

---

Technische Universität Dresden

Institut für Geoinformationssysteme

**Dokumentation**  
**„GDI und Sensorweb“**

**- Entwicklung eines Hochwasserwarnsystems -**



Studierende:

Ulrike Hagemann, Kartographie  
Luise Hutka, Geographie  
Rico Kronenberg, Hydrologie  
Meike Viehweger, Kartographie  
Oliver Zeinz, Geographie

Betreuer:

Prof. Lars Bernard  
Dipl.- Geoinf. Johannes Brauner  
Dipl.- Geogr. Matthias Müller

Dresden, den 10. Juli 2009

# Inhaltsangabe

<b>1. Einleitung</b>	<b>3</b>
1.1. Motivation	3
1.2. Zielstellung	3
<b>2. Datengrundlage</b>	<b>4</b>
<b>3. Architektur</b>	<b>5</b>
<b>3.1. Sensor Observation Service</b>	<b>6</b>
<b>3.2. Warndienst</b>	<b>8</b>
<b>3.3. Web Processing Service</b>	<b>10</b>
WATSCH-Algorithmus	10
<b>3.4. Web Map Service</b>	<b>13</b>
<b>4. Zusammenfassung, Fazit, Ausblick</b>	<b>16</b>
<b>5. Literaturverzeichnis:</b>	<b>18</b>

# 1. Einleitung

## 1.1. Motivation

*„Die Natur versteht gar keinen Spaß; sie ist immer wahr, immer ernst, immer strenge, sie hat immer recht“ (Johann Wolfgang von Goethe)*

In den letzten Jahren kam es häufig zu extremen Wettersituationen. So war das Hochwasser der Elbe im Sommer 2002 eines der höchsten, die es je gegeben hat. Zu Beginn des Frühlings 2006 kam es erneut zu starken Überflutungen an der Elbe. Die Ursachen waren heftige Regenfälle im Sommer und ein starkes Tauwetter im Winter. Aus diesen Erkenntnissen kann geschlossen werden, dass auch in den nächsten Jahren mit einer erhöhten Überflutungsgefahr an der Elbe gerechnet werden muss.

Das Hochwasser 2002 zeigte sehr deutlich, dass bessere und schnellere Warnsysteme benötigt werden. Dabei ist es nicht unbedingt wichtig, hochpräzise Voraussagen zu treffen. Es werden grobe, dafür aber schnellere Prognosen benötigt.

Durch das immer häufigere Auftreten solcher Umweltkatastrophen nimmt die Daseinsberechtigung von Warnsystemen mehr und mehr zu. Und so ist es nicht verwunderlich, dass ausgerechnet in Dresden, welches vom Elbehochwasser 2002 sehr stark betroffen war, die Idee zu einem Warnsystem entstanden ist. Um Menschenleben zu schützen und Schäden zu verringern, sollten solche Warnsysteme installiert werden, damit entsprechende Vorbereitungen sowohl durch Anwohner als auch die für die Organisation verantwortlichen Behörden getroffen werden können.

Im Rahmen der Veranstaltung „Geodateninfrastrukturen 2“ soll ein Prototyp eines Warnsystems aufgebaut werden.

## 1.2. Zielstellung

Ziel dieses Projektes ist der Aufbau einer GeoDatenInfrastruktur (GDI) zur zeitnahen Ermittlung von Gefahrenpotentialen im Einzugsgebiet von Flüssen. Hierfür wurden bestehende Konzepte genutzt, um den Informationsfluss zu organisieren. Jedoch machte die Zielsetzung auch die Einführung einer neuen Komponente (Warnsystem) und die Entwicklung eines Überflutungsalgorithmus (WATSCH) notwendig, welche in die GDI integriert wurden. Es soll ebenfalls gezeigt werden, dass eine anwenderunabhängige Prozessierung innerhalb einer GDI möglich ist.

Die Pegelstände werden von einem Sensor Observation Service von der PEGELONLINE-Homepage abgefragt und mit vorgegebenen Alarmstufen abgeglichen. Diese Informationen werden dann in einem Web Processing Service prozessiert. Am Ende wird das Ergebnis mittels eines Web Map Services visualisiert. Dabei werden die berechneten, überfluteten Flächen in einem digitalen Geländemodell dargestellt. Eine hoch präzise

Berechnung der Daten soll bei diesem System vermieden werden, da es sich um ein einfaches Modell handelt, welches eine schnelle Warnung möglich macht.

## 2. Datengrundlage

Das Projekt WATSCH basiert auf zwei Datensätzen und beschränkt sich räumlich auf den sächsischen Teil der Elbe. So beruht der WATSCH-Algorithmus, für den Ausschnitt von Sachsen, auf SRTM90-Daten (PROCESSED SRTM DATA VERSION 4.1), welche über die NASA frei verfügbar sind. Aus diesem DGM wurden ein Elbe-Raster und ein Pegel-Raster im ArcGrid-Format abgeleitet.

Das Elbe-Raster besitzt nur den Flusslauf und Fehlwerte. In Abbildung 1 ist beispielhaft ein Rasterausschnitt dargestellt.

```

0 0 0 0 0 0 0 0 0 0 355 0 0 0 0
0 0 0 0 0 0 0 0 0 0 340 0 0 0 0
0 0 0 0 0 0 0 0 0 0 340 0 0 0 0
0 0 0 0 0 0 0 0 0 0 355 0 0 0 0
0 0 0 0 0 0 0 0 0 0 340 0 0 0 0
0 0 0 0 0 0 0 0 0 310 0 0 0 0 0
0 0 0 0 0 0 0 0 310 0 0 0 0 0 0
0 0 0 0 0 0 320 0 0 0 0 0 0 0 0
0 0 0 0 0 360 0 0 0 0 0 0 0 0 0
0 0 0 0 355 0 0 0 0 0 0 0 0 0 0
0 0 0 0 340 0 0 0 0 0 0 0 0 0 0
0 0 0 0 345 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 356 0 0 0 0 0 0 0 0 0
0 0 0 0 0 360 0 0 0 0 0 0 0 0 0

```

**Abbildung 1: Ausschnitt aus dem Elbe-Raster ohne ArcGrid-Header, mit einer Fehlwertkennung von Null**

Das Pegel-Raster besitzt nur Pegelnummern und Fehlwerte. In der Abbildung 2 ist beispielhaft ein Rasterausschnitt dargestellt.

