SEVENTH FRAMEWORK PROGRAMME

Crop Monitoring as an
E-agricultural tool in
Developing Countries

E-AGRI
E-Agriculture

E-AGRI VIEWERS FOR ANALYSIS AND VISUALISATION OF WEATHER AND CROP INDICATORS

Reference: *E-AGRI_D26.1_E-AgriViewers_D1.0*

Author(s): Allard de Wit, Jappe Franke, Hugo de Groot, Bas vanMeulebrouk

Version: 1.0

Date: 26/02/2013

DOCUMENT CONTROL

Signatures

Author(s) : Allard de Wit, Jappe Franke, Hugo de Groot, Bas vanMeulebrouk

Reviewer(s) : Qinghan Dong

Approver(s) :

Issuing authority :

Change record

Release	Date	Pages	Description	Editor(s)/Reviewer(s)
1.0	26/2/2012	21	First release	Allard de Wit
2.0				

TABLE OF CONTENT

LIST OF FIGURES

EXECUTIVE SUMMARY

The overall objective of the e-Agri project is to “Transfer, adapt and validate the European MARS Crop Yield Forecasting System and its multi-component extensions to three developing regions in Asia and in Africa. Despite the advanced technologies that are available in the MARS Crop Yield Forecasting System, these technologies have only value if the results can be visualized and transferred to analysts and decision makers in an effective way. Results from the MARS Crop Yield Forecasting System are visualized operationally through the so-called MARS viewers. These viewers have been developed over the last couple of years within the MARSOP3 project and provide a web-based solution to effectively convey information to analysts at JRC that are monitoring the agricultural season.

Within e-Agri the need for developing viewers that can help monitoring the agricultural campaign has been clearly expressed by the users of the results, and in particular by the Moroccan partners. However, the setup and database schema behind the current MARS viewers is very much tied to the MARS Crop Yield Forecast System. Moreover, some of the functionality of the MARS viewers was found to be too complicated. Within e-Agri, a development has been started by Alterra to build a new viewer taking into account the experiences learnt from the MARS viewers. This documents describes the efforts that have been invested so far in developing this viewer and the status of it.

1. Introduction

The overall objective of the e-Agri project is to "Transfer, adapt and validate the European MARS Crop Yield Forecasting System and its multi-component extensions to three developing regions in Asia and in Africa, namely the eastern China (HUAIBEI and JIANGHUAI Plains), Morocco and Kenya." Ultimately, these tools should lead to more effective monitoring of the agricultural season in order to make yield predictions that can help to manage and mitigating risks associated with food price volatility.

Despite the advanced technologies that are available in the MARS Crop Yield Forecasting System, these technologies have only value if the results can be visualized and transferred to analysts and decision makers in an effective way. Usually this means that 'raw' results of weather stations, satellite imagery or crop model simulations have to be visualized in tables, charts and maps. Moreover, such maps can often only be interpreted if they can be compared to a reference such as the previous year or the long term average conditions. This reference is needed to understand if the conditions during the current agricultural campaign are progressing as usual, or if there are large differences (often due to weather conditions) that call for additional analysis or even field visits.

Results from the MARS Crop Yield Forecasting System are visualized operationally through the so-called MARS viewers. These viewers have been developed over the last couple of years within the MARSOP3 project in order to effectively convey information to analysts at JRC that are monitoring the agricultural season. A characteristic of the MARS viewers is that they can build maps and charts on-the-fly based on settings specified through the user interface. This is a large improvement over the system that was used previously where only a predefined set of indicators and maps was available for the analysts. Moreover, the MARS viewers present a web-based solution allowing users in different institutes at different physical locations to make an analysis using the same database.

Within e-Agri the need for developing viewers that can help monitoring the agricultural campaign has been clearly expressed by the users of the results, and in particular by the Moroccan partners. It has already been outlined in the CGMS Usability reports that a web-based solution is also preferred in E-Agri in order to serve multiple users in different institutes at different locations.

However, the setup and database schema behind the current MARS viewers is very much tied to the MARS Crop Yield Forecast System with many dependencies. It also carries some legacy with it (such as table names and data structures), which is generally undesirable for new installations. Moreover, some of the functionality of the MARS viewers was found to be too complicated and superfluous during the training for Moroccan users that was carried out as part of WP6.

Within e-Agri, a development has been started by Alterra to build a new viewer taking into account the experiences learnt from the MARS viewers. This viewer should have the following characteristics:

- Be a web-based solution to support users on different physical locations through the internet.
- Have the ability to visualize weather and crop indicators through user-defined selections. The latter can be provided by model simulations, satellite retrievals or other data.
- Have the ability to visualize results at grid and regional level.
- Have the ability to generate charts for selected grids and regions. Thereby visualising the time-series of different variables as well as the comparison of the current year to the long-term average and the previous years.
- Have a simplified interface compared to the MARS viewers, while still allowing actions like storing user settings and favourites.
- Have a simplified and logical database structure that does not carry MARS legacy, simultaneously supporting the storage and display of different variables.
- Allow different database systems to be used, standardizing on an open source database system instead of ORACLE.

This documents describes the efforts that have been invested so far in developing this viewer.

2. Database architecture

2.1. Background and terminology

During the development of the E-Agri viewers, it was chosen to standardize on the PostgreSQL relational database system for several reasons. First of all, this is an open source database that poses no additional cost for licensing for end users. Second, Alterra has good experiences with this database in terms of management and performance. Finally, the database has some good tools available to manage, load and query the database (e.g. PgAdmin III, http://www.pgadmin.org/).

