

# **Processing chains with OGC-standards on top of an European MODIS satellite data archive**

**Jonas Eberle <sup>(1)</sup>, Dr. Christian Strobl <sup>(2)</sup>, Prof. Dr. Christiane Schmullius <sup>(1)</sup>**

*<sup>(1)</sup> Friedrich-Schiller-University, Department for Earth Observation*

*Loebdergraben 32, 07749 Jena, Germany*

*E-Mail: jonas.eberle@uni-jena.de*

*<sup>(2)</sup> German Aerospace Center (DLR), Remote Sensing Data Center*

*Muenchnerstr. 20, 82334 Wessling, Germany*

*E-Mail: christian.strobl@dlr.de*

## **ABSTRACT**

With the implementation specification of the Web Processing Service (WPS), a standard to process geodata in the web is available since 2007. The new possibilities in web-based geographic information systems allow an interoperable processing of different data based on service-orientated architectures. This paper describes the implementation of a Web Processing Service to derivate products from MODIS satellite data. Software including a WPS was evaluated with focus of handling large data. Several processes were developed to create products for this satellite type, available on the web. On top of the Web Processing Services, software for implementing and executing processing chains as a Web Processing Service was evaluated and several chains were implemented. The Remote Sensing Data Center at the German Aerospace Center (DLR) owns a MODIS satellite data archive covering European states. An automatically running service is processing near-real time data downloaded direct after acquisition. An interface to get and process archived data was integrated in the developed web-based information system. The user of this system can search interactively for archived MODIS data and process it with the developed Web Processing Service and processing chains. The software evaluation shows that there is additional work to do to improve the WPS standard. Specific profiles for working with large raster data are needed to define standardised rules. For processing chains no standardised data format describing these chains exist, but it is also possible to create processing chains with several software solutions. This paper shows that there is a high potential in the usage of Web Processing Services and processing chains to add values for data archives with interoperable services.

Keywords: process chains, web processing, ogc, webgis, modis, services

## **INTRODUCTION**

Since spatial data are distributed in the web over specifications like Web Map Service, Web Feature Service and Web Coverage Service, it is a further step to process these data. Web-based processing of spatial data is more and more on the way since the OpenGeoSpatial Consortium (OGC) published a specification to handle web-based processing. This new possibilities in web-based information systems allow an interoperable processing of data based on service-orientated architectures. This paper is based on a master thesis written at German Aerospace Center, Remote Sensing Data Center, and University of Jena, Department for Earth Observation. The main objective was to analyse the handling of remote sensing data in the field of providing and processing in a web-based geographic information system.

This paper is structured in the chapter "Introduction" with basics, status of research and requirements, the "Web-based processing system" showing the technical layout with spatial data infrastructure as well as software for implementing the OGC Web Processing Service and process chains. The third part describes the developed processes and process chains based on remote sensing data and automatically derived

products. Finally the developed web-based information system with its components is described and at last the scientific findings will be summarized and discussed.

## **Basics**

Basics for this paper are the work with service-orientated architectures especially with REST-like services and Spatial Data Infrastructures includes OGC and ISO standards. Furthermore basics for the work with process chains have to be known.

### **Service-orientated architectures**

Nowadays web-based information systems are based on service-oriented architectures. Services can be implemented by SOAP (“Simple Object Access Protocol”) or by using HTTP REST (“Representational State Transfer”) specification. Services from OGC are commonly using a REST-like specification with plain XML and HTTP-GET using KVP encoding or HTTP-POST using XML encoding to transfer data and information. Additionally some specifications from OGC can’t be used with SOAP. For these reasons the usage of HTTP-GET and HTTP-POST was preferred in the following developments.

### **Spatial Data Infrastructures**

Building-up spatial data infrastructures the usage of standards from OGC and ISO has to be studied. For web-based processing especially the specification for an OGC Web Processing Service needs to be mentioned. This specification is based on processes and has three methods: “GetCapabilities”, “DescribeProcess” and “execute”. “GetCapabilities” lists all published processes, “DescribeProcess” lists further details (e.g. inputs and outputs) of a given process and “execute” starts the execution of a process with given input data. If calling “execute” the user can decide whether the process will be synchron or asynchron executed. Asynchrony means that the process runs in the background and the user can pull the status of the executed process. A process contains several inputs and outputs with a defined quantity and format. It can be distinguished between literal, complex and bounding box data. Literal data are strings or numbers; complex data are binary and text files like XML [1].

### **Process chains**

Complex processings can be realised by chaining simple processes. In process chains these processes can then be executed parallel or after another; processes can depend on prior processes. [2] defines a process chain as “sequence of process services, where the execution of the first process is necessary for the second”. Three different types of chaining are mentioned there:

1. Transparent or user defined chaining
2. Semi-transparent or workflow-managed chaining
3. Opaque or aggregate service chaining

Chaining transparent the user calls the processes one after another. Using the semi-transparent chaining the processes were linked to one workflow, but the user knows which processes are inside the workflow. With the opaque chaining the user doesn’t know which processes are called. The workflow runs in the background.

Beside ISO 19119 the OGC WPS specification defines three solutions to chain processes [1]:

1. Usage of BPEL engine
2. WPS process as orchestration engine
3. Simple service-chaining within execute query

The third possibility is only for simple chaining of OGC-services. For complex chaining, the Business Process Execution Language [3], the WPS process or other software solutions has to be used. The Business Process Execution Language (BPEL) is an IT standard to chain complex processes based on SOAP with an extension for HTTP-GET/POST services. The second possibility can be solved with any software integrating the WPS standard or software only for chaining processes as a process following the OGC WPS specification.

