Web-based Altimeter Service
Final Review

ACCESS 07

Phil Callahan, PI

Brian Wilson, Co-I

Rob Raskin, Co-I

Zhangfan Xing, Lead Developer

March 3, 2011

Final Review Outline

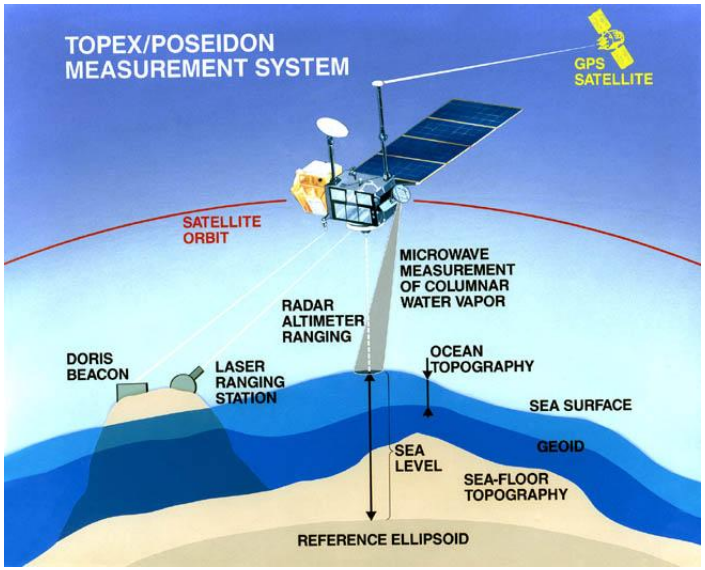
- Review Quad Chart
- Objectives and Accomplishments
- Background
  - Altimetry
  - SciFlo
- Accomplishments – Technical Discussion
  - Altimetry Data
  - SciFlo Implementation
  - Data Display – “Webification”
- Service Overview and Demo
- PODAAC Transition
- Followup Issues

Web-based Altimeter Service

Phil Callahan, JPL

Objectives

- Develop a web-based tool for subsetting and updating altimeter data
  - Altimeter data consist of several specialized "components" that are updated by different groups at irregular intervals.
  - Specialized data exist for localized areas.
- Work with providers to get tool access to data and models as they are updated.
  - Locate specialized models for coastal areas.



Altimeter Sea Surface Height Measurement Components for Updating:

- Orbit
- Tides
- Radiometer
- Atmospheric – range, inverse barometer
- Geoid
- Range processing, corrections

Approach

- Build on SciFlo system for user interface, data access, algorithm control
- Modularize Geophysical Data Record (GDR) update algorithms to provide processing functionality
- Integrate SciFlo and GDR algorithms
- Work with scientists, data centers to get access to models and data sets

Co-Is

- **Rob Raskin (JPL), Brian Wilson (JPL)**

Accomplishments

- ✓ Web interface has email login and status checking capabilities
- ✓ Data can be selected by cycle, pass ranges or set of passes that go through a lat, lon box
- ✓ Produced capability to update orbits using standard POD format
- ✓ Produced capability to update tides in Goddard format
- ✓ Produced capability to update any point-by-point correction
- ✓ Produced TOPEX data in netCDF format compatible with ongoing Jason missions
- ✓ Generate updated sea surface height (SSH) and anomalies (SSHA) data
- ✓ Introduced "webification" – inline display of attributes and plotting of netCDF data
- ✓ Will link to new PODAAC pages

TRL_in = 5-6; TRL_out = 7

Web-Based Altimetry Service Objectives and Accomplishments (1 of 2)

- Develop a web-based tool to allow subsetting and updating of components of altimeter data
  - ✓ Web interface has email login and status checking capabilities
  - ✓ Data can be selected by cycle, pass ranges or set of passes that go through a lat, lon box
- Update TOPEX data to same components (e.g., orbits, tides, mean sea surface) and introduce known corrections as Jason
  - ✓ Produced capability to update orbits using standard GSFC POD format
  - ✓ Produced capability to update tides in Goddard (GOT) format
  - ✓ Produced capability to update any point-by-point correction
  - ✓ Produced TOPEX data in netCDF format compatible with ongoing Jason missions
  - ○ Mean sea surface interpolation in progress
- Generate updated sea surface height (SSH) and anomalies (SSHA) data
  - ✓ When update is done, netCDF files of SSH, SSHA are generated

Web-Based Altimetry Service Objectives and Accomplishments (2 of 2)

- Transition service to PODAAC
  - Service will be linked from new PODAAC pages
  - Migration continuing, still resolving some issues
- Add components as they are updated by experts in the altimeter community
  - ✓ Standard GSFC POD orbits and tide updates can be added by simple ftp of new data files
  - Additional data sets (tides, mean sea surface) are known, but some code modifications are required to use them

✓ Introduced "webification" to view netCDF data

✓ TRL Advancement

- ✓ SciFlo: TRL 6 ➔ 7 based on use in PODACC environment
- ✓ "Webification" : TRL 5 ➔ 7 based on use in PODACC environment

Altimetry Background

- Series of successful science missions from TOPEX/POSEIDON through Jason-1, 2; Jason-3 being planned
- Large international community uses data for oceanographic studies (original intent) and increasingly other applications, particularly coastal and inland water
- Extreme accuracy (~2 – 3 cm absolute, << 1 mm/yr change) is key to success, particularly for climate change applications
- Measurement concept is simple but requires numerous corrections (atmosphere, sea state), “components” (orbits, tides) to reach required accuracy
- Accuracy is ensured, improved by
  - New orbits
  - Calibration, correction of errors, drifts in components
  - New components for new applications
- New uses may also require new, specific (re)processing

SciFlo Workflow Engine

- SciFlo Distributed Dataflow System
  - Large-scale, loosely-coupled, distributed computing using Web Services (Simple Object Access Protocol - SOAP) and Cloud computing (on-demand virtualization)
    - Specify a processing stream as an XML document
  - Leverage Web Services standards, open source
  - Dataflow engine for automated execution and load balancing – parallel, asynchronous processes

- Automate large-scale, multi-instrument science processing by *authoring* a dataflow document that specifies a *tree* of *executable* operators.
  - iEarth Visual Authoring Tool (VizFlow)
  - Distributed Dataflow Execution Engine, implemented in python
  - Move data "granules" to the operators using FTP, HTTP, or OpenDAP URLs.
  - Move operators (executables) to the data.
  - Built-in reusable operators provided for many tasks such as subsetting, co-registration, regridding, data fusion, etc.
  - Custom operators easily plugged in by scientists.

