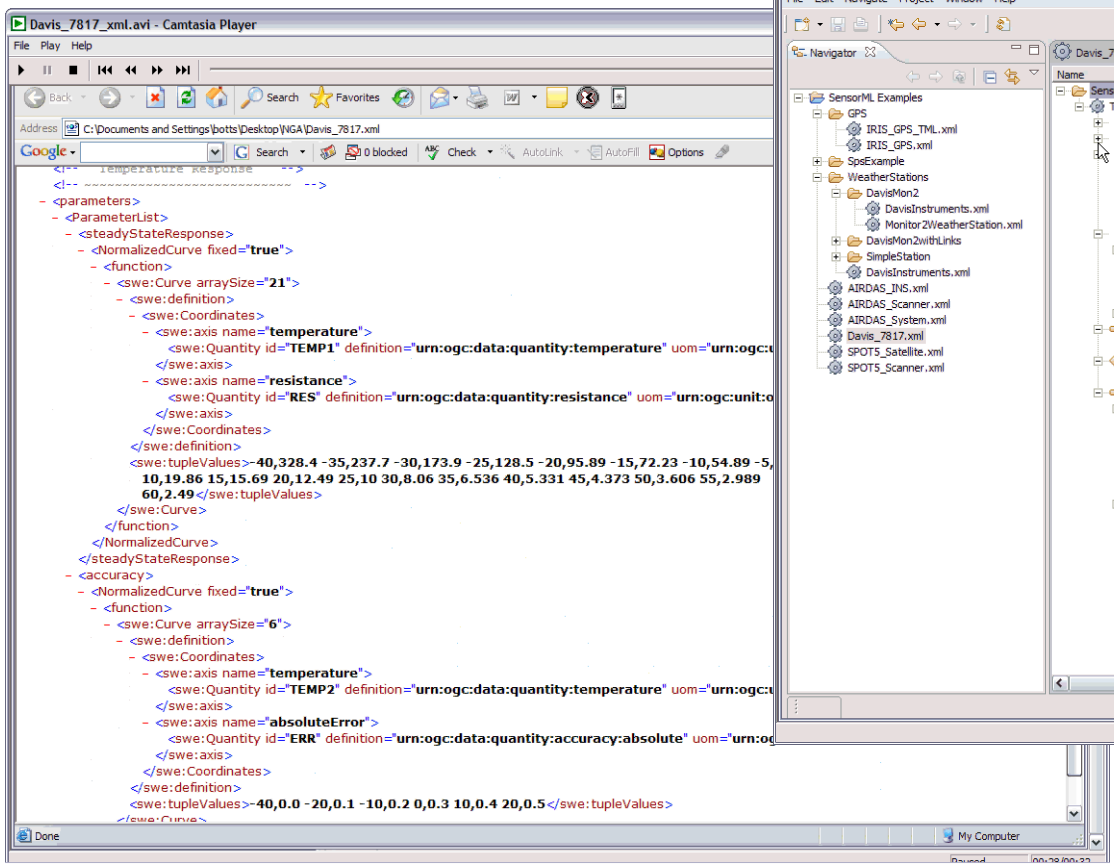
How SensorML can be used -1-

- **“Left-side” versus “Right-side” use**
  - **“Left-Side”**
    - Describes how existing data was obtained
    - Traditional approach
    - Relies on software that has knowledge of sensor system and how to apply particular parameters
    - Useful for defining lineage of observations
    - Useful for simulations of sensor system
  - **“Right-Side”**
    - Describes how to take existing data and derive on-demand higher level information
    - Our vision on how we support discovery and processing of previously unknown observations without a priori knowledge of the sensor system
    - Can utilize process models and parameters from left-side processes

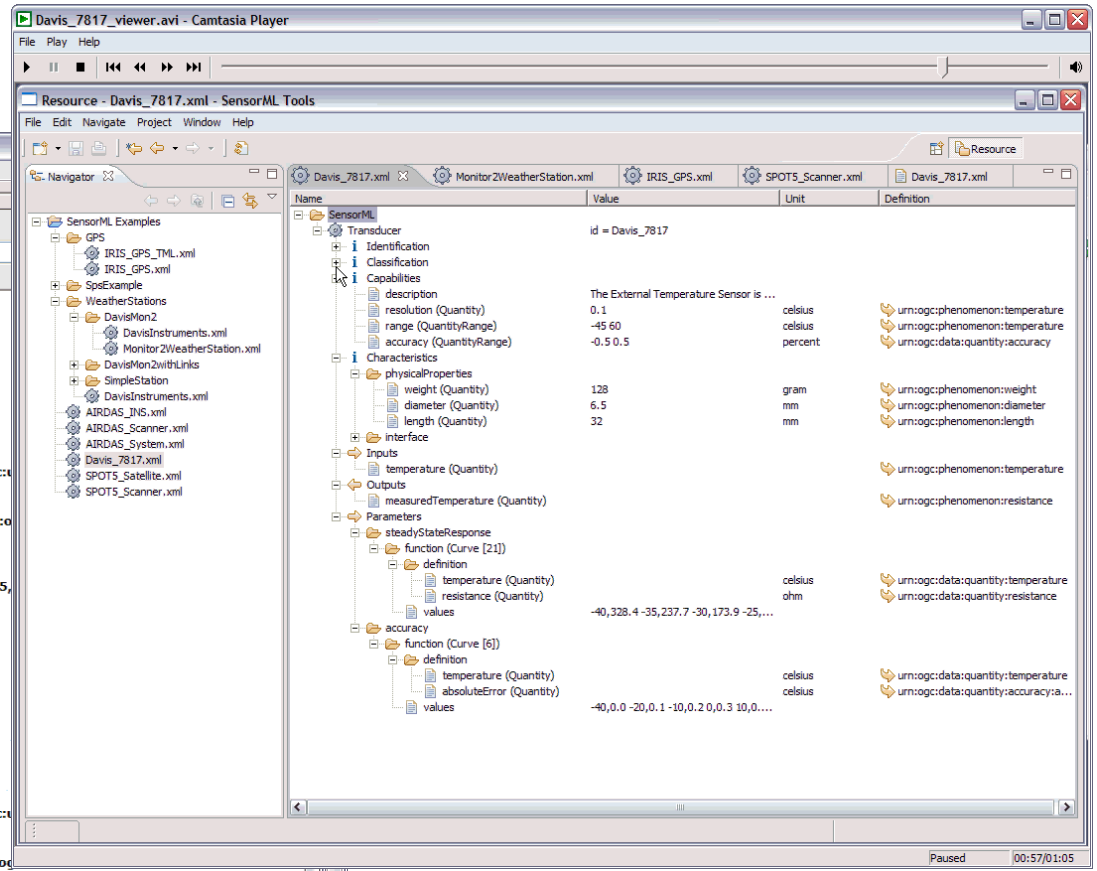
How SensorML can be used -2-

- **Time instance of process versus process factory**
  - **Time instance of process**
    - Typically might accompany a particular instance of an observation and be valid for that time
    - All parameter values for the process are defined inline
    - SPOT DIMAP and NITF approach
  - **Process Factory**
    - Process is defined but variable parameter values may not be specified
    - One treats process like factory assembly line; supply appropriate parameter values and then pump through the input values to obtain appropriate output values

