Accepted for publication at:
23rd FIG Congress, Munich, Germany, 8 – 13 October 2006

Towards a Standard for Interoperable Earth System Raster Services

Peter BAUMANN, Germany

Key words: raster data, image data, coverages, geo services, standards, interoperability

SUMMARY

Online services for multi-Terabyte satellite imagery are becoming integral part of Internet users' life. The general concept of multi-dimensional spatio-temporal raster data covers 1-D sensor time series, 2-D imagery, 3-D image time series and exploration data, 4-D climate models, and many more.

Standards for interoperable geo services are mainly developed by the Open GeoSpatial Consortium (OGC). Among them is the Web Coverage Service (WCS) which allows to retrieve all or part of a coverage (i.e., raster) object. Recently an initiative has started on a Web Coverage Processing Service (WCPS) allowing, based on the conceptual model of WCS, to submit QL-style requests for online data navigation and analysis in a set-oriented coverageexpression language. WCPS currently is an OGC approved Best Practices Paper. A reference implementation is under work.

This paper presents WCPS concepts with the intent of involving the research community at an early standardization stage to gain feedback and comments, and also to stipulate research in this area.

Towards a Standard for Interoperable Earth System Raster Services

Peter BAUMANN, Germany

1. MOTIVATION

Since the launch of World Wind and Google Earth and its high publicity at the latest it is clear that online services for multi-Terabyte satellite imagery are becoming integral part of Internet users' life[1].

This was not always clear. Over the years we have experienced a – sometimes even unwilling – shift from "our users don't want that" to "we need to offer this", actually again and again with every technology step. The result often was adoption of some well-known, but severely limited system which barely fulfilled the needs of the day; every subsequent requirement tended to involve a major system evolution or exchange. Based on these observations, one main motivation of our work is to establish services powerful enough so that they can evolve together with the requirements; a good example there of is database technology where DBMSs nowadays are extremely configurable to adapt to changing needs by adjusing the configuration (schema, indexing, etc.) rather than without having to deploy a completely new system. Hence, our first requirement on server technology is built-in extensibility in terms of both data structures and operations available on them.

Actually, 2-D imagery is but the tip of the iceberg - the general concept of multi-dimensional spatio-temporal raster data covers 1-D sensor time series, 2-D imagery, 3-D image time series (*x/y/t*) and exploration data (*x/y/z*), 4-D climate models (*x/y/z/t*), and many more. Application domains for raster data with some spatio-temporal semantics encompass geodesy/mapping, geology/geophysics/geochemistry, oceanography, environmental monitoring, climate modelling, architecture, engineering, and many more. Hence, our second and third requirement is support for n-dimensional rasters and flexibility for cross-domain applications.

The Open GeoSpatial Consortium[2] (OGC), in collaboration with ISO, today is the main driving force in open standards for interoperable geo services. A particularity of the significantly industry driven work of OGC is that every standard, in order to get approved, must have an implementation; further, user and interoperability experiments are a regular means to challenge and verify standards concepts. These mechanisms help to prevent from standardising concepts that are not usable or off topic. Hence, on the side we note a requirement of implementability and technical appropriateness.

---

[1] even though both are not the first of their kind.
[2] www.opengis.org

A family of standards have been developed for various tasks, such as the Web Feature Service (WFS) for vectorial data; the Web Map Service (WMS) for geo data rendering; catalog services; sensor services; and many more. The OGC Web Coverage Service (WCS) is the standard for basic retrieval from large-scale, multi-dimensional raster data. OGC and ISO prefer to use the term coverage and define it as follows [ISO 2002]: "*A coverage is a function from a spatiotemporal domain to an attribute domain. A coverage associates a position within a spatiotemporal domain to a record of values of defined data types.*" This abstract notion of a (raster) coverage as a function is also supported in [Baumann 1999]. Actually the term coverage is seen wider by OGC, generally encompassing "*space-varying phenomena*" including rasters, triangulated irregular networks, point coverages, and polygon coverages etc. [ISO 2002]. In today's practice, however, only raster-type coverages are really considered in both WCS and WCPS specifications. It is intended, though, to extend this in future.

WCS offers a fixed set of six operations, tailored towards retrieval of a coverage or some cutout thereof (see Section 2.2). The author's opinion that this should be paired by a mechanism to allow nested expressions over time was more and more supported by change requests wishing to add this and that special function to WCS. Among the examples was the NDVI (Normalized Difference Vegetation Index) and aggregation functions like sum, max, min, and avg. It turned out that this developed into an open-ended list which calls for a generic mechanism allowing users to phrase their specific data retrieval operation. In terms of requirements we have captured this core criterion already under "extensibility".