## Status of research

Having a look at the status of research the web-based processing was divided into the research for Web Processing Services and the research for processing chains. Following questions should be answered at the end of this paper: What are the requirements for web-based processing of remote sensing data and can they be achieved with actual standards? How can the combination of OGC standards and processing of remote sensing data be realised? What requirements need to be solved for the web-based client?

### Web-based processing

Since early developments of the Web Processing Service standard from the OGC in 2005, first individual software solutions were developed. Two years later (2007) the first final version of OGC WPS was published and further software like 52 North WPS and pyWPS were developed. Already before the final release of this standard, researchers developed systems for web processing for example for the generalization of vector data [4] and the management of the resource water [5]. In this time first requirements from the users to chain processes and further improvements [4] were published. After publication of WPS standard as version 1.0.0 [6] said that the performance of a WPS must be improved. [7] were engaged in the processing of remote sensing data and showed that the main problem is the transport and the processing of large data amounts.

In 2009 [8] published a “research agenda for geoprocessing services” and defined three critical areas:

1. Service orchestration
2. Semantic descriptions
3. Strategies to improve performance

The aim of the service orchestration is to allow value added chains as well as the creation of new SDI business models based on external processes. Limiting factor of the fully automatic orchestration are the missing semantic descriptions of each process. Strategies to improve performance are in demand, for example by using GRID infrastructures, cloud computing or other mechanism. Further prospective research has to be for example in the areas of WPS profiles for specific domains [9].

### Chaining WPS processes

Since the beginning of the work with the WPS specification the first requirements for chaining processes came up. Already [4] remarked that consistent rules for process chaining are missing. In many projects the OASIS-standard BPEL is used, but problems with SOAP-compatibility within OGC-standards were always mentioned [10,11]. [12] proposed the usage of a WPS process itself for chaining processes. For the usage with BPEL [13] developed a XML transformation to create WSDL documents, [14] implemented a proxy to solve the missing WSDL documents for WPS processes. So there are many approaches to chain processes with BPEL. For other software solutions implemented a WPS process itself for chaining WPS processes, there is no standardised way for the creation and execution of chaining WPS processes.

## SDI AND DATA MANAGEMENT

A spatial data infrastructure is normally divided into three parts: the data layer, the service layer and the client layer (figure 1). The data layer describes the different data sources: the external remote sensing data archive, the internal data for processing and the geodatabase for created products for publishing. An interface (HMA/CSW) establishes a connection to the metadata catalogue of the archive. Several services are communicating with the data layer. The central component is a Web Processing Service, which communicates with the data archive, the internal data and the geodatabase. A service for executing processing chains communicates with the WPS and the geodata access service communicates with the geodatabase. These services provide interfaces like OGC WMS, WFS, WCS and WPS, so that a client can work with data and services. The services can then be implemented in a webGIS or other GIS software.

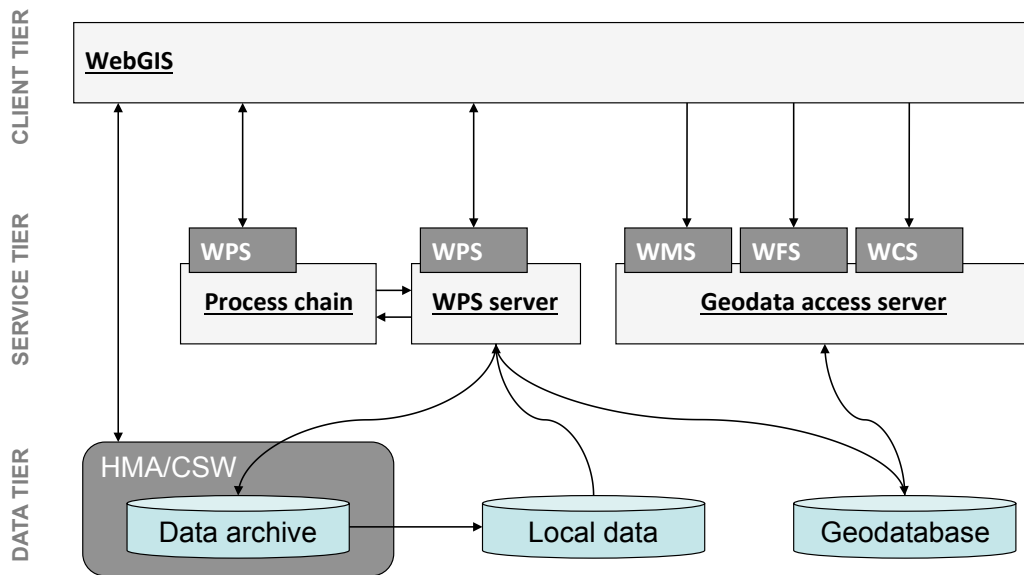


Figure 1. Technical schema of spatial data infrastructure.

A closer look to the data layer shows that the data management between several systems must also be defined (figure 2). Data from DIMS (“Data Information and Management System”), the data archive, can be pulled from PSM interface to the local WPS server and then transferred via WPS to the local processing data directory. This is divided into several products and outputs directories. The final products and outputs are stored in the webserver output directory or the geodatabase. The programming language of the chosen WPS is Python so that the interface to geodatabase is developed as Python module. Requested data from the archive (DIMS) will be transported via FTP in the “pulldir” directory. A WPS process transfers this data to the local processing directory, another process calls GRASS modules and finally stores the outputs as linked files in the webserver directory. Final products from automatic process chains are stored additionally in the specified geodatabase.

Any process can specify a processing directory where its local data is stored. This local directory is linked from the webserver for direct referencing the output data of the process. This mechanism limits the transfer of data within process chains from one process to the next.