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 50102 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 50234 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 50232 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 50236 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

**Abbildung 2: Ausschnitt aus dem Pegel-Raster ohne ArcGrid Header, mit einer Fehlwertkennung von Null**

Datenlücken innerhalb dieses Datensatzes konnten durch Interpolation gefüllt werden. Des weiteren benötigt der WATSCH-Algorithmus

Wasserstände der Elbe. PEGELONLINE stellt zeitaktuelle, gewässerkundliche Informationen zur Verfügung. So sind für das Projekt WATSCH vor allem die aktuellen und die statistisch ermittelten Wasserstände der Elbe im Bundesland Sachsen von besonderer Bedeutung. Diese Daten werden ungefiltert als Rohdaten online gestellt. Es erfolgt von Seiten der Wasser- und Schifffahrtsverwaltung des Bundes keine Qualitätskontrolle über Plausibilität dieser Daten. So werden Messfehler ohne Korrektur zum Download angeboten, diese bleibt dem Anwender vorbehalten.

### 3. Architektur

Das WATSCH-Projekt hat das Ziel, automatische Überflutungsflächen mit zugehörigen Warnstufen zu berechnen und online für einen Client, jeweils in der aktuellsten Version, zur Verfügung zu stellen.

Die GDI des WATSCH!-Projekt es ist in Abbildung 3 dargestellt.

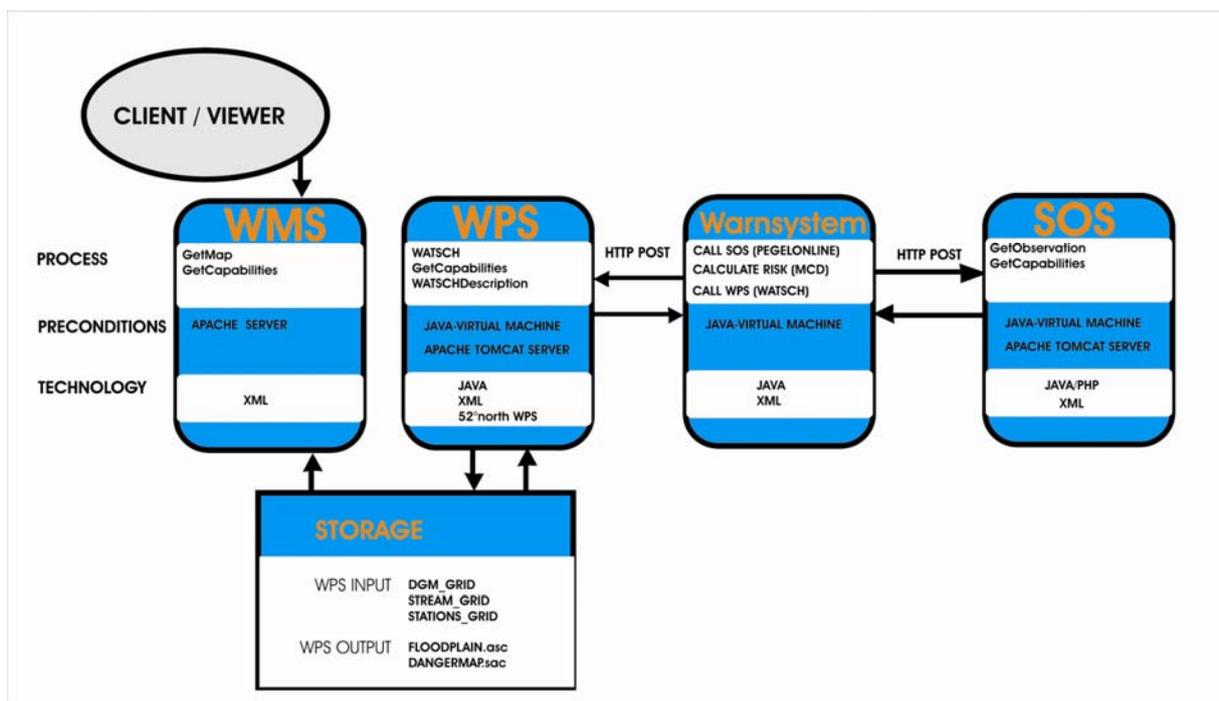


Abbildung 3: WATSCH-Architektur

Um die Zielstellung des Projektes zu erreichen, sind folgende Komponenten notwendig:

Gesteuert wird die GDI über ein Warnsystem. Dieses fragt die aktuellen Pegeldata von einem Sensor Observation Service (SOS) ab und lässt einen Web Processing Service (WPS) den WATSCH-Algorithmus ausführen. Anschließend werden die Ergebnisse über einen Web Map Service (WMS) visualisiert. Dabei arbeiten alle Komponenten der GDI unabhängig voneinander, d.h. sowohl WPS als auch SOS sind unabhängig vom

Warnsystem ausführbar und über OGC-spezifische Schnittstellen ansprechbar. Ausschließlich WPS und WMS sind auf den gleichen Speicher (STORAGE) angewiesen, denn der WPS benötigt diesen zur Abfrage des Elbe-, Stations- und Höhen-Raster. Die Ergebnis-Raster werden auf dem STORAGE abgelegt, so dass der WMS die Ergebnisse veröffentlichen kann.

Die Sequenzkette besteht aus zwei Teilen. Der Hauptprozess wird clientunabhängig vom Warnsystem gesteuert. So fragt dieses in einem 15-minütigen Intervall den SOS ab und bewertet die empfangenen Daten. Diese Daten werden an den WPS gesendet, welcher den WATSCH-Algorithmus auslöst und die Ergebnisse im STORAGE ablegt.

Der Client hat über den WMS die Möglichkeit, die im STORAGE gespeicherten Ergebnisse des WPS, abzurufen. Die Sequenzabfolge ist der Abbildung 4 zu entnehmen.

