Integration of data and computing infrastructures for Earth Science: an image mosaicking use-case

Mazzetti, Paolo; Roncella, Roberto; Mihon, Danut; Bacu, Victor; Lacroix, Pierre Marcel Anselme; Guigoz, Yaniss; Ray, Nicolas; Giuliani, Gregory; Gorgan, Dorian; Nativi, Stefano

How to cite

MAZZETTI, Paolo et al. Integration of data and computing infrastructures for Earth Science: an image mosaicking use-case. In: Earth science informatics, 2016, vol. 9, n° 3, p. 325–342. doi: 10.1007/s12145-016-0255-5

This publication URL: https://archive-ouverte.unige.ch//unige:83032

Publication DOI: 10.1007/s12145-016-0255-5

Integration of data and computing infrastructures for Earth Science: an image mosaicking use-case

Paolo Mazzetti[1], Roberto Roncella[1], Danut Mihon[2], Victor Bacu[2], Pierre Lacroix[3,4], Yaniss Guigoz[3,4], Nicolas Ray[3,4], Gregory Giuliani[3,4], Dorian Gorgan[2], Stefano Nativi[1]

1. National Research Council of Italy, Institute of Atmospheric Pollution Research (CNR-IIA), Division of Florence – Via Madonna del Piano, 10 - 50019 Sesto Fiorentino (FI) – Italy

2. Technical University of Cluj-Napoca, Computer Science Department, Str. G. Baritiu 28, 400027 Cluj-Napoca, Romania.

3. University of Geneva, Institute for Environmental Sciences, enviroSPACE Lab., 66 Bd Carl-Vogt, CH-1205 Geneva, Switzerland.

4. Global Resource Information Database – Geneva, International Environment House, 11 chemin des Anémones, CH-1219 Châtelaine, Switzerland

Paolo Mazzetti: *paolo.mazzetti@cnr.it*

Roberto Roncella: *r.roncella@iia.cnr.it*

Danut Mihon: *vasile.mihon@cs.utcluj.ro*

Victor Bacu: *victor.bacu@cs.utcluj.ro*

Pierre Lacroix: *pierre.lacroix@unige.ch*

Yaniss Guigoz: *yguigoz@unepgrid.ch*

Nicolas.Ray: *Nicolas.Ray@unige.ch*

Gregory Giuliani: *gregory.giuliani@unepgrid.ch*

Dorian Gorgan: *dorian.gorgan@cs.utcluj.ro*

Stefano Nativi: *stefano.nativi@cnr.it*

Abstract

In the recent years several solutions have been proposed to implement digital infrastructures for sharing and processing scientific data and observations. Spatial data infrastructures currently enable effective and efficient geo-information data sharing in many disciplinary communities, and innovative solutions are under development to support new open data and linked data paradigms. In parallel, High Performance Computing systems, computing grids and more recently cloud services, enable fast processing of big data. However, the integration of data and computing e-infrastructures is a raising issue in multidisciplinary research. In the context of the Global Earth Observation System of Systems (GEOSS) initiative, an innovative approach has been proposed. Taking into account that the heterogeneity of data and computing e-infrastructures and related technologies cannot be reduced beyond a certain extent, since it is due to the need of supporting use cases and scenarios from different scientific communities, a brokering solution has been designed and developed. A Business Process Broker (BPB) is a component which takes a formal description of a scientific business process, and translates it in an executable process which can be run on multiple and remote processing and workflow services. In doing this it solves all the interoperability issues in a (semi-)automated way. It allows lowering the entry barrier for both computing service providers - who do not have to comply with strict specifications - and users - who can work at higher level of process description. The paper presents and discusses a BPB use-case from the European project IASON, implementing an Earth Observation application involving satellite image mosaicking, HPC computing services and spatial data e-infrastructures.

Keywords

Brokered architectures, HPC, GEOSS, e-infrastructures, Business Process Broker, Earth Science

Introduction

In the recent years several solutions have been proposed to implement digital infrastructures for sharing and processing of scientific data and observations. Federated systems based either on the implementation of common specifications from standardization bodies and working groups - e.g. ISO[1], Open Geospatial Consortium (OGC)[2], or Taxonomic Database Working Group (TDWG)[3] in the biodiversity community - or on common tools – such as the widespread THREDDS Data Server (TDS)[4] in the meteo-ocean community, or the Geodetic Seamless Archive Centers (GSAC)[5] in geodesy domain – have been developed to implement Spatial Data Infrastructures for international initiatives or to serve communities of practice. Brokered architectures have been adopted for implementing system of systems integrating existing infrastructures for multidisciplinary applications, like the Global Earth Observation System of Systems (GEOSS) developed by the Group on Earth Observations (GEO) (Nativi, Craglia, & Pearlman, Earth Science Infrastructures Interoperability: The Brokering Approach, 2013) (Nativi, Mazzetti, Craglia, & Pirrone, 2014). Innovative solutions are under development to support new open data and linked data paradigms, opening data sharing to information collected from crowdsourcing or social media mining (Edwards, 2013) (Diaz, Granell, Huerta, & Gould, 2012).

In parallel to this effort on data sharing, High Performance Computing systems, computing grids and more recently cloud services enable fast processing of big data, that is data characterized by big volume, large variety and need of high velocity (Giuliani, Ray, & Lehmann, Grid-enabled Spatial Data Infrastructure for environmental sciences: Challenges and opportunities, 2011) (Giuliani, Nativi, Lehmann, & Ray, 2012) (Lecca, et al., 2011) (Petitdidier, et al., 2009) (Mazzetti, Nativi, Angelini, Verlato, & Fiorucci, 2009). The increasing amount of geospatial data coming from new Earth Observation (EO) instruments with high spatial, temporal and radiometric resolution (such as the new ESA Sentinel missions), and new ways to collect in-situ data such as the mentioned crowdsourcing or social media mining, raises the need of integrating data sharing and computing infrastructures (ESA, 2013). This is recognized as a requirement in the major initiatives on EO and research infrastructures. For example, the Belmont Forum E-Infrastructures and Data Management Collaborative Research Action recognizes "the need to integrate high performance computing and data analysis, extensive data storage" in its (draft) Strategy and Implementation Plan (Belmont Forum E-Infrastructures and Data Management Steering Committee, 2015), and also engaging the "Data /Computation Interface" and "Harmonisation of global data infrastructure" working groups in joint activities. Moreover, several projects on e-Infrastructures are addressing such relevant issue, such as MEDINA[6], SeaDataNet[7], ODIP[8], GEOWOW[9], EUBON[10], ECOPOTENTIAL, MED-SUV[11] funded under the European Union funding programmes.

Besides the technical issues on the integration of data and computing infrastructures, a usability aspect arises. Research e-Infrastructures end-users – i.e. mainly scientists, policy-makers and decision-makers- need to interact with the system at a high semantic level where only relevant concepts (business processes, models, indicators, etc.) are exposed, without any technical detail that is not necessary (for them), such as job workflows, data formats, service interfaces, etc. In the context of GEOSS related projects, an innovative approach on that direction has been proposed. It is based on the Business Process Broker (BPB), a component which takes a formal description of a scientific business process, and translates it in an executable process which can be run on multiple and remote processing and workflow services. In doing this

[1] http://www.iso.org
[2] http://www.opengeospatial.org
[3] http://www.tdwg.org
[4] https://www.unidata.ucar.edu/software/thredds/current/tds/
[5] https://www.unavco.org/software/data-management/gsac/gsac.html
[6] http://www.medinaproject.eu/puplic/home.php
[7] http://www.seadatanet.org/
[8] http://www.odip.org/
[9] http://www.geowow.eu/
[10] http://www.eubon.eu/
[11] http://med-suv.eu/

it solves all the interoperability issues in a (semi-) automated way. It allows lowering the entry barrier for both computing service providers - who do not have to comply with strict specifications - and users - who can work at higher level of process description.

In the context of the EU/FP7 IASON project (IASON Consortium, 2015) a use-case on image mosaicking has been developed using the BPB. It is aconceived as a proof-of-concept of integration of data sharing infrastructures based on OGC standards, brokered architectures and grid computing services.

Scenario description

IASON (Fostering sustainability and uptake of research results through Networking activities in Black Sea & Mediterranean areas) was a two year Coordination and Support Action ended in 2015 and funded by the European Union under the European Community's Seventh Framework Programme (FP7). It was coordinated by the Aristotle University of Thessaloniki in Greece, with the participation of European research centers and private companies and including international partners from the North Africa and Black Sea regions. The project objective was the establishment of a permanent and sustainable network of scientific and non-scientific institutions, stakeholders and private sector enterprises belonging to the EU and third countries located in two significant areas: the Mediterranean and the Black Sea regions. The main focal point of the project was the usage and application of Earth Observation (EO) in the following topics: climate change, resource efficiency and raw materials management.

As part of a set of actions specifically dedicated to the uptake of Earth Observation applications resulting from past projects, a specific task was dedicated to demonstrate an effective re-use of EO applications and tools for the Black Sea and Mediterranean regions according to the GEO approach.

In the course of the enviroGRIDS (Building Capacity for a Black Sea Catchment Observation and Assessment System supporting Sustainable Development; 2009-2013) project, the GreenLand platform (Mihon, Colceriu, Bacu, & Gorgan, 2013) (Gorgan, et al., 2013), a grid-based infrastructure, was developed and improved (Gorgan, et al., 2011) (Balcik, et al., 2013). The GreenLand platform provides GIS services related to spatial data management, processing, analysis, and visualization. In general, this framework addresses large volume of data processing by using the Grid infrastructure computational resources. Due to their physical distribution, these resources provide faster execution times than a regular machine. The basic operators and the complex workflows represent the main theoretical concepts that GreenLand is built upon. The basic operator is used as a virtual container that encapsulates relatively simple algorithms that cannot be divided into sub-problems. Actually they exist as atomic structures. The concept is similar to the functions (or methods) from computer software domain. Usually a function is used to implement one simple feature (just like the basic operator), while the entire software application is a combination of functions that are linked in a specific order (just like the workflow concept).