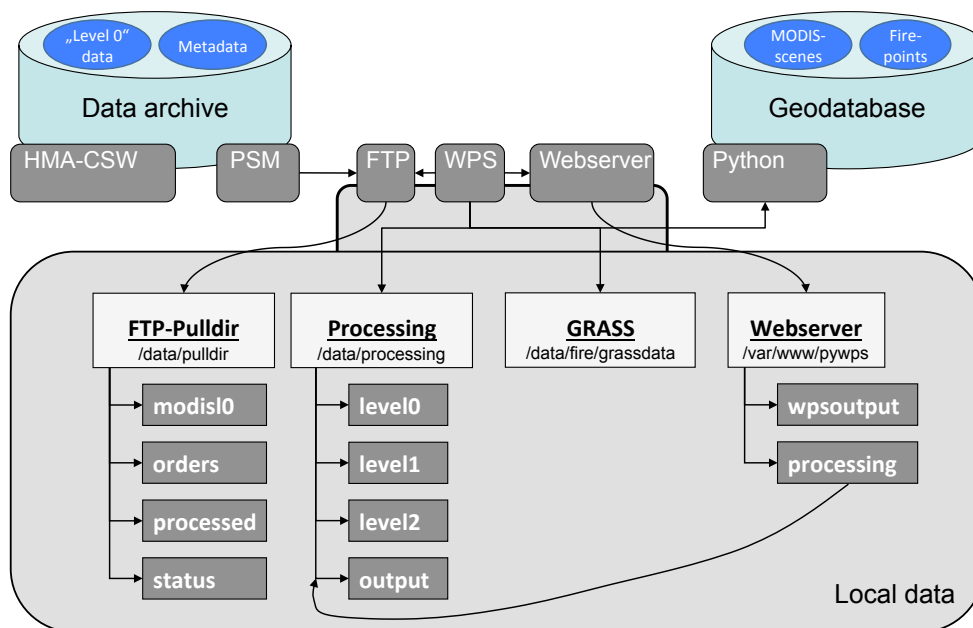


Figure 2. Technical schema of data management within the Web Processing Services.

## OGC Web Processing Service

To provide OGC Web Processing Services software solutions were tested and evaluated. The analysis of requirements for processing remote sensing data showed that it is important to regard the handling of large data and of good performance results. While working with remote sensing data, the processing of raster data is more important than the processing of vector data.

A closer look at 52 North WPS and pyWPS showed that both software solutions implement the OGC WPS specification well. 52 North WPS [15] is developed with Java, processes must be integrated in the given Java framework. Publishing processes in this Java framework is a complex task. After integrating the algorithms the sourcecode must be compiled, deployed and the Java servlet container must be restarted. These steps imply a high effort while developing. Especially while working with special data formats, it is impossible to use some standard framework functions, so that the implementation needs further effort, for example manual adjustments of XML files for generating the response for “getCapabilities” and “describeProcess” requests.

pyWPS [16] in contrast to 52 North WPS is simple constructed. It is based on Python language. One process is defined in one python file, which is available after linking it to the instance of the Web Processing Service. The need of compilation is not existent which keeps the development simpler. The processing of input data is flexible and extensions or further developments can be easily integrated. The process is developed by writing a process class with an initial and an executing method. The initial method describes the main properties of this process and defines the inputs and the outputs. The executing method accesses the information from input data, runs the process algorithm and changes values of the output data. Working with Python modules like OGR are available to work with geodata. Furthermore modules for connecting to databases, executing programs from commandline or accessing and editing special file formats can be used.

For providing the Web Processing Service pyWPS was taken because of its simplicity. But it should be possible to create these processes also with 52 North WPS software. For the implementation of pyWPS some adjustments were conducted to have better performance working with large data. To reference output data direct to the processing directory, the sourcecode was changed, so that output data can be referenced by a defined URL. Changing the value of an output data to this URL, the data will not be copied to the webservers output directory. Another function was integrated, with the result that a process can define which output has to be a reference data value. Normally the user defines which data comes back as reference or within the response.

## Orchestration Engines

Working with many processes lead to the requirement to chain processes and create workflows. Orchestration engines can execute these workflows. Several steps to derive final products based on remote sensing data are necessary. For most products the first steps are always the same. For this reason every single step was developed in one process. To chain these processes to one workflow software is needed to execute this workflow. An additional requirement for this software is the feature, that users should create these workflows on the web by linking process outputs and inputs dynamically. Three software solutions, mentioned in several former research works, were tested and evaluated: 52 North WPS Orchestration, Apache ODE with BPEL orchestration and the internal software DIMS-WPS at the German Aerospace Center.

52 North WPS Orchestration [17] is developed based on the 52 North WPS. A special process is integrated for handling workflows calling other WPS processes, which is assigned by an input parameter. Unfortunately this orchestration engine seems to be not concipated for the work with referenced raster data. All linked data between processes should be integrated in the requests and responses. But data couldn't be transferred from one process to the next using references. Additionally it had many problems in the combination of processes published by pyWPS caused a non-standard conform process requesting. The creation of workflows from a client was solved in an own web-application based on Google WebToolkit otherwise it's possible to develop an own Java application for workflow creation. Both client solutions couldn't be easily integrated in a web-based information system developed without Java.

Another solution was the orchestration with Apache ODE [18], which follows the BPEL standard. BPEL is a XML based language where any kind of process can be executed e.g. within sequences, loops or if-conditions. The workflow is defined in XML files in which the calls to the processes can be created and the responses parsed individually. In context of providing workflows for Web Processing Services there is no special linking beside a HTTP POST interface to call these services. Every Web Processing Service has to be defined as a HTTP binding and the workflow process itself as SOAP binding. This makes the creation of a workflow very complex. But for providing these BPEL workflows additionally as WPS processes, further software around Apache ODE providing this WPS is needed. Therefore for normal workflows overloaded. To develop an implementation to create own workflows over a web-application is another big task.