Viewers for SensorML (XML, Tree View)



SensorML: XML View of Davis Thermometer

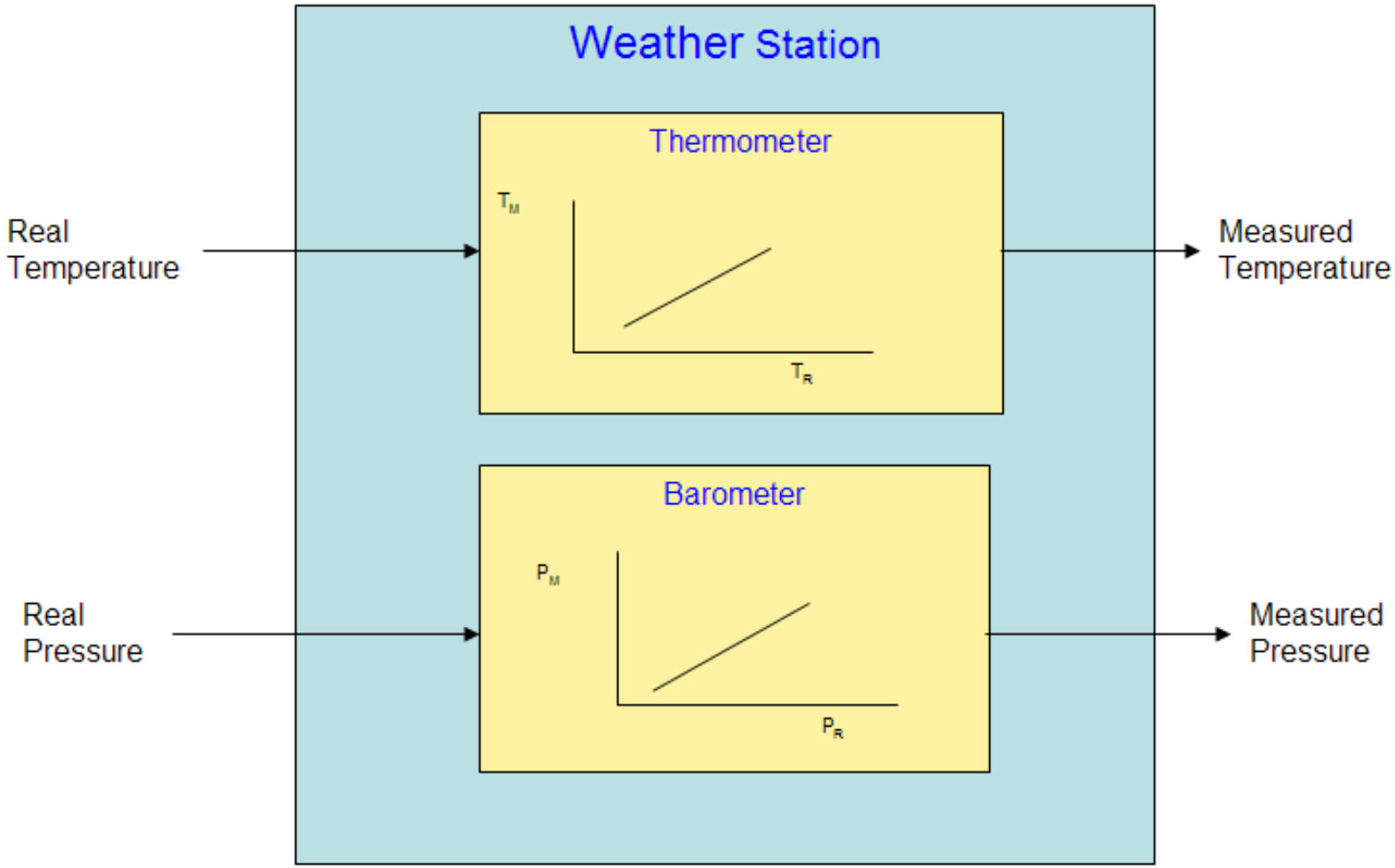


SensorML: Tree View of Davis Thermometer

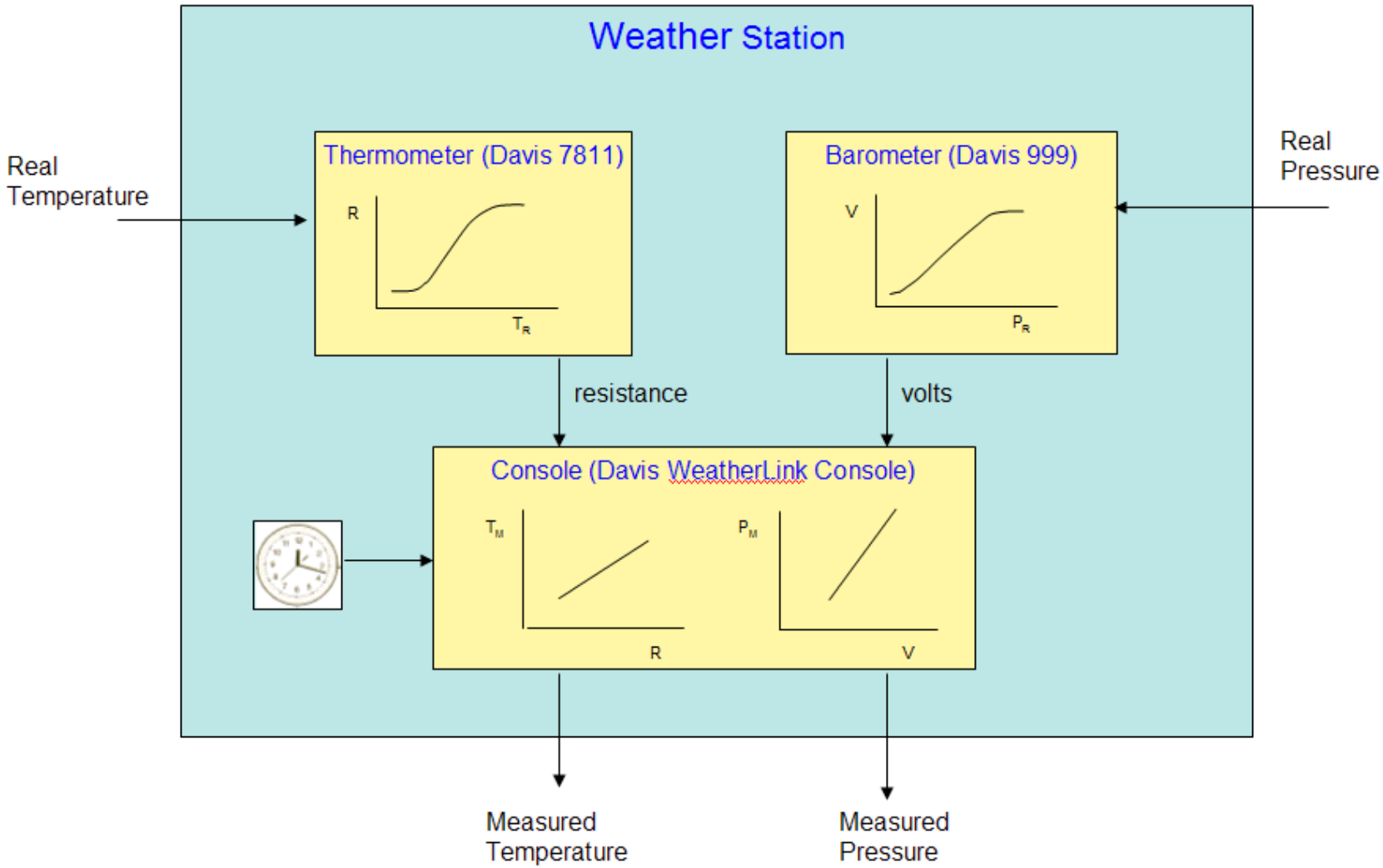
Process Models

- **Transducers** (detectors, actuators, samplers, etc.)
- **Spatial transforms (static and dynamic)**
  - **Vector, matrix, quaternion operators**
  - **“Sensor models”**
    - **scanners, frame cameras, SAR**
    - **polynomial models (e.g. RPC, RSM)**
    - **tie point model**
  - **Orbital models**
  - **Map projection and datum transformations**
- **Digital Signal Processing / image processing modules**
- **Decimators, interpolators, synchronizers, etc.**
- **Data readers, writers, and access services**
- **Human analysts**

