Which Service Interfaces fit the Model Web?

Sven Schade, Nicole Ostländer, Carlos Granell,
Michael Schulz, Daniel McInerney, Gregoire Dubois,
Lorenzino Vaccari, Michele Chinosi
Institute for Environment and Sustainability
European Commission – Joint Research Centre
Ispra, Italy
e-mail: {sven.schade, nicole.ostlaender, carlos.granell,
michael.schulz, daniel.mc-inerney, gregoire.dubois,
lorenzino.vaccari, michele.chinosi}@jrc.ec.europa.eu

Laura Díaz
Institute of New Imaging Technologies
University Jaume I
Castellón, Spain
e-mail: laura.diaz@uji.es

Lucy Bastin, Richard Jones
Aston University,
Birmingham, UK
e-mail: {l.bastin, jonesrm1}@aston.ac.uk

Abstract - **The Model Web has been proposed as a concept for integrating scientific models in an interoperable and collaborative manner. However, four years after the initial idea was formulated, there is still no stable long term solution. Multiple authors propose Web Service based approaches to model publication and chaining, but current implementations are highly case specific and lack flexibility. This paper discusses the Web Service interfaces, which are required for supporting integrated environmental modeling in a sustainable manner. We explore ways to expose environmental models and their components using Web Service interfaces. Our discussions present work in progress for establishing the Web Services technological grounds for simplifying information publication and exchange within the Model Web. As a main outcome, this contribution identifies challenges in respect to the required geo-processing and relates them to currently available Web Service standards.**

Keywords - Model Web; Web Services; SOA; WPS; WSDL

I. INTRODUCTION AND PROBLEM STATEMENT

Historically, the Web played a role only in environmental data transport, but is now proposed as the foundation of the 'Model Web', - a distributed, multidisciplinary network of interoperating infrastructures of data and models communicating with each other using Web services [1]. As a central concept, complex environmental models and the data required to execute them should be modularized into self-containing geo-processing units [2]. These modules, as well as their compositions, should be made available as services on the Web [3]. The benefits of a Model Web are clear; models, such as similarity calculation between ecosystems or predictions of forest change, and associated data can be more easily accessed, reused and chained for multiple purposes, increasing the repeatability of research and allowing end-users to address more complex issues than when models are used in isolation. Difficult operations can also, to a certain extent, be made available to end-users who do not have the expertise required to solve an indispensable step in a compound modeling process [4]. The gain in flexibility when linking models will be directly proportional to the granularity of the services provided. Basic and generic Model Web components are more likely to be shared than sophisticated ones that are less likely to meet end-users' requirements.

Nonetheless, scientists are currently not using the Model Web to discover, re-use or chain models to the extent that was envisaged. There are significant drawbacks to the full implementation of an information system offering increased access to resources. In this paper, we analyze these drawbacks by outlining some of the central requirements and discussing the lessons learned during our previous work in numerous research projects. In doing so, we provide the foundation for standards-based implementations of the Model Web.

For this work, we assume that models are available and discoverable, so we focus on issues relating to geo-processing for the Model Web, such as ways of approaching model exposure and chaining challenges. Furthermore, we focus on the exposure of models, i.e., sets of algorithms to be used over the Web, and exclude pure offerings of geo-processing tools (such as Sextante [5] or GRASS [6]). Challenges such as the use of ontologies for achieving high-level semantic interoperability, as for example discussed in [7] and [8], are out of the scope of the presented work.

Following the concept of Service Oriented Architecture (SOA) [9], we use standards-based Web service interfaces, namely the Open Geospatial Consortium (OGC) Web Processing Services (WPS) and the Web Service Description Language (WSDL) of the World Wide Web Consortium (W3C) as a starting point for our discussions. We decided to focus on these two technologies, because of their popularity in the scientific community and many research efforts have attempted to use them in a variety of ways in service composition settings.

The following Section II presents relevant background and pointers to related work. In Section III, we outline the central requirements of services for the Model Web and reflect on the specific roles of WPS and WSDL interfaces. We conclude with a brief summary and our future roadmap in Section IV.

