ISPRS Int. J. Geo-Inf. **2015**, *4*, 1076-1096; doi:10.3390/ijgi4031076

OPEN ACCESS

ISPRS International Journal of

Geo-Information

ISSN 2220-9964

www.mdpi.com/journal/ijgi/

Article

Conception and Implementation of an OGC-Compliant Sensor Observation Service for a Standardized Access to Raster Data

Juergen Sorg * and Ralf Kunkel

Research Center Juelich, Leo Brandt Strasse, Juelich 52425, Germany; E-Mail: r.kunkel@fz-juelich.de

***** Author to whom correspondence should be addressed; E-Mail: j.sorg@fz-juelich.de; Tel.: +49-2461-61-5535.

Academic Editors: Thierry Badard, Franz-Josef Behr and Wolfgang Kainz

Received: 30 January 2015 / Accepted: 24 June 2015 / Published: 6 July 2015

Abstract: The target of the Open Geospatial Consortium (OGC) is interoperability of geographic information, which means creating opportunities to access geodata in a consistent, standardized way. In the domain of sensor data, the target will be picked up within the OGC Sensor Web Enablement Initiative and especially reached through the Sensor Observation Service (SOS) standard. This one defines a service for a standardized access to time series data and is usually used for *in situ* sensors (like discharge gauges and climate stations). Although the standard considers raster data, no implementation of the standard for raster data exists presently. In this paper an OGC-compliant Sensor Observation Service for a standardized access to raster data is described. A data model was developed that enables effective storage of the raster data with the corresponding metadata in a database, reading this data in an efficient way, and encoding it with result formats that the SOS-standard provides.

Keywords: TERENO; sensor observation service (SOS); SWE; OGC; raster data

---

1. Introduction

In situ measurement stations for observing physical phenomena (e.g., temperature, soil moisture, *etc.*) are always related to a single geographic point. The result of these measurements quantifies the observed phenomenon for this particular point as a time series of scalar values. In contrast, remote sensing stations deliver area differentiated data related to a certain geographic area, and quantify the

observed phenomenon as a time series of raster data. The Open Geospatial Consortium (OGC) provides two standards for the access to raster time series data: WCS and SOS.

The common approach for a standardized access to raster data is the OGC Web Coverage Service (WCS) standard. In general, the WCS specification considers no temporal component; however, since version 2.0 the WCS standard is able to handle multi-dimensional arrays, which can be used to include a temporal dimension. The most popular and comprehensive implementation of the WCS standard is the rasdaman project (http://www.rasdaman.org). Beside the mandatory implementation of the core WCS 2.0 standard it also implements the WCS-T extension for a standardized insert of data, the WCS-EO extension for access to earth observations, and the OGC WMS standard for access to images and is therefore state of the art for managing raster data. The WCS-EO, for instance, uses a two-dimensional grid (EO-coverage), a WGS84 bounding box, and an additional metadata set (EO-Metadata). In this additional metadata the temporal relation of each dataset is specified. A new operation (DescribeEOCoverageSet) has been defined, which returns a description of each dataset including its temporal relation. The dataset itself can be selected by temporal filters and requested by the GetCoverage operation using the unique identifiers of the datasets.

The OGC sensor observation service (SOS) standard comprises methods for standardized access to all kinds of time series data with spatial relation to the earth. In contrast to the WCS(-EO), which has no intrinsic consideration of the temporal reference, the SOS inherits the temporal relation of each dataset, but more over the solid integration in the Sensor Web Enablement (SWE) framework of the OGC, which enables "*the discovery of sensors and corresponding observations, exchange, and processing of sensor observations, as well as the tasking of sensors and sensor systems*" [1]. In particular, the OGC SensorML standard provides capabilities to describe the installed equipment and the entire data collection and preparation process, which is not available for the WCS(-EO). Furthermore, a SOS supports the application of thematic filters to extract thematic attributes of data. A number of implementations of the OGC-SOS standard exist providing *in situ* sensor time-series data. However, no implementations exist to provide time series raster data.

In this paper we describe the conception and implementation of an OGC-compliant SOS for standardized access to raster time series data, which allows us to select raster datasets using temporal, spatial, and thematic filters and to deliver them in a standardized way. This work originated within the TERENO (Terrestrial Environment Observatories) initiative, an interdisciplinary long-term research project that gathers the long-term ecological, social, and economic results of global change on a regional scale [2,3]. Four terrestrial observatories are established, each providing time series data on soil moisture, soil temperature, and a water gauge as well as energy and fluid flux from a number of sensor networks. Four weather radar and rain scanner devices to measure precipitation rates have been installed. The developed SOS is used to give standardized access to data and metadata measured by these devices.

2. OGC Sensor Observation Service

The Open Geospatial Consortium (OGC) is a confederation of leading GIS manufacturers, data producers, authorities, organizations, research facilities, and universities. It was established in 1994 and develops standardized interfaces for interoperable access to geographic data. The interoperability

of sensor and sensor networks is covered by OGC's Web Enablement Initiative [4], which defines several of these in terms of their aligned standards and interfaces with each other:

- The Sensor Observation Service (SOS) standard defines a standardized web service to search, filter, and retrieve observational data and sensor information [5,6].
- Observation & Measurement (O&M) is a standard to describe, exchange, and encode geodata [7–10].
- The Sensor Model Language (SensorML) provides an information model to describe sensors—the observed properties as well as the data collection and preparation process [11].
- Sensor Alert Service (SAS) gives users the possibility to receive alert messages from sensors over a standardized interface [12]. Presently there is a discussion about extending the SAS to a Sensor Event Service (SES), which is able to handle all kinds of event messages [13].

In addition, SWE also defines a Sensor Planning Service (SPS) and a Web Notification Service (WNS), which are not within the scope of this paper.

The SOS standard comprises 11 operations, but only three operations—*GetCapabilities*, *DescribeSensor*, and *GetObservation*—are mandatory. The *GetCapabilities* Operation yields general information about the service and all information necessary to call the supported operations. The *DescribeSensor* Operation yields a description of a sensor encoded by the SensorML language [11] and contains amongst others identifiers for the observed properties, coordinates of the station, and a time domain, for which data is available. Finally, data can be requested by means of the *GetObservation* operation. The XML fragment in Figure 1 shows an example of a *GetObservation* request with all available parameters [5].

```
<GetObservation>
  <offering>Reflectivity</offering>
  <eventTime>
    <o:TM_During>
      <o:PropertyName>om:samplingTime</o:PropertyName>
      <gml:TimePeriod>
        <gml:beginPosition>2012-01-07T10:28:54</gml:beginPosition>
        <gml:endPosition>2012-01-07T12:28:54</gml:endPosition>
      </gml:TimePeriod>
    </o:TM_During>
  </eventTime>
  <procedure>WeatherradarSophienhoehe</procedure>
  <observedProperty>Reflectivity</observedProperty>
  <featureOfInterest>
    <o:Intersects>
      <o:PropertyName>urn:ogc:data:location</o:PropertyName>
      <gml:Point srsName="EPSG::31466">
        <gml:pos>5657550.5 2606247.0</gml:pos>
      </gml:Point>
    </o:Intersects>
  </featureOfInterest>
  <om:result>
    <o:PropertyIsGreaterThan>
      <o:PropertyName>Reflectivity</o:PropertyName>
      <o:Literal>36</o:Literal>
    </o:PropertyIsGreaterThan>
  <om:result>
  <responseFormat>text/xml;subtype="OM/1.0.0"</responseFormat>
  <resultModel>om:Time seriesObservation</resultModel>
  <responseMode>Inline</responseMode>
</GetObservation>
```

Figure 1. An SOS GetObservation request with all available elements.

The unique and mandatory *offering* identifies a free selectable, use case dependent logical combination of sensors (e.g., by observed properties) [5]. The parameters for temporal, spatial, and thematic filters (eventTime, lines 3–11, featureOfInterest, lines 14–21 und result, lines 22–27), the unique identifiers of the station (procedure, line 12), and the observed properties (observedProperty, line 13) are optional.

