Automating the web publishing process of environmental data by using semantic annotations

Jürgen Moßgraber
Fraunhofer IOSB
Fraunhoferstraße 1
76131 Karlsruhe
juergen.mossgraber@iosb.fraunhofer.de

Désirée Hilbring
Fraunhofer IOSB
Fraunhoferstraße 1
76131 Karlsruhe
desiree.hilbring@iosb.fraunhofer.de

ABSTRACT

Large amounts of environmental data are still hidden away in databases only accessible by domain experts. There is the need to make this data available to other experts for further data fusion. To implement standards like the Sensor Observation Service (SOS) huge efforts on the side of environmental agencies are required. At the same time, the pressure to make this data available to the interested public arises in form of Linked Open Data (LOD). This additional demand requires even more programming resources to fulfill the new requirements and interfaces. In this paper, we describe a system architecture, which simplifies and automates this problem of publishing environmental data in different data models. Ontologies are applied to map the different models' syntax and semantics. Additionally, we present a proof-of-concept implementation supporting both SOS and LOD interfaces.

Keywords
Linked Open Data, Semantic, Sensor Observation Service (SOS), Web Publishing, Software Architecture.

1. INTRODUCTION
Geographical data play an increasingly important role in many application fields. Especially in the environmental domain, large amounts of measurement data are stored in expert databases. However, these are not accessible to other public bodies and to the citizens. One reason for this is, among others the lack of use of standards for accessing the data.

The challenge is not to address a specific standard but the increasing number of standards that have to be supported by an environmental information system. Examples are standards of the Open Geospatial Consortium (OGC) such as Web Feature Service (WFS) and the Sensor Observation Service (SOS). At the same time, the pressure to make this data available to the interested public brings up the requirement to support also standards from the Linked Open Data (LOD) domain.

Huge efforts on the side of environmental agencies would be required to support all of them, which is way beyond the budgets of these institutions. Not only the plain programming work needs to be considered but also the mapping of the syntax and semantics of the different data models. The difficulty lies especially in the

semantics, which require time-consuming discussions between domain and IT experts. Furthermore, the domain experts need to be in control of which data are published. Since this is daily business, no programming should be required.

In the following section 3, relevant standardized and proprietary service interfaces for environmental data and their data models are described. The challenges of mapping data models are explained in section 4. After that, we present a method to simplify the task of mapping the data models by facilitating ontologies (section 5) and show a system architecture and experimental implementation based on our Extensible Database Application Configurator (XCNF) framework.

2. RELATED WORK
A lot of research has been executed in the area of mapping (data) models. Especially, mapping schemas of relational databases, which have been available for a long time, were in focus. A good overview of the state-of-the-art is given by [5]. More current research focuses on XML and ontology models [7] of which the later have the advantage providing the semantics of the model as well. In addition, mapping between these different kinds of models has been researched. However, until now there is no fully automatic mapping algorithm, which solves the problem 100% [6]. Therefore, we center the following work on simplifying the manual mapping of models by facilitating semantic annotations, which can be applied by a domain expert.

An overview of the state-of-the-art in Linked Data is given in [4]. Tools such as "D2R Server" [12] are used to publish data stored in relational databases. The data publisher defines a mapping between the relational schema of the database and the target ontology vocabulary with a declarative mapping language. Due to this static nature domain experts cannot apply changes easily. Exemplary works are described in [13] and [14].

3. RELEVANT INTERFACES AND DATA MODELS
The Open Geospatial Consortium (OGC) is concerned with the definition of standardized interfaces in the domain of geographical information and increasingly in the area of sensor data ("Sensor Web Enablement").

3.1 Sensor Observation Service (SOS)
The SOS specification [1] provides operations to retrieve sensor data and specifically "observation" data.

The observations themselves are defined by another OGC standard: the Observation and Measurement Model (O&M) [2]. Observations described by O&M can be seen directly as

measurements from sensors, but they can also represent other data structures.

## 3.2 Web Feature Service (WFS)

The Web Feature Service (WFS) represents a change in the way geographic information is created, modified and exchanged on the Internet. Rather than sharing geographic information at the file level using File Transfer Protocol (FTP), for example, the WFS offers direct fine-grained access to geographic information at the feature and feature property level [3].

