GNC on Demand

Final Report

Developers:

Andres Westermann <andres.westermann@gmail.com>

Christoph Mülligann <cmuelligann@uni-muenster.de>

Damian Lasnia <d.lasnia@gmail.com>

Jafar Hafiz <javar.h@gmail.com>

Goal

The overall goal is the design, implementation and setup of a prototypical core spatial data infrastructure for GEONETCast data. The web service should make its products available for department 14 of University of Muenster (around 1600 students and 150 employees) through an intuitive web interface. Our approach improves the accessibility of incoming geographic data by introducing scheduling, spatial querying and format conversion capabilities to GEONETCast products.

Objectives

Easy to use "Task registration View"

Due to the limited IT knowledge in the target audience, the web "Task Registration View" has to be structured as a straightforward usability concept. Providing only one simple web site where inputs can be filled in from top to bottom and submitted via button after the last form element is such a concept. The combination of form element inputs should be implemented in a manner that the amount of false input exceptions is minimized. However, the exceptions triggered by illegal user inputs should make the user clearly aware of his/her mistakes.

Limit the access to people from University of Münster

Due to licensing issues the user group must be restricted to the people from University of Münster.

Provide available data in a "Folder View"

The outcome of a task is one or several georeferenced image files and their associated XML file(s). The software must provide an interface where users are able download their data.

Methodical and technical approach

On demand

Like geodata in common, GEONETCast data refreshes continuously (>5min) so that a great amount of data is received. In general we can distinguish between two primitive alternatives how to handle this incoming data stream in order to provide certain products at a certain

timestamp: The first approach is to store the whole stream as long as storage capacity is available and overwriting existing data afterwards, starting with the oldest ones. The other approach is that the system administrator specifies which products will be stored at certain dates. This way specific data is available within a longer timeframe but the stream becomes incomplete. We designed our service by combining both approaches und mixed them with empowered users: Users are able to specify their needed data. This geodata will be stored directly in their private user folders when it's available. Furthermore historical stream geodata is available in a buffer.

Java + GWT

Google Web Toolkit (GWT) is an open-source development kit optimized for building complex browser-based applications. GWT is applicable for the gnc on demand web service because it integrates client and server components within one application, is entirely implemented in Java and thus very flexible in handling interactions between client and server. Before building the WAR (Web Archive) file and deploying it to the server, client based Java code is compiled to JavaScript and HTML source code. Server sided source code that interacts with client sided source code extends the "Java Survlet Superclass".

Extreme Programming

Due to the available time for finishing the project we decided to apply Extreme Programming. While in traditional programming analysis, design, implementation and testing is proceeded one after another, Extreme Programming runs these phases simultaneously in several periods. Regarding this method we did not create a detailed workplan or schedule but met for quick and intensive meetings to be aware of each other's progress and for planning further steps. Figure 1 illustrates the outcome of such a meeting where the initial interface specification between the different working groups took place.

