Grid Based Processing of Satellite Images in GreenLand Platform

Danut Mihon, Vlad Colceriu, Victor Bacu, Dorian Gorgan
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
{vasile.mihon, vlad.colceriu, victor.bacu, dorian.gorgan}@cs.utcluj.ro

***Abstract*—Geographical Information System (GIS) applications that process large amount of data require intensive usage of hardware capabilities provided by distributed platforms, such as the Grid infrastructure. Due to the constant demand of data availability and data sharing, without concerning its format and size, a new software solution is needed. GreenLand is a system capable to provide such a solution, based on its constituent modules: GreenLandGUI, gProcess, ESIP, WorkflowEditor, and OperatorEditor. This paper highlights each of them and how they interact in order to create a platform capable of fetching, processing, and visualizing large amount of data exposed in a uniform and standardized manner.**

I. Introduction

The description and processing of natural phenomena and experiments, from different domain fields, is a complex process that usually involves: a solid understanding of the background context, the collection of the adequate input data set, the syntactic and semantic description of the adopted solutions, the execution over distributed environments in order to speed up the entire process, optimized tools for partial results integration, and some special interaction techniques for visualizing and analyzing the final outputs.

This paper describes the theoretical concepts and practical solutions involved in solving the previous mentioned issues, through the perspective of the GreenLand platform [1]. This system was developed within the enviroGRIDS project [2], and its functionality was validated through three case studies: Black Sea catchment hydrologic modeling, land cover/land use analysis of the Istanbul geographic area, and the Rioni river hydrologic analysis [3].

Modeling large scale environmental use case scenarios is most of the times a challenging task, due to the multitude of conditions, restrictions, and algorithms that need to interconnect in order to provide the desired output. Regarding this aspect, the new Geographic Information System (GIS) applications try to provide advanced interaction techniques that facilitate the end-user work and increase the usability of the entire platform.

In order to overcome these issues, the adopted solution was to represent the entire use case as a workflow, where each node identifies one of the algorithms (function) of the main process. The uni-directional edges of the graph specify the interaction between the algorithms and how they communicate in order to generate the output results.

This type of approach is useful in many cases, but when the user is required to manually specify all the connections, errors may occur. This is the main reason why the GreenLand platform provides the WorkflowEditor tool [4] for an easy and flexible description of the workflows.

Executing such large use cases on standalone machines is not a feasible solution. On the other hand, the correct approach is to use the storage and computation benefits of the distributed infrastructures (e.g. Grid, Cloud, clusters, multi-core machines). This way an execution speed up will be obtained, by partitioning the main process into smaller tasks and execute them in parallel.

The GreenLand platform uses the Grid infrastructure [5] in order to improve the execution time, where each node (or a group of nodes) of the workflow is processed onto a different physical machine. The gProcess platform [6] connects the two environments and acts like a middleware between them. The input data set and the expanded structure of the workflow are the only information required by this platform.

Based on the process complexity, the gProcess is able to group the tasks and to discover the optimal execution schema. Monitoring the Grid-based processing and sending the feedback to the GreenLand system is another feature offered by this platform.

In order to provide useful results that could be reused by external applications (without further processing) the GreenLand platform implements the WMS, WCS, and WPS OGC standard services [7]. They allow the satellite images retrieval and exposure in a standardized manner, and facilitate the user actions regarding this types of tasks.

II. Related work

The availability of high performance applications, broadband Internet access, high storage and processing capability devices, and the Web technologies accelerate the usage of geographic information into our daily lives. GIS applications are widely spread across Earth science domains, such as: hydrology, meteorology, agriculture, air and water pollution, urban planning, etc. They offer standard services for storing, processing, analyzing, and visualizing spatial data of different types and formats.

Some of the most known such platforms work either on standalone or distributed infrastructures. In the first category we can include Sextante [8], uDig [9], and GRASS [10]. As for the distributed environments class, the QuantumGIS [11] tool could be mentioned.

The Sistema EXTremeno de ANalisis TEritorrial (Sextante) is an open source spatial data analysis library that contains more than 300 geospatial algorithms that handle raster and vector data types, and provides rich common functionalities, useful for the entire geospatial communities. It allows the creation of complex workflows, in an interactive manner, but it does not support the sub-workflow concept (nodes imbrications within other nodes) as the GreenLand platform does.

The main goal of the uDig is to fill the functional gaps between the geospatial standards and the open source communities. It provides integration support with the latest Open Geospatial Consortium (OGC) standards, and it is mostly used to represent database geospatial information in a simple and interactive manner. Similar with this tool, the GreenLand platform adheres to the latest OGC standards, by offering support in data retrieval, execution, and visualization.

The Geographic Resources Analysis Support System (GRASS) is based on GDAL and OGC libraries, and provides features for reading and writing various raster and vector data formats. It offers more than 400 geospatial algorithms and it can be easily implemented in other platforms (this is the case of the GreenLand system that integrates its functions directly within the Web services, consumed by the end-user).

The Quantum GIS system is useful for spatial data processing, displaying data layers over interactive maps, performing distance measurements, creating map symbologies, data re-projection, etc. Another important aspect is the support it offers for distributed and parallel computations, in case of large experiments. The workflow-based description of the scenarios is the main advantage of the GreenLand platform, and proves useful especially when dealing with a large set of algorithms that need to be connected by certain rules.

