

# KONZEPTION UND IMPLEMENTIERUNG EINES OGC-KONFORMEN SENSOR OBSERVATION SERVICE FÜR DEN STANDARDISIERTEN ZUGRIFF AUF RASTERZEITREIHENDATEN

Jürgen Sorg, Timon Kirschke, Ralf Kunkel

**Zusammenfassung:** Interoperabilität von Geoinformationssystemen (GIS), d. h. die Schaffung von Möglichkeiten, heterogene Geodaten in einer einheitlichen, standardisierten Weise abrufbar zu machen, ist das Ziel des Open Geospatial Consortiums (OGC). Im Bereich der Sensordaten wird dieses Ziel innerhalb der OGC Sensor Web Enablement Initiative aufgegriffen und im Speziellen durch die Sensor Observation Service (SOS) Spezifikation erreicht. Diese definiert einen Dienst für den standardisierten Zugriff auf Zeitreihendaten und wird meist für In-situ-Sensoren (Abflusspegel, Klimastation etc.) verwendet. Die Spezifikation berücksichtigt auch Zeitreihendaten von Fernerkundungssensoren (Rasterdaten), es existieren allerdings hierfür bislang keine Implementierungen. In diesem Beitrag wird die Konzipierung und Implementierung eines OGC-konformen SOS beschrieben, mit dem Rasterzeitreihendaten verwaltet und interoperabel bereitgestellt werden. Hierfür wird ein Datenmodell entwickelt, das es erlaubt, die Rasterdaten in einer Datenbank mit den zugehörigen Metadaten zu speichern, diese Daten effizient für SOS-Anfragen auszulesen und in die von der SOS-Spezifikation vorgegebenen Rückgabeformate zu codieren. Zwei reale Fallstudien belegen die Anwendbarkeit des entwickelten Dienstes.

**Schlüsselwörter:** TERENO, Sensor Observation Service (SOS), Rasterdaten

## CONCEPTION AND IMPLEMENTATION OF AN OGC-COMPLIANT SENSOR OBSERVATION SERVICE FOR A STANDARDIZED ACCESS TO RASTER DATA

**Abstract:** The target of Open Geospatial Consortium (OGC) is interoperability of Geographic Information Systems (GIS), 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) specification. This one defines a service for a standardized access to time series data and is usually used for in-situ sensors (like discharge gauge, climate stations). The specification also considers raster data, but presently, no implementation of the standard for raster data exists. In this paper an OGC-compliant Sensor Observation Service for a standardized access to raster data is described and implemented. A data model was developed, which enables an effective storage of the raster data with the corresponding meta-data in a database, reading this data in an efficient way and encoding it with the result formats that SOS-specification provides. Two realistic case studies will show the usability of the developed service.

**Keywords:** TERENO, Sensor Observation Service (SOS), raster data

### Autoren

M. Sc. Jürgen Sorg  
Forschungszentrum Jülich GmbH  
Leo Brandt Straße  
D-52060 Jülich  
E: j.sorg@fz-juelich.de

B. Sc. Timon Kirschke  
Forschungszentrum Jülich GmbH  
Leo Brandt Straße  
D-52060 Jülich  
E: t.kirschke@fz-juelich.de

Dr. Ralf Kunkel  
Forschungszentrum Jülich GmbH  
Leo Brandt Straße  
D-52060 Jülich  
E: r.kunkel@fz-juelich.de

## 1 EINLEITUNG

Langfristige Änderungen der Temperatur, des Niederschlags und anderer Klimaparameter beeinflussen direkt und indirekt die terrestrischen Systeme Boden, Luft und Wasser und bedingen damit wirtschaftliche, soziale und politische Auswirkungen auf die Gesellschaft. Aufgabe der Forschung ist es, quantitative Aussagen über diesen globalen Wandel zu geben. Hierzu werden von den Wissenschaftlern der verschiedenen Umweltforschungsdisziplinen Modelle entwickelt, die es ihnen ermöglichen, diese Aussagen treffen zu können. Zentrale Grundlage dieser Modelle sind Umweltdaten in verschiedenen zeitlichen und räumlichen Skalen (Bronstert et al. 2009). TERENO (Terrestrial Environment Observatories) ist ein von der Helmholtzgemeinschaft ins Leben gerufenes, interdisziplinäres und langfristiges Forschungsprojekt, das „die langfristigen ökologischen, gesellschaftlichen und wirtschaftlichen Folgen des Klimawandels auf regionaler Ebene“ erfasst (Zacharias et al. 2011). TERENO trägt hierzu bei, indem eine solche Umweltdatenbasis erstellt und die Daten der wissenschaftlichen Gemeinschaft zur Verfügung gestellt werden. In TERENO werden vier terrestrische Observatorien betrieben, die von fünf Helmholtzzentren koordiniert und betrieben werden.

In diesen Observatorien werden seit 2009 Sensornetzwerke zur Intensivmessung von Bodenfeuchte und Bodentemperaturen, Wasserständen sowie Energie und Gasflüssen eingerichtet. Darüber hinaus wurden aber auch Fernerkundungsmessstationen wie Wetterradare bzw. Rainscanner zur Bestimmung von Niederschlagsmengen an insgesamt vier Standorten installiert. Die von den einzelnen Observatorien erhobenen Daten werden eigenverantwortlich in dezentralen, von den Observatorien administrierten Dateninfrastrukturen gespeichert und veröffentlicht. Im Datenportal TEODOOR (TEreno Online Data repository; <http://www.tereno.net>) können Wissenschaftler und Entscheidungsträger nach diesen Umweltdaten in Metadatenkatalogen suchen, ermittelte Daten visualisieren und herunterladen. Die Heterogenität der gemessenen Daten, aber auch die unterschiedlichen Messtechniken und die Verwendung von verschiedenen Datenbanksystemen machen es zwangsläufig erforderlich, Metadatenstandards für die Be-

schreibung der Daten und Messtechniken zu nutzen sowie die verteilten Daten von den einzelnen Observatorien über standardisierte Schnittstellen zur Verfügung zu stellen (Botts et al. 2008). Das Open Geospatial Consortium (OGC) definiert diese Standards und bietet Spezifikationen, die Dienste implementieren müssen, um einen interoperablen Zugriff auf Geodaten gewährleisten zu können.

In-situ-Messstation zur Bestimmung einer physikalischen Größe (Temperatur, Bodenfeuchte, etc.) beziehen sich immer genau auf einen geographischen Punkt. Hingegen liefern Fernerkundungsmessstationen flächenbezogene Daten, die sich auf ein ausgedehntes geographisches Gebiet beziehen. Der gängige Ansatz für den standardisierten Zugriff auf Rasterdaten stellt die von der OGC definierte Web-Coverage-Service(WCS)-Spezifikation dar (OGC 2003, OGC 2010b). Für die Verwaltung von Rasterdatenzeitreihen ist ein WCS allerdings ungeeignet, da die in einem WCS gespeicherten Rasterdatensätze keinen zeitlichen Bezug besitzen. In OGC (2011b) wird daher eine Spezialisierung der WCS Spezifikation für Erdbeobachtungen (WCS-EO) definiert, die zum einen durch die verpflichtende Angabe von Metadaten zu jedem Datensatz einen zeitlichen Bezug herstellt. Zum anderen können Datensätze (Datasets) zu identifizierbaren und anfragbaren Mengen (Dataset Series) zusammengefasst werden. Eine Beschreibung der jeweiligen Menge mit den darin enthaltenen Datensätzen kann durch die neue Operation *DescribeEOCoverageSet* ermittelt werden, wobei die enthaltenen Datensätze durch einen zeitlichen Filter selektierbar sind. In einem zweiten Schritt werden anhand von eindeutigen Bezeichnern die Daten durch *GetCoverage*-Operationen abgerufen.

Einen bislang nicht verfolgten Ansatz, Rasterzeitreihendaten zu verwalten, bietet die vom OGC definierte Sensor-Observation-Service(SOS)-Spezifikation. Diese umfasst Methoden für einen standardisierten Zugriff auf alle Arten von Zeitreihendaten mit räumlichem Bezug zur Erde. Allerdings existiert keine Implementierung für Fernerkundungsdaten (remote sensors). Der Vorteil, einen SOS anstatt eines WCS-EO zu nutzen, liegt darin, dass zum einen der zeitliche Bezug den einzelnen Datensätzen inhärent ist, da ein SOS für die Verwaltung

von Zeitreihen konzipiert ist. Die zeitliche Selektierung von Datensätzen erfolgt direkt und muss nicht durch eine zusätzliche Operation und anschließender Extraktion der benötigten eindeutigen Bezeichner erfolgen. Zum anderen ermöglicht es ein SOS, Filter anzugeben, die es erlauben, thematische Merkmale in Rasterdatensätzen zu extrahieren. In diesem Artikel werden die Konzeption und die Implementierung eines OGC-konformen SOS für den standardisierten Zugriff auf Rasterdaten beschrieben. Dieser ist in der Lage, unter Berücksichtigung von zeitlichen, räumlichen und thematischen Filtern Rasterdaten auszuwählen und in einer standardisierten Weise auszuliefern. Darüber hinaus wird eine Lösung beschrieben, die Rasterdaten für eine schnelle oberflächliche Kontrolle der Daten in einer zeitkritischen Visualisierung in einer Website aufbereitet und standardisiert zur Verfügung stellt. Anhand von zwei praxisorientierten Anwendungsfällen wird die Anwendbarkeit der entwickelten Methoden belegt.