The third possibility to chain processes is the usage of an internal software at the German Aerospace Center (DLR), called DIMS-WPS [19]. This internal closed-source software was developed for the DLR based on technologies integrated in prior years for the existing remote sensing data archiving system within the Data Information and Management System (DIMS). It provides a Web Processing Service for WPS workflows and defined its own XML based language for describing these workflows. Figure 3 shows this XML with the process name, two WPS calls (steps 1, 2) and the definition of the outputs (3). Every process is stored as XML file in a specific directory of this orchestration engine. The software provides the request “execute” as defined by the OGC WPS specification. The requests “getCapabilities” and “describeProcess” have to be defined by a proxy service to provide a standard-conform OGC WPS.

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:wps-workflows xmlns:wpsw="http://wps.dims.dfd.dlr.de/wps-workflows"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wps.dims.dfd.dlr.de/wps-workflows wps-workflows.xsd">
  <wps:workflow identifier="ModisWorkflow">
    <wps:sequence>
      (1) <wps:step identifier="modis_lib" successor="http://pyros.intra.dlr.de/pywps/wps.cgi" successorIdentifier="modis_lib">
        <wps:input identifier="l0_url" sourceIdentifier="l0_url" globalInput="true"/>
      </wps:step>
      (2) <wps:step identifier="modis_mod14" successor="http://pyros.intra.dlr.de/pywps/wps.cgi" successorIdentifier="modis_mod14">
        <wps:input identifier="data_l1b1km" sourceStep="modis_lib" sourceIdentifier="data_l1b1km"/>
        <wps:input identifier="data_l1bgeo" sourceStep="modis_lib" sourceIdentifier="data_l1bgeo"/>
      </wps:step>
    </wps:sequence>
    (3) <wps:outputs>
      <wps:output identifier="data_l1b1km" sourceIdentifier="data_l1b1km" sourceStep="modis_lib" asReference="true" />
      <wps:output identifier="data_l1bgeo" sourceIdentifier="data_l1bgeo" sourceStep="modis_lib" asReference="true" />
      <wps:output identifier="data_mod14gml" sourceIdentifier="data_mod14gml" sourceStep="modis_mod14" asReference="true" />
      <wps:output identifier="numberOfFires" sourceIdentifier="numberOfFires" sourceStep="modis_mod14" />
    </wps:outputs>
  </wps:workflow>
</wps:wps-workflows>

```

Figure 3. XML based definition of workflow for DIMS-WPS.

For providing an orchestration service DIMS-WPS was taken because 52 North Orchestration service couldn't work well with raster data and external clients couldn't define workflows. Apache ODE with BPEL was too complex to provide a real Web Processing Service as an orchestration service. A proxy for DIMS-WPS was implemented in Drupal CMS to provide the functions “getCapabilities” and “describeProcess”. Workflows were created by a web-application within Drupal CMS generating the workflow XML file. Additional WPS processes for administrations of this service were developed to insert, update and delete workflows in DIMS-WPS.

## USE CASE

To use the developed web-based processing system data from MODIS sensor were selected because the data and defined product algorithms are highly available. The German Aerospace Center owns a satellite data archive with data from MODIS sensor over Europe. Other data from all over the world is freely available from NASA servers. Software components to derive final products are available too.

## Data and products

The data are available at unprocessed “level 0” files from the satellites Aqua and Terra. With several software components from NASA it is possible to create pre-processed “level 1b” files. From this level several final products can be created with further software components. These software can be executed

from commandline with specific parameters. Final products from subjects like atmosphere, cryosphere, land and ocean can be derived, for example a cloudmask, landsurface- and water temperature, snow- and fireproducts as well as landcover information. There are 34 standard products in five categories: calibration, atmosphere, land, cryosphere and ocean [20]. For testing the derivation of fire hotspot points, landsurface-temperature and corrected surface-reflectance were implemented.

The output data of NASA’s software components are data in form of the Hierarchical Data Format (HDF). This is used from NASA as output data format of their Earth Observation System. Another software component can transfer specific HDF files to GeoTIFF files for further processing and visualisation.

