

Bereitstellung und Visualisierung hydrologischer Zeitreihen mit Hilfe standardisierter Webdienste

Arne BRÖRING und Oliver MEYER

1 Zusammenfassung

Die Technologien der Sensor Web Enablement (SWE) Initiative des Open Geospatial Consortium (OGC) ebnen den Weg für einen interoperablen Zugang zu den von Sensornetzwerken erfassten Geodaten. Dieser Artikel beschreibt die erste Ausbaustufe der für den Wupperverband umgesetzten SWE Infrastruktur. Diese umfasst die Realisierung eines Sensor Observation Service (SOS), der aktuelle Zeitreihen, nach ihrer Prüfung durch eine Fachabteilung, über das Internet anbietet. Neben diesem Web Service wird die Entwicklung einer Client Applikation beschrieben, die dem Anwender das Abrufen und Visualisieren ausgewählter Zeit- und Messreihen ermöglicht.

2 Einleitung

Das Ziel des Wupperverbandes ist es, die wasserwirtschaftlichen Aufgaben im Einzugsgebiet der Wupper zu erfüllen (WUPPERVERBAND 1992). Zur Durchführung der Aufgaben des Flussgebietsmanagements wird der Zugang zu erfassten Sensordaten benötigt. Wasserwirtschaftlich relevante Kennwerte sind hier beispielsweise Pegel- oder Niederschlagsdaten. Den Entscheidungsträgern müssen Zeit- und Messreihen zur Verfügung stehen, um beispielsweise Hochwasserereignisse einschätzen zu können. Diese raumzeitvariante Information muss analysiert und bewertet werden, um angemessen reagieren zu können.

Der Zugang zu Sensordaten kann interoperabel durch die vom Open Geospatial Consortium (OGC) bereitgestellten Techniken der Sensor Web Enablement (SWE) Initiative geleistet werden. Während Sensornetzwerke bislang heterogene Datenformate und Kommunikationsprotokolle verwendet haben, hat die SWE Initiative die Realisierung des Sensor Webs zum Ziel. Durch die Schaffung eines Frameworks von Web Service Spezifikationen und assoziierten Beschreibungssprachen werden Sensornetzwerke über das Internet einheitlich nutzbar. Die Definition aufeinander abgestimmter Standards ermöglicht die Integration raumzeitlicher Sensordaten in Geodateninfrastrukturen. Sensoren und Sensorsysteme werden interoperabel über das Web auffindbar, zugreifbar, und wenn möglich auch kontrollierbar (BOTTS, PERCIVALL, REED & DAVIDSON 2006).

Für den Wupperverband bietet die Anwendung dieser Technologien den Vorteil, dass sich die SWE Services nahtlos in die bereits existierende Dienstarchitektur OGC-konformer Web Services integrieren lassen. Die vom Wupperverband erfassten Pegel- und Niederschlagsdaten werden somit standardisiert über das Internet zur Verfügung gestellt. Applikationen des Wupperverbandes können aufgrund der einheitlichen Schnittstellen sowohl auf interne als auch auf externe Dienste zugreifen. Dies macht es möglich, Sensordaten kooperierender Organisationen in den Entscheidungsprozess mit einzubeziehen.

Die im Rahmen des hier vorgestellten Projekts entstandenen Implementierungen wurden von der SWE Arbeitsgruppe der 52°North Initiative (KRAAK, SLIWINSKI & WYTZISK 2005)

durchgeführt. Als Open Source Lösungen stehen sie daher auch anderen Projekten frei zur Verfügung.

Im Folgenden werden zunächst die für die Applikation relevanten Spezifikationen der SWE Initiative geschildert. Abschnitt 4 stellt das Konzept der für diesen Anwendungsfall realisierten SOS Implementierung vor. Im Anschluss an die Beschreibung der Server Komponente werden in Abschnitt 5 die Eigenschaften der Client Applikation beleuchtet.

