

Flexible webbasierte Visualisierung von Sensordaten mit Lage- und Tiefeninformationen zur Hochwasserfrühwarnung

Stephan JANSEN^a and Thomas BRINKHOFF^b

^a*OFFIS – Institut für Informatik, Oldenburg, Germany*

^b*Jade Hochschule, Oldenburg, Germany*

Kurzfassung. Hochwasserereignisse in kleineren Flusseinzugsgebieten können durch die Beobachtung der Pegelstände nicht ausreichend früh erkannt werden. Um eine Früherkennung in diesen Gebieten zu gewährleisten, wurde im Projekt WEBBOS ein Sensorsystem entwickelt, das Bodenfeuchteprofilmessungen mit Lage- und Tiefeninformationen ermöglicht. Die erfassten Sensordaten werden über die Schnittstellen des Sensor Web bereitgestellt, die in einen Web Client visualisiert und analysiert werden können. Es wurden verschiedene Ansätze für bei der Umsetzung eines Web Clients verfolgt und prototypisch umgesetzt. Eine sehr flexibel einsetzbare Lösung konnte durch Nutzung der Open Source Bibliothek „OpenLayers“ erreicht werden.

Keywords. Sensor Web Enablement, Sensor Observation Service, Sensor Event Service, OpenLayers, Web Client

1. Einführung

Mit der steigenden Zahl der katastrophalen Hochwasserereignisse in den Einzugsbereichen der Flüsse in Deutschland wächst der Bedarf an geeigneten Frühwarnsystemen. Es zeigt sich, dass bei der Hochwasservorhersage und -warnung in kleineren Flusseinzugsgebieten anders als in den großen mesoskaligen Gebieten eine ständige Beobachtung der Pegelstände keine ausreichende Vorwarnzeit gewährleistet werden kann. Bei starken Regenfällen kommt es in diesen Bereichen zu verheerenden Abflüssen, da diese sich sehr schnell aufbauen. Deshalb ist in diesen Regionen die Pegelablesung kein geeignetes Mittel zur Früherkennung eines Ereignisses.

In dem vom BMBF geförderten Projekt “Webbasiertes Sensorsystem zur Bodenfeuchteprofilmessung in der Hochwasserfrühwarnung” (WEBBOS) kooperierten die Hochschule Mannheim und die Jade Hochschule Oldenburg, um ein System zu entwickeln, dass eine Frühwarnung auch für kleinere Flusseinzugsgebiete ermöglicht. Dazu wurde eine Sonde entwickelt, die eine Bodenfeuchtemessung in mehreren Tiefen ermöglicht. Die Sonden sind in Gruppen an eine Funkeinheit angeschlossen, über die erfasste Messwerte an ein Empfangssystem gesendet werden. Die Sonden werden flächenhaft ausgebracht und erfassen so ein dreidimensionales Feuchteprofil des Bodens [1].

Die Bereitstellung der Profildaten wurde mit Zielsetzung der Übertragbarkeit, Skalierbarkeit und Interoperabilität über Web Services realisiert. Gerade im Bereich

der Bereitstellung von Geodaten hat sich die Nutzung von Geoinformationsdiensten (z.B. der Web Map Service (WMS) oder der Web Feature Service (WFS)) in verschiedenartigen Anwendungen bewährt. Für Sensordaten sind allerdings spezielle Geodienste erforderlich, da es sich hier um dynamische, zeitlich veränderliche Daten handelt. Im Rahmen der Sensor Web Enablement Initiative des Open Geospatial Consortiums (OGC) wurden Spezifikationen zur Behandlung von Sensordaten entwickelt: So finden im WEBBOS-Projekt das Observations & Measurements Schema (O&M) zur Kodierung von Sensormessungen [2] und die Sensor Model Language [3] zur Beschreibung von Sensoren Anwendung. Im Bereich der Web Services kommen der Sensor Observation Service (SOS) zur Anfrage und Filterung von Messwerten [4] und der Sensor Event Service (SES) zur Ereignisverarbeitung [5] zum Einsatz. Die Sensor Web Services ergänzen die bestehenden Geoinformationsdienste um den realzeitlichen Zugriff auf Sensordaten.

