Towards Simplified Coverage Services: 
*The New WCS 2.0 Standard*

**INSPIRE 2010**
Cracow, 2010-jun-24

**Peter Baumann**
Jacobs University Bremen, rasdaman GmbH

supported by ESA as part of the HMA-FO project

Disclaimer: my opinions, not necessarily those of OGC / ESA
Large-Scale Scientific Inf’ Systems (L-SIS) Research Group

- Jacobs University Bremen
  - 91 nations, 75% non-German, all-english on campus

- L-SIS Research Group:
  Flexible, efficient, & scalable services on massive multi-dimensional scientific data
  - rasdaman raster DBMS
  - OGC: co-chair of coverage WGs, editor of 8+ specs
OGC Web Coverage Service

- Coverage = "space-time varying phenomenon"
  - ISO 19123 (=OGC Abstract Topic 6)
  - Today typically raster, but more defined (curved grids, TINs, meshes, ...)

- Web Coverage Service (WCS) = coverage access service
  - Get original data (or subset thereof), suitable for further processing
  - As opposed to WMS: pictures, WFS: cannot subset
  - www.ogcnetwork.net/wcs

- Coverage-related working groups within OGC:
  - WCS.SWG (co-chair), Coverages.DWG (co-chair)
(Part of) The OGC Quilt
Sample WCS Based 3-D Service

[DFD-DLR, Diederich et al, 2001]
WCS History

- **WCS 1.0**
  - First attempt
  - Semantics not completely clear → restricted interoperability
  - Many implementations

- **WCS 1.1**
  - More comprehensive for improved interoperability
  - Monolithic (135 pages, of which ~65 about CRSs)
  - → perceived as complex, few implementations

- **WCS 2.0 design goals:**
  - Crisp, easy to handle
  - Harmonization with GML
  - coverages without WCS
  - Non-raster coverages
  - OGC's core/extension model
  - Concise semantics (incl. „pixel“ level)
  - broad range of different domains (EO, web mapping, climate/ocean, geology, …)
  - Support efficient & scalable implementations
Let’s Talk Tech: The Coverage Model

- **09-146: GML Application Schema for Coverages**
  - GML 3.2.1 coverage model
  - All coverage types of GML:
    - (non) referenced grids, point clouds, multi-curve, multi-surface, multi-solid

- ...with some missing information added
  - Range („pixel“) type description (data type, uom, ...), based on SWE Common
  - Hook for metadata

- **backwards compatible**: existing GML apps can safely ignore new parts

- Designed by WCS.SWG + GML.SWG + SWE writers jointly
WCS 2.0 Coverage Offering

class CoverageOfferings

«Data Type»
CoverageOfferings

+ wcsServiceMetadata: WCSServiceMetadata

+ offeredCoverage 0..*

«Data Type»
OfferedCoverage

1
+coverage

1
serviceParameters

GML 3.2.1 Application Schema for Coverages::
AbstractCoverage

details omitted here

WCS offering can be seen as a single virtual document

Hook for future service-related coverage metadata
WCS 2.0 Subsetting

- subset = trim | slice

- Core: CRS-based subsetting
  - raster, point clouds

- Extension for grid array indexing

- Extension for subsetting of other coverage types
The Big Picture: WCS 2.0 Core & Extensions

- Core
  - Data Model
    - Null Values
    - GeoTIFF
    - netCDF
    - JPEG2000
  - Coverage Formats
    - GML
    - netCDF
  - Service Model
    - WCS-T
    - WCPS
    - JPEG2000
    - CRS/EPDG
    - CRS/general
  - Protocol Bindings
    - GET-KVP
    - POST-XML
    - SOAP
  - Usability
    - Multilinguality
Inset: WCPS (Web Coverage Processing Service)

- "raster SQL": ad-hoc navigation, extraction, aggregation, analysis
- Time series
- Image processing
- Summary data
- Sensor fusion & pattern mining

- current value is 2220.6;
- average over all values up to now currently is 7461.7692307692305.
WCPS 1.0, WCS 2.0 Reference Implementation

- rasdaman=
  raster data manager

- Supports
  WCS, WCS-T, WCPS, WPS
  - WCS 2.0 expected by fall 2010

- Fast, scalable raster server
  - proven with dozen-TB objects

- Free, open-source
  - www.rasdaman.org

PostgreSQL

relational DBMS

rasdaman:
  translator
  parser
  raster engine
  metadata

clients
  visual clients
  cmd line clients

Baumann :: WCS 2.0 :: INSPIRE 2010
Some Current Activities

- **HMA-FO task 3 (ESA)**
  - WCS for EO product distribution
  - WCS 2.0; WCS EO AP; OS implementation, …

