

# Standardizing Value-Added Geo Raster Services

Peter Baumann, Sorin Stancu-Mara, Georgi Chulkov



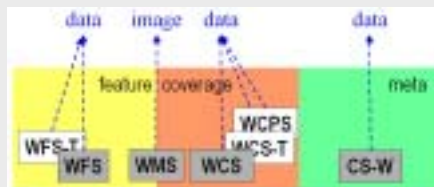
JACOBS  
UNIVERSITY

School of Engineering & Science  
Electrical Engineering & Computer Science  
Database Lab

The *Web Coverage Processing Service* (WCPS; OGC 08-068) specifies an expression language for server-side processing of large, multi-dimensional geo raster data for navigation, download, and ad-hoc analysis & mining purposes, in short: "SQL for raster data".

Raster data are gathered in the Earth Sciences in huge, rapidly growing amounts and in manifold variations. Examples include 1-D sensor time series, 2-D maps, 3-D  $x/y/t$  satellite image time series and  $x/y/z$  exploration data, and 4-D  $x/y/z/t$  climate and ocean models.

WCPS currently is under voting for adoption as a standard by the Open GeoSpatial Consortium (OGC, [www.opengeospatial.org](http://www.opengeospatial.org)). WCPS extends OGC's raster access standard, Web Coverage Service (WCS; OGC 07-067r5) which only offers data subsetting, rescaling, and reprojection of raster data.



## Background

The OGC, in collaboration with ISO, develops standards for geo-spatial and location based services. Three core standards accomplish Internet access to feature data (*Web Feature Service*, WFS; OGC 04-094), coverage data (*Web Coverage Service*, WCS; OGC 07-067r5), and meta-data (*Catalog Service*, CS-W; OGC 07-006r1), resp. While ISO 19123 (identical to OGC Abstract Topic 6; OGC 07-011) defines the term "coverage" in a very broad sense as "A feature that associates positions within a bounded space (its spatiotemporal domain) to feature attribute values (its range)"; in practice today only regularly gridded, i.e., "raster", coverages are standardised.

## WCPS Expression Language

The language allows to phrase operations combining one or more coverages, sent to a coverage server for evaluation. The response is either a coverage set or a scalar set (e.g., when summarizations or metadata are retrieved).

The general structure of a request is (nonterminals in italics):

```
for cov1 in ( coverageList ),
...
covn in ( coverageList )
[ where condition(cov1,...,covn) ]
return processingExpr(cov1,...,covn)
```

Following database standards, the language is declarative and optimizable. Further, it is safe in evaluation by avoiding explicit loops and recursion.

A semi-formal specification of the language is given in OGC 08-068. Concepts are based on experience with the *rasdaman* array algebra and query language (Baumann, 1994 & 1999).

## Advantages

- no client-side programming necessary; e.g., JavaScript request generation
- Clear semantics, allowing for automatic chaining and other optimizations
- integrates with OGC WCS and Web Processing Service (WPS)
- concepts proven for coverages up to 13 TB and 1D – 4D

## Syntax Example

"Inspect coverages *Modis1* to *3*. Mask each coverage's red channel using the *RegionMask* coverage, return it encoded in *HDF-EOS*."

```
for m in ( Modis1, Modis2, Modis3 ),
  r in ( RegionMask )
return
  encode( m.red * (r>0), "HDF-EOS" )
```

## Status & Outlook

As of May 2008, WCPS has been accepted by the WCS Working Group and is now forwarded to the OGC TC for final approval as a standard.

Future work will include

- promotion of the standard once adopted;
- continued standardization work in OGC, CGI/IUGS, etc.;
- Further research on efficient processing, such as hardware/software parallelization and automatic service orchestration;
- evaluation in as many different fields as possible, preferably in cooperation with domain experts.

## Selected Functionality Examples

- Coverage Processing

- Thresholding

$$NDVI(\alpha) = \frac{\alpha \cdot NIR - \alpha \cdot red}{\alpha \cdot NIR + \alpha \cdot red}$$



- Cross-Coverage Searching & Mining

- "Which time series exceed threshold T?"



- Coverage Summarization

- ...into scalars:

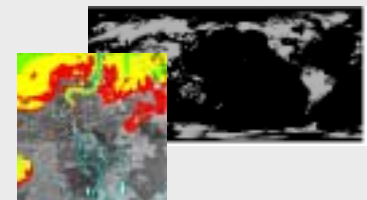


- ...into summary coverages



- Combining Coverages (sensor fusion)

- "Clouds over land"



- "WMS over WCPS"

## Implementation

WCPS has been implemented using a service stack consisting of clients (JavaScript, Java) and a server using Java servlets with Xerces and DOM for request translation, *rasdaman* for request execution, and PostgreSQL for coverage storage. As it turns out, the *rasdaman* array query optimization techniques increase performance significantly.

The implementation is showcased on the EarthLook website, [www.earthlook.org](http://www.earthlook.org).

Contact: Peter Baumann, [p.baumann@jacobs-university.de](mailto:p.baumann@jacobs-university.de)

Research sponsored in part by BIG Bremen.