A New User Interface for On-Demand Cutomizable Data Products for Sensors in a SensorWeb

Daniel Mandl[1], Pat Cappelaere[2], Stuart Frye[3], Rob Sohlberg[4], Vuong Ly[1], Steve Chien[5], Don Sullivan[6]
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

***Abstract*- A SensorWeb is a set of sensors, which can consist of ground, airborne and space-based sensors interoperating in an automated or autonomous collaborative manner. The NASA SensorWeb toolbox, developed at NASA/GSFC in collaboration with NASA/JPL, NASA/Ames and other partners, is a set of software and standards that (1) enables users to create virtual private networks of sensors over open networks; (2) provides the capability to orchestrate their actions; (3) provides the capability to customize the output data products and (4) enables automated delivery of the data products to the users' desktop.**

A recent addition to the SensorWeb Toolbox is a new user interface, together with web services co-resident with the sensors, to enable rapid creation, loading and execution of new algorithms for processing sensor data. The web service along with the user interface follows the Open Geospatial Consortium (OGC) standard called Web Coverage Processing Service (WCPS). This presentation will detail the prototype that was built and how the WCPS was tested against a HyspIRI flight testbed and an elastic computation cloud on the ground with EO-1 data. HyspIRI is a future NASA decadal mission. The elastic computation cloud stores EO-1 data and runs software similar to Amazon online shopping.

I. INTRODUCTION

This SensorWeb research effort has been ongoing since 2003 with a team comprised of researchers three NASA centers; GSFC, JPL and Ames and the University of Maryland. The vision of the SensorWeb effort is to turn the Earth's sensors into data feeds which can be published over the Internet and can be accessed via subscription to authorized users. Users can create workflows which automatically create higher level data products, such as flood maps, and in turn make them data feeds. Figure 1 shows the overall high level SensorWeb architecture. The components outlined in red are the main categories of SensorWeb components. The Web Coverage Processing Service (WCPS) is outlined in a heavy red line and is the subject of this paper.

At the last ESTF in June 2010, a paper [1] was presented on the initial formulation efforts for the development of WCPS interfaces. At that time, some preliminary concepts were presented along with some benchmarks for performance of a WCPS in a HyspIRI testbed environment using EO-1 algorithms and EO-1 data to simulate HyspIRI data and the algorithms that would be used for that mission in the future. An actual WCPS interface had not been developed. At the writing of this paper, we have developed an actual WCPS that is running in our elastic compute cloud and on our HyspIRI testbed, both of which will be described later in the paper.

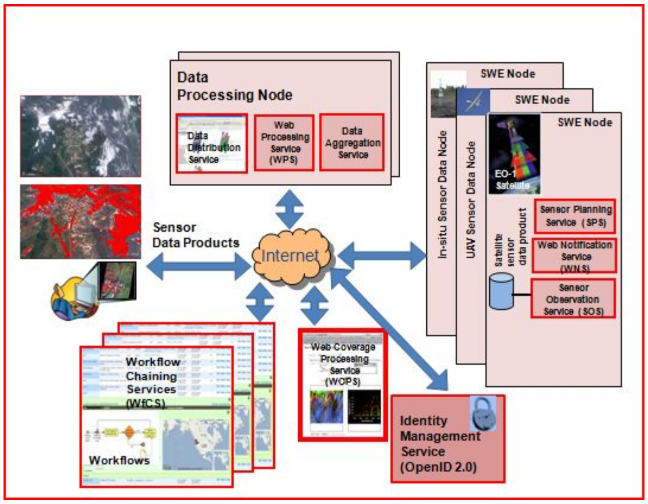

Figure 1 High level SensorWeb architecture with key components outlined in red and WCPS outlined in a heavy red line

II. TARGET MISSION ENVIRONMENT FOR WCPS

Traditional flight software uploads tend to be complex and usually require software developers and operators. The vision of WCPS as part of the SensorWeb Toolbox 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.

This architectural concept is extended into the future operations concept for the HyspIRI NASA Decadal mission via the Intelligent Payload Module (IPM). HyspIRI is presently scheduled to launch after 2020 versus the original target launch date in 2014. The insertion of the IPM into the HyspIRI mission concept, is to allow for a second onboard processor which will be separate from the C&DH computer, to service low latency users. Figure 2 shows the Computer Aided Design (CAD) generated picture of the HyspIRI satellite as it is presently conceived.