In den letzten Jahren starteten mehrere andere Projekte, bei denen die Vorteile der Sensor-Web-Dienste insbesondere auch im Bereich der Hochwasserfrühwarnung genutzt werden. Als Beispiel seien hier die Projekte Tsunami Service Bus of the German Indonesian Tsunami Early Warning System (GITEWS) [6] und das Wupperverband SWE Projekt [7] genannt. Vorwiegend kommt in den Projekten der Sensor Alert Service (SAS) zur Warnung zum Einsatz, die Nutzung von SES ist derzeit noch wenig verbreitet (vgl. auch [8]).

Der Beitrag gliedert sich wie folgt: Zunächst wird die Gesamtarchitektur des Systems zur Hochwasserfrühwarnung mit der eingesetzten Sensor-Hardware und den verwendeten Diensten erörtert. Im darauf folgenden Abschnitt werden die ersten Lösungsansätze auf Basis von Standarddiensten und der im Projekt eingesetzte „OpenLayers“-Client vorgestellt. Des Weiteren wird die von der „OpenLayers“-Community bereitgestellte Lösung zur Anbindung von SOS beschreiben. Der Vergleich der Lösungen soll deren unterschiedliche Eigenschaften und Zielsetzungen verdeutlichen. Im Abschnitt 4 wird die Erprobung des Systems an verschiedenen Einsatzorten dargestellt. Der letzte Abschnitt enthält eine Zusammenfassung und einen Ausblick auf weiterführende Arbeiten.

2. Gesamtarchitektur

Das WEBBOS-Projekt hatte zwei wesentliche Ziele: Zum einen die Entwicklung von Sensorhardware zur Erfassung des Gebietszustands und zum anderen die Realisierung eines Geoinformationsdienstes, der die Sensordaten bereitstellt und Warnmeldungen generiert. Abbildung 1 gibt einen groben Überblick über die Gesamtarchitektur.

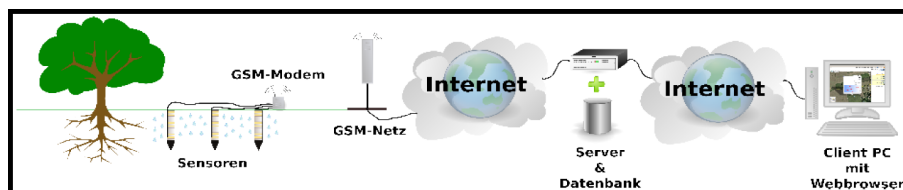


Abbildung 1. Gesamtarchitektur des Sensorsystems.

Anforderungen an das neue Sensorsystem, das die dielektrischen Eigenschaften des Bodens nutzt, waren ein ausreichend großes Messvolumen und Empfindlichkeit, sowie ein zuverlässiger und wartungsfreier, jahrelanger Einsatz auch unter erschwerten Feldbedingungen. Die dielektrischen Eigenschaften eines Bodens sind nicht allein vom Wassergehalt abhängig, auch Dichte und Art des Bodens, die Granularität und Temperatur sowie weitere sind Faktoren, die berücksichtigt werden müssen. Dies erfordert eine spezifische Kalibrierung für unterschiedliche Böden. Es wurden unterschiedliche Bauformen von Sensoren entwickelt und getestet: ein Ringsensor, wie auf der linken Seite in Abbildung 2 dargestellt, und ein mehrteiliges Sensorsystem mit Planarsensoren, wie auf der rechten Seite zu sehen.