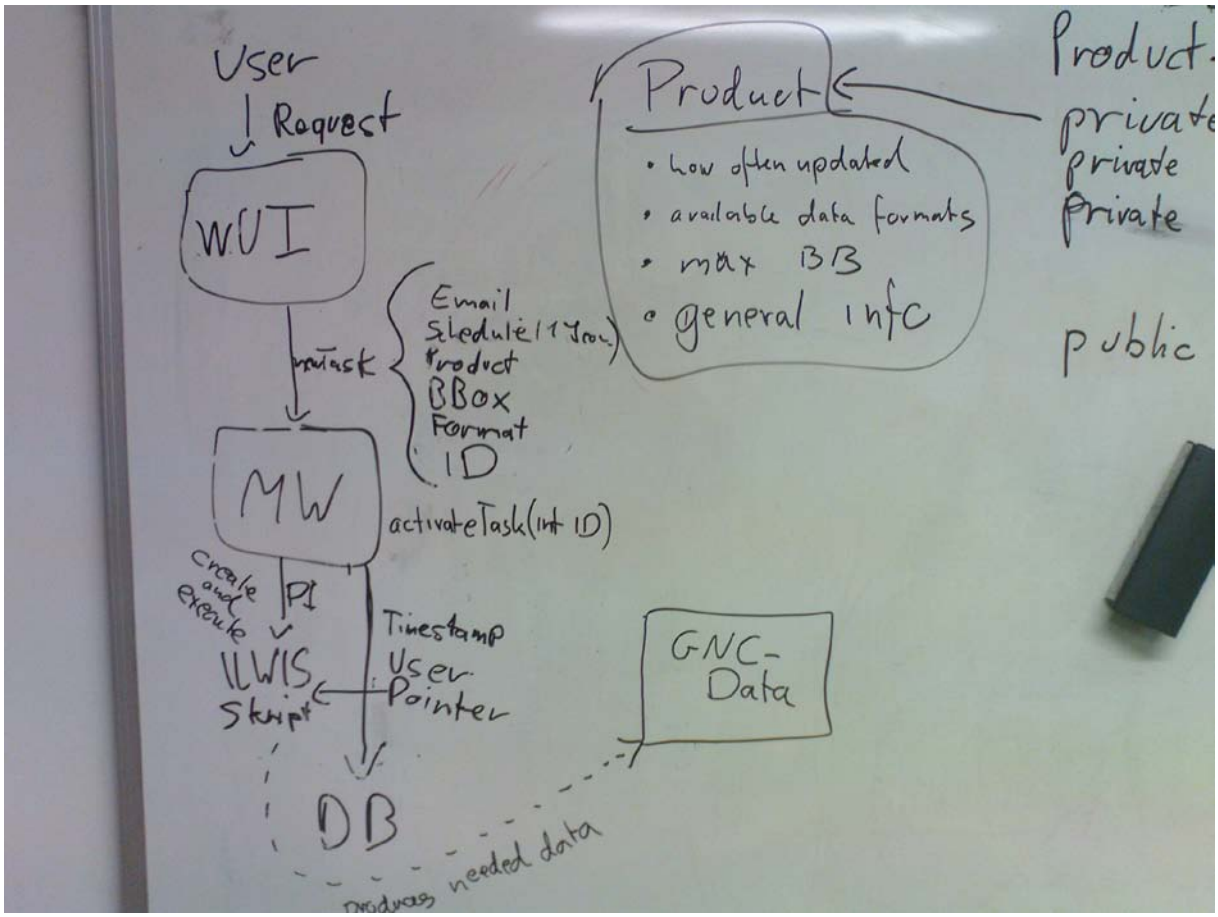

Figure 1: Whiteboard after a quick meeting

Result

Software

Our implementation hides and automates the whole process chain, beginning with GEONETCast raw data, continuing with ILWIS scripts and ending with the GeoTIFF plus metadata XML file. GNC on Demand consists of an uncomplex and intelligent web user interface, powerful task scheduling component, generic script writer & executer for ILWIS scripts and personal folder view. The following paragraph depicts the main software architecture of the web service (see Figure 2). The two main components of GEONETCast on Demand are the Folder and Task Registration View. Before being able to create a task, the input values have to be specified in the Task Registration View.

First, When submitting a new task (see “Create Task” in Figure 2), a new entry is created in the “Task-Database”. The new database entry has an attribute giving information about the status of activation. Additionally an e-mail, including an activation link with an encrypted

identifier, is sent to the user. By executing the activation link the task is activated and edits the activation attribute for the database entry. If users have registered a task for the first time an additional e-mail containing the Folder View link is sent. After the task is activated the “Database Observer” comes into play. This component has two main parts which get automatically checked every five minutes. First, it checks for tasks which got recently activated and generates runnable ILWIS scripts for this request and stores metadata XML file(s) to the user’s folder for each image. Second, it checks for tasks which rely on already available data, meaning that the time (Linux time in milliseconds) of the requested source files is lower than the current time (Linux time in milliseconds). When scripts for available data are executed, ILWIS including GEONETCast Toolbox comes into play. This software component builds a submap fitting the requested bounding box and converts raw data to the output file format specified by the user (e.g. GeoTIFF). Afterwards ILWIS stores the converted data onto the GNC Network Drive.

The second part of GNC on Demand covers the visualization of the user’s personal folder. The Folder View gives users the possibility to see their requested data. A description of the image data is also available in a metadata XML file which has the same name as the image file. Additionally a Log file containing general information about success or failure during the execution of ILWIS scripts.