ISBN: 978-1-61208-178-6

II. BACKGROUND AND RELATED WORK

The requirement to chain scientific models has led to a wide variety of coupling approaches and frameworks of varying specificity and interoperability [10]. These include standards such as Common Component Architecture (CCA) [11] and Open Modeling Interface (OpenMI) [12], but also orchestration tools such as the Invisible Modelling Environment (TIME) [13] and Taverna [14]. Although some of these solutions have been successfully applied to specific modeling settings, we focus our work on the use of the Web service interfaces, starting from WPS and WSDL, as the necessary standards-based enablers for a sustainable Model Web.

A. Web Processing Service

Initiated in 2004/2005, the idea of the WPS [15] was to provide a generic interface for the publication of any kind of operation or model, since at that point in time, none of the OGC attempts to create specific geo-processing services had been successful. The WPS specifies *GetCapabilities*, *DescribeProcess* and *Execute* operations (Figure 1). The response to the *GetCapabilities* request contains generic information about the WPS and details how to launch the other two request types. The second operation (*DescribeProcess*) identifies all available processes, which might be executed on the concrete WPS and defines all model inputs as parameters in the generic *Execute* request. These can be described as simple types, such as integers, Boolean or String, and Complex types, which have their own schema. There is no restriction on how to define inputs, which results in a user having to define input and output types. Finally, the *Execute* operation triggers a specific model run.

Notably, the WPS was not developed specifically for chaining purposes, but it was envisioned to allow generic client applications to read the *DescribeProcess* document, and, on the fly, to present a corresponding input form to a user. Thus it requires human-to-machine communication and interpretation of the presented form by the user.

The WPS standard has been welcomed at the time as a means by which scientific models may be published and linked, consuming as input parameters data from other standard OGC services, such as the Web Feature Service (WFS) [16] and Sensor Observation Service (SOS) [17]. Projects, such as UncertWeb [18] focus on handling and propagating uncertainty in Web-based models [19]. Several WPSs have been used in modeling chains as proofs of concept and these include: INTAMAP WPS for the automatic interpolation of measured point data [4], the eHabitat ecosystems and habitat similarity modeling WPS [20], and WPS to monitor forest change in the context of the European Forest Fire Information System (EFFIS) [21]. However, describing and profiling models as Web-based processes, and making them available as stable nodes for the scientific community is still a demanding task, with few examples of best practice [22].

B. Web Service Description Language

WSDL [23] is a W3C developed standard for describing Web services. As a mature technology, it has a large community and broad range of software support. It is an XML-based language to describe the functional properties of a service such as its method signatures, input and output messages, details of the transport protocol used (endpoint, SOAP envelope, etc.). As a description language, a WSDL file contains all of the operations or methods offered by a given service. However, WSDL does not specify how client applications access to WSDL files because it is not a communication interaction protocol like WPS. This means that each service provider may offer proprietary rules to access WSDL files.

Figure 2 reveals the differences in the level of granularity between WSDL and WPS. While a WPS provides well-defined interaction operations and each WPS process is a resource by its own, i.e., it has a unique endpoint (see also Figure 1), a WDSL file acts as a public endpoint to access all methods contained in a service. Every single WSDL method remains hidden behind this endpoint. In contrast to the case of the WPS, dedicated tools can automatically generate clients from a WSDL description file to invoke a certain service's method (machine-to-machine communication).

Any service described using WSDL can be only orchestrated by WSDL-compliant workflow software and standards. This means that WSDL is coupled to specific workflow languages, which can be a limitation in certain modeling settings. On the contrary, WPS services can be effectively composed by themselves (e.g., service cascading) because the communication protocol is made explicit. A number of wrapper solutions exist [24], where WSDL documents are created for WPS processes. These contain either abstract message descriptions, or concrete schemas for each process.

III. WPS AND WSDL IN THE MODEL WEB

For the context of geo-processing, we identified five Model Web challenges (Figure 3). We excluded challenges that are related to wider topics, such as model and service discovery, as well as technical issues, such as network fragility or general trust in model results. In this section, we discuss the possible roles of WPS and WSDL service interfaces as well as

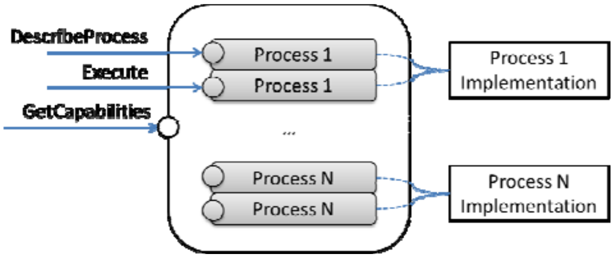

Figure 1. WPS description level.

