ISPRS Int. J. Geo-Inf. **2015**, *4*, 647-660; doi:10.3390/ijgi4020647

OPEN ACCESS

ISPRS International Journal of
Geo-Information
ISSN 2220-9964
www.mdpi.com/journal/ijgi/

Article

Architecture of a Process Broker for Interoperable Geospatial Modeling on the Web

Lorenzo Bigagli [†,*], Mattia Santoro [†], Paolo Mazzetti [†] and Stefano Nativi [†]

Institute for Atmospheric Pollution Research, National Research Council of Italy,
Area della Ricerca di Firenze, Via Madonna del Piano 10, 5019 Sesto Fiorentino (FI), Italy;
E-Mails: m.santoro@iia.cnr.it (M.S.); mazzetti@iia.cnr.it (P.M.); nativi@iia.cnr.it (S.N.)

[†] These authors contributed equally to this work.

[*] Author to whom correspondence should be addressed; E-Mail: l.bigagli@iia.cnr.it;
Tel.: +39-055-5226582; Fax: +39-055-5225000.

Academic Editors: Steve H.L. Liang and Wolfgang Kainz

Received: 16 December 2014 / Accepted: 27 March 2015 / Published: 20 April 2015

Abstract: The identification of appropriate mechanisms for process sharing and reuse by means of composition is considered a key enabler for the effective uptake of a global Earth Observation infrastructure, currently pursued by the international geospatial research community. Modelers in need of running complex workflows may benefit from outsourcing process composition to a dedicated external service, according to the brokering approach. This work introduces our architecture of a process broker, as a distributed information system for creating, validating, editing, storing, publishing and executing geospatial-modeling workflows. The broker provides a service framework for adaptation, reuse and complementation of existing processing resources (including models and geospatial services in general) in the form of interoperable, executable workflows. The described solution has been experimentally applied in several use scenarios in the context of EU-funded projects and the Global Earth Observation System of Systems.

Keywords: model web; brokering; processing services; mediation; service composition; geographic information systems; spatial data infrastructures

1. Introduction

The identification of appropriate mechanisms for process sharing and reuse, namely by means of composition (of which pipelining, or chaining, may be considered a special case), is considered a key enabler for the effective uptake of a global Earth Observation infrastructure, currently pursued by the international research community in the Earth and Space Sciences, in initiatives like the Global Earth Observation System of Systems (GEOSS).

Such a facility could effectively enable the integration and interoperability of scientific models, for what several approaches have been proposed and developed, over the last years, at increasing levels of abstraction and generalization [1].

In fact, the initial stove-pipe software tools have evolved to community-wide modeling frameworks, then to Component-Based Architecture solutions, and, more recently, empowered by the Web, started to embrace Service-Oriented Architecture technologies, which have emerged as a mechanism for assembling individual services to create customized applications, also in the geospatial sector [2].

In fact, GEOSS is specifically tasked with the development of the so-called “Model Web”: a dynamic modeling infrastructure to serve researchers, managers, policy makers and the general public. This will be composed of loosely coupled models that interact via Web services, and are independently developed, managed, and operated [3].

However, to date, several challenges remain and both service providers and consumers must address far too complex technological aspects [4]. In particular, the level of abstraction of the current solutions for service composition seems too low for implementing the Model Web vision, which results in limited usability and difficult uptake.

To address this problem, it has been suggested that users in need of running complex workflows, such as those originating from environmental models, may benefit from outsourcing the composition activities into a dedicated external service, according to the Composition-as-a-Service (CaaS) approach [5].

CaaS generalizes the concept known as Workflow-Managed Service Chaining (aka translucent chaining) [2] and presents the same advantages over tightly coupled, closed, integrated systems running in local GIS setups, which generally require hard-wired customization, lack flexibility, and provide limited support to process distribution, sharing and reuse.

In this work, we present an advanced CaaS solution that delegates the composition of services to a smart mediating capacity acting as a broker.

We introduce the architecture of a Process Broker, as a distributed information system for creating, validating, editing, storing, publishing, and executing geospatial-modeling workflows on the Web. The described solution has been implemented, experimented and assessed in several use scenarios in the framework of the FP7 UncertWeb project. It has also been applied to modeling scenarios in the framework of the FP7 MEDINA project and the GEOSS Architecture Implementation Pilot.

Our main contribution is the application of the brokering approach, [6,7] which we successfully applied to discovery [8–10], semantic [11] and access services [12,13] (e.g., in the FP7 EuroGEOSS project [14,15]), to enable high-level, user-friendly service composition, in compliance with the GEOSS Model Web architecture.

The next section provides a background overview on the evolution of modeling approaches and related technologies. In Section 3 we elaborate on the Process Broker approach and its added value. In

Section 4 we describe the architecture of our Process Broker solution, focusing on its conceptual model and its main functionality of transforming abstract workflows into executable ones. In Section 5 we report on its prototype implementation and evaluation. Lastly, in Section 6 we conclude and delineate our prospective work on the topic.

2. Related Work

A simple modeling approach exploits individual software tools for running selected simulations targeted at specific communities. For example, OpenModeller [16] is an open source tool implementing a correlative approach for Ecological Niche Modeling. Usually, a Graphical User Interface (GUI) allows the user to select the parameters and algorithms of the simulation, and to access and visualize the output. These tools are typically open source, to support extension and customization. Sometimes more advanced functionalities are exposed through an open service interface. However, it can be difficult to integrate such tools into more complex scenarios, since useful capabilities for composition, such as logging and event handling, may not be implemented.