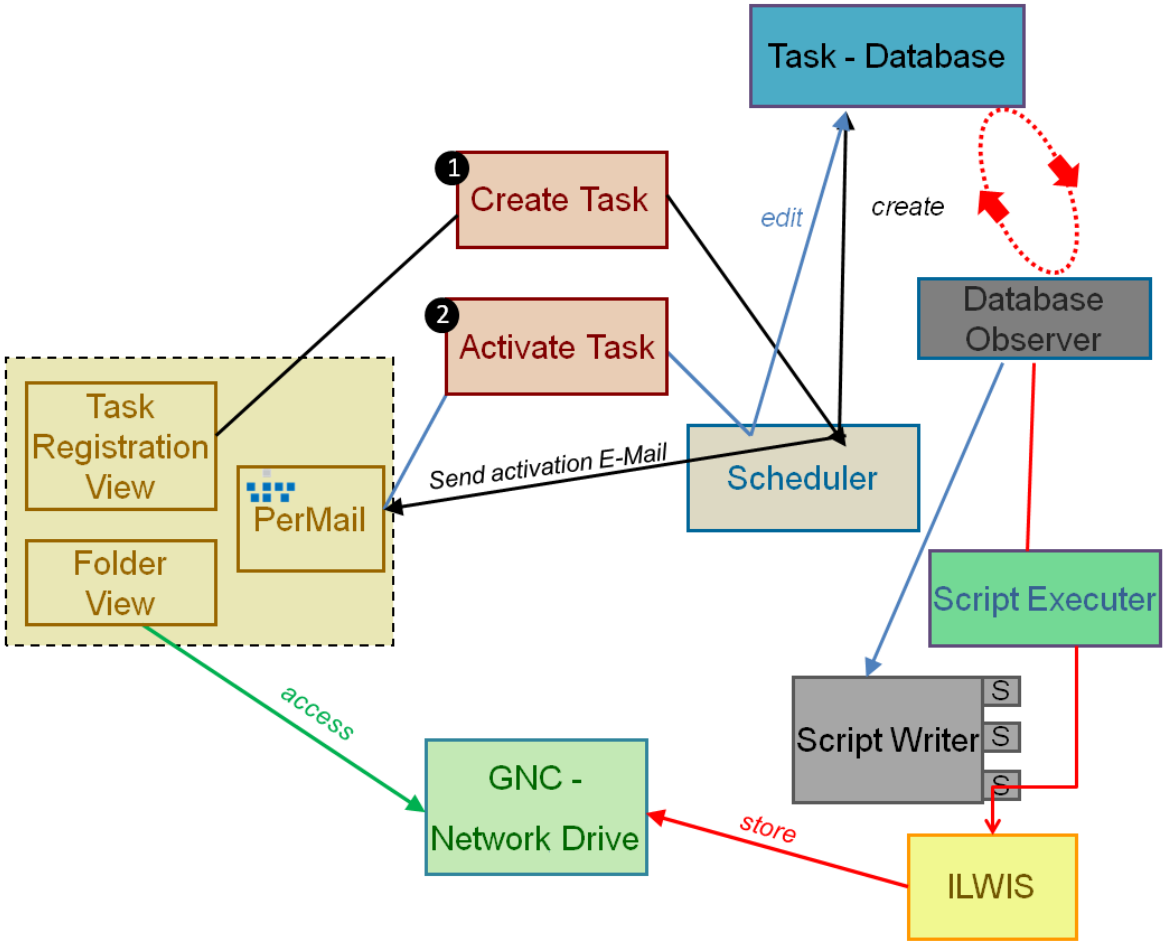

Figure 2: Showing the main architecture of the software components.

Setup http://gnc-data.uni-muenster.de:52472/god/

The setup of GNC on Demand consists of three servers which form the main hardware architecture for the GNC on Demand web service.

The first component of the main hardware architecture is the “Groundstation”. It is a Windows computer onto which it is possible to connect via VNC client. The raw data, collected via several satellite channels, is saved onto the hard disk of the “Groundstation”. Additionally the raw data is moved into a sorted folder structure by using ILWIS and GEONETCast Toolbox. The second server, called “GNC Data Server”, runs on a Windows Server 2008 R2 Enterprise. It synchronizes the previously described structured folders of the “Groundstation”. A Glassfish Server including the newest Java Runtime Environment is also installed on it because the Web Archive (WAR) file generated by the Google Web Toolkit is deployed onto it. Third, a server running on Linux got installed. Apache Tomcat is installed

on the server which serves to mount the network drive of the “GNC Data Server”. This network drive contains all user folders and makes them traceable for the Folder View. The network drive is mounted on the Linux server because Apache Tomcat runs safer on Linux than on Windows servers.

Available Products

Name	**Information**
Albedo	Fraction of incident light or radiation reflected by a surface or body
DSLF	Down-welling Surface Long-wave Radiation Flux
DSSF	Down-welling Surface Short-wave Radiation Flux
FVC	Fraction of Vegetation Cover
LAI	Leaf Area Index
LST	Land Surface Temperature

Further Developments

Launch developer platform

We handled to deploy a prototype of the GEONETCast on Demand web service. The next step after the project will be to establish a sufficient project home, including SVN/CVS (that supports more than 3 users, not like the one we used for our project), a bug tracking system, a road map and user feedback forms.

Storage Manager

Due to the server’s large physical device capacity our storage management regarding to efficiency is only implemented rudimentary.

More Products, more formats

Adding new products to our service is not trivial and takes much effort. At least we managed to make some product available, but more have to follow.

Describe Products

The products are not sufficiently described in the associated metadata XML file. One of the most important missing descriptions is a legend attached to the imagery.

Standards

The ILWIS scripting and ILWIS script execution for data conversion and additional commands could be implemented as to the OGC Web Processing Standard.

Lessons learned

- Extreme Programming
- ILWIS Scripting
- Deploying Web Archives
- GWT Coding
- Integrating several different software and hardware components