To fill this gap an initiative has started to develop a Web Coverage Processing Service (WCPS) allowing, based on the conceptual model of WCS, to submit query language style requests for online data navigation and analysis in a set-oriented coverage expression language. WCPS currently is an OGC approved Best Practices document [OGC 2006][3]. A full reference implementation is under way, based on the rasdaman raster server middleware, which also serves the scientific purpose of studying modelling and optimization of WCPS queries.

This paper presents WCPS concepts with the intent of involving the research community at an early standardization stage to gain feedback and comments, and also to stipulate research in this area.

The remainder of this paper is organized as follows. In Section 2 we discuss some related standards. Section 3 introduces WCPS coverage model as well as the expression language. Two use cases drawn from earth observation and climate modelling are presented in Section 4. Section 5 addresses implementation issues, and Section 6 gives conclusion and outlook.

2. RELATED STANDARDS

In this section we briefly review the most relevant existing coverage standards.

[3] Best Practice Documents, in OGC nomenclature, "*are an official position of the OGC and thus represent an endorsement of the content of the paper*." (see OGC Portal)

2.1 ISO 19123

The historically first standard for geo raster data is ISO 19123 [ISO 2004]. This standard attempts a comprehensive definition of the full spectrum of coverages, subdividing the general CV_Coverage class into discrete and continuous coverages. Discrete coverages are addressed by the class CV_DiscreteCoverage. The CV_ContinuousCoverage class is further subdivided into CV_ThiessenPolygonCoverage, CV_ContinuousQuadrilateralCoverage, CV_HexagonalGridCoverage, CV_TINCoverage, and CV_SegmentedCurveCoverage. For each class a set of properties and operations is indicated which precisely specifies the particular coverage type's semantics.

19123 does not specify a particular implementation or even a service on coverages stored on some computer, it rather defines the coverage data structure. As such, it is not useful (nor intended) to build a Web service upon, however, it may serve as the terminological basis for other standards, such as WCS and WCPS below. Consequently, the 19123 compliance testing method is "*Inspect the documentation of the application schema or profile*".

One current challenge is to update 19123. For example, in its definition section ISO 19123 describes raster as a "*usually rectangular pattern of parallel scanning lines forming or corresponding to the display on a cathode ray tube*". This obviously misses to include n-dimensional rasters, it lacks the commonly accepted definition of raster images as functions from a (spatio-temporal) domain into some (possibly complex) value space, and the description style actually may introduce a – technically not justified – line-by-line access.

The WCS Revision Working Group has spotted several items in this respect (such as the underlying coverage model); by exploiting the tight OGC/ISO liaison it is can be assumed that the according change requests will soon be formulated.

2.2 WCS

WCS specializes on retrieval from coverages, with emphasis on providing available data together with their detailed (metadata) descriptions and returning data with its original semantics (instead of pictures, like, e.g., WMS).

A coverage is seen as a 2-D, 3-D, or 4-D matrix (aka tensor) of values. The axes have an *x*, *y*, *z*, or *t* semantics. Following the functional viewpoint, the axes collectively are called coverage *domain*, the coverage values form the coverage *range*. A coverage's range consists of a list of fields; each field either can be atomic (such as temperature), or composite again. A composite field consists itself of an n-D tensor which can be addressed along the axes (in WCS speak: keys chosen from keylists) very much like array indexing. The difference between domain and range axes is that domains have a strictly spatio-temporal semantics, while range keylists allow abstract entities to serve for addressing. This way "abstract" dimensions such as *x/y/z* windspeed in a climate simulation can be modelled. To accommodate orthorectified/georeferenced, georeferenced, and non-georeferenced imagery, a coverage can bear both a ground Coordinate Reference System (CRS) and an image CRS. Addressing, then, is possible via either CRS. Finally, a bounding box is associated with each coverage, expressed in the resp. CRS(s).

Operationally, WCS offers three request types: `GetCapabilities`, `DescribeCoverage`, and `GetCoverage`. Requests and responses are exchanged via http, using either key/value pair (KVP) or XML encoding; the XML structures are laid down using XML Schema.