3.3 Linked Open Data (LOD)

In computing, linked data (often capitalized as Linked Data) describes a method of publishing structured data so that it can be interlinked and become more useful. It builds upon standard Web technologies such as HTTP, RDF and URIs, but rather than using them to serve web pages for human readers, it extends them to share information in a way that can be read automatically by computers. This enables data from different sources to be connected and queried [4].

3.4 XCNF

XCNF (eXtensible database application CoNFigurator) is a Java based client/server framework by Fraunhofer IOSB for developing information systems for time series analysis. While the framework can be applied to any domain, we mainly apply it to the domains of water management and water quality. Most of the data are time series with spatial relationships.
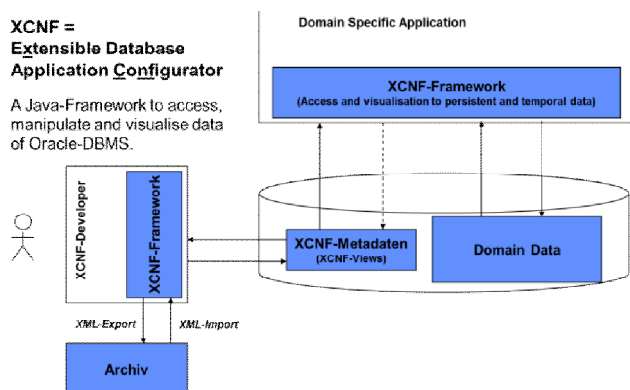
XCNF uses a proprietary metadata model, which not only describes the data but also the layout of input forms and search masks. XCNF uses a concept called *View*. A View provides access to a part of one or more connected databases quite similar to a database view. In contrast to a database view, it provides additional annotations to add semantics to its attributes and link attributes to other views. This has the consequence that every end user creates or extends its own data model by creating or modifying a XCNF View.

4. PUBLISHING AND MAPPING OF DATA MODELS

Figure 1 depicts the problem that needs to be solved. Several interface standards with their specific data models have to be mapped with respect to their syntax and semantics to a proprietary data model of an existing information system in the backend.

As noted above in 3.4 XCNF provides a metadata model to describe data models, which can change dynamically. This means that we cannot apply a once-only mapping of the models. Instead, the mapping always needs to be adjusted if an end user makes a change and therefore needs to be dynamic too.
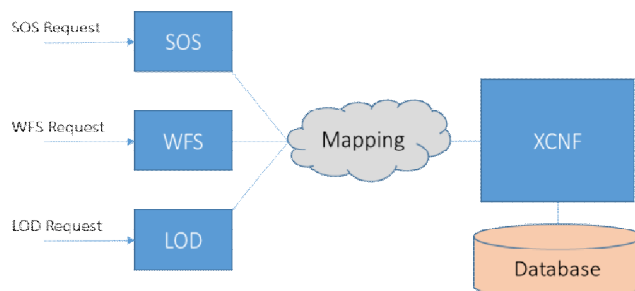
Figure 1. Required mapping for accessing time series with different standards

4.1 Concept

To publish data from XCNF the existing features are used and extended by ontology annotations:

- An ontology is required for each interface which should be supported (SOS, WFS, etc.). The ontology must contain the specific concepts and properties to describe the model. Preferably, an existing ontology should be reused.

- All required concepts and their accompanying properties contained in the used ontology must be mapped to existing XCNF Views and their attributes. This is done by annotating them with the URIs of ontology resources. For example if available datasets shall be published as SOS Observations the appropriate XCNF View is annotated with #Observation (this is only the hash part of the URI for better readability). The attributes of the view need to be annotated with properties from the ontologies too, e.g. #hasValue, #hasTime, etc.

- Other interfaces (e.g. LOD) can be supported by annotating the views with URIs from the ontology used for the other interface.

- The specific publishing service (SOS, WFS, etc.) can now read all of the entries from the related XCNF Views, annotated by concepts of its ontology.