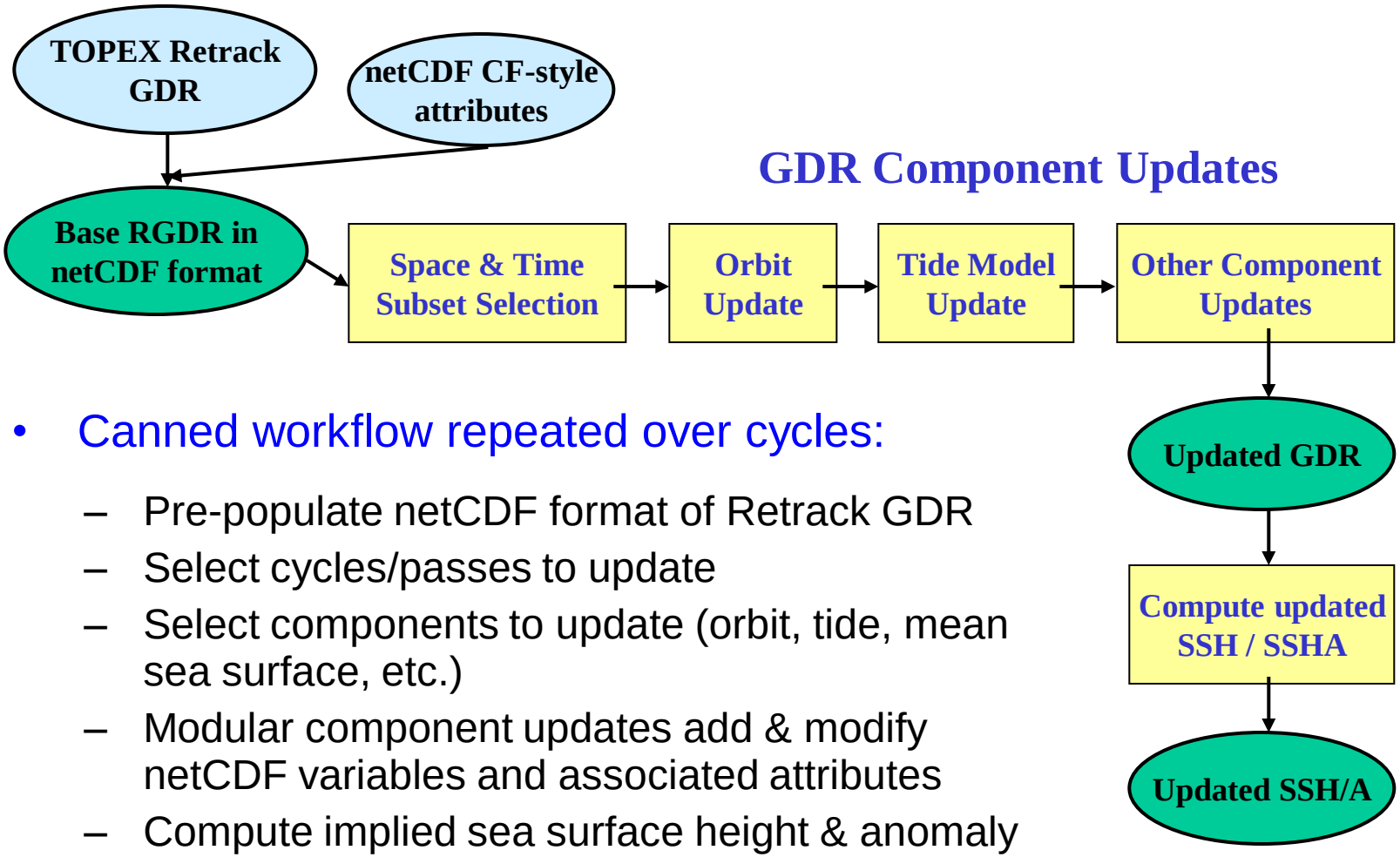
Service/Operator Choreography

- Each SciFlo processing step is one of:
  - Template for XML (or string) generation
  - REST (http GET) call: e.g. WMS/WCS, DAP
  - SOAP service call: "have WSDL, will call"
  - XPath 2.0 transformation for XML mediation
  - XQuery 1.0 query/transformation
  - Command-line script or executable
  - Python method call
  - Scientist's custom IDL or MATLAB script
  - Other (What do you need?)

Technical Accomplishments

- Modernized TOPEX GDR, on equal footing with JASON -1/2
  - netCDF format with self-describing attributes, as similar to JASON-1/2 files as possible
  - New base GDR is the Retrack GDR
- Modularized software components to update GDR
  - Updateable TOPEX components: orbit, tide model, along-track correction, etc.
  - Easily extendable to JASON-1/2 netCDF
- Expose GDR workflows as web services
  - Space/time subsetting
  - Workflows to update multiple components
  - Web interface to submit & monitor GDR update jobs
- Webification of science data containers (joint AIST & ACCESS work)
  - Drill into any netCDF, HDF4/5, GRIB, or FITS file anywhere in the world
  - Open standard, W10n-sci
  - Open source python implementation (downloadable pomegranate library)

Workflow for GDR Updates



- Canned workflow repeated over cycles:
  - Pre-populate netCDF format of Retrack GDR
  - Select cycles/passes to update
  - Select components to update (orbit, tide, mean sea surface, etc.)
  - Modular component updates add & modify netCDF variables and associated attributes
  - Compute implied sea surface height & anomaly
  - User downloads updated products (in green)