## 2 OGC SENSOR OBSERVATION SERVICE

Das Open Geospatial Consortium (OGC) ist ein Zusammenschluss von führenden GIS-Herstellern, Datenproduzenten, Behörden, Organisationen, Forschungseinrichtungen und Universitäten. Es wurde 1994 gegründet und entwickelt seitdem Metadatenstandards und standardisierte Schnittstellen für den interoperablen Zugriff auf geographische Daten. Die interoperable Nutzung von Sensordaten wird innerhalb des OGCs durch die Sensor Web Enablement Initiative (SWE) abgedeckt (Bröring et al. 2011). Darin sind mehrere aufeinander abgestimmte Standards bzw. Schnittstellen definiert:

- Sensor Observation Service (SOS) definiert einen Standardwebedienst für die Suche, das Filtern und den Abruf von Sensordaten und -informationen (OGC 2006a, OGC 2012).
- Observation & Measurement ist ein Standard zur Beschreibung und zum Austausch von Beobachtungsdaten (OGC 2007a, OGC 2007b, OGC 2010a, OGC 2011a). Dieser wird vor allem zum Codieren von Geodaten verwendet, die ein SOS liefert.
- Sensor Model Language (SensorML) ist ein Informationsmodell zur Beschrei-

bung von Sensoren und der von ihnen gelieferten Informationen (OGC 2007d).

- Sensor Alert Service (SAS) ermöglicht es Nutzern, über eine Standardschnittstelle Alarmnachrichten von einzelnen Sensoren zu erhalten (OGC 2007c). Derzeit wird diskutiert, den SAS-Standard durch einen erweiterten Sensor Event Service (SES), der alle Arten von Sensorereignissen zur Verfügung stellen kann, zu ersetzen (OGC 2008).

Neben diesen Standards umfasst das SWE auch die Standards Transducer Model Language (TML), Sensor Planing Service (SPS) und Web Notification Service (WNS), die aber hier nicht weiter betrachtet werden.

Die SOS-Spezifikation umfasst für den standardisierten Zugriff auf Beobachtungsdaten insgesamt elf Operationen, wobei allerdings nur die Operationen *GetCapabilities*, *DescribeSensor* und *GetObserva-*

*tion* verpflichtend sind. Die *GetCapabilities*-Operation liefert neben allgemeinen Informationen zum SOS alle notwendigen Informationen für den Aufruf der unterstützten Operationen. Die *DescribeSensor*-Operation liefert eine Beschreibung eines Sensors, codiert im SensorML-Standard (OGC 2007d), die unter anderem die Bezeichnungen der gemessenen Parameter, die Koordinaten der Messstation und ein Zeitintervall, für das Daten vorhanden sind, enthält. Mittels einer *GetObservation*-Operation können schließlich die Daten vom Dienst abgefragt werden. Das in Beispiel 1 angegebene XML-Fragment zeigt ein Beispiel einer *GetObservation*-Anfrage mit allen zur Verfügung stehenden Parametern (OGC 2006a).

Der eindeutige und obligatorische Parameter *offering* bezeichnet eine frei wählbare, vom Anwendungsfall abhängige logische Kombination von Sensoren (z. B.

anhand der gemessenen Parameter) (OGC 2006a). Optional sind hingegen die Parameter zur zeitlichen, räumlichen und thematischen Filterung (*eventTime* Zeilen 3-11, *featureOfInterest* Zeilen 14-21 und *result* Zeilen 22-27) und die eindeutigen Bezeichner der Messstation (*procedure* Zeile 12) und der gemessenen Parameter (*observedProperty* Zeile 13).

Abbildung 1 zeigt einige Beispiele für räumliche Filter. Links ist ein Liniensegmentfilter dargestellt, der alle Werte extrahiert, die entlang eines Straßenverlaufs liegen. Mittig ist ein Rechteckfilter und rechts ein nicht orthogonaler, polygonaler Filter dargestellt. Beide Filter selektieren Objekte, die innerhalb der Filterflächen liegen.

Neben räumlichen Filtern sieht die SOS-Spezifikation auch thematische Vergleichsfilter vor. Hierbei werden Objekte nicht räumlich selektiert, sondern anhand eines Werts einer Variablen. In den Zeilen

```

1  <sos:GetObservation>
2    <sos:offering>Reflectivity</sos:offering>
3    <sos:eventTime>
4      <ogc:TM_During>
5        <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
6        <gml:TimePeriod>
7          <gml:beginPosition>2012-01-07T10:28:54</gml:beginPosition>
8          <gml:endPosition>2012-01-07T12:28:54</gml:endPosition>
9        </gml:TimePeriod>
10     </ogc:TM_During>
11   </sos:eventTime>
12   <sos:procedure>WeatherradarSophienhoehe</procedure>
13   <sos:observedProperty>Reflectivity</sos:observedProperty>
14   <sos:featureOfInterest>
15     <ogc:Intersects>
16       <ogc:PropertyName>urn:ogc:data:location</ogc:PropertyName>
17       <gml:Point srsName="urn:ogc:def:crs:EPSG::31466">
18         <gml:pos>5657550.508276477 2606247.091316845</gml:pos>
19       </gml:Point>
20     </ogc:Intersects>
21   </sos:featureOfInterest>
22   <om:result>
23     <ogc:PropertyIsGreaterThan>
24       <ogc:PropertyName>urn:ogc:def:phenomenon:Reflectivity</ogc:PropertyName>
25       <ogc:Literal>36</ogc:Literal>
26     </ogc:PropertyIsGreaterThan>
27   </om:result>
28   <sos:responseFormat>text/xml;subtype="OM/1.0.0"</sos:responseFormat>
29   <sos:resultModel>om:TimeSeriesObservation</sos:resultModel>
30   <sos:responseMode>inline</sos:responseMode>
31 </sos:GetObservation>

```

Beispiel 1: *GetObservation*-Anfrage mit allen zur Verfügung stehenden Parametern

22-27 im Beispiel 1 wurde ein thematischer Filter angegeben, der alle Rasterma-schen eines Rasterdatensatzes selektiert, deren Attribut „urn:ogc:def:phenomenon: Reflectivity“ einen Wert größer 36 besitzen.

Der SOS verarbeitet Anfragen unter Berücksichtigung der Filterbedingungen und gibt das Ergebnis zusammen mit den beschreibenden Metadaten als O&M-Dokument (OGC 2007a, OGC 2007b, OGC 2010a, OGC 2011a) zurück. Dabei ermöglicht es der O&M-Standard, auf der Basis von XML-Dokumenten sowohl Vektordaten als auch Rasterdaten zu codieren. Entscheidend dabei ist, dass im O&M-Standard es trotz der sehr unterschiedlichen Struktur möglich ist, sowohl Raster- als auch Vektordaten zu codieren (OGC 2007b). Hier wird einerseits das im ISO19123 (ISO 2005) standardisierte verallgemeinerte Coveragemodell zur Codierung von Rasterdaten benutzt. Andererseits werden Vektordaten durch mehrere speziell hierfür entwickelte Modelle abgedeckt. Innerhalb einer *GetObservation*-Anfrage wird das gewünschte Rückgabemodell durch den Parameter *resultModel* (Zeile 29 im Beispiel 1) spezifiziert (OGC 2006a). Theoretisch ist es damit möglich, Rasterdaten als auch Vektordaten mithilfe eines SOS standardisiert abzurufen. Allerdings werden Rasterdaten von den gängigen SOS-Implementierungen nicht unterstützt.

### 3 IMPLEMENTIERUNG

Es existieren eine Vielzahl von Implementierungen des SOS-Standards in den unterschiedlichsten programmiersprachlichen Ausprägungen (Nengcheng et al. 2009).