The need for interoperable modeling leads to the design of frameworks such as the Object Modeling System, ModCom, The Invisible Modeling Environment, OpenMI, *etc*. These differ in several aspects reflecting the varying needs of the target communities, including domain scope (single *vs*. multi-disciplinary), functionality (model chaining *vs*. step-by-step simulations), and technology (single *vs*. multi-platform). Although these frameworks provide valuable functionalities, they also impose constraints on model developers and integrators, such as requiring a specific programming language or development/deployment platform. Such constraints limit the scope of application and can increase entry barriers. As these frameworks are usually closed environments, the interoperability of spatially distributed models, or of models implemented in different frameworks, can be limited.

Component-Based Architectures (CBAs) embrace mechanisms and techniques for developing coarse yet reusable technical implementation units that are context-aware. In a CBA, units of software are encapsulated as “components” that interact with each other through well-defined interfaces. These manageable units of software can be composed in complex applications within specific “container applications”. Examples are the well-known Kepler [17] and Taverna [18] tools, widely used by specific research communities for scientific workflow management. To improve usability in other communities, user-friendly Web-based user interfaces are being investigated.

A further step towards interoperable modeling is based on Service-Oriented Architectures (SOAs). Here, models are exposed as services, thus shifting the interoperability agreements from the technical environment to the interface specifications, as in CBAs. An example is the eHabitat Web service for ecological modeling [19]. This approach enables the integration of legacy systems, and spatially distributed models, in a way closer to the Model Web principles, that are: [1]

- Open access—anybody can create a service to share their model and anybody (or any machine) can access it;
- Minimal entry barriers—easy uptake for both resource providers (modelers sharing their model on the Web) and users (e.g., other modelers incorporating those models in their own, or end users visualizing the output in their browser);

- Service-oriented approach—access is provided by services (*i.e.*, Web services) of a general-purpose distributed services framework (*i.e.*, the WWW) and resources are a specialization of generic distributed resources (*i.e.*, WWW resources);
- Scalability—exponential growth is inherently supported by Web technologies.

In the last decade, several solutions to enable service composition and workflow management have been presented, often in relation to Semantic Web issues [20,21].

W3C and OASIS are among the main promoters of interoperability and standardization, focusing on building consensus and driving development, convergence and adoption in the context of Web technologies and e-Business standards, respectively. The Open Geospatial Consortium (OGC) activated a Workflow Domain Working Group, to establish a forum for addressing issues related to geospatial workflows.

Open standards, such as the WS-* stack of W3C specifications for service composition, WSDL and the OGC Web Processing Service (WPS) [22] help to abstract the resulting system from the adopted implementation technologies, and support modeling on the Web. The widely adopted OASIS standard for workflow implementation in the e-Business domain, WS-BPEL (or BPEL, in short), has been considered for scientific applications [23].

Given the complexity and portability issues of BPEL, the standard Business Process Model and Notation (BPMN) has been proposed as a user-friendlier format to capture workflows, from which a BPEL process may be generated [24].

3. The Process Broker Approach

The provider of an environmental model must perform complex technological tasks to make it available on the Web. A formal interface to the model must be designed and developed, preferably according to a standard specification (e.g., SOAP or WPS) and, more importantly, it is necessary to provide an actionable, precise (and ideally unambiguous) description of that interface and publish it in a register, for the model to be searchable and usable. Likewise, possible consumers of the model must be able to retrieve and interpret its description, to integrate it in their workflows.

This is different from the current Web scenario, where publishing a document can be done without much knowledge about how the Web works, thanks to intuitive tools that generally do not require programming skills. Specific composition tools support the visual wiring of Web resources and GUI widgets together, allowing the production of mashups that enrich the value of data. These tools contribute to a new vision of the Web, the so-called Web 2.0, where users contribution is substantial.

The Process Broker approach applies the same concept to service composition, by means of a dedicated external service providing the necessary interoperability framework for adaptation, reuse and complementation of existing processing resources (including models and geospatial services in general) in the form of executable workflows [25].

This way, the user can be freed from the need of a composition infrastructure and alleviated from the technicalities of workflow definitions (type matching, identification of external services endpoints, binding issues, *etc*.), so as to better focus on his/her intended application.

Moreover, the Process Broker can provide recommendations derived from an aggregated knowledge base of user interactions and feedback, possibly underpinned by Web 2.0 technologies, to assist the user during design and execution, e.g., suggesting additional models or parameter values commonly used in the case of interest. Or the user may submit a partially abstract workflow definition and leverage the recommendations of the Process Broker to bind it to executable service instances.

This is of particular interest in multidisciplinary scientific contexts, where different communities may benefit from each other, by leveraging and sharing the knowledge acquired through model composition.

Indeed, the Process Broker approach supports combining the recent advances in service-oriented computing with the principles of collaborative research and social networking in general. Arguably, it may be considered a fundamental capability of the Model Web.

The recent ICT trend of resource virtualization has lead to the introduction of concepts such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). IaaS provides virtualized computing power and storage space where the consumer can run arbitrary software, often with advanced capabilities like load balancing; a typical example is Amazon EC2/S3. PaaS provides application development environments supporting different programming languages and common functionalities like database access, caching, *etc*.; an example is the Google App Engine. SaaS provide direct access, usually through a Web browser, to applications running on the computational infrastructure; its evolution based on the cloud services model is also referred to as Business Process as a Service (BPaaS); Google Docs is one of the most used SaaS/BPaaS [26]. The CaaS approach of the Process Broker is somewhat situated at an intermediate level between PaaS and SaaS: its "process execution" functionality may be related to the application level (*i.e.*, SaaS), whereas its "process composition" functionally resembles a lower-level computational platform, onto which arbitrary applications can be realized (*i.e.*, PaaS).