- Since the structure is given by the ontology the service can relate multiple views which belong together.
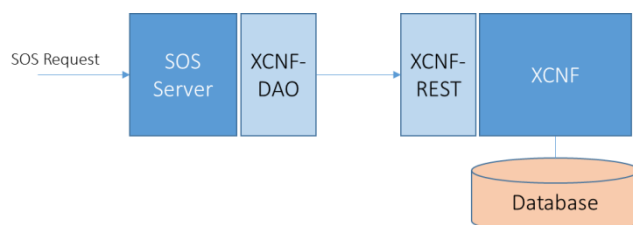
Figure 2. Architecture for SOS accessing XCNF

5. ARCHITECTURE AND IMPLEMENTA-TION

The architecture and implementation of an SOS interface is described in the following. Other interfaces can be supported in the same way.

The following figure depicts the components of the system that will be described in the following sub-sections:

5.1 Ontology

Several translations to an ontology are available for the Observation and Measurement Model (O&M) [8]. Since they tend to be rather complex we have extracted only those concepts and properties which were necessary for the mapping. The following concepts and their properties are used:

- Observation

    hasObservedProperty
    measuredByProcedure
    hasValue
    relatesToFeatureOfInterest
    hasTime
    hasUnit

- Phenomenon

    hasName
    hasID

- Procedure

    hasName
    hasID

- FeatureOfInterest

    hasName
    hasID
    hasNorthing
    hasEasting

5.1.1 Mapping Example

Our test data is taken from the *Fachinformationssystem Gewässer Qualität* (FISGeQua) which contains water quality data from all measurement stations of the German state Baden-Württemberg.

The following tables show how the XCNF-Views of FISGeQua have been annotated with resources from the SOS ontology to support the SOS interface:

Ontology Concepts and Properties	XCNF Views and Attributes
#Observation	GEW_MESSWERT_GUETE, GEW_PROBE
#hasObservedProperty	GEW_MESSWERT_GUETE.PARAMETER_NR
#measuredByProcedure	GEW_MESSWERT_GUETE.MESSVERFAHREN_NR
#hasValue	GEW_MESSWERT_GUETE.MESSWERT
#relatesToFeatureOf-Interest	GEW_PROBE.PNST_NR
#hasTime	GEW_PROBE.DATUM

#hasUnit	GEW_MESSWERT_GUETE.DIMENSION_NR
#Phenomenon	GEW_PARAMETER
#hasID	GEW_PARAMETER.BASIS_NR
#hasName	GEW_PARAMETER.KURZNAME
#Procedure	UIS_SL_MESSVERFAHREN
#hasName	UIS_SL_MESSVERFAHREN.LANGNAME
#hasID	UIS_SL_MESSVERFAHREN.MESSVERFAHREN_NR
#FeatureOfInterest	GEW_PNST, GEW_MST, GEW_POSITION
#hasName	GEW_MST.NAME
#hasID	GEW_PNST.PNST_NR
#hasNorthing	GEW_POSITION.HW
#hasEasting	GEW_POSITION.RW

Since the example above contains German words and acronyms here is a little glossary:

- MESSWERT: measurement
- PROBE: observation
- GUETE: quality
- MESSVERFAHREN: measurement procedure
- DATUM: date
- KURZNAME: short name
- LANGNAME: long name
- HW + RW: the geo location

5.2 SOS Requests and Results

By facilitating the above mapping, it is possible to receive data from the FISGeQua database to make it accessible via an SOS interface. A typical SOS request can be formulated in the following way:

- Give me all available data which matches the following conditions:
    - The #Phenomenon shall be water temperature.
    - The #Procedure, which has been used to determine the water temperature is electrometry.
    - The data has been measured in the time range of 2nd to 4th January 2005.
    - The #FeatureOfInterest, which defines the spatial region, is the measuring point with id 1051.

This request in the SOS XML notation looks like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservation service="SOS" version="2.0.0"
    xmlns:sos="http://www.opengis.net/sos/2.0"
    xmlns:fes="http://www.opengis.net/fes/2.0"
```

```
   xmlns:gml="http://www.opengis.net/gml/3.2"
   xmlns:swe="http://www.opengis.net/swe/2.0"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:swes="http://www.opengis.net/swes/2.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/sos/2.0
http://schemas.opengis.net/sos/2.0/sos.xsd">
   <!-- optional, multiple values possible -->
   <sos:procedure>10</sos:procedure>
   <!-- optional, multiple values possible -->
   <sos:offering>8289</sos:offering>
   <!-- optional, multiple values possible -->
   <sos:observedProperty>TW</sos:observedProperty>
   <!-- optional -->
   <sos:temporalFilter>
       <fes:During>