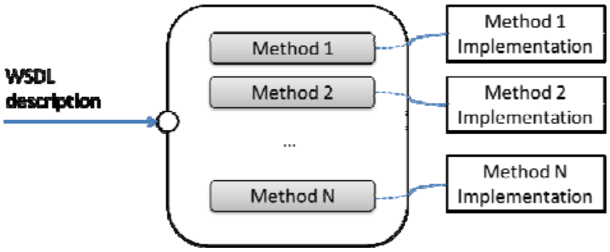

Figure 2. WSDL description level.

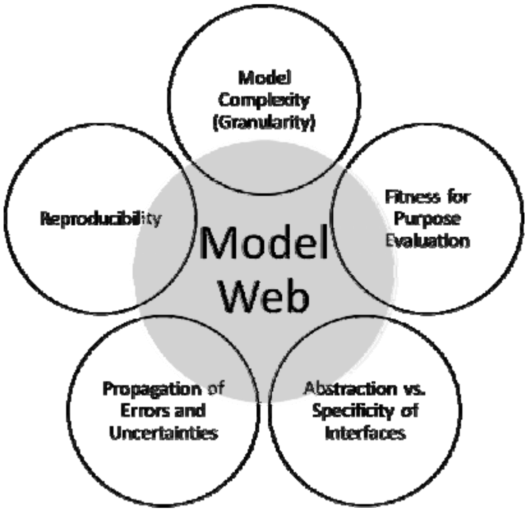

Figure 3. Overview of identified Model Web challenges, which are related to geo-processing.

their combination in relation to each of these challenges. All sub-sections have the same structure: presentations of the challenge are followed by reflections on the use of WPS and WSDL in the given context. Examples are included where appropriate.

A. *Model Complexity*

Models can be wrapped to be exposed with standard interfaces (WPS or WSDL) at different levels of abstraction. Exposing models necessitates finding the right level of service granularity, i.e., the amount of exposed functionality [25]. Coarse-grained services encapsulating a whole model within a single interface reduce the number of service requests from the client; however they might be difficult to reuse in new scenarios. Examples of the coarse-grained approach are the EFFIS WPS, where a forest change model is encapsulated as a service; and eHabitat, which provides access to a similarity calculation for ecosystems and habitats. More generic or finer-grained services normally require less complicated input and output data, and they are more easily reused in new chains, although multiple calls are needed to run more complex models [2]. The INTAMAP WPS, while offering access to complex interpolation algorithms, also fits essentially into this category because of its clear and modular purpose: automated spatial interpolation of point data on a requested region, with or without a consideration of input uncertainties.

A careful consideration of the appropriate granularity level for services could have a positive impact on component reusability and performance [2]. Wrapped models ease the publishing and execution of entire models, but limit re-use, since successful sharing will require some adaptation of inputs and outputs for the new context. In contrast, highly distributed model compositions pose new challenges such as an increase in traffic and network latency as potentially huge quantities of data are exchanged between model components. Above all, the fragmentation of a model over distributed nodes increases the likelihood of breakable nodes and reduces the overview on the processing chain unless each transaction is documented and consistent error handling is implemented.

The well defined relationship between the WPS specification and other geospatial data services, such as WFS and SOS, has encouraged its use within the geospatial community [26]. However, though multiple authors note the need to increase interoperability with mainstream approaches to SOA, such as WSDL, several design decisions make this difficult to implement [27]. While services interfaces that are described with WSDL expose separate execution endpoints to execute an individual process, depending on the kind of binding selected (e.g., SOAP, HTTP Get/Post), WPSs offer a common end point mechanism to expose a service that holds several processes and which delivers each process description as response to a particular *DescribeProcess* request. In this sense a WPS endpoint, both at service- and process-level, serves as a unique identifier. We re-visit this issue in the Section III.C.