A WCS client is supposed to first issue a `GetCapabilities` request (like with any OGC service) to obtain a description of the services available (such as the coordinate systems and data formats supported) and a summary description of the coverages offered by this server. The client then can issue one or more `DescribeCoverage` requests to get all relevant metadata details about particular coverages (such as CRSs, bounding boxes, and range structure), sufficient for subsequent retrieval. This is done via a `GetCoverage` request. `GetCoverage` offers a set of six operations to be executed on a coverage, controlled by the request parameters. Processing steps not needed can be skipped by appropriate parameter settings. The six operations are assumed to be executed in a particular sequence (to obtain a well-defined result), although a server implementation may internally well deviate for optimization purposes, as long as the result is guaranteed to be identical. The operations are, in canonical sequence:

- spatio-temporal domain subsetting
- range subsetting (aka “band” selection, addressing via fields and their keys, if defined)
- resampling (e.g., scaling)
- reprojection
- data format encoding
- result file(s) delivery; results can optionally be delivered directly in the GetCoverage response, or stored by the server for later download by the client.

This WCS mimics as described is currently being finalised as version 1.1, to be released in Fall 2006. In parallel, evaluation implementations are conducted to assure implementability of the concepts.

2.3 WPS

The OGC Web Processing Service (WPS) Implementation Specification standard [OGC 2005] serves to describe any sort of GIS functionality, on both vector and raster data. A WPS may offer calculations as simple as subtracting one set of spatially referenced numbers from another (e.g., determining the difference in influenza cases between two different seasons), or as complicated as a global climate change model. WPS provides mechanisms to identify the spatially-referenced data required by the calculation, initiate the calculation, and manage the output from the calculation so that it can be accessed by the client. Description of a service is in terms the service name, input and output parameter structures, and a fulltext derscription.

As such, WPS is particularly suitable to “webify” legacy geo services by wrapping them into aka “geo SOAP”. Like SOAP, a particular shortcoming of WPS is that the real semantics of an operation (“what it really does”) is hidden in the operation name and the full text description – with the consequence that there is no machine-readable semantics specification, no automated verification, and no automatic service chaining.

WCPS, on the other hand, by having a clear and concise semantics specification, overcomes this and allows indeed to advertise services ready for assessment by machines. For example, a WCPS engine might analyse a request, find out that it can’t handle all operations needed to fulfil it, and automatically forward sub-tasks and intermediate results to other well-known services, and collect back and integrate their partial results.

To clarify this relation, WCPS contains a non-normative Annex describing a mapping of WCPS service description to the (much looser) WPS structure.

3. WCPS

Among the challenges in developing WCS is to draw the fine line between a high-profile standard which is function-rich, but hard to implement and use, and an “80:20” standard which may not support all imagineable use cases, but is easier to handle. Indeed, this question is carefully being pondered with each particular facet to achieve the optimal balance.

One decision was to deliberately exclude processing functionality beyond subsetting, scaling, and reprojection. The Web Coverage Processing Service (WCPS) fills this gap by extending WCS functionality with a coverage expression language allowing requests of unlimited nesting complexity.

The advantage of having two different standards, WCS and WCPS, lies in the modularity gained: implementers can choose to implement only the – relatively simple – WCS, or to undertake a WCPS implementation, which is significantly more challenging. The same choice is available to service operators, i.e., the customers of the software vendors.

In the remainder of this section we outline the WCPS concepts. For details we refer to the OGC WCPS document [OGC 2006].

3.1 Coverage Model

The WCPS coverage model deliberately lends itself towards the WCS coverage model, and it is foreseen that both models in future keep in lock-step synchronization to ensure smooth interoperability.

One current exception is an extension of WCPS to allow not only spatio-temporal domain axes, but also abstract domains, i.e., axes without any spatial or temporal semantics associated[4]. Examples of such abstract axes are simulation time for climate model computations, input parameter spaces for climate models, or statistical applications. All axes are treated equally in the operations; for example, subsetting and slicing operations can be performed on every axis.

We introduce the coverage model only informally here, a formal treatment of a similar model can be found in [Baumann 1999]. A *coverage* consists of a multi-dimensional array (in the programming language sense) plus associated metadata that describe it and allow access to it. The array elements, called *cells*, all share the same *cell type*. Each cell has a cell coordinate taken from $\mathbf{Z}^d$ where $d \in \mathbf{N}$ is the coverage’s number of dimensions. Dimensions, called *axes*, are ordered by numbering them from 0 to d-1. Coverage constituents are grouped into four levels:

[4] This feature is being discussed in the WCS group for possible inclusion in WCS 1.2.

- **Level 0: coverage values.** This is “the array data themselves”. In a 2-D image, for example, these are the pixel intensity values. No assumption is made about their storage structure, such as files vs. databases, tiled or one unit, etc.
- **Level 1: technical meta data.** Level 0 data can only be understood and accessed based on their Level 1 metadata. These consist of: coverage name; cell type (based on the range structure concept of WCS); coverage dimensionality; coverage domain, denoted by lower and upper bounds in the image CRS, i.e., the (integer) cell array coordinates; an optional null value; interpolation methods supported when operations involve resampling. There is no further semantics (such as space / time) known on this level; axes are numbered (not named) starting with 0, and cells are addressed using integer coordinates in all axes (using positive or negative axis bounds).