The Normalized Difference Vegetation Index (NDVI) is an example of a basic operator, since it is simple enough to be implemented as one routine, without the need to divide it into multiple functions and then reference them together. On the other hand a more complex scenario should be implemented as a set of different operators, each of them storing one simple feature.

The workflows are useful for representing large use case scenarios, where the entire problem can be decomposed within simpler algorithms that are connected together by specific relations and constraints. This way, the GreenLand platform provides algorithms for vegetation index computation, arithmetical formulas, pseudo-coloring of satellite images, etc. Initially, all these functionalities were implemented to be interoperable with other software platforms, based on REST paradigm. With the popularity growth of OGC standard, the GreenLand started to extend through this direction. One important step was the development of the Normalized Difference Vegetation Index (NDVI) as an OGC WPS service, followed by other basic operators. This way, the GreenLand framework can now share data and algorithms with other GIS related platforms that implement the OGC standards.

IASON aimed to uptake GreenLand, integrating it in a more flexible infrastructure assuring interoperability with other data infrastructures, and facilitating access by scientists and decision-makers.

A specific use-case on EO imagery mosaicking was selected for several reasons: a) it is a common processing facilitating the general re-use of EO from disparate sources; b) it involves basic steps of data retrieval, processing, publishing and visualization; c) an implementation of basic building blocks was already available.

There are plenty of cases when a satellite image does not cover the entire area of interest, and it is necessary to put together two or more such layers. The main goal of this use case scenario is to tile together a set of satellite images from different sources (e.g., Landsat, MODIS, and possibly others) and make the result accessible and visible through a common platform such as Google Earth (Google Inc., 2015) in order to improve the analysis process for the end-users that are interested in studying the Earth Observation phenomena.

The scenario includes the following phases:

1. Data retrieval consisting in the discovery of and access to the relevant datasets
2. Mosaicking process consisting in the merging of the retrieved datasets
3. Publishing, consisting in the publication of the result for visualization

The objective of the IASON scenario was the uptake of this mosaicking procedure and its provision through a brokered architecture decoupling services and their implementation. It would demonstrate:

- The possibility to use different data sources for the mosaicking procedure, for example merging datasets available for the Black Sea or Mediterranean regions;
- The possibility to use different mosaicking procedures provided on different infrastructures (e.g. for comparison) or, on the other hand, make the GreenLand based services available to different process;
- The possibility to use different workflow engines for running the same process.

Architectural approach

The brokered architecture for a Model Web

The access to and integration of scientific models is a complex task with a long history and many different proposed approaches and solutions - for a short overview see (Nativi, Mazzetti, & Geller, Environmental model access and interoperability: The GEO Model Web initiative, 2013). Paralleling model interoperability with document interoperability as it is achieved in the World Wide Web, Nativi, Mazzetti and Geller (2013), identified four principles for a Model Web:

P1. **Open access**: In a Model Web, anybody can create a service to share a model – it becomes simply another resource accessible via the Web – and anybody (or any machine) can access it.

P2. **Minimal barriers to entry**: The Model Web seeks to minimize the entry barriers of both resource providers (modelers who share their model via Web services) and users (other modelers who desire input for their model, or end users on a website).

P3. **Service-driven approach**: Model access is provided by services (i.e. Web services), making a Model Web a subset of a general-purpose distributed services framework (i.e. the WWW) and Model Web resources are a specialization of generic distributed resources (i.e. WWW resources).

P4. **Scalability**: The design of the Web makes it completely scalable, a critical factor to its explosive growth. Scalability is important to a Model Web and also inherent in the concept because it is based on Web services.

As a vision, a Model Web can be implemented according to different architectures, based on different architectural styles. As far as an architecture is compliant with the principles above, it is a realization of the Model Web.

The Group on Earth Observations (GEO) recognized the importance of model sharing in both the Work Programme 2009-2011 and the Work Programme 2012-2015 for building the Global Observation System of Systems (GEOSS). They dedicate specific activities to the Model Web development and in general "to develop innovative methods for harmonized access and use of heterogeneous data, services, and models" (GEO Secretariat, 2014).

In the context of such GEO activities, CNR-IIA designed and developed an architecture for the Model Web based on the broker pattern (Chang, 2005). It assumes that a set of models is accessible in a way that it is compliant with principles P1 and P3 above, i.e. "open access" and "service-driven approach". This means that modelers provide open access to models, and that they are published as services using web technologies. Those assumptions seem reasonable, especially in the GEO context where open sharing, at least for data, is a priority, and regulated by clear data sharing principles. In particular, in the current IT scenario dominated by the Web, the adoption of web technologies is a very loose constraint.

It is noteworthy that although those assumptions sound reasonable, it does not imply that actions on: a) fostering open data and model sharing, and b) capacity building to facilitate model publication in the Web are not needed. However, they are out of the scope of the Model Web architectural design activity.

Instead, the brokered Model Web architecture specifically addresses the principle P2: *minimal barriers to entry*. Assuming that models are published as web services does not mean actually almost anything. Indeed, a web service invocation may be as simple as a HTTP request/response interaction or as complex as a SOAP workflow. This is the reason why we said that the use of web technologies is a very loose requirement, easily fulfilled. The problem is how to deal with the possible heterogeneity of a web service ecosystem.

In building a distributed system, it is possible to adopt two different approaches to integrate heterogeneous nodes. They can be summarized as *federated* and *brokered* approaches.

In the *federated approach*, a common set of specifications (e.g., *federated model*) is agreed between the participants. In a federated model web, participants would agree on common specifications that can be as loose as a set of communication protocols (including service interfaces, metadata and data models) or as strict as the use of common software platforms and tools. This is the approach adopted by most of the existing solutions for model integration and workflow.

In the *brokered approach* (Nativi, Craglia, & Pearlman, Earth Science Infrastructures Interoperability: The Brokering Approach, 2013)(Giuliani, Ray, & Lehmann, Grid-enabled Spatial Data Infrastructure for environmental sciences: Challenges and opportunities, 2011), no common model is defined, and participating systems can adopt or maintain their preferred interfaces, metadata and data models. Specific components (the *brokers*) are in charge of accessing the participant systems, providing all the required mediation and harmonization functionalities. The only interoperability agreement is the availability of documentation describing the published interfaces, metadata and data models. In a brokered model web, modelers can publish their models, as they prefer, just providing the needed documentation to make brokering possible. This approach greatly lowers the entry barriers for model providers helping to comply with the *minimal barrier to entry* principle. Brokered architectures have also another advantage: introducing a middle-tier (brokers) between producers (servers) and consumers (clients), they can provide added value. Brokers, beside mediation and harmonization, can enrich the interaction introducing advanced functionalities including transformations, fusion, semantics, etc.

Obviously, with the brokered architecture approach, the overall complexity of model sharing is not removed, it is just shifted from model servers, which do not need to publish services complying with a federated model, to the broker(s) which must be able to interact with many heterogeneous systems. However, this approach is effective since it allows applying the general *separation-of-concern* engineering pattern. Model providers can focus on model logic, while interoperability experts can focus on the broker(s) design and development.

The Business Process Broker

In the context of European Union funded projects, a model broker prototype, named Business Process Broker (BPB), has been developed and then tested in different use-cases (Bigagli, Santoro, Mazzetti, & Nativi, 2015). In particular: a) in FP7 UncertWeb[12] (Uncertainty Model Web) it was used to compose scientific models supporting uncertainty representation and transformations; b) in FP7 MEDINA[13] (Marine Ecosystem Dynamics and Indicators for North Africa) it was used to expose a Seagrasses Habitat Suitability Model for the Mediterranean area.

As its name implies, a Business Process Broker aims to work at a higher semantic level in respect to scientific models. It aims to work at the level of scientific business processes. In a general sense, a scientific business process is the process to be followed for answering a scientific challenge. It includes the application of scientific models to generate data, information and knowledge. A scientific business process can be represented by an abstract business process detailing the needed interaction between data and models involved (see **Fig. 1** for an example).