The SOS standard supports a data query according to thematic and spatial filters. A thematic filter selects objects according to values of a variable. In lines 3–11 of Figure 1, a temporal filter is specified, which selects the data from a given sampling time period. In lines 22–27 of Figure 1, a thematic filter is specified that selects all meshes of a raster dataset; the attribute "Reflectivity" has a value greater than 36. Figure 2 shows some examples of spatial filters that are supported by the SOS standard. On the left side a line segment filter is depicted, which extracts all values located on a street course. In the center a rectangle filter and on the right hand side a not orthogonal polygonal filter is depicted. Both filters are selecting objects located within the filter faces.

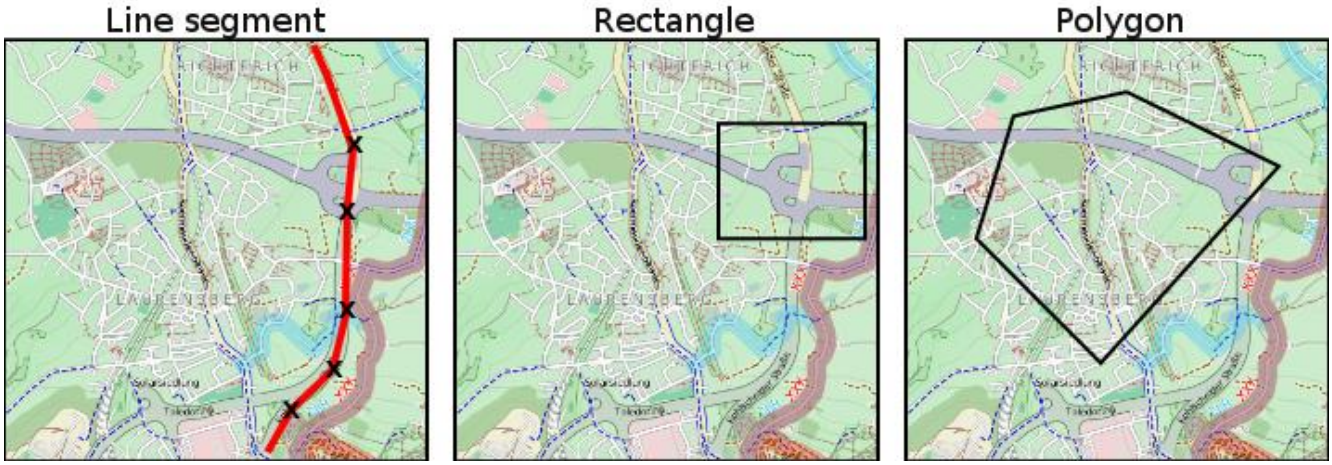


Figure 2. Spatial filters, (**left**) a line segment filter, (**center**) a rectangle filter, and (**right**) an orthogonal polygonal filter).

The SOS processes requests using the specified filter conditions and returns the results as O&M documents [7–10]. Thereby the O&M standard facilitates encoding vector data as well as raster data as XML documents [10]. Raster data are GML-encoded according to a conceptual model for spatial coverages defined in ISO19123, whereas for vector data a number of different encodings are used. Within a *GetObservation* request the desired result model is specified by the Parameter *ResultModel* (line 46 in Figure 1) [5]. Therefore, it is possible to request raster as well as vector data from an SOS in a standardized way.

3. Implementation

A number of implementations of the SOS standard in different programming languages exist [14]. We selected the SOS implementation of the 52 ℃N company as a basis for the extension to a Raster-SOS, since this implementation is commonly used, open source and easily modifiable. The extension comprised the following steps:

- A data model to store the raster data and its describing metadata in an efficient way has been developed, based on the 52 ℕ SOS data model (see Figure 3).
- Efficient algorithms to apply filters have been implemented.
- Several O&M models to return the raster data in a standardized and flexible manner have been realized.

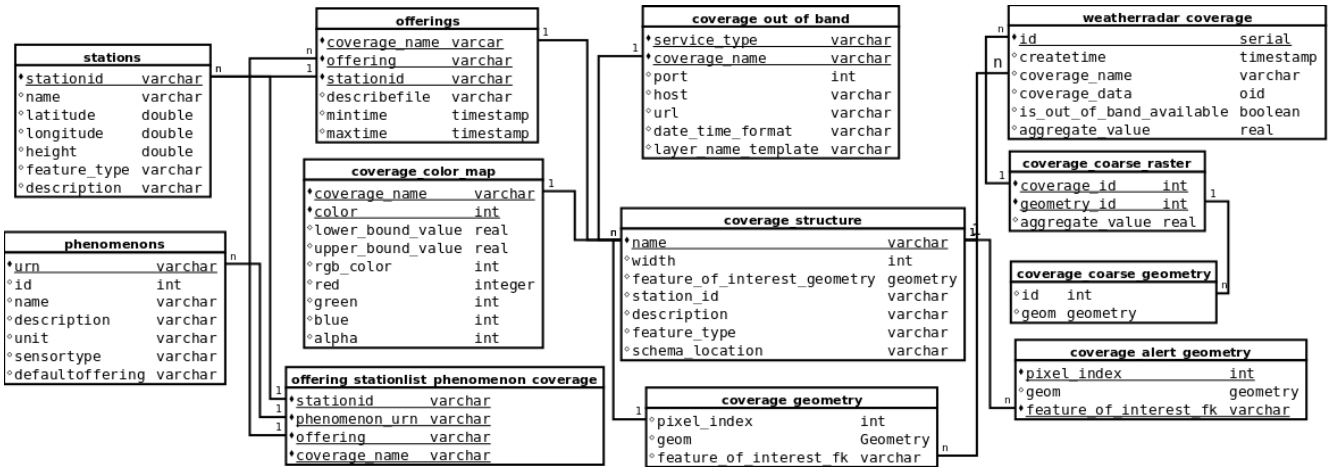


Figure 3. Data model.

Figure 3 shows the developed data model. The metadata describing the time series data is stored in the normalized tables coverage_structure, phenomenons, stations, offerings, and coverage_-out_of_band. Moreover, the coarser grid is stored normalized in the table coverage_coarse_raster and coverage_coarse_geometry. The raster data itself is stored as a BLOB field in the table weatherradar_coverage with a foreign key link to its time series in table coverage_structure.

3.1. Data Model

The raster data are stored in a relational database management system (RDBMS). In our implementation PostgreSQL [15] with PostGIS [16] extension for spatial data was used, however, any other RDBMS able to handle geodata (e.g., Oracle with OracleSpatial extension) may be used instead. We want to stress the fact that database-intrinsic raster processing, like is provided by the PostGIS version (2.0), cannot be used here. Since PostGIS, for instance, uses one table for each raster dataset [17], a huge number of data tables may be created. Within the TERENO project period we expect to create up to 1.5 Mio. weather radar datasets, which makes the usage of PostGIS raster data tables not practicable.

Therefore, a data model was developed that holds all raster data in one single data table. This can be accomplished in two different ways: row based or as a binary large object (BLOB). A BLOB is a database type for large, unspecified binary objects, using a minimized amount of storage. On the other hand, storage as a BLOB is inconvenient for thematic filtering of data, because the entire binary object must be read and the filter condition must be proofed for each raster mesh. Keeping each raster mesh in a row of a data table, on the other hand, has the advantage of an effective search. The disadvantage

is the large amount of inserts and rows that accrue for each raster dataset (e.g., for a relatively small raster with 800 × 800 pixels 640.000 inserts and rows must be handled).

In order to provide a compromise between effective storage and effective searching, we use the BLOB storage of the data, but an additionally generated coarser grid to achieve efficient thematic search capabilities. The coarser grid resolution is freely selectable and keeps the maximum and minimum of the underlying original raster meshes for each of its raster meshes. The resolution of the grid is selected in such a way that on the one hand inserts can be done in an adequate time frame and on the other hand not too many original pixels have to be read and checked. The biggest advantage is, however, that database indices can be used to have much more efficient access to data than with the sequential method [18].