The GreenLand platform allows the parallel and distributed execution of the tasks, and benefits from the computing and storage characteristics of the Grid infrastructure. One of the important advantages of this solution (compared with the previous mentioned environmental applications) is the execution speedup, obtained for large scale use cases (experiments). Because the entire process is partitioned into multiple tasks, the system is able to schedule them onto different physical Grid nodes. This means that the total processing time is significantly reduced, the only overhead appears when transferring input data set and combining the partial results in order to generate the final output.

The ability of executing the use cases over the Grid infrastructure is the main feature that differentiates the GreenLand platform from the previous mentioned standalone applications, and makes it suitable for implementing large environmental scenarios from different Earth Science domains.

The flexible and interactive use cases description is the main advantage of the GreenLand platform compared with the QuantumGIS application. Instead of independent execution of all the inner algorithms, this solution allows the relationships definition between them and the possibility to create a single execution thread for the entire workflow.

The gProcess platform is used as a middleware between the Grid infrastructure and the user requests, and provides support for: workflows partition into tasks, scheduling mechanisms, and execution and monitor features. The GANGA [12] and DIANE [13] tools represent two of the alternatives to this approach. The first one is a job management tool, capable of scheduling the entire execution process. On the other hand, the DIANE is mostly used for monitoring the processing, and gives periodic feedback about its status (e.g. the number of executed jobs, on what Grid nodes the tasks are resident, etc).

The main advantage of the gProcess platform (compared with the features provided by these two alternative applications) is the ability to interpret the workflow-based data structures, and to create groups of nodes, similar in complexity. This way a balanced Grid execution is obtained.

III. Theoretical concepts and implementation

This section highlights the main concepts related to the possibility of describing the spatial data execution process as complex workflows that encapsulate within their nodes an abstract representation of an algorithm, function, or experiment.

A. *Spatial data classification*

Based on the data structure and on the collection mechanisms, the spatial data are grouped into: satellite images, airborne images, and ground data measurements.

The satellite images are obtained onboard the artificial satellites that orbit around the Earth, collecting information about its surface (e.g. temperature, humidity) by scanning it in multiple frequency levels. The collected data are organized in bands that contain on each layer one of the measured characteristic. The GreenLand platform supports various satellite images, regardless of their number of bands: Landsat (organized on 7 layers), ASTER (15 bands), 36 levels MODIS images, etc.

The airborne data are useful in applications that require high accuracy results, because these images scan the Earth's surface in more detail. Some of the most known products (e.g. SPOT and QuickBird) are also supported in the GreenLand framework.

The information obtained from ground based measurements has the best accuracy and penetrate in dense areas where the artificial sensors are not able to record the data. They are used especially for calibrating different experimental models, related to a small geographic region (due to the limited measurement capacity).

The GreenLand platform offers support for all these data categories, but in this paper only the satellite images are presented in more detail, due to the requirements of the three case studies highlighted in the introduction section.

The GreenLand platform uses the workflow concept for use cases development. The physical execution of such graphs can be defined as a multi-variable function P that produces, in a finite amount of time, a valid result, based on a specific input data set. It also contains sub-processes (represented as the nodes of the graph) combined in a specific order that corresponds to the use case description flow. Two types of processes were identified in the context of the GreenLand platform: basic operators and complex workflows.

B. Basic operators

The operator is the smallest unit that can be processed, without the possibility to divide it into atomic modules. It integrates the representation of an algorithm (e.g.: vegetation indices, atmospheric correction functions, statistics computation, distance measurements, etc.) under the form of an executable file that is further used at runtime over the computing infrastructures.

A formal description of the basic operator is given bellow,

$$O(IN, OUT, DATA) \tag{1}$$

where:

- $IN = \{in_1, in_2, \ldots, in_n\}$: all the available inputs data set;
- OUT: the output of the operator;
- $DATA = \{d_1, d_2, \ldots, d_m\}$: all the available data resources that can be used for inputs instantiation.

Each input ($in_k, k = \overline{1, n}$) and the output is a triplet <name, value, type> that has a name, a value (or resource from the *m* possible entries), and an associated type. The order in which the inputs are specified has a major impact on the final result of the core process. In this case, there should be a perfect match between the *n* arguments and the variables of the algorithm described by the operator

C. Complex workflows

The description of natural phenomena (experiments, use cases) that belongs to different Earth Science domains can be modeled as workflows (graphs) that contain a collection of operators, interconnected by uni-directional edges. Using this approach, we can achieve the goal of optimal representation and data model organization of the natural phenomena.

From mathematical point of view, the workflows can be described as in (2)

$$W(IN, OUT, DATA, N, C) \tag{2}$$

The first three arguments of the function *W* have the same significance as in the case of the basic operators. The only difference is the fact that the workflows allow the possibility to specify multiple outputs ($out_1, out_2, \ldots, out_s$), compared to a single operator's output. Information about the inner layout of the graph is stored in the last two arguments of the function:

- $N = \{n_1, n_2, \ldots, n_u\}$: a finite list of nodes that, in the basic form are identified as operators. A more complex node is called sub-workflow that has the ability of storing other graphs within;
- $C = \{c_1, c_2, \ldots, c_v\}$: a list of uni-directional edges that describe the execution flow inside the graph.

The conceptual representation of a complex workflow is described in Figure 1. As can be seen, it contains *u* operators (marked with $OP1, OP2, \ldots, OPu$). The inputs list of the workflow is distributed to its inner basic operators. This part is extremely important because it influences the final output result. For example, if we switch the inputs of the $OP1$ and OPu than the obtained results are different from the original ones. This change propagates to the next description levels ($OP3$ in this case) and affects the out_3 and *outs* of the core workflow.

