SensorWeb 3G: Extending On-Orbit Sensor Capabilities to Enable Near Realtime User Configurability

Daniel Mandl[1], Pat Cappelaere[2], Stuart Frye[3], Rob Sohlberg[4], Vuong Ly[1], Steve Chien[5], Daniel Tran[5], Ashley Davies[5], Don Sullivan[6], Troy Ames[1], Ken Witt[7], Jason Stanley[7]

1 NASA/GSFC Greenbelt MD, daniel.j.mandl@nasa.gov
2 Vightel Inc. Ellicott City MD
3 SGT Inc. Greenbelt, MD
4 University of Maryland, Department of Geography, College Park MD
5 NASA/JPL Pasadena CA
6 NASA/AMES Moffett Field, CA
7 West Virginia High Tech Foundation, Fairmont, WV

***Abstract*- This research effort prototypes an implementation of a standard interface, Web Coverage Processing Service (WCPS), which is an Open Geospatial Consortium(OGC) standard, to enable users to define, test, upload and execute algorithms for on-orbit sensor systems. The user is able to customize on-orbit data products that result from raw data streaming from an instrument. This extends the SensorWeb 2.0 concept that was developed under a previous Advanced Information System Technology (AIST) effort in which web services wrap sensors and a standardized Extensible Markup Language (XML) based scripting workflow language orchestrates processing steps across multiple domains. SensorWeb 3G extends the concept by providing the user controls into the flight software modules associated with on-orbit sensor and thus provides a degree of flexibility which does not presently exist. The successful demonstrations to date will be presented, which includes a realistic HyspIRI decadal mission testbed. Furthermore, benchmarks that were run will also be presented along with future demonstration and benchmark tests planned. Finally, we conclude with implications for the future and how this concept dovetails into efforts to develop "cloud computing" methods and standards.**

I. INTRODUCTION

Our team has been developing various ongoing prototypes with increasing complexity to demonstrate an approach to interconnect sensors around the world and to enable easy access to the data from the sensors. Furthermore, we enable easy methods to combine various sensor data along with applying processing algorithms to provide users with customized data products.

In our demonstrations, we have used up to four satellites, one Unmanned Aerial System (UAS), multiple ground sensors, data algorithms and models in a variety of disaster management scenarios such as wildfires, floods and volcanoes. Mashup displays integrate a variety of data sources and trigger workflows that task sensors via a common interface based on Open Geospatial Consortium web service standards.

SensorWeb 3G takes the process one step further and provides an easy to use standardized user interface for the user to specify algorithms that will customize the raw sensor data being ingested onboard a satellite. The user interface is called a Web Coverage Processing Service (WCPS) which is compliant with the OGC specification of WCPS 1.0. The key challenge is how to actually build the user interface so that it is easy to use. Figure 1 shows the conceptual SensorWeb architecture with the WCPS along with the other affected portions of the architecture outlined in red.

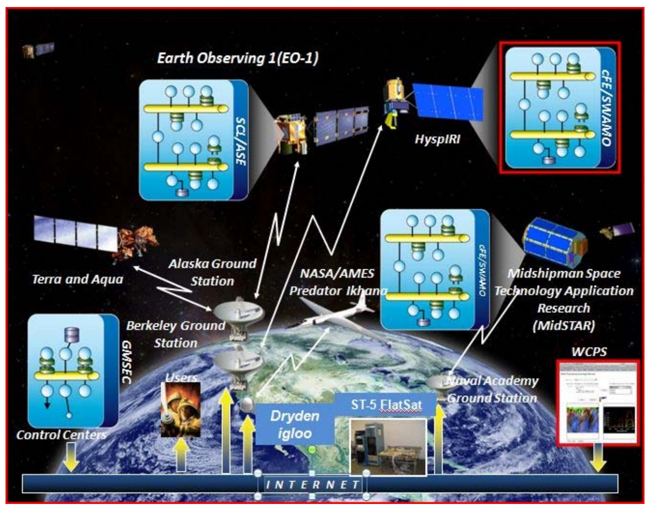

Figure 1 High level SensorWeb 3G architecture showing how the WCPS acts as an interface on the ground to define and dynamically upload algorithms to run onboard a satellite