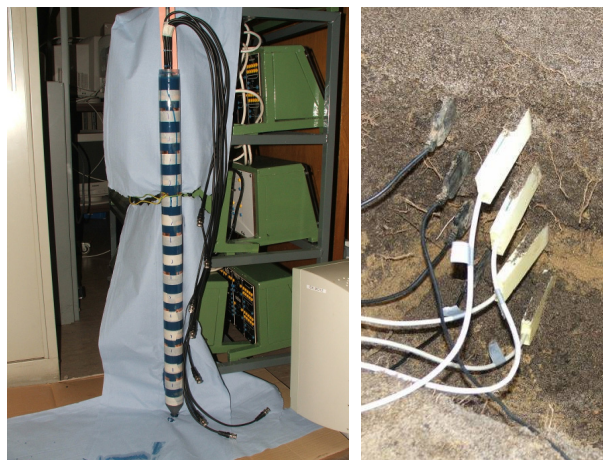


Abbildung 2. Zwei unterschiedliche Sensortypen.

Eine wichtige Anforderung war, dass der Informationsdienst möglichst flexibel für verschiedene Sensortypen geeignet ist und die durch die SWE-Initiative initiierten Sensor Web Services nutzt. Als Basis-Software wurde hierfür das SWE-Framework von 52°North (<http://52north.org/>) verwendet. Dieses hat die größte Verbreitung in der Community und unterstützt die meisten SWE-Funktionen unter Berücksichtigung der Besonderheiten des WEBBOS-Projektes (u.a. die Repräsentation, Speicherung und Abfrage von Messwerten an einem Standort in verschiedenen Tiefen).

3. Client

Der in WEBBOS verwendete Client sollte flexibel einsetzbar und erweiterbar sein. Dies konnte am geeignetsten durch die Entwicklung eines hardwareunabhängigen Web Clients erreicht werden, der unterschiedliche Web Services integriert und deren Daten visualisiert. Um die Eigenschaften verschiedener Lösungsansätze miteinander vergleichen zu können, wurden verschiedene Prototypen realisiert, die auf unterschiedlichen Frameworks aufbauen.

3.1. Auf Standarddiensten beruhende Lösungsansätze

Zunächst wurde eine Lösung auf Basis des „UMN Map Server“ (<http://mapserver.org>) und dem Geoportal „Mapbender“ entwickelt. Alternativ wurde ein Client auf Basis der Open-Source-Bibliothek „OpenLayers“ (<http://www.openlayers.org>) implementiert, der anstelle des „Mapbender“-Portals zum Einsatz kam (vgl. auch Abbildung 3).

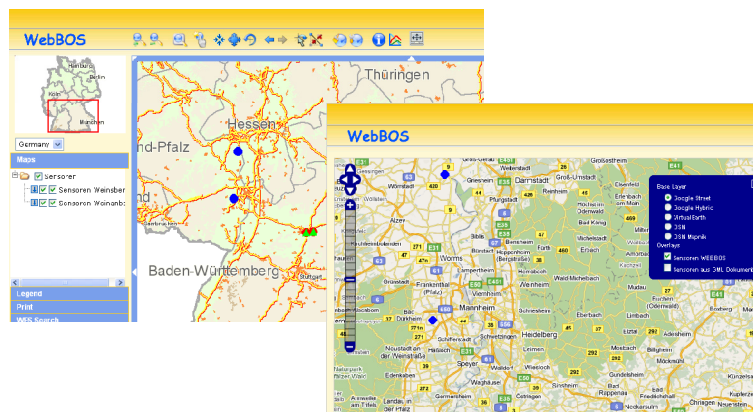


Abbildung 3. Versuche der SOS-Anbindung als WMS mit „Mapbender“ (links) und „OpenLayers“ (rechts).