Figure 2 Picture of satellite as it is presently conceived

Figure 3 depicts the general architecture of the IPM. In addition to a high speed onboard processor, the IPM contains a Direct Broadcast antenna and electronics to allow rapid downlink of subsets of instrument data or higher level data products that would be processed onboard from the subset of data. The IPM will tap off of two instrument data streams that are on the satellite, the Visible through ShortWave InfraRed (VSWIR) imaging spectrometer and Thermal Infrared (TIR) scanner.

Note that the composite data rate of the two instruments is over 900 Mbps. Therefore the CPU for the IPM has to be very fast in order to tap off of this fire hose of data and to be able to process the data into higher level products. In order to have any chance to keep up with the data and produce the data products in a timely manner onboard, the team has been examining the use of multi-core and multi-tiled CPU architectures. Furthermore, the other question that the team is trying to answer is how to interface the WCPS to such a parallel processing environment.

Although this effort is mainly targeted for the HyspIRI mission, many of the future NASA Decadal Survey missions have a similar profile of high speed and high volume instrument data. There is a desire to provide data to low latency users such as government agencies and Non-governmental Organizations (NGO) that deal with disasters. Therefore, the IPM concept, along with the corresponding WCPS software would help to fill that need.

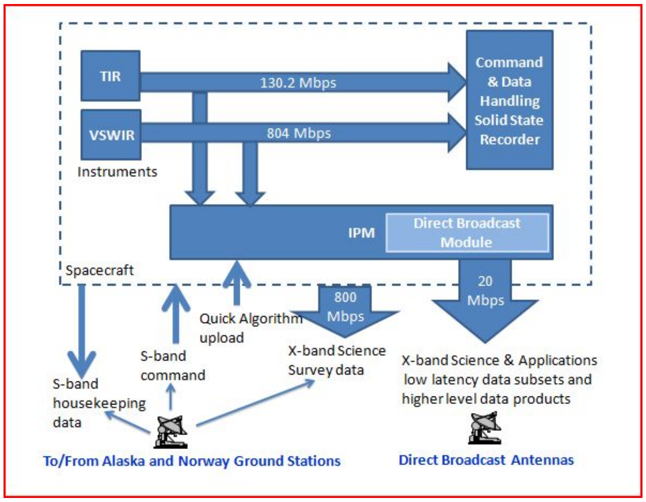

Figure3 Architecture of the IPM to be used for HyspIRI Decadal mission and possibly other NASA Decadal missions.

III. WCPS Operations Concept

Figure 4 shows a typical data flow for the use of a WCPS. Note that in this diagram, the Weka data mining tool is used to generate a desired algorithm. The custom data product pictured in the diagram is a flood extent classifier used over the country of Namibia. The satellite images used are from the Advanced Land Imager on EO-1. The output of the Weka tool is input into the Weka to WCPS translator which is another piece of software from the SensorWeb Toolbox. The output of the Weka to WCPS translator is the input into the WCPS client. The WCPS client is the user interface that is used to design the algorithm and upload the final results into a buffer of available algorithms. The output of the WCPS client is called a “parse tree” which is the instruction set that specifies to the target environment how to process the target data set from the instrument. The WCPS Runtime (WCPS-R) is used by the user to execute the algorithm against the specific data set specified by the user. This can be a data stream from an instrument when the WCPS-R is used in a satellite environment, or it can be one image of a stored data set when used on the ground . For example, when the WCPS-R is used in the elastic cloud environment which stores EO-1 data, the user is queried for the algorithm desired and the EO-1 scene ID that is desired.

Figure 4 is a generalized operations concept for the use of WCPS to generate an algorithm and to inject that algorithm into one of three environments; (1) satellite, (2) airborne vehicle such as an unmanned aerial system and (3) an elastic compute cloud on the ground. Note that in the case shown, the output product is in KMZ format and is put onto a Google Map which resides in the elastic compute cloud. The SensorWeb software component that ingests the flood extent classified image is the Flood Dashboard. The vision for our future process for flood extent images is that once the algorithm is