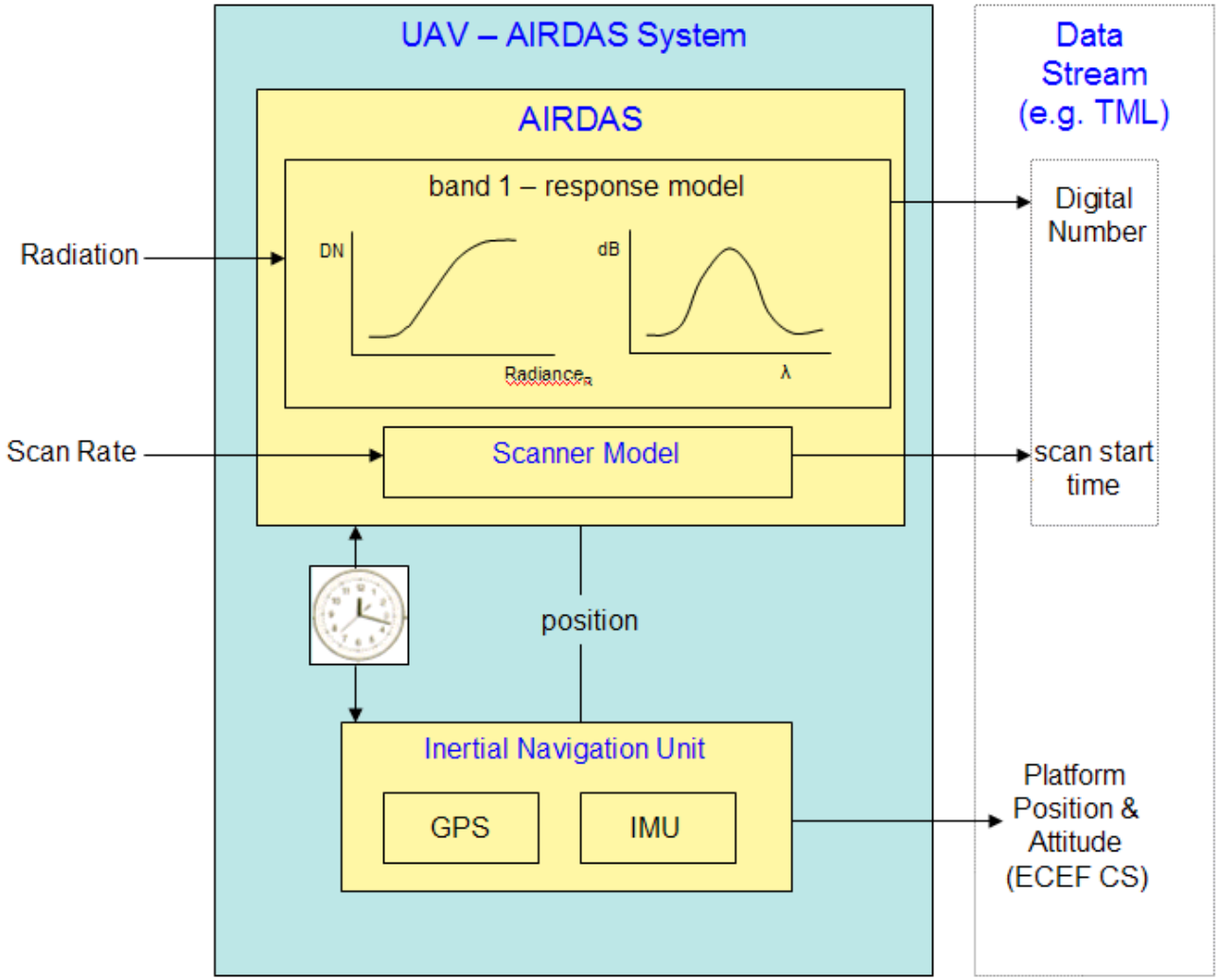
System Example – Simple Weather Station



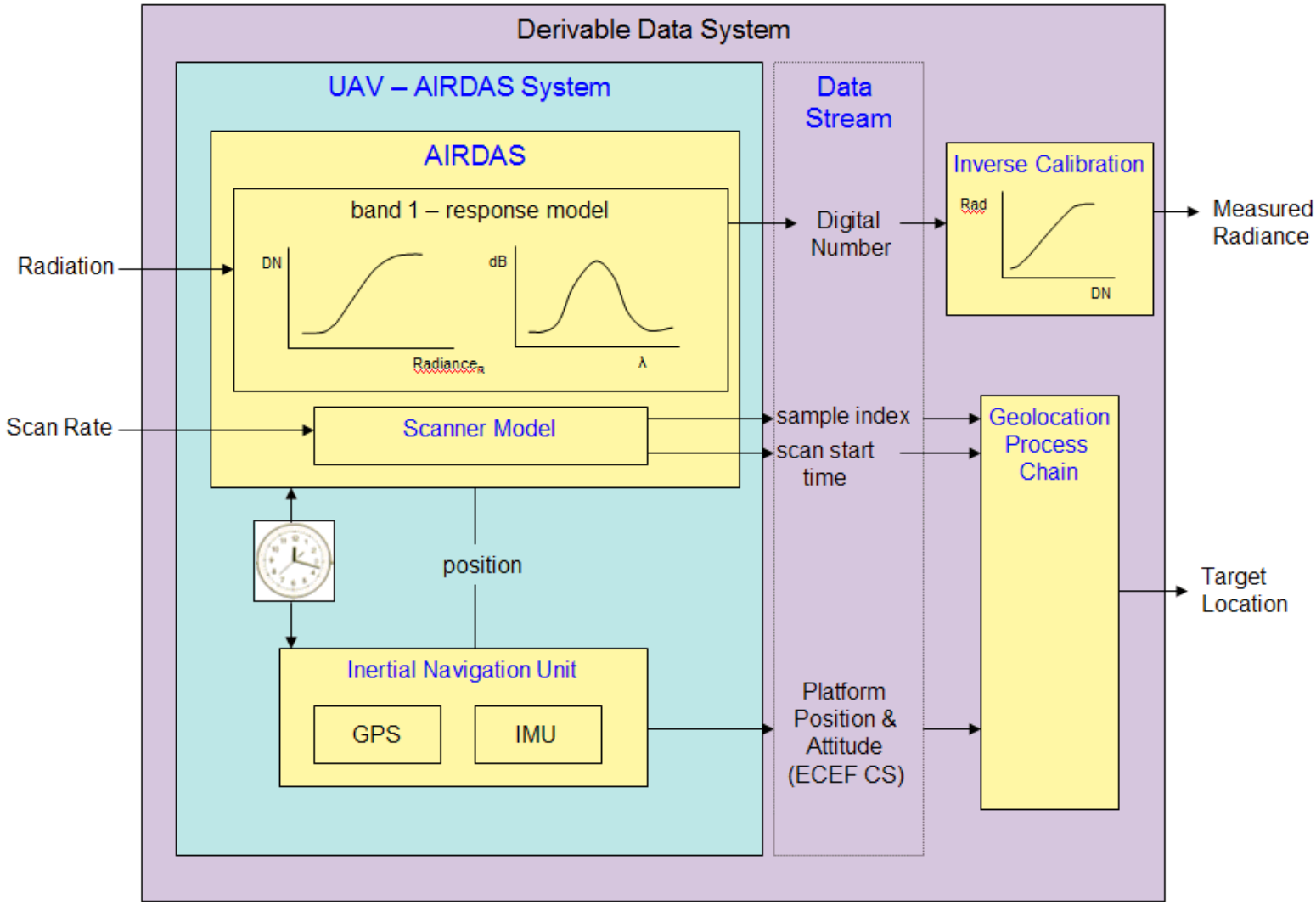
System Example – More explicit weather station