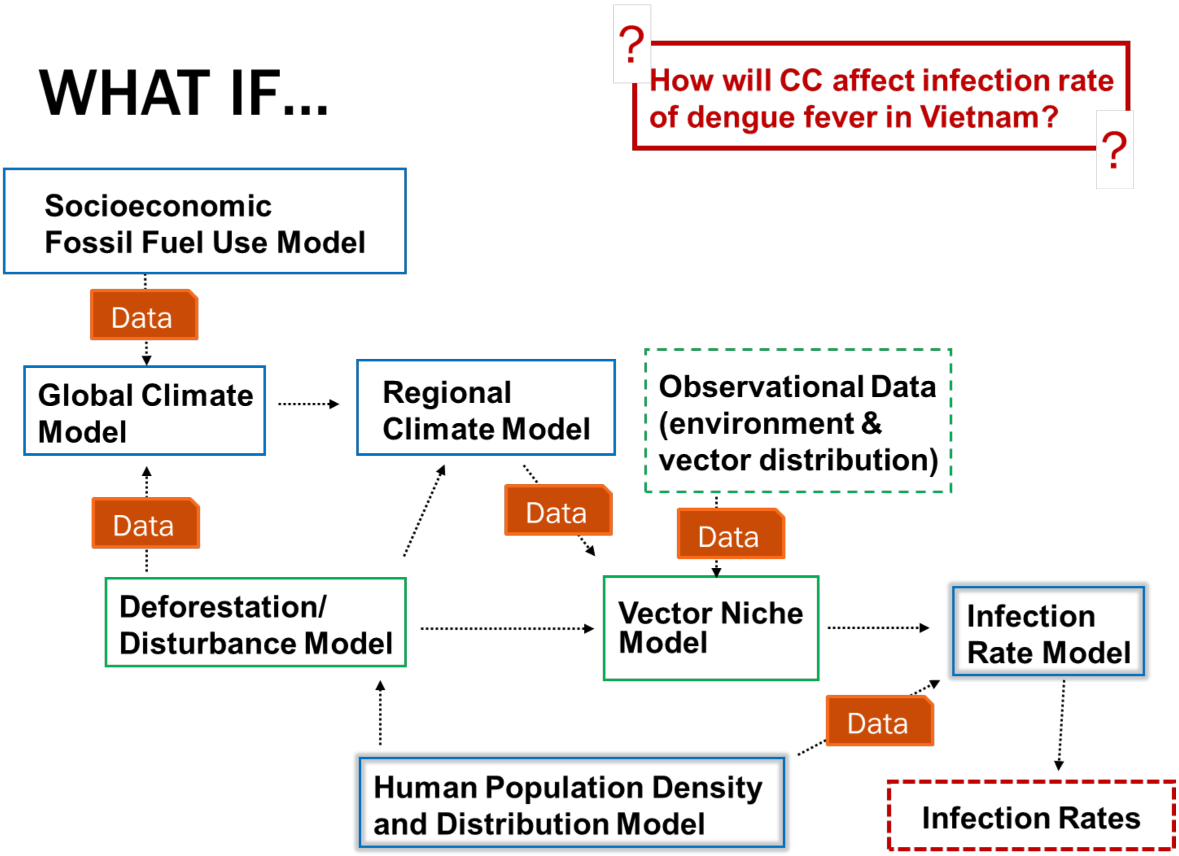

WHAT IF...
?
How will CC affect infection rate of dengue fever in Vietnam?
?
Socioeconomic Fossil Fuel Use Model
Data
Global Climate Model
Regional Climate Model
Observational Data (environment & vector distribution)
Data
Data
Data
Deforestation/ Disturbance Model
Vector Niche Model
Infection Rate Model
Data
Human Population Density and Distribution Model
Infection Rates

Fig. 1 Example of an abstract diagram for answering to a "what if" scientific challenge

It is noteworthy that the abstract diagram is the scientist's view of the problem, and it does not include technical details. An abstract business process can be encoded using high-level notations like the Business Process Modeling Notation (BPMN) (Object Management Group, 2015). Since it does not include technical details, the abstract diagram needs to be translated in a concrete workflow and encoded in an executable language for running. Since several workflow engines exist, a broker should be able to translate (compile) the abstract business process in an executable language, adapting interfaces and data models, and run the execution in the most suitable workflow engine.

Following this approach, the BPB prototype is able to:

[12] http://www.uncertweb.org
[13] http://www.medinaproject.eu/

1. accept abstract business process encoded in BPMN enriched with annotations facilitating the execution and expressed according to a set of defined conventions
2. compile the BPMN in a concrete workflow introducing required components that do not appear in the abstract business process (e.g. data format transformation services)
3. encode the concrete workflow in one of the supported execution language (only Business Process Execution Language (BPEL) is currently implemented)
4. run the execution language in one of the local or remote available workflow engines (only JBoss jBPM[14] open source software currently supported)

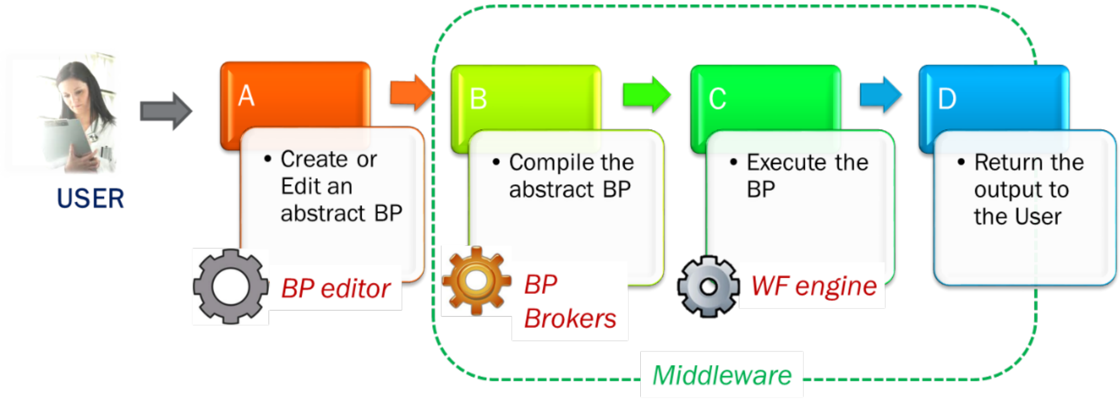

USER
A
• Create or Edit an abstract BP
BP editor
B
• Compile the abstract BP
BP Brokers
C
• Execute the BP
WF engine
D
• Return the output to the User
Middleware

Fig. 2 *Implementing model sharing use-cases in a brokered architecture for the Model Web*

The BPB is conceived as a component of a more general architecture supporting model sharing and execution (see **Fig. 2**). A BP Editor enables users to create abstract diagrams of scientific business processes. The prototype uses the Yaoqiang BPMN Editor[15] for BPMN creation, and includes a dedicated Graphical User Interface (GUI) for exploring available abstract business process, setting up input data and parameters, monitor business process execution and finally retrieve the results. The abstract business process, e.g. a BPMN document, is the input of the BPB that compiles it in an executable business process. A semi-automated procedure would introduce other technical components, possibly asking the user for further missing information needed in the compilation phase. The outcome of this phase is an artifact representing the concrete business process, e.g. a BPEL document or any other supported execution language such as Grid Execution language or Taverna Flow Language. It is passed to a suitable external workflow engine which runs it. The prototype currently supports BPEL and JBoss jBPM workflow engine. The data results are then provided back to the user.

Allowing working at the abstract diagram level, the BPB complies with principle P2 *minimal barriers to entry* concerning users. So, in the end, the BPB lowers the entry barrier for both providers - by brokering heterogeneous model services without need of changes on the server side – and users – by allowing them to work at the abstract diagram level without dealing with technical interoperability aspects.

The brokered Model Web aims to comply with the fourth principle – i.e. *scalability* – along the main axes: *variety* (i.e. scalability in terms of increasing variety of model and model workflow services), *volume* (e.g., amount and complexity of scientific business processes), and *velocity* (e.g., time of response). The brokered approach, with its focus on mediation and harmonization, specifically targets variety. In comparison with federated approach, in a brokered approach there is not any pre-defined federated model, which can be outdated by some new model service. The BPB is specifically designed as a modular framework, which can be extended to support new model services and workflow engines.

[14] http://www.jbpm.org/
[15] http://bpmn.sourceforge.net/

One of the typical concerns about brokered architectures is the introduction of a middle tier made of components – i.e. the brokers - that stay between clients and servers. The brokers can become a single point-of-failure or a bottleneck in a distributed architecture. However, it is noteworthy that brokers are the logical center of a star topology (with data sources and users as end nodes), but they can be physically replicated. What is a single broker by a logical point-of-view, could be actually implemented with many redundant components, implementing load-balancing and high-availability functionalities. Therefore a brokered architecture can effectively exploit many scalability solutions including elastic computing to address volume and velocity scalability. Moreover, a brokered Model Web can exploit the capabilities of external model services and workflow engines choosing the "best" available solution between those available for running a specific concrete workflow. The BPB prototype is specifically designed for deployment either on local infrastructures or cloud environments and to broker different model servers and workflow engines.

Inside the Business Process Broker

Fig. 3 shows the internal architecture of the BPB highlighting the main internal components and the connections to external systems.

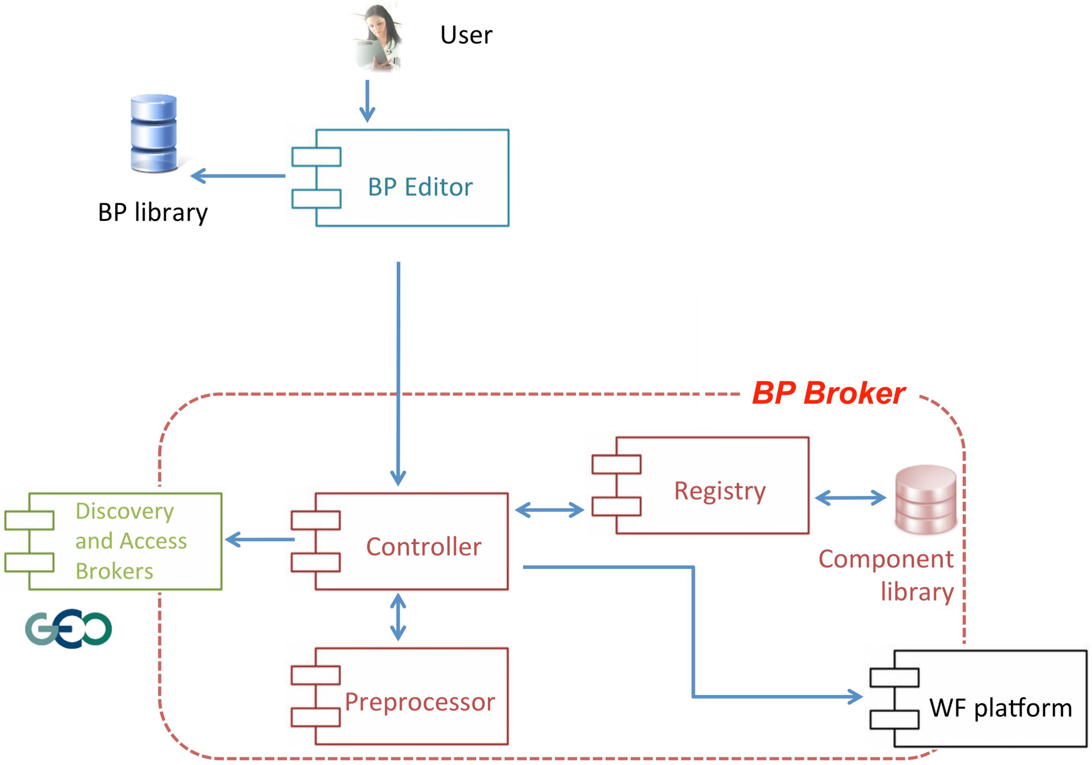


Fig. 3 Component view of the BP Broker

A short description of the main external components follows:

- BP Editor: allows to create and edit Business Processes publishing BPMN artifacts compliant with OMG specifications (BPMN 2.0). In the prototype we used the Yaoqiang BPMN Editor.
- Discovery and Access Brokers: provide seamless discovery of and access to heterogeneous data sources (Nativi & Bigagli, Discovery, Mediation, and Access Services for Earth Observation Data, 2009). In the prototype we adopted the GI-suite Brokering Framework (ESSI-Lab, 2014), also used to implement the GEO DAB component of the GEOSS Common Infrastructure (GEO, 2014).