3 Grundlagen

3.1 SensorML

Die Spezifikation der Sensor Model Language (SensorML) (BOTTS & ROBIN 2007) definiert ein Modell sowie dessen XML-Kodierung zur Beschreibung von Sensoren und Sensorsystemen. Der vollständige vom Sensor durchgeführte Messprozess oder auch die Weiterverarbeitung erfasster Daten im Sinne einer Post-Prozessierung kann beschrieben werden.

Es können zahlreiche Metadaten definiert werden. Hierzu zählen Angaben zur Identifikation und Klassifikation aber auch Charakteristiken wie die zeitliche Verfügbarkeit oder die räumliche Beschreibung eines Sensorsystems.

3.2 O&M

Die Spezifikation Observations and Measurements (O&M) (COX 2007) definiert ein domänenunabhängiges, konzeptionelles Modell zur Repräsentation der Resultate von Messbeobachtungen. Dazu stellt die O&M-Spezifikation XML-Schemata bereit, die eine XML-basierte Kodierung raumzeitvarianter Messdaten ermöglichen. Das O&M-Format wird insbesondere für den Aufbau der Antwortdokumente der GetObservation-Operation des Sensor Observation Service benötigt. Es kann aber auch als generelles Hilfsmittel für den standardisierten Umgang mit technischen Messungen benutzt werden.

3.3 SOS

Aufgabe der Sensor Observation Service (SOS) Spezifikation (NA & PRIEST 2006) ist die Beschreibung einer Web Service Schnittstelle, die den standardisierten Zugang zu den von Sensoren erfassten Daten möglich macht. Der Service dient dabei als Mittler zwischen einem Client und einem Sensordatenarchiv beziehungsweise einem Echtzeit-Sensorsystem. Die unterschiedlichen Kommunikationsprotokolle, Datenformate und -quellen sowie Standards der verschiedenen Sensorsysteme verbirgt der SOS hinter seiner standardisierten Schnittstelle. Vom Client angefragte Sensordaten werden gemäß der O&M-Spezifikation in Form von Observations kodiert, Metadaten zu Sensoren als SensorML.

4 Bereitstellung der Daten

4.1 Datenerhebung und -archivierung

Der Wupperverband verwaltet ein Netzwerk von zahlreichen Sensoren. Zum einen werden hydrologische Messwerte wie Wasserstände oder Durchflüsse an Talsperren und Flüssen

erfasst. Darüber hinaus werden meteorologische Kennwerte gemessen, beispielsweise Niederschlagsmengen. Über verschiedene Kommunikationskanäle (zum Beispiel DSL, GSM, manuelle Auslesung) werden die Messwerte abgerufen und zunächst zentral gesammelt. Bei diesen Werten handelt es sich in der Regel um numerische Größen. Gemäß ihrer Art liegen sie in unterschiedlichen Wertebereichen und gehören zu unterschiedlichen Skalen. Nach einer eventuellen Vorverarbeitung und Plausibilitätsprüfung werden die Daten in einer Datenbank archiviert. Dabei werden sie mit den zugehörigen Phänomenen, Sensoren und Stationen assoziiert. Ebenfalls wird eine entsprechende raumbezogene Information zu den Messstationen gespeichert.

4.2 Architektur und Implementierung des SOS

Die Architektur des SOS ist in drei Schichten organisiert (siehe Abbildung 1). Die unterste Ebene stellt die Datenschicht dar. Die SOS-Implementierung definiert hier abstrakte Schnittstellen für den Zugriff auf das darunterliegende Datenbankmanagementsystem. Die höheren Ebenen der Architektur bleiben somit unabhängig von der Datenbank. Implementierungen dieser Schnittstellen dienen als Plugins (GAMMA, HELM, JOHNSON & VLISSIDES 1995). Sie kann somit einfach für unterschiedliche Datenquellen angepasst werden. Im Falle des SOS für den Wupperverband wird der Zugriff auf die Daten durch ein relationales Datenbanksystem mit raumbezogener Erweiterung ermöglicht, um auch geometrische Anfragen bearbeiten zu können.