Binary TOPEX GDR to netCDF

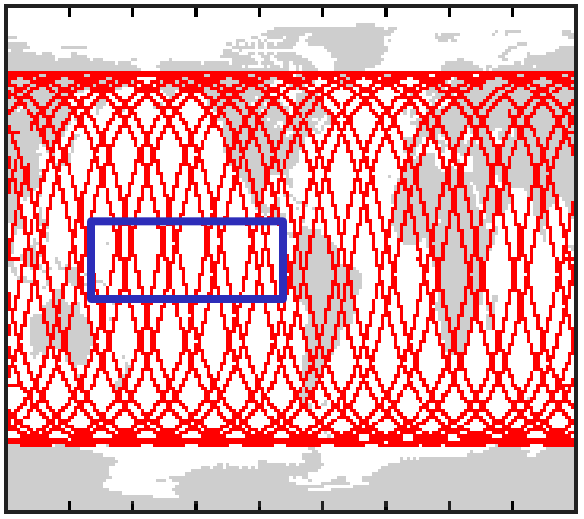
- Native TOPEX GDR is record-oriented binary format
  - Header contains global metadata (saved as netCDF global attributes)
  - Followed by packed integer records
  - Converted to netCDF vectors/arrays with CF-style attributes
  - Added CF-compliant time, lat, lon dimension variables
- Design decisions:
  - Retain scaled integer values since TOPEX users are used to them
  - Variable attributes: scale, offset, unit, long_name, original_name
  - Modeled variable names & attributes after JASON netCDF products
  - If variable renamed, original name retained in attribute as documentation
- Developed general binary-conversion software
  - Started with C program, leveraging PO.DAAC activity
  - Abandoned: turned out to be inflexible, hard to change quickly
  - Developed general python program that can read any binary-record file, given a format specification, and write netCDF3/4
  - Converter adds netCDF attributes expressed in NcML (simple XML)

Example CF-style Attributes

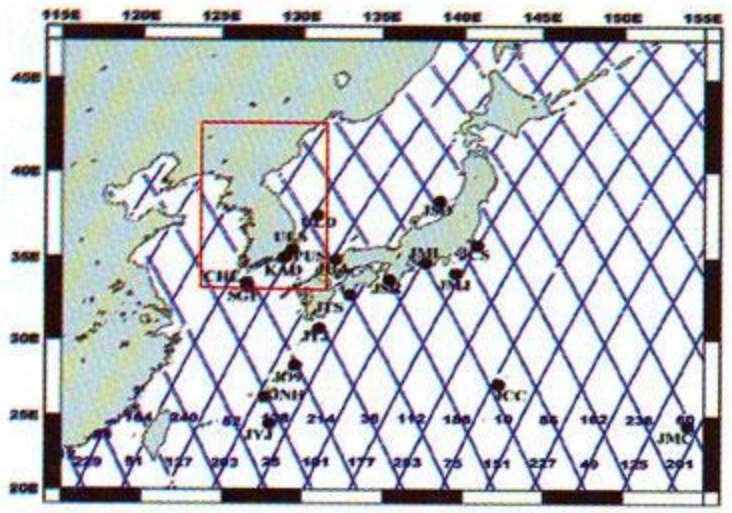
```
<?xml version="1.0" encoding="UTF-8"?>
<netcdf xmlns="http://www.unidata.ucar.edu/namespaces/netcdf/ncml-2.2">
  <dimension name="time" length="3303" />
  <dimension name="meas_ind_ku" length="10" />
  <dimension name="meas_ind_c" length="5" />
  <attribute name="Conventions" value="CF-1.1" />
  <attribute name="institution" value="JPL" />
  <attribute name="source" value="radar altimeter" />
  <attribute ...>
  <variable name="time" shape="time" type="double">
    <attribute name="long_name" value="time (sec. since 2000-01-01)" />
    <attribute name="standard_name" value="time" />
    <attribute name="calendar" value="gregorian" />
    <attribute name="tai_utc_difference" type="double" value="-33." />
    <attribute name="leap_second" value="0000-00-00 00:00:00" />
    <attribute name="units" value="seconds since 2000-01-01 00:00:00.0" />
    <attribute name="comment" value="[tai_utc_difference] is the difference between TAI
    and UTC reference time (seconds) for the first measurement of the data set. " />
  </variable>
  <variable name="sat_alt_1" shape="time" type="int">
    <attribute name="_FillValue" type="int" value="2147483647" />
    <attribute name="long_name" value="1 per frame altitude of satellite" />
    <attribute name="standard_name" value="height_above_reference_ellipsoid" />
    <attribute name="units" value="m" />
    <attribute name="scale_factor" type="double" value="0.001" />
    <attribute name="coordinates" value="lon lat" />
    <attribute name="comment" value="Altitude of satellite above the reference ellipsoid;
    copied from MGDR POE_1, hp_sat_alt" />
  </variable>
</netcdf>
```

Space/Time Subsetting

- TOPEX and JASON-1/2 Repeating Orbits
  - 254 passes per cycle, cycles repeat
  - Use pre-computed lookup tables
  - Which cycle/passes intersect 1-by-1 degree latitude & longitude bins



- Space/time Query Service
  - Given lat/lon table & time range, return list of cycles/passes
  - Callable from python, also exposed as web service
  - User can select subset by cycle/pass numbers or space/time box
  - Update GDR globally, by ocean, or in small region



Binding Algorithms into Python

- Reuse existing algorithms for updating GDR
  - Modularize codes into callable Fortran or C subroutines
  - Algorithms: Functions that read & write numeric arrays
  - No globals: All file names and config. parameters are passed in arguments
  - Error handling: Error codes returned to caller, no Fortran aborts
- Bind algorithms into python
  - Use f2py or SWIG to generate glue code, or write by hand
  - F2py tool: Fortran arrays become python numpy arrays
  - Compile into shared object library
  - Library dynamically linked into python using 'import'
  - Example: Precise Orbit Estimation (Fortran algorithms):
    - from poe import poe_interp_pt, poe_interp_highrate_pt
- Service Workflows
  - SciFlo workflow document invokes python routines as processing steps
  - Algorithms can also be published as Web Services
  - Regional Tide Model: could call remote service exposed at university