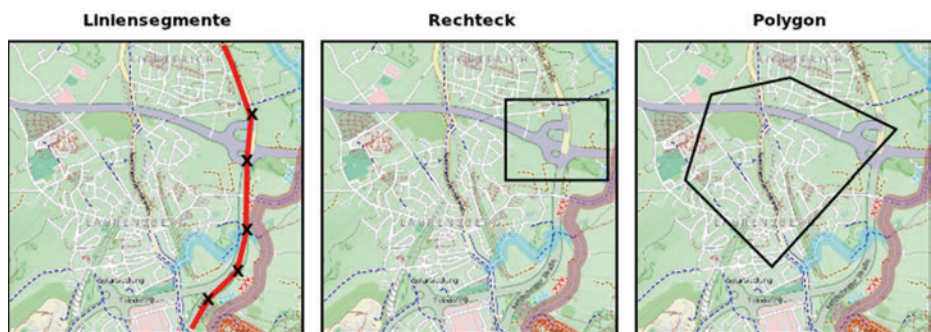


Abbildung 1: Räumliche Filter (links Liniensegment-Filter, mittig Rechteckfilter, rechts nicht orthogonales polygonales Filter)

Dabei stellt sich die Implementierung der Firma 52°North unter anderem durch ihr Dreischichtenmodell als eine sehr gut erweiterbare Lösung dar, die deshalb für den hier entwickelten SOS für Rasterdaten als Basis verwendet und für den standardisierten Zugriff auf Rasterdaten erweitert wurde. Hierzu waren die folgenden Schritte erforderlich, die in den folgenden Unterkapiteln erläutert werden:

- Für die effiziente Speicherung von Rasterdaten und den dazugehörigen Metadaten wurde ein auf dem 52°North-SOS basierendes Datenmodell für Rasterdaten entwickelt (Abbildung 2).
- Für die Anwendung von Filtern wurden effiziente Algorithmen implementiert.
- Für eine standardisierte und flexible Rückgabe der Rasterdaten wurden mehrere O&M-Modelle realisiert.

greSQL (Pfeiffer & Wenk 2010) mit der Erweiterung PostGIS (Obe & Hsu 2011) für räumliche Daten verwendet wird. PostGIS unterstützt in seiner neuesten Version (2.0) Rasterdaten. Allerdings kann für die Verwaltung von Rasterzeitreihen diese Erweiterung nicht genutzt werden, da PostGIS für jeden Rasterdatensatz eine Tabelle vorsieht (Holl 2012). Da davon ausgegangen werden kann, dass eine hohe Anzahl von Datensätzen (z. B. wird das in Kapitel 4 verwendete Wetterradar bis zum Ende des TERENO-Projekts voraussichtlich ca. 1,5 Millionen Rasterdatensätze erzeugen) pro Zeitreihe anfallen werden, ist die Verwendung von PostGIS-Rastertabellen nicht praktikabel. Aus diesem Grund wurde ein Datenmodell entwickelt und eingesetzt, mit dem die Rasterdaten in einer einzigen Tabelle abgelegt werden können. Die Speicherung in konventionellen Tabellen kann auf zwei unterschiedlichen Wegen erfolgen: als *Binary Large Object (BLOB)* oder zeilenbasiert. Unter einem BLOB versteht man einen Datenbankdatentyp für große,

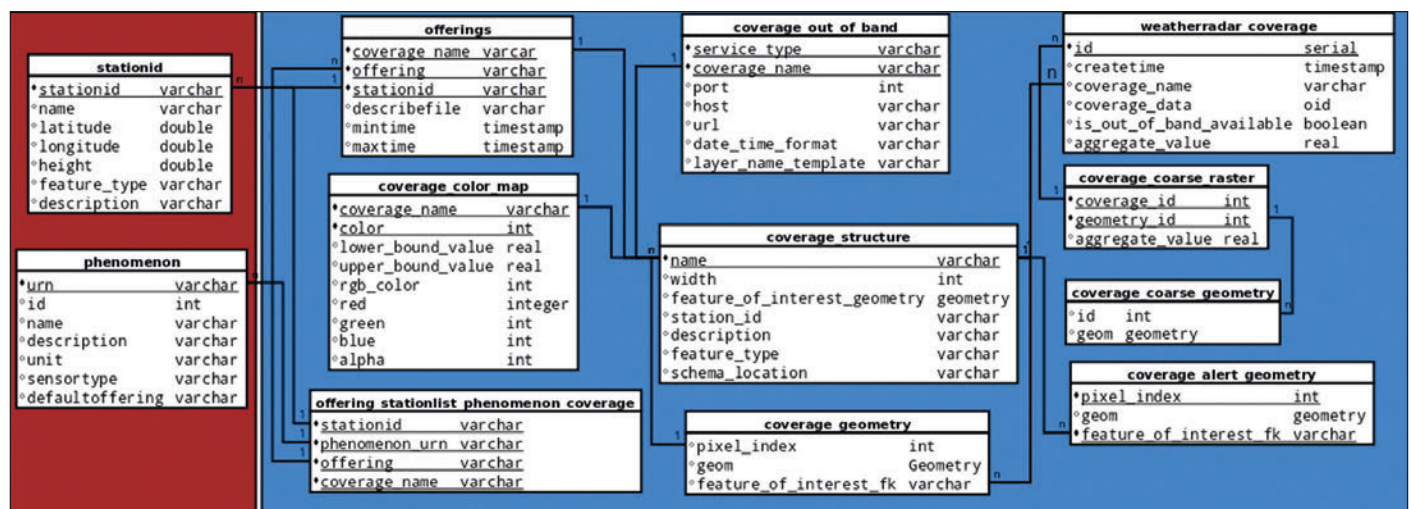


Abbildung 2: Datenmodell



nicht weiter spezifizierte binäre Objekte. Da in einem BLOB keine weiteren Informationen zum gespeicherten Objekt abgelegt werden, wird das Speichervolumen auf Tabellenebene minimiert. Allerdings ist die Speicherung in einem BLOB für die thematische Filterung der Daten sehr ungünstig, weil dadurch das komplette binäre Objekt gelesen und für alle Rastermaschen die Filterbindung überprüft werden muss. Die Speicherung jeder Rasterzelle als einzelnes Datenbankobjekt hat zwar den Vorteil, dass effizient mittels Indizes thematisch in den Rasterdaten gesucht werden kann, der erhebliche Nachteil ist hingegen die hohe Anzahl von Einfügeoperationen und Zeilen, die für jeden Rasterdatensatz anfallen (bei einem relativ kleinen Raster mit  $800 \times 800$  Pixeln beläuft sich dies schon auf 640.000 Einfügeoperationen bzw. Zeilen pro Datensatz). Um effizient thematisch in den Rasterdaten suchen zu können, wurde daher ein größeres Gitter erstellt, das einen Kompromiss zwischen geringem Speichervolumen und effizientem Zugriff darstellt. Auf diese Weise kann durch Datenbankindizes ein viel effizienterer Zugriff eröffnet werden, als dies durch eine sequenzielle Methode möglich wäre (Kemper & Eickler 2006). Das grobe Gitter ist in seiner Auflösung frei wählbar und beinhaltet für jede seiner Rastermaschen den minimalen sowie den maximalen thematischen Wert der darunterliegenden originalen Rastermaschen. Die Auflösung des Gitters muss so gewählt werden, dass einerseits die Einfügeoperationen in einer adäquaten Zeitspanne durchgeführt werden können, andererseits aber nicht zu viele Originalpixel gelesen und überprüft werden müssen.

### 3.2 FILTER

Die in der 52°North Implementierung enthaltenen Filter sind an die datenbank-intrinsischen Funktionen gebunden und unterstützen derzeit nur vektorbasierte Abfragen. Gerade bei diesem hier vorgestellten SOS für Rasterdaten sind aber Filteroperationen erforderlich, die auf die Rasterdaten zugreifen müssen. Dies wird unter Verwendung eines groben Rasters realisiert. Das grobe Raster hat zum einen den wesentlichen Vorteil, dass ein Datenbankindex für die Datensuche benutzt werden kann. Zum anderen müssen nur relevante Teile aus den jeweiligen binären Objekten gelesen werden. Die selektierten

Pixel im groben Raster stellen immer ein orthogonales Polygon dar, das allerdings nicht notwendigerweise zusammenhängend sein muss. Deshalb wurde für diese orthogonalen geometrischen Flächen ein Sweep-Algorithmus (Güting & Dieker 2004) entwickelt. Abbildung 3 zeigt einen Ausschnitt eines solchen orthogonalen Polygons. Der Sweep-Algorithmus tastet das Polygon in vertikaler Richtung von oben nach unten ab und führt in einem temporären Speicher Liniensegmente, die Bereiche markieren, welche blockweise gelesen werden. Im Beispiel werden im Abschnitt  $t_i$  bis  $t_{i+1}$  Blöcke von originalen Daten gelesen, die die Liniensegmente  $P_1P_2$  und  $P_3P_4$  markieren. Zum Zeitpunkt  $t_{i+1}$  wird der temporäre Speicher aktualisiert, wobei ab hier Blöcke gelesen werden, die das Liniensegment  $P_1P_4$  markieren. Für

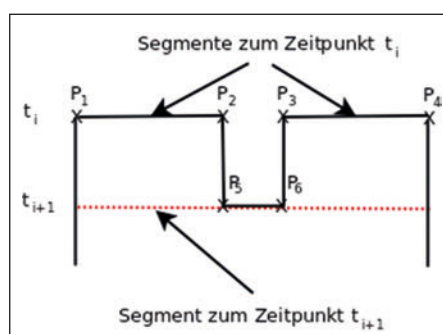


Abbildung 3: Sweep-Algorithmus zum Abtasten orthogonaler Polygone

die gelesenen Datenwerte wird die Filterbedingung überprüft, und bei bestandener Prüfung werden die Koordinaten der Rastermasche errechnet.