II. WCPS OPERATIONS CONCEPT

Traditional flight software uploads tend to be complex and usually require software developers and operators. The vision of WCPS and SensorWeb 3G is to transform this software upload process to a user-driven self-service process. This is accomplished by separating the critical onboard functions from the science applications that run onboard and then to firewall the two processes. For example, on Earth Observing 1 (EO-1), there are two flight computers. One computer runs the

Command and Data Handling (C&DH) applications. In particular, the data handling portion refers to telemetry. The other flight computer handles the science recorder and any functions which have to do with science processing and onboard scheduling of images. There is also bridge software which allows the two computers to communicate so that the scheduling software can task the C&DH system. EO-1 has rudimentary capability to upload new science processing algorithms. But it is not user-driven and more like the traditional flight software upload process.

The WCPS adds the user interface to make a configuration like that of EO-1 more user-friendly and cost-effective to make changes. Figure 2 depicts one possible operations concept for the WCPS. In this scenario, we used a flight testbed that represents a conceptual design for the Intelligent Payload Module (IPM) for the HyspIRI decadal mission. Note that this configuration mirrors that of EO-1 in that the IPM is a separate computer system from the main C&DH computer targeted for HyspIRI.

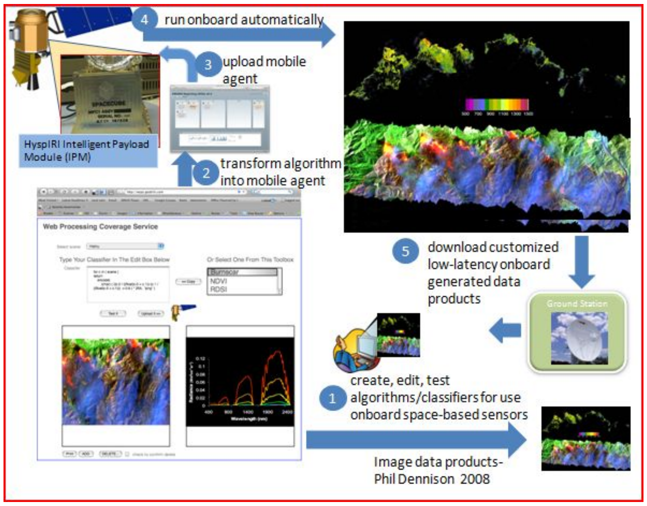

Figure 2 One possible operations concept for the WCPS using a HyspIRI (future decadal mission) IPM testbed

In the scenario in figure 2, a user either selects a pre-existing algorithm or defines a new algorithm in a standardized workflow language called Business Processing Management Notation (BPMN). Eventually, the BPMN will be generated by a visual tool, but for now, it has to written in XML following the WCPS 1.0 standards. The user can then test the algorithm using test data and view the result locally. If satisfied, the WCPS allows the user to upload the algorithm. Once the algorithm is uploaded, it is enabled , processes data from the instruments and then downloads the resultant data products. In the case of HyspIRI, the data products are downloaded via a Direct Broadcast antenna.

III. WCPS Functionality

The basic functions needed by the user to specify the onboard algorithms are a combination of basic primitive manipulation functions such as spectral band arithmetic, image generation and encoding in various formats. Table 1 below shows a partial list of WCPS 1.0 functions that were implemented and the percentage completion status as of the writing of this paper for our prototyping efforts. We only developed the ones we needed for the prototype effort.