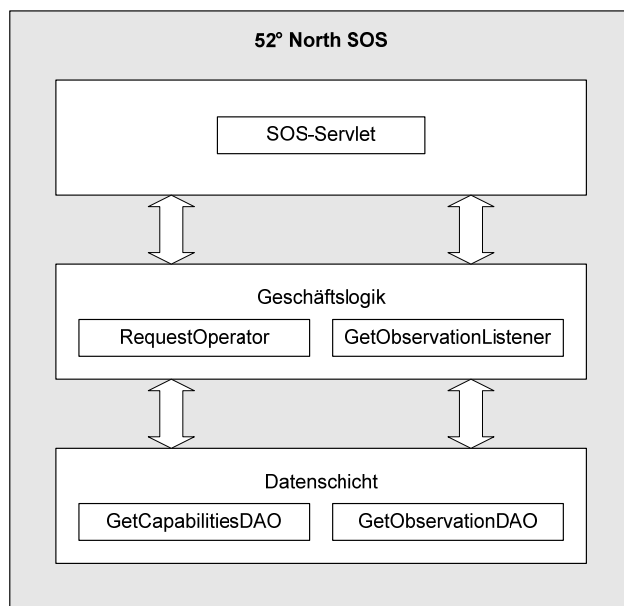


Abb. 1: SOS Architektur

Darauf aufbauend ist die Geschäftslogik realisiert. Die zentrale Komponente ist hier der Request Operator, der die Anfragen an den Dienst entgegennimmt, validiert und an den korrespondierenden Listener weiterreicht. Jede vom SOS unterstützte Operation ist durch

einen eigenen Listener realisiert, die wiederum alle eine gemeinsame Schnittstelle implementieren. Über die Konfiguration des Dienstes können die einzelnen Listener aktiv gesetzt werden. Darüber hinaus finden sich in dieser Schicht weitere Komponenten zum Parsen der eingehenden Anfragen und Enkodieren der Antworten.

In der obersten Ebene nimmt das SOS-Servlet die HTTP Anfragen des Clients entgegen und leitet sie an den Request Operator weiter.

Der SOS des Wupperverbands verfügt darüber hinaus über eine Schnittstelle, mit der verschiedene Generalisierungsmechanismen für numerische Messwerte umgesetzt werden können. Der verwendete Algorithmus dünnt Zeitreihen abhängig von der angefragten Zeitspanne aus. Je größer der Anfragezeitraum, desto höher wird ein Schwellenwert gesetzt, unterhalb dessen die Werte nicht in die Ergebnismenge aufgenommen werden. Der Schwellenwert bezieht sich immer auf die Spannweite der Werte im Anfragezeitraum. Unterschreiten Zwischenwerte in einem kontinuierlich ausgedehnten Intervall den Schwellenwert, werden sie entfernt. Bei starken Änderungen im Kurvenverlauf werden so Stützpunkte gesetzt. Dadurch ist gewährleistet, dass die Charakteristik der Zeitreihe und somit lokale Mini- und Maxima erhalten bleiben.

4.3 Kodierung der Daten

Bei der Bereitstellung der Daten wird zwischen Metainformation zu Sensoren und Messstationen und den eigentlichen Messergebnissen unterschieden.

Die `GetFeatureOfInterest`-Operation liefert Metainformation zu einem oder mehreren Messstationen. Diese beinhaltet neben den geografischen Koordinaten die zur jeweiligen Station zugehörigen Sensoren sowie deren Phänomene.

Über die `DescribeSensor`-Operation kann der Client Metadaten zu einem Sensor abrufen. Diese werden als `SensorML` kodiert. Listing 1 zeigt einen Ausschnitt eines solchen Dokuments. Die Inhalte der Antwortdokumente sind frei konfigurierbar und werden global für einen Sensortyp in einer Vorlage festgelegt. Zur Laufzeit wird diese dann mit der entsprechenden Information für den angefragten Sensor gefüllt. Bestimmte Abschnitte von `SensorML`-Dokumenten können von Katalogdiensten indiziert werden, um Sensoren einfacher auffindbar zu machen.