## Processing chains

For any product it is necessary to create a process chain based on several processes. Most of the processes are used in several process chains; only a few product-specific processes have to be changed. Figure 4 shows exemplary a workflow to derive fire hotspot points. The first task is to download data (“modis\_download”), the second to process level 0 to level 1b (“modis\_l1b”), the third task is to apply the product-generation (“modis\_mod14”) and the normal visualisation of the satellite scene as well as the export to GeoTIFF (“modis\_l1b\_crefl” and “modis\_l1b\_crefl\_geotiff”). Further processings are the intersection of firepoints with landcover- and countryborder informationen (“info\_points\_grass”) with GRASS GIS and the export from GML to Shapefile (“convert\_gml\_to\_shp”). This workflow has defined input and output data, which where specified on execution of the workflow. Every process chain can work with internal data from the data archive system or with external data reference by a HTTP URL. The final outputs of this processing chain are a GeoTIFF file visualising the satellite scene, a GML and a shape file representing the fire points and a GeoTIFF file visualising the classification mask generated for detecting fire points. Similar workflows were generated for the derivation of landsurface-temperature, for the visualisation of a MODIS satellite scene and for the on near-realtime data running fire-service (derivation of firepoints with deployment on PostgreSQL and data-exchange to a webserver).

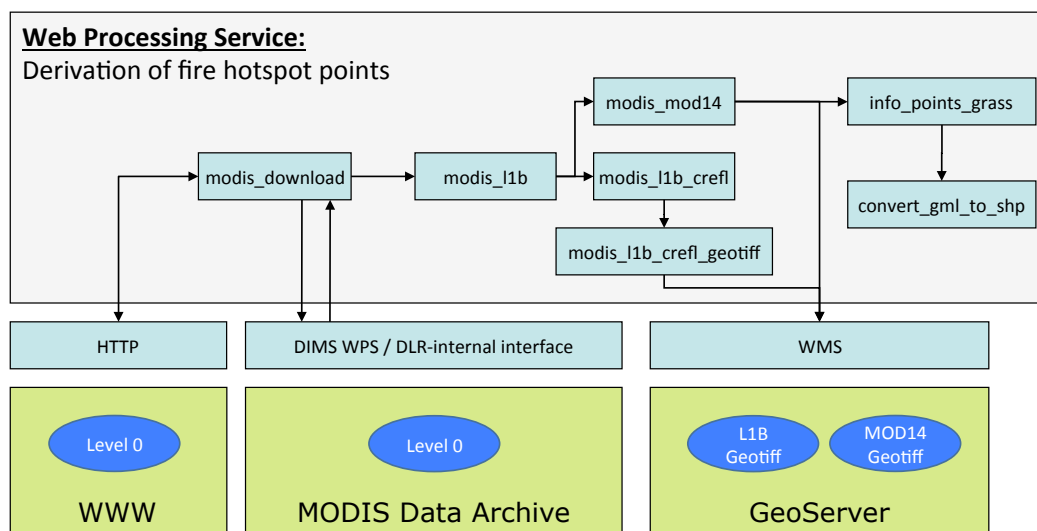


Figure 4. Sequence of processes called for derivation of fire points with interfaces for input and output data

## Web Processing Services

For the given use case 17 processes were developed. These processes were parted into seven groups (figure 5): initialisation, data download, pre-processing, product-derivation, post-processing, dissemination and administration. The processes within groups one to five are accessible from everywhere, the processes within groups six and seven run on a separate WPS instance protected for the usage from defined clients. The processes for “dissemination” are only used in the automatically running workflow on near-realtime data. The processes for “administration” will be called from the web-based information system to administrate the orchestration engine and the WPS.

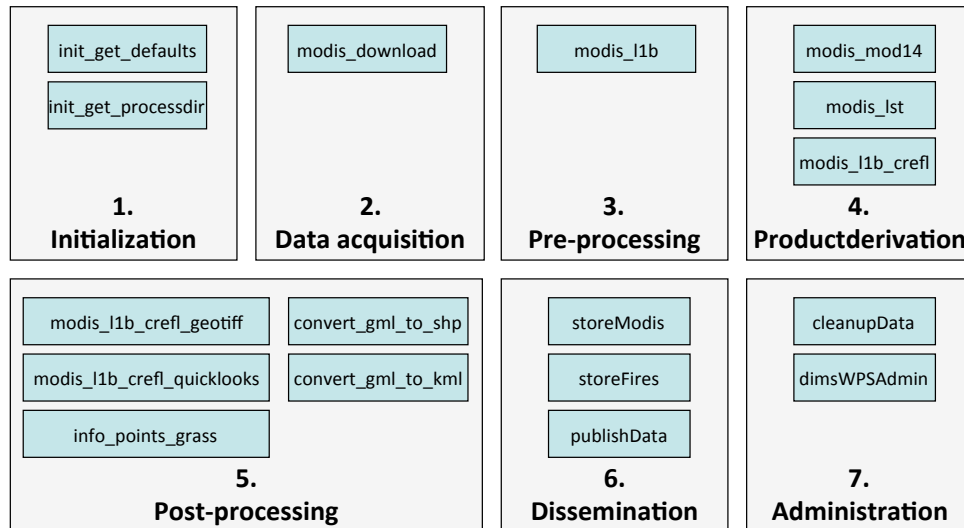


Figure 5. Developed processes in different categories.

In any process there is an optional feature declaring a processing directory as an additional input. In that directory all processed files are stored. It replaces the temporary processing directory of pyWPS. Aim of this processing directory is the unique possibility to access the processed data from another process to a later time and to directly reference this output data without copying to the webservers directory for wps-outputs. While copying large data from temporary processing directory to webservers output directory needs unnecessary time. Declaring one processing directory for the whole workflow has the further possibility to create a logfile with information of each called process in the workflow. After finishing a process chain the operations of the workflow can be followed by viewing this logfile.

## WEB-BASED INFORMATION SYSTEM

The web-based information system acts as a client for the developed processing system. The defined requirements indicate a map to visualise satellite data and products, a search component for archived satellite data, a monitoring tool of started processes as well as the creation of workflows for product creation. For this information system open-source software was used. The frontend was developed with ExtJS, GeoExt and Openlayers. The backend was built on the framework Drupal.

### Drupal CMS

Drupal is a web content management system and a web application framework with a fine-grained, modular design which can be easily extended by lots of modules. This framework provides the possibility to organize structured content in several ways with a sophisticated user authorisation mechanism. To act as a backend tool, content-types for handling WPS servers, WPS execute requests and processing chains were integrated. Every start of process from the developed information system will be registered and stored as content in the specific content-type. Any content is assigned with the user who creates it. If a user logs in, he will see his own started processes and can view their output data.