3.2. Filters

The existing filters of the 52 °North implementation are coupled to the intrinsic database functions, which support only vector-based requests. However, here we need thematic and spatial filter operations that can be applied to raster data, too. Thematic filters select raster cells by means of values of a variable, for which the aforementioned coarse grid was used. This has the advantage, as only relevant pieces of the original dataset have to be read. Because the selected pixel of the coarse grid always represents an orthogonal polygon, we implemented an efficient sweep algorithm [19], which scans the polygon in a vertical direction from top to bottom and holds those line segments in a temporary cache, which marks domains that can be read in blocks from the original raster dataset (see Figure 4). The filter condition needs to be checked only for the cached segments.

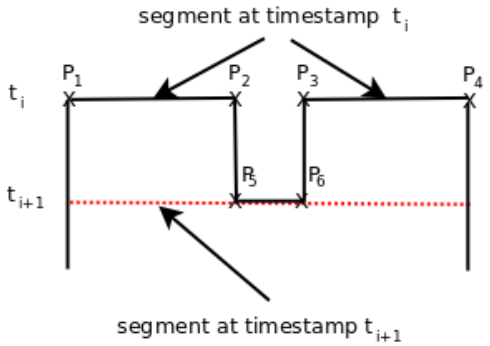


Figure 4. Sweep algorithm to scan orthogonal polygons.

The sweep algorithm can be used for all spatial filters that deal with the original raster dataset directly (not over the coarse grid), *i.e.*, if the spatial filter is orthogonal (see Figure 2 right). If the spatial filter is not orthogonal, the sweep algorithm cannot be used. Instead, a vectorized representation of the raster is used, which holds the geometry and the cell index of the original raster dataset in each mesh. With this vectorized raster the PostGIS intrinsic functions like intersects, contains, *etc.* can be used [16]. In this way, our implementation supports any spatial filter geometry, which can be encoded with GML within the *GetObservation* request and transformed to the WKT (Well Known Text) format [20].

Table 1. Filter combinations and their results.

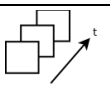
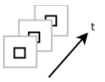
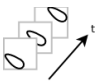
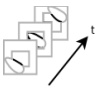
Temporal	Spatial	Thematic	Result
x	--	--	
x	x	--	
x	--	x	
x	x	x	

Table 1 shows all combinations of temporal, spatial, and thematic filters and the results after their application. If only a temporal filter is applied (row 1 of Table 1) then the matched results are entire raster datasets. Row two of Table 1 is an example of a combination of a temporal and a spatial polygonal filter. All pixels within the polygonal clipping area of each dataset, which has been selected by the temporal filter, are added to the result set. As row 3 of Table 1 shows, a combination of a temporal and a thematic filter (row 3) facilitates tracking a moving object. Within each temporal selected dataset, the object is extracted by filtering its thematic attribute. For example, in the case of our weather radar the object can be a precipitation cloud, which can be extracted by applying a thematic filter with a precipitation or reflectivity threshold of a certain value (e.g., 48 mm/h, which meets 50 dbZ—compare Section 0). The last row shows a combination of all three filter types. After a temporal selection of datasets, pixels are selected by a polygonal filter but will only be added to the result set if the thematic filter criterion is matched. A possible use case is finding weather events occurring only in a certain region.

3.3. Result Models

The SOS standard allows packing data as responses to a GetObservation request as INLINE, OUT-OF-BAND, and ATTACHED. We use all three possibilities to handle different use cases. The O&M models used for the direct (INLINE) encoding of unfiltered and thematically and spatially filtered raster data are the *DiscreteCoverageObservation*, the *TimeSeriesObservation*, and a generic *O&M* model [10,21]. Entire or parts of raster datasets are provided as georeferenced images (not the data values themselves) over an external (OUT-Of-BAND) OGC compliant Web Map Service. ATTACHED encodes entire or parts of raster datasets in the binary NetCDF format to provide access to the data values itself in a scientific well known format.

For the data transfer using the INLINE method (see Figure 1), two different models, both included in the ISO 19123 standard [22], have been implemented. The *DiscreteCoverageObservation* O&M model can be used for the encoding of spatial and thematic filtered data, while the *TimeSeriesObservation* O&M model is suitable for the encoding of a time series of a selected raster mesh. In this case the requests use a point filter. Figure 5 shows the geometry-value pair of a raster mesh encoded with the *DiscreteCoverageObservation* model.

```
<ns:element>
  <ns:CV_GeometryValuePair>
    <ns:geometry>
      <ns:CV DomainObject>
        <ns:spatialElement>
          <gml:Polygon srsName="EPSG::31466"
              xsi:type="gml:PolygonType">
            <gml:exterior>
              <gml:LinearRing
              xsi:type="gml:LinearRingType">
                <gml:coordinates>
            5640414.72905 2468620.92429,
            5640664.72905 2468620.92429,
            5640664.72905 2468870.92429,
            5640414.72905 2468870.92429,
            5640414.72905 2468620.92429
                </gml:coordinates>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </ns:spatialElement>
      </ns:CV DomainObject>
    </ns:geometry>
    <ns:value uom="dbz" xsi:type="gml:MeasureType">
      8.944881889763703
    </ns:value>
  </ns:CV GeometryValuePair>
</ns:element>
```

Figure 5. Geometry value pair of a raster mesh encode with the *DiscreteCoverageObservation*-model.

The geometry is encoded as a polygon in the Geography Markup Language (GML), while the thematic value is encoded as a decimal value with its unit of measure (UOM). In the *TimeSeriesObservation* model the dataset is encoded similarly with the difference that the geometry is replaced by a timestamp. Because both models are XML based and very character intensive, a third generic O&M model [10] was used to encode data with comma-separated values. For each raster cell the index or geometry and the data value are given. An additional reduction of the data volume is accomplished by a gzip compression (RFC 1952 Gzip file format) of the response document. For this purpose the SOS standard allows us to use the mime type “application/zip” for the *ResponseFormat* (see Figure 1 line 28) parameter.

For an indirect data transfer the SOS standard provides the possibility to reply to a *GetObservation* request with a reference to an external source [10]. We use this option to provide the raster as a georeferenced image from an OGC compliant Web Map Service (WMS) [23]. References are encoded as a WMS *GetMap* URL and included in the response document, which is encoded as the generic O&M model. Within the *GetObservation* request the response mode is given by the value OUT-OF-BAND for the parameter *ResponseMode* (see Figure 1 line 30). Figure 6 shows a reference to a raster image provided by a WMS, which is included into a generic O&M model.

```
<om:member>
  <om:Observation>
    <om:samplingTime>
      <gml:TimeInstant
         xsi:type="gml:TimeInstantType">
        <gml:timePosition>
          2012-10-10T06:00:00.000+02:00
        </gml:timePosition>
      </gml:TimeInstant>
    </om:samplingTime>
    <om:procedure xlink:href="WRradar"/>
    <om:observedProperty
         xlink:href="Reflectivity"/>
    <om:featureOfInterest>
      <gml:GridCoverage gml:id="WRadar">
        <gml:name>Weatherradar</gml:name>
          <gml:boundedBy>
            <gml:Envelope>
              <gml:lowerCorner
              srsName="EPSG::31466">
               5543664.729 2432120.924
              </gml:lowerCorner>
              <gml:upperCorner
              srsName="EPSG::31466">
               5743664.729 2632120.924
              </gml:upperCorner>
            </gml:Envelope>
          </gml:boundedBy>
          <gml:gridDomain/>
          <gml:rangeSet>
            <gml:ValueArray/>
          </gml:rangeSet>
        </gml:GridCoverage>
      </om:featureOfInterest>
      <om:result
        xsi:type="gml:ReferenceType"
        xlink:href=
"http://www.lomepa.ru:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&
layers=tereno:2012-10-10_06-00-
00&styles=&bbox=2463655,5436699,2663655,5636699&width=800&
height=800&srs=EPSG:31466&format=image/png"/>
    </om:Observation>
  </om:member>
```

Figure 6. Reference to a raster image provided by a WMS and encoded with a generic O&M model.