Messergebnisse werden über die `GetObservation`-Operation abgerufen. Dabei können verschiedene Parameter definiert werden, die die zurückzugebende Wertemenge einschränken. Ein Beispiel für ein im O&M-Format kodiertes Antwortdokument findet sich in Listing 2. Eine Observation beginnt stets mit einer Beschreibung der Kodierung der Ergebniswerte. Die Messergebnisse finden sich dann entsprechend dieser Beschreibung formatiert im Ergebnis-Teil.

```
<identification>
  <IdentifierList>
    <identifier name="stationName">
      <Term definition="urn:ogc:identifier:stationName">
        <value>Beyenburg</value>
      </Term>
    </identifier>
    ...
  </IdentifierList>
</identification>
<classification>
  <ClassifierList>
    <classifier name="sensorType">
      <Term definition="urn:sensor:classifier:sensorType">
        <value>Wasserstandmesser</value>
      </Term>
    </classifier>
    ...
  </ClassifierList>
</classification>
```

Listing 1: SensorML-Auszug

```
<om:Observation>
  <om:resultDefinition>
    <swe:components>
      <swe:field>
        <swe:Time definition="urn:ogc:data:time:iso8601"/>
      </swe:field>
      <swe:field>
        <swe:Text definition="urn:ogc:data:feature"/>
      </swe:field>
      <swe:field>
        <swe:Quantity definition="Wasserstand"/>
      </swe:field>
    </swe:components>
    <swe:encoding>
      <swe:TextBlock decimalSep="." tokenSep="," blockSep="|"/>
    </swe:encoding>
  </om:resultDefinition>
  <om:result>
    2007-02-02T06:45:00+01, Pegel_Beyenburg, 4.30|
    2007-02-02T07:00:00+01, Pegel_Beyenburg, 4.34|
    2007-02-02T07:15:00+01, Pegel_Beyenburg, 4.37
  </om:result>
</om:Observation>
```

Listing 2: O&M-Auszug

5 Visualisierung der Daten

5.1 Anforderungen

Das typische Anwendungsgebiet der Client Applikation ist die Präsentation der erfassten Sensordaten zu informativen Zwecken. Für den Anwendungsfall des Wupperverbands kann das Programm beispielsweise als flexible Informationsquelle bei der Gewässermodellierung, für Planungszwecke oder zur Überwachung von Wasserständen im Flusssystem und an Talsperren dienen sowie für die Öffentlichkeitsarbeit eingesetzt werden.

Aus diesen möglichen Anwendungsgebieten der Applikation ergeben sich im Wesentlichen zwei Nutzertypen: Neben dem erfahrenen Fachanwender, der den Client bei der täglichen Arbeit einsetzt, soll die Applikation auch vom Laien verwendet werden können, der Interesse an der darstellbaren Information besitzt.

Bei der Entwicklung der Bedienoberfläche werden die technischen Details des zugrundeliegenden Systems, wie die SWE Dienste und Datenformate, verborgen. Für beide Nutzergruppen sollen keine Erfahrungen mit diesen Technologien vorausgesetzt werden und es wird versucht den Lernaufwand möglichst gering zu halten.

Hauptaugenmerk bei der Entwicklung des Clients ist, ein hohes Maß an Funktionalität bei maximaler Plattformunabhängigkeit zu erreichen. Die technischen Voraussetzungen des Client-Rechners zum Ausführen der Applikation sollen möglichst gering gehalten werden. Die Applikation wird daher als webbasierter Thin-Client¹ umgesetzt. Das heißt der Client ist in einem herkömmlichen Browser ausführbar. Somit wird einem breiten Nutzerkreis ein einfacher Zugang ermöglicht.