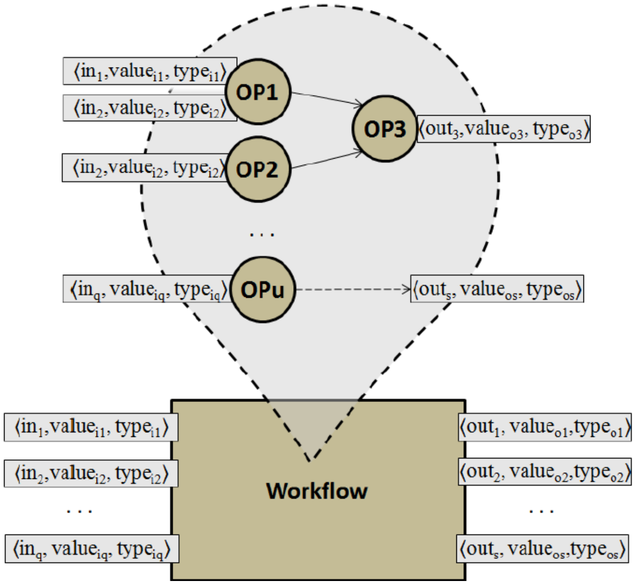

$\langle in_1, value_{i1}, type_{i1} \rangle$
$\langle in_2, value_{i2}, type_{i2} \rangle$
OP1
$\langle in_2, value_{i2}, type_{i2} \rangle$
OP2
OP3
$\langle out_3, value_{o3}, type_{o3} \rangle$
. . .
$\langle in_q, value_{iq}, type_{iq} \rangle$
OPu
$\langle out_s, value_{os}, type_{os} \rangle$
$\langle in_1, value_{i1}, type_{i1} \rangle$
$\langle in_2, value_{i2}, type_{i2} \rangle$
. . .
$\langle in_q, value_{iq}, type_{iq} \rangle$
Workflow
$\langle out_1, value_{o1}, type_{o1} \rangle$
$\langle out_2, value_{o2}, type_{o2} \rangle$
. . .
$\langle out_s, value_{os}, type_{os} \rangle$

Fig. 1. The abstract representation of the workflow

The example from Figure 1 highlights only the mathematical significance of the concepts, but at runtime, these inputs are instantiated with the data specified by the end-user. In the GreenLand framework multiple data types are supported, such as: generic satellite images (e.g. Landsat, MODIS, Aster, etc.), vector shape files, projection files, integers, strings, etc.

Another important aspect is the connection establishment between the operators, because it describes the entire execution flow of the use case (scenario). Even though the GreenLand platform offers support for multiple users' categories (e.g. data providers, decision makers, specialists in Earth Sciences, regular users), this step is recommended to be realized by a domain field specialist.

To exemplify the basic operator and workflow concepts, the Istanbul case study is very useful. Shortly, this experiment consists in classifying the vegetation, water, and urban areas around the geographic region of Istanbul, by implementing multiple algorithms that interconnect at four stages: spatial data pre-processing, vegetation index computation, satellite image classification, and the accuracy statistics generation.

The algorithms that are used in each stage can be defined as basic operators (e.g. geometric correction of satellite images, NDVI, EVI, Density slicing, etc.). The entire Istanbul scenario can be described as a workflow, where the nodes are identified as operators and the data flow process is described through uni-directional edges.

One important aspect is the fact the GreenLand platform limits to the acyclic graph structure. This means that the system

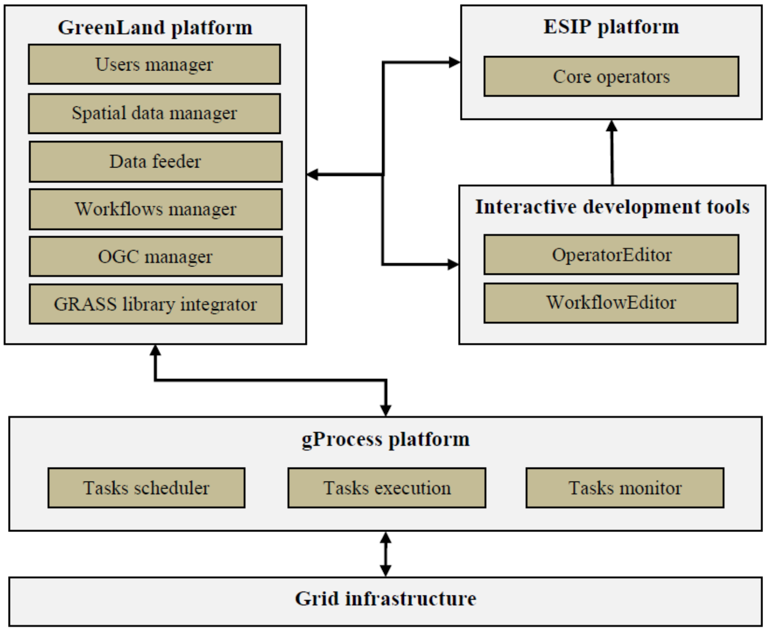


Fig. 2. System related architecture

is protected from infinite looping cycles. One future research direction is to offer support for repetitive structures (e.g. for, while) and the conditional ones.