Language Primitives	Completion Status
Processing Expression	100%
Store Coverage Expression	100%
Encoded Coverage Expression	100% (added kmz)
Boolean Expression	100%
Scalar Expression	80%
Get Metadata Expression	-
Set Metadata Expression	-
Coverage Expression	100%
Coverage Identifier	100%
Induced Expression	100%
Unary Induced Expression	80%
Unary Arithmetic Expression	90%
Trigonometric Expression	-
Exponential Expression	-

Table 1 Partial list of WCPS functions implemented for effort

IV. WCPS And Related Architecture

The WCPS is also a seamless part of SensorWeb architecture and thereby leverages all of the collaboration software and the corresponding standard interfaces. Figure 3 depicts the generalized SensorWeb architecture with the WCPS integrated with the Campaign Manager. The WCPS can reside anywhere, but for our SensorWeb, a close coupling of with the Campaign Manager made sense.

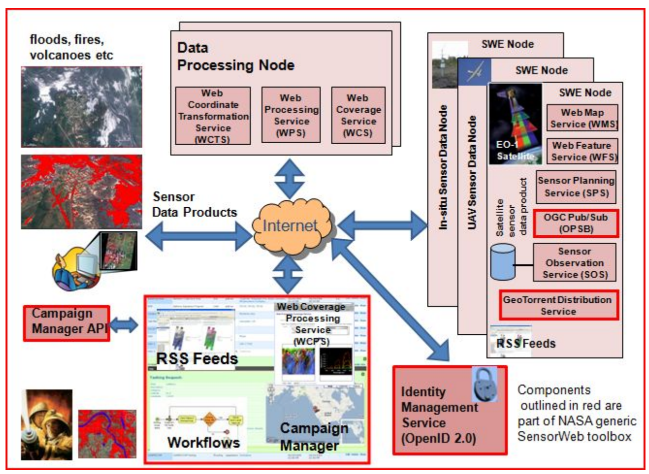

Figure 3 Generalized SensorWeb architecture with WCPS embedded within the Campaign Manager

We tested the WCPS concept by developing the IPM testbed, creating or selecting test algorithms via a WCPS, simulating the uploading and executing of the algorithms on the testbed and returning the customized data products via a simulated downlink. The WCPS uploads algorithms by transforming the algorithm into a mobile agent compliant with the Sensor Webs for Mission Operations (SWAMO) agent architecture standards that were previously defined on a NASA Advanced Information System Technology (AIST) 2005 award with West Virginia High Tech Foundation (WVHTF) as the Principle Investigator. The testbed and ultimately the satellite are configured with a flight executive called Core Flight Executive (cFE) that was developed at NASA/GSFC. WVHTF along with GSFC personnel integrated SWAMO and cFE so that when a mobile agent is uploaded, it knows how to plug itself in via cFE and then execute. Figure 4 shows the top level architecture for SWAMO/cFE that is integrated with the ML507 which is the board that we use as the primary portion of the HyspIRI testbed.

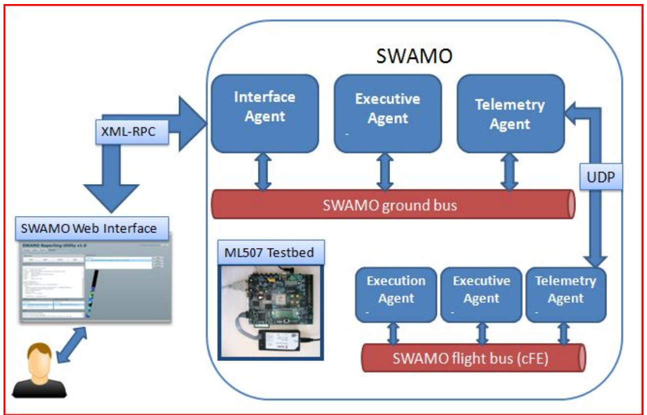

Figure 4 SWAMO/cFE architecture that uploads algorithms from the WCPS to execute on the testbed

V. BENCHMARKS

In order for this operations concept to be successful, a question that needs to be answered is whether the WCPS in conjunction with an onboard configuration such as an IPM has sufficient performance to be able to handle typical user requested algorithms. The team is in the process of investigating this question by using the HyspIRI science scenarios and the HyspIRI target configurations. In place of HyspIRI data, the team used EO-1 data which was scaled up in size to simulate the performance requirements needs of HyspIRI. Figure 5 shows the present baselined configuration of the IPM for HyspIRI.