### Structure

The web-based information system is structured in three parts (figure 6): header with main navigation, left part with metadata catalogue search and processing system and the middle (content) part where the interactive map or other content is displayed. The map is integrated with OpenLayers and GeoExt functions. The components in the left part are based on ExtJS and GeoExt. To display content from Drupal a module called “ext\_json” was used to request content items and special views on this content. With simple Ajax calls the ExtJS components could be filled with their content.



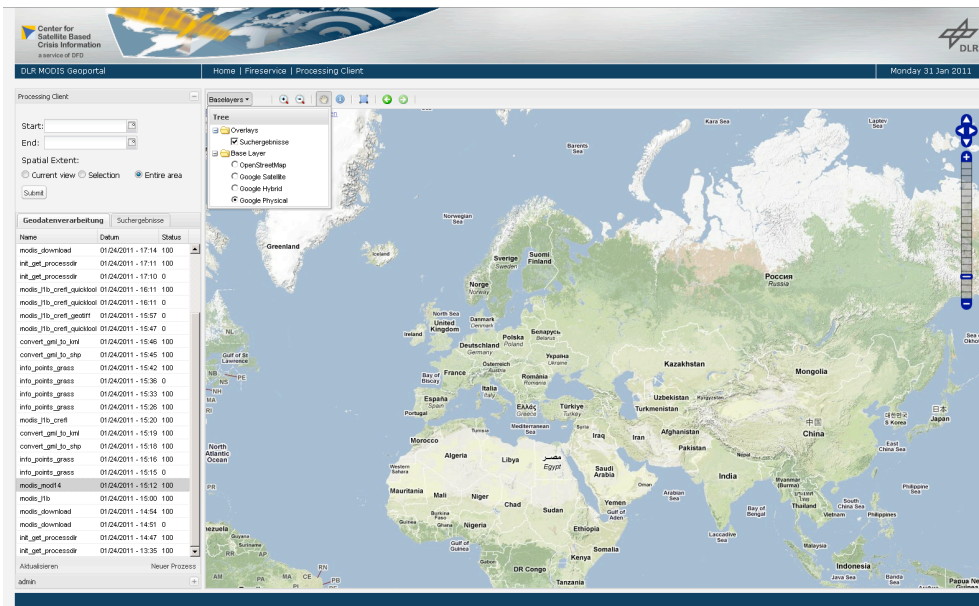


Figure 6. Screenshot of the developed web-based geographic information system based on open-source software.

## Satellite data search

The internal archive with remote sensing data is provided by a standard-compliant OGC CSW interface with SOAP binding. A “GetRecords” request integrated in a SOAP envelope leads to a “GetRecordsResponse” response and lists available remote sensing data with specific information fields. A developed Drupal module takes the search parameters and returns a GeoJSON object for further processing within the frontend. This GeoJSON object contains the bounding box of the data and metadata information. With the GeoExt FeatureStore component it is possible to visualize the bounding box in the GeoExt map automatically after loading the GeoJSON object (figure 7). Based on the dataset id given within the metadata it is possible for a WPS process to request the data itself for further processing. Figure 7 shows the search form and the response items in the left part and the bounding boxes in the map. An integrated paging makes it clearly represented.

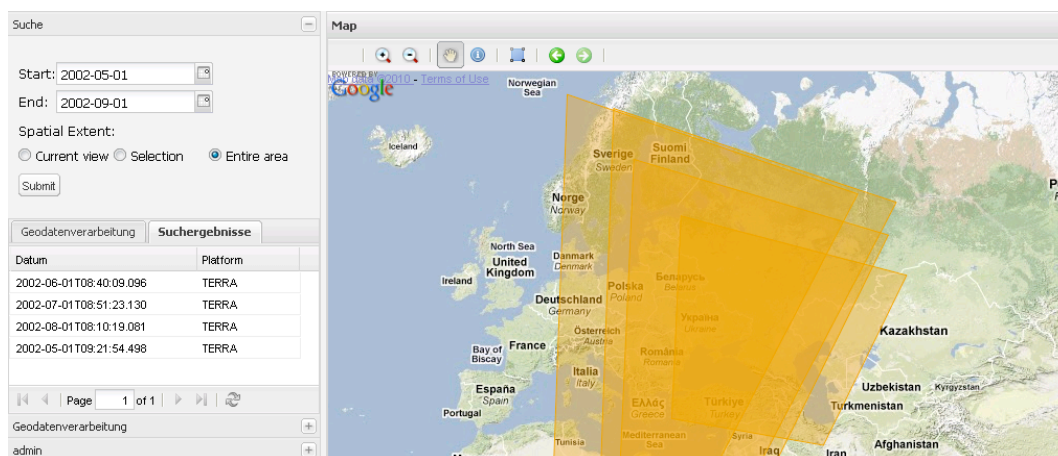


Figure 7. List of searched satellite scenes and visualisation of their bounding boxes.

## Processing services

Based on the Web Processing Services registered in Drupal CMS a client for WPS with ExtJS components was developed to start processes and request their status. After loading the system a list of processes is generated from the registered Web Processing Services. Choosing a process the “DescribeProcess” request will be executed. From the given information the process is graphically displayed and input-fields are generated. This development is based on a WPS extension for OpenLayers

published by pyWPS. To register the started process as content in Drupal CMS a proxy for executing WPS processes was implemented. The proxy parses the execute request, inserts a new content object and forward the request without changes to the Web Processing Service. To check status of each running process regularly, a cronjob calls a developed service from Drupal CMS. This service pass through all registered and running processes and calls the status URL. The new status information will be updated in the content object of each process. To protect both writing mechanism (insert and update) the cronjob logs in to Drupal CMS via HTTP authorisation, which was configured only for these services. The running and completed processes are listed in an ExtJS Gridpanel. For each process it is possible to request the actual status information with their input and output data (figure 8). A direct connection to open-source software GeoServer allows an instant visualisation of the vector- and raster-based output.

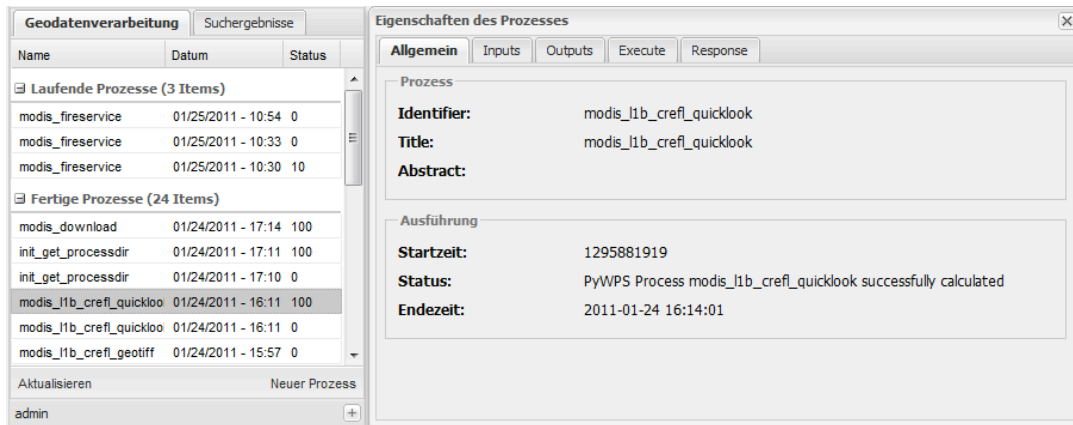


Figure 8. Representation of running and completed processes, serving detailed information for a selected process.

## Processing chains

Based on DIMS-WPS the process chains are described in XML documents. To create these chains in the web-based information system an interface between information system and DIMS-WPS was developed. To chain elements the JavaScript library jsPlump was integrated. In an editor window (figure 9) it is possible to add several processes and bind output data to input data. Furthermore some input data can be selected for global inputs so that the user can declare this input by starting this new opaque process. After saving a new workflow it acts as a normal process. Every new workflow is stored in Drupal CMS and transferred to DIMS-WPS over a administration process on the same server. With the individual rights management within Drupal CMS it is possible to assign every process chain only to registered users with respective rights.