For designing the structure of the tables a harmonized terminology has been developed that has been implemented within all database tables (where appropriate):

- Grid: A rectangular shaped collection of rectangular grid cells of the same size and area. Each grid cell is identified by its grid_id (which is a unique meaningless integer identifier) and the coordinates of the cell centre in latitude, longitude and in project coordinates. The projected coordinates are assumed to be regularly spaced and the extent of the grid is defined by the minimum and maximum projected coordinates. Finally, each cell has the same area and the cell size of the grid can be retrieved from the square root of the projected area.

- Region: A region (e.g. province, district, country, etc.) with defined spatial boundaries often designed by the statistical office of a country for publishing regional statistics. Regions must be hierarchical (a region belongs to a larger region) and mutually exclusive (one region only belongs to one higher level region). A region is defined by its region_id, (which is a unique meaningless integer identifier), its region_code (usually assigned by the statistical office, e.g. 'NL12'), its region_name, its region_level (the level in the hierarchy) and its belongs_to attribute which identifies the hierarchical position of that region.

- Landcover: The land cover for which the indicator is representative. This means landcover in a wide sense and refers both to simulated crop types (e.g. wheat) or land cover types from general landcover maps (e.g. CORINE class 'arable land'). Each land cover type is defined by its landcover_id, its landcover_name and its landcover_type_id. The latter refers to the type of land cover to which it refers, e.g. 'simulated crop types' or 'derived from GLC2000'.

- Indicator: a general name for variables that are to be visualized. The name indicator is used for all database tables that store information for both weather variables (weather_indicators) and crop related variables (crop_indicators) which can be retrieved both from crop model simulations or satellite data.

- Theme: An overall term providing a differentiation between main data sources that are available to the users. Common themes are: 'interpolated weather', 'crop simulation results', 'satellite observations'. The division is somewhat arbitrary as themes can be split (e.g. 'satellite observations - VGT' , 'satellite observations - AVHRR'), although fundamentally different themes cannot be merged.

2.2. Table structure

The database schema for storing the indicators makes a principle distinction between 'grid' data and 'region' data. The distinction is the result of the fact that the e-Agri viewers handle these two data structures in a fundamentally different way:

- Gridded data are assumed to be on a rectangular grid which means that they can be retrieved from the database and stored as an image (a PNG file in this case) with a legend and geo-location information attached. The E-AGRI viewer will just load this image and use the geo-location information to geo-reference the image in the map viewer on the right location.

Region data are retrieved from the database as a Styled Layer Descriptor (SLD). The web map server than joins the SLD on the region_code attribute of the shape file. The result of this operation is that each region received the correct colour which is then converted into an image for displaying in the map viewer.

The table structure of the E-Agri database is shown in Figure 1. Overview of the database for storing data for the e-Agri viewers. The schema contains the following tables:

- GRID: Stores the characteristics of the grid/
- REGIONS: Stores the ID, code, name and hierarchy of the regions.
- LANDCOVERS: Stores the land covers used in the system.
- LANDCOVER_TYPES: Stores the land cover types used in the system
- GRID_WEATHER_INDICATORS: Stores the different gridded weather variables that are relevant for crop growth such as temperature, radiation and rainfall. Each set of daily weather variables is stored as a record which is identified by its grid_id and day_obs.
- REGION_WEATHER_INDICATORS: Stores the different regionally aggregated weather variables that are relevant for crop growth such as temperature, radiation

and rainfall. Each set of daily weather variables is stored as a record which is identified by its region_id and day_obs.

- GRID_CROP_INDICATORS: Stores the different gridded crop indicators that provide information about crop growth and status. Crop indicators can consist of satellite observations, output of crop simulation models or other information about crop status. Crop indicators are stored in a normalized way in that each indicator is identified by an indicator_code as well as a grid_id and a day. This has the advantage that new indicators can be easily added without having to add additional columns to the table. The latter is less of a problem with weather indicators because this is a relatively fixed set.
- REGION_CROP_INDICATORS: Stores the different regionally aggregated crop indicators that provide information about crop growth and status. Crop indicators can consist of satellite observations, output of crop simulation models or other information about crop status. Crop indicators are stored in a normalized way in that each indicator is identified by an indicator_code as well as a grid_id and a day.

Overall, the database schema that was designed for the e-Agri viewer has a very small number of tables. Moreover, it is compatible with the database schema of the CGMS although some additional scripts will be needed for transferring data from the CGMS tables towards the tables in the schema of the E-AGRI viewers. Finally, different crop indicators from different themes (e.g. ‘satellite observations’, ‘interpolated weather’, etc.) can be integrated into to the schema easily, which even is quite difficult to accomplish with the MARSOP3 viewers.

In addition to the data schema described in this section, the E-Agri viewers also have a configuration database which stores all the components of the interface. Although this is a complex database, it has only to be filled once and then the contents of the database remain static.