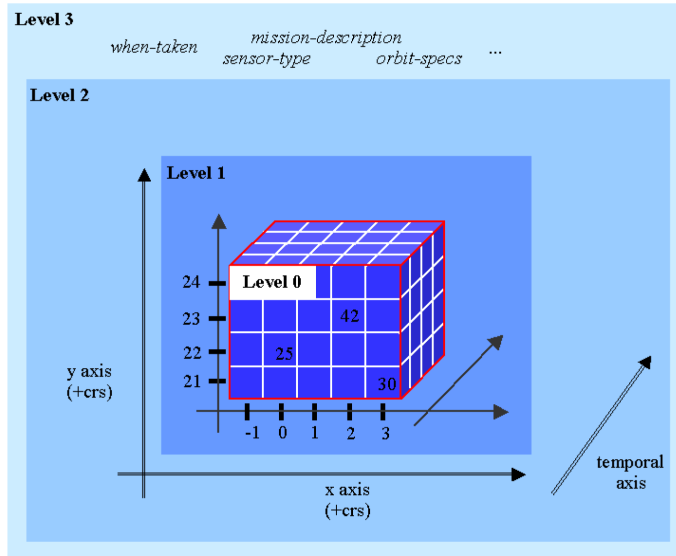

Fig. 1: Coverage conceptual model and terminology.

- **Level 2: spatio-temporal metadata.** This information materializes the space/time semantics that axes can bear. Components are: an optional ground CRS in which spatial coordinates are expressed; named axes, with a spatio-temporal or abstract semantics assigned; the coverage’s domain, represented by lower and upper bound geographic / temporal / abstract coordinates in the resp. dimension.

  To associate semantics with a direction, each axis has an axis type associated; type *x* and *y* designate an x or y spatial semantics to the corresponding axis; type *elevation* denotes a height dimension; type *temporal* denotes a temporal semantics, and *other* means some abstract axis without any spatio-temporal semantics. Coverages can have at most one *x*, *y*, and *t* axis, resp. Operations that refer to geo coordinates – such as reprojection – can only be applied to *x* and *y* axes, if present.
- **Level 3: general metadata.** This encompasses „everything else“, such as application-dependent metadata. Level 3 data are out of the scope of WCPS.

Coverages are described in an object-oriented manner: a coverage object knows several “getter” methods which report about its internal state; the total of all getter methods describes the state information of a coverage. The coverage locations containing values are referred to as *cells*; the (atomic or composite) values associated with a particular cell are called its *cell values*. See Fig. 1 for the terminology.

Levels do not encapsulate (in the sense of hiding) other layers; applications are free to address coverages on any level, for example on Level 2 using geo coordinates or on Level 1 using cell cordinates.

3.2 Operational Model

The WCPS request structure is similar to WCS in that it offers a `GetCapabilities` and a `DescribeCoverage` request. Both have the same request parameters and differ only in the response which, in the case of WCPS, is slightly enhanced to offer the additional information about the server's processing capabilities. The WCPS workhorse is the `ProcessCoverage` request; here the retrieval expression is passed to the server for evaluation. In the sequel we therefore concentrate on the `ProcessCoverage` operation.

Syntactically, WCPS uses an abstract language which subsequently is mapped to request KVP and XML encodings.

A WCPS `ProcessCoverage` request consists of a central request loop where each of a list of coverages in turn is visited to instantiate the associated processing expression if an optionally provided predicate is fulfilled. Coverages must be taken from the list advertised by the service in the `GetCapabilities` response.

The request loop is denoted as

```
for c in ( coverageList )
[ where cond(c) ]
return pExpr(c)
```

Variable *`c`* iterates over the coverages enumerated in *`coverageList`*, considering only those where the predicate *`cond(c)`* is fulfilled. The processing expression `pExpr(c)` consists of either a metadata accessor operation (such as *`tdom(c)`* returning *`c`*'s temporal domain extent), or of an encoding expression

```
encode(e,f)
```

where *`e`* is a coverage-valued expression and *`f`* is the name of a data formats supported.

Example: "*Coverages A, B, and C, TIFF-encoded*":

```
for c in ( A, B, C )
return encode( c, "tiff" )
```

Obviously the data format must be able to hold the result coverage – e.g., a 3-D result cannot be encoded in JPEG, neither can a 7-band multispectral image.