Nachdem der Nutzer den Client aufgerufen hat, kann er über Eingabemasken beliebige SOS-Dienste ansprechen und deren angebotene Daten abfragen. Die empfangenen Sensordaten können dann in Form eines Diagramms oder einer Tabelle dargestellt werden. Im Diagramm sollen die Daten mehrerer Messgrößen verschiedener Phänomene, die an unterschiedlichen Messstellen aufgenommen wurden, kombinierbar sein. Eine Legende soll Informationen zu Farb- und Schraffurkodierungen der dargestellten Messreihen geben und zudem Interaktionsmöglichkeiten zur Anpassung dieser bieten. Weiterhin soll der Client Eingabeelemente zur Konfiguration der betrachteten Zeitspanne bieten, beispielsweise das Verschieben, Vergrößern und Verkleinern des zeitlichen Ausschnitts.

Neben den von den Sensoren erfassten Daten muss der Client auch die von einem SOS bereitgestellten Metadaten einzelner Sensoren abrufen und dem Nutzer in ansprechender Form präsentieren können.

Darüber hinaus soll der Client den Nutzer befähigen, die von ihm konfigurierte Darstellung von Messreihen zu exportieren, das heißt die aktuelle Ansicht der Daten in einer lokalen Datei eines herkömmlichen Formats (zum Beispiel PDF) abzuspeichern.

¹ Eine Thin-Client Applikation wird in einem Webbrowser ausgeführt. Die Präsentationskomponente einer solchen Applikation teilt sich in Server und Client auf. Der Client-Code wird vom Browser interpretiert und über die Web Schnittstelle des Servers bezogen. In Kontrast dazu führt ein Thick-Client (auch Rich- oder Fat-Client) die komplette Präsentation auf dem Client-Rechner aus (FOWLER, RICE & FOEMMEL 2003).

5.2 Architektur und Implementierung des Clients

Um die Applikation als Thin-Client mit hohem Funktionsumfang und großer Bedienfreundlichkeit zu entwickeln, wird die Konzeption der Architektur des Systems auf die Nutzung der AJAX-Technologie optimiert². Durch den Einsatz von AJAX können Webanwendungen auf Eingaben des Nutzers schneller und dynamischer reagieren.

Um die beschriebenen Anforderungen zu erfüllen und die Vorteile der asynchronen Kommunikation zwischen Client und Server optimal ausnutzen zu können, ist die Architektur des Gesamtsystems in Komponenten separiert. Diese Komponenten sind in distinkten Schichten, gemäß der Konzeption einer Schichten-Architektur (THOMAS 2006), angesiedelt.

In der obersten Ebene, der Präsentations-Schicht, ist der im Browser ausgeführte Client angesiedelt. Dieser greift nicht direkt auf den SOS zu. Stattdessen erfolgen die Datenanfragen des Clients indirekt über einen Fassadendienst (vgl. Abbildung 2).

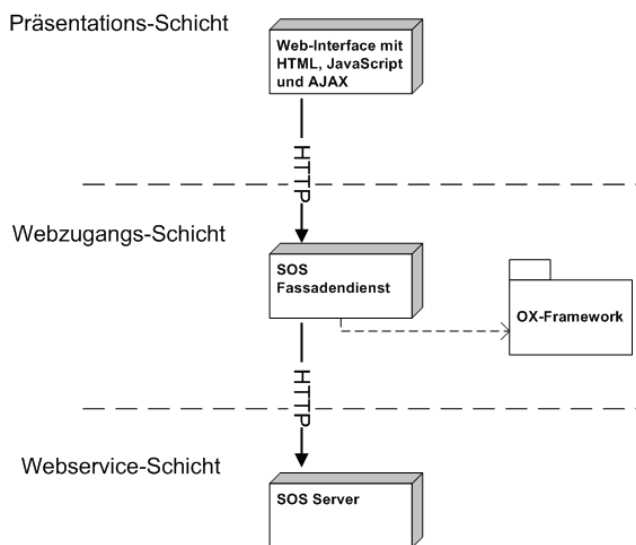


Abb. 2: Architekturschema des Gesamtsystems

Der Fassadendienst realisiert die Webzugangs-Schicht und erzeugt aus der Anfrage des Clients einen spezifikationskonformen GetObservation-Request, der an den SOS gesandt wird.