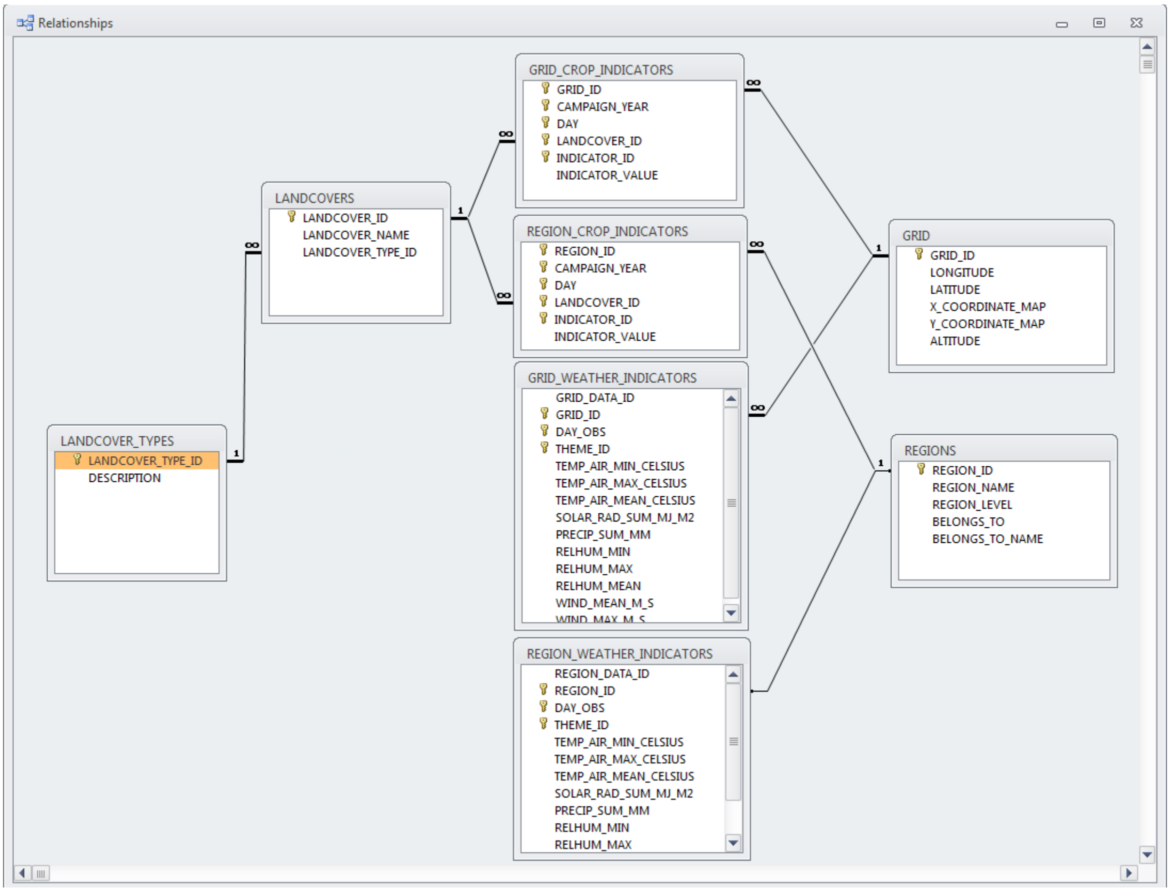


Figure 1. Overview of the database for storing data for the e-Agri viewers.

2.3. Performance issues

So far no performance issues have been encountered, but this can only be tested with a full-size database covering the complete archive of meteorological data and crop indicator data.

3. Viewer architecture

3.1. Overall architecture

The viewer architecture is a typical example of a so called Service Oriented Architecture (SOA). In a SOA functions are separated into distinct units, or services that are made accessible over a network in order to allow users to combine and reuse them in web applications. These services communicate with each other by exchanging data in a well-defined, shared format, or by coordinating an activity between two or more services. In a SOA data, logic and presentation are separated in different tiers. Figure 2 depicts the E-Agri SOA.

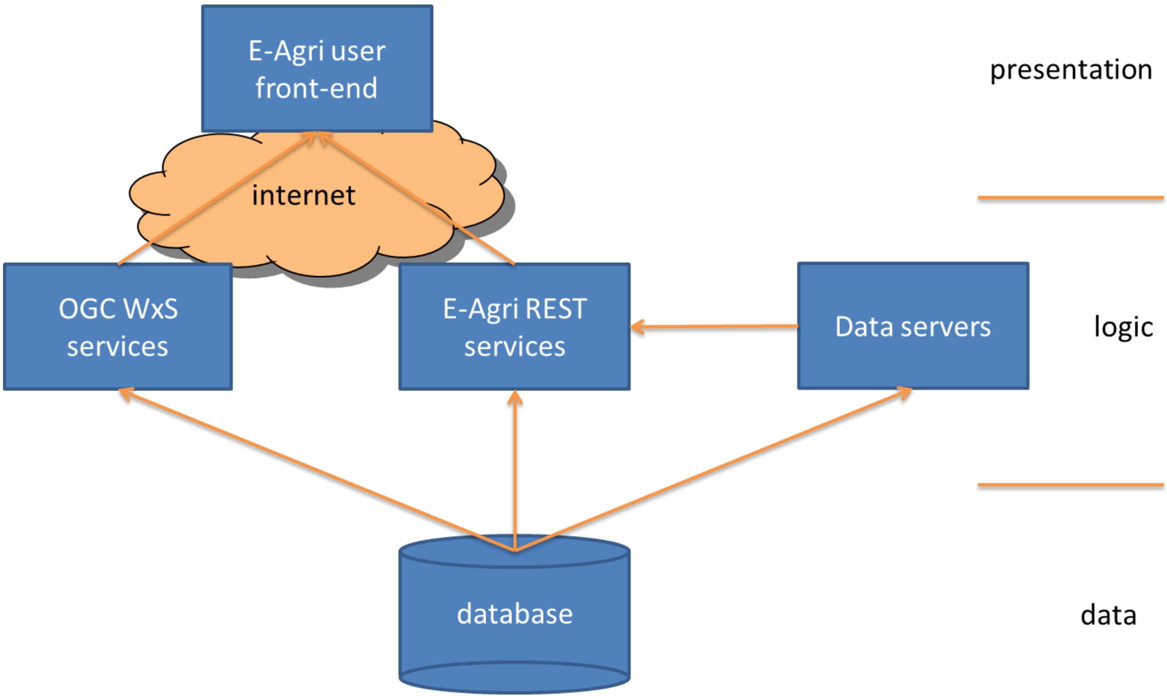


Figure 2 E-Agri service oriented architecture.