Among the list of coverage operations are induce expressions, subset expressions, scaling expressions, crs transform, coverage expressions, and condense expressions. Among the auxiliary functions is mapping of spatial and temporal coordinates into cell coordinates; the server does not need to disclose the underlying mapping algorithm. Except for the CRS transformation (reprojection) of a coverage, all operations can be described algebraically.

A *subset expression* allows for temporal subsetting along an axis. This can either be an interval cutout (which preserves the dimensionality) or slicing at a given position (which reduces dimensionality by one). Arguments are the coverage, the axis, and the subsetting coordinate(s). Arguments are given on Level 1, i.e., cell coordinates.

Example: "*Time slice 17 from coverage A"* (assuming time axis has number 3; this information is indicated in the `DescribeCoverage` response):

```
for c in ( A )
return encode( sect(c,3,17), "tiff" )
```

If a client wishes to express Level 2 subsetting, i.e., spatio-temporal coordinates, it needs to apply the coordinate mapping function.

Example: "*A section at time 'Thu Nov 24 01:33:27 CET 2005' through coverage A*":

```
for c in ( A )
return encode( sect(c,3,
          ttransform(c, "Thu Nov 24 01:33:27 CET 2005")), "tiff" )
```

This may seem as complex and unwieldy at first sight. However, this allows addressing on both geo and pixel level; further, this language is not intended for humans; rather, some GUI client will offer click-and-point request composition, and then internally the corresponding XML request will be generated and shipped to the server.

An *induce expression* allows to simultaneously apply a cell operation (such as addition of corresponding cells from two coverages, or access to some cell component) to a coverage as a whole. All usual unary and binary operations available in programming languages (cast, record selection, arithmetic, boolean, trigonometric, and exponential functions) can be induced.

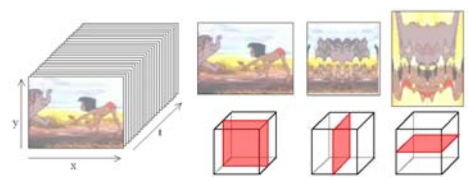

Fig. 2: Slicing illustrated with a movie clip cutout

Example: "*Sum of red and near infrared band from coverage A, as 8-bit integer*":

```
for c in ( A )
return encode( (char) c.red+c.nir, "tiff" )
```

Note that by avoiding an explicit coverage iteration sequence the server is free to transparently optimise storage and access, e.g., by using tiling schemes as in rasdaman [Furtado 1999].

A *scaling expression* performs scaling of a coverage along a specified axis. As this usually involves resampling and interpolation, an interpolation method must be specified in addition.

Example: "*Coverage A, scaled to (a,b) along the time axis*"; this assumes that the server offers cubic interpolation on this coverage:

```
for c in ( A )
return encode( scale(c,t,a,b,"cubic"), "tiff" )
```

The most general operations, finally, are coverage constructor and condenser. A *condenser* corresponds with aggregation in relational languages. Given some condense (i.e., aggregation) operation, a domain to iterate over, and a cell expression, a summary value is computed. Shorthand condensers, along the line of SQL, like `count_cells()`, `add_cells()`, `some_cells()`, `all_cells()`, etc., are available bearing the obvious meaning; see [Baumann 1999, OGC 2006] for more background.

Example: "*The maximum near infrared cell value in Coverages A, B, and C*". Note that no assumption is made about the coverage's dimensionality. As the result is a scalar, no coverage, no encoding is required:

```
for c in ( A, B, C )
return max_cells( c.nir )
```

A *coverage constructor*, finally, allows to build up a coverage with completely new dimension, domain, and cell contents.

Example: "*Histogram for 8-bit greyscale coverages A, B, and C, as comma-separated list*". This creates a 1-D coverage of 256 buckets and fills these using a condenser for counting the values; note that again no assumption is made about the dimensionality of A, B, and C.

```
for c in ( A, B, C )
return encode(
            coverage bucket in [0:255]
            values   count_cells( A = bucket ), "csv" )
```

By combining the coverage constructor and the (general) condense operator, a large class of imaging and signal processing operations including, e.g., filter kernels can be expressed. The limit tentatively has been set with recursive operations. Recursion, such as matrix inversion, is excluded to obtain a "safe" language – i.e., every request is guaranteed to terminate after a finite number of steps.

3.3 Request and Response Encoding

Requests conforming to the abstract WCPS language are shipped to the server in either KVP (key-value pair) or XML encoding, described in XML Schema.