Im „UMN Map Server“ wurden versuchsweise zwei Sensorgruppen als WMS und WFS erzeugt, die anschließend mit „Mapbender“ respektive „OpenLayers“ über die entsprechenden Schnittstellen angefordert und visualisiert werden konnten. Zur Anfrage von Sensordaten wurde der WMS zum Absetzen einer GetFeatureInfo-Anfrage an einen SOS konfiguriert. Dafür wurde ein Modul integriert, welches bei der Auswahl eines Sensors eine Eingabemaske bereitstellt, in der anzufragenden Phänomene und der Zeitraum ausgewählt werden können. Abschließend wird eine Anfrage aus den Eingabedaten generiert. Auf Basis der Anfrage werden dann die Messwerte innerhalb des angegebenen Zeitraums ermittelt, die in einem Graphen im Web Client dargestellt werden. Dies ist beispielhaft in Abbildung 4 dargestellt.

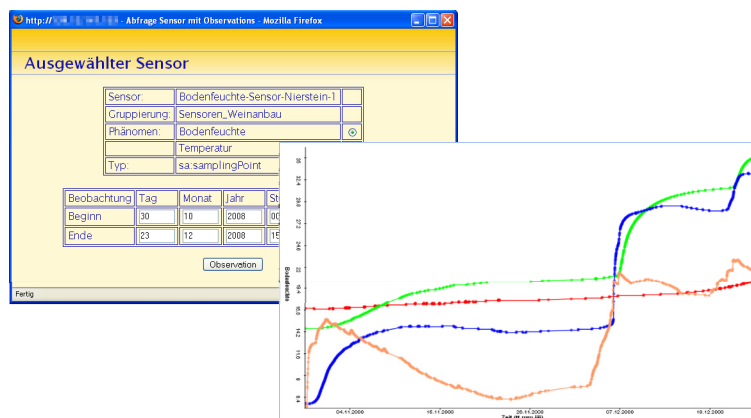


Abbildung 4. Anfragemaske und Antwort als Graph.

Die Kopplung des WFS mit „OpenLayers“ wurde ebenfalls getestet. Obwohl der SOS GML-kodierte Inhalte liefert stellte, sich die Nutzung der GML-Funktionalität von „OpenLayers“ als wenig praktikabel heraus. Die GetFeatureOfInterest-Anfrage lieferte das GML-Dokument als Zeichenkette und nicht als URL, wodurch die Nutzung als GML-Layer verhindert wurde. Alternativ wurde deshalb ein generischer Vector-Layer aus einer eigenen HTTP-POST Anfrage erzeugt.

Beide Lösungen zeigen, dass sich eine nahtlose Integration eines SOS in vorhandene Webdienste schnell realisieren lässt. Aufgrund des nicht auf SWE-Dienste abgestimmten Rahmens sind beide Ansätze relativ starr und inflexibel.

3.2. WEBBOS-Lösung mit „OpenLayers“

Der WEBBOS-Client (vgl. auch Abbildung 5) nutzt daher eine im Projekt entwickelte Erweiterung für „OpenLayers“, die den Zugriff clientseitig realisiert. Die konkrete Umsetzung dieses Clients und die Vorteile, die sich aus der Erweiterung der „OpenLayers“-Bibliothek ergeben, werden im Folgenden detailliert vorgestellt.