The data layer is the layers in which data used by the system are stored (see paragraph 3.2). The service layer is used to make logic available (see paragraph 3.3) The presentation layer consists of the user front-end application. This layer is crucial since it connects the end user to the information he or she needs. It will be discussed in paragraph **Error! Reference source not found.**.

3.2. Data layer

The data for the E-Agri viewer are stored in a PostgreSQL relational database management system (RDBMS). The PostGIS extension is used to store spatial data in this PostgreSQL database. Two databases are created in PostgreSQL: a database for the configuration information and a database containing the actual data for the data servers (see chapter 2). In addition, shape files are used to store the geometries needed for rendering the maps. This information could be stored in a PostgreSQL database as well.

3.3. Service layer

Logic is made available using a service layer. These are the components that serve a specific goal and they are created as independent but well standardised pieces of software. The E-Agri service layer consists of three types of services: out of the box OGC WxS services (see paragraph 3.3.1), custom made REST services (see paragraph 3.3.2) and custom made data servers (see paragraph 3.3.3).

3.3.1. OGC WxS services

E-Agri relies on out of the box OGC WMS and WFS services for rendering maps. A Web Map Service (WMS) produces maps of spatially referenced data dynamically from geographic information. A "map" is in this case a portrayal of geographic information as a digital image file suitable for display on a computer screen. The WMS standard is used for data visualization. Optionally, symbolization of a WMS layer can be controlled by a client using a Styled Layer Descriptor (SLD). A Web Feature Service (WFS) provides a client with access to the actual features encoded in Geography Markup Language (GML). A transactional WFS (WFS-T) allows a client to both retrieve and update geospatial data.

The E-Agri viewer uses GeoServer (http://www.geoserver.org) for this purpose.

3.3.2. E-Agri REST services

Within the frame of E-Agri, a number of tailor made REST services were developed. REST stands for Representational State Transfer. It is a client-server software architecture used to read and write resources. The E-Agri REST services implement the following functionality: read which regions of interest, themes, resolutions etcetera are available, read and write favourites and connect to the data servers. The E-Agri REST services are implemented in Java/J2EE using the Restlet framework (http://www.restlet.org)

3.3.3. The data servers

The data servers are the key to providing access to the actual data in a format suitable for the E-Agri user front-end. The E-AGRI data server aggregates data from the E-AGRI database following a REST (Representational state transfer) query over HTTP. The database content is cached in a hashtable mechanism for fast lookup. After the REST query is parsed either a PNG image or XML is generated. A PNG image is generated when grid data is requested, XML is generated when polygon data is requested. Legend colours are applied to the image or XML polygon tags according to specified user configuration retrieved from the database. In case of the PNG image, a world file is also generated so that the client can position it correctly in relation to the map visible in the user front end.

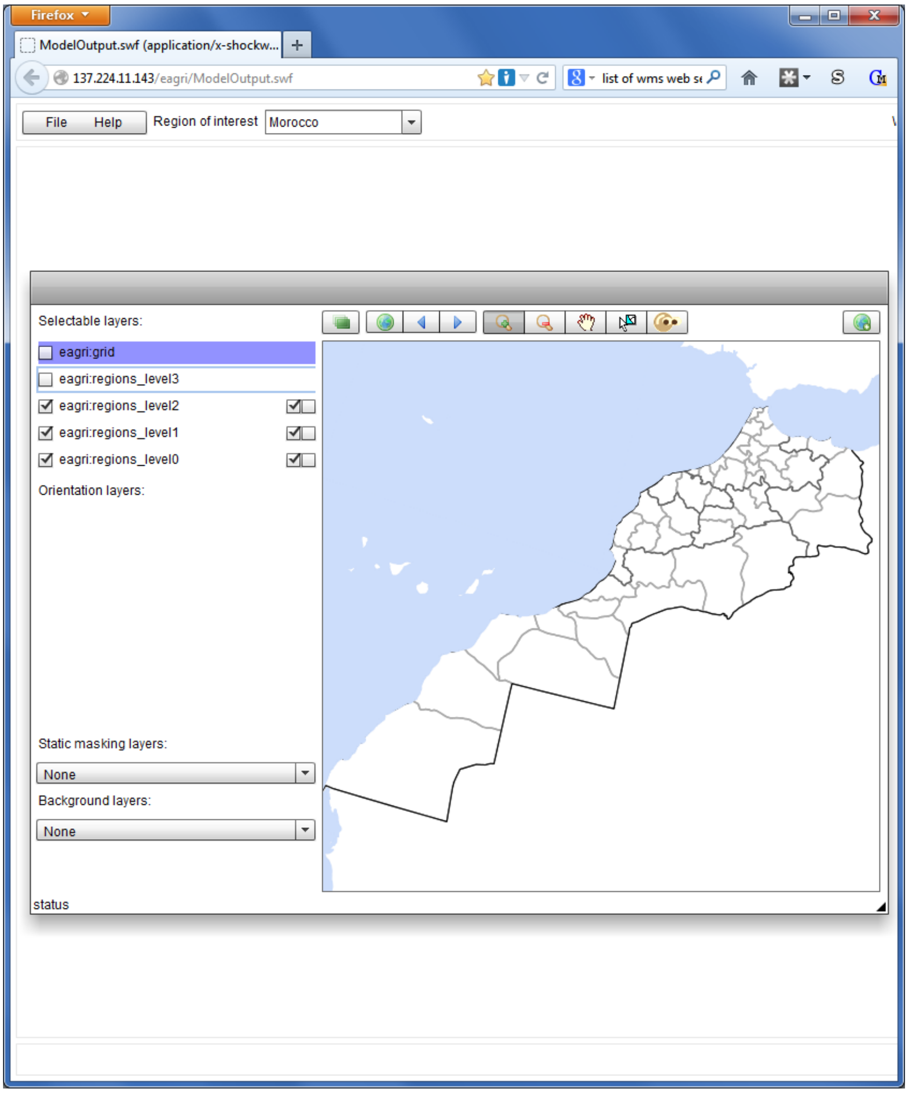
3.4. Presentation layer, the E-Agri user front-end