Because a WMS supports only rectangular spatial filters, the given filter conditions within the *GetObservation* request are transformed to the enclosed rectangle and forwarded to the WMS. If a thematic filter is specified, the enclosed rectangle of all determined raster meshes is calculated. We implemented this kind of data provision by a self-developed extension [24] of the open source, WMS compliant GeoServer project (http://geoserver.org). Our GeoServer plugin creates *ad hoc* the requested georeferenced image by retrieving the necessary data from the SOS data model. The plugin implements the mandatory methods *GetCapabilities* and *GetMap* compliant with the WMS version 1.3 standard to provide a service description as well as access to the data.

The SOS standard allows us to deliver data in an arbitrary format as an additional attachment of a *GetObservation* response [5]. In this case the attachment needs to be referenced by the response O&M document. To get requested data as a binary attachment the value of the response mode parameter of a *GetObservation* request must be “ATTACHED” (see Figure 1 line 30). Thereby, the O&M document and also the data will be transferred within a multipart HTTP body (RFC 2046 MIME). All parts are separated by a unique boundary token, which is defined in the HTTP Content-Type header.

We encode the raster data in the NetCDF file [25] format and transfer it as an attachment. NetCDF is a self-descriptive binary matrix oriented file format for large datasets. An API for reading and writing data is provided in almost all common programming languages, like JAVA, C++, or Python. The data is organized within matrices (variables), each of them described by a metadata attribute.

In spite of the inherent metadata attributes of the NetCDF format, interoperability is not directly given, because the structure and content of metadata elements must follow formal conventions to accomplish interoperability. For this purpose several conventions to organize data as well as mandatory metadata attributes with well-defined vocabularies have been defined. We use the CF-Convention [26], which is particularly appropriate for geospatial data. Therein a raster time series is defined by two one-dimensional and two two-dimensional arrays.

```
dimensions:
     X_DIM = 800 ;
     Y_DIM = 800 ;
     BOUNDS = 2 ;
     timeDim = UNLIMITED ; // (277 currently)
variables:
     int EPSG\:31466 ;
            EPSG\:31466:grid_mapping_name = "transverse_mercator" ;
            EPSG\:31466:scale_factor_at_central_meridian = 1 ;
            EPSG\:31466:longitude_of_central_meridian = 6 ;
            EPSG\:31466:latitude_of_projection_origin = 0 ;
            EPSG\:31466:false_easting = 2500000 ;
            EPSG\:31466:false_northing = 0 ;
     double X(X_DIM) ;
            X:standard_name = "projection_x_coordinate" ;
            X:long_name = "x distance on the projection plane from the origin" ;
            X:units = "m" ;
            X:bounds = "X_BOUNDS" ;
     double Y(Y_DIM) ;
            Y:standard_name = "projection_y_coordinate" ;
            Y:long_name = "y distance on the projection plane from the origin" ;
            Y:units = "m" ;
            Y:bounds = "Y_BOUNDS" ;
     double X_BOUNDS(X_DIM, BOUNDS) ;
     double Y_BOUNDS(Y_DIM, BOUNDS) ;
     double time(timeDim) ;
            time:calendar = "noleap" ;
            time:axis = "T" ;
            time:units = "days since 0000-1-1" ;
            time:long_name = "time" ;
     double data(timeDim, X_DIM, Y_DIM) ;
```

Figure 7. NetCDF file formatted in CDL.

Figure 7 shows the header of a raster record encoded in NetCDF and represented in human readable CDL format. The x-coordinate (latitude) of the center of each raster mesh is stored in the x-direction and y-coordinates (longitude) in the y-direction within the two one-dimensional arrays. Associated mesh boundaries to each x as well y coordinates are stored within the two two-dimensional arrays. The x-bounds array holds left as well as right and y-bounds top as well as bottom boundaries.

In our implementation we create the basis raster as a CF-compliant NetCDF file on startup of the service for each managed raster time series. To create the requested NetCDF files in an efficient way, a NCML file is generated additionally, which includes the complete structure, metadata, and, with a reference to the aforementioned NetCDF basis raster file, the data of the basis raster. NCML is an XML-based descriptive language designed exactly for this purpose [27]. To create an SOS-compliant response we have to reference each NetCDF attachment in an O&M document. In our implementation

the reference is encoded by a CID-URL, which is actually defined to reference attachments in a mime email [28]. Figure 8 shows an example of such an O&M document. You can see that each observation section describes its referenced data with some useful metadata elements and references the attachment via the CID-URL. Observation data is grouped by stations (procedures) and parameters (phenomena), which mean that each NetCDF attachment contains one raster time series.

```
<om:Observation gml:id="ot_527">
   <om:samplingTime>
      <gml:TimeInstant xsi:type="gml:TimeInstantType">
         <gml:timePosition>2011-10-06T08:50:00.000+02:00</gml:timePosition>
      </gml:TimeInstant>
   </om:samplingTime>
   <om:procedure xlink:href="WeatherradarSophienhoehe"/>
   <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:Reflectivity"/>
   <om:featureOfInterest>
      <gml:GridCoverage gml:id="ot_527.WeatherradarSophienhoehe">
         <gml:description>NOT_SET</gml:description>
         <gml:name>WeatherradarSophienhoehe</gml:name>
         <gml:boundedBy>…</gml:boundedBy>
         <gml:gridDomain>…</gml:gridDomain>
      </gml:GridCoverage>
   </om:featureOfInterest>
   <om:result xlink:href="cid:527@pro.lmpr.org"/>
</om:Observation>
```

Figure 8. O&M response XML fragment where the data is linked by a CID-URL.

4. Applications

In this section we consider two applications dealing with data retrieved from the described SOS for raster data. This SOS is used to give standardized access to the weather radar data of the Jülicher Weatherradar. The weather radar is located at a 30 m high tower on the 270 m high hill Sophienhoehe near Juelich. The radar transmits microwaves, which are reflected by precipitation clouds. From the reflection intensity precipitation rates can be calculated e.g., by using the Marshall–Palmer equation [29]. Within a scope of 100 km the radar observes any kind of precipitation (rain, hail, snow, *etc.*). The collected data of the radar is a time series with a temporal resolution of five minutes and a spatial resolution of 250 m within a 31,416 km^2 area. Each record of the time series is a PNG image with a resolution of 800 × 800 pixels, which provides in each pixel a homogenous reflectivity value for the associated 62,500 m^2 sized area.

The first application visualizes the weather radar data as an animation on a web page for a quick quality check by a user. The second is a system of software components to alert registered users about heavy precipitation events, which also retrieves weather radar data from the SOS in a standardized way. Before these applications are described, the configuration of and the data import to the raster SOS are considered.

4.1. Data Import

In the first step, each pixel of the basis raster is vectorized to enable spatial filters application. For the weather radar data considered here, the raster is nonrecurring and segmented into 640,000 polygons, which are inserted into a database table (see weatherradar_geometries in Figure 3) together with their pixel indices. A coarse grid must be created to apply a high/low pass filter in an efficient

manner. Here, the coarse grid with an extension of the original raster and a resolution of 20 km is used (see coverage_coarse_geometry in Figure 3). A color table provides a mapping between reflectivity and color values. Moreover, metadata like extension and resolution of the raster, unit of reflectivity (dbZ decibel Z, where Z stands for reflectivity), a name, and a description of the radar device as well as the geographic coordinates are bundled together into a SensorML-file. Last but not least, parameters to provide georeferenced images within a WMS are kept in a database table (see coverage_out_of_band in Figure 3).