4. Process Broker Architecture

The design of the Process Broker [27] is based on the following rationale: support the reuse and integration of existing resources (System-of-Systems approach); integrate with the GEOSS Common Infrastructure (GCI) and the Model Web principles; comply with the OGC standard baseline.

Applying the brokering-SOA approach [6], we designed a smart mediating service that plays two main roles: as an orchestrator, it is able to invoke the necessary services in a correct order to execute a complex workflow; as a mediator, it is able to handle mismatches in the interface and (meta)data models of the invoked services, as well as to assist the user during the design and the execution phases.

A key technological aspect of the Process Broker is the use of BPMN for workflow encoding. Hence, any standard BPMN editor could be used to edit a workflow, in the design phase. Unlike the more commonly used BPEL, BPMN has been developed to enable the users to define readily understandable graphical representations of workflows, as a standardized bridge between process design and implementation [24]. We preferred BPMN to BPEL also because the latter relies on WSDL descriptions, currently not well adopted within the OGC.

A Web execution interface allows the user to select, start, and monitor the asynchronous execution of the workflows made available by the Process Broker. For machine-to-machine access, the Process Broker exposes a standard WPS interface for workflow execution, as well as an interface to an internal registry based on the ubiquitous ISO 19115 and ISO 19119 geospatial standards [28,29], where the published workflows are stored, along with the library of models and components supported by the

Process Broker. This internal registry integrates with the GCI Service Registries, which references all the geospatial service instances contributed to GEOSS. This design supports the reusability of existing software modules, such as the growing number of mediation plugins that empower the GEO Discovery & Access Broker [30].

In the rest of this chapter, we elaborate on the conceptual model of the Process Broker and on one of its main functionalities: the mechanism supporting the transformation of abstract BPs into concrete (executable) ones. The complete system architecture of the Process Broker defines a full service framework for modeling resources that implements several other functionalities [27].

4.1. Conceptual Model

Figure 1 depicts the concepts of interest for the Process Broker, with their mutual relationships.

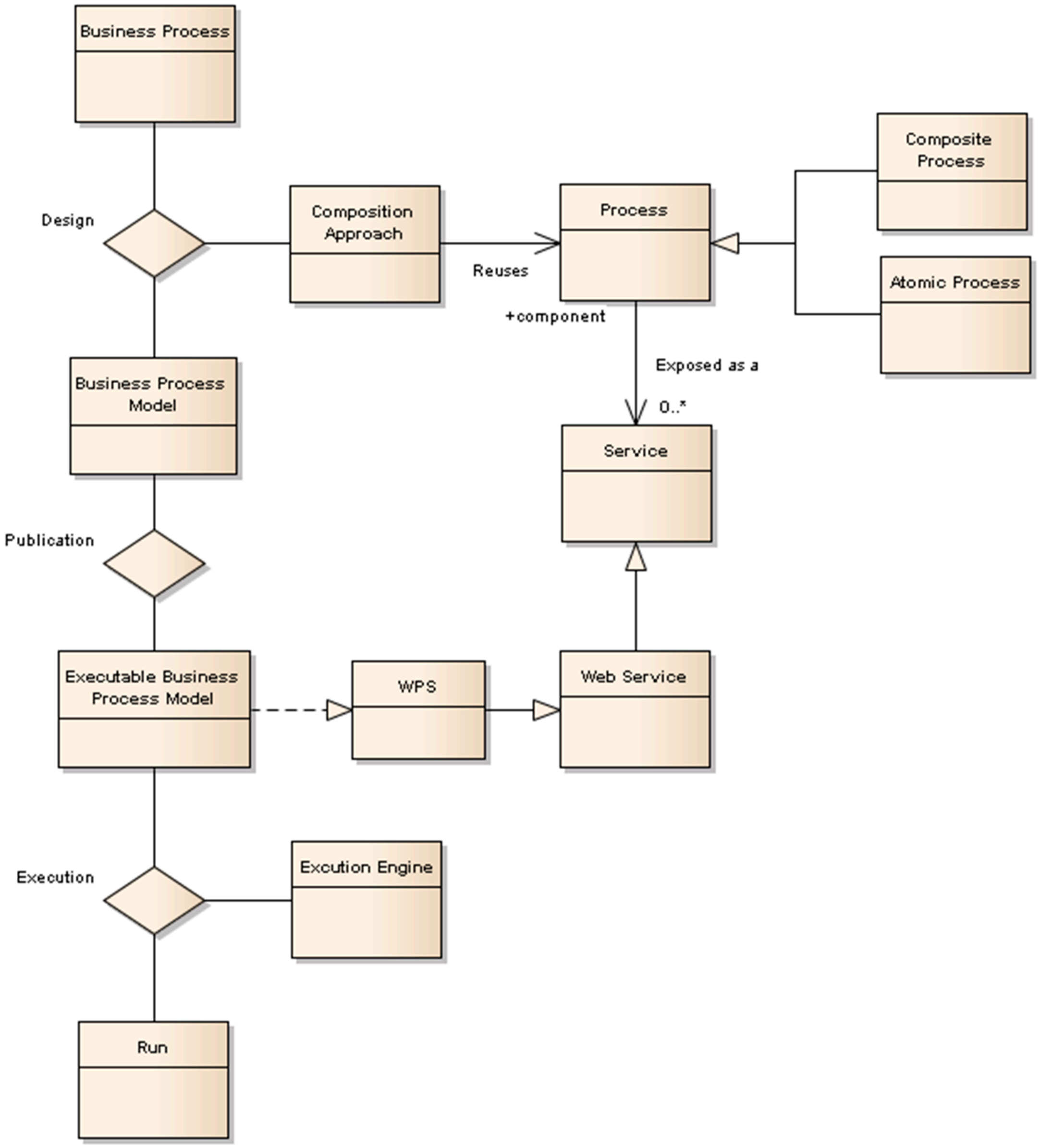


Figure 1. Process Broker conceptual model.