D. System related architecture

The GreenLand platform is built upon other modules that interconnect in order to fulfill the satellite images processing requirements (Figure 2). The main entry point is the GreenLand system that listens for the user-actions at the graphical interface level. Once a new event is triggered, it is automatically interpreted and converted into an internal representation. In cases that involve satellite images processing, the system also stores the operators and the workflows specified by the user together with their input data set.

The Environment oriented Satellite Image Processing (ESIP) is responsible for providing the core algorithms (operators) to the GreenLand platform. It contains only the main operators that are useful in different Earth Science domains, while the GreenLand allows the development of customized operators and workflows (based on the OperatorEditor and WorkflowEditor tools).

This delineation is also useful when installing the platform on other environments, because only the ESIP is exported together with the data schema, while the GreenLand is ported with an empty data repository.

Once the use case (scenario) development process is complete, it can be executed over the Grid infrastructure. This is possible based on the gProcess platform that acts like a middleware, translating the client requests into commands recognized by the Grid environment. It is also used for sending the execution feedback to the application's graphical user interface.

When the execution is finished the final output can be analyzed by using the online specific tools or it can be downloaded onto the user's local machine. The GreenLand platform offers support for visualizing these results in an interactive manner and promotes data sharing with external systems and applications.

1) GreenLand general overview: It is a GIS platform that provides services for geospatial data retrieval, processing, and visualization. The frontend of this platform acts like a gateway that masks all the complex mechanisms that are implemented within the system, such as: workflow partition into tasks, the scheduling process, Grid based execution, data interoperability with external platforms, standards implementation, etc. [1].

The GreenLand is an open platform that allows data import from three main sources: directly from the user local machine (regular upload), from File Transfer Protocol (FTP) data repositories, and by OGC means. Depending on the requirements, the users are able to utilize one of these methods, or to combine them as desired.

In case of near real time processing algorithms the idea of automatic data fetching from different remote repositories is very useful. This feature is especially used in prediction experiments that require a large data set for the calibration process. The GreenLand platform allows the automatic data extraction, based on the FTP protocol.

The Mosaic Black Sea catchment workflow is a perfect example to offer insights about this solution. The user is requested to specify the remote repository that stores these data, the processing time period, and the satellite image bands that he is interested in. Once these steps are completed the system processes the user's request and automatically starts collecting data, by applying the filters selected at the GreenLand graphical interface level.

Because various spatial data types are used in the GreenLand framework, new scripts were needed in order to interpret and process these data. GRASS library proved to be flexible enough to allow its functionality extension to fulfill the GreenLand requests.

On the other hand the GRASS library fits perfectly on standalone platforms, but needs several adjustments in order to run properly on distributed environments.

The adopted solution was to identify the GRASS version that comply the best with the Grid infrastructure, and to describe all the operators based on that type of library. At runtime, the GRASS files were packed together with the inputs specified by the user and transferred to the Grid machines. Locally, on each Grid node, the operators (algorithms) are executed similar to standalone machines.

One of the GIS fundamental ideas is to develop an open platform that is able to interoperate with external systems in terms of data sharing. Based on the OGC standard that was implemented within the GreenLand framework, the system is able to: query and download remote data directly in the GreenLand repository (through WCS operation), interactively visualize spatial information (based on the WMS service), execute the scenarios in a standardized manner (by using the WPS service), and to publish the Grid processing results to external remote storages.

The flexibility characteristic is another main aspect that was taken into account when developing the GreenLand system. First of all it can be used as a Web-based platform. In this case the users are able to perform different actions directly from the application frontend, in an interactive and user friendly manner. The complexity of the internal mechanisms is hidden from the users, and only light weighted operations are exposed.

Extending the GreenLand functionalities to other activity domains (e.g. archeology, physics, etc.) represents the second utilization mode of this platform. This feature can be achieved by integrating the constituent services directly into the backend architecture of other systems. Because of the GreenLand flexibility, the modules described in Figure 2 do not necessarily need to work in their original schema. Instead their installation can be extended to different physical machines (e.g. the operators repository can be resident on other servers).

Platform interoperability is the third way of using the services exposed by the GreenLand system. Because it implements the OGC standard, external applications are able to connect to the GreenLand (by means of standard services) in order to: query, visualize, and download the satellite images made available by their owners, and to process the GreenLand workflows exposed as WPS items

2) ESIP platform: The Environment oriented Satellite Image Processing (ESIP) [14], [15] can be defined as a set of basic operators (e.g. radiometric correction, vegetation index computation, histogram generation, mathematic computations, etc.) that handle various types of data, such as: satellite images, vector data, ground based measurements, etc.

The GreenLand platform provides services for the GIS domain. By default, when a fresh copy of the platform is installed, it contains a predefined basic operators set, resident in the ESIP platform. As the system develops, new operators can be added to the ESIP repository.

The content of this platform is rarely updated, and once an operator is implemented it is recommended to maintain its functions (because it may be already used in other workflows and the change of its internal structure will affect the entire data flow). Adding new operators to this repository can be done interactively, through the OperatorEditor tool (Figure 2). More details about this application are presented in the next section.

In other words the ESIP platform is recommended to be used as a repository of operators that provides information to different instances of the GreenLand system.

3) Interactive development tools: There are two interactive applications (OperatorEditor and WorkflowEditor) [4] which are integrated within the GreenLand platform and used for basic operators and workflows development. They are called interactive because they facilitate the entire implementation process, allowing the users to easily describe the inner functionalities of the algorithms (as operators) and complex use cases (as workflows).

There are several important characteristics about the operator concept: a list on inputs, an output, and the inner algorithm (function or formula) that describes the operator's behavior. Each input/output has the triplet form (<name, value, type>) that makes it easier to distinguish among other items, and to map various data formats.