B. *Fitness for Purpose Evaluation*

Assuming that a model has been wrapped as a service and discovered, the evaluation of whether that model is fit for a given purpose might not be answered by simply consuming the process description or the corresponding metadata. It might be a matter of interpreting various model runs with varying input parameters, or of running a sensitivity analysis for the potential user's inputs. The encoding of inputs and outputs can also be an issue requiring some investigation and testing, e.g., if the desired input or output data model is not supported. Additionally, a misconfiguration of the model could lead to wrong results and some of the input parameters could be conditional, i.e., several model runs might be required before the final results will provide the user with the desired response. The more complex and unique a model is, the more model runs might be required before a user considers the result as final and is able to assess whether the model is fit for the intended purpose.

Unlike WSDL, the WPS standard has been built assuming human intervention. Thus, the formulation and execution of processing requests controlled by the user is foreseen, and the interface is tailored to human-to-machine communication. Thus, the focus in WPS developments has been at least partly on the development of specific clients through which they can be accessed and generic clients to allow the immediate visualization of results returned by a WPS.

However, conditional inputs and outputs cannot be expressed through any of the interfaces; they can only be described in the process documentation, which might be misunderstood. This might result in various invalid test runs, which could have been avoided if the *Execute* request had been validated against the conditions foreseen by the service provider.

C. *Abstraction and Specificity*

As model complexity increases, more translations and validations are needed for the various input and output data schemas. The huge variety of available data encodings to define inputs and outputs in modeling scenarios and their inaccuracy in defining parameter types are becoming a burden in delivering the Model Web vision.

WPS profiles can help to describe interface specifics in a more detailed and reusable way. For example, it is possible to refer to a Geography Markup Language (GML) application schema [28] in order to specify that only certain types of spatial features should be accepted, or to exploit the validation rules of a schema (such as the XML implementation of the Uncertainty Markup Language (UncertML) [29]) to specify that an input grid, which contains probabilities can be expected to contain continuous values between 0 and 1. These additional data models and dictionaries can be extremely useful in clarifying whether a discovered service is suitable for a user's data, but they must be used rigorously and consistently – an extra pressure on users and clients and an entry barrier. It is also impossible to automate all the necessary validation simply through profiling, when more complex scientific data exchange formats such as netCDF [30] are required. Developments within the UncertWeb project try to support this validation of netCDF datasets, e.g., by extending the existing netCDF Climate and Forecast (CF) metadata convention to encode and identify variables using UncertML references.

Wrapping approaches for WPS use a generic WSDL document to describe any WPS instance at once or on a per process basis [24]. In the WPS specification, a *DescribeProcess* request reveals additional process details, such as required inputs and formats. Due to the generic nature of WSDL, not all the information of a WPS and its processes can be adequately represented. This extra layer of complexity and lack of precision leads to a drastic reduction in the benefits of using WSDL. Graphical workflow composition and code generating tools require the user to know the information provided by the *DescribeProcess* operation, and how to subsequently build an *Execute* request document. In practice, this means that a user has to examine the WPS responses and understand the required parameter types and formats, before they can actually benefit from WSDL's widespread support in chaining environments and orchestration engines in order to enable automation of a process chain.

When defining schemas for each process, which appears to be the solution for the described above problems of a generic WPS, the benefits of WPS appear to be negated, since the request and response messages defined do not validate against WPS schemas. Therefore, the additional overheads of implementing the WPS specification become unnecessary, if not argued by any of the other challenges. In this case, a simpler solution would be to implement the processes using a SOAP/WSDL framework [9]. Such frameworks are able to automatically convert code into a usable Web service.

It is vital to reach the right balance between specific and generic/abstract interfaces and data specifications so as to increase usability and subsequently model sharing. Complex spatio-temporal data and models require careful description and validation, which is beyond the current capacity of the generic interfaces available.

D. Propagation of Errors and Uncertainties

When diverse data sources and processes are composed within a chain whose ultimate outputs will be used for decision-making, a need arises for properly-documented propagation of inherent or introduced errors and uncertainties.