Example: The fragment "*encode(C.red,'tiff')*" would appear in the XML encoding as:

```
<formatEncoding format="tiff" params="">
  <coverageExpr>
    <structSelection>
      <coverageExpr>
        <coverageName>C</coverageName>
      </coverageExpr>
      <selector>red</selector>
    </structSelection>
  </coverageExpr>
</formatEncoding>
```

The encoded result coverage(s) are, at the client's discretion, either shipped back immediately as XML response, or they are stored at the server for some time and only the URL is returned. Scalar results (like condensers and metadata) are returned immediately.

Multiple results (such as lists of coverages, or coverages stored in multiple files) makes use of the multi-file response being part of WCS 1.1.

4. USE CASE SCENARIOS

WCPS has been designed to accommodate observation and simulation data from all earth sciences – among them remote sensing, geography/geodesy, geophysics, geochemistry, oceanography, climate modelling – and beyond. We illustrate the use of WCPS-based services through two use cases: 2-D (*x/y*) satellite maps and 4-D (*x/y/z/t*) climate models.

4.1 Use Case 1: Satellite Map Services

The first use case consists of raster map services for navigation and bulk download.

We assume a coverage representing a large seamless map. Overlapping mosaics are excluded because otherwise the value of a pixel may not be uniquely determined, hence the definition of a WCS coverage is not fulfilled any longer; if overlapping is desired then each contiguous part (such as a single satellite image scene) has to go into a separate coverage. Cells within the bounding box may be undefined, which is indicated by storing a null value there.

A typical workflow for ordering of image products is to first interactively navigate, determining product, bounding box, and maybe overlays, and then order the selection in some (usually high) resolution, CRS, and delivery data format.

Such a workflow is implemented and in commercial use in the rasdaman Web client, rasgeo, since several years. Navigation via WMS allows to compose a layer stack;

In future, WCPS can be used alongside WMS so that only open standards are used for server access; actually, WMS requests can be mapped to WCPS. To give a flavour of such a mapping, we present a (simplified) rasgeo-generated rasdaman query resulting from a WMS GetMap request:

```
select jpeg(      scale(img0[...],[1:246,1:300])
      overlay (  (scale(img1[...],[1:246,1:300])<71.0))* { 51c, 153c, 255c}
      overlay bit(scale(img2[...],[1:246,1:300]),4)    * {191c, 255c, 255c}
      overlay bit(scale(img3[...],[1:246,1:300]),1)    * {  0c, 255c, 255c}
        )
from ...
```

Clearly visible are the overlay stacks; within each stack, subsetting plus result colouring takes place on thematic layers. The induced comparison with 71.0 implements a DEM colouring to highlight flood areas. Evaluation of such requests takes about 0.3 seconds on a 1.7 GHz PC.

After the user has chosen his/her desired map stack, the order component can be used to order this stack in high resolution in one of many image formats, optionally mosaicked. Currently the order component generates a rasdaman query to produce the requested file set.

In a similar manner, the Normalized Vegetation Index (NDVI) can be provided as either a virtual WMS layer or a virtual WCS coverage. For read and near infrared bands *red* and *nir*, the NDVI is defined as *abs(red+nir)/(red–nir)*.

Example: In WCPS language the NDVI translates immediately into the following request:

```
for c in ( A, B, C )
return encode( (char)(100*abs(c.red-c.nir)/(c.red+c.nir)),“tiff” )
```

The result type of the division is `float`; as this doesn’t fit into TIFF, so it is mapped into [0..255] and cast to the 8-bit cell type `char`.

Here the advantage of a language shows up prominently: derived data can be provided ad-hoc, without extra programming – either on server side as aka virtual data sets based on stored requests, or by generating appropriate requests on client side. Obviously for the latter case an easy, user-convenient way to phrase complex requests must be provided (see Section 5.2).

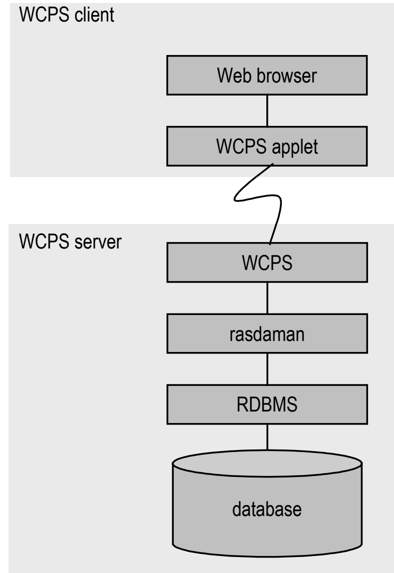

Fig. 4: WCPS reference implementation architecture

4.2 Use Case 2: Climate Model Data

Earth system research increasingly needs and produces a variety of raster data. In the GALEON network[5] the WCS standard is investigated on its suitability for 4D/5D climate model data; basis is the netCDF data model and exchange format which is widely used in the climate modelling community and beyond. As first results, several change requests have been brought into the WCS group, and a WCS application profile for netCDF has been drafted.

