Supporting Content Provision in Environmental Information Infrastructures

Sven Schade[1] and Laura Díaz[2]

[1]Institute for Environment and Sustainability
European Commission, Joint Research Centre
Ispra, Italy

[2]Institute of New Imaging Technologies
University Jaume I
Castellón, Spain

sven.schade@jrc.ec.europa.eu, laura.diaz@uji.es

Abstract. Information Infrastructures for managing and providing environmental resources are requested by numerous initiatives on regional, national, and international scales. While much research focuses on the discovery and consumption of provided content, environmental data and model provision is hardly addressed. Each time an expert creates new information or develops a novel scientific algorithm it is delegated to an expert in information and communication technology to make this content available in a given Environmental Information Infrastructure (EII). From our point of view, this workflow is a bottleneck in sharing environmental content that impedes the efficient maintenance of EII. Accordingly, we have extended the classical three-layered EII architecture with a middleware for assisted content publication and deployment. A first implementation for data publishing is in place, while investigations on the publication of environmental models are ongoing. In this position paper, we briefly present the status of our work and discuss possibilities for publishing environmental models. We point to related activities and outline our future plans. We hope that our contribution will help to increase content availability in EIIs in standard basis and thereby will aid content discovery and model composition.

Keywords: INSPIRE, GEOSS, SDI, Environmental Information Infrastructures, standard services, model web, publishing, deployment.

Introduction

Due to rising challenges of climate change and natural hazards, environmental content sharing became a central need in environmental sciences [1]. Since many considered phenomena, such as wild fires, floods or change in biodiversity, cross administrative borders initiatives for establishing Environmental Information Infrastructures (EII) on different scales emerged over the past years. Those include Infrastructure for Spatial Information in Europe (INSPIRE) and Global Monitoring for Environment and

Security (GMES) on European level, and Global Earth Observation System of Systems (GEOSS) worldwide. An analysis of these three and their interplay has been published recently [2]. In a nutshell, efforts on following INSPIRE implementing rules in GMES are ongoing, while both can be seen as part of European contribution to GEOSS. INSPIRE and GMES also contribute to the Shared Environmental Information System (SEIS) initiative. The concepts behind SEIS focus mainly on reporting but they are still evolving.

So far, EIIs assume that only Information and Communication Technology (ICT) experts can provide content as services. This is because current EIIs are based on Service Oriented Architectures (SOA) where services are implemented according to international agreements and standards. The deployment mechanisms imply to understand these standard service specifications and their implementations. ICT experts became the only mediator between the environmental experts, who create the content, and the infrastructure for content sharing [3] [4].

In order to improve the given situation, we suggested extending existing architectures with components and mechanisms, which assist EII users in content deployment [4]. The GEOSS Service Factory (GSF) realizes this proposal. In the next section, we describe the GSF, and point to related work. As we intent to extend GSF with deployment capabilities for (environmental) models, we discuss possibilities and a future stepwise development. We conclude this position paper by summarizing our findings and by outlining our future development plan.

Recent GSF Developments and Related Work

We enable content deployment by adding a (fourth) layer to the classical EII architecture (Figure 1). This middleware layer (GSF, dashed lines in the figure), acts as a mediator to provide content 'as a Service' to an EII, which is compliant with INSPIRE and GEOSS. With the GSF, applications became able to push newly created content into EIIs using common formats. Appropriate access services are selected based on content types: Metadata, Data, Model, and Warning.

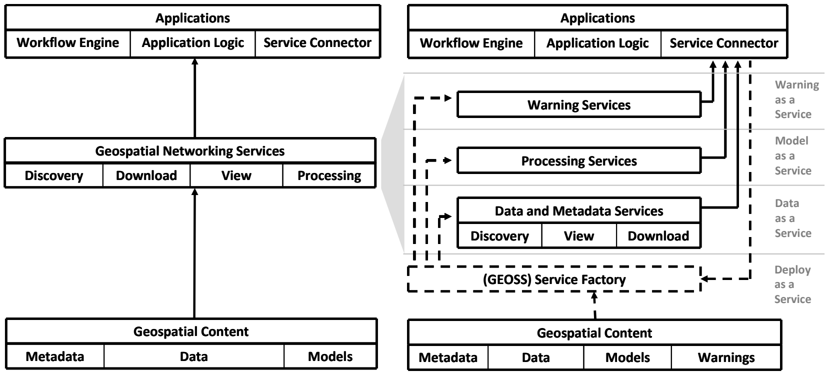

Figure 1: Extending the classical EII architecture (left) with GSF (right).