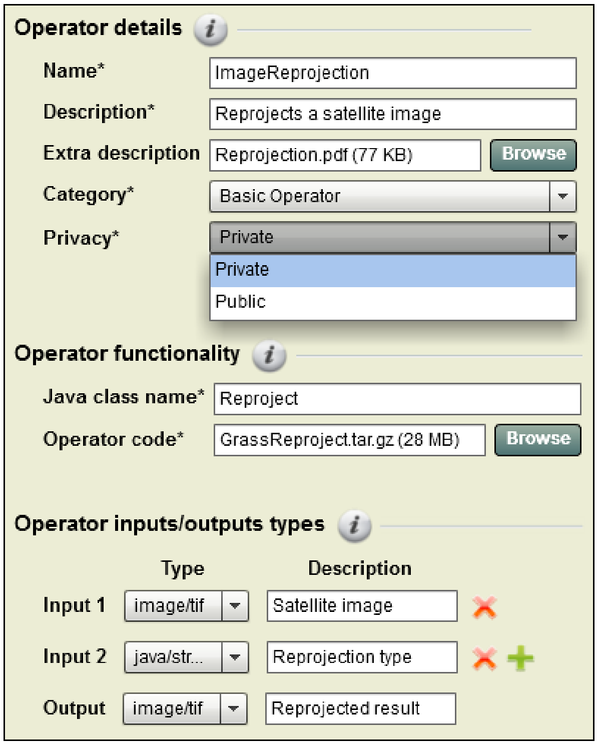
The OperatorEditor tool allows the user to describe the inner functionality of the operator by extending a specific Java API. The resulting algorithm has several input variables and one output that stores the result generated when instantiating the algorithm with the input data resources.

Once the operator is implemented its owner has the possibility to make it available to other users. These users do not have access to the kernel of the algorithm and do not know what inputs it expects and what its functions are.

For this reason the OperatorEditor tool provides to the owner of the operator an interactive mapping technique for specifying all these features. As can be seen in Figure 3 the user is able to specify the operator's name and a short description. It is also recommended to give a full description of its functionality and to attach it to the operator by means of external files (PDF in this case). Once the operator is completed, it can be shared with the entire users' communities (by making it public).

The user is also able to map the algorithm's inputs and output to the operator's ones, by using the same interaction technique as the one presented in the bottom side of Figure 3. The order of the inputs must completely match the order in which they were specified within the Java algorithm, otherwise the final output result will be altered.

As mentioned in previous sections, the GreenLand use cases are described as workflows. When this process is done manually (e.g. by means of XML tags) it is most likely that errors may occur. The WorklfowEditor tool was developed in order to avoid such issues and to facilitate the workflows implementation by providing: several interactive techniques, validation mechanisms, layout algorithms, and proper adjustments performed automatically by the system itself.



Operator details

Name* ImageReprojection

Description* Reprojects a satellite image

Extra description Reprojection.pdf (77 KB) Browse

Category* Basic Operator

Privacy* Private

Private
Public

Operator functionality

Java class name* Reproject

Operator code* GrassReproject.tar.gz (28 MB) Browse

Operator inputs/outputs types

	Type	Description
Input 1	image/tif	Satellite image
Input 2	java/str...	Reprojection type
Output	image/tif	Reprojected result

Fig. 3. Interactive description of the ImageReprojection operator

Usually each workflow (Figure 4) contains a list of operators (marked with the circle graphical symbol) and sub-workflows (highlighted as rectangles) that integrate other inner nodes. The imbrications can extend to multiple levels, while the user has the possibility to navigate through these structures by using the mouse device.

On the other hand the workflow development process itself is highly interactive, and includes:

- Placing the graph nodes with the drag and drop actions;
- Connecting the items by tracing a uni-directional edge with the mouse (once a node is selected the system enables only the inputs that have the same data type). At this stage it is worth mentioning that only nodes that have the same type are allowed to be connected;
- Nodes reposition on the canvas surface, while the corresponding edges update automatically;
- Navigating through the sub-workflows hierarchy;
- Auto-arrangement of the nodes by using one of the automatic workflow layout algorithms that minimize the surface on which the workflow is represented and reduces as much as possible the number of intersections between the edges. Creating edges of similar lengths and preserving their angular resolution are also taken into account when using the automatic layout algorithms.

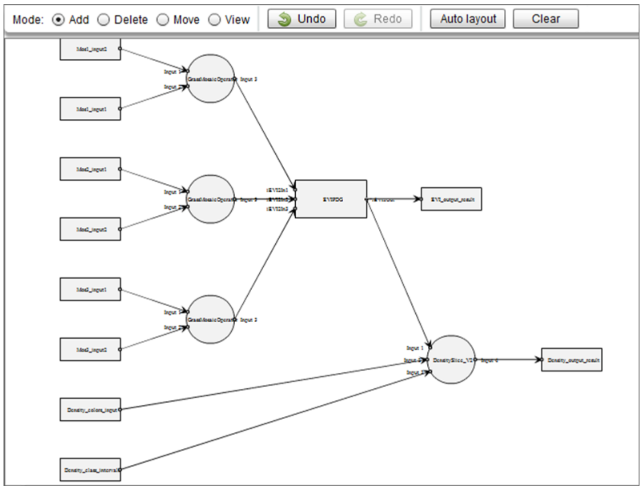


Fig. 4. The workflow development process based the WorkflowEditor tool

Once the workflow development process is complete, it becomes available in the GreenLand platform, and can be used in further Grid processing. The inputs of the workflow represent the inputs collection of all its internal nodes, excepting the case when one input is connected to the output of another item.