Currently the GALEON data we store on our WCS also serve as test vehicle for the WCPS im plementation. A typical request might ask for the average of the absolute of the wind speed over *x* and *y* dimensions over ground, i.e., at height 0.

Example: This translates into the following WCPS request expression:

```
for c in ( ClimateModel )
return avg_cells( sqrt( c.vx*c.vx + c.vy*.vy ) [*:*,*:*,0,*:*] )
```

5. REFERENCE IMPLEMENTATION

The concepts outlined in the previous section have been implemented at IUB to a great extent; when finished this code will serve as reference implementation. It consists of a database-backed WCPS server plus GUI as well as command line client.

5.1 Server

Our WCPS server [Chulkov 2006] maintains any-size, any-dimensional coverages in a relational database (Fig. 4). To this end, coverage objects are partitioned into so-called tiles. The rasdaman [Ritsch 1998] middleware accomplishes automatic, transparent tiling and offers a raster query language, rasql, which extends SQL92 with array expressions. Based on this, the WCPS service is implemented as a Java servlet which transforms WCPS requests into rasql queries. The rasdaman server actually is the workhorse that computes and packages coverages into the data exchange format requested. The WCPS servlet pipes the rasdaman results through to the client.

5 www.ogcnetwork.net/?q=galeon

Grounding coverage services on databases is a rare exception supported by only a few systems [Ritsch 1998, ESRI 2005, Oracle 2005]; only rasdaman has an algebraically founded query language, optimization, and storage management. Databases, however, offer a large toolkit of tools as well as tuning and optimization facilities which have proven advantageous for large-scale data management.

To support this we briefly discuss one element, storage management. Climate data tend to be very large. A rather low-resolution model, ECHAM-T42, developed by the European Centre for Medium-Range Weather Forecast[6] (ECMWF) and the World Data Centre for Climate[7] (WDCC), has the following characteristics: spanning the whole earth surface, it has a spatial resolution of 64 x 128 cells and 17 elevation levels. With a temporal resolution of 25 min and a simulation time of about 200 years, the result data volume is about 2.5 TB per variable. Each variable represents a physical parameter, such as temperature, humidity, *x/y* wind speed.

Such data are produced by a supercomputer in time slices (usually several slices are packed into one file), yielding a data organisation as shown in Fig. 5. On the other hand, a large class of operations performs spatial rather than temporal selection, resulting in a slicing orthogonal to the storage. Obviously reading data in the unit of such input fields results in a tremendous amount of unnecessarily read data. The tiling concept of rasdaman allows to optimize storage for specific retrieval situations, such as multi-dimensional slicing. From the range of options available, partitioning into cube tiles of only a few MB size is particularly suitable in the case on hand where tile size, among other criteria, is determined by the server hardware characteristics such as disk transfer rate vs. CPU speed.

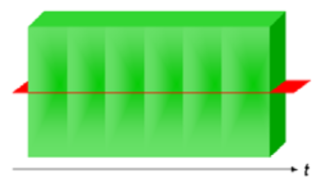

Fig. 5: time-oriented data generation (vertical slices) vs. spatial user access pattern (horizontal slicing)

5.2 Clients

First studies towards domain-specific clients on top of a general-purpose raster query language have been conducted in the ESTEDI project[8]. Currently a visual WCPS Java client [Delchev 2006] is under development which allows to graphically compose complex requests by drawing and connecting boxes on screen. Built-in semantics checks on the operation input and output types ensure that operations can only be combined in a meaningful way.

6. CONCLUSIONS AND OUTLOOK

Currently signs of a convergence between the various earth sciences disciplines are observable. Naturally, integration occurs at the level of measured and observed data, where raster / coverage data comprise an information category of prime importance. Hence, the outreach of WCPS is rather wide, encompassing all of the earth sciences and beyond.

---

[6] www.ecmwf.org
[7] www.mad.zmaw.de/wdc-for-climate/
[8] www.estedi.org

The Web Coverage Processing Service (WCPS) Implementation Specification has been designed to support non-trivial navigation and analysis on large-scale, multi-dimensional sensor and image data. WPCS is not aiming to substitute image processing; rather, it defines a data service whose main task is fast and flexible extraction from extremely large data assets. Specialized tools may well be front-ends operating on the result data, eg, image analysis, visualization; actually in the ESTEDI project the Khoros and IDL On The Net packages have been plugged successfully onto rasdaman.