Job Control

- Job control important
  - Have long-running jobs, updating many TOPEX cycles
  - Large jobs broken into sub-jobs for each cycle
  - Job progress visible a cycle at a time
  - Job execution is throttled (only a few large jobs at a time)
- Jobs & user database
  - Must enter an email address to execute a job (but only that)
  - Low barrier: user does not have to pre-register to use system
  - Simple jobs database saves state of job execution
- Job interface is REST URL's sent to user in an email
  - User web page lists all his jobs as links (most recent first)
    - http://altiserv.jpl.nasa.gov/aws/user/<hashed-user-email-address>
  - Each job page shows status of job execution
    - http://altiserv.jpl.nasa.gov/aws/jpb/<hashed-unique-job-id>
  - Completed jobs page points to downloadable results
  - Results page is webification-enabled, with default visualizations

"Webification" Technology – Pomegranate

- Pomegranate (http://pomegranate.jpl.nasa.gov) is a python application, that implements the nascent Webification Science (w10n-sci or p9e) API. Developed by Zhangfan Xing under several NASA tasks
  - It can be used as
    - Standalone python module/library
    - Command line tool
    - Callable RESTful web service
  - Formats supported: NetCDF, HDF4, HDF5, GRIB and FITS
  - Client examples in command line, python, php, java, AJAX,
  - Open specification drafted by JPL, the HDF group, UCAR, others
  - Current version available at http://w10n-sci.jpl.nasa.gov
  - Wiki at http://wiki.esipfed.org/index.php/W10n-sci
  - Installer available for Apache web server

- The idea of webification: the inner contents of a data store, such as attributes and data arrays should be directly addressable by well-defined, meaningful urls

"Webification" Examples

Inner components of a data store, such as attributes and data arrays, should be directly ***addressable*** and ***accessible*** by **well-defined** and **meaningful** URLs.

Examples:

- http://w10n.jpl.nasa.gov/test/data/nc/coads_climatology.nc
  is a NetCDF file on host w10n.jpl.nasa.gov
- http://w10n.jpl.nasa.gov/test/data/nc/coads_climatology.nc/
  represents meta information of file coads_climatology.nc
- http://w10n.jpl.nasa.gov/test/data/nc/coads_climatology.nc/SST/
  represents meta information of variable SST inside the file
- http://w10n.jpl.nasa.gov/test/data/nc/coads_climatology.nc/SST[]?output=netcdf
  returns entire data array of SST as NetCDF
- http://w10n.jpl.nasa.gov/test/data/nc/coads_climatology.nc/SST[0:2,45:55,85:95]?output=json
  returns data array slice SST[0:2,45:55,85:95] in JSON format

Summary of Webification W10N-SCI API

- An arbitrary data store is exposed as a simple tree.
- The tree has only 2 types of entities: node and leaf.
- A node may contain sub-nodes and leaves.
- A node or leaf may have attributes.
- Meta information is defined for both node and leaf.
- Data information is currently defined for leaf only.
- Meta or data information is addressable by extended URLs.
- Meta or data information is readable/writable via HTTP.

Extended URL Syntax:

http://host:port/path/identifier?query_string

In which,

- identifier indicates what to read/write
- query_string dictates how to do it

Extended URLs
(1 of 3)

Identifier Examples

Identifier	What is identified
/	Meta information about the store.
/node/	Meta info of a particular node in the store.
/node/leaf/	Meta info of a leaf under a node.
/node/leaf[]	Entire byte array for that leaf.
/node/leaf[indexer]	A subset of that leaf indicated by an indexer.
more can be defined.	

Indexer Examples

Indexer	Type	Example
[start:end:step,start:end:step,…]	range	[0:100,100:200]
[n0,n1,n2,…]	list	[2,4,6,8,10]
[(x,y)width*height]	tile	[(50,50)300*200]
More can be introduced.		

Extended URLs
(2 of 3)

Parameters defined for query_string

Parameter	Value	Type	Meaning
output	json/html/raw/nc/…	String	Format of output
reCache	false/true	Boolean	If cached output is used
flatten	false/true	Boolean	If array is flattened
traverse	false/true	Boolean	If traverse
More can be introduced.			

query_string is a string of parameter-value pairs concatenated by '&'.

Reading and Writing via HTTP Request

Whether meta or data info, the HTTP request is unambiguous by the URL used.

GET Request – Read API

http://host:port/path/identifier?query_string

Message body is absent.

PUT Request – Write API

http://host:port/path/identifier?query_string

Message body contains data.

HTTP Response

Data Response:

- Defined only for a leaf in current w10n-sci API.
- Output formats can be
  - raw (default, binary or text)
  - json (text)
  - big-endian, be (binary)
  - little-endian, le (binary)
  - netcdf, nc (binary)

  More possible, depending on service implementation.

Webification Type:

Designation of ways a data store can be webified: `major.minor`

- `major` can be `nc, hdf4, hdf5, grib,` etc.
- `minor` can be `basic, eos.swath, eos.grid,` etc.

For example, a EOS granule file may be webified as either `hdf4.basic` or `hdf4.eos.swath`

Client Support

- Developing Clients is easy
  - Request: HTTP GET call
  - Response: parse JSON, or write binary data to file
  - No complicated data model
  - Can always request netCDF bundle as response
- Current Client support
  - Command line (wget, curl)
  - Python, php, java
  - Matlab, IDL
  - Quickview, an AJAX client for web browsers
  - <Any language that can do a HTTP-GET call & parse the returned JSON string>