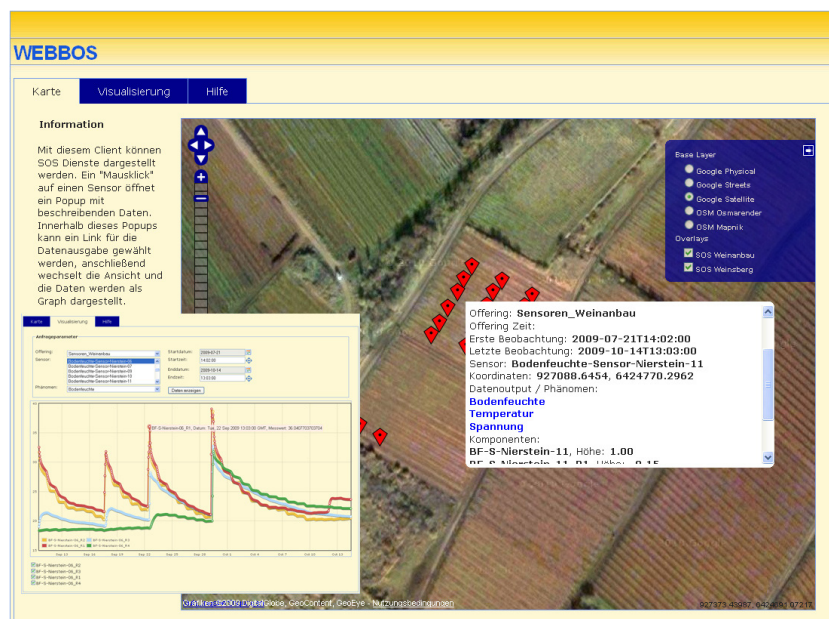


Abbildung 5. Web Client auf Basis von „OpenLayers“ mit SOS-Anbindung.

In der finalen Lösung wurde eine direkte Anbindung eines SOS in einem „OpenLayers“-Client realisiert. Zur Integration eines neuen Dienstes in „OpenLayers“ werden zwei Komponenten benötigt, eine die den Zugriff auf das Datenmodell und eine zweite die die Visualisierung der Daten übernimmt. Dabei wurde die Klasse `OpenLayers.Format.SOS`, welche von `OpenLayers.Format.XML` erbt, zur Verarbeitung der SensorML- und Observation-Dokumente erzeugt. Diese enthält drei Methoden zur direkten Anfrage des SOS. Die verfügbaren Sensoren und andere Metadaten des Service werden mit Hilfe der Methode `write_GetCapabilities` angefragt.

Die Anfrage von Details zu einem Sensor oder dessen aufgenommene Sensorwerte übernehmen die Funktionen *write_DescribeSensor* respektive *write_GetObservation*. Die Antworten des Service werden von entsprechenden Parser-Methoden übernommen. Es werden dabei nur die für den Client benötigten Inhalte extrahiert. Im Falle einer *GetCapabilities*-Anfrage sind das die zur Verfügung stehenden Sensoren und den Zeitstempeln des ältesten und des jüngsten aufgenommenen Messwertes. Mit Hilfe der Sensornamen werden die *DescribeSensor*-Anfragen erzeugt und an den Server geschickt. Aus den Antworten werden dann zunächst die Koordinaten der Sensoren ermittelt, um sie zur Anzeige zu bringen. Anschließend werden die Sensorkomponenten aus dem zurückgelieferten SensorML-Dokuments ausgelesen. Die im Projekt entwickelte Sonde misst die Feuchtigkeit in unterschiedlichen Tiefen, was im SensorML-Dokument durch verschiedene Teilkomponenten abgebildet wird. Mit Hilfe der Klasse *OpenLayers.Layer.SOS*, die von *OpenLayers.Layer.Markers* abgeleitet wurde, werden die Sensorgeometrien in einem eigenständigen Layer als Marker visualisiert. Aktuelle Messwerte werden nach Auswahl eines Markers zusammen mit der Beschreibung des Sensors in einem Popup-Menü dargestellt.

Der Web Client enthält eine zusätzliche Komponente zur Analyse von Messreihen, die wahlweise über die aufgelisteten Sensorkomponenten im Popup-Menü direkt oder über den Reiter „Visualisierung“ im Client aufgerufen werden kann. Die Daten werden mit Hilfe von „flot“ (<http://code.google.com/p/flot/>) einer JavaScript-Visualisierungskomponente dargestellt.