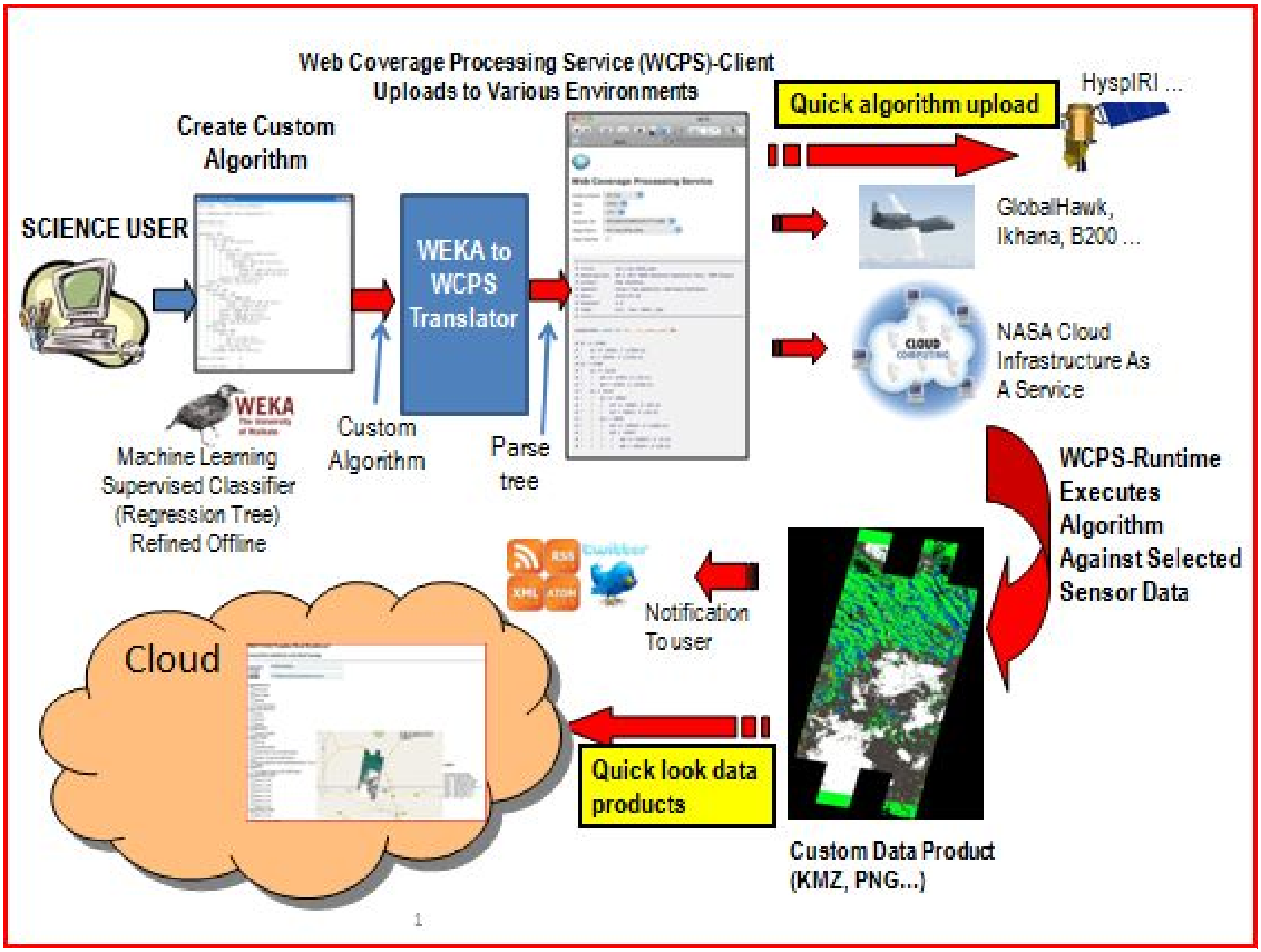

Figure 4 Top level operations concepts for WCPS in which a ground based WCPS client is used to design a algorithm and then the algorithm is uploaded to one of three environments. The user selects one of the created algorithms to run via the WCPS-R which causes the quick look data product to be produced and sent to the elastic compute cloud.

created and made available, a corresponding workflow would automatically run the algorithm and place the flood extent on the Flood Dashboard to be displayed in Google Map whenever an image of a flooded region was taken by EO-1.