<fes:ValueReference>phenomenonTime</fes:ValueReference>
           <gml:TimePeriod gml:id="tp_1">
               <gml:beginPosition>2005-01-
02T14:00:00.000+01:00</gml:beginPosition>
               <gml:endPosition>2005-01-
04T15:00:00.000+01:00</gml:endPosition>
           </gml:TimePeriod>
       </fes:During>
   </sos:temporalFilter>
   <!-- optional, multiple values possible -->
   <sos:featureOfInterest>1051.0</sos:featureOfInterest>
   <!-- optional -->

<sos:responseFormat>http://www.opengis.net/om/2.0</sos:responseFor
mat>
</sos:GetObservation>
```

As you can see, the request contains no FISGeQua specific nomenclatures. Here is the response to this request:

```
<?xml version="1.0" encoding="UTF-8"?>
<sos:GetObservationResponse
xmlns:sos="http://www.opengis.net/sos/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:om="http://www.opengis.net/om/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/sos/2.0
http://schemas.opengis.net/sos/2.0/sos.xsd
http://www.opengis.net/om/2.0
http://schemas.opengis.net/om/2.0/observation.xsd
http://www.opengis.net/sampling/2.0
http://schemas.opengis.net/sampling/2.0/samplingFeature.xsd
http://www.opengis.net/samplingSpatial/2.0
http://schemas.opengis.net/samplingSpatial/2.0/spatialSamplingFeat
ure.xsd">
  <sos:observationData>
    <om:OM_Observation gml:id="o_1377870212996">
      <gml:identifier
codeSpace="http://www.opengis.net/def/nil/OGC/0/unknown">abc8dbd3-
13ff-442a-9e23-80a9ec96881f</gml:identifier>
      <om:type
xlink:href="http://www.opengis.net/def/observationType/OGC-
OM/2.0/OM_Measurement"/>
      <om:phenomenonTime>
        <gml:TimeInstant gml:id="phenomenonTime_abc8dbd3-13ff-
442a-9e23-80a9ec96881f">
          <gml:timePosition>2005-01-
03T12:10:00.000+01:00</gml:timePosition>
        </gml:TimeInstant>
      </om:phenomenonTime>
      <om:resultTime xlink:href="#phenomenonTime_abc8dbd3-13ff-
442a-9e23-80a9ec96881f"/>
      <om:procedure xlink:href="10"/>
      <om:observedProperty xlink:href="TW"/>
      <om:featureOfInterest xlink:href="1051"/>
      <om:result xmlns:ns="http://www.opengis.net/gml/3.2"
uom="185" xsi:type="ns:MeasureType">6.5</om:result>
    </om:OM_Observation>
  </sos:observationData>
</sos:GetObservationResponse>
```

It says that a water temperature (TW) of 6.5°C has been measured at 1051 on January 3rd.

5.3 XCNF REST Service

This interface allows us to query the data of a XCNF system. It is implemented in a RESTful style [9]. All replies are JSON encoded. New methods are added to query the model ontologies and the annotations of the views. The service provides the following methods:

- **/xcnfrestservice/capabilities** Provides a list with all supported models/ontologies

- **/xcnfrestservice/capabilities/viewNames** Get the names of all published XCNF views, e.g.:

  ```
  {"viewNames":["GEW_MESSWERT_GUETE",
  "GEW_PROBE","GEW_PARAMETER","UIS_SL_M
  ESSVERFAHREN","GEW_PNST","GEW_MST","G
  EW_POSITION"]}
  ```

- **/xcnfrestservice/capabilities/mapping** Get the mapping of the views to the ontology concepts and properties. Note that multiple annotations from different ontologies could be applied if the data should be available via different interfaces! The following shows the mapping part for #Phenomenon and #FeatureOfInterest and with deleted URIs to keep it shorter::