- **VAROS (ESA)**
  - WCPS as WPS Application Profile
  - Bridging coverages & processing: WCS – WCPS – WPS

- **Vightel / NASA:**
  - WCPS as ground/space interface for EO-1 / Hispiri
  - satellite can answer ad-hoc raster queries
Summary

- OGC WCS 2.0: standards suite for open, interoperable, scalable coverage access
  - Crisp, modular, powerful (non-raster!), thoroughly evaluated across domains
  - Towards unified coverage model across all OGC, harmonized with SWE Common, GML, WCPS, WPS, soon O&M
  - Status: under TC vote, adoption expected by August 2010

- Implementation
  - Demo: www.earthlook.org

- Next: extension writing, collaborative application projects
Motivation

- Exponential data growth on the Web
  - Today often raster data

- Move data to evaluation site?
  - 1 TB: > 1 day; 1 PB: 3 years
  - Huge volumes moved, only part needed (10:1) [Kleese 2000]

- Consequences:
  - Analyse close to data source
  - Deliver to exact needs → no bandwidth waste, higher quality of service
  - Flexible, intelligent retrieval

- Transition from data interoperability to service interoperability
Raster Data Ev'rywhere...

sensor feeds

raster server

[Image source: unknown/SWE]
Raster Data Ev'rywhere...

sensors feeds

raster server

Semantic Web
Optimization Example 1:
Just-In-Time Compilation

- **Observation:** interpreted mode slows down

- **Approach:**
  - cluster suitable operations
  - compile & dynamically bind

- **Benefit:**
  - Speed up complex, repeated operations

- **Variation:**
  - compile code for GPU

```
for x in (float_matrix)
return x*x*...*x
```

Times [ms] for $512^2 \cdot n$ ops

[Jucovschi, Stancu-Mara 2008]
"From MODIS scenes M1, M2, and M3, the absolute of the difference between red and nir, in HDF-EOS"

```python
for c in (M1, M2, M3):
    return encode(abs(c.red - c.nir), "hdf")
```
"From MODIS scenes $M_1$, $M_2$, and $M_3$, the absolute of the difference between red and nir, in HDF-EOS"

• ...but only those where nir exceeds 127 somewhere

```python
for c in ( M1, M2, M3 )
    where
        some( c.nir > 127 )
    return
        encode
            abs( c.red - c.nir ),
            "hdf"
        )
```
"From MODIS scenes \textbf{M1}, \textbf{M2}, and \textbf{M3}, the absolute of the difference between \textbf{red} and \textbf{nir}, in HDF-EOS" 

- ...but only those where nir exceeds 127 somewhere
- ...inside region R

```python
for c in (M1, M2, M3), r in (R)
where
  some(c.nir > 127 and r)
return
  encode
    abs(c.red - c.nir),
    "hdf"
(hdf_A)
```
So Why "Semantic"?

- Formal semantics for language allows machine-machine communication, no human intervention required
  - Clients (other services?) can compose requests

- Ex:
  - Client: "Let's see, which server can handle reprojection / exponentials / ... ?"
  - In a cloud: "hm, this subexpression I better pass on to node X"
  - "Evaluating this request will take an estimated 3.5min, over 500 objects match."
  - "Sorry, this request's complexity exceeds your CPU quota"
class GML 3.2.1 Application Schema for Coverages

AbstractCoverage

+ coverageFunction: gml:CoverageFunction [0..1]
+ metaData [0..1]

inherits attribute id from AbstractGML

domainSet
rangeType
rangeSet

Union
GML 3.2::DomainSet

GML 3.2 is the namespace of GML 3.2.1 [OGC 07-036]. Substructure omitted here.

Data Type
GML 3.2::AbstractFeature

Data Type
swe:DataRecordPropertyType

swe is the namespace of SWE Common 2.0 [OGC 08-094]. Substructure omitted here.

Data Type
GML 3.2::RangeSet

GML 3.2 is the namespace of GML 3.2.1 [OGC 07-036]. Substructure omitted here.
WCS 2.0 Service Model

- **Request Types:**
  - GetCapabilities
  - DescribeCoverage
  - GetCoverage

- …but now concisely defined semantics:
  
  response = GML document, pruned from coverage offerings
  
  - Conceptually!
Outlook: WCS Earth Observation Application Profile

Legend:
- **mandatory**
- **choice**

WCS core

WCS extensions

WCS app profiles
- Sensor AP
- EO AP
- MetOcean AP
- ...
Multi-Source Merge & Delivery

SOS

coverage

database

WCS-T

WCS

WCPS

s_1

s_2

... s_n

s_1

s_2

... s_n

u_1

u_2

... u_m