So the key question with this configuration was whether the IPM can keep up with the very high data rates generated by the instruments, select subsets of the data and then also possibly process higher level data products for downlink. For the case of HyspIRI, one scenario might be user requests for a subset of the spectral bands when overflying certain geographical areas. Another possible scenario would the creation of customized data products based on uploaded algorithms. In both cases, data products would be downlinked as soon as possible when a direct broadcast ground antenna is in view.

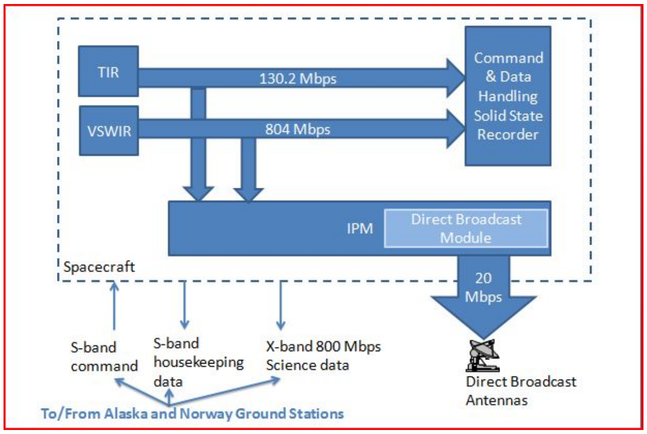

Figure 5 HyspIRI IPM configuration; note the high data rates emanating from the two instruments on HyspIRI, the Visible to Short Wave Infrared (VSWIR) imager and the Thermal Infrared (TIR) imager. Also, note the relatively low IPM downlink rate compared to the data rates generated by the instruments and the normal high science downlink bandwidth

Our tests consisted of running science algorithms that we run on EO-1 at present. We used a Space Cube board with ML507 Virtex boards consisting of 2 PowerPC's per board. We determined that at the composite data rate of about 930 Mbps from the instruments, we would have to break the stream into 4 parallel streams to allow the ML507 Virtex board to keep up with the ingest rate as shown in figure 6. Although for the testbed, we only used one ML504 board. We measured how long it would take to run the algorithm on each processor board with ¼ swath of the data stream.

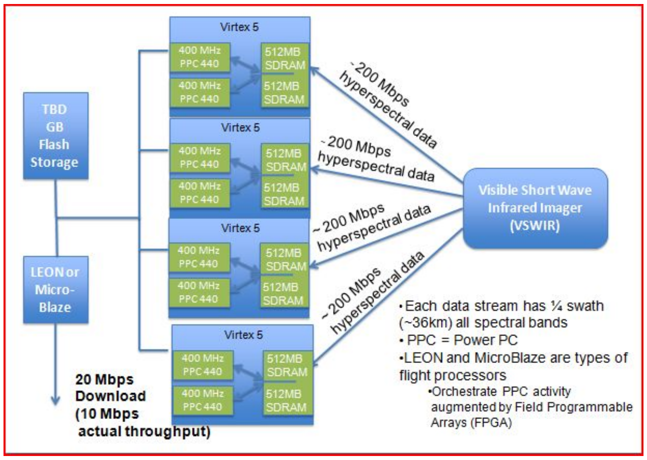

Figure 6 Space Cube configuration with four ML507 Virtex boards to handle the composite instrument data stream broken up into four parallel data streams

Table 2 shows the results of our benchmark tests. We did two types of tests. The first was to see how fast we could ingest the data to show that we could read the data into memory and get it out of memory fast enough to keep up with the data rates. We also ran six different algorithms that are used on EO-1 and benchmarked how long it would take to complete a ¼ swath of VSWIR for a 5 second data ingest.