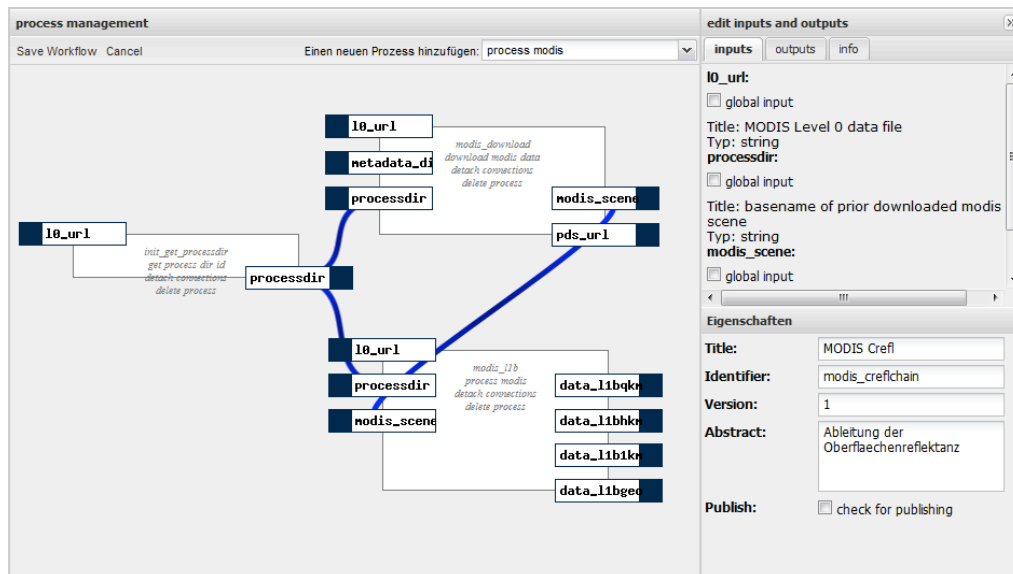


Figure 9. Window to create own workflows.

## CONCLUSION AND DISCUSSION

In the context of this work requirements and software solutions for web-based processing of remote sensing data were tested and evaluated. A web-based information system was developed to tether and manage this processing system. With the whole system, it is possible to derive implemented products based on MODIS satellite data, create own processing workflows, monitor the started processes and visualise the output data from the completed processes. Based on the status of research the potential of web-based processing with workflows is very distinguishable. But also areas of problems in the processing of large data are present und research have to be enlarged in this areas. Furthermore the chaining of processes is not standardised and descriptions of processes are not yet clearly structured and defined.

For the usage of large data as usual in the earth observation, [7] emphasize to bring the algorithms to the data. A good basic idea but in contrast to centralized web-based processing services, as the idea of spatial data infrastructures, in which data is distributed on several systems. For the developed system the data, which were processed, were already available on the same network. With an interface the processing system could be connected to the data archive and the web-based information system from the catalogue system to the data archive. To process external data it is necessary that the server downloads the input data first. The idea from [21] to implement a mechanism to list all available data from the processing server leads to an efficient datamanagement. This is important for processing workflows with separate processes because data must be exchanged a lot between processes. Other data, often used as input data for processes, like a digital elevation model, can be placed at the server and reused as input data. Working with large data the management of data used for processing has to be clearly defined. This was a main aspect of this work. Therefore centralized processing directories as well as the corresponding program logic were established in the processing system.

For the development of a web-based processing system software solutions providing OGC Web Processing Services and Orchestration Engines were tested and evaluated. Already in the year 2006 [4] remarked that there is no standardised way to chain processes. For complex workflows BPEL can be used, but for easier workflows other standardised solutions have to be developed. The possibility to define process chains with a XML based scheme is obvious. This could be created by any kind of software, Web Processing Services has just the task to parse this file and execute the specified processes. An exemplary XML file is used in the internal software DIMS-WPS to provide the orchestration engine. In a developed web-application this XML file was generated for the given process chains. Only the status messages from DIMS-WPS can be criticized. Those and other additional information to processes and workflows should be available in a standard format. So far no detailed rules were defined.

Another part of this work was the implementation of a web-based geographic information system. Using Drupal CMS as a backend and frontend framework showed the possibilities of a user-dependent information system. User can store and list their running and completed processes and create their own workflows. Web Processing Services and workflows can be provided user-dependent or additional security authorisations can be stored in the Content Management System. Another possibility is the use of Drupal CMS as WPS proxy so that additional security limits can be controlled using the rights management of the CMS. Using a Content Management System to control the content of a geographic information system have to be well defined, but the possibility to store user-dependent content like context information of maps or executed processes and their final outputs shows the advantages.

