Distributed Environmental Modeling

Ari Jolma, Anas Altartouri, Ioan Ferencik
Aalto University, Department of Civil and Environmental Engineering, Espoo, Finland
ari.jolma@aalto.fi

Abstract: Environmental modeling projects consider increasingly complex systems and thus require integration of skills of several people and information from several sources. To overcome problems that modelers face in such projects calls for new approaches. We define “distributed environmental modeling” as the organized use of Internet technology for environmental modeling. This paper examines the hypothesis that distributed environmental modeling has two main forms: collaborative modeling and use of shared resources. We examine this hypothesis in the light of experiences obtained from case studies. We present a case study where a prototype was developed for shared development of Bayesian networks. We present and discuss other cases where we have explored services for modeling and ways to link such services together for modeling. A general conclusion is that it is a benefit to be able to work with text files and/or formal structured descriptions of the objects and tasks. This kind of data lends itself well to be used with existing tools like databases and version control systems. Object-orientation allows the division of the overall task into parts, which can then be integrated in the distributed system level. Another conclusion is that standardization efforts regarding high-level environmental modeling languages would be useful from the viewpoint of distributed modeling. However, another side of distributed modeling is the change in modeling paradigm, which may require changes in mindsets.

Keywords: Environmental modeling; Web services; Modeling services; Environmental modeling languages

1 INTRODUCTION

1.1 Definition of distributed environmental modeling

Environmental modeling projects consider increasingly complex systems and thus require integration of skills of several people and information from several sources. Typically, an important part of integration in projects happens informally, using electronic communication. Traditionally the main method of electronic communication is email but social media technology such as wikis, and various specialized tools like online literature reference databases are becoming more frequently used. We define “distributed environmental modeling” as the organized use of Internet technology for environmental modeling.

This paper examines the hypothesis that distributed environmental modeling has two main forms. The first (without any emphasis on the order) can be defined as collaborative modeling and the second as use of shared resources. Collaborative modeling may just mean communication in the Internet, but it can also be more organized shared development of models or components of models and testing and use of those. Shared use of resources is a form of distributed modeling, where a
modeler uses modeling resources like data, ontologies, and computational tools that are available on the Internet.

In an earlier paper (Jolma and Karatzas 2011) a first look was given at the concept and it was considered from the viewpoint of modeling workflows and software architectures that support them.

1.3 Aims and scope of this paper

This paper aims to analyze the two forms of distributed environmental modeling by presenting cases where we are currently developing prototypes. The prototype, which is used as an example of collaborative modeling, was initially developed within a project focusing on risk analysis in environmental management in the Gulf of Finland (the IBAM research project). The main objective of IBAM was to create an environmental decision model and deliver the results to the decision makers. The requirements of distributed modeling, i.e., to support the collaborative creation and use of Bayesian networks as integrated models of environmental-social systems, were added to what was initially a platform for model delivery. The prototypes which exemplify the use of shared resources are developed in various projects, but mainly in one which aims to create a modeling client for distributed and standards-based services (or services, which could become standards-based).

In the case of modeling clients the existing standards are usually developed within OGC (Open Geospatial Consortium). OGC is an international industry consortium which runs a consensus process to develop publicly available interface standards, especially for web services. OGC has developed, among others, standards for environmental sensors, geospatial data, and for a web processing service. The standards typically define services for workflows that include steps of type “get capabilities of this service”, “get a data resource”, “get more information about a location”.

2 METHODS OF DISTRIBUTED ENVIRONMENTAL MODELING

2.1 Collaborative development of environmental models

Collaborative development is a practical method to develop integrated models. Jakeman (2010) discussed integration in modeling in a presentation he gave at an OGC technical meeting. According to him, integration can be done along several dimensions: issues, geographic features, drivers, disciplines, stakeholders, approaches, and tools. Collaborative development can thus mean involving experts that cover this integration space. Each expert needs a meaningful way to make a contribution into the integrated whole. Collaboration calls for methods to link a particular input into the whole.