Webification Infusion

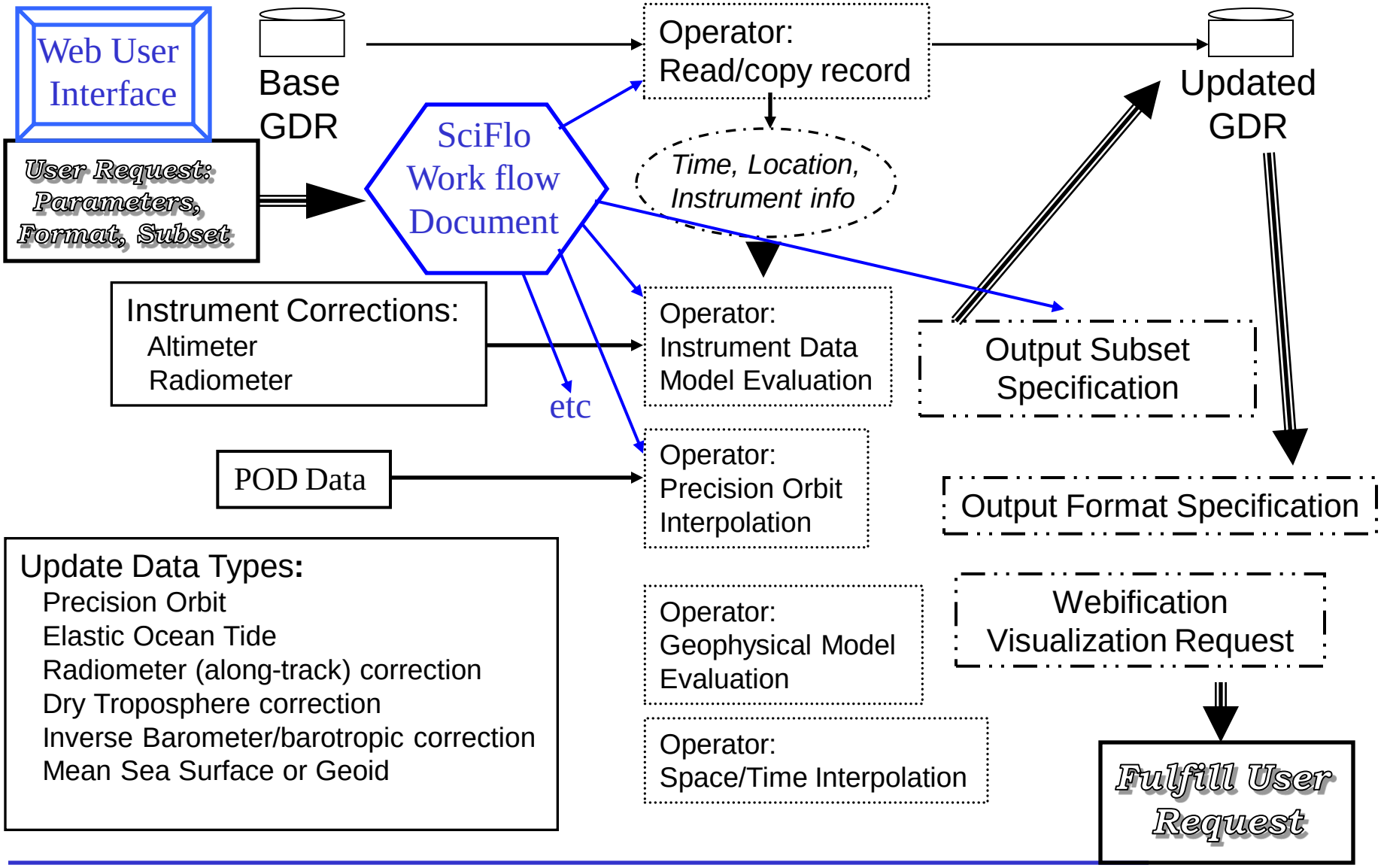
- Webification / Pomegranate Origination
  - OSCAR AIST project to provide on-demand troposphere calibrations for SAR images (used for data subsetting & access via URL)
  - Altimetry services (on-demand netCDF variable slicing, visualization)
- Infusion into other Earth Science projects
  - Water Vapor Climatology MEASUREs project (HDF4/5 variable slicing)
  - SMAP decadal-survey mission plans to use it
  - Evangelized to JPL's CDX project (compare sat. obs. to model grids)
- Other JPL projects
  - Java version for a subset of file formats
  - Planetary image processing (handling VICAR image formats)
  - Multi-mission ground systems (HDF, VICAR)
  - Zhangfan is overloaded with feature requests

Technical Accomplishments

- Modernized TOPEX GDR, on equal footing with JASON -1/2
  - netCDF format with self-describing attributes, as similar to JASON-1/2 files as possible
  - New base GDR is the Retrack GDR
- Modular software components to update GDR
  - Updateable TOPEX components: orbit, tide model, mean sea surface, etc.
  - Extending to JASON-1/2 netCDF updates (in dev.)
- Expose GDR workflows as web services
  - Space/time subsetting
  - Workflows to update multiple components
  - Web interface to submit & monitor GDR update jobs
- Webification of science data containers (joint AIST & ACCESS work)
  - Drill into any netCDF, HDF4/5, GRIB, or FITS file anywhere in the world
  - Open standard, W10n-sci
  - Open source python implementation (downloadable pomegranate library)

Altimeter GDR Update Process Flow

Web-based Altimeter Service



Web Interface: Functional Capabilities

- Provide system to allow users to access and combine various parts of altimeter GDRs with new components (data, corrections, models) on demand.
  - Projects generate fundamental data record from telemetry with time, orbit, instrument information and basic corrections
  - Producers of other "components" of the altimeter record provide data to the Altimeter Service
    - Components can be data or data+operator(s)
    - Producers need to provide some documentation to guide use

- Use Scenario:

- **Login** to the Altimeter Service using email
- **Select** a cycle/pass range or a latitude/longitude **region**
- **Select** the desired choices of **components** from a menu
- **Update** altimeter GDR data on demand, using the selected components. Automatically generates Sea Surface Height Anomalies
- **Notify** user via email that job is complete
- **User retrieves** the updated GDR containing the parameters of interest in netCDF
- **Can Visualize** the key parameters SSH and SSHA on web page via "webification"

Some Usage Scenarios

- Generate basic altimeter data records: Project produces initial IGDR; all subsequent updates done with service – add POD, improved atmospheric models (observed replaces predict), any other improvements available on ~1 month time scale
- Update existing data with improved components: Particularly revised orbits, improved tides, radiometer calibrations, barotropic corrections, geoid, mean sea surface
  - Faster, independent of Project update cycle
  - Test versions of components
- Advanced/future development - Produce regional/coastal products: Select regional data, apply local tide models, radiometer corrections (processed to remove land effects), local barotropic models
  - Special retracking could be linked to original points
  - Additional Data Sources Available
    - Updated radiometer wet tropo computation, specifically for coastal areas
    - New version high resolution of Dynamic Atmospheric Correction (DAC)
    - Additional tide models (requires different software than GSFC models)
    - Additional mean sea surfaces (grid interpolation fixed)