32-bit Memory Test	Write (ms)	Read + Verify (ms)
128MB	711	1179
256MB	1564	2365
512MB	2942	4731
1024MB	6673	10670

Not Optimized!
FPGA not leveraged

Algorithms	Linux (ms)	Standalone (ms)	Linux (ms)	Standalone (ms)
	EO1 scene (256 x 1000 pixels)		HyspIRI ¼ swath (640 x 565 pixels)	
Cloud	1791	431	2170	589
Flood	3024	937	3782	1311
SWIL	7350	2872	10226	4058
Sulfur	116362	29515	164978	42026
Thermal	1103	304	1475	431
SIWI	580	44	823	62
NDVI	630	44	904	62
NDWI	589	44	836	62

Disclaimer: Code not optimized. Performance based on a 400MHz PPC design.

Table 2 Benchmark test results: The input and output tests showed that we could write and then read up to at least 256 Mbytes of data within the required 5 second window. A five second scene ¼ swath has 640 pixels x 540 pixels x 213 bands x 2 bytes per pixel = 155 Mbyte. Part two of the benchmark was to run eight algorithms with times being recorded for a standard EO-1 scene and also for a projected ¼ HyspIRI scene.

The concept of how the IPM would process the data is that there will be a ping pong buffer that ingests 5 seconds of data while the other buffer is processing the previous 5 seconds. Therefore, the worst case is multiple adjacent 5 second scenes, forcing completion of data ingestion, algorithm processing and downlink of the data products within 5 seconds for the worst case. Based on the second part of the benchmark in table 2, only some of the algorithms could meet this requirement. On the other hand, the tests with the ML507 were not optimized. It is expected that significant optimization will occur over the next year or so.

VI. Examples of WCPS Usage

This section lays out an example of how the WCPS might be used. In the first example, we assume a survey mission such as HyspIRI. The HyspIRI VSWIR instrument has 213 spectral bands. As it surveys land, it could be performing a search for desired spectral signature such as depicted in Figure 7. Depending on the available spectral signature libraries, this becomes a powerful method to monitor the environment for various materials of interest whether they are harmful algal blooms, fires, volcanoes or materials that are manmade and aggregated such as pollutants.

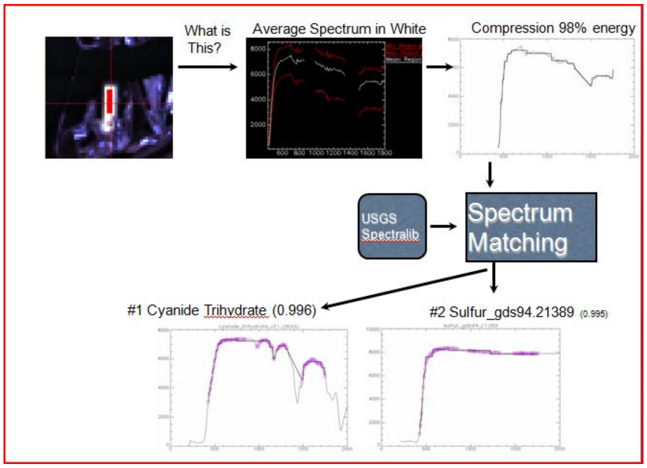

Figure 7 Spectral matching to locate natural phenomena or materials of interest

As an example, a WCPS algorithm was run on Hyperion data. An algorithm was run to find matching spectral signatures within the scene for potassium nitrate which can be the precursor to improvised explosives. Figure 8 depicts this process.