It might be noticed that the service layer was extended by Warning Services in order to address upcoming requirements for event notification. Among other information resources, warnings are created in the application layer, for example by spatial decision support systems that integrate available models and execute them on the available data sets. Warning Services are used for managing all warnings that have been published within the EII and distribute according notifications to the EII users. These notifications may follow push or pull based approaches.

In order to provide GSF as a (web) service itself. We decided to use a common geospatial standard, the Web Processing Service (WPS) [5]. WPS specification allows encapsulating all kinds of functionality, and it has been proven as mature to expose processing functionality in EII [6], therefore it looks appropriate to describe the GSF interface to provide content publication capability. The added value is that due to the fact that the use of WPS is increasing as well as the number of implementations both for service and client side. To access to GSF from any application any generic WPS client can be used [new7][1].

We designed the GSF using the Abstract Factory pattern from software engineering [8]. The pattern provides a central entry point for content creation/deployment (the Factory), which encapsulated task delegation to specialized deployment components. In this way, the current implementation is easy to extend. As an abstract factory, the GSF holds a group of concrete factories; each of them dealing with a distinct service type and deploying content via transactional service interfaces.

A proof of concept is provided for data deployment in the context of the European Forest Fire Information System EFFIS [9]. Here, GSF is able to provide a unique entry point to deploy vector data (shape files), raster data (GeoTIFF) and even user contributed content (KML) in a View Service and in a Download Service existing in EFFIS. GSF is also able to register basic metadata in a Discovery Service. So far, GSF is not able to deploy processing content such as environmental models, like fire risk calculations or procedures for burned area assessment.

Much research on model deployment has been carried out, including the Model Web concept [10], work on model decomposition [6], and outcomes of the projects that are listed on the web page of this workshop (envip 2010). Still we require scenarios of using GSF for model deployment and have to specify a development plan.

Model Deployment with GFS

At this stage, we remark the following options for model deployment (order indicates complexity, from simple to most difficult implementation):

[1] We have used a generic HTTP client for testing, and a self developed java client [9] but many projects such as 52North and uDIG already provide graphical user interfaces for WPS execution.

(1) *Deploy conceptual model descriptions to a repository.* First, we concentrate on the description of scientific models, which create environmental information. For example, the processing steps, required to generate a burned area map, may be described and deployed as standard encoding of a workflow language. We suggest using the Business Process Modeling Notation (BPMN) [11] for this purpose. Such information helps to understand the model to generate environmental data. Discussions on model improvements may be triggered.

However, BPMN does not provide information on model semantics. We propose using a vocabulary, which is shared within the EII community, for this purpose. It should be available from a shared registry and should be used for labeling the various BPMN elements.

(2) *Deploy executable files to repository.* Following the initial ideas of the Model Web [10] and in line with GEOSS, models may be provided as executables (as *.exe, *.jar, etc) in a repository or Web Accessible Folder (WAF) following GEOSS terminology. Users looking for models may just be provided with a simple list of available files and their formats. For example, a package for statistical calculations of burned area characteristics can be offered stand alone. Execution will still require download and invocation in a suited environment, but at least models become sharable.

In order to make such an implementation usable, we face a model description problem. We require metadata for model evaluation and use. To make a model executable, users will require information about: input and output parameters, required operation systems, versions, and libraries. All may be affected by licensing issues. Additionally, supported interfaces should be described in a common manner. Again, descriptions of model semantics are required. Approaches, such as Web Service Modelling Ontology (WSMO) [12], try to address such issues for web services, but can this be projected to multiple types of executables?

Complementary to the above, we face the practical problem of obsolescence, i.e. required basic technology or software may simply be outdated and not available anymore. Open archives try to tackle these problems of long-term data preservation [13]. Nevertheless, it has to be ensured that all needs to execute the specific environmental modeling algorithms are covered.

(3) *Deploy executable model descriptions to a Processing Service.* WPS has been proven as a technology useful to expose and share processing capabilities in the EII domain [6]. Existing WPS software like 52North implements transactional capabilities [14], to be added to extend the upcoming version 2.0 of WPS. Among other new functionalities WPS 2.0 is considering to deploy new processes in running instances. In other words WPS will support the concept of Composition as a Service (CaaS) [15].

We consider deploying executable process descriptions (using, for instance, Business Process Execution Language (BPEL)) as a next deployment step. In this case, chainable service instances have to be available within the EII and the model has to be provided as an executable process description. For example, assuming all required data is available as Download Service and each processing task as a WPS, the complete burned areas calculation workflow could be made available. A Processing Factory within the GSF would deploy that script and the composed model will be directly available as a distributed Processing Service described with WPS interface. The BPEL deployment functionality of the 52north implementation of WPS [14] may serve as a starting point. However, the use of BPEL limits possible service compositions to components, which operate via SOAP/WSDL.