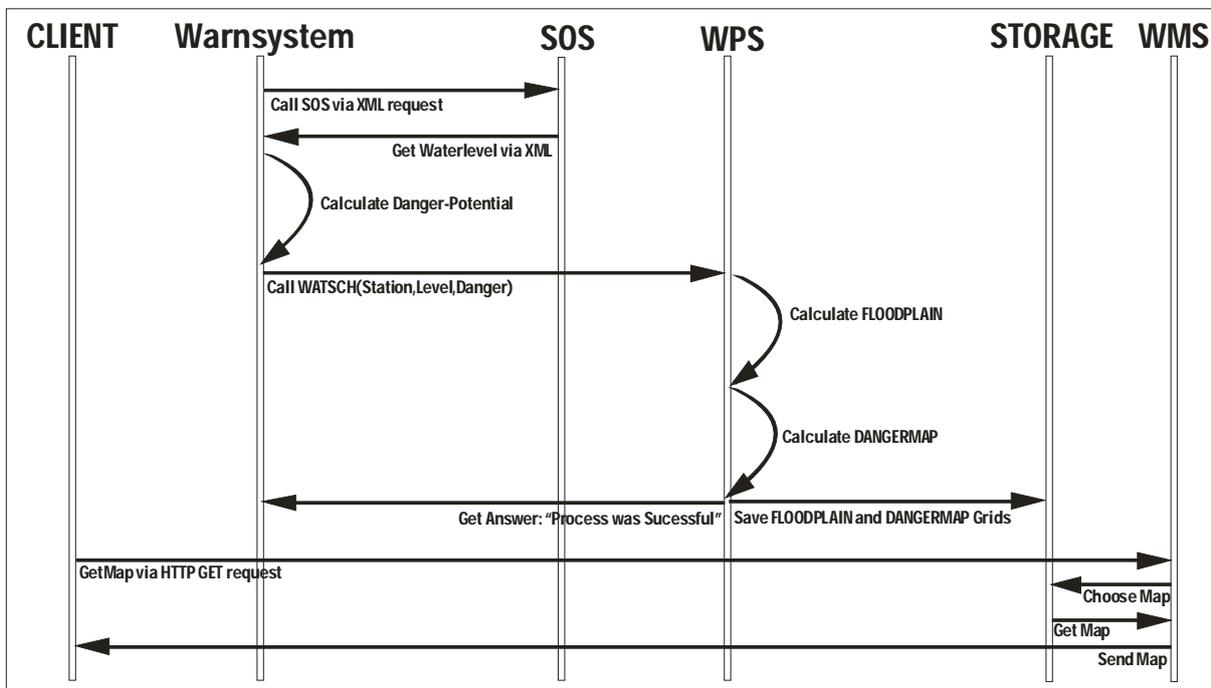


Abbildung 4: Sequenzdiagramm des Projektes WATSCH

### 3.1. Sensor Observation Service

Bei dem PEGELONLINE Sensor Observation Service (SOS) handelt es sich um einen OGC-konformen Dienst, über den Sensordaten verschiedener Pegelmessstationen Deutschlands abgerufen werden können. Dabei nutzt der SOS die *Observation and Measurements (O&M)* Spezifikation für die Modellierung von Sensordaten. Über die obligatorische GetCapabilities-Funktion, die in Ausschnitten im folgenden Listing zu sehen ist, erhält man einen Überblick, welche Sensorparameter („ows:Value“) für welche

Pegelstationen („sos:featureOfInterest“) abgefragt werden können. (vgl. [OGC01])

```

<?xml version="1.0" encoding="UTF-8" ?>
<sos:Capabilities version="1.0.0" updateSequence="2005-12-14T10:12:39"
  xsi:schemaLocation="http://www.opengis.net/sos/1.0 http://schemas.opengis.net/sos/1.0.0/sosAll.xsd"
  xmlns:sos="http://www.opengis.net/sos/1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink">
+ <ows:ServiceIdentification xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:swe="http://www.opengis.net/swe/1.0">
+ <ows:ServiceProvider xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:swe="http://www.opengis.net/swe/1.0">
- <ows:OperationsMetadata xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:swe="http://www.opengis.net/swe/1.0">
+ <ows:Operation name="GetCapabilities">
- <ows:Operation name="GetObservation">
+ <ows:DCP>
+ <ows:Parameter name="version">
+ <ows:Parameter name="service">
+ <ows:Parameter name="srsName">
+ <ows:Parameter name="offering">
+ <ows:Parameter name="eventTime">
+ <ows:Parameter name="procedure">
- <ows:Parameter name="observedProperty">
- <ows:AllowedValues>
  <ows:Value>Wasserstand</ows:Value>
  <ows:Value>mittlerer niedrigster Wasserstand</ows:Value>
  <ows:Value>Pegelnulpunkt</ows:Value>
  <ows:Value>mittlerer hoechster Wasserstand</ows:Value>
  <ows:Value>Mittelwert Wasserstand</ows:Value>
</ows:AllowedValues>
...
<sos:featureOfInterest xlink:href="Emshoern (9340010)" />
<sos:featureOfInterest xlink:href="Grabow Op (596330)" />
<sos:featureOfInterest xlink:href="Anklam (9660001)" />
<sos:featureOfInterest xlink:href="Stoer-Sperrwerk Bp (5970040)" />
<sos:featureOfInterest xlink:href="Gundelsheim Uw (23800620)" />
<sos:featureOfInterest xlink:href="Hann.muenden (43100109)" />
<sos:responseFormat>text/xml;subtype="om/1.0.0"</sos:responseFormat>
<sos:resultModel xmlns:ns="http://www.opengis.net/om/1.0">ns:Observation</sos:resultModel>
<sos:responseMode>inline</sos:responseMode>
</sos:ObservationOffering>
</sos:ObservationOfferingList>
</sos:Contents>
</sos:Capabilities>

```

**Listing 1: Ausschnitt aus der GetCapabilities-Funktion des PEGELONLINE SOS**

Die eigentliche Abfrage der Pegeldata erfolgt über die GetObservation-Funktion des SOS. Diese wird als XML-Dokument via HTTP-POST an die von PEGELONLINE bereitgestellte URL <http://sensorweb.dlz-it-bvbs.bund.de/PegelOnlineSOS/sos> gesendet. Die Abfrage enthält den abzurufenden Parameter (z.B. Wasserstand), den erforderlichen Zeitraum sowie die gewünschte Pegelstation. Ein Beispiel für die GetObservation-XML-Anfrage ist in Listing 2 abgebildet.

Als Antwort liefert der SOS ein XML-Dokument zurück, welches die Messwerte für den angegebenen Zeitraum der gewünschten Pegelstation enthält. Ein Antwortbeispiel der GetObservation-Funktion ist in Listing 3 abgebildet.

## 3.2. Warndienst

Als Warndienst wird eine selbst entwickelte Java-Schnittstelle bereitgestellt, in der festgelegt werden kann, in welchem Intervall der SOS abgerufen werden soll.

Nach dem Start der Applikation werden zunächst Informationen über die gewünschten Messstationen abgerufen. Dafür wird die von PEGELONLINE bereitgestellte Client-Schnittstelle des PEGELONLINE Webservice-Aktuell genutzt. Anschließend werden für jede Messstation drei Hochwasserstufen gespeichert, welche auf der Webseite des *Sächsischen Staatsministeriums für Umwelt und Landwirtschaft* beschrieben sind.