Error and uncertainty propagation are necessary ingredients in assessing the effects of data and model uncertainty on the reliability of the outputs of a model chain. Uncertainty must be properly quantified and communicated to decision makers – for example, by supplying error estimates, quantiles or examples of equally likely alternative scenarios as outputs. In a Model Web context, the language and formats used to do this must be standardized and interoperable. There are several examples of WPS, which use the UncertML approach to characterize the uncertainty on their inputs and outputs. The INTAMAP WPS accepts an Observation and Measurement (O&M) [31] request document containing point observations, which may have associated measurement uncertainties. Depending on the nature of that uncertainty, an appropriate algorithm is selected, and a document returned, which contains interpolated grids of predicted means and variances. Some cross-validation is performed, but this is internal to the service rather than directly accessible to the user as a model-evaluation service. The uncertainty-enabled version of eHabitat, currently under development at the Joint Research Centre (JRC) of the European Commission, samples (or simulates) datasets from inputs with known or inferred uncertainty, and produces summary statistics such as exceedance probabilities and example realizations (again encoded using UncertML), on top of the usual mean predicted habitat suitability map. Again, the simulation (which could effectively be seen as a form of sensitivity analysis) is embedded inside the service and not exposed separately.

An alternative approach, which more clearly illustrates service chaining, is to use a model that in itself does not handle uncertainty. Instead, requests on that model are executed multiple times, using perturbed parameters and inputs, which are sampled from statistical estimates of the uncertainty on those inputs. This allows fairly straightforward propagation of uncertainty on inputs, model parameters and initial conditions, and with some adaptation might even help to estimate and propagate the uncertainty within the model itself. The approach was successfully demonstrated for an air quality assessment WPS by Gerharz and others [32] but raises interesting questions about the pressures of increased network traffic, especially with large and multi-dimensional datasets.

E. Reproducibility

Complex chains of diverse models make it difficult to ensure the reproducibility of model runs, and raise particular curation challenges when the recorded implementations become outdated. Given that most integrated modeling approaches use Monte Carlo simulation to incorporate and assess the impact of uncertainties [33]; this also raises the issue of ensuring the reproducibility of model runs and simulations with a random element. For instance, someone else should be able to reproduce the results of a published model, even though a component might have changed in the meantime, or some manipulation to the input data has been performed.

The eHabitat WPS is a typical example where, depending on the actual values of an input, the model algorithm might use assumptions or omit values and currently, because of the encapsulated nature of these decision rules, there is no way of recording or propagating the branches that occurred for a particular run. While lineage and provenance information [34] provide a partial solution, full reproducibility would also require some form of workflow curation and versioning.

The most obvious means of storing and documenting workflows are orchestration tools, such as Taverna, Kepler [35], or Vistrails [36]. While these tools are designed to produce workflows that can be run and shared, they are far more frequently used to describe the logic, parameters and components of a sequence of processing steps. Even in this limited role, workflow tools are extremely useful as a step towards reproducibility. For this reason, it is very relevant in this context that models described using WSDL documents are far more immediately interoperable with and easier to chain using these tools. On the other hand, this straightforward interoperability is partly because of an assumed simplicity in model inputs/outputs. For example, Kepler has little capacity for declaring complex types and ensuring a correct mapping between them, which caused difficulties in an experimental attempt to expose it as a WPS [37].

IV. CONCLUSIONS AND FUTURE WORK

This paper presented a condensed view on our currently ongoing work that investigates suitable service interfaces for the Model Web. Table 1 (below) provides a direct comparison between the WPS and the WSDL approach. While OGC's WPS requires human-to-machine and machine-to-human communication in order to fully exploit the automated capabilities, W3C's WSDL addresses purely machine-to-machine interactions. Both separate and combined approaches have their role in addressing central geoprocessing tasks in the Model Web. However, it is clear that neither of the approaches for generating WSDL wrappers for WPS-based services/processes is an adequate solution for supporting interoperability outside the OGC community.

Due to the variety of issues and approaches, the Model Web is likely to evolve towards a set of ecosystems of components of different granularities that will evolve independently, largely because of the many chasms (e.g., scientific disciplines, independent networks of developers and projects) between the different communities (see also [38]). This will certainly require in depth investigations on the relation between the Model Web and the Geospatial Semantic Web [39]. Starting points are for example provided in [7] and [8].

Besides further elaborations on service interfaces, our future work will particularly address the impact of harmonized data models for environmental information, as currently being developed in the context of the Infrastructure for Spatial Information in Europe (INSPIRE) [40]. We will base our investigations on the aforementioned eHabitat and EFFIS case studies, following a holistic approach.