- WorkFlow (WF) platform: supports the execution of an executable BP. In the prototype the jBPM 5.4 instance is adopted.

The BP broker is in charge of transforming the user-defined abstract BP into executable BP. It has three main internal components:

- *Controller* : makes use of a set of modules to implement specific adaptations according to the needed component to invoke;
- *Preprocessor* : is the core module of the BP broker and it implements the actual transformation from an abstract BP to an executable BP, executing a set of *dry runs* to simulate the execution of the BP and identify possible I/O mismatches;
- *Registry* : provides registry functionalities of available components (models and re-usable BP).

Use-case development

Creation of the BPMN

The BPMN diagram for the IASON use-case scenario is illustrated in **Fig. 4** according to the BPMN specification and the BP broker conventions defined in UncertWeb project. More details on the conventions are available in (Bigagli, Santoro, Angelini, Mazzetti, & Nativi, 2011).

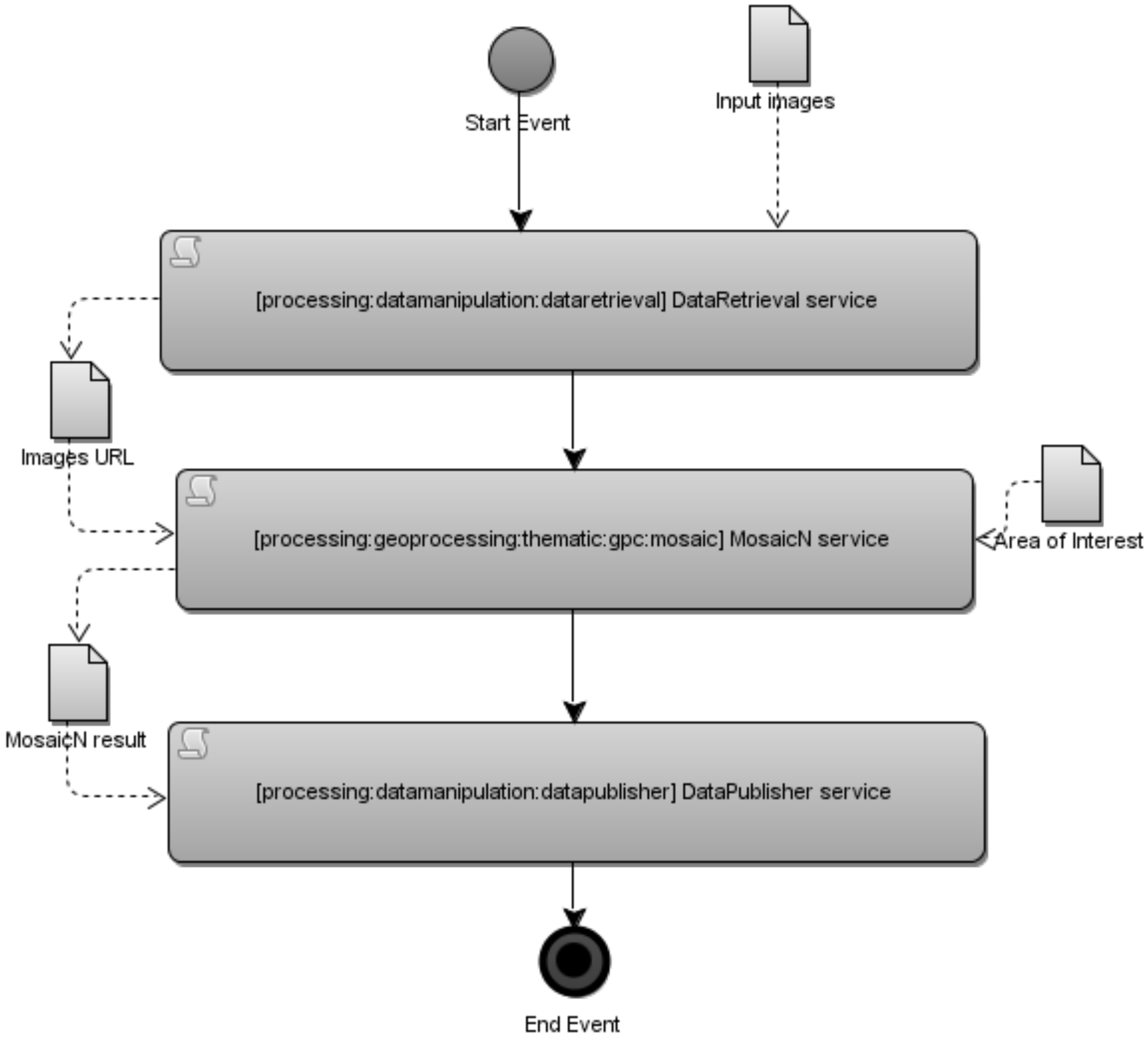

Start Event
Input images
[processing:datamanipulation:dataretrieval] DataRetrieval service
Images URL
[processing:geoprocessing:thematic:gpc:mosaic] MosaicN service
Area of Interest
MosaicN result
[processing:datamanipulation:datapublisher] DataPublisher service
End Event

Fig. 4 *BPMN Diagram for the use-case*

Every rectangular block in the diagram is a BPMN Script Task and is implemented by a model or a service. Inputs and outputs are graphically represented by the BPMN Data Object elements (the document icons, e.g. Area of Interest).

Inputs for the scenario are:

- Input images: URL to an XML file that contains a list of satellite images.
- Area of Interest: a bounding box describing the Area of Interest.

Output for the scenario is:

- Satellite image: representing the mosaicked image. It is expressed in three different formats: GeoTIFF, PNG and KML.

The following main components (adapters) appear in the diagram:

- *DataRetrieval service*: extract the input images links from an XML file and return an URL to a list of N GeoTIFF images.
- *MosaicN service*: mosaic a list of N GeoTIFF images (from Images URL) into a single satellite image that covers the area of interest.
- *DataPublisher service*: publish the data result into GeoServer and expose the output in three different formats: GeoTIFF, PNG and KML.

Implementation of BPMN Script Tasks

In the IASON scenario the required BPMN script tasks are implemented as OGC WPS services.

To implement the use case scenario described in this paper, we developed three new WPS processes: *Data retrieval*, *MosaicN*, and *GeoServer publisher*. Their implementation particularities are presented in the following sections.

Data retrieval service

The IASON use case scenario described within this paper handles satellite images. Before stepping into the actual data processing, we have to assure that there is relevant data to be processed. The Data retrieval service aims at transferring such information from GIS remote repositories to the GreenLand local storage using OGC standards.

The service implementation is based on the PyWPS programming language allowing an easy integration of OGC WPS API into Python scripts (Cepicky, 2007). Without giving full software implementation details, we only highlight here how the OGC requirements can be integrated within the PyWPS language.

The implementation process starts by defining a DataRetrieval.py file and registering this operator inside `__init__.py`, an internal PyWPS configuration file. The actual implementation can be divided into three sections:

1. **Service overview data**
   Contains specific information (e.g. name, internal identifier, etc.) that differs from one service to another. For the DataRetrieval operator such a structure is similar to:

```
WPSProcess. init (self,
   identifier = ”DataRetrieval”,
   title = ”Data retrieval process”,
   abstract = ”Useful for transferring satellite images”,
   version = ”1.0”)
```

2. **Inputs and output specification**

There are cases when the WPS services have specific inputs to be considered within their implementation. Such an example is the DataRetrieval that transfers all data from a specific URL address into the GreenLand local storage. This operator expects as input an XML file that stores the physical path for all satellite images. In the PyWPS implementation this input is defined as:

```
self.URLPath = self.addComplexInput(
    identifier = ”imageList”,
    formats = [{‘mimeType’:’text/xml’}],
    title = ”XML encoded list of satellite images”)
```

The DataRetrieval output is defined in the same way, and contains the URL path to the new created folder that stores all satellite images described within the XML input:

```
self.Result = self.addLiteralOutput(
    identifier = ”result”,
    type = type(“”),
   title = ”Path to the GreenLand local folder”)
```

3. **Processing implementation**
   Contains the actual implementation of the DataRetrieval service. It starts by parsing the XML file (received as input), creating the GreenLand related folder, and then transfering all satellite images from the GIS remote repository. All these features are developed by using Python specific commands and libraries.
   The DataRetrieval output should reflect the GreenLand folder, created earlier. This correspondence is performed by using:

```
Self.Result.setValue(http://<GreenLand_server_address> + uniqueName + “/”)
```

Since DataRetrieval is defined as a PyWPS service and it implements the OGC requirements, it can be used as a WPS operator. This means that operations like GetCapabilties, DescribeProcess, and Execute are allowed for listing all OGC services, displaying detailed information about DataRetrieval, and processing this operator with specific XML inputs.