4) gProcess platform: Once the workflows are instantiated with the inputs specified by the user at the graphical interface level, the Grid execution begins. The gProcess acts like a middleware that interprets the processing user-requests, converts them into an internal representation structure, and forwards them to be executed over the Grid infrastructure [6].

The communication with the gProcess platform is accomplished by using a customize XML format that contains the description of the entire GreenLand workflow. The XML structure is slightly different from the one generated by the WorkflowEditor tool, meaning that the sub-workflow concept is not included. Instead, each node of the graph is represented on the same hierarchy level.

The main advantage of this representation approach consists in the possibility of creating execution groups (that run on different Grid machines) in order to balance the processing of the entire workflow.

Taking into account this aspect, we can say that the gProcess platform is able to partition an execution graph into smaller tasks that are interconnected upon the relations specified within the XML structure. Each task is then submitted to a specific Grid machine, together with all its input data and additional dependencies.

The gProcess platform is also responsible for monitoring the entire Grid execution process. When interconnected tasks are executed on multiple machines, it is most likely that one node has to wait for the other one to finish. In this case the gProcess is involved in managing the data transfer between the two entities, and to generate the final output of the workflow based on these partial results.

Each gProcess task is considered to have one of the following statuses: submitting, running, completed, canceled, and failed. The input data transfer to the Grid nodes takes place in the submitting stage. At this moment the gProcess is also partitioning the workflow into tasks, and schedule their execution by mapping each task on a specific Grid node.

It is worth mentioning that initially a list of all available Grid machines is retrieved and tasks are assigned to consecutive worker nodes from this queue. If the number of tasks is greater than the length of the list, the remaining processes are marked with the pending status. Once a Grid machine completes, it receives a new task to process. The running stage consists in executing the workflow modules over the Grid infrastructure. When a task finishes, its output is automatically transferred to other nodes that require this information as input.

The workflow execution completes when all its tasks are processed correctly. The final result of the workflow is generated by combining the partial outputs of each task, based on the XML representation file.

The failed status identifies an error that was encountered during the execution process. The user is also able to stop the Grid based workflow processing. In this case the gProcess will mark this execution as cancelled.

5) Grid infrastructure: It can be described as a worldwide computer network that offers support for storing and processing large volume of data. The storage nodes are called Storage Elements (SEs), while the computational stations are referred as Computing Elements (CEs) [5].

The motivation behind using the Grid, as a processing infrastructure for the GreenLand platform, is that in case of complex scenarios the standalone machines do not provide enough computation power to execute them in reasonable amount of time. In order to speed up the entire execution process, this platform benefits from the Grid parallel and distributed capabilities regarding the large data processing.

On the other hand the GreenLand platform can also be used for executing the basic operators that are simpler algorithms that take a few seconds to compute. In these cases the Grid infrastructure is not needed, because it will slow down the entire computation process, taking into account:

- The time required to partition the workflow into tasks and to schedule them onto the available Grid nodes;
- The time required for transferring the input data sets, together with the additional dependencies;
- The actual Grid execution of all tasks and the final output generation, based on the partial results of each task;
- The time required to transfer the workflow result from the SE node to the GreenLand server and to make it available to the user.

In order to avoid using the Grid for unnecessary executions, one of the research directions for extending the GreenLand platform is to implement a decision module that is able to redirect the processing (to Grid or multi-core infrastructures) based on the complexity of the workflow. This research is only at the beginning, but it proves to be useful in increasing the platform flexibility and scalability.

IV. Experiments

This chapter exemplifies the theoretical concepts, described in the previous sections of this paper. The goal of the conducted experiment was to analyze the water quality/quantity for the Black Sea catchment in the last 10 years. The MODIS satellite images were used as input data sets for this use case. In order to simplify the entire execution process, one additional request was to automatically collect the data from remote repositories, by keeping the user graphical interface as simple as possible [3].

A. General description

The MODIS satellite produces data by scanning the Earth's surface on an 8-days time basis. This sensor partitions the entire Black Sea catchment into 12 adjacent tiles, represented as satellite images. Regarding all these aspects a new GreenLand workflow was needed in order to:

- Recompose the Black Sea catchment area from the 12 adjacent tiles, and apply the analysis algorithms on the extended model;
- Automatically collect MODIS satellite images from remote repositories, over a specific time period;
- Handle both MOD15 and MOD16 products. The differences between them are the internal bands organization and the data contained within each frequency interval;
- Extract information relevant to the use case requirements. Because the MODIS data are organized in multiple bands, only specific information is required for this particular experiment (e.g. the Evapo-Transpiration, the Photosynthetically Active Radiation, etc.);
- Optimize the entire execution process by performing parallel computations over the Grid infrastructure;
- Expose the results to external platforms, by using the OGC standard.

B. GreenLand workflow development

A new workflow was implemented (called BlackSeaMosaicPDG or Mosaic12) that based on 12 MODIS satellite image input generates a single model for the entire Black Sea catchment. The internal algorithm is based on the classical Mosaic operator that combines 2 bands in order to generate a single satellite image, containing the extended area.