Lastly, it should be noted that the interface issues stressed in this paper represent only one research field in geoprocessing for the Model Web. Assuming that generic models will become available as modules, better means for orchestration and chaining will be required. We doubt that solutions from the business sector, such as Business Process Modelling Language (BPEL) [41] or Business Process Modelling Notation (BPMN) [42], will suit the arising needs, but this is a different story.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements n° 248488 (UncertWeb project), n° 226487 (EuroGEOSS project) and n° 284898 (ENVIROFI project). The authors would also like to thank the anonymous reviewers, which have helped to improve the overall value of this paper.

TABLE I. SUMMARY OF WPS AND WSDL COMPARISON.

WPS	***WSDL***
Niche-market	Mass-market
Several endpoints (resource-based approach)	Single endpoint (service-based approach)
Functional description of process/method signatures + other descriptive fields	Functional description of method signatures
Support for profiling	No support for profiling
No need of third-party languages (cascading composition) to enable service composition	Need third-party languages (BPEL, etc.) to enable service composition
Human-to-machine interaction	Machine-to-machine interaction
Support for (mimic) WSDL description	No support for WPS description

REFERENCES

[1] G. Geller and W. Turner, "The model web: a concept for ecological forecasting", Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International, pp. 2469 – 2472.

[2] C. Granell, L. Díaz, and M. Gould, "Service-oriented applications for environmental models: Reusable geospatial services", Environmental Modelling and Software, 25(2), 2010, pp. 182-198, ISSN: 1364-8152.

[3] D. Roman, S. Schade, A. J. Berre, N. Rune Bodsberg and J. Langlois, "Model as a service (MaaS)", AGILE Workshop - Grid Technologies for Geospatial Applications, Hannover, Germany, 2009.

[4] E. Pebesma, D. Cornford, G. Dubois,G. Heuveling, D. Hristopoulos, J. Pilz, U. Stöhlkerg, G. Morinh, and J. Skøien, "INTAMAP: the design and implementation of an interoperable automated interpolation Web service", Computers & Geosciences, 37(3), 2011, pp. 343-352.

[5] Sextante, official web page, http://sextante.forge.osor.eu/ (last access: November 18, 2011).

[6] Geographic Resources Analysis Support System (GRASS), official web page, http://grass.fbk.eu/ (last access: November 18, 2011).

[7] L. Vaccari, P. Shvaiko, J. Pane, P. Besana, and M. Marchese, "An evaluation of ontology matching in geo-service applications", GeoInformatica, 2011, DOI: 10.1007/s10707-011-0125-8.

[8] D. Fitzner, "Formalizing cross-parameter conditions for geoprocessing service chain validation, International Journal of Applied Geospatial Research 2 (1), 2011, pp. 18-35.

[9] G. Alonso, F. Casati, K. Harumi, and V. Machiraju, "Web services: concepts, architectures and applications", Springer, Heidelberg, 2004.

[10] H. Jagers, "Linking data, models and tools: an overview", proceedings of iEMSs, 2010.

[11] Common Component Architecture (CCA), official web page, http://www.cca-forum.org/ (last access: November 18, 2011).

[12] Open Modeling Interface (OpenMI), official web page, http://www.openmi.org/reloaded/ (last access: November 18, 2011).

[13] Invisible Modelling Environment (TIME), official web page, http://www.toolkit.net.au/Tools/TIME/ (last access: November 18, 2011).

[14] Taverna, official web page, http://www.taverna.org.uk/ (last access: November 18, 2011).

[15] Open Geospatial Consortium (OGC), “OGC web processing service (WPS) version 1.0.0”, OGC Standard Document, 2007.

[16] Open Geospatial Consortium (OGC), “ OpenGIS web feature service (WFS) implementation specification – version 1.1.0” OGC Standard Document, 2004.

[17] Open Geospatial Consortium (OGC), “OpenGIS Sensor Observation Service (SOS) implementation specification”, OGC Standard Document, 2007.

[18] UncertWeb project, official web page, http://www.uncertweb.org/ (last access: November 18, 2011).

[19] D. Cornford, R. Jones, L. Bastin, M. Williams, E. Pebesma, and S. Nativi, “UncertWeb: chaining web services accounting for uncertainty”, Geophysical Research Abstracts, Vol. 12, EGU 2010, p. 9052.