Importing of the png images is implemented by a JAVA program, which first extracts the timestamp from the filename, temporally stores gray-scale values of each pixel in an array, and inserts this array as a binary large object (BLOB) in the database by means of the PostgreSQL API for large objects [30]. While importing, the values of the coarse grid are calculated and inserted in the coarse grid table (see Figure 3 coverage_coarse_raster).

4.2. Visualization of Weather Radar Data

For an animated visualization of the weather radar data, a web application has been implemented that uses the OUT-OF-BAND method (see Section 3.3, Result Models). A link to a WMS layer is provided by the data response, which delivers weather radar images for a given time interval. The layer data are presented as layers upon a base layer provided by the OpenStreetMap (OSM) (http://www.openstreetmap.org) web service. The application is developed with the Google Web Toolkit [31] and an OpenLayers wrapper for GWT (https://bitbucket.org/gwtopenlayers/gwt-openlayers) and provides the graphical user interface depicted in Figure 9. The user interfaces provide a visualization of the localization of the weather data. An overlaid circle represents the boundaries of the scope of the radar. In Figure 9 a colorized weather front tracks to the domain, with different precipitation levels as specified in the legend on the bottom. On the left part of the interface several buttons to control the animation and to visualize the time series of a selected pixel are placed. If the value “time series” is selected in the combo box, then a click to a pixel on the map creates a diagram, which visualizes the time series of the selected raster mesh.

Figure 10 shows the interaction and communication between the SOS and the client application. In the first step the client identifies all WMS references for the last two hours with a GetObservation request. Within the request the *ResponseMode* parameter is set to OUT-OF-BANND in order to get WMS references instead of direct encoded data. As long as the service has data for the given interval, each data record is referenced by a WMS GetMap URL within an O&M result document. For each obtained WMS URL a layer is created, but not immediately shown. This is done in a loop over all layers, where each layer is visible for 500 ms, resulting in an animated picture.

Besides the possibilities to control the animation as well as zoom in/out or displace the map, an option to visualize a time series of one raster mesh (see Figure 9 popup diagram) was included. Time series data are delivered from a standard compliant *GetObservation* request, which contains besides the value *om:TimeSeriesObservation* for the *ResultModel* parameter a time interval and coordinates of the selected raster mesh as a spatial filter.

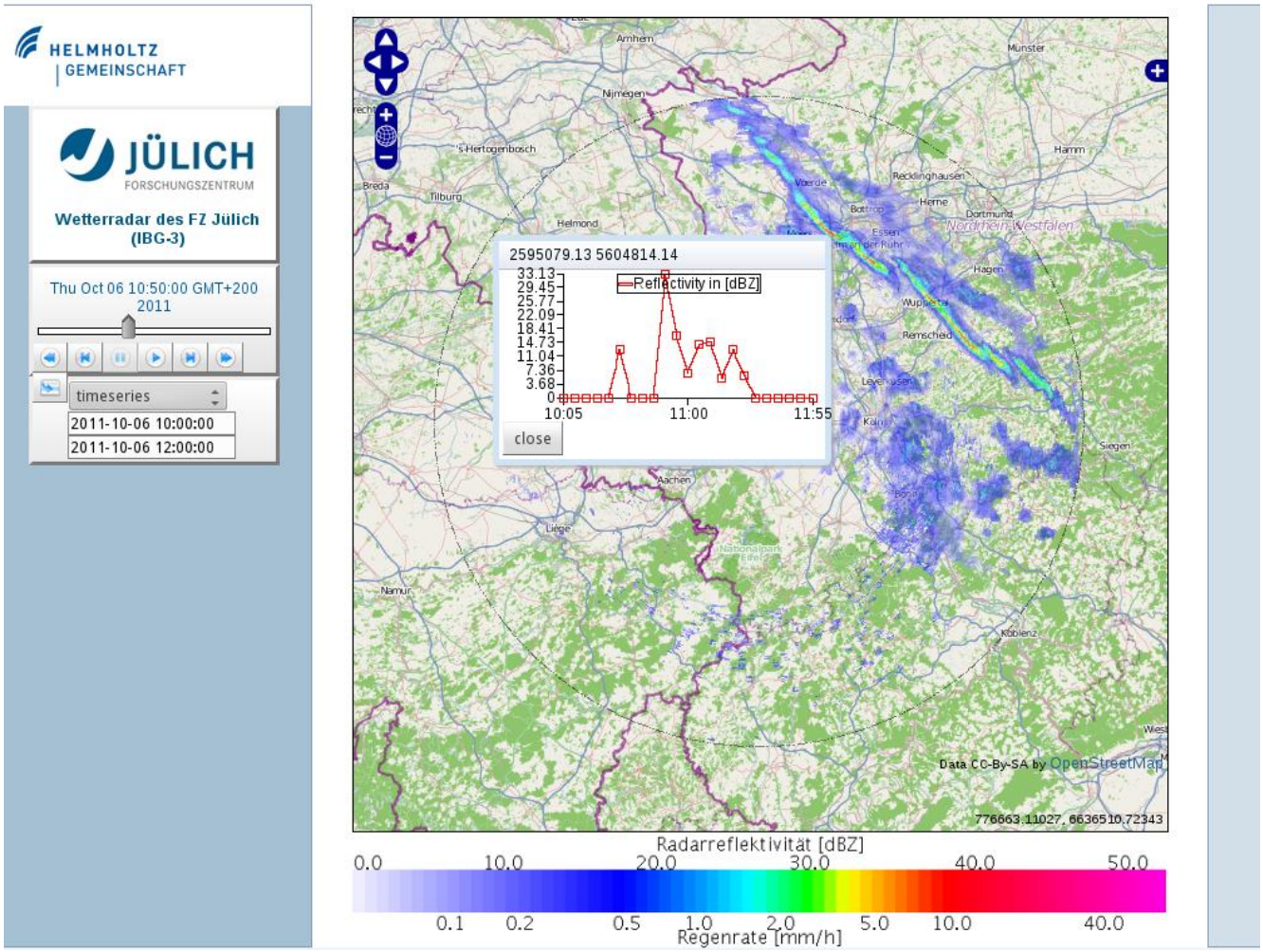


Figure 9. Screenshot of the web application visualizing the weather radar data.

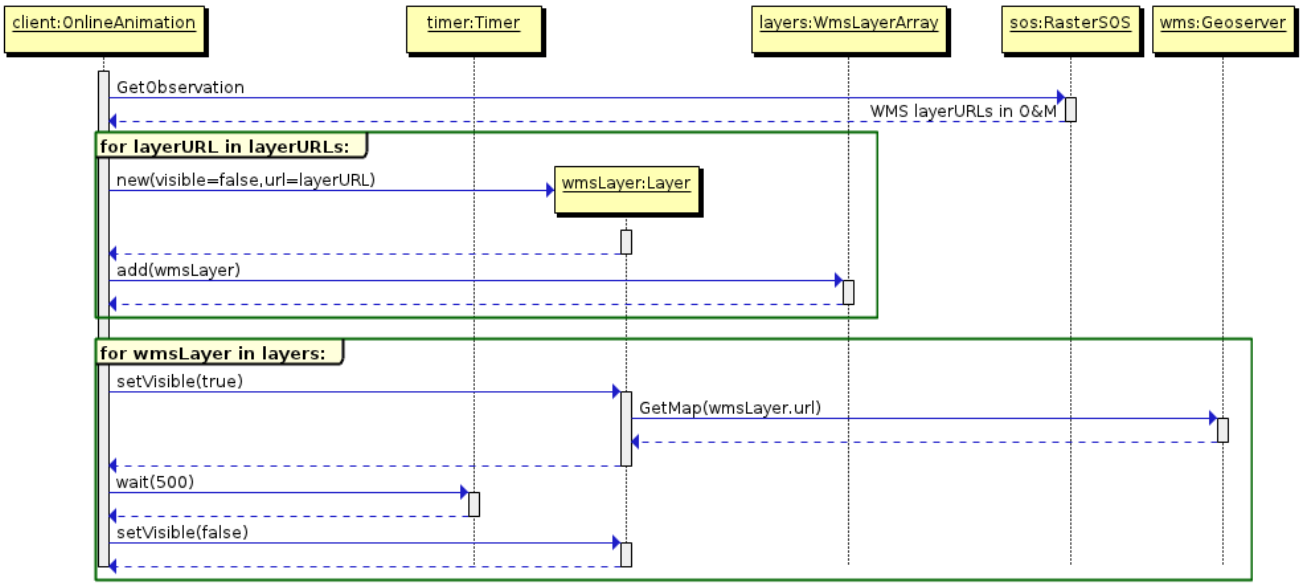


Figure 10. UML diagram of the communication between client, SOS, and WMS.