The E-Agri user front-end consists of a user friendly internet GIS client application. It runs in the web browser of the user using the Flash Player plugin. It is developed in Adobe Flex 4.5. Adobe Flex is a platform independent platform for developing rich internet applications. The LuigiOS framework for internet GIS applications is used to develop the E-Agri user front-end. LuigiOS is a framework which can be used to develop user friendly internet GIS applications for the Flex platform. It implements a number of standards of the Open Geo-spatial Consortium (OGC) as a client. The E-Agri user front-end uses the Web Map Service (WMS) standard and the Web Feature Service (WFS) standard.

The E-Agri user front-end is set up according to the widely used model-view-controller (MVC) software pattern. The model consists of the data and the logic providing access to these data. The view is used to visualize the model. The controller links model and view. The E-Agri user front-end is entirely data driven. This means that all the information the front-end needs to function is read from the database.

E-Agri supports both raster (e.g. grid) and vector (e.g. region) datasets. In case of a raster dataset, the data servers produce a geo-referenced bitmap. This bitmap is downloaded to the user front-end for rendering purposes. In case of a vector dataset, the data servers produces an OGC Styled Layer Descriptor (SLD). A SLD is used to control the visual portrayal of a map layer. This SLD is not downloaded to the client. Instead, the URL is passed on to the OGC WMS service which uses it to render a map. The result of this rendering process is downloaded to the user front-end and displayed.