On the left side (in Figure 5) there are 6 Mosaic operators that receive the 12 input images. Each of the next levels reduces the number of the operators, until the final result image is generated. The inputs order is important and has to match the horizontal or vertical position of the adjacent tiles. The Mosaic12 workflow can be created directly from the WorkflowEditor tool that allows the interactive placement of the operators and the specification of the inter-nodes relations by using the mouse device.

C. Input data specification process

The main goal of this experiment is to model the Black Sea catchment area, based on information dated from 2000 up to 2010. In order to optimize the entire execution, the workflow was implemented to process one year at a time (Figure 6

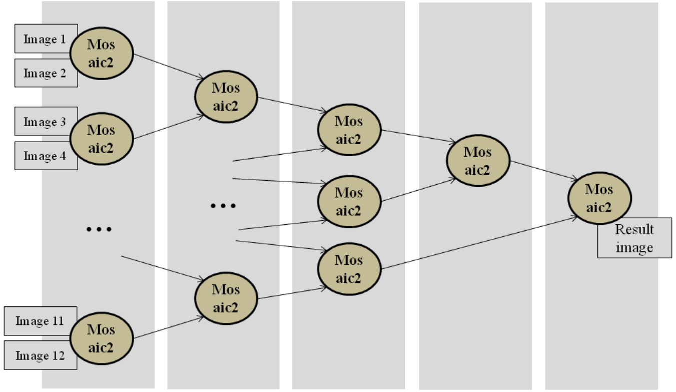


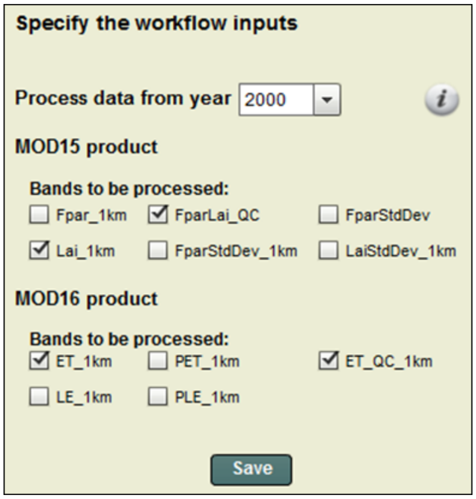
Fig. 5. The internal representation of the Mosaic12 workfow

highlights how the user is able to specify the processing time period).

One of the requirements of this experiment was related to the workflow capability of being able to handle MOD15 and MOD16 products. This is the reason why the graphical interface (Figure 6) allows the user to select bands from both products. By default the Evapo-Transpiration and the Photosynthetically Active Radiation items are already selected.

Until now the user specified only the metadata for the workflow, meaning the processing time and the relevant satellite image bands. But there were no specifications about the actual data. This process is done internally by the GreenLand platform that query at runtime the entire content of the NTSG (ftp://ftp.ntsg.umt.edu/pub/MODIS/Mirror/) and USGS (ftp://e4ftl01.cr.usgs.gov/MOTA) data repositories.

If the content information from the two storages match the metadata specified by the user, then they are automatically downloaded to the machines that process the Mosaic12 workflow and mapped to the corresponding inputs of the graph.


Specify the workflow inputs

Process data from year 2000

MOD15 product

Bands to be processed:

□ Fpar_1km ☑ FparLai_QC □ FparStdDev

☑ Lai_1km □ FparStdDev_1km □ LaiStdDev_1km

MOD16 product

Bands to be processed:

☑ ET_1km □ PET_1km ☑ ET_QC_1km

□ LE_1km □ PLE_1km

Fig. 6. The inputs specification for the Mosaic12 workflow

D. Optimizing the execution process

When using the workflow for a large time interval (e.g. 10 years) the entire execution process will take a long time to complete. Based on the Grid parallel computing capabilities the GreenLand is able to complete the entire process in approximately two hours.

The MODIS sensor generates data for the same geographic area every 8 days. This means that in a year we have 45 samples for the same tile for each product, and 90 data samples for both MOD15 and MOD16.

This experiment requires that a result is generated for each data sample, meaning that for one processing year the Mosaic12 workflow will generate 90 independent results (if taking into account both MODIS products).

In order to optimize the Grid execution, the gProcess platform partitions the use case into multiple tasks and schedules them to be executed on a different Grid node. Each task contains a group of 9 data samples. Using this approach, we've obtain a parallel execution that significantly improves the total workflow execution time.

E. Results visualization

Another important aspect about the Mosaic workflow is the ability of sharing the results among different scientific communities or between regular users that are not necessarily registered within the GreenLand system.

The implementation of the OCG services proved to be the best solution, regarding the fulfillment of data level interoperability between multiple platforms. The GreenLand offers support for the majority of the OGC products, such as: Web Map Service (WMS) for spatial data visualization, Web Coverage Service (WCS) for remote data retrieval, and Web Processing Service (WPS) useful for standardized execution of the workflows.

The results visualization is managed by the WMS service that provides a standardized method of accessing spatial data, regardless of the location of the remote repository. This service does not offer access to the original information, instead it generates at runtime a graphical representation of the data (under the form of JPEG, TIF, or PNG files). Such a representation is known as layer and can be identified as a frequency band of the satellite images.

The results visualization using the WMS service is an open feature that can be used by any platforms, regardless of its location. The only requirements are the availability of the results (resident on a GeoServer or MapServer) and the Internet connection of all the systems that are implementing the visualization feature.

The WMS service can be accessed directly as a Web based resource (http://<server_domain>/service=WMS&request=GetCapabilities&version=1.1.1) with multiple parameters that specify the results that need to be visualized, the image type that is used for exporting the result (e.g. JPEG, PNG, etc.), the projection type, etc.