MosaicN service

The *MosaicN* operator takes *N* satellite image inputs and merges them together in order to generate a single satellite image whose extent is the ensemble of input images. It is based on the classical Mosaic algorithm that handles only two images at a time. Progressively, the inner levels of the operator are linked to the outputs generated earlier, and so on until the final result is computed (**Fig. 5**).

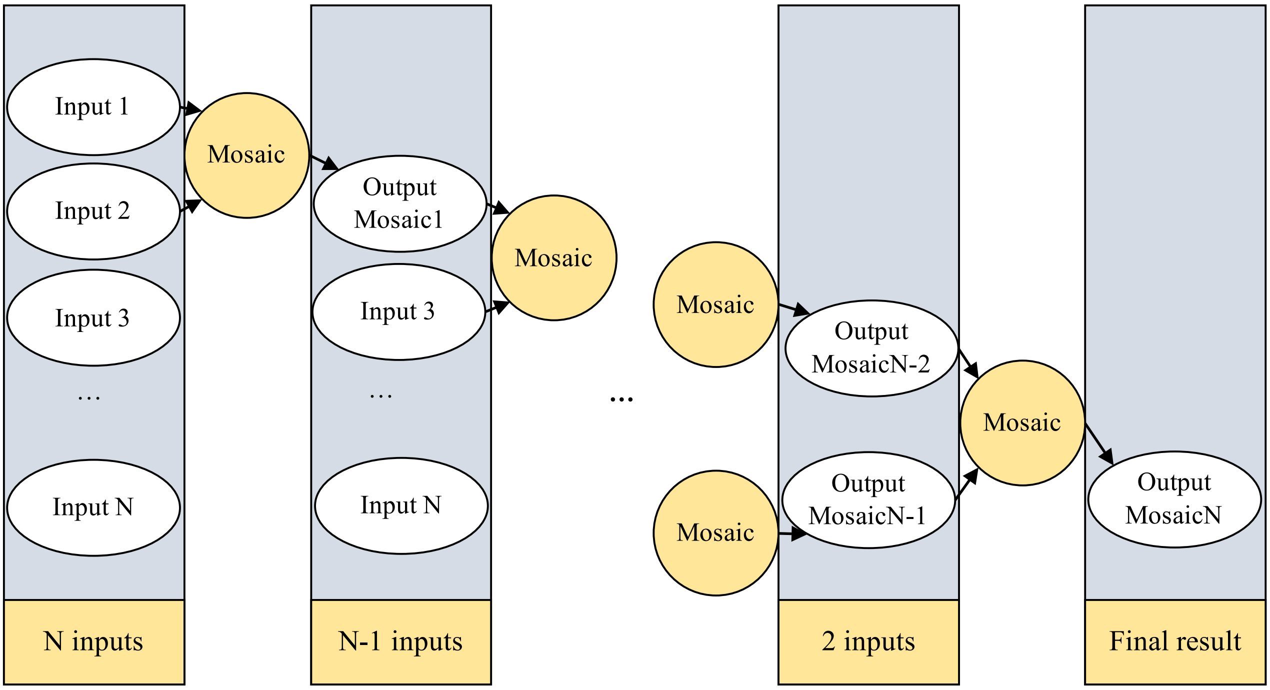


Fig. 5 *Inner functionality of the MosaicN operator*

Since the order in which the images are tiled together does not alter the overall result, but only the computation performance, the *MosaicN* operator begins by grouping the first two inputs. The result is used as first input for the next instance of the algorithm, while the second input is represented by the satellite image called *Input 3*. The process continues iteratively, until there are no more inputs to process.

The *MosaicN* implementation follows the same OGC pattern as the data retrieval operator, meaning that it can be listed together with all other services (using GetCapabilities), it can be queried for more details about its development process (using DescribeProcess), and it also supports URL based remote executions through the Execute WPS feature.

The *MosaicN* operator expects two inputs. The first one is the URL path of the repository that stores all satellite images that need to be tiled together. These inputs are generated on demand by the data retrieval service that creates a folder with a unique name and store inside all data. Once the mosaicking process finishes, this folder (with all inner information) is deleted, since we only need the final processing output.

The core functionality of the *MosaicN* service is to generate one large satellite image that includes, in terms of geographical position, all the above inputs layers. To make it more flexible and suitable for user demands, this main functionality was extended with a new feature: the ability to reduce the area of interest to a particular geographical region. This is possible by cropping the mosaicked satellite image against a virtual bounding box defined by the latitude and longitude coordinates of the top left and bottom right corners. The second *MosaicN* input therefore represents the geographical coordinates of this rectangular area.

Fig. 6a illustrates the *MosaicN* service with four input images. **Fig. 6c** highlights the area of interest obtained by applying the cropping algorithm, based on the latitude and longitude values specified for the bold bounding box. This area represents the final result of the *MosaicN* operator and all further use case processing will be performed on this satellite image.

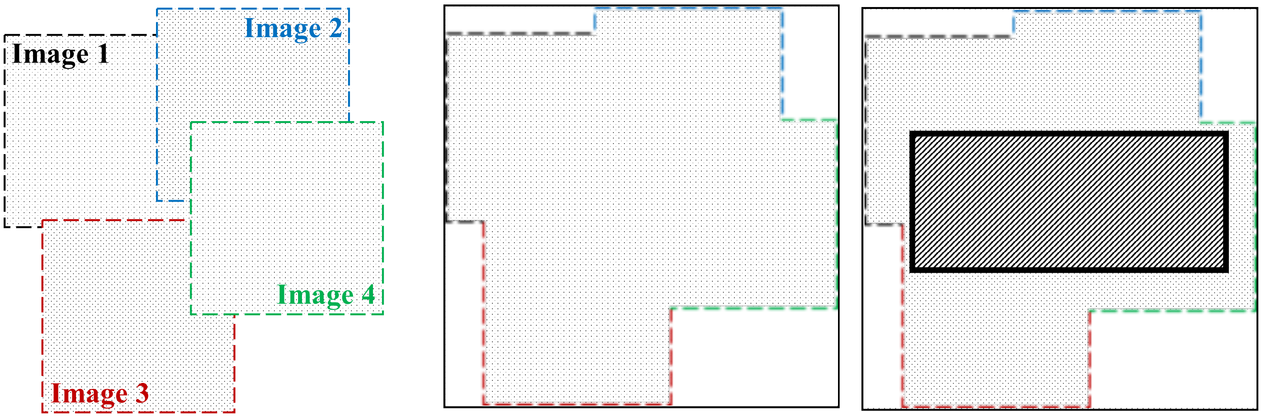


Fig. 6a Input images data set *Fig. 6b Tiled images* *Fig. 6c MosaicN result*

***Fig. 6** MosaicN operator applied on four input images*

The final output of the *MosaicN* WPS service is represented as a satellite image and stored within the GreenLand repository, where this service is also resident. From here it is HTTP accessible, downloadable to the end-user machines, or included within specialized geospatial platforms (e.g. GeoServer) for further processing.

GeoServer publisher service

GeoServer (GeoServer, 2010) is an OGC Web accessible platform that handles geo-spatial data of various formats and types and makes it available as an interoperable resource. GeoServer publisher service is implemented by the same principals as the previous two WPS services. The main goal is to insert the *MosaicN* result into the GeoServer tool that in its turn is able to convert it into several formats (e.g. GeoTIFF, JPEG, KML, etc.) depending on the requirements of the users. The implementation of this feature is possible due to the REST API provided by the GeoServer instance. This service is useful for the last step of the use case scenario, described within this paper.

The integration with the GeoServer is a four steps process, and it was performed by using the Python programming language:

1. **Connecting to the GeoServer instance**
   It is possible through the *geoserver* library that was specifically written for the Python language. The connection method to the GeoServer catalogs is highlighted below:

   ```
   cat = Catalog(http://<geoserver_address>/geoserver/rest, <username>, <password>)
   ```

2. **Access the corresponding workspace**
   Inside GeoServer, data is organized in workspaces that are useful for grouping similar information, usually related to a specific project. This way, a new workspace (called *IASON_Workspace*) was manually created for the use case described within this paper. Before storing the *MosaicN* result inside this workspace, we need to get its reference. This is possible through the following Python command:

   ```
   workspace = cat.get_workspace(“IASON_Workspace”)
   ```

3. **Adding the *MosaicN* result**
   It requires the full system path of the resource that is going to be uploaded to GeoServer. Since we already have the references of the catalog and the workspace, the following instruction is used to insert the result in the GeoServer instance:

cat.create_coveragestore_external_geotiff(<unique_name>, "file://" + <mosaicN_result_path>, workspace, True)

A coverage store is a container that stores a reference to a raster resource (a GeoTiff image in this case) and it is associated with the *IASON_Workspace*.

4. **Access the GeoServer resource**
   At this point, the *MosaicN* result is available in different formats (e.g. JPEG, PNG, GeoTiff, etc.) and can be accessed by using the OGC related services.

Implementation of the BPB adapters

BPB adapters are implemented in Java programming language. They can be used to implement standard OGC services, such as WPS, WCS and utility functions for data manipulation or data transformation.

BPB adapters are categorized according to their functionalities. Four main categories are defined:

- **Access**: adapters are in charge of executing data access operations;
- **Publish**: adapters are able to publishing one or more dataset on standard access services;
- **Processing**: adapters provide all functionalities that execute some kind of processing;
- **Utils**: adapters provide some utility functions (e.g. matrix inversion, format conversion)

Three BPB adapters are developed for the image mosaicking use-case (one for each WPS module): Data retrieval, MosaicN and Data publisher. These BPB adapters interact with the WPS modules through WPS Execute request operations.

Data retrieval adapter

Data retrieval adapter (**Fig. 7**) allows retrieving a list of satellite images stored in an XML file and sending them to Data retrieval module through a WPS Execute request. The WPS service response is an URL path to a folder where the images, included in the XML file, are stored; the result is passed on to the next adapter, namely *MosaicN* adapter.