Additional Data Sources

- Updated radiometer wet tropo computation, specifically for coastal areas
- Updates of orbits from GSFC: ftp://dirac.gsfc.nasa.gov
- AVISO (French data center): new version high resolution of Dynamic Atmospheric Correction (DAC) in netCDF available by ftp: ftp://ftp.cls.fr/pub/oceano/AVISO/auxiliary/dac/
- Baltic, North Sea models from Danish Meteorological Institute
- Coastal altimetry projects: PISTACH (CNES, CLS), ALTICORE (ESA, several agency consortium)
- Tide Gauge calibration check (G. Mitchum, U. South Florida)

First User Screen

Jet Propulsion Laboratory
California Institute of Technology

JPL HOME | EARTH | SOLAR SYSTEM | STARS & GALAXIES | SCIENCE & TECHNOLOGY
BRING THE UNIVERSE TO YOU: JPL Email News | RSS | Podcast | Video
search

Home

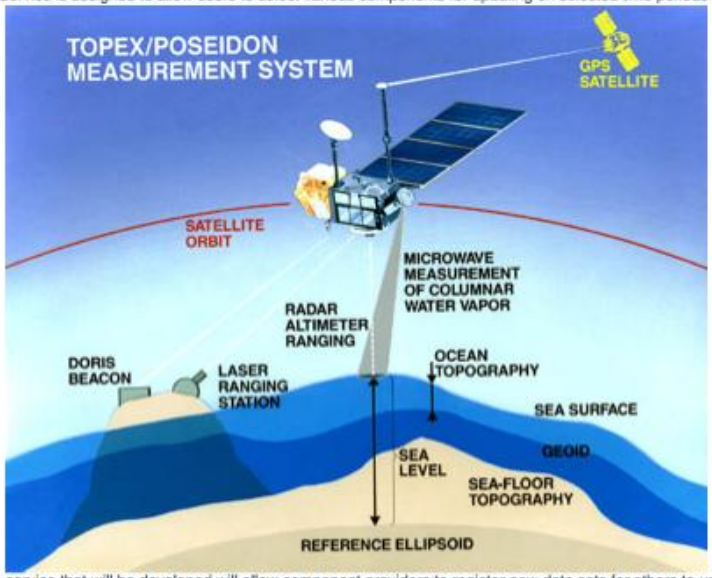
Datasets
- Base RDGR
- Canned RGDR 1
- Canned RGDR 2

Services
- Submit
- Check

Docs
- proposal
- 1st report
- poster

Welcome to the NASA ACCESS-sponsored Web-based Altimetry Service

Altimeter measurements consist of many components in order to arrive at the desired Sea Surface Height (SSH), SSH Anomaly (SSHA, difference of current measurement from a long term Mean Sea Surface (MSS)), or Ocean Surface Topography (difference of SSH from the geoid). The Altimetry Service is designed to allow users to select various components for updating on selected time periods or regions of data.



Another feature of the service that will be developed will allow component providers to register new data sets for others to use. This will speed the dissemination of the latest data to the broad community. Until the web-based feature is implemented, please communicate with Principal Investigator (Philip.s.callahan@jpl.nasa.gov, telephone: +1-818-354-4753) about having your component added to the service.

Second User Screen – Submit

Home

Datasets
- Base RDGR
- Canned RGDR 1
- Canned RGDR 2

Services
- Submit
- Check

Docs
- 1st report
- poster

Submit A Job

Please provide your email address:

Click to select a method:

by spatial and temporal constriants

by cycle and pass number

Check operators:

- [x] aws.component.orbit.Updater
- [ ] aws.component.orbit1.Updater
- [x] aws.component.tide.Updater
- [ ] aws.component.tide1.Updater
- [x] aws.product.ssha.Creator

- User enters email to send link to results, selects updates to do.
- Click on links for data subsetting: Cycle/Pass or Space/Time.
- When all selections are made, a job is submitted on the server. When the job is completed, an email is sent to the user with a link to retrieve the data.

- Data can be viewed through a browser as shown below.

Third User Screen – Webification

Web-based Altimeter Service

http://altiserv/aws/dataset/base_rgdr/cyc362/Trgdr_362_021.nc/?output=html

Parent URL

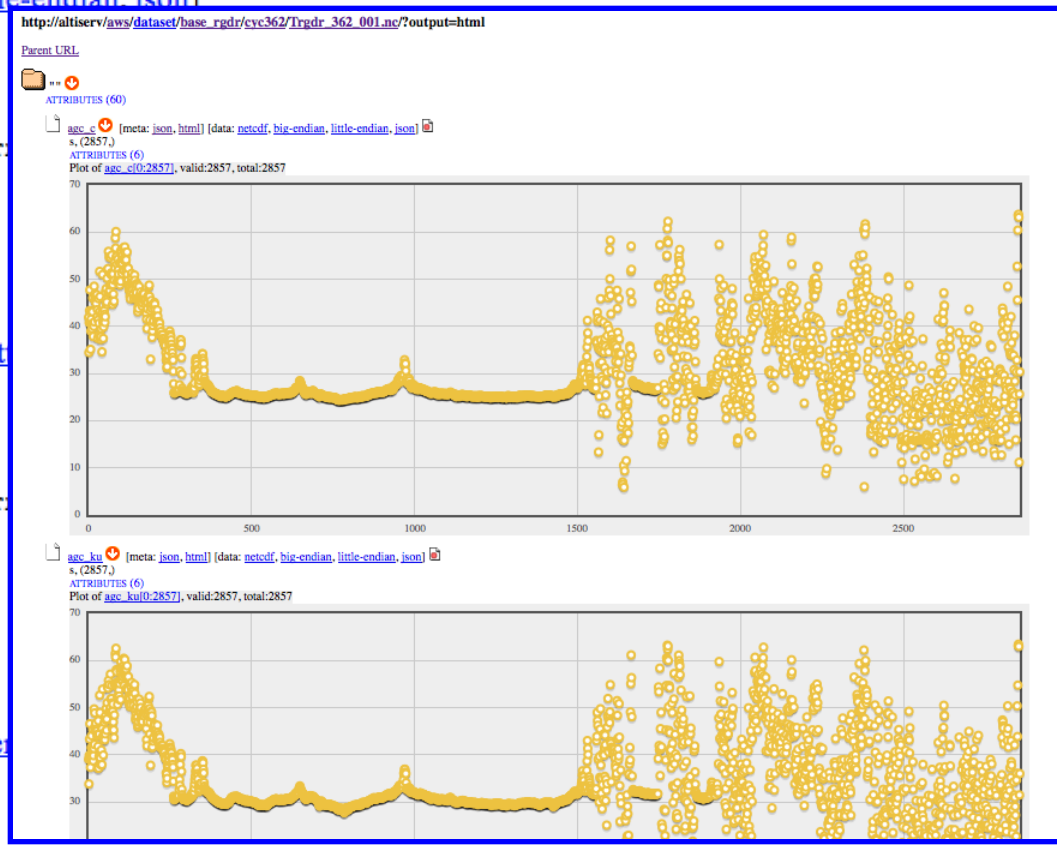
""