The base concept is the **Business Process**, *i.e.*, the overall abstract phenomenon that the user wants to simulate, typically to answer a "what if" question, as exemplified in Figure 2. A Business Process is modeled into a formal **Business Process Model** through the design phase. Since the referred conceptual entity should be clear from the context, in the rest of this paper we may refer to "Business Process Model" simply as "Business Process" (BP in short). The design phase is characterized by the **Composition Approach**, *i.e.*, realizing the desired functionality by means of existing **Processes** offered by third parties, or by the Process Broker itself.

A process may be exposed as a **Service**, *i.e.*, a distinct functionality provided by an entity through a named set of operations that characterize its behavior. In the context of the Process Broker, a Service is an engineering concept to support system distribution, hence local processes (e.g., internal tasks) are typically not exposed as services. Moreover, the scope of the Process Broker is restricted to software services, typically implemented as **Web Services**.

A BP must be published as an **Executable Business Process Model**, before it can be interpreted and run by the internal **Execution Engine** of the Process Broker. Similar to the compilation of source code, publication entails the resolution of all possibly ambiguous and incomplete aspects (e.g., missing endpoints to concrete service instances, binding issues, type mismatches), as well as BP validation. In the rest of this paper we may refer to "Executable Business Process Model" simply as "Executable Business Process" (EBP in short). A **Run** represents the outcome of the execution of an EBP, including the resulting output, as well as provenance and additional ancillary information (error logs, *etc.*)

Notably, the described Process Broker solution implements an EBP as a **WPS** endpoint, which is itself a (Web) Service, and can hence become part of other BPs. In other words, the newly published BP is immediately available for reuse and, unlike other workflow solutions, the Process Broker allows composing and deploying services on the fly.

This is a key aspect of the Process Broker, since it provides a standard interface for invoking BPs and allows their recursive composition.

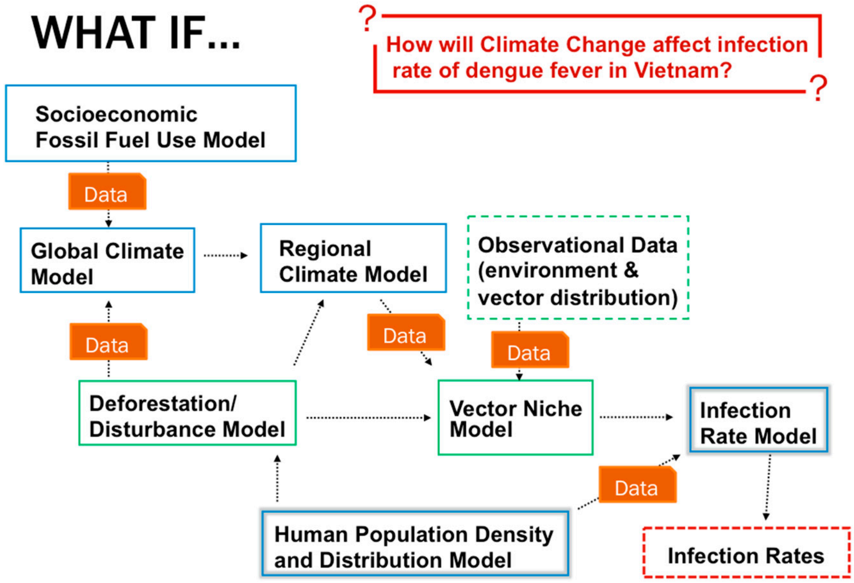

WHAT IF...
? How will Climate Change affect infection rate of dengue fever in Vietnam? ?
Socioeconomic Fossil Fuel Use Model
Data
Global Climate Model
Regional Climate Model
Observational Data (environment & vector distribution)
Data
Data
Data
Deforestation/ Disturbance Model
Vector Niche Model
Infection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

Figure 2. Example of an abstract Business Process.

4.2. Process Broker Conventions

One of the main functionalities provided by the Process Broker is the transformation of abstract BPs into concrete (executable) ones. To achieve this, the Process Broker associates the various sub-components of a BP to appropriate software artifacts, either statically, upon publication, or dynamically, upon execution. Such associations are facilitated by a set of conventions that must be observed in the design of a BP. That is, the resulting BPMN document must conform to the Process Broker conventions.

For example, the conventions define the taxonomy of the components supported by the Process Broker. According to their functionalities, four main categories of BP components are defined:

- Access: executing data access operations;
- Publish: publishing data to standard access services;
- Processing: executing some kind of processing, including geoprocessing based on ISO 19119 taxonomy;
- Utility: basic processing capabilities (e.g., matrix inversion, FFT) used as simple user processes.

Each of the above categories is structured in sub-categories, which specialize the functionalities of the parent. Thanks to the conventions, the Process Broker is able to bind each abstract component in a BP to an appropriate software module interfacing to the (possibly remote) service actually realizing the process. All the syntactical and semantic mediation required to incorporate the service in the workflow is segregated in the module. At present, the supported services include the main OGC Web services and specific WPS profiles for selected models.

The Process Broker conventions are implemented by means of appropriate annotations added to the BPMN encoding of a BP. Hence, both BPs and EBPs are just standard BPMN documents, differing for some annotations (EBPs may also contain additional mediation components introduced by the Process Broker, if needed; see below).