The extended operations concept for the WCPS will be to make use of a Workflow Chaining Service (WfCS), as depicted in figure 1, to orchestrate the use of the WCPS. For example, one WfCS used in the SensorWeb Toolbox is GeoBPMS. It is used to task EO-1 and to control data processing and delivery. GeoBPMS has an entry field for theme. So if a user specifies that the image has a flood theme, GeoBPMS would specify the use of a flood classifier for data processing via the WCPS-R. Then a notification would be sent to the user to let them know that their flood classified image, along with the raw data, is available on the cloud.

Figure 5 shows a sample image from the EO-1 ALI

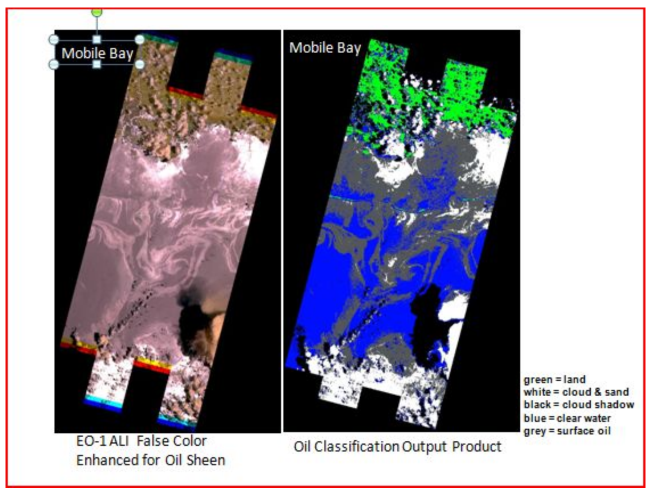

Figure 5 Sample EO-1 ALI image as input on left and classified with WCPS created algorithm for oil on water on right. The image is Mobile Bay in Louisiana

that was used to detect the oil spills in the Gulf of Mexico and created via the WCPS. The left panel is the input data and the right panel is the algorithm processed image.

Figure 6 depicts the present WCPS user interface that resides on the Geobliki on another cloud. It allows users to exercise the WCPS-C locally and to access the WCPS-R remotely.

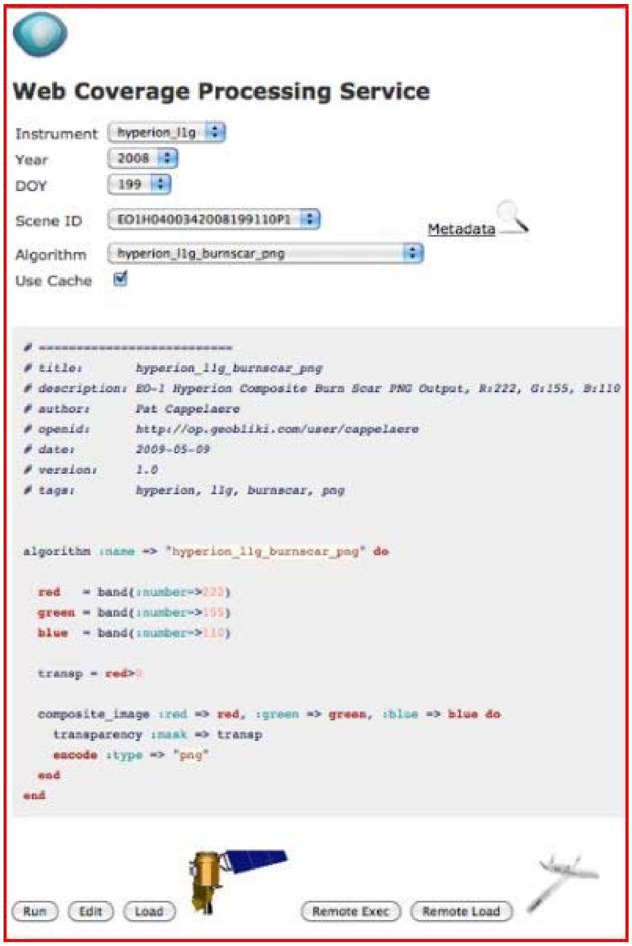

Figure 6 User interface for access to WCPS-C and WCPS-R

IV. WCPS AND CONCURRENT PROCESSING

As previously mentioned, the key to successfully implementing WCPS will be the ability to specify algorithms and then to run them on a concurrent architecture. This means the WCPS might have to specify how an algorithm can be broken up so that different parts can be run on different cores or tiles in parallel and thus increase the throughput performance. Figure 7 depicts an example in which a pan sharpening algorithm is specified and each of the three bands is allocated to different cores or tiles.