Der Sweep-Algorithmus kann auch für räumliche Filter verwendet werden, die direkt auf das Originalraster zugreifen. Dies ist jedoch nur möglich, wenn der räumliche Filter eine orthogonale Fläche ist (vgl. Abbildung 1, Mitte). Bei nichtorthogonalen räumlichen Filtern (Abbildung 1, links bzw. rechts) können die Daten nicht mit dem Sweep-Algorithmus gelesen werden. Für diesen Fall wurde eine vektorisierte Repräsentation des Rasters verwendet, in der jede Rastermasche des Originalrasters durch die Geometrie und den Rasterzellenindex realisiert ist. Durch Anwendung der im Datenbanksystem vorhandenen PostGIS-Funktionen (Intersects, Contains etc.) (Obe & Hsu 2011), die als Argumente den Fil-

ter und die Geometriespalte der Geometrierelation erhalten, werden die dem Filter genügenden Rastermaschen ermittelt. Der Zugriff auf die im BLOB abgelegten Daten erfolgt in diesem Fall über den Rasterzellenindex.

### 3.3 RÜCKGABE DER WERTE

Die Rückgabe der Werte kann auf zwei verschiedene Arten erfolgen, um die in Kapitel 4 beschriebenen Arten von Anwendungen zu bedienen. Zum einen werden die vom O&M-Standard für die direkte Codierung von Rasterdaten benutzten Modelle *DiscreteCoverageObservation* und *TimeSeriesObservation* sowie ein generisches O&M-Modell (OGC 2007b) (Broering & Meyer 2008) dazu benutzt, räumlich wie auch thematisch gefilterte Daten zu codieren. Zum anderen werden komplette Rasterdatensätze codiert und als georeferenzierte Bilder indirekt über einen externen Referenzmechanismus (WMS) dem Anwender bereitgestellt.

Für die direkte Übertragung (im Beispiel 1 durch den Wert „inline“ für den Parameter *ResponseMode* spezifiziert) von gefilterten Rasterdaten wurden zwei Modelle des O&M-Standards benutzt: Das *DiscreteCoverageObservation*-Modell für die Codierung von thematisch wie auch räumlich gefilterten Daten und das *TimeSeriesObservation*-Modell für die Codierung einer Zeitreihe einer selektierten Rastermasche. Die Rastermasche wird dabei in der Anfrage durch einen punktuellen räumlichen Filter angegeben. Allerdings ist es auch hier möglich einen räumlichen Filter mit einer beliebigen Fläche anzugeben, wobei dann, im Gegensatz zum *DiscreteCoverageObservation*-Modell, nicht alle Rastermaschen innerhalb der Filterfläche zurückgegeben werden, sondern der aus diesen resultierende Mittelwert. Dadurch ist es beispielsweise möglich, mit dem *TimeSeriesObservation*-Modell Mittelwerte eines Einzugsgebiets abzufragen. Beide Modelle werden aus dem ISO19123 Standard (ISO 2005) in den O&M-Standard inkludiert. Beispiel 2 zeigt das Geometrie-Wert-Paar einer Rastermasche, codiert mittels des *DiscreteCoverageObservation*-Modells.

Die Geometrie der Rastermasche wird als Polygon in der Geography Markup Language (GML) angegeben; der zugeordnete thematische Wert als Dezimalzahl

mit der dazugehörenden Einheit (UOM). In ähnlicher Weise wird auch im *Time-SeriesObservation*-Modell ein Datensatz codiert, wobei anstelle der Geometrie ein Zeitstempel benutzt wird. Da beide Modelle durch die ausführliche Beschreibung der Rasterzellen ein hohes Datenvolumen bedingen, wurde eine dritte Möglichkeit geschaffen, welche die Datenwerte kommasepariert in einem generischen O&M-Modell zurückgibt (OGC 2007b). Dabei werden für jede Rasterzelle der Index bzw. die Geometrie und der Wert angegeben. Eine weitere Reduktion des Datenvolumens wird durch eine auf dem gzip-Algorithmus (IEFT 1996) basierte Komprimierung des Rückgabedokuments erreicht. Die SOS-Spezifikation sieht hierfür den Wert „application/zip“ (Content-Type) für den Parameter *responseFormat* (Zeile 28 im Beispiel 1) vor.

Für die indirekte Übertragung (im Beispiel 1 würde dies durch den Wert „out-of-band“ für den Parameter *ResponseMode* spezifiziert werden) bietet der O&M-Standard die Möglichkeit, als Antwort auf eine *SOS-GetObservation*-Anfrage eine Referenz auf eine externe Quelle zu liefern (OGC 2007b). Für die Implementierung des SOS wird diese Option genutzt, indem die Rasterdaten als georeferenzierte Bilder

von einem OGC-konformen WebMap-Service (WMS) bereitgestellt werden und die einzelnen Rasterdatensätze jeweils mittels einer Referenz auf diesen Dienst in einem O&M-Dokument zurückgegeben werden. Dabei bietet ein WMS einen standardisierten Zugriff auf georeferenzierte Rasterbilder (OGC 2006b). Innerhalb der *SOS-GetObservation*-Anfrage wird durch die Angabe des Rückgabemodus (*ResponseMode*) *out-of-band* die Datenübertragung über den WMS spezifiziert. Da ein WMS ausschließlich rechteckige räumliche Filter unterstützt, werden die in der *GetObservation*-Anfrage angegebenen Filterflächen in die umschließenden Rechtecke (Bounding Box) transformiert und an den WMS weitergeleitet. Bei thematischen Filtern wird das umschließende Rechteck aus allen ermittelten Rasterzellen errechnet. Auch sind die zurückgegebenen Daten durch Farbwerte codiert, da ein WMS Bilder und keine rohen Rasterdaten liefert. Beispiel 3 zeigt eine Referenz auf ein Rasterbild codiert im O&M-Format.

Implementiert wurde diese Art der Datenbereitstellung durch die Erweiterung des quelloffenen, WMS-Standard konformen GeoServer-Projekts (<http://geoserver.org>), um ein Plug-in für den Zugriff auf die im SOS-Datenmodell gespeicherten Rasterda-

ten (Kirschke 2012). Dadurch wird die zusätzliche Speicherung von Daten im GeoServer selbst vermieden. Die WMS-Spezifikation schreibt zwei obligatorische Operationen vor. Für eine Dienstbeschreibung ist die *GetCapabilities*-Operation und für den Zugriff auf Daten die *GetMap*-Operation definiert. Formuliert werden beide Operationen mittels GET-Anfragen über das Hypertext-Transfer-Protokoll (HTTP) an den Dienst. Konform mit der Implementierungsspezifikation für einen WMS der Version 1.3.0 stellt das Plug-in die Methoden *GetMap* und *GetCapabilities* zur Verfügung (OGC 2006b). Für die dynamische Erstellung der Bilder bei einer *GetMap*-Anfrage werden die Rasterdaten mit ihren beschreibenden Metadaten aus der Datenbank geladen, daraus ein georeferenziertes Bild unter Berücksichtigung der Anfrageparameter gerendert und dieses im gewünschten Format (hier: PNG) zurückgegeben. Die Erweiterung beinhaltet zusätzlich einen Cache, der die gesamten Metainformationen aus der Datenbank mit einer definierten Aktualisierungsrate liest. Dieser vorgelagerte Cache ist insbesondere deshalb sinnvoll, da sonst bei einer hohen Anzahl von Datensätzen das Laden der Metadaten zu jedem Datensatz eine hohe Antwortzeit bedingen würde.

```
<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>
```

**Beispiel 2:** Geometrie-Wert-Paar einer Rastermasche codiert im *DiscreteCoverageObservation*-Modell

## 4 ANWENDUNGEN

Im Folgenden werden zwei Anwendungen betrachtet, die auf ihre benötigten Daten über den hier beschriebenen SOS für Rasterdaten zugreifen. Dieser SOS wird verwendet, um einen standardisierten Zugang zu den Wetterradardaten des Jülicher Wetterradars bereitzustellen. Das Radar befindet sich auf einem 30 m hohen Turm auf der 270 m hohen Abraumkippe Sophienhöhe des Tagebaus Hambach in der Nähe von Jülich. Das Radar versendet Mikrowellen, die von Niederschlagswolken reflektiert werden. Aus der Intensität der Reflektion kann mit der Formel von Marshall-Palmer der Niederschlag in mm/h errechnet werden (Marshall & Palmer 1948). Innerhalb einer Reichweite von 100 km misst das Radar alle Arten von Niederschlag (Regen, Schnee, Hagel etc.). Die erhobenen Radardaten stellen eine flächenbezogene Rasterzeitreihe mit einer zeitlichen Auflösung von fünf Minuten und einer räumlichen Auflösung von

250 m in einem 40.000 km<sup>2</sup> großen Bereich dar. Jedes Element der Zeitreihe entspricht einem PNG-Bild mit einer Auflösung von 800 x 800 Pixeln, das in jedem Pixel einen homogenen Reflektivitätswert, codiert als Grauwert, für das darunterliegende 62500 m<sup>2</sup> große Areal fasst. Die erste Anwendung visualisiert in einer Animation die Radardaten auf einer Website für eine schnelle Kontrolle durch den Benutzer. Die zweite Anwendung stellt einen Dienst dar, der registrierte Benutzer über eingetretene Starkregenereignisse alarmiert. Zunächst werden vorbereitende Konfigurationen und der Import der Daten in das Datenmodell des SOS erläutert.