After the binding phase, the Process Broker executes an I/O compatibility check of the BP components, matching their inputs and outputs. This is possible because each mediation module has a well-defined set of I/O parameters. When an I/O mismatch is detected, the Process Broker may solve it by inserting additional mediation components (e.g., a format conversion block), chosen among those available in its component library. Input parameters that remain unmatched are supposed to be provided by the user upon execution. This information is used to generate a standard ProcessDescription document for the OGC WPS interface of the Process Broker.

5. Implementation and Evaluation

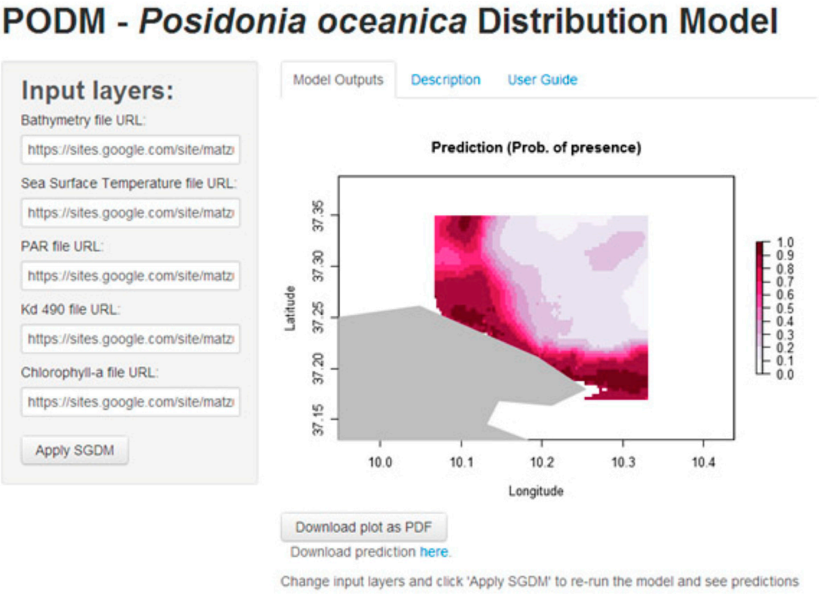
The described solution has been initially prototyped and experimented in the framework of the FP7 UncertWeb project. The reference implementation is based on Java EE technology and on the Apache Tomcat Web container [31]. The Execution Engine of the Process Broker prototype is based on the JBoss Business Process Management technology (jBPM). The core of jBPM is a lightweight, extensible workflow engine written in pure Java that allows executing annotated BPs using the latest BPMN 2.0 specification. However, jBPM is based on a generic process engine, and thus is able to support multiple process languages natively [32].

The Process Broker has been evaluated in two use-scenarios: a habitat assessment scenario based on an ecological niche model, encapsulated in a WPS, to predict the geographic range of a species from occurrence records; and a scenario for evaluating land-use response to climatic and economic change, as output by a land-use model chained to other statistical models. It has proven able to successfully execute the above workflows, starting from intuitive BPMN diagrams designed by non-technical users. The Process Broker has effectively isolated the users from most of the technical aspects of workflow design (which is in fact equivalent to programming), leaving them with the simpler task of modeling geospatial processes in terms of abstract, high level diagrams, hence allowing them to better focus on their application of interest.

More recently, the Process Broker has been applied to a modeling scenario predicting the suitability of coastal habitats to sea grasses proliferation, in the framework of the FP7 MEDINA project and the GEOSS Architecture Implementation Pilot–Phase 6 (AIP-6), demonstrated at the GEO-X Plenary meeting. The demo aimed at showcasing the Posidonia Oceanica Distribution Model (see Figure 3), a tool for sustainable management of sea grass meadows along the Mediterranean coastline, integrating species distribution models with available GEOSS resources [33].

At present, we are assessing the Process Broker in an image mosaicking scenario, in the framework of the FP7 IASON project. The main goal of the scenario is to tile a set of input images and expose the final output in KML format. The scenario is implemented by chaining three WPS instances, the Data Retrieval, the MosaicN and the Data Publisher services, performing the following operations:

- *Data Retrieval service*: parses the URLs of the input GeoTIFF images from an XML file, downloads them and returns the URL to a Web Accessible Folder containing the images;
- *MosaicN service*: creates a mosaic a list of the GeoTIFF images into a single image covering the area of interest;
- *Data Publisher service*: uploads the data result to a GeoServer instance, in three different format: GeoTIFF, PNG and KML.



PODM - *Posidonia oceanica* Distribution Model

Input layers:

Bathymetry file URL: https://sites.google.com/site/matzi

Sea Surface Temperature file URL: https://sites.google.com/site/matzi

PAR file URL: https://sites.google.com/site/matzi

Kd 490 file URL: https://sites.google.com/site/matzi

Chlorophyll-a file URL: https://sites.google.com/site/matzi

Download prediction here.

Change input layers and click 'Apply SGDM' to re-run the model and see predictions

Figure 3. Web interface of the Posidonia Distribution Model. The Process Broker produced the output, based on the input layers provided in the control panel (on the left).