ATTRIBUTES (60)

agc_c [meta: json, html] [data: netcdf, big-endian, little-endian, json]
s, (2833,)
ATTRIBUTES (6)
_FillValue: [32767]
comment: AGC is corrected for instrumental er
coordinates: lon lat
long_name: C band corrected AGC
scale_factor: [0.01]
units: dB

agc_ku [meta: json, html] [data: netcdf, big-endian, lit
s, (2833,)
ATTRIBUTES (7)
_FillValue: [32767]
comment: AGC is corrected for instrumental er
coordinates: lon lat
long_name: Ku band corrected AGC
mgdr_name: agc_k
scale_factor: [0.01]
units: dB

agc_numval_ku [meta: json, html] [data: netcdf, big-e
1, (2833,)
ATTRIBUTES (7)



PODAAC Transition

- Made presentations at PODAAC Users Working Group, Coastal Altimetry Workshop in Oct '08. Received positive feedback
- Presented AltiServ demo to PODAAC in June 2009
- Worked with Thomas Huang (Infrastructure Lead), Cynthia Chen (Tools and Services Lead) and Andy Bingham (Manager) to integrate AltiServ into PO.DAAC infrastructure (more time consuming than expecteded)
- Originally planned to directly turn hardware and running server over to PODAAC.
  - Not possible, violates DAAC support parameters
- Provided installable package to PODAAC
  - Adapted to PODAAC-prescribed Linux OS version & web server configuration
  - Currently testing new installation
  - Public web site should be visible soon
- Operational within PODAAC (May 2011)
  - Public and private (for our use) web sites

Outstanding Work

- Finish transition to PODAAC, evangelize public services
  - Collect user metrics
- Some known additional components have not been installed. Data & software available, but ran out of resources.
  - Tides (different software than Goddard tides being used)
  - Mean Sea Surfaces (interpolation problems)
- Generalize update components to JASON-1/2:
  - Framework will work with Jason netCDF, just need to provide xml spec
  - Updated Goddard orbits (in progress)
  - Tides: add more global & regional models
- Component update is still a manual process: download files, change links/pointers
- Scientific vetting:
  - Vet additional update components within altimetry community

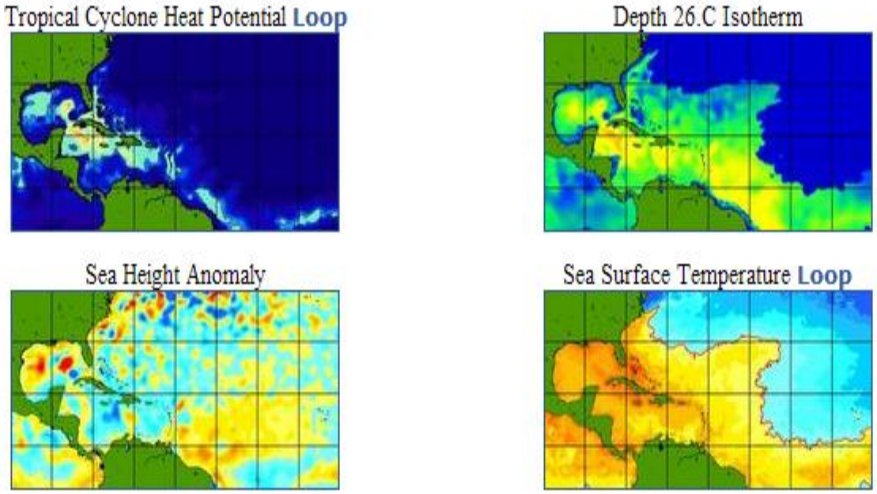
Backup Material

Details

Altimetry Benefits

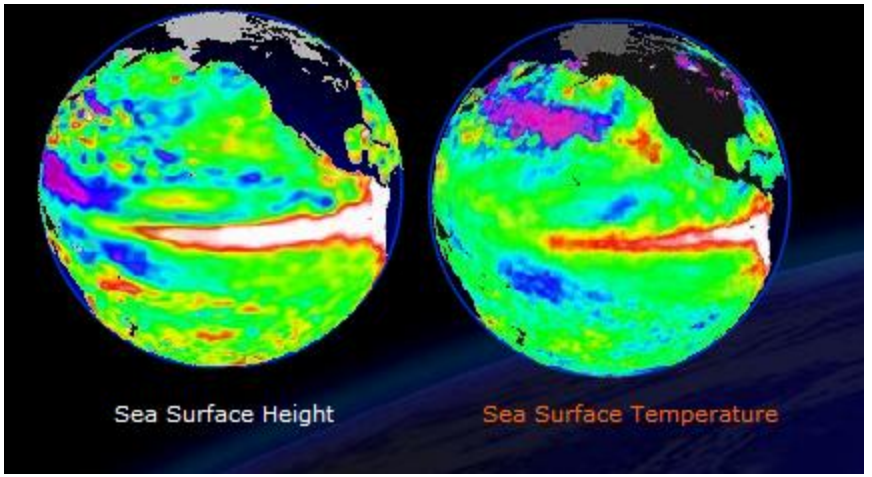
- Climate Studies – sea level rise, heat transport/distribution
- El Nino/La Nina – weather effects
- Hurricane Studies – Ocean heat content
- Ocean Currents – ship routing, off-shore industries, as well as science
- Physical-Biological ocean effects
- Environmental monitoring – pollution transport