Schließlich kann der Warndienst mit der gewünschten Abrufperiode gestartet werden, sodass z.B. automatisch alle 15 Minuten die Daten vom SOS abgefragt werden. Es wird nun für jede Messstation eine XML-Datei erzeugt, in der der abzurufende Zeitraum auf das jeweils aktuelle Datum eingegrenzt wird, um den neuesten verfügbaren Pegelstand für diese Messstation vom PEGELONLINE-SOS zu erhalten. Für die Pegelstation Riesa sieht die XML-Anfrage beispielsweise wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ns:GetObservation service="SOS" version="1.0.0" xmlns:ns="http://www.opengis.net/sos/1.0">
  <ns:offering>urn:ogc:offering:offering0</ns:offering>
- <ns:eventTime>
  - <ogc:TM_Equals xmlns:ogc="http://www.opengis.net/ogc">
    <ogc:PropertyName>urn:ogc:data:time:iso8601</ogc:PropertyName>
    - <gml:TimePeriod xmlns:gml="http://www.opengis.net/gml">
      <gml:beginPosition>2009-06-15T00:00:00+0200</gml:beginPosition>
      <gml:endPosition>2009-06-15T23:59:00+0200</gml:endPosition>
    </gml:TimePeriod>
    </ogc:TM_Equals>
  </ns:eventTime>
  <ns:procedure />
  <ns:observedProperty>Wasserstand</ns:observedProperty>
- <ns:featureOfInterest>
  <ns:ObjectID>Riesa (501110)</ns:ObjectID>
  </ns:featureOfInterest>
  <ns:responseFormat>text/xml;subtype="om/1.0.0"</ns:responseFormat>
  <ns:resultModel xmlns:gml="http://www.opengis.net/gml">gml:Observation</ns:resultModel>
  <ns:responseMode>inline</ns:responseMode>
</ns:GetObservation>
```

### Listing 2: GetObservation-Anfrage für Pegelstation Riesa

Diese Anfrage wird schließlich an die URL des PEGELONLINE-SOS geschickt.

Die Antwort des SOS enthält, wie das folgende Beispiel zeigt, bei erfolgreichem Abruf die letzten aktuellen Messwerte, inklusive Datum und Uhrzeit, an dieser Pegelstation.