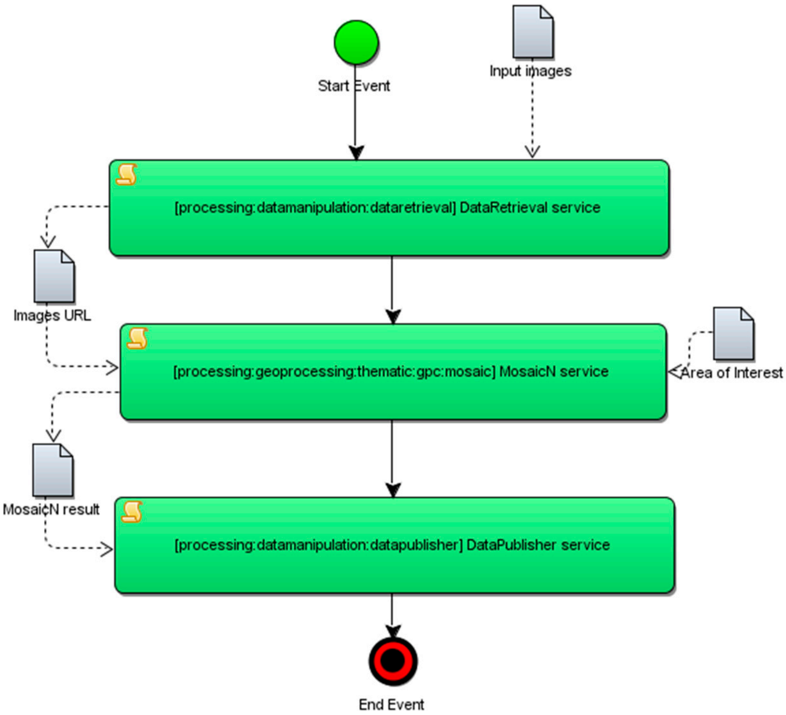

Start Event
Input images
[processing:datamanipulation:dataretrieval] DataRetrieval service
Images URL
[processing:geoprocessing:thematic:gpc:mosaic] MosaicN service
Area of Interest
MosaicN result
[processing:datamanipulation:datapublisher] DataPublisher service
End Event

Figure 4. BPMN diagram for the mosaicking scenario.

Figure 4 illustrates the BPMN diagram for the scenario, according to the BPMN specification and the Process Broker conventions. Every BPMN component (green block in the diagram) represents a model or a service. Inputs and outputs are graphically represented by the BPMN Data Object elements (white document icons). Figures 5 and 6 illustrate the output of this workflow, encoded in the Keyhole Markup Language (KML) and the Portable Network Graphics (PNG) format, respectively.

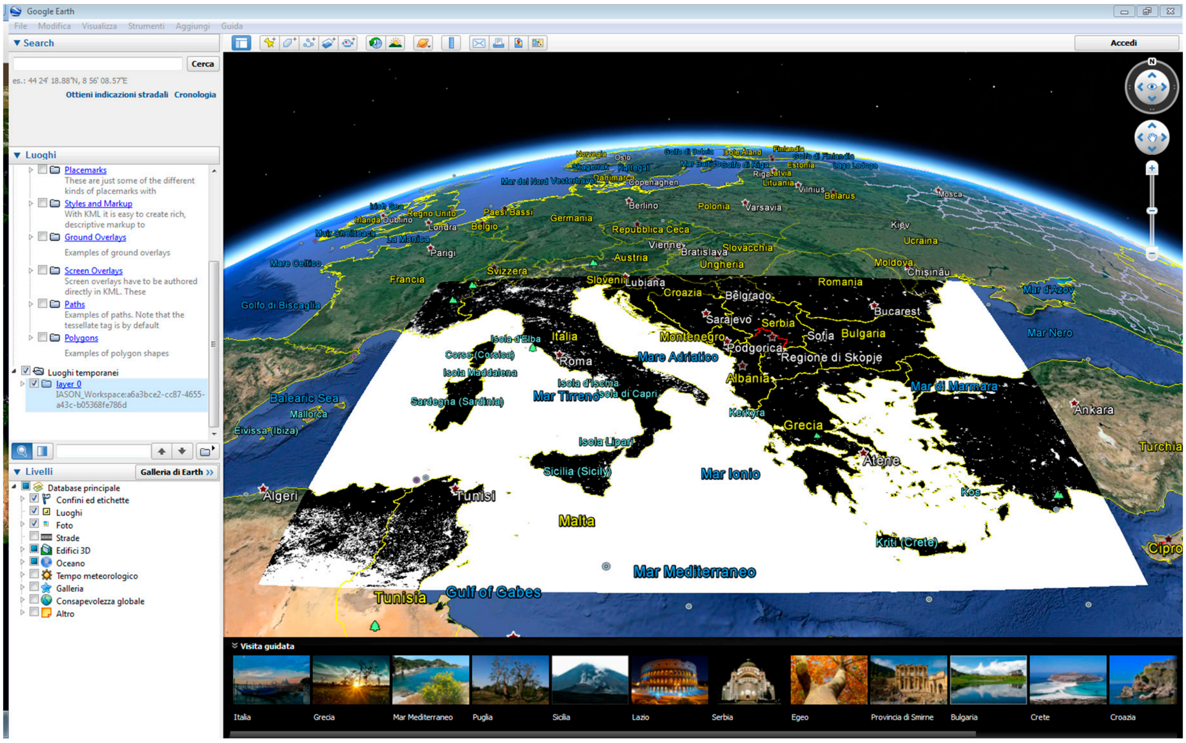


Figure 5. Output of the mosaicking scenario in KML format.