But all the possibilities of web-based processing and a web-based geographic information system are useless when no data or algorithms for processing are available. Beside some starting parameters algorithms also have to run automatically. Having near-realtime data additional advantages are available by processing automatical products and providing official services like fire points or flood areas. For data from MODIS sensors a cupple of algorithms and near-realtime data from the German Aerospace Center are available. Only the resolution of this data is a bit outdated. Better resolutions will come with the Sentinel Family from the European Space Agency, whose data is free of charge.

## REFERENCES

- [1] – OGC: OpenGIS Web Processing Service. OGC 05-007r7. Open Geospatial Consortium. (2007)
- [2] – ISO: Geographic information - Services (ISO 19119). Deutsches Institut für Normung e.V. (2006)
- [3] – OASIS: Web Services Business Process Execution Language Version 2.0. OASIS WSBPEL TC. (2007)
- [4] – T. Foerster, J. Stoter: Establishing an OGC Web Processing Service for generalization processes. In: Workshop of ICA Commission on Map Generalisation and Multiple Representation. (2006)
- [5] – L. Diaz, S. Costa, C. Granell, M. Gould: Migrating geoprocessing routines to web services for water resource management applications. In: Proceedings of AGILE 2007. (2007)
- [6] – A. Friis-Christensen, N. Ostlaender, M. Lutz, L. Bernard: Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions. In: Transactions in GIS 11.6, pp. 799-818. (2007)
- [7] – C. Granell, L. Diaz, M. Gould: Managing Earth observation data with distributed geoprocessing services. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium. Ieee, pp. 4777-4780. (2007)
- [8] – J. Brauner, T. Foerster, B. Schaeffer, B. Baranski: Towards a Research Agenda for Geoprocessing Services. In: Proceedings of 12th AGILE International Conference on Geographic Information Science. Hannover. (2009)
- [9] – E. Nash: WPS Application Profiles for Generic and Specialised Processes. In: Proceedings of GI Days. Published by F. Probst and C. Kessler. Muenster. (2008)
- [10] – A. Weiser, P. Neiss, A. Zipf: Orchestrierung von OGC Web Diensten im Katastrophenmanagement am Beispiel eines Emergency Route Service auf Basis der OpenLS Spezifikation. (2007)
- [11] – B. Schaeffer: Towards a Transactional Web Processing Service. In: Proceedings of GI Days 2008. (2008)
- [12] – B. Stollberg, A. Zipf: OGC Web Processing Service Interface for Web Service Orchestration Aggregating Geo-processing Services in a Bomb Threat Scenario. In: Web and Wireless Geographical Information Systems. Issue 4857. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 239-251. (2007)
- [13] – J. Brauner, B. Schaeffer: Integration of GRASS Functionality in Web based SDI Service Chains. In: Proceedings of the FOSS4G 2008. Cape Town, South Africa, pp. 420-429. (2008)
- [14] – G. Sancho-Jimenez, R. Bejar, M. A. Latre, P. R. Muro-Medrano: A Method to Derivate SOAP Interfaces and WSDL Metadata from the OGC Web Processing Service Mandatory Interfaces. In: ER Workshops 2008, LNCS 5232. Springer Verlag, pp. 375-384. (2008)
- [15] – B. Schaeffer: Standardized Geoprocessing with 52 North Open Source Software - Web Processing Service Tutorial. Techn. Report. 52 North Initiative. (2009)
- [16] – J. Cepicky: Welcome to pyWPS. <http://pywps.wald.intevation.org/> (2011)
- [17] – 52 North Initiative: WPS Orchestration & Geoprocessing Workflows Introduction. (2010)
- [18] – The Apache Software Foundation. Apache ODE. <http://ode.apache.org/index.html> (2010)
- [19] – Werum AG: DIMS WPS Prototype. Software user manual (unpublished). (2010)
- [20] – V. Salomonson, W. Barnes, E. J. Masuok: "Introduction to MODIS and an Overview of Associated Activities". In: Earth System Science Satellite Remote Sensing. Volume 1: Science and Instruments. Springer Verlag. pp. 12–32. (2006)
- [21] – C. Michael, D. P. Ames: Evaluation of the OGC Web Processing Service for Use in a Client-Side GIS. In: OSGeo Journal 1<sup>st</sup> May 2007. (2007)

Acknowledgement: This paper is based on a master thesis written at German Remote Sensing Data Center within the German Aerospace Center (DLR) Oberpfaffenhofen in cooperation with Department of Earth Observation within Friedrich-Schiller-University (FSU) Jena. I thank my supervisors, Prof. Dr. Christiane Schmullius (FSU) and Dr. Christian Strobl (DLR). Further I thank my colleagues at DLR and FSU supporting me to complete my work.

Author's biography: Jonas Eberle, Master of Science in Geoinformatics with focus of remote sensing and spatial data infrastructures and Bachelor in Applied Computer Sciences with focus of environment informatics. He works since several years as web-developer.