```

<?xml version="1.0" encoding="UTF-8" ?>
<om:ObservationCollection xmlns:om="http://www.opengis.net/om/1.0" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:swe="http://www.opengis.net/swe/1.0.1" xmlns:sa="http://www.opengis.net/sampling/1.0"
  gml:id="oc1245055026035" xsi:schemaLocation="http://www.opengis.net/om/1.0
  http://schemas.opengis.net/om/1.0.0/om.xsd http://www.opengis.net/sampling/1.0
  http://schemas.opengis.net/sampling/1.0.0/sampling.xsd">
+ <gml:boundedBy>
- <om:member>
- <om:Observation>
  + <om:samplingTime>
  <om:procedure xlink:href="Wasserstand - Riesa (501110)" />
  + <om:observedProperty>
  + <om:featureOfInterest>
  - <om:result>
    - <swe:DataArray>
      + <swe:elementCount>
      + <swe:elementType name="Components">
      - <swe:encoding>
        <swe:TextBlock decimalSeparator="." tokenSeparator="," blockSeparator="@@"/>
        </swe:encoding>
        <swe:values>2009-06-15T00:00:00+02,Riesa (501110),211.0@@2009-06-
          15T00:15:00+02,Riesa (501110),211.0@@2009-06-15T00:30:00+02,Riesa
          (501110),211.0@@2009-06-15T00:45:00+02,Riesa (501110),211.0@@2009-06-
          15T01:00:00+02,Riesa (501110),211.0@@2009-06-15T01:15:00+02,Riesa
          (501110),211.0@@2009-06-15T01:30:00+02,Riesa (501110),211.0@@2009-06-
          15T01:45:00+02,Riesa (501110),211.0@@2009-06-15T02:00:00+02,Riesa
          (501110),211.0@@2009-06-15T02:15:00+02,Riesa (501110),211.0@@2009-06-
          15T02:30:00+02,Riesa (501110),211.0@@2009-06-15T02:45:00+02,Riesa
          (501110),211.0@@2009-06-15T03:00:00+02,Riesa (501110),211.0@@2009-06-
          15T03:15:00+02,Riesa (501110),211.0@@2009-06-15T03:30:00+02,Riesa
          (501110),211.0@@2009-06-15T03:45:00+02,Riesa (501110),211.0@@2009-06-
          15T04:00:00+02,Riesa (501110),211.0@@2009-06-15T04:15:00+02,Riesa
          (501110),211.0@@2009-06-15T04:30:00+02,Riesa (501110),211.0@@2009-06-
          15T04:45:00+02,Riesa (501110),211.0@@2009-06-15T05:00:00+02,Riesa
          (501110),211.0@@2009-06-15T05:15:00+02,Riesa (501110),211.0@@2009-06-
          15T05:30:00+02,Riesa (501110),211.0@@2009-06-15T05:45:00+02,Riesa
          (501110),211.0@@2009-06-15T06:00:00+02,Riesa (501110),211.0@@2009-06-
          15T06:15:00+02,Riesa (501110),211.0@@2009-06-15T06:30:00+02,Riesa
          (501110),210.0@@2009-06-15T06:45:00+02,Riesa (501110),210.0@@2009-06-
          15T07:00:00+02,Riesa (501110),210.0@@2009-06-15T07:15:00+02,Riesa
          (501110),210.0@@2009-06-15T07:30:00+02,Riesa (501110),210.0@@2009-06-
          15T07:45:00+02,Riesa (501110),210.0@@2009-06-15T08:00:00+02,Riesa
          (501110),210.0@@</swe:values>
        </swe:DataArray>
      </om:result>
    </om:Observation>
  </om:member>
</om:ObservationCollection>

```

**Listing 3: GetObservation-Antwort für Pegelstation Riesa**

Im Falle einer erfolgreichen Abfrage des SOS, wird die Antwort geparkt und der letzte Messwert sowie dessen Datum und Uhrzeit für diese Pegelstation gespeichert.

Anschließend wird für jede Pegelstation die aktuelle Alarmstufe berechnet. Im Falle eines fehlerhaften Messwertes wird anstatt der Alarmstufe der Error-Value "0" gesetzt. Liegt der Messwert des Pegels unterhalb der definierten Hochwasserstufen, wird "1" als Alarmstufe gesetzt. Beim Überschreiten einer der Hochwasserstufen wird die entsprechende Alarmstufe gesetzt: niedrige Hochwasserstufe "2", mittlere "3" und höchste "4". Für die Hochwasserstufen werden die offiziellen Werte des *Sächsischen Staatsministeriums für Umwelt und Landwirtschaft* verwendet, die auf der Webseite [WWW02] als PDF-Datei heruntergeladen werden können.

Nun werden Pegelnummer, jeweiliger Messwert und jeweilige Alarmstufe zu einem String zusammengefügt und schließlich bei Aufruf des Web Processing Service (WPS) an diesen übergeben.

Die GetObservation-Antwort des SOS kann allerdings auch eine Fehlermeldung zurückliefern. Das ist bei Pegelstationen der Fall, deren Name einen Umlaut enthält (z.B. Schöna oder Mühlberg). Dies scheint ein Fehler des PEGELONLINE-SOS zu sein, denn auch wenn die Anfrage entsprechend der GetCapabilities-Funktion formuliert wird, wird eine Fehlermeldung für diese Pegelstationen zurück geliefert.

### **3.3. Web Processing Service**

Ein weiterer Pfeiler der Projektarchitektur ist der Web Processing Service (WPS). Dieser Dienst stellt eine unabhängige Prozessierung dem Anwender online zur Verfügung. Er unterliegt den OGC-Spezifikationen Web Processing Service [WPS, OGC 2007]. Seine Prozesse sind via HTTP POST ausführbar. In diesem Projekt wurde der WPS von 52°north [WWW03] genutzt. Dieser Dienst erfüllt den aktuellen Standard. Somit sind drei Grundfunktionen im WPS implementiert:

1. *GetCapabilities*, welche Meta-Daten des Dienstes liefert
2. *DescribeProcess*, welche die zur Verfügung gestellten Prozesse beschreibt, einschließlich ihres Inputs und Outputs
3. *Execute*, welche den Output eines Prozesses zurückliefert

Sollte der Fall eintreten, dass mehrere Anfragen für einen Prozess den WPS erreichen, so arbeitet der WPS diese Anfragen der Reihe nach ab. Der Client erhält in diesem Fall eine Mitteilung mit dem Hinweis, dass die Anfrage so bald wie möglich ausgeführt wird. Der 52°north WPS ist OpenSource-Software, damit ist es möglich, diesen um zusätzliche Prozesse zu erweitern. Für das Projekt wurde dem WPS ein neuer Algorithmus (WATSCH) hinzugefügt.

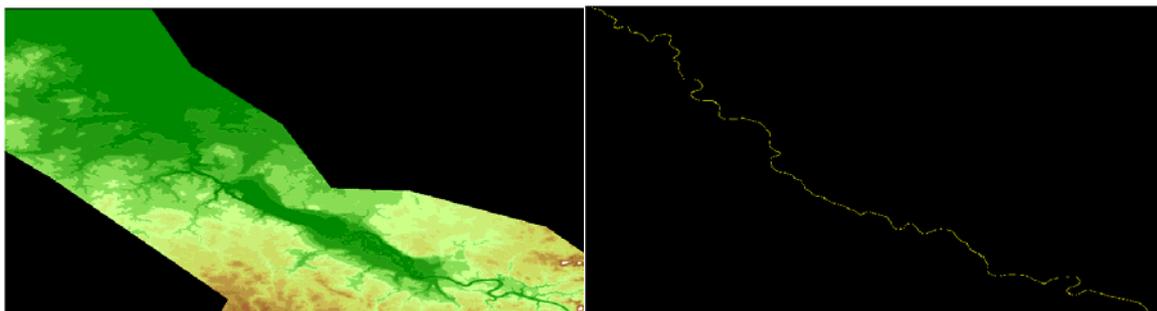
#### **WATSCH-Algorithmus**

Der WATSCH-Algorithmus hat das Ziel, rasterbasiert Überflutungsflächen mit der zugehörigen Warnstufe auszuweisen. WATSCH beruht dabei auf einfachen Annahmen und ist ein prototypischer Algorithmus. Die Voraussetzungen für diesen Algorithmus sind ein Digitales Geländemodell (DGM), der untersuchte Fluss, sowie die anliegenden Stationen. Beispiel-Inputdaten sind in Abbildung 5 dargestellt. Zusätzlich wird, durch den Execute-Request via XML-Dokument, als literal Data, jedem Pegel ein Wasserstand und eine Warnstufe zugeordnet. Mittels dieser Daten interpoliert der Algorithmus linear zwischen zwei Pegeln den Wasserstand. Der resultierende Flusswasserstand wird zur Geländehöhe addiert. Das Ergebnis-Pixel wird anschließend mit den angrenzenden DGM-Pixeln

flussaufwärts verglichen. Sollte das berechnete Pixel höher liegen, so wird das zu vergleichende DGM-Pixel auf das entsprechende Niveau erhöht. Nachfolgend ist der Algorithmus vereinfacht als Pseudocode zusammengefasst.

- Parse aus xml-Request: Pegelnummer, Wasserstand, Warnstufe
- Erstelle neues Raster „FloodplainMap“ und Raster „DangerMap“
- Suche Pegelnummer im Pegel-Raster und weise Floodplain bzw. DangerMap den entsprechenden Wert mit der entsprechenden Position zu
- Suche höchsten Punkt des Elbe-Rasters aus dem Vergleich der DGM Werte -> Startpunkt für die Interpolation
- gehe den Fluss entlang und suche den ersten Pegel (1) und merke dir die Position
  - geh weiter und suche den nächsten Pegel-> wenn nächster Pegel (2) gefunden
    - interpoliere über die durchgelaufene Fließstrecke bis zum ersten Pegel (1)
    - übergib das Ergebnis an FloodplainMap (Wasserstand) und DangerMap (Warnstufe)
  - anschließend wird der Pegel (2) zu Pegel (1) von dort wird der nächste Pegel flussabwärts gesucht, bis es keinen Pegel mehr gibt
- Vergleiche FloodplainMap+DGM-Raster mit umliegenden Rasterzellen von FloodplainMap -> fülle die Rasterzellen auf, die niedriger sind in FloodplainMap (Wasserstand) und DangerMap(Warnstufe)
- Speichere FloodplainMap und DangerMap ab

#### Listing 4: Pseudo Code



**Abbildung 5: Rastergrundlage für den WATSCH-Algorithmus. Links: Ausschnitt des DGM der Elbe innerhalb von Sachsen. Rechts: Flusslauf der Elbe DATASET-SRTM90**

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://geoserver.itc.nl:8080/wps/schemas/wps/1.0.0/wpsDescribeProcess_response.xsd"
  version="1.0.0" xml:lang="en-US">
  <ProcessDescription wps:processVersion="2" storeSupported="true" statusSupported="false">
    <ows:Identifier>org.n52.wps.server.algorithm.foss4g.Watsch</ows:Identifier>
    <ows:Title>WATSCH</ows:Title>
    xmlns:wps="http://www.opengis.net/wps/1.0.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    service="WPS"
  </ProcessDescription>
</wps:ProcessDescriptions>
```

```

<ows:Abstract>Calculation of Floodplains</ows:Abstract>

<DataInputs>
  <Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>WaterLevel:LevelNumber</ows:Identifier>
    <ows:Title>Need LevelNumber and current WaterLevel</ows:Title>
    <ows:Abstract>Input String Format should be :
    Level1@@Value1@@Warn1@@Level2@@Value2@@Warn2@@..</ows:Abstract>
    <LiteralData>
      <ows:DataType ows:reference="xs:string"></ows:DataType>
      <ows:AllowedValues>
        <ows:Value></ows:Value>
      </ows:AllowedValues>
    </LiteralData>
  </Input>
</DataInputs>
<ProcessOutputs>
  <Output>
    <ows:Identifier>FLOOD_RASTER</ows:Identifier>
    <ows:Title>FloodPlain Raster</ows:Title>
    <ows:Abstract>ASCII Format output File</ows:Abstract>

    <LiteralOutput>
      <ows:DataType ows:reference="xs:string"></ows:DataType>
    </LiteralOutput>
  </Output>
</ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>

```

### Listing 5: Prozessbeschreibung als abrufbares XML-Dokument

Die Ausführung eines Prozesses ist im 52°north WPS ausschließlich per HTTP POST möglich. Das entsprechende XML Dokument ist nachfolgend zu sehen.