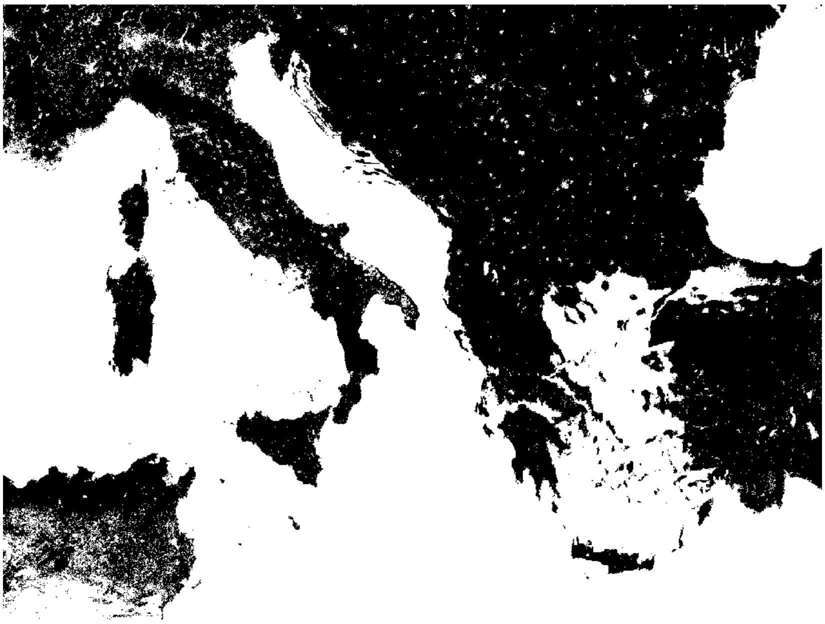


Figure 6. Output of the mosaicking scenario output in PNG format.

6. Conclusions and Future Work

We have applied the brokering approach to service composition, leveraging the capabilities of the GEOSS service framework to enable high-level, user-friendly implementation of geospatial-modeling workflows, according to the Model Web vision.

In fact, delegating composition to a brokering middleware alleviates from the technicalities of workflow definitions (identification of external services, binding issues, type matching, *etc.*) and allows the user, possibly a scientist or a decision-maker, to focus on the process of interest at an abstract level, without having to install and manage a local composition infrastructure. In an advanced scenario, the user may even submit an incomplete workflow, and leverage on the broker recommendations (which may derive from an aggregated knowledge base of user feedback), for completing it. This is of particular interest in multidisciplinary scientific contexts, where different communities may benefit of each other knowledge through model composition.

Another novelty of the proposed solution is that the Process Broker is not bound to an individual workflow platform, usually tailored to a specific running model, or to a specific community of practice. In fact the workflow engine is just an internal, pluggable component of the Process Broker, in charge of interpreting and running the EBPs.

In the future, we plan to continue populating the Process Broker component library, as well as to improve its capabilities to extract knowledge from users' interaction, to provide recommendations during the design and the execution phase, and to collect usage feedback. We also plan to develop a specialized Web editor integrating the Process Broker component library, to better assist the user in the design phase, and to support additional workflow engines.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under Grant Agreement n° 248488.

The authors are grateful to their colleagues Enrico Boldrini, Fabrizio Papeschi, Roberto Roncella, and Valerio Angelini for their contributions. The authors would also like to thank the reviewers for their constructive comments.

Author Contributions

L.B. coordinated the design, prototyping and experimentation of the described solution in the framework of the FP7 UncertWeb project, and wrote the paper. M.S. contributed to the design, implementation and evaluation. S.N. and P.M. performed research on the state of the art and contributed to the vision. All authors read and approved the final manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References and Notes

1. Nativi, S.; Mazzetti, P.; Geller, G. Environmental model access and interoperability: The GEO Model Web Initiative. *Environ. Model. Softw.* **2013**, *39*, 214–228.
2. Alameh, N. Chaining geographic information web services. *IEEE Internet Comput.* **2003**, doi:10.1109/MIC.2003.1232514.
3. GEO 2009–2011 Work Plan, Sub-Task AR-09-02d, Model Web Development Task Sheet, 2009. Available online: http://www.earthobservations.org/documents/tasksheets/latest/AR-09-02d.pdf (accessed on 23 June 2014).
4. Granell, C.; Gould, M.; Esbrí, M. Geospatial web service chaining. In *Handbook of Research on Geoinformatics*; Karimi, H., Ed.; Hershey: New York, NY, USA, 2008; pp. 189–195.
5. Rosenberg, F.; Leitner, P.; Michlmayr, A.; Celikovic, P.; Dustdar, S. Towards composition as a service—A quality-of-service-driven approach. In Proceedings of 25th International Conference on Data Engineering (ICDE), Shanghai, China, 29 March–2 April 2009; IEEE CS Press: Los Alamitos, CA, USA, 2009; pp. 1733–1740.
6. Nativi, S.; Craglia, M.; Pearlman, J. The brokering approach for multidisciplinary interoperability: A position paper. *Int. J. Spat. Data Infrastruct. Res.* **2012**, *7*, 1–15.
7. Nativi, S.; Craglia, M.; Pearlman, J. Earth science infrastructures interoperability: The brokering approach. *J. Sel. Topics Appl. Earth Obs. Remote Sens.* **2013**, *6*, 1118–1129.
8. Nativi, S.; Bigagli, L. Discovery, mediation, and access services for earth observation data. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.* **2009**, *2*, 233–240.
9. Bigagli, L.; Nativi, S.; Mazzetti, P.; Villoresi, G. GI-Cat: A web service for dataset cataloguing based on ISO 19115. In Proceedings of the 15th International Workshop on Database and Expert Systems Applications, Zaragoza, Spain, 30 August–3 September 2004.
10. Nativi, S.; Bigagli, L.; Mazzetti, P.; Boldrini, E.; Papeschi, F. GI-cat: A mediation solution for building a clearinghouse catalog service. In Proceedings of the International Conference on Advanced Geographic Information Systems & Web Services, GEOWS ’09, Cancun, Mexico, 1–7 February 2009.