² Der Begriff AJAX steht für „Asynchronous JavaScript and XML“ und fasst den Einsatz von Web-Technologien zusammen, die eine asynchrone Übertragung von Daten zwischen Web-Server und Client ermöglichen. Es können HTTP-Anfragen durchgeführt werden, ohne dass die komplette Webseite vom Webbrowser neugeladen werden muss. Ein spezielles Browser-Plugin ist zur Ausführung von AJAX nicht notwendig, wodurch die Anwendung vom Betriebssystem unabhängig ist. Allerdings muss der Webbrowser Javascript unterstützen.

Die vom SOS zurückgegebenen und im O&M-Format kodierten Sensordaten werden vom Fassadendienst empfangen. Dieser transformiert die Daten in das kompaktere Format der „JavaScript Object Notation“ (JSON) und sendet sie an den Client. Schließlich generiert der Client aus den JSON-Daten die vom Nutzer gewünschte Repräsentation, wie ein Diagramm oder eine tabellarische Darstellung.

Der Einsatz eines Fassadendienstes wirkt sich positiv auf die Performanz der Applikation aus. Während der Client auf den Fassadendienst über das Internet zugreift, ist die Verknüpfung von Fassadendienst und SOS hingegen enger. Im operativen Einsatz ist dies eine Netzwerkverbindung mit hoher Bandbreite. Somit wird das wortreiche Datenformat O&M lediglich über die schnelle Verbindung ausgetauscht. An den Browser wird über das Internet nur das kompaktere JSON-Format gesendet. Für andere Client Applikationen, welche die applikationsspezifische JSON-Kodierung nicht interpretieren können, stehen die Sensordaten trotzdem standardisiert über einen direkten Zugriff auf den SOS zur Verfügung.

Ein weiterer Vorteil dieser Vorgehensweise ist, dass das aufwendige Parsen der O&M-Daten nicht im Browser geschieht. Der auf JavaScript basierende Client kann das JSON-Format auf einfache Weise interpretieren und daraus beispielsweise ein Diagramm erzeugen. Dieses clientseitige Rendern der Daten entlastet wiederum den Server und sorgt für eine erhöhte Reaktionsgeschwindigkeit der Anwendung.

Zur Umsetzung des Fassadendienstes wird das im Rahmen der 52°North Initiative entwickelte OX-Framework (BRÖRING, FÖRSTER & SIMONIS 2006) verwendet. Dieses kann im Sinne von BALZERT 2001 als Softwareframework verstanden werden. Entwickler können es in verschiedenen Applikationen nutzen, um auf beliebige OGC Web Services zuzugreifen und empfangene Daten zu visualisieren oder in anderer Form zu verarbeiten. Für die hier entwickelte Anwendung wird das OX-Framework zum Zugriff auf den SOS, zum Parsen der empfangenen O&M-Daten und schließlich zur Erzeugung der JSON-Objekte genutzt.

Neben der Darstellung der gemessenen Sensordaten als Diagramm oder Tabelle, erfordert der Anwendungsfall ebenfalls die menschenlesbare Anzeige der Metadaten eines Sensors. Das vom SOS zurückgegebene SensorML-Dokument muss dazu in ein HTML-Dokument überführt werden. Hierzu wird von XSL Transformations (XSLT) Gebrauch gemacht³.

Der zentrale Vorteil, der sich aus der Verwendung von XSLT ergibt, ist die Kapselung der Darstellungsvorschriften in einer separaten Datei. Dadurch kann die Formatierung der Ausgabe auf einfache Weise editiert und eventuell für andere Anwendungsfälle angepasst werden. Aus diesem Grund wird auch bei der Umsetzung der Exportfunktionalität der Applikation, also beispielsweise die Erzeugung von PDF-Dokumenten, auf die XSLT-Technologie zurückgegriffen.