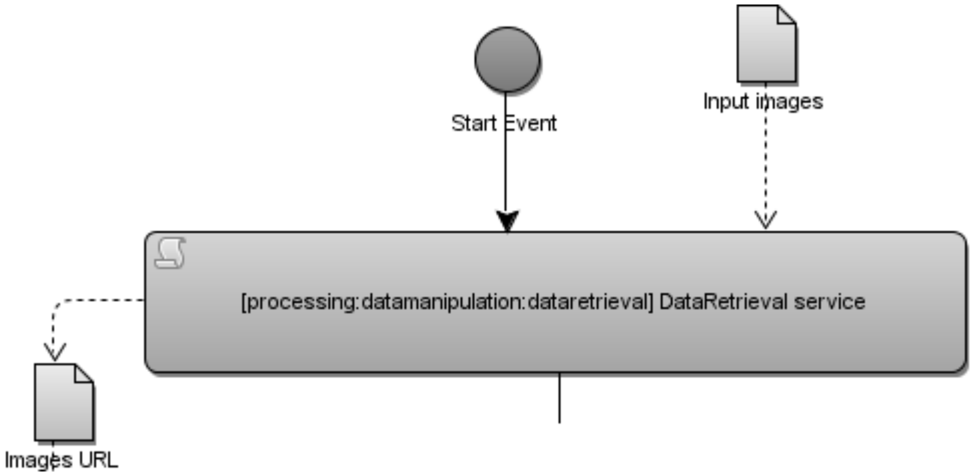

Start Event
Input images
[processing:datamanipulation:dataretrieval] DataRetrieval service
Images URL

***Fig.** 7 Data retrieval adapter*

MosaicN adapter

MosaicN adapter (**Fig. 8**) sends a WPS Execute request with the reference URL of satellite images and the Area of Interest chosen by users to the *MosaicN* service. The WPS service response is a reference URL to the mosaicked GeoTIFF image.

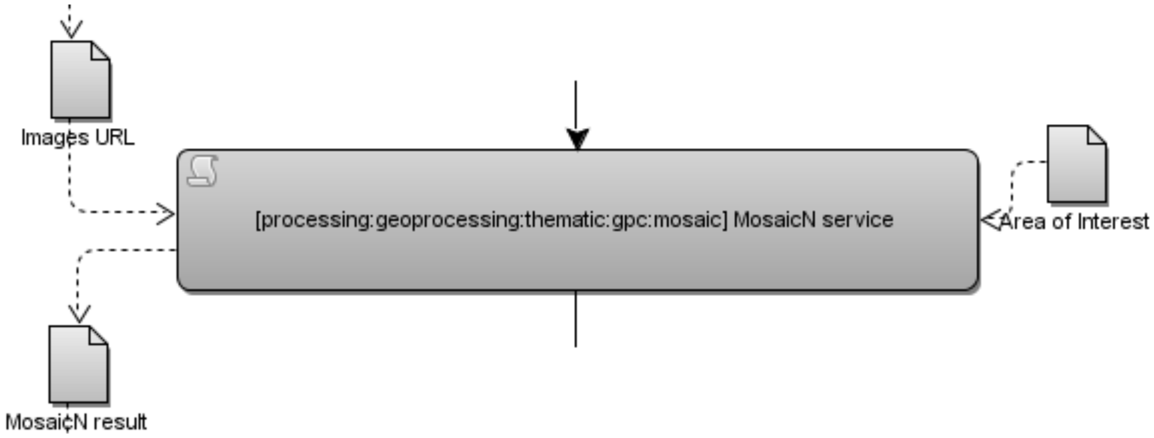

Images URL
[processing:geoprocessing:thematic:gpc:mosaic] MosaicN service
Area of Interest
MosaicN result

Fig. 8 *MosaicN adapter*

Data publisher adapter

As for the previous adapter, Data publisher adapter (**Fig. 9**) allows sending the mosaicked image to the GeoServer service through a WPS Execute request. The WPS service publishes the image into the GeoServer and returns the GeoTIFF image. After the WPS response, the Data publisher adapter executes two GetMap operation requests to GeoServer in order to get the image in PNG and KML format. Finally, the Data publisher adapter exposes the mosaicked image in GeoTIFF, PNG and KML format to the users.

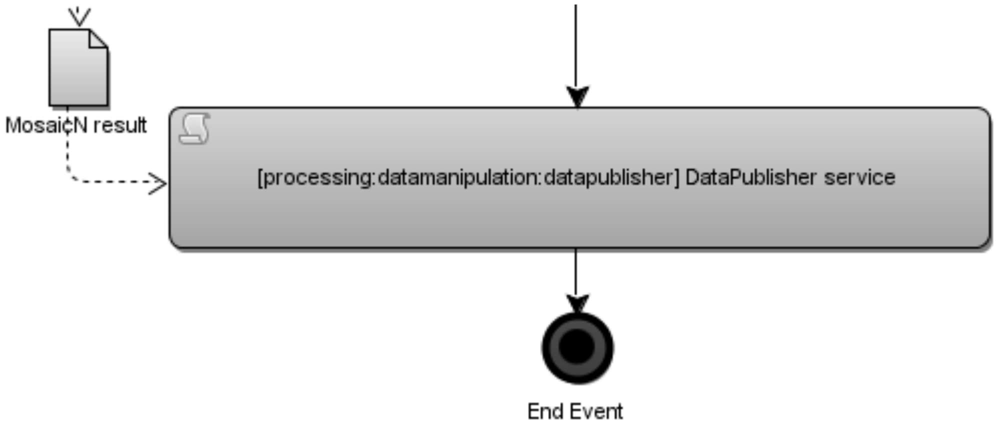

MosaicN result
[processing:datamanipulation:datapublisher] DataPublisher service
End Event

Fig. 9 *Data publisher adapter*

Deployment

All the WPS modules[16] are deployed on a machine located at University of Cluj-Napoca, while the BP broker[17] is deployed on a machine located at University of Geneva.

Test and validation including performance evaluation

CNR-IIA developed a prototype web-application portal (**Fig. 10**) to test and evaluate the BP broker. The portal allows users to discover and execute processes expressed in BPMN.

[16] http://cgis2ui01.mediogrid.utcluj.ro/wps/wps.py?service=wps&version=1.0.0&request=GetCapabilities
[17] http://eopower.grid.unep.ch:8080/gi-portal/

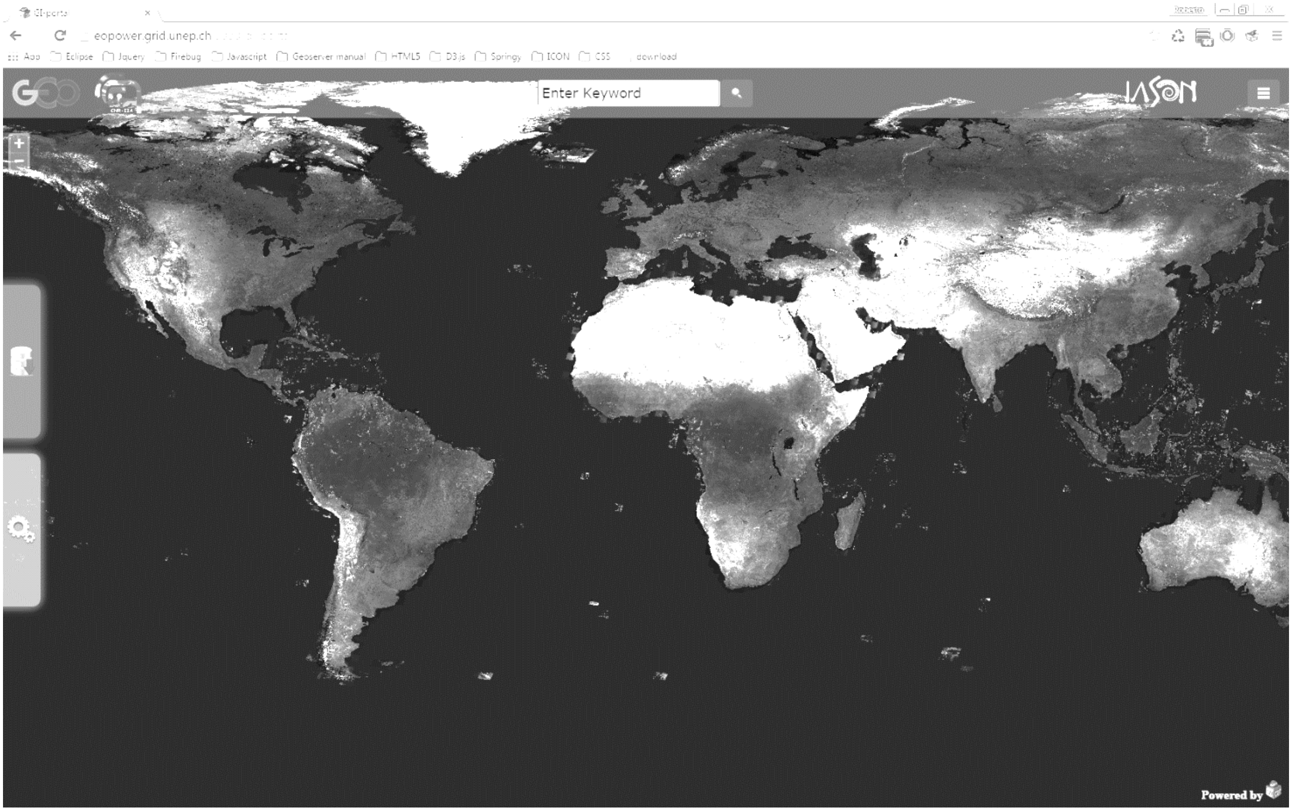
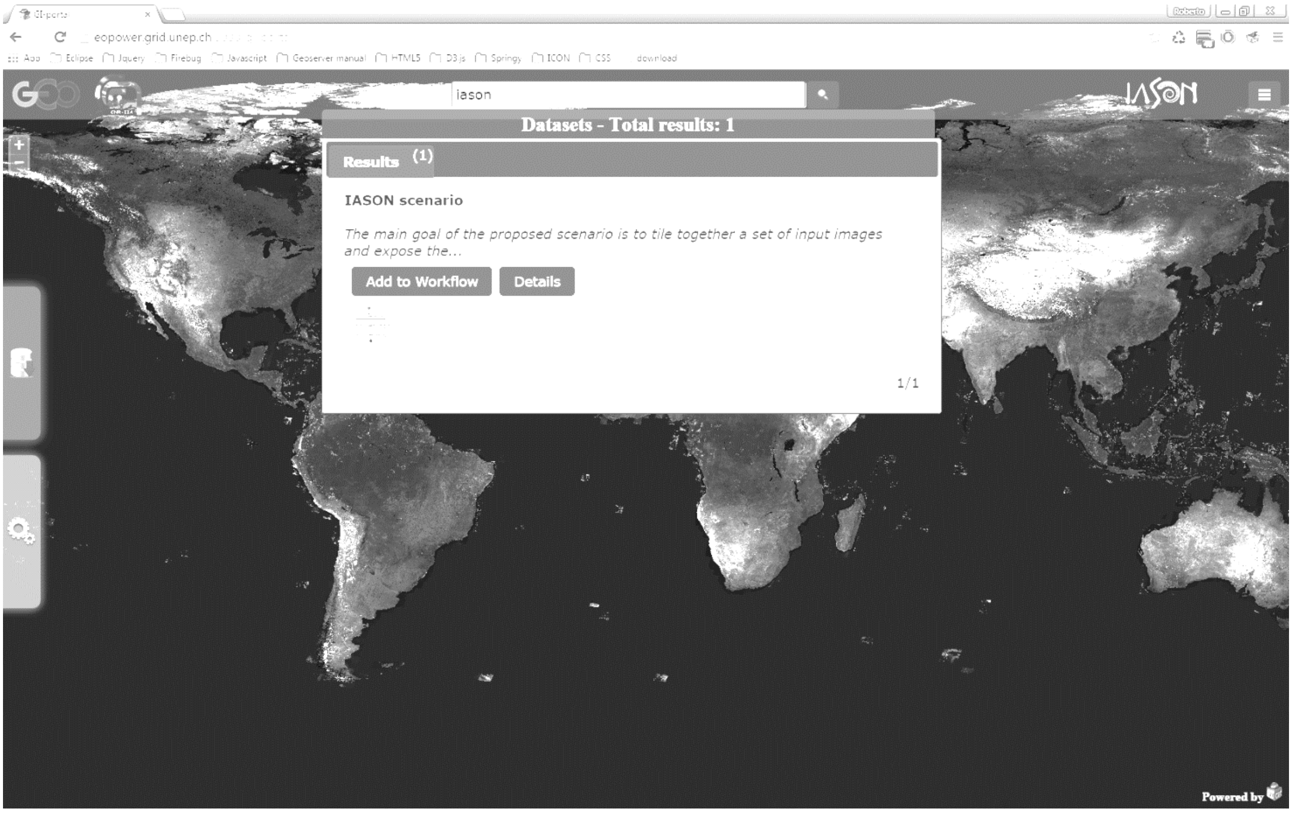