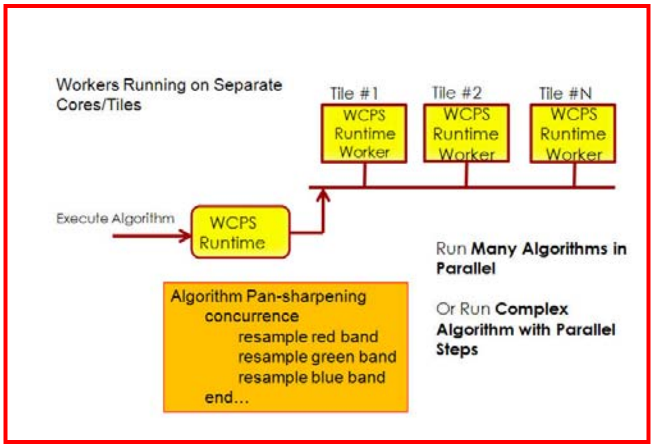

Figure 7 WCPS specified pan sharpened algorithm specified to run concurrently

The initial experiments are being done on the elastic compute cloud since it has over 300 cores. This is the easier task because it totally resides on the ground and more tools exist to control the parallel processing for ground systems than multi-core flight computers. In parallel, team members are devising methods to make use of the multiple cores on the HyspIRI flight testbed. Initial performance benchmarks were measured using the SpaceCube processors with EO-1 algorithms. These benchmarks will be used to compare the increase in performance as the parallel processing is integrated.

V. TESTING THE WCPS OPS CON WITH AN AIRBORNE MISSION

Since HyspIRI launch is far off into the future at this time, the team has been in discussions with ER-2 personnel and with Global Hawk personnel to possibly fly a WCPS configuration. A multicore processor would be put into a box and mounted

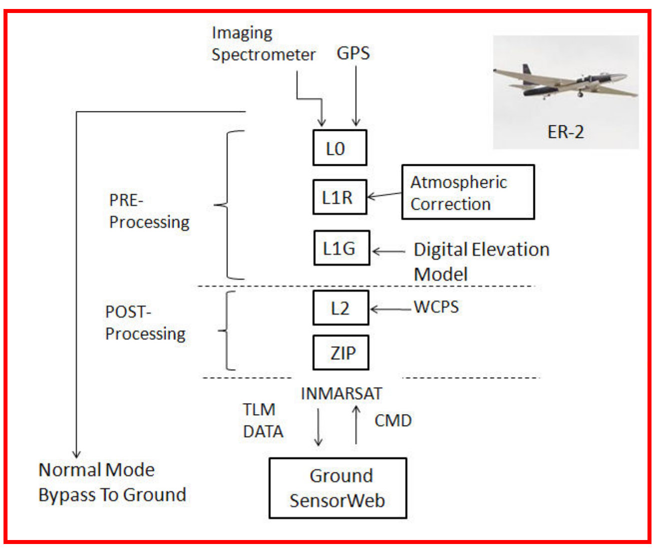

Figure 8 Possible ER-2 WCPS configuration for future test

into an ER-2 or Global Hawk with an instrument that is similar to that of HyspIRI. Figure 8 depicts the desired configuration of the IPM and the communications links that would be used to emulate the HyspIRI IPM operations concept. This would include an embedded WCPS. One possible mission being considered is the Enhanced MODIS Airborne Simulator (EMAS) mission which will be flown in summer 2012. The team is negotiating possibly flying in one of the engineering flights after the initial mission flight.

VII. Conclusion: Evolution to cloud computing

The WCPS concept increases user access and provides more flexibility for creating data products for sensor data. The WCPS paired with cloud computing and onboard parallel processing provides the next evolutionary step in data processing from the user perspective.

References

[1] Mandl, D; P. Cappelaere; S. Frye; R. Sohlberg; V. Ly; S. Chien; D. Tran; A. Davies; D. Sullivan; T. Ames; K. Witt; J. Stanley, "SensorWeb 3G: Extending On-Orbit Sensor Capabilities to Enable Near Realtime User Configurability", ESTF 2010, June 2010.

[2] http://hyspiri.jpl.nasa.gov

[3] http://sensorweb.nasa.gov

[4] http://eo1.gsfc.nasa.gov