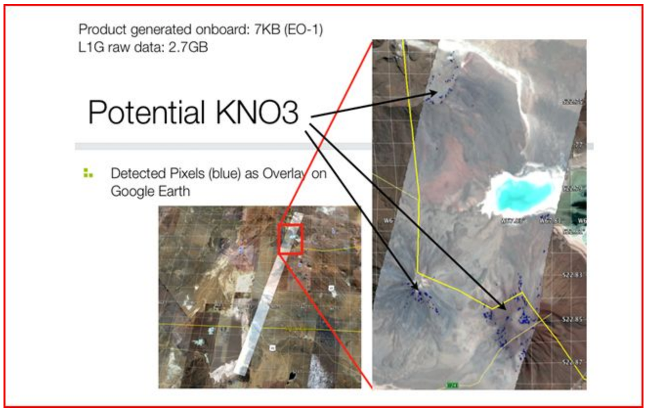

Figure 8 Detection of potassium nitrate with spectral matching using EO-1 imagery and generating the higher level product on a simulated onboard environment. The final product is a Keyhole Markup Language (KML) overlay that can be superimposed on Google Earth as shown in the figure.

Another example was conducted in the Galveston Bay area for the detection of sulfur. These is a known pile of sulfur, so an algorithm to detect sulfur was used and the detection was compared to the know location of the sulfur for validation. Figure 9 depicts this test.

VII. Conclusion: Evolution to Cloud Computing

The SensorWeb 3G concept with the evolution of the WCPS concept extends the idea of abstracting the world of sensors to the average users. This concept dovetails into the idea of "cloud computing" in which information is provided on-demand similar to electricity on a power grid. So in the case of SensorWeb, sensor data is provided on-demand with the user needing to know very little about the sensor itself. Sensing resources are abstracted and shared over the Internet.

This concept was presented at a session at Ground Systems Architecture Workshop in March 2010 in Pasadena CA for a session on "Data Center Migration for Ground Systems: Geospatial Clouds". So the concept of cloud computing and SensorWebs have a close link and the concept of a WCPS which allows configuration of one of the shared sensor resources further extends the power of the user in this environment.

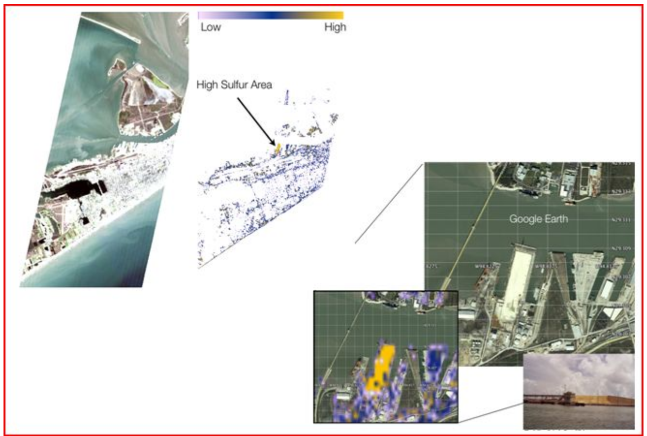

Figure 9 Detection of know sulfur pile in Galveston Bay.

REFERENCES

[1] D. Mandl, "Experimenting with Sensor Webs Using Earth Observing 1," IEEE Aerospace Conference Proceedings, vol. 1, pp. 183, March 2004.

[2] Mandl, Daniel, Rob Sohlberg, Chris Justice, Stephen Ungar, Troy Ames, Stuart Frye, Steven Chien, Pat Cappelaere, Daniel Tran, Granville Paules, "Experiments with User-Centric GEOSS Architectures", IGARSS 2007, July 22-27, 2007, Barcelona, Sp.

[3] Mandl, D., "OGC Standards to Enable SensorWebs for Disaster Management", GSAW 2010, Session 11D: Data Center Migration for Ground Systems: Geospatial Clouds, Manhattan Beach, CA

[4] http://eo1.gsfc.nasa.gov/new/extended/sensorWeb/sensorWeb.html