Fig. 10 Prototype Web Portal

The image mosaicking use-case can be added to the list of executable processes clicking "Add to Workflow" button (**Fig. 11**).



Datasets - Total results: 1

Results (1)

IASON scenario

The main goal of the proposed scenario is to tile together a set of input images and expose the...

Fig. 11 Discovery of IASON scenario

Fig. 12 shows the main phases of BPMN execution.

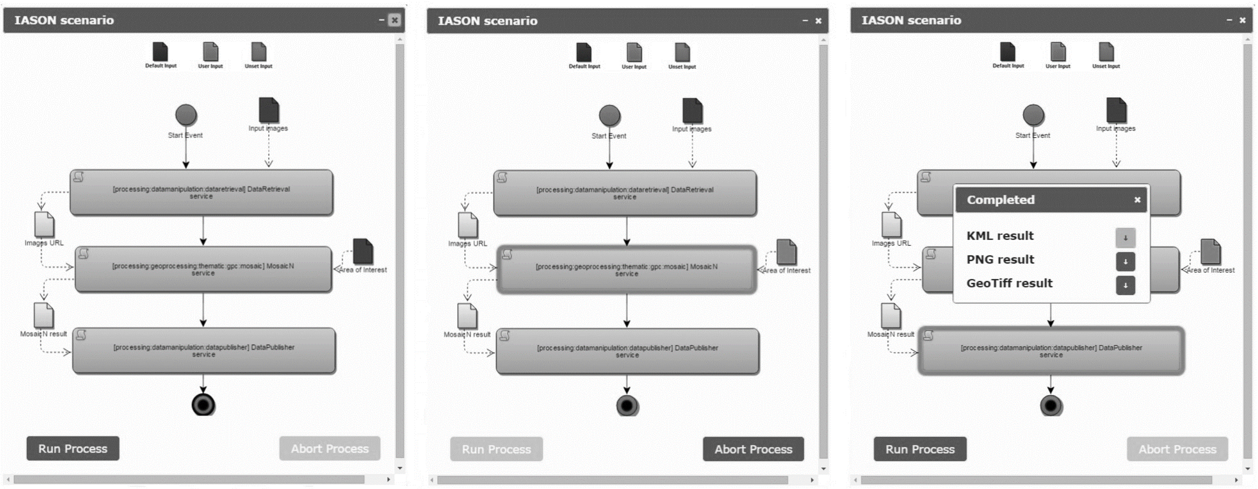

IASON scenario
Default Input
User Input
Unset Input
Start Event
Input images
[processing:datamanipulation:dataretrieval] DataRetrieval service
Images URL
[processing:geoprocessing:thematic:gpc:mosaic] MosaicN service
Area of Interest
MosaicN result
[processing:datamanipulation:datapublisher] DataPublisher service
Run Process
Abort Process
Completed
KML result
PNG result
GeoTiff result

Fig. 12a Executable BPMN ready to be runned.

Fig. 12b BPMN process during the execution.

Fig. 12c Execution results.

Fig. 12 BPMN execution phases

Users can run the process with default values or select a new Area of Interest in the Mediterranean area before running the scenario (**Fig. 12a**). Users can also trace the execution through a graphical feedback: the active component (adapter) in the BPMN diagram is marked with a (yellow) outline during the execution (**Fig. 12b**). At the end of execution, users can download the results through a pop-up window (**Fig. 12c**).

Fig. 13 illustrates an output example of the workflow, encoded in KML format.

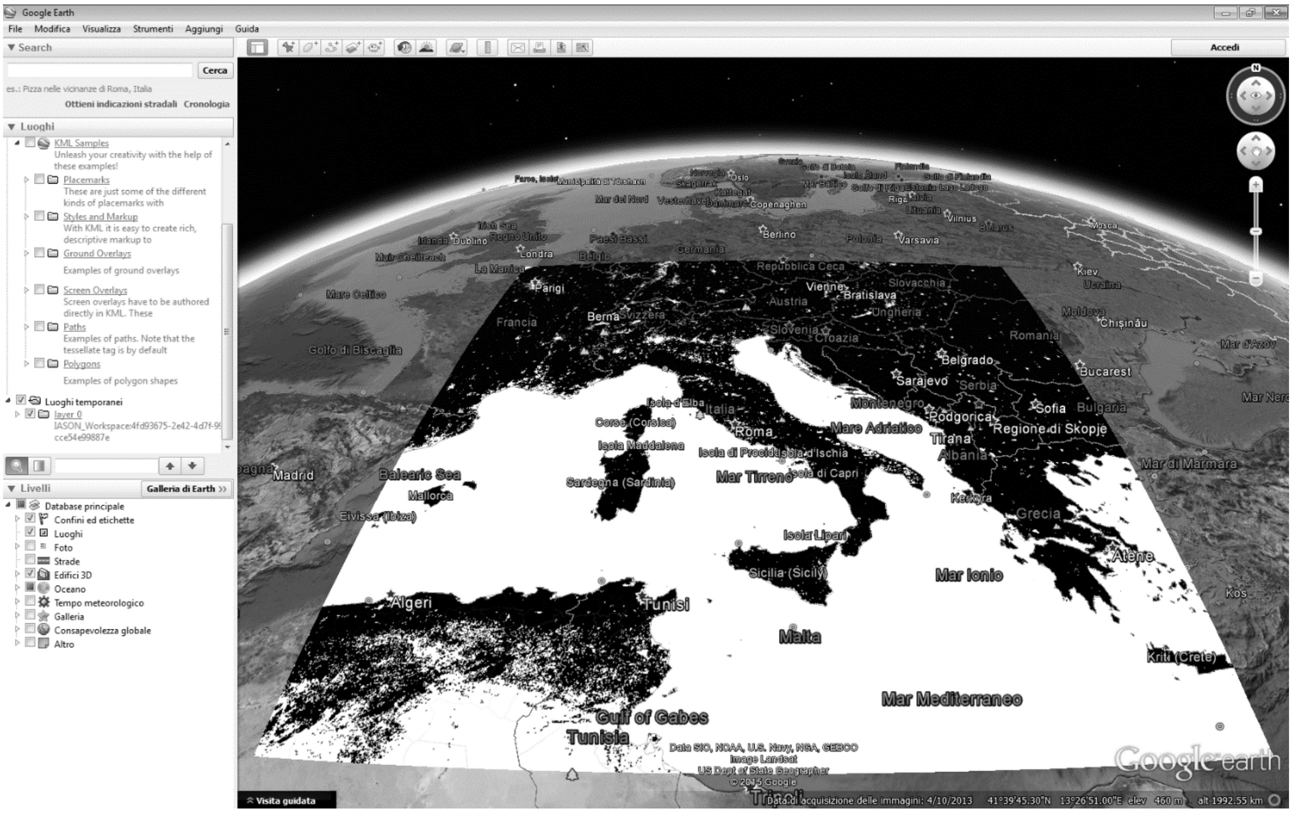


Fig. 13 KML result for the scenario

Data retrieval and Data publisher services take only few seconds to perform their tasks. Completion time mainly depends on *MosaicN* service. In particular, the key factors for the execution time are:

- Number of input images
- Size of Area of Interest

The number of images was N=12 for all the tests carried out. *MosaicN* service with N=12 takes several minutes to mosaic and merge the list of satellite images. Performing the mosaic algorithm on a larger Area of interest (e.g. all the Mediterranean area) requires more time (about 5-6 minutes) than performing the algorithm on a smaller area (about 2 minutes).