System Example – Basic UAV-AIRDAS system

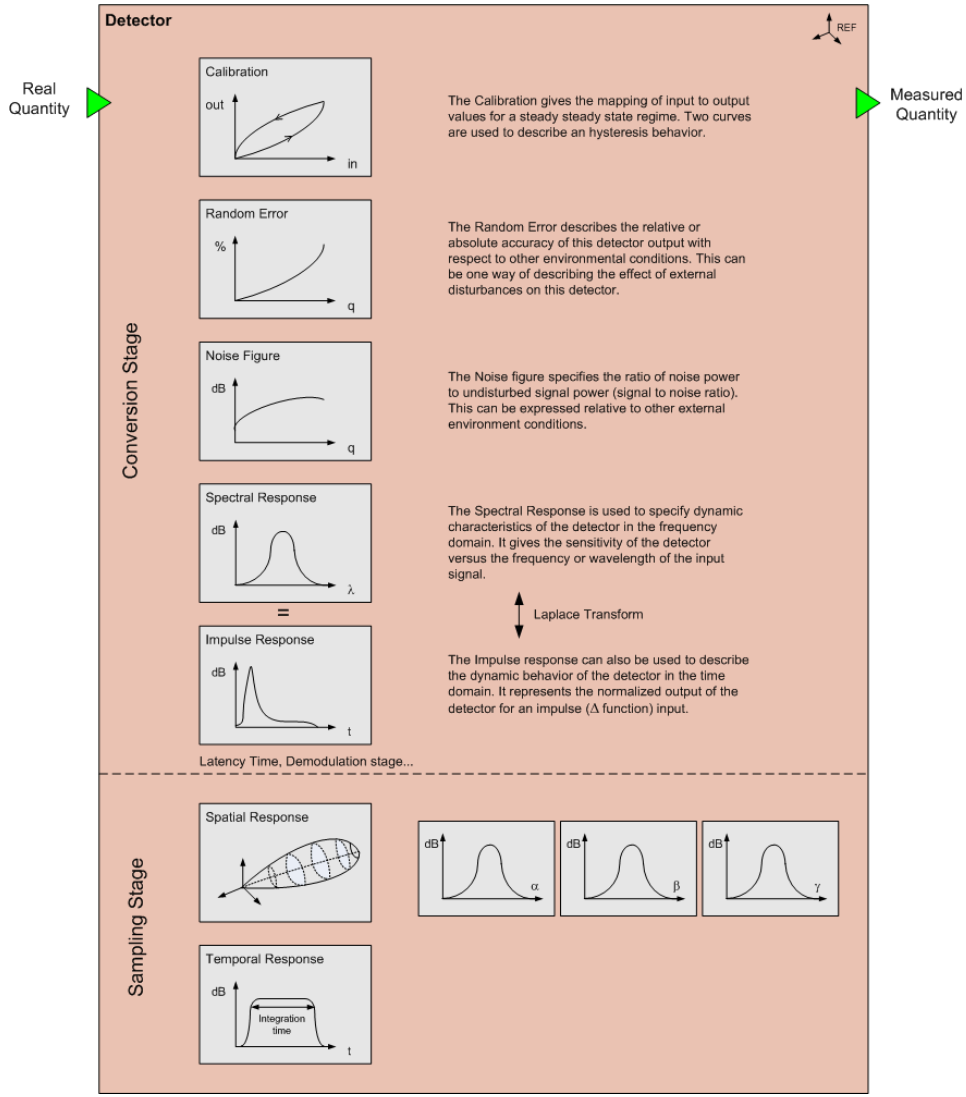


System Example – AIRDAS with derivable data



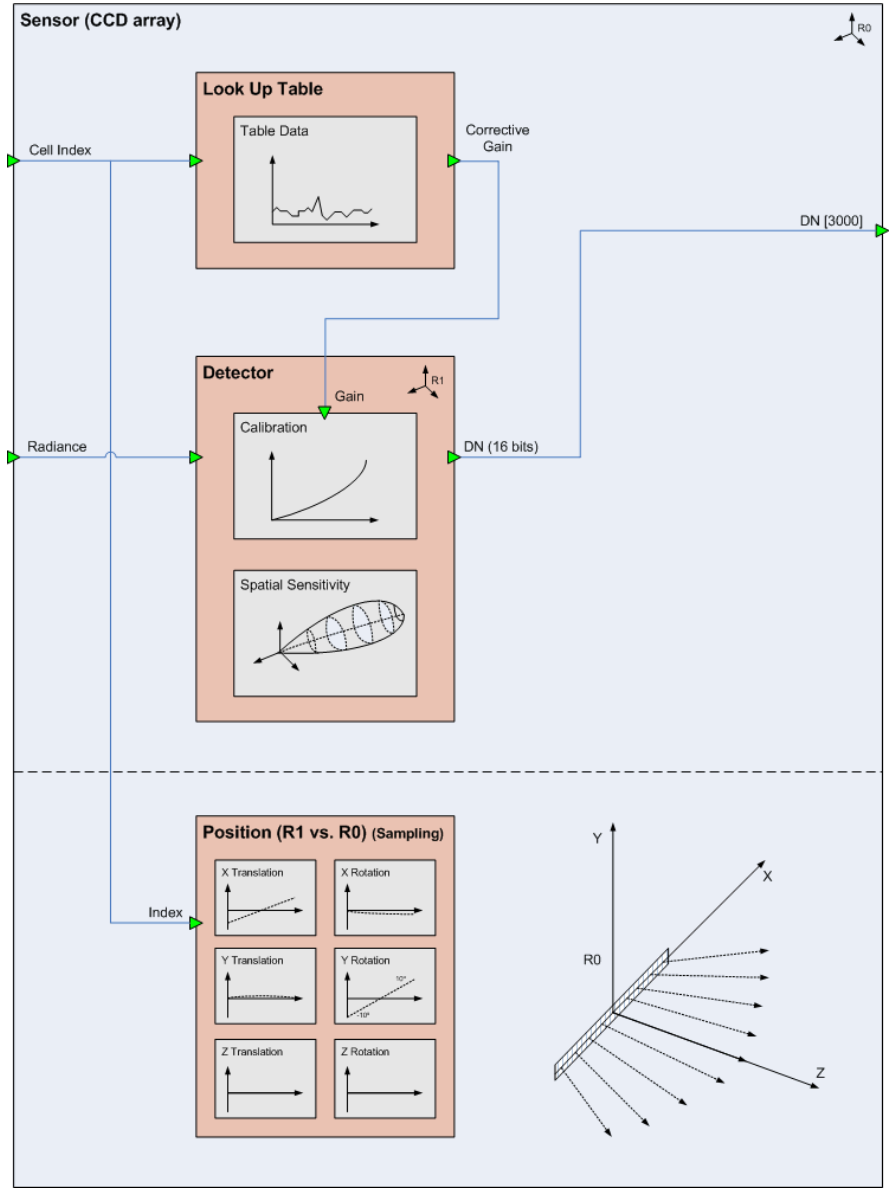
General Detector Model

- Potentially one of the most complex process models
- All parameters expressed in SensorML as curves
- Describes response of a single detector (e.g. thermometer, pixel, etc.)