Neben der Darstellung von Messreihen ist auch die Verfolgung von Sensorereignissen eines SES innerhalb des Web Clients realisiert worden. Löst ein SES ein kritisches Ereignis aus, wird dieses von einem SES-Client auf dem Clientrechner registriert. Die Nachricht wird zwischengespeichert und sowohl vom SES-Client als auch im eventuell aktiven Web Client angezeigt. Der Nutzer hat die Möglichkeit, alle zwischengespeicherten Ereignisse im Web Client anzuzeigen und gegebenenfalls genauer zu analysieren (vgl. auch [9]).

3.3. Community-Lösung von „OpenLayers“

Parallel zur und unabhängig von der WEBBOS-Entwicklung wurde „OpenLayers“ ab Version 2.9 von der „OpenLayers“-Community um Funktionen zur Anbindung von Sensor Observation Services erweitert. Bei dieser Umsetzung wurden insgesamt drei neue Klassen zur Verarbeitung von SOS-Anfragen eingeführt, die wie die Eigenentwicklung von *OpenLayers.Format.XML* erbt. Zunächst werden die Metadaten des Service mit Hilfe der Klasse *OpenLayers.Format.SOSCapabilities* abgefragt und geparkt. Anschließend wird eine *GetFeatureOfInterest*-Anfrage erzeugt die die im Capabilities-Dokument enthaltenen *featureOfInterest*-Elemente vom Service anfragt. Die Anfrage und auch die Verarbeitung der Antwort werden in der Klasse *OpenLayers.Format.SOSGetFeatureOfInterest* realisiert. „OpenLayers“ kann neben den Informationen zu den Sensoren auch Sensormesswerte abfragen. Dazu wird die dritte Klasse, *OpenLayers.Format.SOSGetObservation*, benötigt. Zur Darstellung der Sensoren in der Karte wurde die Klasse *OpenLayers.Layer.SOS* erzeugt, welche die Klasse *OpenLayers.Layer.Markers* um Funktionen zur Anfrage aktueller Messwerte erweitert. Die angefragten Messwerte werden dabei in einem Popup-Menü dargestellt.

3.4. Vergleich

Im direkten Vergleich ähneln sich die beiden Ansätze. Die Verarbeitung der Anfragen wurde bei der Eigenentwicklung innerhalb einer Klasse realisiert, was im direkten Vergleich die Übersichtlichkeit etwas einschränkt. Ein großer Unterschied ist bei der Anfrage von Sensorinformationen zu sehen, da hier zwei unterschiedliche Wege eingeschlagen werden. Während die im Projekt eingesetzte Lösung die Sensorgeometrie über eine DescribeSensor-Anfrage erhält, werden die geometrischen Eigenschaften bei der „OpenLayers“ Version 2.9 über das FeatureOfInterest angefragt. Hierdurch wird die unterschiedliche Zielsetzung deutlich. Die primäre Zielsetzung von „OpenLayers“ liegt in der Anbindung des Service zur Darstellung der überwachten Phänomene in der Karte. Mit der Projektrealisierung wird ein anderer Ansatz verfolgt. Hier ist ein besonderer Gesichtspunkt, die Unterstützung von dreidimensionalen Feuchteprofilen (d.h. mit Lage- und Tiefeninformation) und eine Analyse deren Messwerthistorie zu realisieren. Die Funktionen von „OpenLayers“ 2.9 beschränken sich im Grunde auf die Anfrage eines Messwertes zu einem Zeitpunkt. Die Anfrage einer Messreihe zur visuellen Analyse ist somit nicht vorgesehen. Bei der Eigenentwicklung wurde die gesamte Verwaltung von Anfragen über „OpenLayers“-Erweiterung realisiert, so auch die Anfrage von Messreihen.

4. Test und Erprobung

Neben Labortests wurden die Sensoren und die Dienstankopplung auch in Feldversuchen erfolgreich an zwei Standorten im Einzugsbereich des Flusses Sulm und – über den ursprünglichen Einsatzzweck hinausgehend – für die Bodenfeuchtemessung im Weinbau erprobt. Abbildung 5 zeigt links die Basisstation, an die die Sensoren gabelgebunden angeschlossen werden. Die rechte Abbildung zeigt ein komplett unterirdisch eingebautes Funkmodul, welches ebenfalls erprobt wurde. Dieses kann in direkter Nähe zur Sensorik platziert werden und erspart längere Kabelverbindungen zu einer Basisstation.