Within the above mentioned presentation Jakeman also listed approaches for integrated modeling. According to him these are systems dynamics, Bayesian networks, coupling complex models, agent-based models, and hybrid expert systems. Of these Bayesian networks have arguably the best developed standards for model representation (XMLBIF, Cozman 1998), although that effort appears to have stalled. Standardized model representation is important from the point of view of creating distributed systems in the same way as standardized data services. Coupling of complex models is the most difficult approach in this respect as it requires working installations of several complex models plus a technology to implement the coupling.

The software engineering community has developed widely used and shared systems for collaborative development of software, the version control systems or source code management systems being the best example. In fact, these systems
are just as useful and usable in collaborative development of any objects that can be described with text.

2.2 Use of distributed environmental modeling resources

Environmental modeling requires both information and computational resources. Information resources provide data about the problem, geographical area in question, and knowledge. Computational resources are needed to process information into more useful forms. A lot of modeling involves interaction with information resources and computations where the goal is to specify modeling context, conceptualize the system, etc.

Web services are applications, which allow direct interaction with other applications using standards-based protocols and interfaces (Alonso et al. 2004). Web services may deliver data or they may deliver data processing. In the first case the client of the service sends a detailed request, which the service interprets and fetches, processes, formats, packages and finally sends the data that meets the request in content and form. In the latter case the client needs either to send data for the processing or inform the service how to obtain the input data, in addition to a detailed request. Before a client can send any detailed data or processing requests it needs to know what the service offers and even before that it needs to know about the service. This is the well-known “(how to) publish and find” problem.

2.3 Collaborative platforms for environmental management

Two broad classes of information systems are commonly used to aid environmental management. The first is decision support systems (DSSs), which are interactive systems designed to help managers to utilize data and models to solve unstructured problems (Gorry and Scott Morton 1971, Turban and Aronson, 2001). The second class of systems is communication systems.

The decision that a DSS is built to support may be operational or it may be a strategic one. Some DSSs rely mostly on data or case descriptions while some DSSs rely on models. The model or models may represent the system in which the decision is embedded or they may represent the value structure of the decision maker. Regarding data, a requirement can be set to DSS to support the transformation of data into information and to knowledge. System models referred to above are computational tools that come out of natural sciences, operations research, artificial intelligence, and other domains. Particularly popular tools are simulation models, optimization models, representational models, and causal models. Characteristically, solving unstructured problems requires innovation, review of existing information, judgment, interaction, building confidence, learning, and other skills. Requirements of DSSs include general analysis capabilities, modeling, and integrated use of tools.

Environmental communication can be defined as planned and strategic use of communication processes to support effective policy-making and project implementation geared towards environmental sustainability (Hamacher and Paulus, 1999). Technological developments have supported transitions from broadcast type communication (one sender multiple receivers) to interactive communication (multiple senders and multiple receivers). Other transitions that are taking place include the emphasis from cognitive to experiential learning, from product-orientation to process-orientation, from conflict positions to shared meaning, from distributive to integrative negotiations, and from compartmental knowledge to explicit treatment of complexity, uncertainty, and risk (Scharl, 2004 and references therein).
3 CASE: A DSS FOR INTEGRATED ENVIRONMENTAL MANAGEMENT

The overall topic of the IBAM project was integrated Bayesian risk analysis of ecosystem management in the Gulf of Finland. A research question in IBAM was how to support collaborative creation of an integrative model-based decision support tool that aims to deliver the research results to a wider audience.

3.1 The ecosystem of the Gulf of Finland and its management

The Gulf of Finland is one of the three gulfs in the Baltic Sea. It has a surface area of close to 30 000 km\(^2\) and it is 400 km long and 70 to 130 km wide. It is relatively shallow with an average depth of 38 m and the maximum of 100 m. Close to 6 million people live on its coast in cities like St Petersburg, Helsinki, and Tallinn. The drainage area of the gulf is approximately 420 000 km\(^2\) and cover Finnish, Russian, and Estonian soils. The Gulf was formed only after the last ice age. Due to a large influx of fresh water the salinity of the Gulf is very low and varies between 0.2 and 8.5 ‰. The Gulf is usually partly frozen from late November to late April, but complete freezing requires a cold winter. The northern Finnish coast of the Gulf is winding with abundant islands and skerries, while the southern Estonian coast is smooth and shallow with only a few larger islands. Several fish species live in the gulf and commercial fishing is carried out, herring fishery being the most important.