[20] G. Dubois, J. Skøien, S. Peedell, J. De Jesus, G. Geller, and A. Hartley, “eHabitat: a contribution to the model Web for habitat assessments and ecological forecasting”, 34th International Symposium on Remote Sensing of Environment, Sydney, Australia, 2011.

[21] European Forest Fire Information System (EFFIS), official web page, http://effis.jrc.ec.europa.eu/ (last access: November 18, 2011).

[22] F. Lopez-Pellicer, W. Rentería-Agualimpia, R. Béjar, P. Muro-Medrano, F. Zarazaga-Soria, “Availability of the OGC geoprocessing standard: March 2011 reality check”, Computers & Geosciences, 2011, DOI: doi:10.1016/j.cageo.2011.10.023.

[23] World Wide Web Consortium (W3C), “Web services description language (WSDL) version 2.0 part 1: core language”, W3C Recommendation, 2007.

[24] Open Geospatial Consortium (OGC), “OWS 5 SOAP/WSDL common engineering report”, OGC Discussion Paper, 2008.

[25] R. Haesen, M. Snoeck, W. Lemahieu, and S. Poelmans, “On the definition of service granularity and its architectural impacts”, International Conference on Advanced Information Systems Engineering (CAiSE’08). LNCS, vol. 5078. Springer, 2008, pp. 375–389.

[26] P. Maué, C. Stasch, G. Athanasopoulos, and L. Gerharz, “Geospatial standards for web-enabled environmental models”, Internal Journal for Spatial Data Infrastructures Research (IJSDIR), Vol.6, 2011.

[27] M. Gone and S. Schade, “Towards semantic composition of geospatial web services - using WSMO in comparison to BPEL”, International Journal of Spatial Data Infrastructures Research (IJSDIR), Vol.3, 2008.

[28] Open Geospatial Consortium (OGC), “OpenGIS geography markup language (GML) encoding standard - Version 3.2.1”, OGC Standard Document, 2007.

[29] Open Geospatial Consortium (OGC), “Uncertainty markup language (UncertML)”, OGC Discussion Paper, 2008.

[30] Open Geospatial Consortium (OGC), “OGC network common data form (NetCDF) core encoding standard version 1.0”, Candidate OpenGIS® Encoding Standard , 2011.

[31] Open Geospatial Consortium (OGC), “Observations and measurements – XML implementation version 2.0”, OGC Standard Document, 2010.

[32] L. Gerharz, B. Proß, C. Stasch, and E. Pebesma, “A web-based uncertainty-enabled Information system for urban air quality assessment”, Geophysical Research Abstracts, Vol. 13, EGU2011-5554, 2011.

[33] L. Bastin, D. Cornford, J Richard, G. Heuvelink, E. Pebesma, C. Stasch, S. Nativi, P. Mazetti, and M. Williams , “Managing uncertainty in integrated environmental modelling frameworks”, submitted to Environmental Modelling and Software, 2011.

[34] R. Devillers and R. Jeansoulin, “Fundamentals of spatial data quality, ISTE, London, UK, 2006.

[35] Kepler, official web page, https://kepler-project.org/ (last access: November 18, 2011).

[36] Vistrails, official web page, http://www.vistrails.org/ (last access: November 18, 2011).

[37] Pratt, A., et. al (2010). Exposing the Kepler Scientific Workflow System as an OGC Web Processing Service. iEMSs 2010, Ottawa, Canada.

[38] S. Schade, P. Mazzetti, Z. Sabeur, D. Havlik, T. Uslander, A. Berre, and L. Mon, “Towards a multi-style service-oriented architecture for earth observations”, EGU 2011, Vienna, Austria, 2011.

[39] M. Egenhofer, “Toward the semantic geospatial web. Proceeding GIS”, 10th ACM international symposium on Advances in geographic information systems, 2002.

[40] European Parliament and Council, “Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)”, Official Journal on the European Parliament and of the Council, 2007.

[41] Organization for the Advancement of Structured Information Standards (OASIS) “Web services business process execution language version 2.0”, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf, 2007 (last access: November 18, 2011).

[42] Object Management Group (OMG), “Business process model and notation (BPMN), version 2.0”, http://www.omg.org/spec/BPMN/2.0/, 2011 (last access: November 18, 2011).

ISBN: 978-1-61208-178-6