### 4.1 DATENIMPORT

Für die Konfiguration des SOS wird in einem ersten Schritt jedes Pixel des Rasters vektorisiert, um die so erhaltenen Geometrien mit den Geometrien eines Filters schneiden zu können (siehe Kapitel 3.2). Im konkreten Fall wird das Raster für die

komplette Zeitreihe einmalig in 640.000 Polygone zerlegt. Die Polygone werden mit dem Pixelindex in eine Tabelle (*weatherradar\_geometries*, vgl. Abbildung 2) eingetragen. Um effizient einen Tief/Hochpassfilter auf ein Raster anwenden zu können, muss ein grobes Raster erzeugt werden. Es wird ein Raster mit der Ausdehnung des Originalrasters und einer Auflösung von 20 km erzeugt (*coverage\_coarse\_geometry*, vgl. Abbildung 2). Für die spätere farbliche Darstellung der Wetterradardaten wird eine Tabelle (*coverage\_color\_map*, vgl. Abbildung 2) mit Farbinformationen gefüllt, die jedem Grauwert des Originalbilds einen RGB-Farbwert zuordnet. Darüber hinaus werden Metadaten, wie Größe und Auflösung des Rasters, Einheit der Reflektivität (dBZ Dezibel Z, wobei Z für die Reflektivität steht), Name der Messstation, die Koordinaten der Messstation sowie eine detaillierte Beschreibung der Messstation im SensorML-Standard erzeugt. Für die Bereitstellung der Daten mithilfe eines

```
<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="WeatherradarSophienhoehe"/>
    <om:observedProperty xlink:href="urn:ogc:def:phenomenon:OGC:1.0.30:Reflectivity"/>
    <om:featureOfInterest>
      <gml:GridCoverage gml:id="WeatherradarSophienhoehe">
        <gml:name>WeatherradarSophienhoehe</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.menja.de: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>
```

*Beispiel 3: Referenz auf ein Rasterbild codiert im O&M-Format*

WMS werden die zum Referenzieren eines Rasterbilds benötigten Parameter (*coverage\_out\_of\_band*, vgl. Abbildung 2) gespeichert.

Der Import der Wetterradardaten erfolgt durch ein Java-Programm, das den Zeitstempel aus dem Dateinamen extrahiert, die Grauwerte der einzelnen Pixel in einem binären Feld (Array) zwischenspeichert und die Daten mittels der PostgreSQL-Large-Object-API (The PostgreSQL Global Development Group 2012) in einem binären Objekt (BLOB) im SOS-Datenmodell speichert. Während des Imports werden außerdem die Werte (Minimum, Maximum) der Rastermaschen des groben Rasters errechnet und in der Datenbank abgelegt (*coverage\_coarse\_raster*, vgl. Abbildung 2).

#### 4.2 VISUALISIERUNG DER WETTERRADAR DATEN

Zur animierten Visualisierung der Wetterradardaten wurde eine Clientanwendung implementiert, die mit der *out-of-band*-Methode (siehe Kapitel 3.3) Referenzen von Wetterradardaten eines gegebenen Zeitintervalls vom SOS abrufen, damit Overlays erzeugt und auf einer *OpenStreetMap*-Karte (OpenStreetMap contributors 2012) darstellt. Die Anwendung wurde mithilfe des *Google Web Toolkit* (Hanson & Tacy 2007) und dem *OpenLayers-Wrapper* für GWT (<https://bitbucket.org/gwtopenlay->

*ers/gwt-openlayers*) entwickelt und bietet die in Abbildung 4 dargestellte grafische Oberfläche sowie mehrere Komponenten zur Interaktion mit dem Benutzer.

Wie in Abbildung 4 zu erkennen ist, wird im Hintergrund eine *OpenStreetMap*-Karte angezeigt. Der darauf liegende Kreis stellt die Reichweite des Wetterradars dar. Im rechten oberen Bereich des Kreises ist eine Wetterfront zu sehen, die verschieden farbig dargestellt ist. Unterhalb des „Forschungszentrum Jülich“-Logos befinden sich Schaltflächen zum Steuern der Animation und Schaltflächen für die Erzeugung der Pixelzeitreihenvisualisierung. Wurde das Auswahlsschaltenelement auf „timeseries“ gesetzt, kann mit einem Klick auf die Wetterfront eine Zeitreihe einer Rastermasche als Diagramm angezeigt werden. Links oben auf der Karte sind Schaltknöpfe zum Vergrößern/Verkleinern vorhanden.

Das Zusammenspiel und die Kommunikation zwischen SOS und Clientanwendung ist in Abbildung 5 dargestellt. Im ersten Schritt ermittelt die Anwendung mittels einer *GetObservation*-Anfrage die WMS-Referenzen der letzten zwei Stunden. Dabei wird der Parameter *ResponseMode* auf den Wert *out-of-band* gesetzt und damit signalisiert, dass die Daten als Referenz auf einen WMS geliefert werden sollen. Sofern der Dienst Daten für die gewünschte Zeitspanne ermitteln konnte, werden diese in einem O&M-Dokument geliefert, wobei jeder Da-

tensatz im Dokument als eine vollständige WMS-GetMap-URL codiert ist. Für jeden übermittelten Datensatz wird ein Layer aus der WMS-GetMap-URL erzeugt, allerdings noch nicht angezeigt. Die Darstellung der erzeugten Layer erfolgt in einer Schleife, in welcher jeweils ein Layer für eine halbe Sekunde (500 ms) sichtbar wird. Dadurch entsteht die gewünschte animierte Visualisierung der Wetterradardaten.

Neben den Möglichkeiten, die Animation zu steuern sowie die Karte zu vergrößern/verkleinern und die Karte zu verschieben, besteht noch die Option, Zeitreihen von einzelnen Rastermaschen zu visualisieren (vgl. Abbildung 4). Die Daten der zu erstellenden Zeitreihe werden mittels einer *SOS-GetObservation*-Anfrage vom SOS standardkonform angefordert, wobei neben dem Wert *om:TimeSeriesObservation* für den *ResponseMode*-Parameter noch die gewünschte Zeitspanne sowie die Koordinaten der durch einen Mausklick selektierten Rastermasche als räumlicher Filter gesetzt werden.

#### 4.3 CLIENTANWENDUNG ZUR DETEKTION VON STARKREGENGEBIETEN

Die zweite Anwendung implementiert einen Dienst, der Starkregengebiete detektiert und Benutzern signalisiert, die sich für potenziell betroffene Gebiete registrieren haben. Als Dienst zur Registrierung und Signalisierung wird die 52°North-Referenzimplementierung der *Sensor-Alert-Service*-Spezifikation (SAS) (OGC 2007c) benutzt. Dieser dient als Middleware zwischen Sensor und Client. Dabei sendet ein Sensor Alarmnachrichten eines bestimmten Typs über einen Kanal, der zuvor beim Registrieren des Alarmtyps vom SAS geöffnet wurde. Registriert sich ein Benutzer für diesen Alarmtyp, so erhält er vom SAS den geöffneten Kanal und hört diesen auf ankommende Alarmbenachrichtigungen ab. Intern wird das auf XML basierende *Extensible Messaging and Presence Protocol* (XMPP) (IETF 2011) dazu verwendet, über sogenannte *Multi-User-Chat-Rooms* (MUC) die Alarmnachrichten zu verteilen. Dies erfolgt hier durch den quelloffenen XMPP-Server *ejabberd* (<http://www.process-one.net/>).