Currently (July 2006) WCPS is OGC Best Practices Paper, an editorially revised version is being put on the OGC Portal for public comment. It is expected that by Fall 2006 WCPS will receive the status of a Draft Implementation Specification, the last step before becoming accepted standard.

The reference implementation conducted at IUB strengthens feasibility of the concepts. The technology basis, relational databases combined with the rasdaman raster server middleware, has matured over many years and can be considered stable. Rasdaman is in operational use, partially for more than five years, in mapping agencies, mining industry, and research. Feasibility has been proven in research projects in several geo, life science, and Grid projects.

Now all affected communities are invited to comment on the WCPS draft and point out their specific requirements. Only if as many communities as possible, spanning as diverse domains as possible, take a critical review WCPS can achieve its goal of a unified, cross-domain, interoperable standard for advanced coverage services.

ACKNOWLEDGEMENTS

The author is indebted to John Evans (NASA; WCS co-editor) and Arliss Whiteside (BEA Systems; WCS co-editor); Steven Keens (PCI Geomatics); Luc Donea (Ionic); Peter Vretanos (Cubewerx); and all the other WCS Revision Working Group members; particular thanks to Sean Forde (Lizardtech; Coverages WG speaker).

Further, many thanks to Georgi Chulkov and Ivan Delchev for their great work implementing WCS/WCPS server and clients, and for patiently exploring all the conceptual pitfalls.

REFERENCES

[Baumann 1999] P. Baumann: *A Database Array Algebra for Spatio-Temporal Data and Beyond.* Proc. Next Generation IT and Systems (NGITS), Zikhron Yaakov, Israel, 1999, pp. 76 – 93

[Chulkov 2006] G. Chulkov: *Architecture and Implementation of a Web Coverage Processing Service Using a Database Back-End*. IUB Bachelor's Thesis, May 2006

[Delchev 2006] I. Delchev: *Graphical Client for a Multidimensional Geo Raster Service*. IUB seminar report, May 2006

[ESRI 2005] n.n.: *Raster Data in ArcSDE 9.1*. ESRI White Paper, September2005

[Furtado 1999] P. Furtado, P. Baumann: *Storage of Multidimensional Arrays Based on Arbitrary Tiling.* Proc. ICDE'99, Sidney - Australia 1999

[ISO 2004] n.n.: *FDIS 19123 Geographic information - Schema for coverage geometry and functions*. ISO/TC 211, document 1740, October 2004

[OGC 2005] P. Schut, A. Whiteside: Web Processing Service. OGC Discussion Paper, artifact 05-007r4, September 2005, available from OGC Portal (www.opengis.org)

[OGC 2006] P. Baumann: Web Coverage Processing Service. OGC Best Practices Paper, artifact 06-102, August 2006, available from OGC Portal (www.opengis.org)

[Oracle 2005] n.n.: *Oracle Database 10g: Managing Spatial Raster Data using Geo-Raster*. Oracle Technical Whitepaper, May 2005

[Ritsch 1998] R. Ritsch, N. Widmann: *The Multidimensional Database System Rasdaman.* Proc. ACM SIGMOD'98, Seattle, USA 1998, pp. 575 - 577

BIOGRAPHICAL NOTES

Peter Baumann is Professor of Computer Science at International Unviersity Bremen (IUB). He holds a PhD from Tech. Univ. Darmstadt. Before joining IUB in 2004, he was with Softlab, Munich, then Assistant Head of the FORWISS Knowledge Base Research Group, Munich. For commercialization of his research results he has founded spin-off companies.

Peter Baumann's research interest focuses on large-scale multidimensional raster databases and their application in geo, life sciences, Grid, e-learning, etc. He has authored and co-authored 60+ book chapters and papers in the areas of raster databases, and holds international patents in the field. He has received several national and international innovation awards for his work.

He is member of OGC, the Commission for the Management & Application of Geoscience Information (CGI), and a founding member of CODATA Germany.

CONTACTS

Prof. Dr. Peter BAUMANN

International University Bremen	rasdaman GmbH
Campus Ring 12	Adam-Berg-Str. 172a
D-28759 Bremen	D-85731 München
Germany	Germany
Tel. +49-(0)421-200-3178, Fax -493178	Tel. +49-(0)89-67000-146, Fax -147
Email: p.baumann@iu-bremen.de	Email: baumann@rasdaman.com
Web: www.faculty.iu-bremen.de/pbaumann	Web: www.rasdaman.com