4.3. Client Application for Detecting Heavy Precipitation Regions

The second application implements a service for detecting heavy precipitation regions and alerting users to this event when they have registered for a notification for the concerning area. For registration and signaling the 52 north implementation of the OGC Sensor Alert Service (SAS) standard, acting as middleware between sensor and client, was used. If a sensor pushes alert messages of a certain type and a user is registered for this alarm type, he receives a handle to a channel opened by the SAS, which listens to incoming alert messages. Internally, the XML-based Extensible Messaging and Presence Protocol (XMPP) [32] is used to distribute alert messages over so-called multi-user chat rooms (MUCs). In our implementation we use the open source software system ejabberd (http://www.process-one.net/) as XMPP-Server. Users have the opportunity to register for 100 different zones, which divide the area observed by the weather radar into a raster with 10×10 raster meshes.

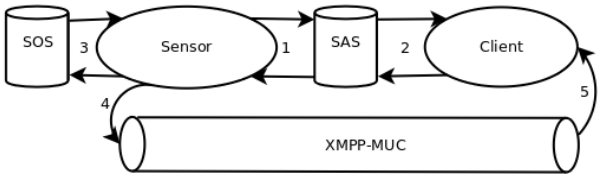


Figure 11. Communication between SOS, sensor, SAS, and client.

Figure 11 depicts the communication between the concerned components, where the interaction between each other is temporally ascending and given by numbers. On the left hand site a software component (Sensor), developed for this purpose registers an alert type on the SAS for each zone and stores the associated zone index to the received channel MUC (1). A zone index is returned to identify the channel. If a user registers for a zone (2), the index of the desired zone is referred and enables the SAS to identify the appropriate channel (XMPP-MUC). The sensor component uses the *GetObservation* requests to detect each five minutes if a heavy precipitation event has occurred (3). For this purpose a thematic filter with a threshold of 50 dBZ (this is equal to a precipitation quantity of about 48 mm/h), a temporal filter selecting the last record, and a *ResultModel* parameter with value *om:DiscreteCoverageObservation* are used. The SOS replies raster meshes, in which a heavy precipitation event has been occurred. All meshes will be unified to a not cogently connected polygon and intersected with the zone grid using the PostGIS "intersect" function. By means of the resulting zone index the appropriated channel (XMPP-MUC) is identified and the alert message is transmitted to all users that have already joined the chat room registered for this zone (4 and 5). Registration of a client and the client site representation of detected heavy precipitation events are done with a swing-based portable JAVA desktop application, which has the graphical user interface (GUI) depicted in Figure 12. On the left the GUI for registration is depicted. According to each activated checkbox a registration for each particular zone is performed as soon as the "subscribe" button is clicked. On the right of Figure 12, the red highlighted zones are those where a heavy precipitation event has occurred.

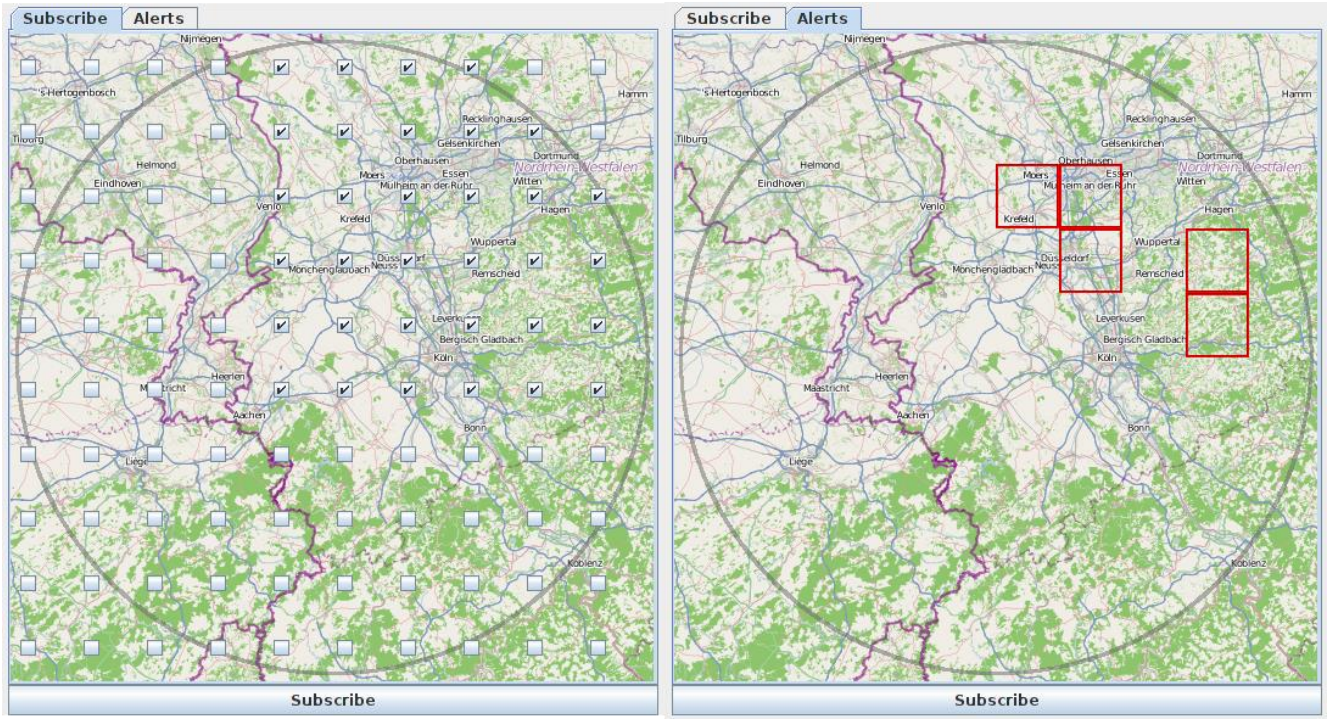


Figure 12. Graphical user interface (GUI) of the heavy precipitation detection application.

5. Results

Because the SOS and especially its filter functions have to extract information from the raster data, software components were designed and implemented to select raster data by geographic or thematic viewpoints. Vector-oriented filters defined by the SOS standard had to be translated into raster data. Therefore the challenge was to find a compromise between efficient data storage and efficient searching within the raster data. On the one hand, the system must be able to use database indices for an efficient search for thematic information within the raster data. On the other hand, it is not possible to store each raster data record in its own table, because of the huge amount of records a time series can have. A solution was accomplished by a free selectable coarse grid and by storing the raster data in a large binary field (BLOB) of the database. The resolution of the coarse grid was selected in such a way that each raster mesh of a record can be inserted in a table as one row and therefore a database index can be used to perform a faster search. For this an efficient sweep algorithm was developed, which reads regions from the binary stored raster data, which were selected by the coarse grid.

Some different O&M response models are allowed by the SOS standard, distinguishable by their accuracy and expressivity. Compared to the other models used, the *DiscreteCoverageObservation* model provides the most accurate and most expressible model to describe raster data. Therefore this model was used to return strong filtered raster data. With increasing accuracy the size of the returned documents increases, which is the reason that it is not possible to retrieve a larger number of data records by means of this model. A second O&M model was used to deliver comma-separated values, which decreased the document size by a factor of 100. To deliver raster time series data in a well-known, common scientifically used format, we implemented a SOS-compliant response in such a way that NetCDF files can be attached to the response and referenced by the responded O&M document.