Die Benutzer haben die Möglichkeit, sich für 100 verschiedene Zonen zu registrieren, die das vom Wetterradar observier-

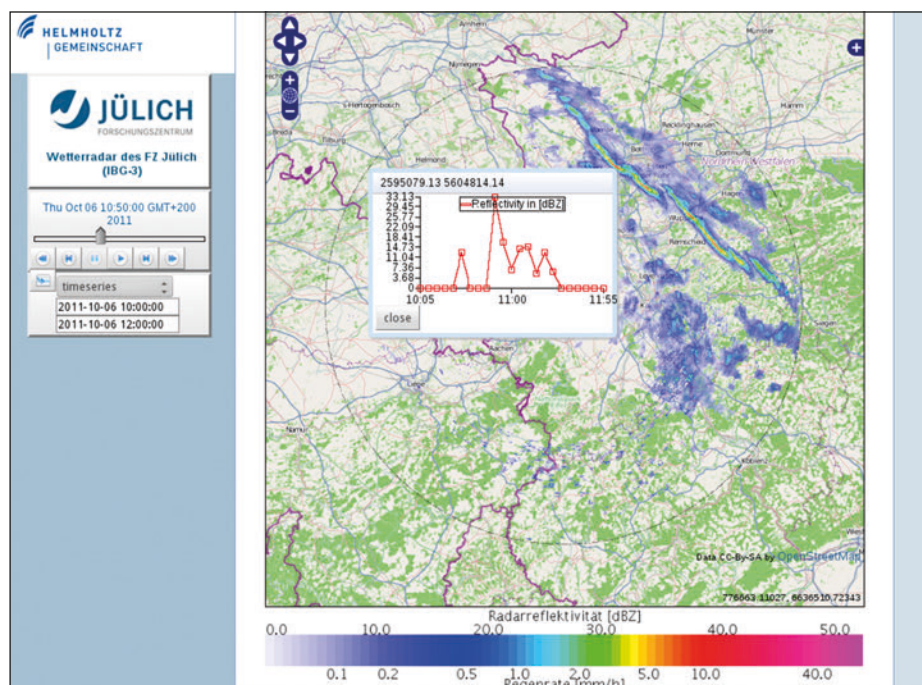


Abbildung 4: Animierte Visualisierung der Wetterradardaten



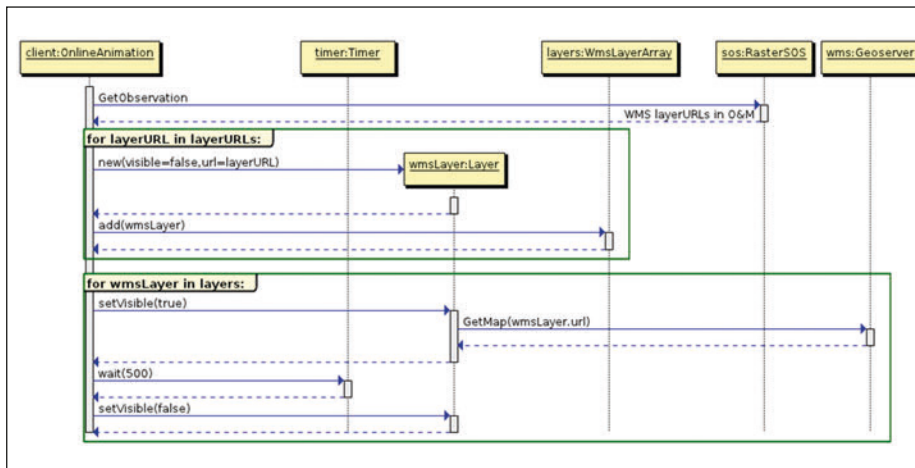


Abbildung 5: Kommunikation zwischen SOS und Clientanwendung

te Gebiet in ein Raster mit 100 Rasterzellen zerlegt. Abbildung 6 zeigt die Kommunikation zwischen den beteiligten Komponenten, wobei die Interaktion untereinander zeitlich aufsteigend durch Zahlen angegeben ist. Auf der einen Seite registriert eine hierfür entwickelte Softwarekomponente (im Weiteren als Sensorkomponente bezeichnet) für jede Zone einen Alarmtyp beim SAS und speichert den Zonenindex assoziativ zu dem als Antwort erhaltenen Kanal (MUC) (1). Dabei wird der Index der Zone zur Identifikation des Kanals übergeben. Registriert sich andererseits ein Benutzer für eine Zone (2), so wird dabei auch der Index der gewünschten Zone übergeben, und der SAS ist in der Lage, den passenden Kanal (XMPP-MUC) zu identifizieren und an den Aufrufer zurückzugeben. Die Sensorkomponente detektiert mittels *GetObservation*-Anfragen alle fünf Minuten, ob ein Starkregenereignis eingetreten ist (3). Hierzu wird in der *GetObservation*-Anfrage der Parameter *Resultmodel* auf *om:DiscreteCoverageObservation*, ein thematischer Filter mit einem Schwellwert von 50 dBZ (dies entspricht einer Niederschlagsmenge von ca. 48 mm/h) und ein

temporaler Filter, der den letzten Datensatz filtert, gesetzt. Die vom SOS zurückgegebenen Daten beinhalten die Rastermaschen, in denen ein Starkregenereignis eingetreten ist. Die Maschen werden zu einem nicht zwingend zusammenhängenden Polygon vereinigt und mittels der PostGIS-Funktion *intersect* mit dem Zonengitter geschnitten. Anhand des resultierenden Zonenindex wird der zu nutzende Kanal (XMPP-MUC) aus dem assoziativen Feld ermittelt und die Alarmnachricht über diesen Kanal an alle dem Chat-Raum beigetretenen bzw. für diese Zone registrierten Benutzer übermittelt (4 und 5). Die Registrierung eines Clients und die clientseitige Darstellung der detektierten Starkregenereignisse geschieht mittels einer auf Swing (Loy et al. 2002) basierenden betriebssystemunabhängigen Java-Desktopanwendung, die die in Abbildung 7 dargestellte grafische Benutzeroberfläche (GUI) bietet. Links ist die GUI zur Registrierung dargestellt, wobei durch Aktivieren der jeweiligen Auswahlsteuerelemente und Betätigen des Registrierungsschaltlements eine Registrierung für die ausgewählten Zonen erfolgt. Rechts in Abbildung 7 werden rot markierte Zonen

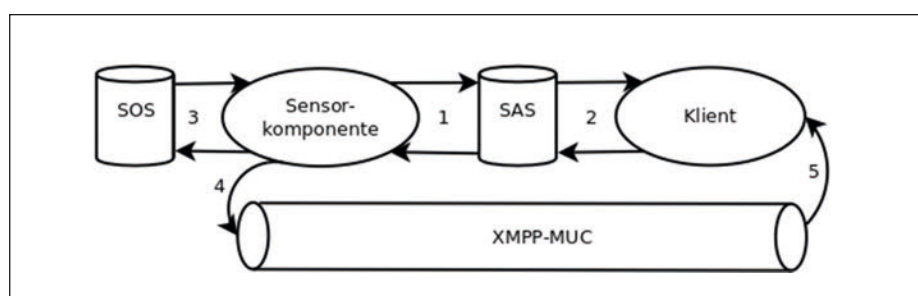


Abbildung 6: Kommunikation zwischen Sensor, SAS, SOS & Klient

angezeigt, in denen ein Starkregenereignis vorliegt.

## 5 ZUSAMMENFASSUNG, DISKUSSION UND AUSBLICK

Beschrieben wurde hier der Entwurf und die Implementierung eines OGC-konformen SOS für den interoperablen Zugang zu flächenbezogenen Rasterzeitreihendaten. Zusätzlich wurde zum einen eine Webanwendung für einen schnellen, standardisierten und interoperablen Zugang zu Rasterdaten für eine schnelle visuelle Überprüfung dieser Daten implementiert. Zum anderen wurde ein System realisiert, das gefilterte Wetterradardaten standardkonform vom SOS abrufen, damit Starkregengebiete detektiert und registrierte Benutzer vor diesen warnt.

Für Anwendungsfälle der ersten Art wurde durch Kombination von SOS und WMS ein System geschaffen, das einen schnellen Zugriff auf die Zeiteihenelemente einer Rasterzeitreihe ermöglicht. Hierzu wurden die vom SOS verwalteten Metadaten mit den in einem WMS hinterlegten Rasterkarten verbunden. Diese Assoziation geschieht mittels der Zeitstempel, die zum einen Teil der Metadaten, zum anderen Teil des Kartennamens (*layername*) sind. Die Rasterbilder werden ad hoc vom WMS aus den im SOS-Datenmodell gespeicherten Rasterdaten produziert. Hierzu wurde für den OGC-WMS-konformen Geoserver ein Plug-in auf Basis von Spring (Walls 2011) entwickelt. Das vom SOS zur Verwaltung der Daten benutzte Datenbanksystem gewährt eine schnelle Suche in den Metadaten und ermöglicht so eine kurze Antwortzeit des Diensts. Das System, bestückt mit über 50.000 Datensätzen, ermittelt die Referenzen auf die Karten im WMS im zwei- bis dreistelligen Millisekundenbereich. Dadurch verkürzt sich die Antwortzeit im Verhältnis zum Altsystem um einen Faktor von 1.000, das die Radarbilder in einem Content Management System (CMS) speichert und einen standardisierten Zugriff darauf über einen OGC-Katalogdienst (CSW) bereitstellt. Die Interoperabilität ist durch den Zugriff auf die Daten mittels eines SOS verbessert worden, da ein SOS speziell für die interoperable Verwaltung von Zeitreihendaten konzipiert ist.