(4) *Deploy existing software to a Processing Service.* The final step, to expose scientific models fully, is to migrate binary-encoded model components as Processing Services [6]. In order to assist the domain expert and to automate this process as much as possible, we require sophisticated deployment mechanisms and a methodology for workflow modeling with domain experts. Challenges, such as deployment of software developed in diverse programming languages (FORTRAN, Java, etc) must be overcome. Similar issues hold for the operating platforms (windows, unix) in which the models have been developed. Distributed computing, and in particular the 'mobile code approach' [16], in which executable algorithms are sent across a network and executed at distinct nodes, may provide solutions. The relations to grid computing, cloud computing, and virtualization require further exploration.

We believe that the overall solution can only be semi-automatic. For example, only distinct parts of EFFIS can be provided as decoupled processes, due to software packages and dependencies. As for all other options, model semantics still have to be defined in some form of metadata.

Conclusions and Future Work

We argued that content deployment to EIIs faces complex deployment issues and a central bottleneck in environmental information sharing. The GEOSS Service Factory (GSF) was proposed as a solution. Although the GSF concept is supported by a proof of concept implementation for providing environmental data, detailed elaborations for model deployment remain challenging. We pointed to related research and projects. On this basis, we argued for four possible approaches for model provision. As these complement each other, we plan to address them sequentially. Implementations will be guided by the EFFIS example. The GSF will help to increase content availability in EIIs and thereby will aid information discovery and model composition.

Offering GSF as a service provides means to secure deployment. According implementations may be considered in future. This notably differs from a 'secure' execution of models on external machines. The latter is out of the scope of our work.

Acknowledgements

The presented work was partially founded by EUROGEOSS (FP7-ENV-2008-1-226487). The authors thank their colleagues from the Spatial Data Infrastructure Unit of the Joint Research Centre for numerous discussions that shaped the GSF principle. Four anonymous reviewers provided constructive comments for improving an earlier version of this document.

References

[1] M.F. Goodchild, Geographic information science: the grand challenges. In J.P. Wilson and A.S. Fotheringham, editors, The Handbook of Geographic Information Science. Malden, MA: Blackwell,, pp. 596–608, 2008.

[2] P. Smits, S. Cox, F. Fierens, J. Schulze Althoff and A. Biancalana, GIGAS Business Model and Exploitation Plan. GIGAS D 1.2b Annex 1, 2010.

[3] L. Díaz, Improving resource availability for Geospatial Information Infrastructures. PhD Thesis. University Jaume I of Castellón, 2010.

[4] L. Díaz, C. Granell, M. Gould, and J. Huerta. Managing user generated information in geospatial cyberinfrastructures. Future Generation Computer Systems. Future Generation Computer Systems, in press..

[5] P. Schut (ed). OGC Web Processing Service (WPS) version 1.0.0. OGC Standard Document, Open Geospatial Consortium, 2007.

[6] C. Granell, L. Díaz, and M. Gould, Service-oriented applications for environmental models: Reusable geospatial services. Environmental Modelling and Software, vol 25, issue 2, pp. 182-198, 2010.

[7] T. Foerster and B. Schäffer, B, A Client for Distributed Geo-processing on the Web. Lecture Notes in Computer Science (LNCS), vol. 4857, 7th International Symposium on Web and Wireless GIS (W2GIS 2007), 252-263, 2007.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns. Addison-Wesley, 1995.

[9] http://effis.jrc.ec.europa.eu/, last acessed 22nd of September 2010.

[10] G. Geller and F. Melton, Looking forward: Applying an ecological model web to assess impacts of climate change, Biodiversity 9, no. 3&4, 2008.

[11] OMG, Business Process Model and Notation (BPMN), Version 1.2. Object Management Group Standard, 2009.

[12] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel, Web Service Modeling Ontology, Applied Ontology, 1(1): 77 - 106, 2005.

[13] CCSDS, Reference Model for an Open Archival Information System (OAIS). Consultative Committee for Space Data Systems – Blue Book, 2002.

[14] B. Schäffer, Towards a transactional Web Processing Service (WPS-T). In Proceedings of the 6th Geographic Information Days, IfGIprints Nr. 32, Institut für Geoinformatik, Münster, 2008.

[15] M.B. Blake, W. Tan, and F. Rosenberg, Composition as a Service. IEEE Internet Computing, vol. 14, no. 1, pp. 78-82, 2010.

[16] A. Fuggetta, G.P. Picco, and G. Vigna, Understanding Code Mobility. IEEE Transactions on Software Engineering, 24(5), pp. 342-361, 1998.