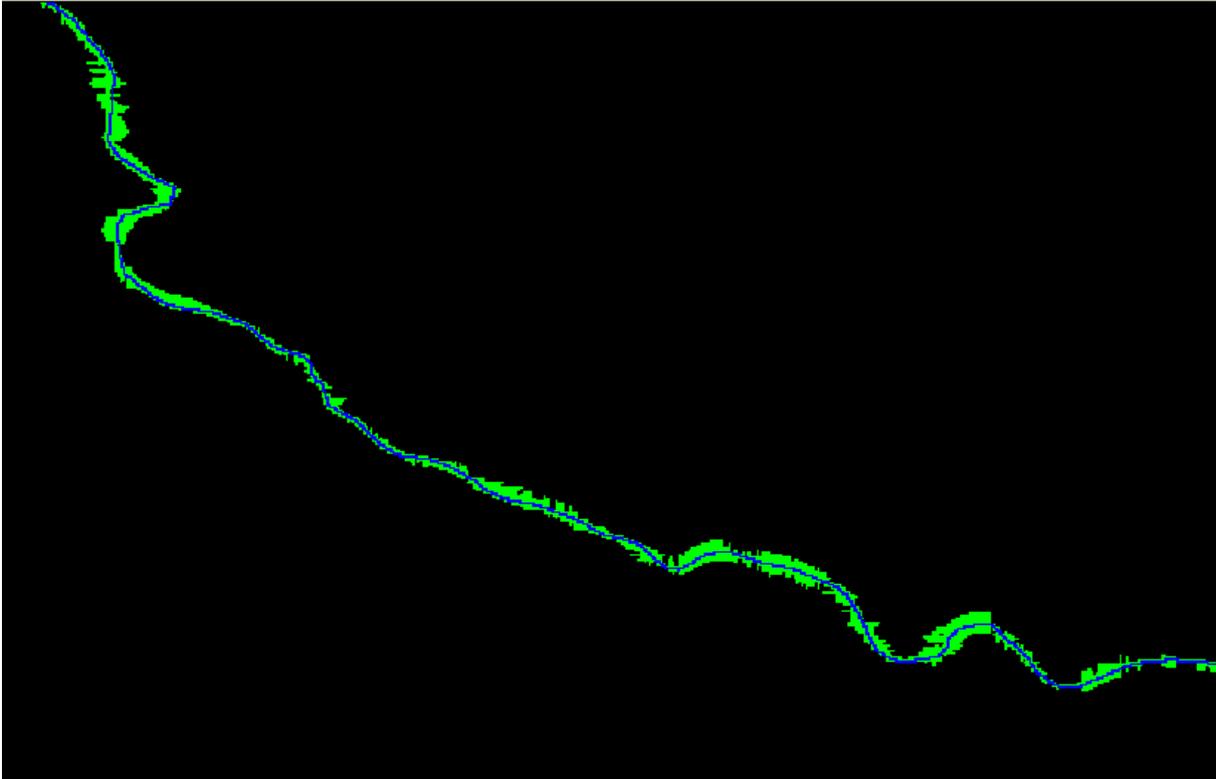
```

<?xml version="1.0"?>
<wps:Execute service="WPS" version="1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://geoserver.itc.nl:8080/wps/schemas/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>org.n52.wps.server.algorithm.snes4s.Watsch</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>WaterLevel:LevelNumber</ows:Identifier>
      <wps>Data>
<wps:LiteralData>17@@23@@1@@21@@24@@2@@13@@25@@3@@14@@27@@2</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps>DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="true">
      <wps:Output asReference="true">
        <ows:Identifier>FLOOD_RASTER</ows:Identifier>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>

```

### Listing 6: WATSCH-Request

Der Prozess liefert eine einfache textuelle Benachrichtigung: "Process was successful". Ein Beispielresultat ist in Abbildung 6 dargestellt.



**Abbildung 6: berechnete Überflutungsflächen**

Blaue Farbe entspricht Ausschnitten mit Elbeverlauf und grüne Farbe der von einer Überflutung betroffenen Fläche mit einer Pixelgröße von  $83 \times 83 \text{m}^2$

Die Ergebnistraster werden im STORAGE abgelegt und nach jeder neuen Anfrage überschrieben. Die Ergebnisse von Watsch sind befriedigend, da sie ausschließlich zur Orientierung dienen.