To verify these issues, 14 scenarios were considered and compared with respect to their latencies. Results are shown in Table 2. The first eight scenarios retrieved complete raster records from the SOS (see Table 1, lines 1 to 8). The system needed more than 10 minutes to encode a data record with the *DiscreteCoverageObservation* model (see Table 2, line 1). The size of the created file containing one individual raster data record (640,000 raster meshes) was more than 600 MB. With the generic O&M model using comma-separated values, response time was reduced to 1.8 seconds and the document size was reduced to 7 MB (see Table 2, line 2). Gzip compression allowed a further file size reduction to 1MB (Table 2, line 3).

Table 2. Scenarios to analyze the response latency.

Sce-nario	Count	Meshes	Filter	Model	Mode	Format	Net	Time (s)	Size (MB)
1	1	640,000	--	Om:Discrete-Coverage-Observation	Inline	Plain	Intranet	600	600
2	1	640,000	--	Om:Observation	Inline	Plain	Intranet	1.8	7
3	1	640,000	--	Om:Observation	Inline	Zip	Intranet	1.08	1
4	60	640,000	--	Om:Observation	Inline	Plain	Intranet	210	438
5	60	640,000	--	Om:Observation	Inline	Zip	Intranet	200	90
6	23	640,000	--	Om:Observation	Inline	Plain	Internet	346	165
7	23	640,000	--	Om:Observation	Inline	Zip	Internet	155	34
8	8605	640,000	--	Om:Observation	Attached	NetCDF	Intranet	3240	365
9	1	40,000	spatial	Om:Observation	Inline	Plain	Intranet	0.2	0.4
10	60	40,000	spatial	Om:Observation	Inline	Plain	Intranet	12	28
11	60	250	thematic	Om:Observation	Inline	Plain	Intranet	26	0.5
12	60	1	spatial (rectangle)	Om:Time series-Observation	Inline	Plain	Intranet	6	0.02
13	60	1	spatial (point)	Om:Time series-Observation	Inline	Plain	Intranet	0.3	0.03
14	60	--	--	Om:Observation	Out-of-band	Plain	Internet	0.1	0.1

Beside single raster records, it is also possible to retrieve time series of complete raster records by means of the generic model with comma separated values or attached in a NetCDF file (see Table 2, lines 4 to 8). The compression is most efficient when data are transferred via the internet (see Table 2, "Net" column), because the cost of the compression is compensated for by the lower size of the transferring data (see Table 2, lines 6 to 7). Scenario 8 retrieves a time series of complete raster data records within a time period of one month, encoded and compressed in the NetCDF file format. The returned file of 365 MB contains 5,507,200,000 data values and was created, compressed, and delivered in less than one hour.

In scenarios nine to 11, 60 spatial and thematic filtered data records were retrieved. The last two scenarios retrieved time series data encoded with the *TimeSeriesObservation* model. Sixty records have been selected, where in scenario 12 the value of a mesh and in scenario 13 the average value of a rectangular region have been returned.

6. Discussion

As far as we know, this is the first implementation of the SOS standard for a standardized access to raster data. Based on version 1.0 of the SOS standard, our implementation comprises the mandatory core SOS operations *GetCapabilities*, *DescribeSensor*, and *GetObservation*. In 2012, version 2.0 of the SOS standard was adapted by the OGC; however, work is based on the older version, since development was started in 2010, when the TERENO project was established. The concepts presented here, especially the result formats, can easily be applied to version 2.0.

The common approach for the publication of raster data is the OGC WCS standard, which is able to handle multi-dimensional arrays since version 2.0 can be used to include a temporal dimension. The WCS(-EO) is designed to deliver big datasets, on one hand; on the other hand, consideration of the temporal relation is implemented as an add-on using an additional metadata set provided with each raster dataset. The OGC SOS standard, as one component of the OGC SWE framework, is designed for standardized access to all kinds of raster and vector time series data with spatial relation to the earth, inherits the temporal relation of each dataset, and allows for the definition of spatial and thematic filters. On the other hand, up to now the SOS has mostly been used for *in situ* sensors, since general concepts to combine the intrinsic vector-based geoprocessing with raster data are not available due, of course, to a lack of efficient data return models. Our implementation of a raster-SOS implements not all the functionalities of a WCS(-EO) due to the different standards. For instance, an SOS only supports 2-D geospatial data. 1-D or 3-D data possibly created as model output are neither in the scope of our implementation nor supported by the SWE components. On the other hand, a raster-SOS provides additional functionalities that are unique for the SOS standard, e.g., thematic and extended spatial filtering.

The analysis of the scenarios showed that the *om:DiscreteCoverageObservation* model is not able to provide raster data due to the vast data volume. On the other hand, we could show that it is indeed possible to deliver raster data in a reasonable time and size using the generic model (see Table 2, scenarios 4 to 7) or attaching a NetCDF file (see Table 2, scenario 8). In practice, even in vector-based SOS implementations data are usually returned by means of the generic model and not through the more expressive and detailed models, which would also be available for this kind of data (like the *om:Measurement* model [10,21]).

In spite of the large difference in responsiveness between the *DiscreteCoverageObservation* and *TimeSeriesObservation* models, the expressivity and accuracy, but also the storage requirement of both models is equal. The reason for the good score of the *TimeSeriesObservation* model is that there is always only one mesh or one average value of a region of the grid encoded with this model. Indeed, this model introduces the possibility of retrieving average values (for instance river catchments), which is not possible to specify elsewhere in a *GetObservation* request. Scenario 8 shows that raster data delivered attached in a NetCDF file give the opportunity to provide considerable raster time series data in a scientifically well-known format. An interoperable usage of the delivered data is still given by means of the CF-Convention.

The OUT-OF-BAND method allows access to raster time series data, which are provided by external services. For our application visualizing weather radar data it was sufficient and from a

performance perspective it is necessary to provide the data encoded as georeferenced colorized images. Beyond that it is possible to provide the actual raster data with this method, not by a WMS but with a Web Coverage Service (WCS) [33,34], which delivers actual raster data [14] and not only derived images. An exciting question about this is: "how can we convert a spatial filter of a SOS GetObservation request into a WCS GetCoverage request?" A possible approach for this would be to execute the GetCoverage request directly from the SOS. In this case the SOS would serve as a proxy between the client and the WCS. If both WMS and WCS response methods should be provided, then the possibility to distinguish this in a GetObservation request must also be available. Because both variants have to set the request parameter RequestMode to OUT-OF-BAND, only the parameter ResponseMode still remains. Therefore a unified specification of potential values for the ResponseFormat parameter would be desirable, which considers both cases (WCS and WMS).

7. Conclusions

In this paper we presented concepts and an implementation of an OGC-compliant SOS for an interoperable access to area-related raster time series data. Additional, a web application for fast, standardized, and interoperable access to raster data for a fast visual quality inspection has been implemented. A system was realized that retrieves filtered weather radar data from the SOS in a standard, compliant way to detect heavy precipitation events and alert registered users about this.

For use cases of the first kind, a system was accomplished that is a combination of SOS and WMS, which provides fast access to raster records of a time series. Concerning this, the metadata managed by the SOS have been combined with the raster maps managed by a WMS. This association is done by the timestamps, which are on the one hand part of the metadata, but are on the other hand also part of the WMS layer name. The raster images are produced *ad hoc* from data that are stored within the SOS data model. For this a Spring [35]-based plugin for the OGC- and WMS-compliant GeoServer was developed. The database system used for managing data by the SOS supports fast retrieval of the metadata and therefore provides short service latency. Assembled with more than 420,000 data records, the system extracts and encodes desired WMS references in a range of two- or one-figure milliseconds. Therefore the latency has been shortened by a factor of 1000 in the portion of the old system that stored the radar images within a content management system (CMS) and gave an interoperable access via an OGC catalog service (CSW). Interoperability could be improved by use of the SOS, because an SOS is designed especially for interoperable managing time series data.