³ XSLT ist Teil der eXtensible Stylesheet Language (XSL) und kann zur Transformation von XML-Dokumenten angewendet werden. Dazu werden in einem XSLT-Skript Regeln zur Überführung des XML-Codes in eine andere Repräsentation definiert. Ein spezieller XSLT-Prozessor wird verwendet, um ein XML-Dokument gemäß den spezifizierten Regeln in das gewünschte Ausgabeformat zu transformieren.

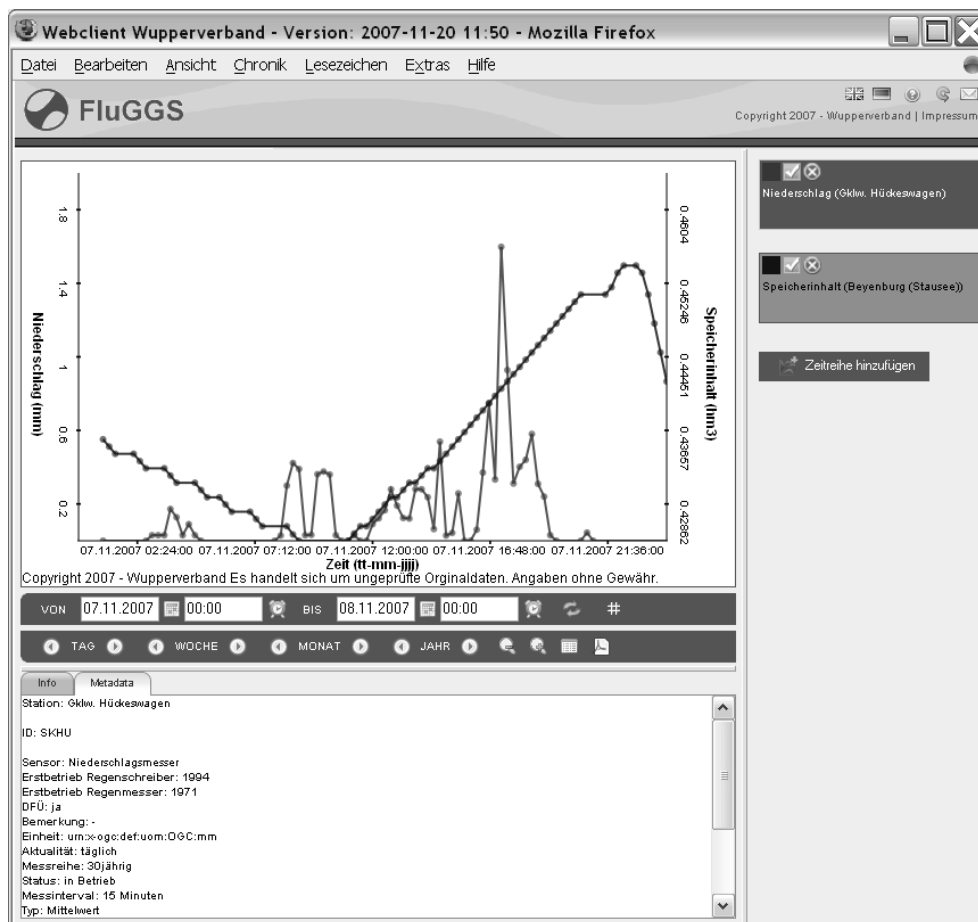


Abb. 3: Client Applikation im Browser

Abbildung 3 zeigt die im Webbrowser aufgerufene Client Applikation. Im Mittelpunkt ist die Diagramm-Ansicht mit dem darunterliegenden Bedienfeld zur Manipulation der Zeitspanne angeordnet. Rechts befindet sich die interaktive Legende, mit deren Hilfe die Zeitreihen ausgewählt, konfiguriert, hinzugefügt oder entfernt werden können. Selektiert der Nutzer eine Zeitreihe in der Legende, so erscheint unterhalb des Diagramms die Darstellung der Metadaten des assoziierten Sensors.