Abbildung 5. Basisstation und Funkmodul.

5. Zusammenfassung und Ausblick

In diesem Beitrag wurde die im Rahmen des WEBBOS-Projektes entwickelte Lösung zur flexiblen webbasierten Visualisierung von Sensordaten vorgestellt. Es wurden verschiedene Herangehensweisen zur Anbindung von Sensor-Web-Diensten dargestellt und gezeigt, dass sich mit von „OpenLayers“ eine sehr flexible clientseitige Variante entwickelt wurde, die auch weiterführende Analysen von Messreihen ermöglicht. Im Vergleich der Community-Lösung mit der WEBBOS-Lösung wurde die unterschiedliche Zielsetzung herausgearbeitet, aber auch Gemeinsamkeiten der Ansätze deutlich gemacht.

Es ist geplant, den Einsatz des entwickelten WEBBOS-Systems auf andere Anwendungsgebiete, wie dem Küstenschutz, auszuweiten. Außerdem wird eine Weiterentwicklung der Ereignisverarbeitung und -analyse im Hinblick auf komplexere Ereignisse in Datenstrommanagementsystemen mit Hilfe von Complex Event Processing (CEP) angestrebt.

Literatur

- [1] Hübner, C. (ed.), Wireless soil moisture sensor networks for environmental monitoring and irrigation, *Geophysical Research Abstracts*, Vol. 12, EGU General Assembly 2010, Vienna, 2010
- [2] Cox, S. (ed.), *Observations and Measurements*, OpenGIS Discussion Paper (Version 0.13.0) OGC 05-087r3, 2006
- [3] Botts, M. (ed.), OpenGIS Sensor Model Language (SensorML) Implementation Specification (Draft Proposed Version 1.0), OGC 05-086r2, 2006
- [4] Na, A., Priest, M. (eds.), Sensor Observation Service, OpenGIS Implementation Specification (Request for Comment 0.1.5) OGC 06-009r1, 2006
- [5] Echterhoff, J., Everding, T., OpenGIS Sensor Event Service Interface Specification (proposed), Version 0.3.0, OpenGIS Discussion Paper, OGC 08-133, 2008
- [6] Fleischer, J., Häner, J., Herrkind, S., Kloth, A., Kriegel, U., Schwarting, H., Wächter, J., 2010. An integration platform for heterogeneous sensor systems in GITEWS – Tsunami Service Bus. *Natural Hazards and Earth System Sciences*, Vol. 10, pp. 1239-1252.
- [7] Spies, K.-H., Heier, C., 2008. OGC Sensor Web in der Praxis – Bereitstellung von Sensordaten in Geodateninfrastrukturen und personalisierter Hochwasserwarndienst, In: *Proceedings 20. AGIT-Symposium*, 2008, Salzburg, Austria, pp. 721-725.
- [8] Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R., 2011. New Generation Sensor Web Enablement. *Sensors* 2011, Vol. 11, pp. 2652-2699, doi:10.3390/s110302652
- [9] Brinkhoff, T., Jansen, S., Sensor Web Services for Early Flood Warnings Based on Soil Moisture Profiles, *akzeptiert für den ISPRS-Kongress 2012*, Melbourne, Australien, 2012

Danksagung

Die hier vorgestellte Arbeit wurde im Rahmen des FHprofUnt-Programms durch das Bundesministerium für Bildung und Forschung gefördert. Dank geht auch an Dipl.-Ing. (FH) Christian Knese, der in Rahmen seiner Tätigkeit im WEBBOS-Projekt entscheidend an den hier vorgestellten Arbeiten beteiligt war.