### **3.4. Web Map Service**

Für die Visualisierung dieses Projektes wurde der von der University of Minnesota entwickelte UMN Mapserver verwendet [WWW04]. Der Web Map Service greift auf das Ergebnistraster zu und lädt dieses und weitere Layer in eine Website. Auf dieser können die Daten visualisiert werden. Ein Zugriff durch einen standardisierten WMS-Client, wie zum Beispiel Sachsenatlas [WWW05], ist ebenso möglich. Nach Eingabe der URL des Mapservers und der Mapfile können auch hier die Ergebnisse visualisiert werden. Bei beiden Varianten greift der UMN Mapserver auf das erstellte Mapfile zu. Dieses regelt die Darstellung der einzelnen Ebenen, deren Layout und Projektion. Im Folgenden wird ein Ausschnitt aus dieser Mapfile gezeigt, in dem das erstellte Ergebnistraster „dangermap.asc“ verwendet wird.

```

MAP
NAME          'Wasserbeobachtung'
IMAGETYPE     JPEG
IMAGECOLOR    128 128 128
EXTENT        4483214 5552857 4717162 5739012
SIZE          800 640
SHAPEPATH     'data'
SYMBOLSET     'symbols/symbols.sym'
FONTSET       'fonts/fonts.list'
...
LAYER
NAME          'DangerMap'
TYPE          RASTER
STATUS        ON
DATA          'dangermap.asc'

METADATA
'wms_title'  'danger'
END

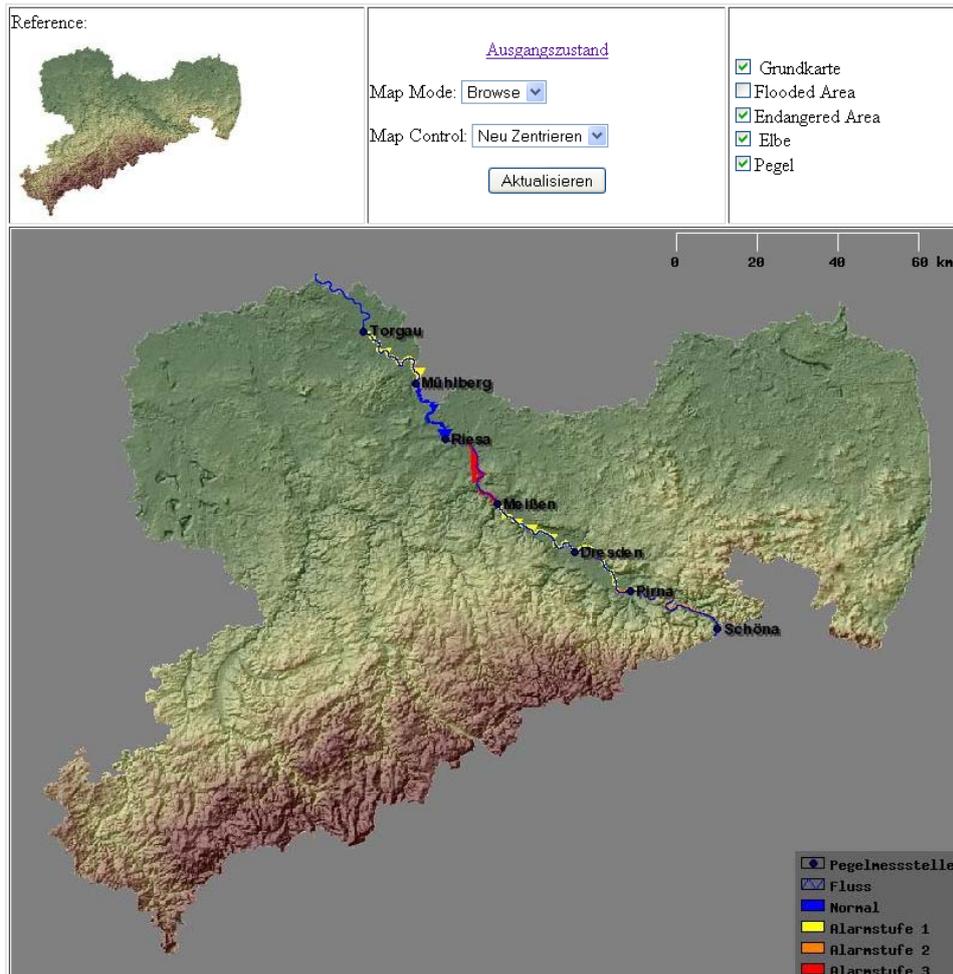
PROJECTION
'init=epsg:4326'
END

CLASSITEM '[pixel]'
CLASS
NAME          'Normal'
EXPRESSION ([pixel]=1)
STYLE
COLOR 0 0 255
END
END
CLASS
NAME          'Alarmstufe 1'
EXPRESSION ([pixel]=1)
STYLE
COLOR 255 255 0
END
END
CLASS
NAME          'Alarmstufe 2'
EXPRESSION ([pixel]=2)
STYLE
COLOR 255 128 0
END
END
CLASS
NAME          'Alarmstufe 3'
EXPRESSION ([pixel]=3)
STYLE
COLOR 255 0 0
END
END
END #ENDE LAYER DangerMap

```

### Listing 7: Ausschnitt aus dem Mapfile

Ein Geländemodell von Sachsen, welches entsprechend der Höhen farblich ansprechend gestaltet ist, und der Flussverlauf der Elbe werden obligatorisch in den Viewer geladen. Der Nutzer kann nun verschiedene Ebenen hinzuschalten und in das Bild hineinzoomen. Dies wird in Abbildung 7 dargestellt.



**Abbildung 7: Darstellung der einzelnen Layer im Browserfenster**

Durch Selektion der Checkbox „Endangered Area“ (Gefährdete Gebiete) können die Hochwasserstufen hinzugeladen werden, welche sich an den offiziellen Vorgaben des Landeshochwasserzentrums Sachsen[WWW02] orientieren und in verschiedenen Farben dargestellt sind. Die Alarmstufe 1 erhält die Farbe Gelb, Alarmstufe 2 Orange und Alarmstufe 3 Rot. Eine hellblaue Färbung erhält der Fluss bei normalem Pegelstand. Weiterhin können die überfluteten Flächen durch Auswählen der Checkbox „Flooded Area“ (überflutete Gebiete) in Grün dargestellt werden. Die Legende verändert sich, je nachdem welcher Layer angeschaltet wurde.

Somit ist es möglich sowohl die Warnstufen als auch die überfluteten Flächen darzustellen.

Bei einer höheren Auflösung der Grundlagendaten, beispielsweise einer Topographischen Karte oder eines DGM mit größerer Auflösung sind natürlich die überfluteten Flächen sehr viel besser darstellbar. Aufgrund der Einteilung in Layer innerhalb der Mapfile, können neue Ebenen leicht eingebunden werden, somit ist eine Erweiterung um diese hochauflösenden Karten problemlos durchführbar.

## 4. Zusammenfassung, Fazit, Ausblick

Das gegenwärtige Hochwassersystem ermöglicht es, aktuelle Pegelstände über PEGELONLINE abzufragen. Dies geschieht über einen eigens programmierten SAS-ähnlichen Dienst, welcher automatisch, in einem Zeitintervall von 15 Minuten, einen SOS abfragt und aufgrund der erhaltenen Werte Gefahrenstufen erstellt. Anschließend wird ein WPS aufgerufen, der mittels eines neuen Prozesses (WATSCH-Algorithmus) die Prozessierung und Erstellung zweier ASCII-Files übernimmt, welche die Gefahrenstufen und Überschwemmungsgebiete darstellen. Diese Ergebnisse werden über eine Website bzw. einen standardisierten WMS-Client visualisiert. Somit ermöglicht die WATSCH-Architektur, für Sachsen schnelle Aussagen über das vorherrschende Gefahrenpotenzial eines Hochwassers an der Elbe zu ermitteln und diese Berechnungen für weitere Entscheidungen verfügbar zu machen. Aber auch interessierte Privatpersonen können sich sowohl über die momentane Hochwasserstufe, als auch über eventuell überflutete Areale entlang der Elbe informieren.

Ungeklärt sind noch die Fragen der Speicherung der Datenmengen der Ergebnisse und die Visualisierung für den Client. Auch die Problematik der Fehlermeldungen, welche bei Anfragen an den SOS entstehen, wenn man hierbei Umlaute in Pegelmessstellen verwendet, muss weiter untersucht werden. Des Weiteren müssen die als Rohdaten übermittelten Pegelstände einer erweiterten Kontrolle unterzogen werden, damit Fehler in diesen Werten erkannt werden und nicht in Berechnungen einfließen. Diese Datenlücken in den PEGELONLINE Daten werden in dieser prototypischen GDI nicht berücksichtigt. Für die Weiterentwicklung dieses Projektes ist es dringend notwendig einen entsprechenden Prüf- und Korrektur-Algorithmus zu integrieren, welcher die Daten auf Plausibilität prüft. Das folgende Beispiel zeigt sehr deutlich, wie fehlerhaft die genutzten Daten sein können.