4. The E-Agri viewers user interface



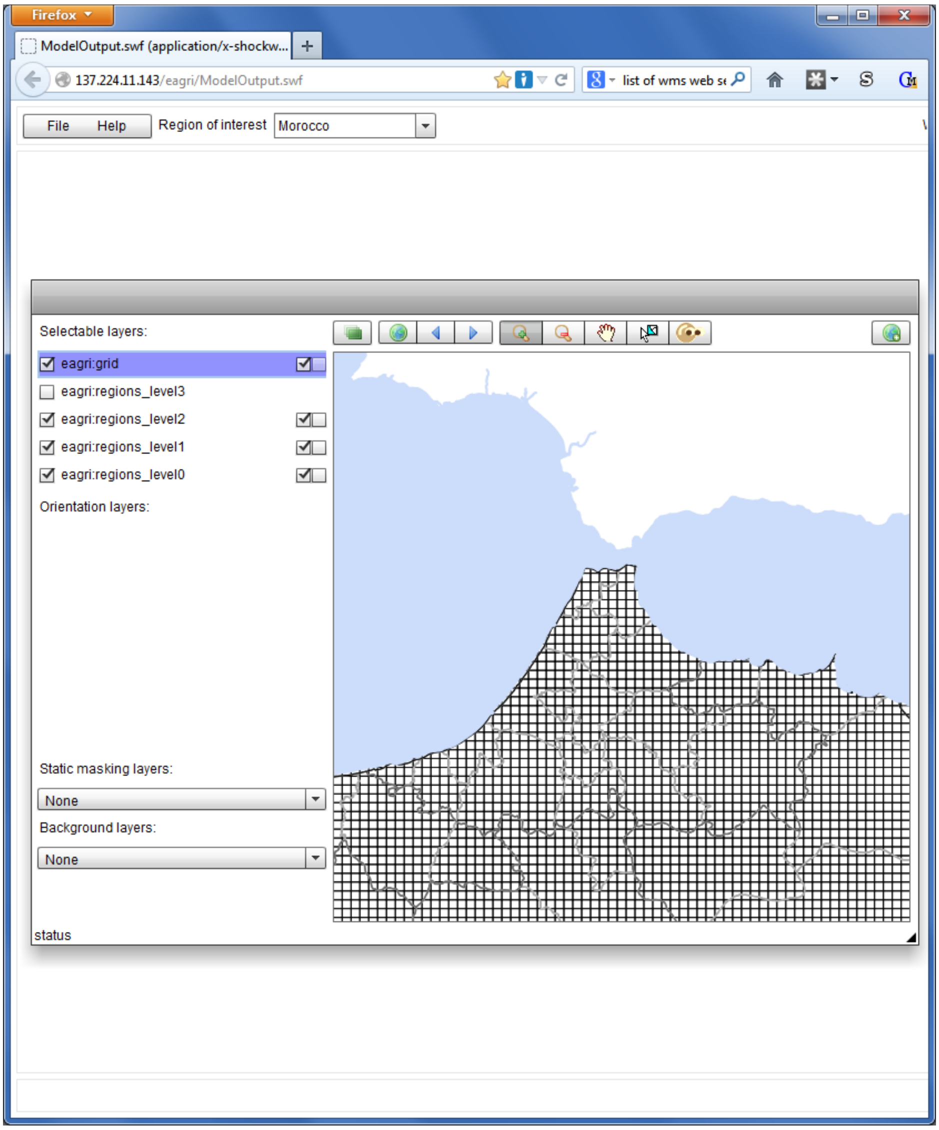
Region of interest: Morocco

- □ eagri:grid
- □ eagri:regions_level3
- ☑ eagri:regions_level2
- ☑ eagri:regions_level1
- ☑ eagri:regions_level0

Static masking layers: None

Background layers: None

Figure 3. The e-Agri map orientation window with the Moroccan regional division at levels 0, 1 and 2.


Firefox
ModelOutput.swf (application/x-shockw...
137.224.11.143/eagri/ModelOutput.swf
list of wms web se
File
Help

Region of interest: Morocco

Selectable layers:

- ☑ eagri:grid ☑
- □ eagri:regions_level3
- ☑ eagri:regions_level2 ☑
- ☑ eagri:regions_level1 ☑
- ☑ eagri:regions_level0 ☑

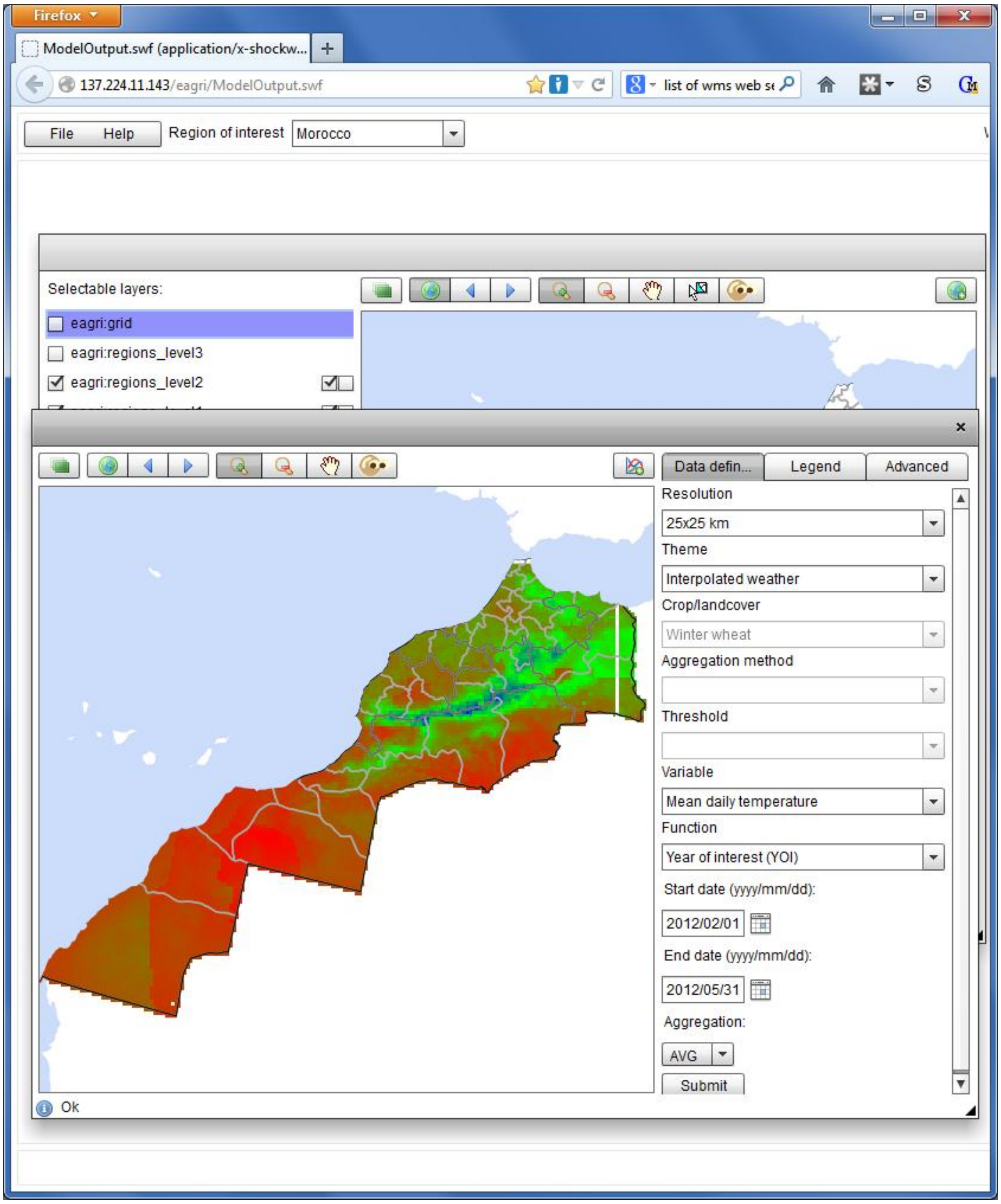
Orientation layers:

Static masking layers: None

Background layers: None

status

Figure 4. Detail of the Moroccan territory with the 9x9 km^2 CGMS grid of the Morocco CGMS enabled.