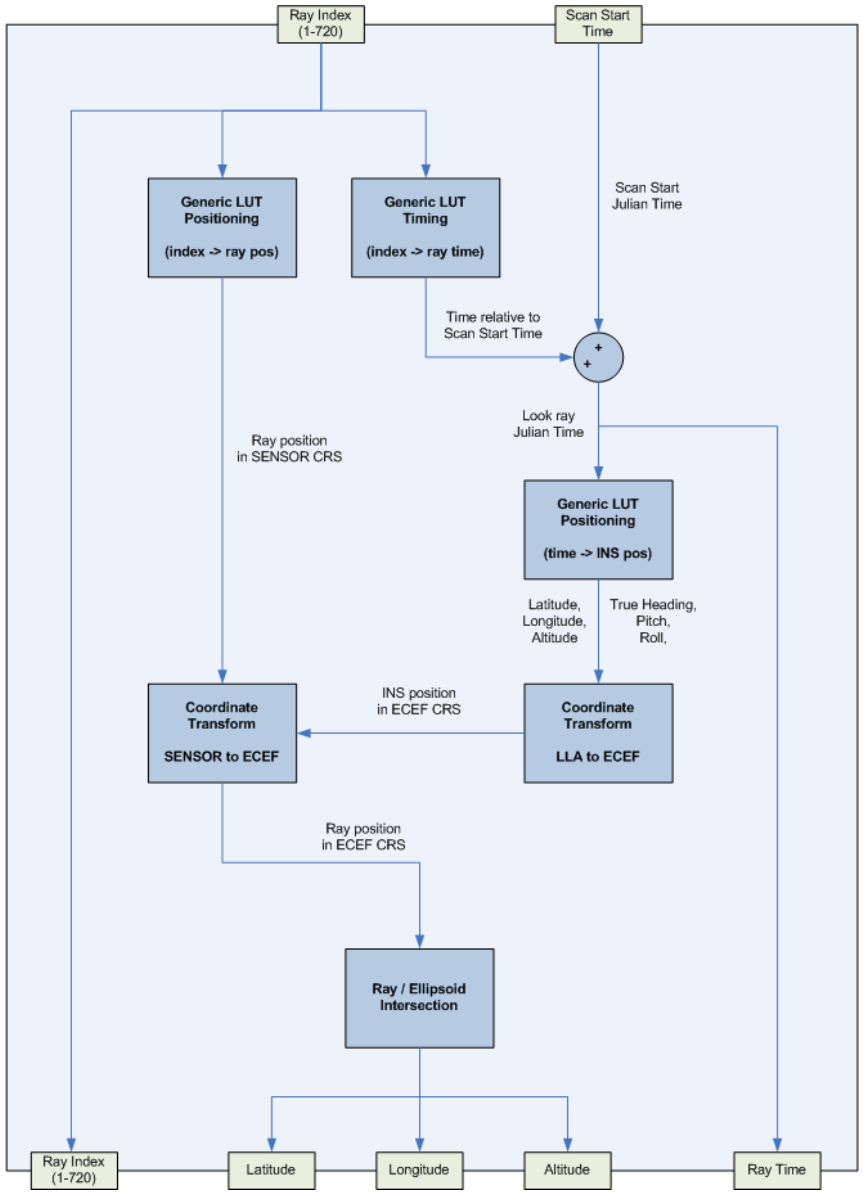
General Detector Array Model

- Detector Array modeled as a System
- Constant properties of individual detectors supported by the Detector ProcessModel
- Variable response of individual detectors supported by index dependent curves (e.g. gain vs. index)
- Spatial variability (e.g. Look ray direction) also supported using index driven curve