6 Fazit und Ausblick

Im Rahmen des vorgestellten Projekts wurde mit Hilfe der SWE Technologie erfolgreich ein System zur standardisierten Bereitstellung und Visualisierung von Sensordaten entwickelt. Die realisierten Komponenten sind dabei nicht auf die Hydrologie und das vorliegende Beispiel des Wupperverbands beschränkt. In Zukunft können sie auf einfache Weise in anderen Anwendungsfällen adaptiert werden und beispielsweise in Domänen wie der Logistik oder dem Umweltmonitoring zum Einsatz kommen. Die Visualisierungskomponente

unterstützt den Anwender dabei, mögliche Zusammenhänge oder Anomalien innerhalb der erfassten Phänomene zu entdecken.

Für den Wupperverband wird in einer nächsten Stufe die existierende SWE Infrastruktur weiter ausgebaut. Ein Sensor Planning Service (SPS) (SIMONIS 2007) wird künftig eine standardisierte Schnittstelle zur Steuerung ausgewählter Sensoren bieten. Weiterhin soll ein Sensor Alert Service (SAS) (SIMONIS & ECHTERHOFF 2007) eingesetzt werden, um registrierte Nutzer in vordefinierten Situationen alarmieren zu können.

Literatur

- Balzert, H. (2001): Lehrbuch der Software-Technik. Software-Entwicklung. 2. Auflage. Heidelberg. Spektrum Akademischer Verlag.
- Botts, M. & A. Robin (2006): Sensor Model Language. OpenGIS Implementation Specification. Open Geospatial Consortium: 07-000, Version: 1.0.
- Botts, M., G. Percivall, C. Reed & J. Davidson (2006): OGC Sensor Web Enablement: Overview And High Level Architecture. OGC White Paper. Open Geospatial Consortium: 06-046r2, Version: 2.0.
- Bröring, A., T. Förster & I. Simonis (2006): An Integrated Software Framework for OGC Web Services. Online unter: <http://www.foss4g2006.org/contributionDisplay.py?contribId=134&sessionId=37&confId=1>. (abgerufen am: 30.01.2008).
- Cox, S. (2007): Observations and Measurements. OpenGIS Implementation Specification. Open Geospatial Consortium: 07-022r1, Version: 1.0.
- Fowler, M., D. Rice & M. Foemmel (2003): Patterns für Enterprise Application-Architekturen. Bonn. mitp-Verlag.
- Gamma, E., R. Helm, R. Johnson & J. Vlissides (1995): Design Patterns. Elements of Reusable Object-Oriented Software. Reading, USA. Addison-Wesley.
- Kraak, M.-J., A. Sliwinski & A. Wytzisk (2005). What happens at 52N? An Open source approach to education and research. Joint ICA commission seminar 6.-8. July 2005, part of the 22nd ICA conference, ICC 2005: Mapping approaches into a changing world, 2005, 16-20.
- Na, A. & M. Priest (2006): Sensor Observation Service. OpenGIS (Draft) Implementation Specification. Open Geospatial Consortium: 06-009r5, Version: 1.0.
- Simonis, I. (2007): OpenGIS Sensor Planning Service Implementation Specification. OpenGIS Implementation Specification. Open Geospatial Consortium: 07-014r3, Version: 1.0.
- Simonis, I. & J. Echterhoff (2007): OGC® Sensor Alert Service Implementation Specification OpenGIS Implementation Specification. Open Geospatial Consortium: 06-028r4, Version: 1.0.0.
- Thomas, S. (2006): Alternativen zu AJAX. In: Java Spektrum, 2006(6). S. 16-18.
- Wupperverband (1992): Gesetz über den Wupperverband. Online unter: [http://www.wupperverband.de/internet/wupperverbandwys.nsf/0/D676C513E1D828F4C125729C004FDA4E/\\$FILE/Wupperverbandsgesetz.pdf](http://www.wupperverband.de/internet/wupperverbandwys.nsf/0/D676C513E1D828F4C125729C004FDA4E/$FILE/Wupperverbandsgesetz.pdf). (abgerufen am: 30.01.2008)