Tropical Storms



From http://www.aoml.noaa.gov/phod/cyclone/data/at.html

El Nino



From http://sealevel/science/time-series-data.html

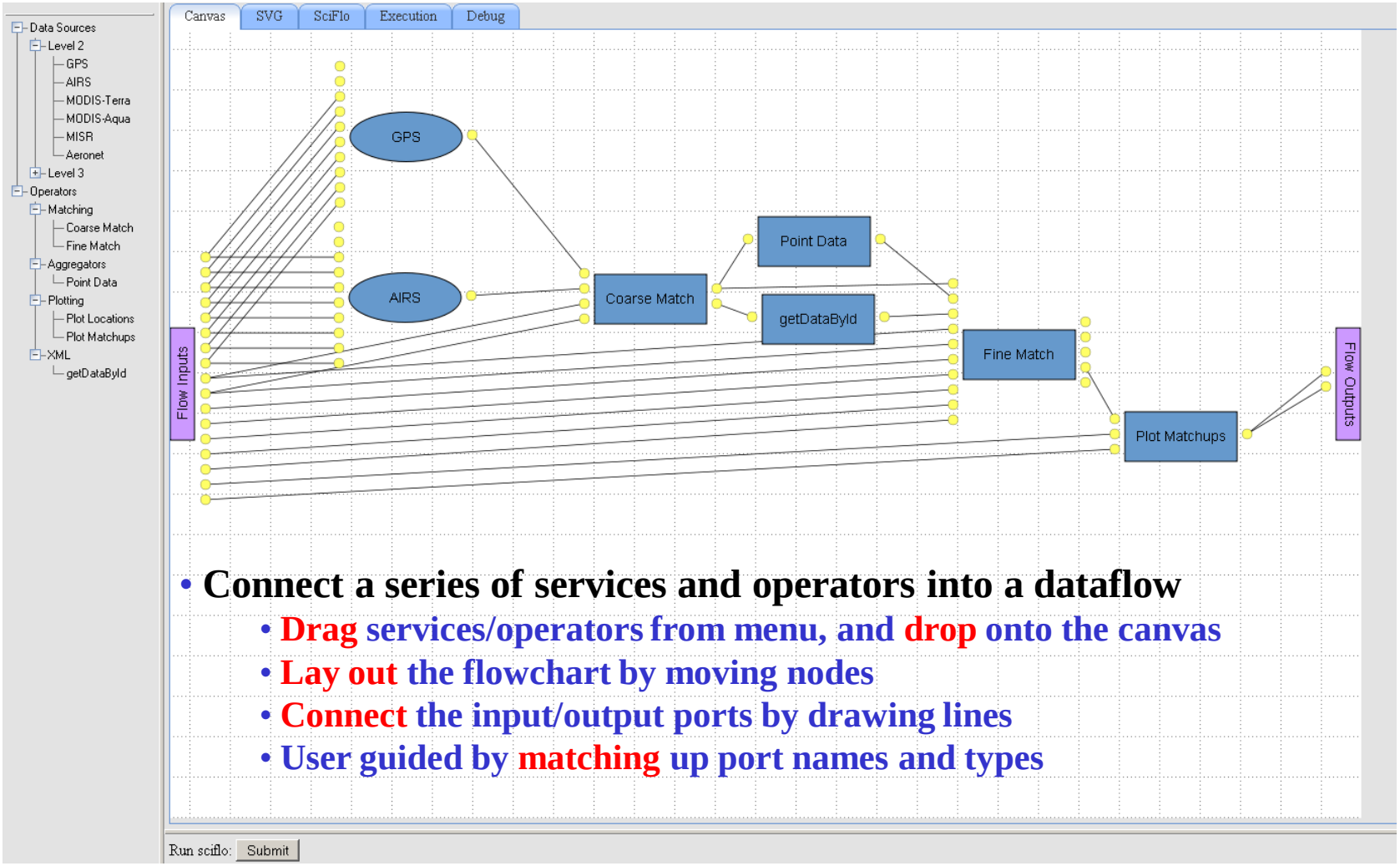
Development Approach

- Create web user interface to select data range, which components to update, output format
- Connect user interface to SciFlo system to generate work flows to carry out user requests: data access, algorithm control, output generation
- Modularize GDR and other algorithms to provide update, processing functionality
  - Algorithms for altimeter components such as orbit, tides, sea state bias, atmospheric corrections
  - Details of binding algorithms using python on next page
- Develop data subsetting (localization)
  - Key short term application to coastal applications
- Provide output formatting, mainly netCDF, parameter selection
- Work with scientists, data centers to get access to models and data sets
  - Key to long term utility is enlisting providers of new, improved components
  - Coordinate with PODAAC, AVISO, other data providers on International Altimeter Service
  - "Register" external models and data sets

Visual Dataflow Programming

Web-based Altimeter Service

GPS & AIRS Level-2 Space/Time Matchup



- **Connect a series of services and operators into a dataflow**
  - **Drag services/operators from menu, and drop onto the canvas**
  - **Lay out the flowchart by moving nodes**
  - **Connect the input/output ports by drawing lines**
  - **User guided by matching up port names and types**

HTTP Response (I)

Two types of responses: meta and data.

Meta Response:

- Defined for both node and leaf.
- Default output format is JSON. It can be HTML or other.
- For a node, the response body in JSON looks like

```
{'name':string,
 'attributes':[...],
 'nodes':[...],
 'leaves': [...],
 'w10n':[...]}
```

- For a leaf, the response body in JSON looks like

```
{'name':string,
 'attributes':[...],
 'type':string,
 'shape':[...],
 'dimensions':[...],
 'w10n':[...]}
```