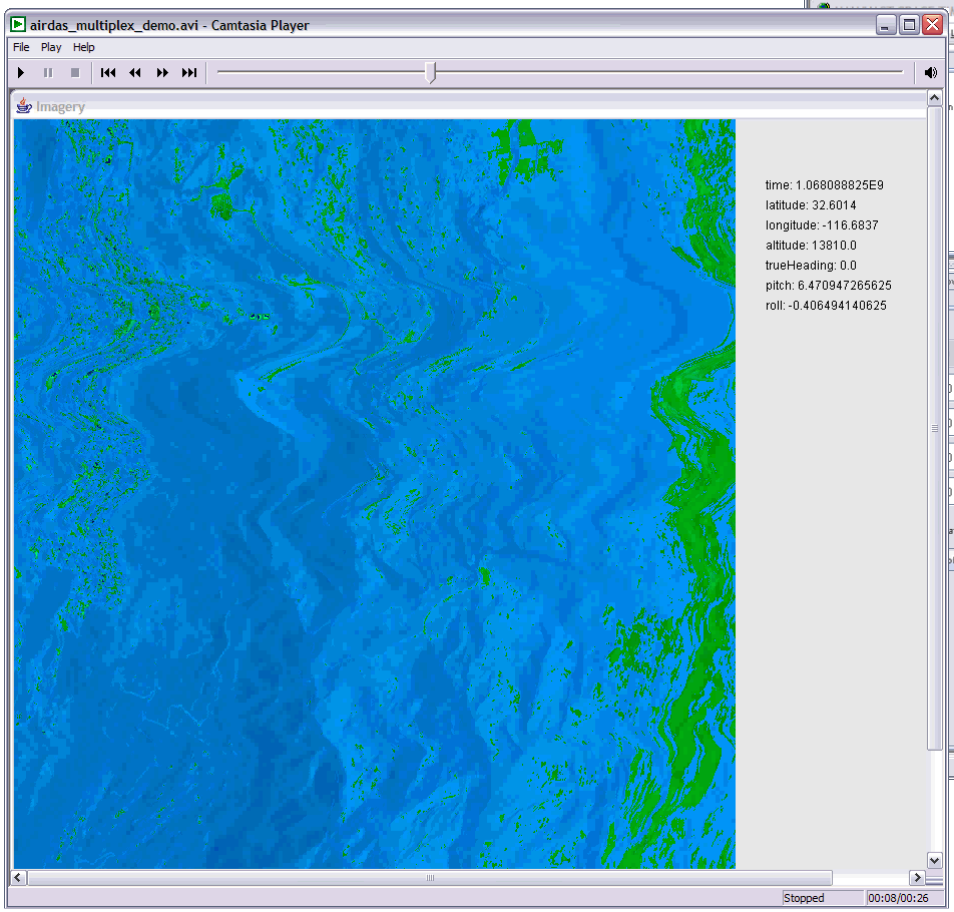


AIRDAS UAV Geolocation Process (Chain)

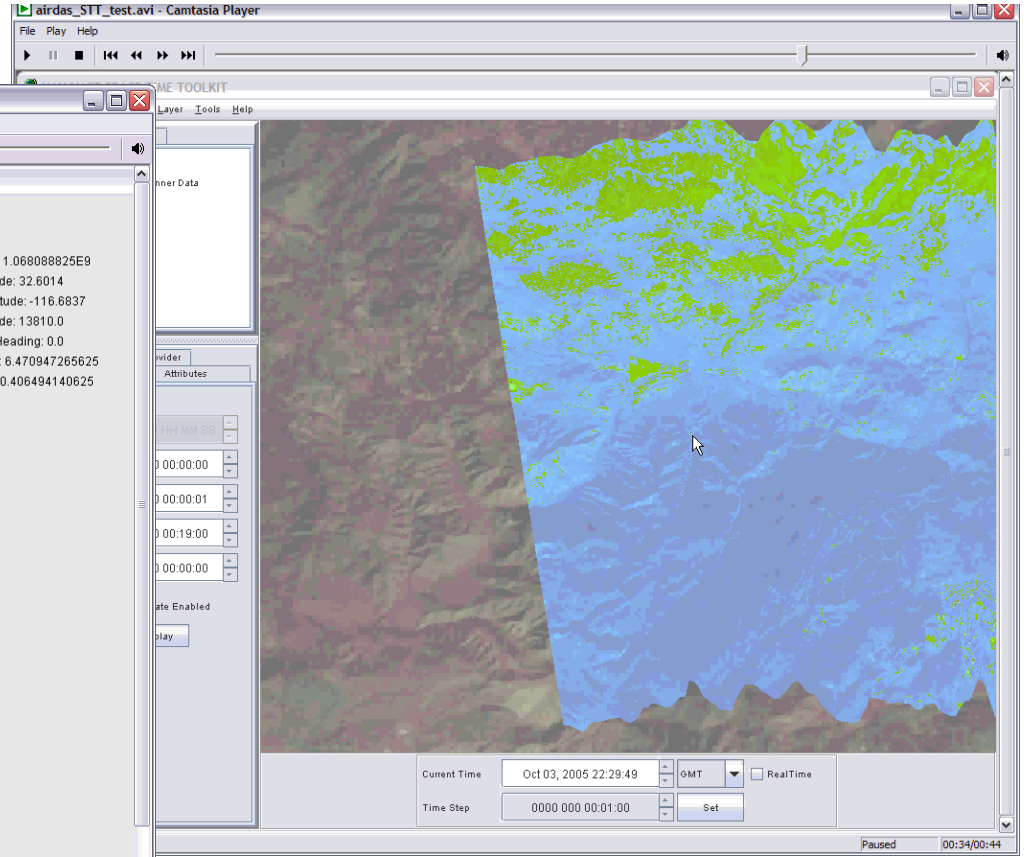
- **Inputs:**
  - Ray index
  - Scan start time
- **Outputs:**
  - Ray index
  - Latitude, longitude, altitude
  - Ray time
- **Process Models used**
  - LUTs
    - Index to ray direction
    - Index to ray time offset
    - Time to UAV state
  - Coordinate Transforms
  - Ray-Ellipsoid intersection



AIRDAS UAV Geolocation Process (Chain) Demo



AIRDAS data stream (consisting of navigation data and 4-band thermal-IR scanline data)



AIRDAS data stream geolocated using SensorML-defined process chain (software has no a priori knowledge of sensor system)

Standard Earthquake Engineering Data Standard (SEEDS)

- **THE standard for seismic sensor data**
- **Currently starting to implement in SensorML**
- **Design of SEEDS allows easy implementation in SensorML**
  - consist of chain of process module descriptions and parameters

SEED volumes usually use complex-valued functions of frequency in response functions. Usually, these functions will not be single expressions, but rather the products of several expressions. Most seismic systems can be regarded as cascades of stages — for example, a seismometer, followed by an amplifier, followed by an analog filter, followed by an analog/ digital converter, followed by a digital filter. A blockette's stage sequence number shows the order of the stages, as shown in figure 1 below:

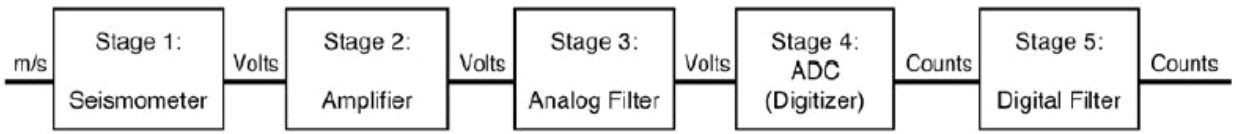


Figure 1: Example of a sequence of stages.