Da der SOS und insbesondere dessen Filterfunktionen für Anwendungen der zweiten Art Informationen aus den Rasterdaten

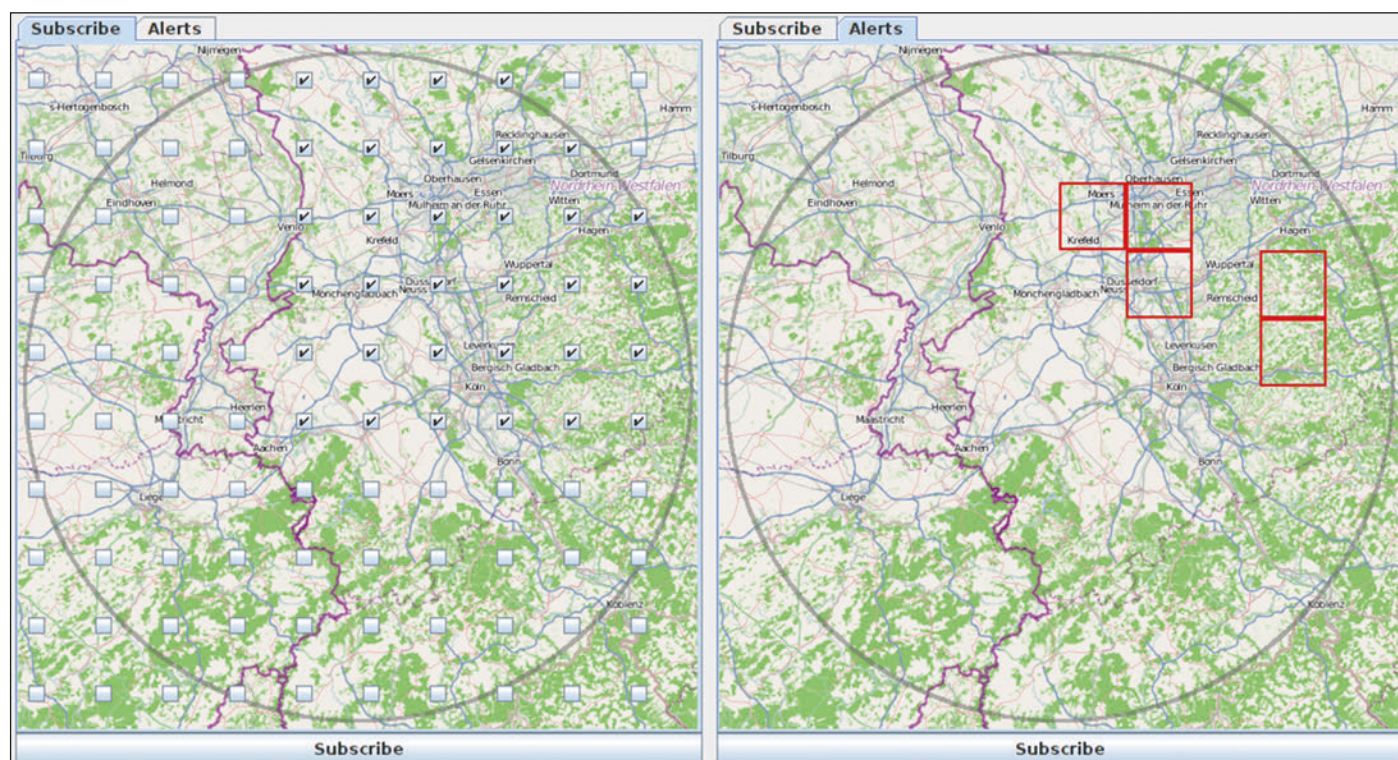


Abbildung 7: Grafische Benutzerschnittstelle der Anwendung

extrahieren müssen, wurden Softwarekomponenten konzipiert und implementiert, mit denen Rasterdaten nach geographischen oder thematischen Gesichtspunkten selektiert werden können. Die von der SOS-Spezifikation definierten vektordatenbasierten Filter mussten hierzu auf Rasterdaten übertragen werden. Die größte Problematik bestand darin, einen Kompromiss zwischen der Datenspeicherung und einer effizienten Suche in den Rasterdaten zu finden. Einerseits musste es möglich sein, für eine effiziente Suche nach thematischen Informationen in den Rasterdaten Datenbankindizes zu nutzen. Andererseits ist es wegen der Anzahl der Rastermaschen eines Datensatzes nicht möglich, jede Rastermasche einzeln in einer Zeile einer Tabelle abzuspeichern. Eine Lösung wurde zum einen durch ein in seiner Größe frei wählbares, gröberes Raster und zum anderen durch die Speicherung der Rasterdaten in einem binären Feld (BLOB) der Datenbank erzielt. Die Auflösung des groben Rasters wurde dabei so gewählt, dass jede Rastermasche als Zeile in eine Datenbank eingetragen und damit ein Datenbankindex für die Suche benutzt werden kann. Hierzu wurde ein effizienter Sweep-Algorithmus entwickelt, der aus den binär gespeicherten Rasterdaten Bereiche ausliest, die durch das grobe Gitter selektiert wurden.

Für die Rückgabe der Daten lässt die SOS-Spezifikation verschiedene Modelle zu, die sich hinsichtlich der Genauigkeit und Ausdrucksstärke erheblich unterscheiden. Im Hinblick auf diese beiden Eigenschaften stellt das *DiscreteCoverageObservation*-Modell das genaueste und ausdrucksstärkste Modell für die Beschreibung von Rasterdaten dar und wurde deshalb auch für die Rückgabe von gefilterten Bereichen von Rasterdaten benutzt. Mit steigender Genauigkeit erhöht sich aber auch die Größe der Dokumente, die das Modell fasst. Es ist aus diesem Grund nicht möglich, eine größere Anzahl von Datensätzen mithilfe dieses Modells vom Dienst abzurufen. Deshalb wurde ein zweites Modell dazu benutzt, Daten als kommaseparierte Werte auszuliefern. Die Größe der mit diesem Modell codierten Dokumente ist um ca. einen Faktor 100 geringer.

Um dies zu überprüfen, wurden 13 Szenarien betrachtet und hinsichtlich ihres Antwortzeitverhaltens verglichen. Die Ergebnisse sind in Tabelle 1 zusammengestellt. Die ersten sieben Szenarien rufen vollständige Rasterdatensätze vom SOS ab (siehe Tabelle 1, Zeilen 1 bis 7). Um einen Datensatz in das *DiscreteCoverageObservation* O&M-Modell zu codieren, brauchte das System über zehn Minuten (siehe Tabelle 1, Zeile 1). Die Größe der

erzeugten Datei belief sich dabei auf über 600 MB für nur einen einzigen Rasterdatensatz (640.000 Rastermaschen). Durch die Rückgabe der Daten im generischen O&M-Modell konnte die Größe des zurückgegebenen Dokuments auf 7 MB und die Antwortzeit auf 1.8 Sekunden reduziert werden (siehe Tabelle 1, Zeile 2). Schließlich konnte durch die gzip-Komprimierung die Größe nochmals auf 1 MB verkleinert werden (siehe Tabelle 1, Zeile 3). Neben Einzelrastern ist es mithilfe des generischen Modells auch möglich, eine Zeitreihe von kompletten Datensätzen abzurufen (siehe Tabelle 1, Zeilen 4 bis 7). Die Komprimierung zahlt sich dabei vor allem beim Zugriff über das Internet aus (siehe Tabelle 1, Spalte „Netz“), da dabei aufgrund der geringeren Übertragungsgeschwindigkeit die Zeit für das Komprimieren durch die geringere Größe der zu übertragenden Daten kompensiert wird (siehe Tabelle 1, Zeilen 6 und 7). In den Szenarien acht bis zehn wurden jeweils 60 räumlich und thematisch gefilterte Datensätze abgerufen. Die letzten beiden Szenarien erfragten vom Raster-SOS im *TimeSeriesObservation*-Modell codierte Zeitreihen. Es wurden 60 Rasterdatensätze selektiert, wobei bei Szenario 12 der Wert einer Rastermasche und in Szenario 13 der Mittelwert einer rechteckigen Region zurückgegeben wurde.

Aus Szenario 1 folgt, dass das vom SOS-Standard zum Codieren von Rasterdaten vorgesehene *om:DiscreteCoverageObservation*-Modell nicht praktikabel ist. Zwar können hierin die Rasterdaten sehr genau beschrieben werden, was der Interoperabilität zugute kommt, doch ist das dadurch bedingte immense Datenvolumen nicht akzeptabel. Hingegen lassen sich die Rasterdaten mit dem generischen Modell in akzeptabler Zeit und Größe ausliefern (vgl. Tabelle 1, Szenario 4 bis 7). Die Interoperabilität ist auch hier durch die Beschreibung der kommaseparierten Felder gegeben. In der Praxis werden ohnehin auch bei vektorbasierten SOS-Implementierungen die Daten mittels des generischen Modells übergeben und nicht durch die auch hier vorhandenen wortreicheren detaillierteren Modelle, wie beispielsweise dem *om:Measurement* Modell (OGC 2007b) (Broering & Meyer 2008).