  ```
  {"mappingStructure":{"viewMappingLi
  st":[{"viewName":"GEW_PARAMETER","con
  ceptNames":["#Phenomenon"],"mappingLi
  st":[{"columnName":"KURZNAME","concep
  t":"#hasName"}]},{"viewName":"
  GEW_PNST","conceptNames":["#FeatureO
  fInterest"],"mappingList":[{"columnNa
  me":"ID","concept":"#hasID"}]},
  {"viewName":"GEW_MST","conceptNames
  ":["#FeatureOfInterest"],"mappingList
  ":[{"columnName":"NAME","concept":has
  Name"}]},{"viewName":"GEW_POSITION","
  conceptNames":["#FeatureOfInterest"],
  "mappingList":[{"columnName":"RW","co
  ncept":hasEasting"},{"columnName":"HW
  ","concept":"#hasNorthing}]},...]}}
  ```

- **/xcnfrestservice/capabilities/model/?uri=*uri*** Get the ontology with the given URI.

- **/xcnfrestservice/data/concept/?uri=*uri*** Query for the data mapped to the concept with the given URI.

- **/xcnfrestservice/data/filter/?uri=*uri*&propertyList=*u ri*&valueList=*value*** Query for the data mapped to the concept with the given URI. The response is filtered with the properties provided in the additional parameters.

5.4 SOS server and XCNF-DAO

52°North, a company developing Geospatial Open Source software, provides a SOS implementation based on Java. As 52°North develops the reference implementation for the OGC SOS specification we chose their software (see http://52north.org/communities/sensorweb/sos/index.html) as the basis for our proof-of-concept implementation.

We chose an early access version 4 (4.0.0 Beta2) of the software since it provides much better modularity than version 3. In this new version there is now a defined way for plugging in your own data access into the server via so-called Data Access Objects (DAO). Out of the box it retrieves its' data from a relational database in a proprietary format which did not fit our

needs since we wanted a direct access to the data stored in an XCNF server for performance reasons.

The implemented XCNF-DAO plugs into the SOS server. It retrieves the data from the XCNF-REST service by utilizing the SOS ontology annotations. The retrieved data is handed over to the SOS server, which handles the syntax formatting and encoding (see Figure 2).

6. DISCUSSION

6.1 Distribution of Concept Properties over several Views

Analyzing the mapping example described in section 5.1.1, one can see that it often happens that the properties of one concept need to be mapped to attributes, which belong to several different XCNF Views. Here is an example:

The #hasObservedProperty property of an #Observation can be found in the XCNF View GEW_MESSWERT_GUETE while the property #relatedToFeatureOfInterest is contained in XCNF View GEW_PROBE.

Requesting the #Observation concept via the integrated XCNF View filtering option filtered with #hasObservedProperty=A or requesting the #Observation concept filtered with #relatedToFeatureOfInterest=B will lead in both cases to too many results if the second filter option is missing.

To support the filtering mechanism of the XCNF REST Service /xcnfrestservice/data/filter/?uri=uri&propertyList=uri&valueList= value, the implementation must provide an additional filtering operation before returning the results via the URI.

6.2 Reducing the Amount of Data to be published

Often only subsets of the data in the database are foreseen for publishing. Therefore, we need a mechanism for defining which subsets of the data in the database can be delivered via the XCNF REST Service.

XCNF foresees the possibility to create so called BDOs ("Benutzerdefiniertes Objekt"), which are user-defined objects. It is possible to create a BDO which reduces the amount of data in the database to the subset which shall be published, e.g. via defining specific measurement points, a specific time range or specific phenomena.

Currently we consider implementing the following mechanism:

1. The #Observation concept in the ontology needs to be extended with a new property #hasBDO.

2. The owner of the database needs to define a specific BDO for the data subset to be published.

3. This BDO needs to be annotated with #hasBDO.

4. The implementation of the XCNF REST Service /xcnfrestservice/data?uri=uri and its filter mechanism need to be extended with an additional filter (propertyList: #hasBDO, valueList: #8289) which is not seen from outside the XCNF Rest Service.

6.3 Ideas for Integrating Linked Open Data

The possible support of Linked Open Data was another idea we had. Therefore, the architecture foresees the possibility to support several interfaces. The additional support of LOD would require that we provide our data in RDF or OWL format. For our current implementation the following two possibilities exist:

- Either the XNCF REST Service would need to map its responses to RDF or OWL or

- we use our extended SOS implementation and map the resulting XML Observation Collection to RDF or OWL.

The first approach will be faster, because it saves one mapping step. However, it will contain a proprietary solution while the second approach can use existing geospatial standards and might reuse mechanisms described in [10] and [11].

6.4 Adapting the approach to other systems

In this paper, we used the XCNF framework as an example to demonstrate our approach but it can be adapted to other systems as well. To facilitate that, the following steps need to be taken:

1. Enable annotation of your relational data (could be done with a standard relational mapper).

2. Support multiple mappings (ontologies)

3. Add the possibility for the user to dynamically change the mapping

4. Provide the means to publish only selections of the data (done by XCNF views in our approach).

7. CONCLUSION

In this paper, we presented a concept for dynamically mapping data models of domain expert systems to different interface standards by annotating the model with resources from an ontology. In contrast to static approaches like D2R shown in the related work section, this allows for quicker adaptions to new requirements by the domain expert.

The described implementation shows that the concept is applicable to a real world scenario. In the future, we will work on removing the discussed drawbacks and improve the user interface for executing the mapping. For example, ontology properties for an annotation could be suggested to the user depending on the data type and the selected ontology concept. Furthermore, since XCNF views already contain some metadata annotations it is interesting to explore to what degree the mappings can be created automatically.

8. REFERENCES

[1] Bröring, A., Stasch, C., Echterhoff, J. (Ed.) 2012. OGC® Sensor Observation Service Interface Standard, Version: 2.0, Open Geospatial Consortium Inc., 12-006

[2] Cox, C. (Ed.) 2011. Observations and Measurements - XML Implementation, Version:2.0, Open Geospatial Consortium, 10-025r1

[3] Vretanos, P. (Ed.) 2010. OpenGIS Web Feature Service 2.0 Interface Standard, Open Geospatial Consortium, OGC 09-025r1 and ISO/DIS 19142

[4] Bizer, C., Heath, T., Berners-Lee, T. 2009. Linked Data - The Story So Far, International Journal on Semantic Web and Information Systems 5 (3): 1–22. doi:10.4018/jswis.2009081901

[5] Bellahsene, Z., Bonifati, A., Rahm, E. 2011. Schema Matching and Mapping, Springer Verlag, doi:10.1007/978-3-642-16518-4

[6] Bernstein, P., Madhavan, J., Rahm, E. Generic Schema Matching, Ten Years Later, 37th International Conference

on Very Large Data Bases (Seattle, Washington August 29th - September 3rd 2011)

[7] Gross, A., Hartung, M., Thor, A., Rahm, A. 2012. How do computed ontology mappings evolve?, Joint Workshop on Knowledge Evolution and Ontology Dynamics (ISWC 2012)

[8] Compton, M., et al. 2012. The SSN ontology of the W3C semantic sensor network incubator group, Web Semantics: Science, Services and Agents on the World Wide Web 17, p25-32.

[9] Fielding, R., Taylor, R. 2002. Principled Design of the Modern Web Architecture, ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery) 2 (2): 115–150, doi:10.1145/514183.514185, ISSN 1533-5399

[10] Page, K., De Roure, D., Martinez, K., Sadler, J., Kit, O. 2009. Linked Sensor Data: RESTfully Serving RDF and GML. In Proceedings of 2nd International Workshop on Semantic Sensor Networks (SSN09), conjunction with the 8th International Semantic Web Conference, ISWC 2009, Washington, DC, USA, October 2009; CEUR-WS: Aachen, Germany, 2010; Volume 522, pp. 49–63.

[11] Probst, F., Gordon, A., Dornelas, I. 2006. Ontology-based Representation of the OGC Observations and Measurements Model, Open Geospatial Consortium

[12] Bizer, C., Cyganiak, R. 2006. D2R Server - Publishing Relational Databases on the Semantic Web. Poster at the 5th International Semantic Web Conference (ISWC2006)

[13] Moraru, A., Fortuna, C., Mladenic, D. 2011. A System for Publishing Sensor Data on the Semantic Web. Journal of Computing and Information Technology - CIT 19, 2011, 4, 239–245, doi:10.2498/cit.1002030

[14] Page, K., Frazer, A., Nagel, B., De Roure, D., Martinez, K. 2011. Semantic Access to Sensor Observations through Web APIs, Fifth IEEE International Conference on Semantic Computing