Figure 7 highlights the results visualization when using the WMS service from different GIS platforms. As can be

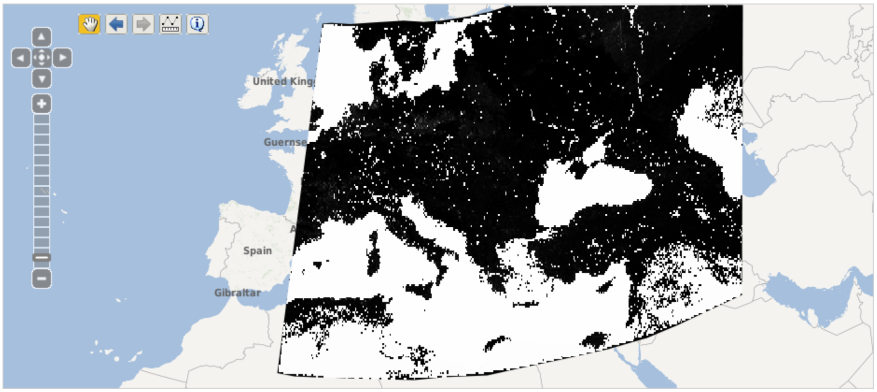


Fig. 7. Mosaic workflow results visualization on different platforms

seen there is the possibility to overlap the WMS result over interactive maps, increasing this way the user satisfaction and the usability of the system.

V. CONCLUSIONS

The natural language description of large use case studies is a complex process that requires a good understanding of the context of the problem. Modeling and implementing software architectures based on these use cases is even harder and usually involves specialists from both computer science and Earth science domains.

This paper describes the GreenLand platform that implements the previous mentioned features and exposed them in a user friendly Web based application. The complexity of the inner mechanisms is hidden from the user. Special interaction techniques were developed in order to ease the use cases description in an interactive and intuitive manner.

The system related architecture highlights all the modules of the GreenLand platform and exemplifies their contributions by modeling the Black Sea catchment scenario as a GreenLand workflow.

OGC standard implementation provides the advantage of achieving data interoperability with other external platforms. This feature is useful especially when retrieving, processing, and visualizing spatial data from different remote repositories.

ACKNOWLEDGMENT

This research is supported by the enviroGRIDS Project funded by the European Commission, through the Contract 226740.

REFERENCES

[1] D. Mihon, V. Colceriu, F. Bektas, K. Allenbach, M. Gvilava, D. Gorgan, ”Spatial Data Exploring by Satellite Image Distributed Processing”, Geospatial Research Abstracts, EGU General Assembly, vol.14, 2012.

[2] enviroGRIDS project,
http://envirogrids.net/

[3] B.F. Bektas, C. Goksel, S. Sozen, K. Allenbach, M. Gvilava, K. Rahman, D. Gorgan, D. Mihon, ”Remote Sensing Services - ESIP Platform and Hot Spot Inventory Case Studies”, enviroGRIDS Deliverable D2.11, 2012. Available at: http://envirogrids.net/index.php?option=com_jdownloads&Itemid=13 &view=finish&cid=139&catid=11

[4] D. Mihon, A. Minculescu, V. Colceriu, D. Gorgan, ”Diagramatic description of distributed spatial data processing”, Romanian Journal of Human-Computer Interaction, pp.59-80, 2012.

[5] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, ”Grid Information Services for Distributed Resource Sharing”, In IEEE International Symposium on High Performance Distributed Computing, (2001), IEEE Press

[6] V. Bacu, T. Stefanut, D. Rodila, D. Gorgan, ”Process Description Graph Composition by gProcess Platform”, 3rd International Workshop on High Performance Grid Middleware, CSCS-17 Conference, vol.2, pp.423-430, 2009.

[7] Open Geospatial Consortium, (2007), ”OpenGIS Web Service Common Implementation Specification”, pp.1-153.

[8] Sextante, ”A Versatile Open-Source Library for Spatial Data Analysis”, 2011.

[9] P. Ramsey, ”User Friendly Desktop Internet GIS (uDIG) for OpenGIS Spatial Data Infrastructures”, Refractions Research Inc., 2004.

[10] M. Landa, ”New GUI for GRASS GIS Based on wxPython”, Departament of Geodesy and Cartography, pp.1-17, 2008.

[11] T. Athan, O. Dassau, A. Ghisla, ”Quantum GIS 1.7.0 Geographic Information System User Guid”, Open Source Geospatial Foundation, 2011.

[12] A. Maier, ”A Job Management and Optimising Tool”, International Conference on Computing in High Energy and Nuclear Physics, Journal of Physics: Conference Series, vol. 119, pp.2-9, 2008.

[13] J.T. Moscicki, ”DIANE Distributed Analysis Environment for Grid-enabled Simulation and Analysis of Physics Data”, Nuclear Science Symposium, vol. 3, pp.1617-1620, 2004.

[14] V. Bacu, ”Error Prevention and Recovery Mechanisms in the ESIP Platform”, IEEE 6th International Conference on Intelligent Computer Communication and Processing, pp.411-417, 2010.

[15] D. Gorgan, V. Bacu, T. Stefanut, D. Rodila, D. Mihon, ”Earth Observation Application Development based on the Grid Oriented ESIP Satellite Image Processing Platform”, Journal of Computer Standards & Interfaces, vol. 34/6, pp. 541548, 2012.