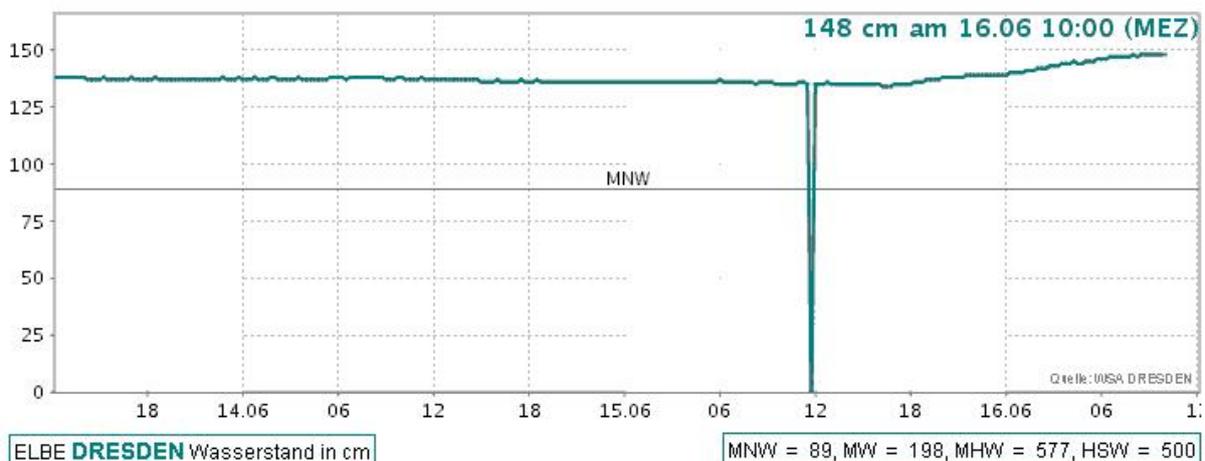


Abbildung 8: Fehler in den Daten bei Pegelonline

Das Projekt konnte zeigen, dass eine automatisierte Prozessierung durch entsprechende Komponenten innerhalb einer GDI, ohne den Eingriff eines Anwenders, möglich ist.

Für die Visualisierung der Ergebnisse müssen großmaßstäbigere, digitale, topographische Karten eingebunden werden, da erst dadurch die eigentlichen Überflutungsgebiete dargestellt werden können. Die aktuelle Visualisierung vermittelt, aufgrund der niedrigen Auflösung des DGM und fehlender Kartengrundlagen einen allgemeinen Überblick über die Hochwassergefahr.

Eine Möglichkeit der Erweiterung bietet die Benachrichtigung der betroffenen Bewohner und Zuständigen per SMS oder E-Mail. Ebenfalls möglich sind die Protokolle Phone, Fax und http. Wird eine bestimmte Alarmstufe erreicht, sollen die betroffenen Personen alarmiert werden, damit diese entsprechende Maßnahmen ergreifen können. Aufgrund des modularen Aufbaus der Architektur ist es möglich, einen WNS (Web Notification Service) einzurichten. Hierbei wird auf einen von 52north bereit gestellten WNS zu gegriffen. Die Namen registrierter Nutzer können dann in einer Datenbank gespeichert werden (Open Source Datenbank Exist). Die für den jeweiligen Pegelmesser gefährliche Hochwasserstufe wird per String durch den SAS an den WNS übergeben. Der SAS steuert also den WNS. Im Alarmfall wird mittels der Funktion DoNotification per XML die Anweisung für eine Alarmmeldung ausgelöst, die die entsprechenden Pegelmessstellen und die Alarmstufe enthält. Hierzu wird das entsprechende XML-File neu geparkt und anschließend per Http-Post an den WNS gesendet. Der WNS soll dann eine E-Mail oder eine SMS an die registrierten Nutzer senden. Der WNS unterliegt ebenfalls den Spezifikationen der OGC.

Eine Optimierung der Visualisierung ist möglich, indem man nicht nur ein Geländemodell nutzt, sondern auch Gebäudemodelle. Mit Hilfe der ermittelten Daten lassen sich dann die entsprechenden Flächen überfluten. Dadurch wird noch eindrucksvoller deutlich, mit welcher Hochwassergefahr gerechnet werden muss. Es können auch die verschiedenen Hochwasserstufen simuliert und visualisiert werden, damit die Gefahr für bestimmte Gebiete hervorgehoben wird. Dazu sind hochauflösende Daten nötig. Ein DGM mit einer Auflösung von maximal 10 Metern sowie eine 3D-Visualisierung der betroffenen Städte, zum Beispiel mittels CityGML sind erstrebenswert.

WATSCH bietet somit einen nützlichen Dienst zur Hochwasserwarnung und Planung, sowohl für öffentliche Entscheidungsträger, als auch für Privatpersonen. Es ist für den Nutzer einfach handhabbar und ermöglicht durch seine Visualisierung auch unerfahrenen Personen einen sofortigen Überblick zu erhalten.

## 5. Literaturverzeichnis:

[OGC01] Open Geospatial Consortium Inc. (Hrsg.): Sensor Observation Service. 1.0. 2007

[WWW01] <http://www.pegelonline.wsv.de/gast/start>

[WWW02] [http://www.umwelt.sachsen.de/de/wu/umwelt/lfug/lfug\\_internet/wasser\\_9564.html](http://www.umwelt.sachsen.de/de/wu/umwelt/lfug/lfug_internet/wasser_9564.html)

[WWW03] <http://52north.org/maven/project-sites/wps/52n-wps-webapp/>

[WWW04] Dokumentation für den UMN Mapserver: <http://mapserver.org/index.html>

[WWW05] Sachsenatlas: <http://atlas.egov.sachsen.de/gps/erweitert.jsp>

[WWW06] [http://52north.org/index.php?option=com\\_content&view=category&layout=blog&id=29&Itemid=44](http://52north.org/index.php?option=com_content&view=category&layout=blog&id=29&Itemid=44)

[WWW07] <http://exist.sourceforge.net/>

Informationen zur Klimaveränderung:

Klimamonographie für Sachsen (KLIMOSA) – Untersuchung und Visualisierung der Raum- und Zeitstruktur diagnostischer Zeitreihen der Klimaelemente unter besonderer Berücksichtigung der Witterungsextreme und der Wetterlagen. In: Sächsisches Staatsministerium für Umwelt und Landwirtschaft (Hrsg.): Sachsen im Klimawandel – Eine Analyse. Eigenverlag, Dresden, 211 S.

Bernhofer C, Goldberg V, Franke J, Häntzschel J, Harmansa S, Pluntke T, Geidel K, Surke M, Prasse H, Freydank E, Hänsel S, Mellentin U, Küchler W (2008)