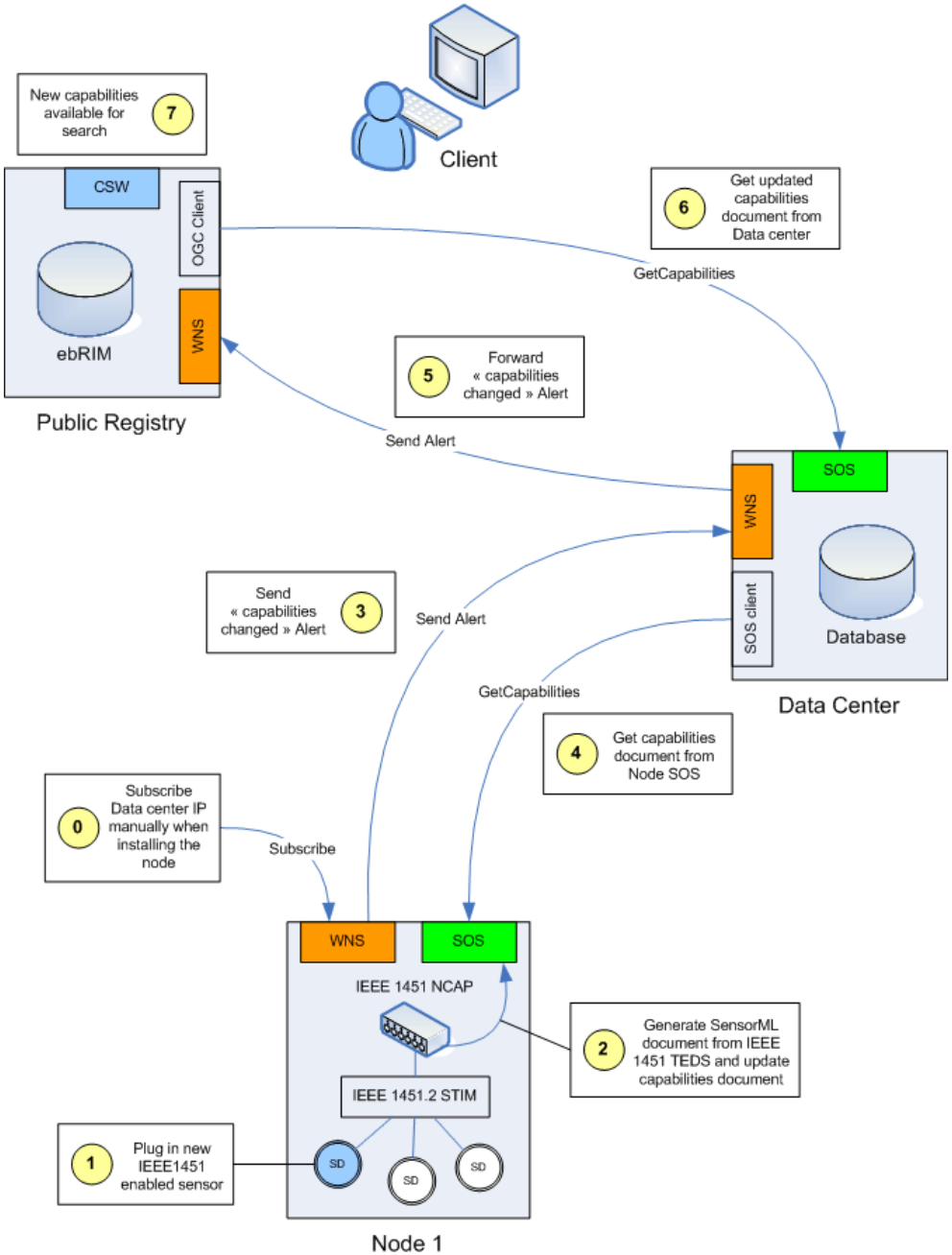
Status of SensorML Specification

- **SensorML Core**
  - Core framework models and encoding complete and stable
  - SensorML document OGC 05-086 approved as Best Practices document in Bonn (Nov 2005)
    - Approved as Public Discussion Paper (2002)
    - Approved as Recommended Paper (2004)
  - SensorML on track for OGC Technical Specification
    - Initiator (OGC Voting Member): Intergraph (?)
    - Initial committee formed
    - Beginning to obtain Letters of Support
    - Gathering and documenting current implementations
    - 90 day formal process
- **SensorML Process Models**
  - Have established a SensorML Forum for information exchange
  - Have established SourceForge project for ProcessModel submission and download
  - Have developed Open Source SensorML parser and process execution environment
  - Have developed API for Java process model software
  - Writing initial Implementation document for defining process models and processes (initial draft available in 1 week)