11. Santoro, M.; Mazzetti, P.; Nativi, S.; Fugazza, C.; Granell, C.; Diaz, L. Methodologies for augmented discovery of geospatial resources. In *Discovery of Geospatial Resources: Methodologies, Technologies, and Emergent Applications*; Diaz, L., Granell, C., Huerta, J., Eds.; IGI Global: Hershey, PA, USA, 2012; pp. 172–203.
12. Boldrini, E.; Santoro, M.; Papeschi, F.; Nativi, S. Data access services interoperability in the geosciences by means of the GI-axe Brokering Framework. In Proceedings of the EGU General Assembly, EGU General Assembly Conference Abstracts, Vienna, Austria, 7–12 April 2013.
13. Nativi, S.; Caron, J.; Domenico, B.; Bigagli, L. Unidata's common data model mapping to the ISO 19123 data model. *Earth Sci. Inf.* **2008**, *1*, 59–78.
14. Vaccari, L.; Craglia, M.; Fugazza, C.; Nativi, S.; Santoro, M. Integrative research: The EuroGEOSS experience. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.* **2012**, *5*, 1603–1611.
15. Nativi, S.; Craglia, M.; Vaccari, L.; Santoro, M. Searching the New Grail: Inter-disciplinary interoperability. In Proceedings of the 14th AGILE International Conference on Geographic Information Science, Utrecht, The Netherlands, 18–21 April 2011.
16. Muñoz, M.E.S.; Giovanni, R.; Siqueira, M.F.; Sutton, T.; Brewer, P.; Pereira, R.S.; Canhos, D.A.L.; Canhos, V.P. OpenModeller: A generic approach to species' potential distribution modelling. *GeoInformatica* **2011**, *15*, 111–135.
17. The Kepler Project. Available online: https://kepler-project.org/ (accessed on 15 December 2014).
18. Taverna—Open Source and Domain Independent Workflow Management System. Available online: http://www.taverna.org.uk/ (accessed on 15 December 2014).
19. Dubois, G.; Skoien, J.; Mendes de Jesus, J.; Peedell, S.; Hartley, A.; Nativi, S.; Santoro, M.; Geller, G. eHabitat: A Contribution to the model web for habitat assessments and ecological forecasting. In Proceedings of the 34th International Symposium on Remote Sensing of Environment, Sidney, Australia, 10–15 April 2011.
20. Dustdar, S.; Schreiner, W. A survey on web services composition. *Int. J. Web Grid Serv.* **2005**, *1*, 1–30.
21. Srivastava, B.; Koehler, J. Web service composition—Current solutions and open problems. In Proceedings of ICAPS Workshop on Planning for Web Services, Trento, Italy, 10 June 2003; AAAI Press: Palo Alto, CA, USA, 2003; pp. 28–35.
22. Schut, P., Ed. *OpenGIS® Web Processing Service*; OGC 05-007r7; Open Geospatial Consortium Inc.: Wayland, MA, USA, 2007.
23. Akram, A.; Meredith, D.; Allan, R. Evaluation of BPEL to Scientific Workflows. In Proceedings of CCGRID'06, Singapore, Singapore, 16–19 May 2006.
24. White, S. Using BPMN to Model a BPEL Process. *BPTrends* **2005**, *3*, 1–18.
25. Brian Blake, M.; Tan, W.; Rosenberg, F. Composition as a Service. *IEEE Internet Comput.* **2010**, *14*, 78–82, doi:10.1109/MIC.2010.2.
26. Mell, P.; Grance, T. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology; Special Publication 800-145; September 2011. Available online: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (accessed on 15 December 2014).

27. Bigagli, L.; Angelini, V.; Santoro, M.; Mazzetti, P.; Nativi, S. Service frameworks for modeling resources. Deliverable 2.2, FP7 UncertWeb, 2011. Available online: http://www.uncertweb.org/documents/deliverables/service-frameworks-for-modelling-resources/download (accessed on 16 April 2015).
28. ISO. *ISO/IS 19115:2003(E)—Geographic Information—Metadata*; ISO/TC 211: Geneva, Switzerland, 2003.
29. ISO. *ISO/IS 19119:2003—Geographic Information—Services*; ISO/TC 211: Geneva, Switzerland, 2003.
30. Santoro, M.; Boldrini, E.; Papeschi, F.; Baldini, A.; Nativi, S. Specification of EUROGEOSS Initial Operating Capacity. Deliverable 2.2.2, FP7 EuroGEOSS, 2010. Available online: http://www.eurogeoss.eu/Documents/EuroGEOSS_D_2_2_2.pdf (accessed on 16 April 2015).
31. Bigagli, L.; Roncella, R.; Boldrini, E.; Santoro, M.; Papeschi, F.; Williams, M.; Jones, R.; Bastin, L.; Cornford, D.; Stasch, C.; *et al.* Reference implementation of UncertWeb framework. Deliverable 2.3, FP7 UncertWeb, 2012. Available online: http://www.uncertweb.org/documents/deliverables/reference-implementation-of-uncertweb-framework/download (accessed on 16 April 2015).
32. Santoro, M. Environmental Model Composition in the GEO Model Web. Doctoral Thesis, Dipartimento di Ingegneria e Fisica dell'Ambiente, Facoltà di Ingegneria, Università della Basilicata, Potenza, Italy, March 2013.
33. Santoro, M.; Pastres, R.; Zucchetta, M.; Venier, C.; Roncella, R.; Bigagli, L.; Mangin, A.; Amine Taji, M.; Malvarez, G.; Nativi, S. Sustainable management of seagrass meadows: The GEOSS AIP-6 Pilot. In Proceedings of the EGU General Assembly, EGU General Assembly Conference Abstracts, Vienna, Austria, 27 April–2 May 2014.