Region of interest: Morocco

Resolution: 25x25 km

Theme: Interpolated weather

Crop/landcover: Winter wheat

Aggregation method:

Threshold:

Variable: Mean daily temperature

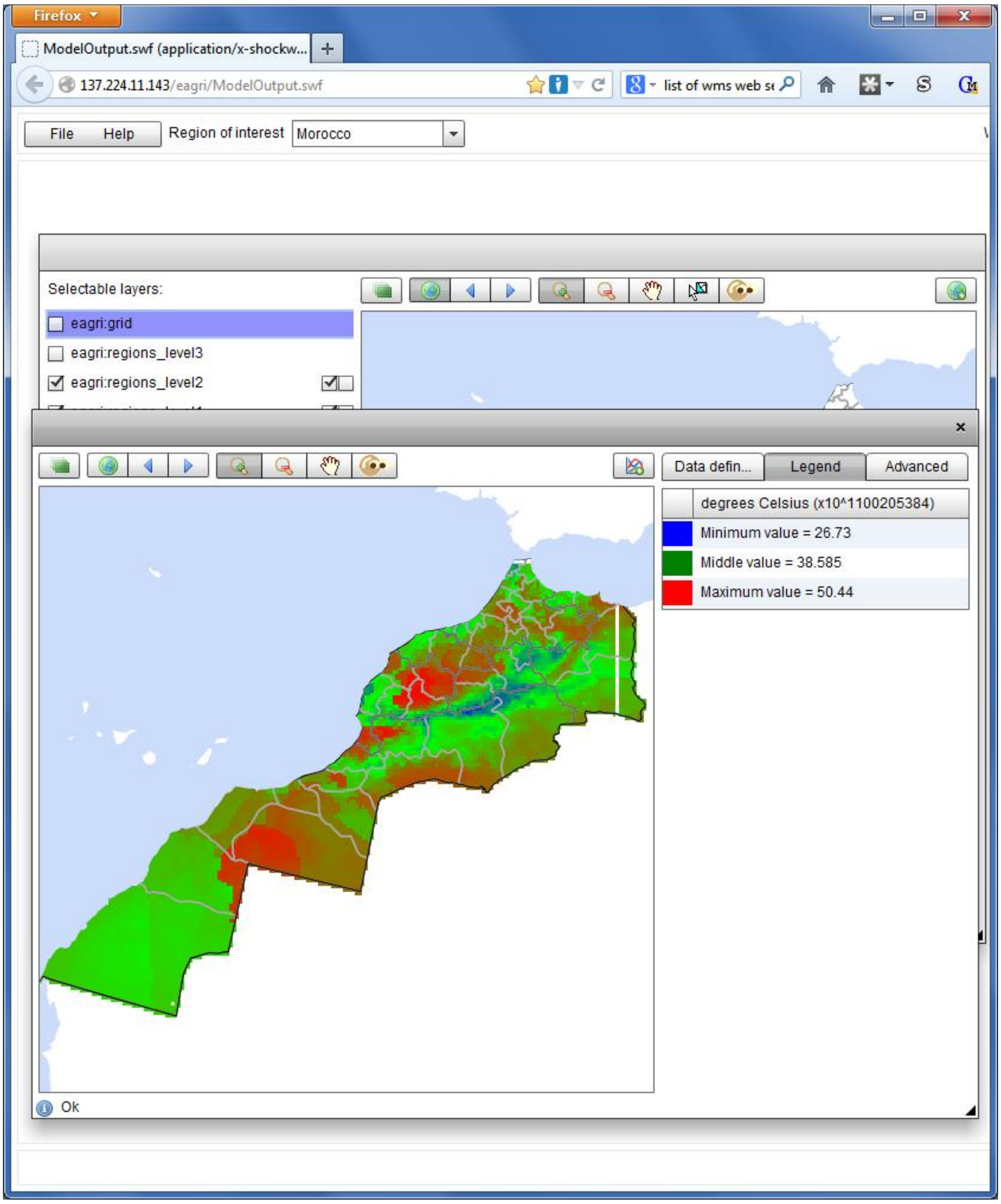
Function: Year of interest (YOI)