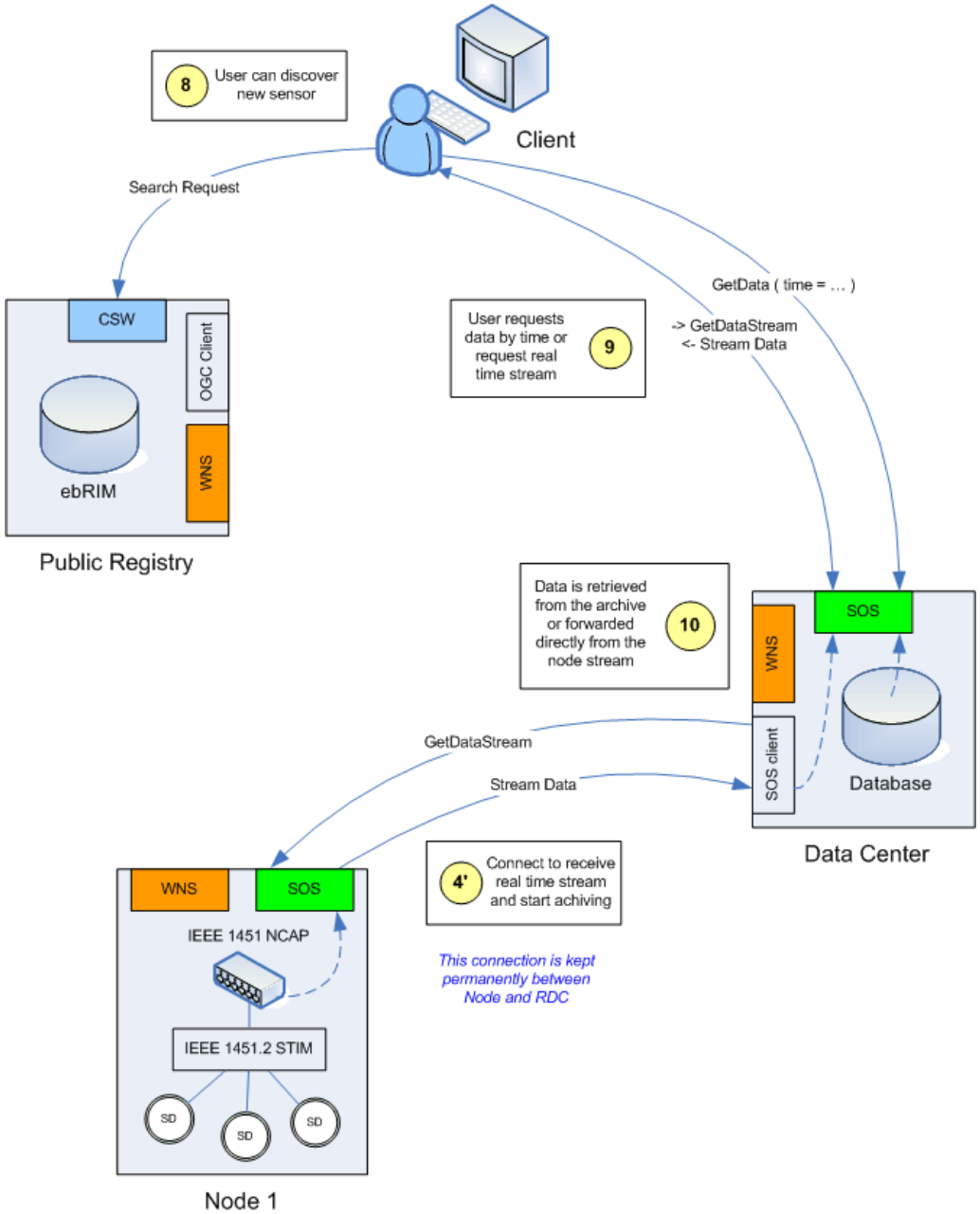
SWE and IEEE1451

IEEE P1541 within OGC SWE

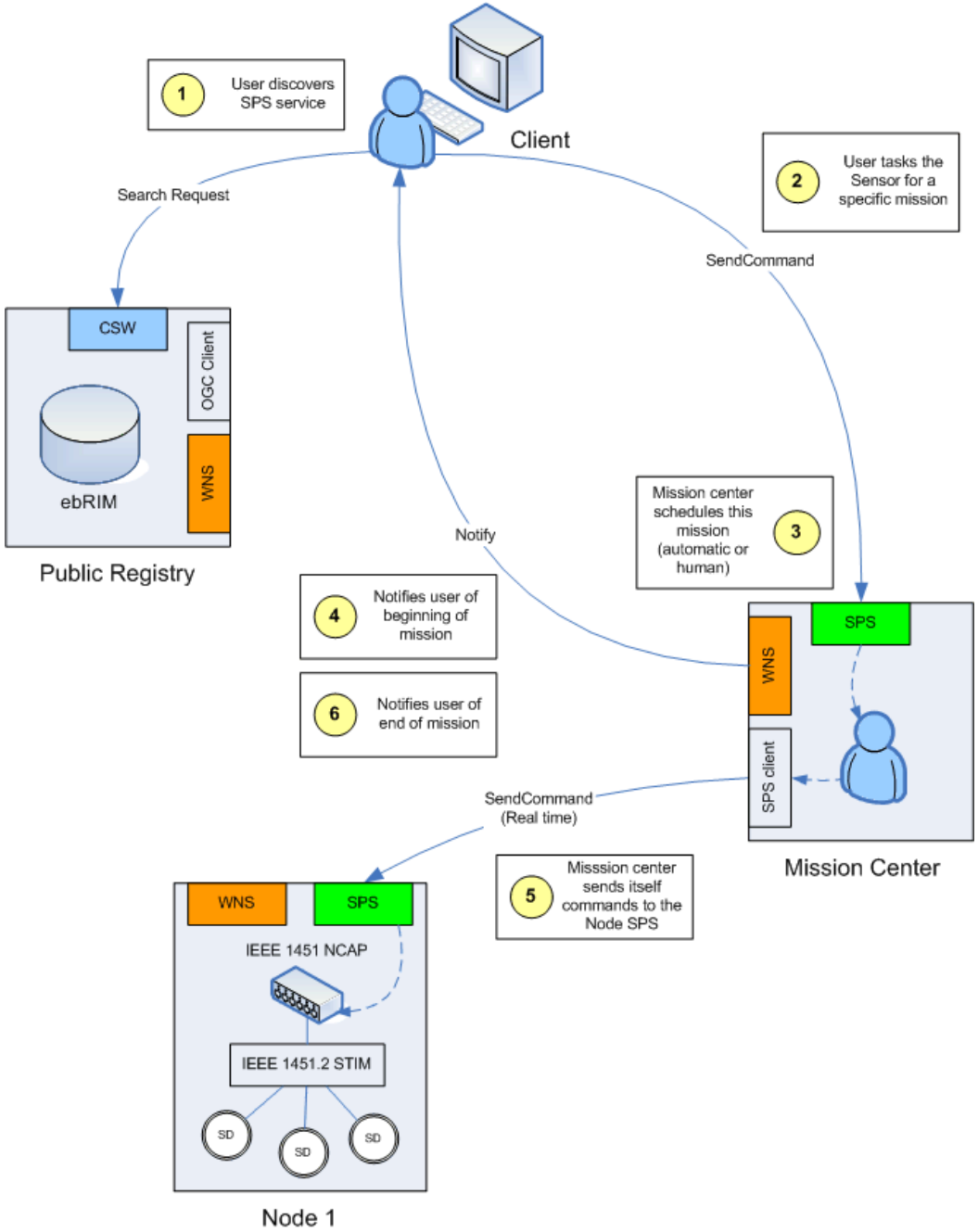
- **IEEE 1451 TEDS description to SensorML**
  - TEDS parameters map to Detector parameters
  - TEDS info on chip can be combined with SensorML profile description
  - Include interface description
- **IEEE 1451 data description to support data stream**
  - Adaptor can use IEEE 1451 data description and easily convert to Common Observation or TML description
- **SensorML and Observation become discoverable**



ORNL
Plug-n-Play



ORNL
Client Access



ORNL
Client Tasking

SWE and EDXL

EDXL and OGC SWE

- **EDXL based on envelope model**
  - Sender role, receiver role, keywords
  - Can be used within a publish/subscribe paradigm, but appears to be a last option (if all else fails)
  - Requires sender to be aware of what receivers would be interested
- **OGC Sensor Alert Service / Web Notification Service**
  - Based fully on a publish/subscribe
  - Any user can subscribe to any alerts based on any parameters
    - keywords, sender, alert type, geographic region
  - Various types of notification are available through WNS
    - XMPP, SMS, email, URI

SWE and CAP

CAP and OGC SWE

- **CAP could be supported as an alert encoding type in SAS**
- **Converter between CAP and OGC Observation/TML data**
  - It would be easily possible to create a CAP profile in OGC Observation (but that may be irrelevant)
  - Very easy to convert between CAP and Observation
- **CAP could include link to SensorML process instance**
  - Provide sensor and process lineage of the alert/observation

Relevant Links

OpenGeospatial Consortium –

http://www.opengeospatial.org

SensorML –

http://vast.uah.edu/SensorML

Space Time Toolkit –

http://vast.uah.edu/SpaceTimeToolkit