Trotz des unterschiedlichen Antwortverhaltens der Modelle *DiscreteCoverageOb-*

*servation* und *TimeSeriesObservation* sei darauf hingewiesen, dass die Ausdruckstärke und die Genauigkeit, aber auch der Speicherbedarf der beiden Modelle identisch sind. Das gute Abschneiden des *TimeSeriesObservation*-Modell liegt darin begründet, dass immer nur eine Rastermasche bzw. der Mittelwert einer Region für einen Datensatz im Modell codiert wird. Allerdings eröffnet dieses Modell die Möglichkeit, Mittelwerte beispielsweise von Einzugsgebieten abzufragen, was sich in einer *GetObservation*-Anfrage nicht spezifizieren ließe.

Die *out-of-band*-Methode eröffnet Anwendungen den Zugang zu Rasterzeitreihendaten, die von externen Diensten bereitgestellt werden. Für die hier beschriebene Anwendung zur Visualisierung von Wetterradardaten war es ausreichend und aus Performanzaspekten notwendig, die Daten codiert als georeferenzierte Farbbilder bereitzustellen. Darüber hinaus ist es möglich, die Rasterdaten mit der hier beschriebenen

Methode, aber nicht mittels eines WMS, sondern durch einen Web Coverage Service (WCS) (OGC 2003), bereitzustellen, der tatsächliche Datenwerte liefert (Nengcheng et al. 2009). Eine interessante Frage dabei ist, wie die in einer *GetObservation*-Anfrage angegebenen räumlichen Filter auf eine WCS *GetCoverage*-Anfrage umgesetzt werden können. Ein möglicher Ansatz dazu ist, die *GetCoverage*-Anfrage direkt vom SOS ausführen zu lassen. In diesem Fall würde der SOS als Proxy zwischen Client und WCS fungieren. Sollen sowohl Rückgabe per WMS als auch WCS unterstützt werden, muss auch eine Möglichkeit bestehen, dies in einer *GetObservation*-Anfrage zu unterscheiden. Da beide Varianten den Anfrageparameter *requestMode* auf „out-of-band“ setzen müssen, bleibt hierfür nur der *responseFormat*-Parameter. Eine einheitliche Spezifizierung der möglichen Werte für den *responseFormat*-Parameter, die beide Fälle (WCS, WMS) berücksichtigt, wäre deshalb wünschenswert.

Szenario	Anzahl	Rasterzellen	Filter	Modell	Mode	Format	Netz	Zeit in s	Größe in MB
1	1	640.000	–	Om:Discrete-CoverageObservation	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	1	40.000	Räumlich	Om:Observation	Inline	Plain	Intranet	0.2	0.4
9	60	40.000	Räumlich	Om:Observation	Inline	Plain	Intranet	12	28
10	60	250	Thematisch	Om:Observation	Inline	Plain	Intranet	26	0.5
11	60	1	Räumlich (Rechteck)	Om:TimeSeries-Observation	Inline	Plain	Intranet	6	0.02
12	60	1	Räumlich (Punkt)	Om:TimeSeries-Observation	Inline	Plain	Intranet	0.3	0.03
13	60	–	–	Om:Observation	Out-of-band	Plain	Internet	0.1	0.1

Tabelle 1: Untersuchte Szenarien zur Analyse der Rückgabezeiten des SOS



Für die Signalisierung der durch die Anwendung von Filtern detektierten Starkregengebiete wurde ein Softwaresystem implementiert, das einerseits einen Sensor Alarm Service (SAS) mit den Alarmanträgen der detektierten Starkregengebiete bedient. Andererseits wird eine grafische

Benutzerschnittstelle für die clientseitigen Aktionen bereitstellt. Aus Gründen der Wiederverwendbarkeit der entwickelten Softwarekomponenten wurde neben einer konkreten Anwendung, die Starkregengebiete detektiert, auch ein *Framework* geschaffen, das die Alarmanträge mittels des XMPP-

Protokolls an den SAS übermittelt. Die entwickelte grafische Benutzerschnittstelle bietet auf Basis von Java-Swing-Komponenten eine einfache und benutzerfreundliche Möglichkeit, eine Registrierung für Alarmanträge durchzuführen und eingetretene Starkregenereignisse anzuzeigen.

## Literatur

- Botts, M. et al. (2008): OGC Sensor Web Enablement: Overview and High Level Architecture. In: Nittel, S.; Labrinidis, A.; Stefanidis, A. (Eds.): *Advances in GeoSensor Networks. Lecture Notes in Computer Science*, 4540. Springer Verlag, Berlin/Heidelberg/New York, S. 175-190.
- Bröring, A.; Meyer, O. (2008): Bereitstellung und Visualisierung hydrologischer Zeitreihen mit Hilfe standardisierter Webdienste. In: Strobl, J. et al. (Hrsg.): *Angewandte Geoinformatik 2008*. Wichmann Verlag, Heidelberg, S. 354-363.
- Bröring, A. et al. (2011): New Generation Sensor Web Enablement. In: *Sensors*, 11 (3), S. 2652-2699.
- Bronstert, A. et al. (2009): Interactions and feedbacks in hydrological change: Relevance and possibilities of modelling. In: *Hydrologie und Wasserbewirtschaftung*, 53 (5), S. 289-304.
- Güting, R. H.; Dieker, S. (2004): *Datenstrukturen und Algorithmen*. Teubner Verlag, Stuttgart.
- Hanson, R.; Tacy, A. (2007): *GWT im Einsatz: AJAX-Anwendungen entwickeln mit dem Google Web Toolkit*. Hanser Verlag, München.
- Holl, S. (2012): *PostGIS Raster Workshop. FOSSGIS 2012*, Dessau-Rosslau.
- IETF (1996): Request for Comments: 1952 – GZIP file format specification version 4.3. Internet Engineering Task Force.
- IETF (2011): Extensible Messaging and Presence Protocol (XMPP): Core. Internet Engineering Task Force.
- ISO (2005): *Geoinformation – Coverage Geometry- und Funktionsschema (DIN EN ISO 19123)*. Beuth Verlag, Berlin.
- Kemper, A.; Eickler, A. (2006): *Datenbanksysteme – eine Einführung*. Oldenbourg Verlag, München.
- Kirschke, T. (2012): Implementierung eines OGC-konformen WMS als Plugin für einen Geoserver. Bachelorarbeit, FH-Aachen.
- Loy, M. et al. (2002): *Java Swing*. O'Reilly, Sebastopol, CA.
- Marshall, J. S.; Palmer, W. M. (1948): The distribution of raindrops with size. In: *Journal of Meteorology*, 5, S. 165-166.
- Nengcheng, C. et al. (2009): A flexible geospatial sensor observation service for diverse sensor data based on Web service. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 64 (2), S. 234-242.
- Obe, R. O.; Hsu, L. S. (2011): *PostGIS in Action*. Manning Publications, Shelter Island, NY.
- OGC (2003): *Web Coverage Service (WCS)*. Open Geospatial Consortium, Inc., Wayland, MA.
- OGC (2006a): *OpenGIS Sensor Observation Service Implementation Specification*. Open Geospatial Consortium, Wayland, MA.
- OGC (2006b): *OpenGIS Web Map Server Implementation Specification*. Open Geospatial Consortium, Wayland, MA.
- OGC (2007a): *Observations and Measurements. Part 2: Sampling Features*. Open Geospatial Consortium, Wayland, MA.
- OGC (2007b): *Observations and Measurements. Part 1: Observation schema*. Open Geospatial Consortium, Wayland, MA.
- OGC (2007c): *OGC Sensor Alert Service Implementation Specification*. Open Geospatial Consortium, Wayland, MA.
- OGC (2007d): *OpenGIS Sensor Model Language (SensorML). Implementation Specification*. Open Geospatial Consortium, Wayland, MA.
- OGC (2008): *OpenGIS Sensor Event Service Interface Specification (proposed)*. Open Geospatial Consortium, Wayland, MA.
- OGC (2010a): *Geographic Information: Observations and Measurements. OGC Abstract Specification Topic 20*. Open Geospatial Consortium, Wayland, MA.
- OGC (2010b): *OGC WCS 2.0 Interface Standard – Core*. Open Geospatial Consortium, Wayland, MA.
- OGC (2011a): *Observations and Measurements. XML Implementation*. Open Geospatial Consortium, Wayland, MA.
- OGC (2011b): *OGC Web Coverage Service 2.0 Interface Standard – Earth Observation Application Profile*. Open Geospatial Consortium, Wayland, MA.
- OGC (2012): *OGC Sensor Observation Service Interface Standard*. Open Geospatial Consortium, Wayland, MA.
- OpenStreetMap contributors (2012): *OpenStreetMap*. <http://www.openstreetmap.org>.
- Pfeiffer, T.; Wenk, A. (2010): *PostgreSQL – das Praxisbuch*. Galileo-Press, Bonn.
- The PostgreSQL Global Development Group (2012): *PostgreSQL JDBC Driver*. <http://jdbc.postgresql.org/>.
- Walls, C. (2011): *Spring in Action*. Manning Publications, Shelter Island, NY.
- Zacharias, S. et al. (2011): A Network of Terrestrial Environmental Observatories in Germany. In: *Vadose Zone Journal*, 10 (3), S. 955-973.