A use case of the second kind, which detects heavy precipitation events by applying filters and alerts registered users about this, was realized by a software system that pushes alarm messages about those events to a Sensor Alarm Service (SAS). On the other hand, the system provides a GUI for client interactions and visualizations. Besides the concrete application detecting heavy precipitation events, a framework transmitting alert message via the XMPP protocol to the SAS has also been created, which enables the reusability of developed software components. On the basis of JAVA swing components, the GUI offers an easy-to-use and user-friendly opportunity to make a registration for alert messages and view highlighted regions, where heavy rain events have been occurred.

Our conclusion is that the SOS standard is a very good opportunity to publish raster time series data in a standardized way. With regard to 2-D geospatial data, the SOS approach is predominant compared

with the common WCS(-EO) approach due especially to its temporal and thematic filtering capabilities, but in particular due to the possibility of describing measurement equipment, measurement processes, and observations in detail by metadata standards, which are exactly defined for this purpose. Additional work will be done to specify an extension of the core SOS 2.0 for raster data, which uses the OGC coverage model [36] to facilitate encoding raster data in a more efficient way than the SOS version 1.0 and its used ISO 19123 concepts allow.

Acknowledgments

We gratefully acknowledge the support of TERENO (Terrestrial Environmental Observatories), funded by the Helmholtz Association.

Author Contributions

Juergen Sorg developed the SOS for raster data and also the client applications; Juergen Sorg and Ralf Kunkel designed the SOS for raster data; Juergen Sorg and Ralf Kunkel wrote the paper; Juergen Sorg designed and performed the test scenarios for the performance analysis.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. OGC. *OGC ® Sensor Web Enablement Architecture*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2008; p. 66.
2. Zacharias, S.; Bogena, H.; Samaniego, L.; Mauder, M.; Fuss, R.; Puetz, T.; Frenzel, M.; Schwank, M.; Baessler, C.; Butterbach-Bahl, K.; *et al.* A network of terrestrial environmental observatories in Germany. *Vadose Zone J.* **2011**, *10*, 955–973.
3. Kunkel, R.; Sorg, J.; Eckardt, R.; Kolditz, O.; Rink, K.; Vereecken, H. TEODOOR: A distributed geodata infrastructure for terrestrial observation data. *Environ. Earth Sci.* **2013**, *69*, 507–521.
4. Bröring, A.; Echterhoff, J.; Jirka, S.; Simonis, I.; Everding, T.; Stasch, C.; Liang, S.; Lemmens, R. New generation sensor web enablement. *Sensors* **2011**, *11*, 2652–2699.
5. OGC. *OpenGIS Sensor Observation Service Implementation Specification*; Open Geospatial Consortium Inc.: Mayland, MA, USA, 2006; p. 91.
6. OGC. *OGC® Sensor Observation Service Interface Standard*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2012; p. 148.
7. OGC. *Observations and Measurements—Part 2—Sampling Features;* Open Geospatial Consortium Inc.: Wayland, MA, USA, 2007; p. 36.
8. OGC. Geographic information: Observations and measurements. In *OGC Abstract Specification Topic 20;* Open Geospatial Consortium Inc.: Wayland, MA, USA, 2010; Volume 2.0.0, p. 57.
9. OGC. Observations and measurements. In *XML Implementation*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2011; Volume 2.0, p. 76.

10. OGC. *Observations and Measurements—Part 1—Observation Schema*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2007; p. 85.
11. OGC. OpenGIS sensor model language (SensorML). In *Implementation Specification*; Open Geospatial Consortium: Wayland, MA, USA, 2007; Volume 1.0.0, p. 180.
12. OGC. *OGC® Sensor Alert Service Implementation Specification*; Open Geospatial Consortium Inc: Wayland, MA, USA, 2007; p. 128.
13. OGC. *OpenGIS® Sensor Event Service Interface Specification (Proposed)*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2008; p. 88.
14. Nengcheng, C.; Liping, D.; Genong, Y.; Min, M. A flexible geospatial sensor observation service for diverse sensor data based on Web service. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 234–242.
15. Pfeiffer, T.; Wenk, A. *PostgreSQL: Das Praxisbuch*; Galileo-Press: Bonn, Germany, 2010.
16. Obe, R.O.; Hsu, L.S. *PostGIS in Action*; Manning Publications: Greenwich, CT, USA, 2011.
17. Holl, S. *PostGIS Raster Workshop*; FOSSGIS: Dessau-Rosslau, Germany, 2012; p. 18.
18. Kemper, A.; Eickler, A. *Datenbanksysteme—Eine Einführung*; Oldenbourg Wissenschaftsverlag GmbH: München, Germany, 2006.
19. Güting, R.H.; Dieker, S. *Datenstrukturen und Algorithmen*; Teubner Verlag: Wiesbaden, Germany, 2004.
20. OGC. *OpenGIS® Implementation Standard for Geographic Information—Simple Feature Access—Part 1: Common Architecture*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2011; p. 92.
21. Broering, A.; Meyer, O. Bereitstellung und visualisierung hydrologischer zeitreihen mit hilfe standardisierter webdienste. In Proceedings of AGIT2008 Symposium und Fachmesse, Salzburg, Österreich, 2–4 July 2008.
22. ISO. *Geoinformation—Coverage Geometrie—und Funktionsschema (ISO 19123: 2005)*; DIN Deutsches Institut für Normung: Berlin, Germany, 2005; p. 71.
23. OGC. *OpenGIS® Web Map Server Implementation Specification*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2006.
24. Kirschke, T. Implementierung eines OGC-konformen WMS als Plugin für einen Geoserver; FH-Aachen: Aachen, Germany, 2012.
25. Rew, R.K.; Davis, G.P.; Emmerson, S. NetCDF: An interface for scientific data access. *IEEE Comput. Graph. Appl.* **1990**, *10*, 76–82.
26. Eaton, B.; Gregory, J.; Drach, B.; Taylor, K.; Hankin, S. NetCDF Climate and Forecast (CF) Metadata Conventions. Available online: http://cfconventions.org/ (accessed on 30 January 2015).
27. Nativi, S.; Caron, J.; Davis, E.; Domenico, B. Design and implementation of netCDF markup language (NcML) and its GML-based extension (NcML-GML). *Comput. Geosci.* **2005**, *31*, 1104–1118.
28. Internet Engineering Task Force (IETF). Request for Comments: 2392—Content-ID and Message-ID Uniform Resource Locators. Available online: https://www.ietf.org/rfc/rfc2392.txt (accessed on 13 October 2014).
29. Marshall, J.S.; Palmer, W.M. The distribution of raindrops with size. *J. Meteorol.* **1948**, *5*, 165–166.

30. The PostgreSQL Global Development Group. PostgreSQL JDBC Driver. Available online: http://jdbc.postgresql.org/ (accessed on 30 January 2015).
31. Hanson, R.; Tacy, A. *GWT im Einsatz : AJAX-Anwendungen entwickeln mit dem Google Web Toolkit*; Carl Hanser Verlag: München, Germany, 2007.
32. Internet Engineering Task Force (IETF). Request for Comments: 6120—Extensible Messaging and Presence Protocol (XMPP): Core. Available online: https://tools.ietf.org/html/rfc6120 (accessed on 20 September 2014).
33. OGC. *Web Coverage Service (WCS)*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2003; p. 67.
34. OGC. *OGC® WCS 2.0 Interface Standard—Core*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2010; p. 45.
35. Amuthan, G. *Spring Mvc Beginner's Guide Your Ultimate Guide to Building a Complete Web Application Using All the Capabilities of Spring Mvc/Amuthan G*; Packt Pub.: Birmingham, UK, 2014.
36. OGC. *OGC® GML Application Schema—Coverages*; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2012; p. 35.