Start date (yyyy/mm/dd): 2012/02/01

End date (yyyy/mm/dd): 2012/05/31

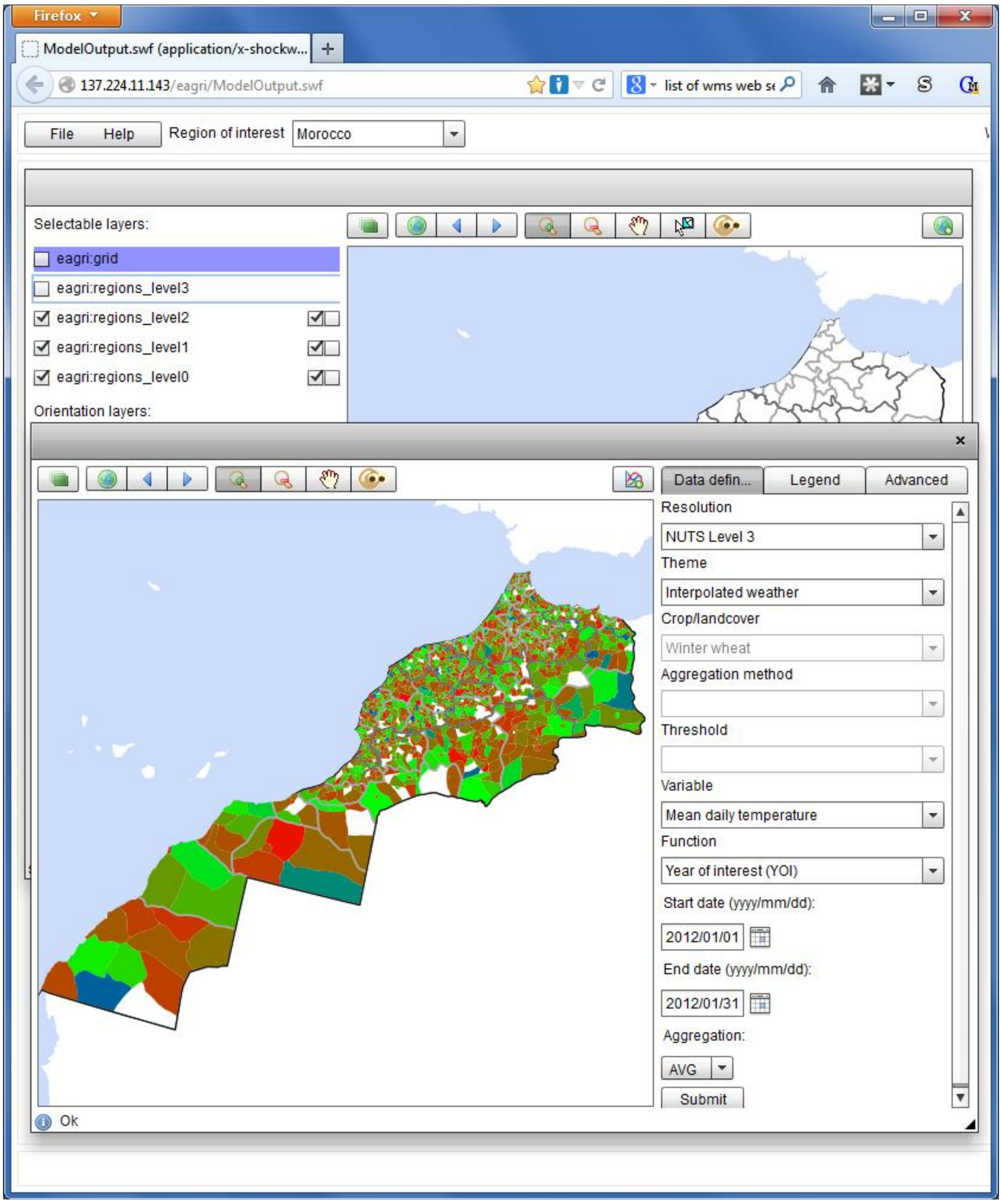
Aggregation: AVG

Figure 5. Map window showing the mean daily temperature in Morocco from 1st of February to 31st of May 2012 (blue = 0 °C, green=10 °C, Red=21 °C).



	degrees Celsius (x10^1100205384)
🟦	Minimum value = 26.73
🟩	Middle value = 38.585
🟥	Maximum value = 50.44

Figure 6. Map window showing the maximum of the maximum daily temperature in Morocco in 2012 at grid level (blue = 26 °C, green=38 °C, Red=50 °C).


Firefox
ModelOutput.swf (application/x-shockw...
137.224.11.143/eagri/ModelOutput.swf
list of wms web se
File
Help
Data defin...
Legend
Advanced
Submit

Region of interest: Morocco

Selectable layers:

- □ eagri:grid
- □ eagri:regions_level3
- ☑ eagri:regions_level2
- ☑ eagri:regions_level1
- ☑ eagri:regions_level0

Orientation layers:

Resolution: NUTS Level 3

Theme: Interpolated weather

Crop/landcover: Winter wheat

Aggregation method:

Threshold:

Variable: Mean daily temperature

Function: Year of interest (YOI)

Start date (yyyy/mm/dd): 2012/01/01

End date (yyyy/mm/dd): 2012/01/31

Aggregation: AVG

Ok

Figure 7. Figure showing the average temperature in January 2012 at NUTS3 level. ***Note****: A bug was discovered causing problems with joining of the data with the regions, this causes the irregular pattern of colours in the map.*

5. Current status

The status of the E-Agri viewers is as follows:

- The main interface has been developed and is mostly functional. Some minor interface details still have to be fixed. These include labelling, responsiveness of buttons and removal of superfluous elements
- The REST server and the Web Mapping Server are operational, tested and are generally working well.
- The E-Agri database has been designed and filled with data from the Moroccan CGMS for the tables GRID, REGIONS, GRID_WEATHER_INDICATORS and REGION_WEATHER_INDICATORS.
- The data server has been designed and tested for retrieving grid and SLD data from the GRID_WEATHER_INDICATORS and REGION_WEATHER_INDICATORS tables. Some deficiencies have been noted and these will be corrected in the near future.

The main work to be carried out in the coming months:

- Development of the interface functionality for making charts
- Development of the interface functionality for templates and favourites
- Testing of the data server for retrieving data for chart and identify from the GRID_WEATHER_INDICATORS and REGION_WEATHER_INDICATORS tables.
- Development of the data servers for retrieving crop indicators from the tables GRID_CROP_INDICATORS and REGION_CROP_INDICATORS.
- Testing of the application
- Development of documentation
- Rolling out of the application at the end-user premises including development of version that can work on ORACLE.