Conclusion

The integration of data and computing infrastructures is an inherently complex task. Indeed many infrastructures exist, and their heterogeneity cannot be easily reduced since they are designed to answer to specific requirements and thus implemented using different technological solutions, tools and standard specifications. In particular, a first level of difference is between infrastructure mainly designed for facilitating data sharing, and infrastructures mainly designed for providing computing capabilities.

This heterogeneity poses big challenges to the effective re-use of data and scientific models on infrastructures other than those they were provided initially. In particular users often need to be experts in the different technologies, and developers have to put a big effort on developing interoperability solutions.

The brokering approach addresses these challenges introducing a set of components specifically dedicated to interoperability. Exploiting the brokers capabilities, users do not need to be technology experts anymore, and providers can make their resources easily accessible and reusable in other contexts. In particular a Business Process Broker would be able to execute abstract business processes (i.e. expressed according to the users' viewpoint) using heterogeneous resources provided by multiple infrastructures. Obviously this is possible shifting the overall complexity from client and servers to the brokers which are necessarily complex tools.

The use-case described in the present paper provided outcomes on several directions:

- Added to the use-cases developed in the context of previous research projects, it supports the validity of the brokering approach for scientific data and model sharing. From an abstract business process it was possible to access data and services without any major change on the providers' infrastructures. In particular it could be an effective way to integrate services offered by data sharing infrastructure and those offered by computing infrastructures.
- It validates the current BPB prototype implementation: data are accessed from heterogeneous sources and processed through heterogeneous processing services. The "simple" development of access modules was the only action needed to connect with existing processes.
- It shows that resources (data and models) can be easily reused also in very different contexts without the need of major effort, allowing their exploitation beyond the duration of the project in which they were developed or generated. In this specific case a processing service from the enviroGRIDS project for the Black Sea could be reused in the Mediterranean region by the IASON project and beyond.
- It demonstrates that many interoperability issues raising when trying to run an abstract business model can be solved in a (semi-)automated way as in the BPB, allowing users, namely scientists and decision-makers, to focus on the representation of abstract business models. This clearly contributes to reducing the gap that currently exists between high-level descriptions of business process produced by non-IT experts (e.g., modellers, scientists, decisions-makers) and executable business processes that are low-level realizations produced by IT experts (e.g., WPS services providers).

The proposed solution can be re-used in different contexts since we tried to avoid solutions tailored for the use-case. In principle any business process represented in BPMN, and based on algorithms and data published by any infrastructure could be implemented. It would require the annotation of the BPMN document according to a set of conventions, to provide some information to assist execution, and the configuration or development of accessors for the model services.

Future works will include: a) new use-cases to test the generality of the approach also addressing different computing infrastructures; b) the implementation of transformation in different executable languages and the support of new workflow engines to improve the flexibility of current implementation.

Acknowledments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 603534 (IASON), grant agreement no. 603500 (EOPOWER), and grant agreement no. 227640 (enviroGRIDS). We thank Jean-Philippe Richard, GRID-Geneva IT administrator, for help with the hosting of the application.

References

Balcik, F. B., Mihon, D., Colceriu, V., Allenbach, K., Goksel, C., Ozgur, D., . . . Gorgan, D. (2013). Remotely Sensed Data Processing on Grids by Using GreenLand Web Based Platform. *International Journal of Advanced Computer Science and Applications(IJACSA), EnviroGRIDS Special Issue on "Building a Regional Observation System in the Black Sea Catchment"*, 58-65. doi:10.14569/SpecialIssue.2013.030307

Belmont Forum E-Infrastructures and Data Management Steering Committee. (2015). *A Place to Stand: eInfrastructure and Data Management for Global Change Research - DRAFT.* Tratto da http://bfe-inf.org/sites/default/files/doc-repository/DRAFT_Belmont%20Forum%20E-Infrastructures%20%26%20Data%20Management%20-%20Community%20Strategy%20%26%20Implementation%20Plan.pdf

Bigagli, L., Santoro, M., Angelini, V., Mazzetti, P., & Nativi, S. (2011). *Service frameworks for modelling resources.* UncertWeb Deliverable D2.2.

Bigagli, L., Santoro, M., Mazzetti, P., & Nativi, S. (2015, June). Architecture of a Process Broker for Interoperable Geospatial Modeling on the Web. *International Journal of Geo-Information (IJGI), 4*(2), 647-660. doi:10.3390/ijgi4020647

Cepicky, J. (2007). PyWPS 2.0.0: The presence and the future. *Geoinformatics.* Prague.

Chang, F. J. (2005). *Business Process Management Systems: Strategy and Implementation.* CRC Press. Tratto da Microsoft Developer Network: https://msdn.microsoft.com/en-us/library/ff647958.aspx

Diaz, L., Granell, C., Huerta, J., & Gould, M. (2012). Web 2.0 Broker: A standards-based service for spatio-temporal search of crowd-sourced information. *Applied Geography, 35*(1), 448-459.

Edwards, C. (2013). *In Search of Dark Data.* Tratto da Earthzine Fostering Earth Observation & Global Awareness: http://earthzine.org/2013/05/22/in-search-of-dark-data-2/

ESA. (2013). *Big Data from Space Event Report.* ESA.

ESSI-Lab. (2014). *GI-cat Homepage*. Tratto da http://essi-lab.eu/do/view/GIcat/WebHome

GEO. (2014). *The GEOSS Common Infrastructure (GCI).* Tratto da https://www.earthobservations.org/geoss.php

GEO Secretariat. (2014). *IN-03-C1: Evolution and Enhancement of the GEOSS Common Infrastructure (GCI)*. Tratto da GEO 2012-2015 Work Plan: http://www.geosec.org/ts.php?id=137

GeoServer. (2010). *GeoServer User Manual.* Tratto da http://docs.geoserver.org/stable/en/user/index.html

Giuliani, G., Nativi, S., Lehmann, A., & Ray, N. (2012). WPS mediation: An approach to process geospatial data on different computing backends. *Computers & Geosciences, 47(0)*, 20-33.

Giuliani, G., Ray, N., & Lehmann, A. (2011). Grid-enabled Spatial Data Infrastructure for environmental sciences: Challenges and opportunities. *Future Generation Computer Systems, 6*(3), 27(3): 292-303.

Google Inc. (2015). *Google Earth: Explore, Search and Discovery.* Tratto da http://earth.google.com

Gorgan, D., Abbaspour, K., Cau, P., Bacu, V., Mihon, D., Giuliani, G., . . . Lehmann, A. (2011). Grid based data processing tools and applications for black sea catchment basin. *IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, *1*, pp. 223-228. Prague. doi:10.1109/IDAACS.2011.6072745

Gorgan, D., Giuliani, G., Ray, N., Anthony, L., Cau, P., Abbaspour, K., . . . Jonoski, A. (2013). Black Sea Catchment Observation System as a Portal for GEOSS Community. *International Journal of*

Advanced Computer Science and Applications EnviroGRIDS Special Issue on "Building a Regional Observation System in the Black Sea Catchment", 3, 9-18. doi:10.14569/SpecialIssue.2013.030301

IASON Consortium. (2015). *IASON Project Home Page*. Tratto da IASON Project: http://www.iason-fp7.eu/index.php/en

Lecca, G., Petitdidier, M., Hluchy, L., Ivanovic, M., Natalia, K., Nicolas, R., & Thieron, V. (2011). Grid computing technology for hydrological applications. *Journal of Hydrology, 403*, 186-199.

Mazzetti, P., Nativi, S., Angelini, V., Verlato, M., & Fiorucci, P. (2009). A Grid platform for the European Civil Protection e-Infrastructure: the Forest Fires use scenario. *Earth Science Informatics, 2*(1), 53-62.

Mihon, D., Colceriu, V., Bacu, V., & Gorgan, D. (2013). Grid Based Processing of Satellite Images in GreenLand Platform. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 41-49.

Nativi, S., & Bigagli, L. (2009, December). Discovery, Mediation, and Access Services for Earth Observation Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2*, 233-240.

Nativi, S., Craglia, M., & Pearlman, J. (2013). Earth Science Infrastructures Interoperability: The Brokering Approach. *IEEE JSTARS*, 6(3), 1118-1129. doi:10.1109/JSTARS.2013.2243113

Nativi, S., Mazzetti, P., & Geller, G. (2013, January). Environmental model access and interoperability: The GEO Model Web initiative. *Environmental Modelling & Software, 39*, 214-228. doi:doi:10.1016/j.envsoft.2012.03.007

Nativi, S., Mazzetti, P., Craglia, M., & Pirrone, N. (2014). The GEOSS solution for enabling data interoperability and integrative research. *Environmental Science and Pollution Research, 21*(6), 4177-4192.

Object Management Group. (2015). *Business Process Model and Notation*. Tratto da http://www.bpmn.org/

Petitdidier, M., Cossu, R., Mazzetti, P., Fox, P., Schwichtenberg, H., & Som de Cerff, W. (2009). Grid in Earth Sciences. *Earth Science Informatics, 2*, 1-3.