Several ecological problems exist in the Gulf. Contaminants like dioxins and methylmercury can be found from the area and for example the Finnish Food Safety Authority (Evira) recommends caution when eating herring or salmon from the Gulf. There are also radioactive materials in the sediments, originating mostly from the Tshernobyl accident. Nutrients entering the Gulf and in the sediments are high causing frequent algae blooms. Several alien species are invading the Baltic Sea and also the Gulf. The Baltic Sea Alien Species Database\(^1\) lists 41 alien species that have been found in the Gulf of Finland.

Drivers or pressures on the ecology of the Gulf include, among others, climate change; pollution from agriculture, waste water, and other sources; overfishing; traffic, especially increasing transport of oil and petrochemicals; and development and construction.

3.2 Designing a collaborative modeling platform for the Gulf of Finland

The previous section attempts to highlight that the Gulf of Finland is both a very active economic area with interest from a lot of people and many countries, and environmentally a very diverse, vulnerable, and threatened area. There are several environmental issues concerning the Gulf and one of the main premises of the IBAM project was that an integrated risk analysis should be the basis of environmental management of the Gulf. A starting point for IBAM was to employ Bayesian network methodology for the integration and risk analysis.

3.3 The IBAM DSS prototype

The IBAM DSS prototype was developed using a relational database management system (PostgreSQL), a Bayesian network engine (HUGIN), network visualization software (Graphviz), HTML and CGI technology, and the Perl programming language. A Perl interface for the HUGIN Bayesian network engine, i.e., a library, was developed within the project for integration purposes.

\(^1\) http://www.corpi.ku.lt/nemo/
Bayesian networks are often relatively lightweight text files, which makes the methodology well-suited for collaborative systems. During the project tests were made to using a version control system (Subversion) for the network files. There is no problem in using such systems but relational databases have some other advantages, i.e., better management of several networks and easy extending of the data model, and this is why we used them. HUGIN implements object-oriented Bayesian networks, which allow complex domains to be described in terms of inter-related objects (Koller and Pfeffer, 1997).

Figure 1. illustrates the IBAM DSS prototype. The core of the functionality of the prototype comes from the HUGIN Bayesian network engine, but extending the functionalities is possible by exploiting the possibilities the relational database technology and programming offer. Opening a page with a Bayesian network in a compiled state is currently slow for models of the size shown in the figure because the application starts from the uninitialized state every time (the standard CGI model). It is possible to covert such applications into continuously running parts of the web server (mod_perl technology in the case of the Apache web server), thus keeping models in memory and in compiled state, if possible.

Figure 1. A combination of four screenshots from the IBAM DSS. Top left: the main page, top right: the main page with editing features enabled, bottom left: a model opened for editing, bottom right: a large model opened in a compiled state. In the networks the green nodes are decisions, the pale brown nodes are chance nodes, the white nodes are submodels (the model on the left is an example of a submodel), and grey nodes represent utilities. Network visualization and interaction (HTML usemap feature) is provided by Graphviz and other interaction (HTML forms) is provided by the CGI Perl module).

4 CASE: WEB SERVICES FOR MODELING

Our motivation for studying web services for modeling stems partly from the requirements of on-site modeling (Ferencik et al. 2010) and partly from the general requirements of distributed modeling. We distinguish between opening models as web services and using web services for modeling.

4.1 On-site modeling

On-site modeling, i.e., using information tools while visiting the site being modeled, requires access to many kinds of information and use of computational tools.
On-site modeling is still very much a predicted activity, but typically there would be easy-to-use client software running in mobile computers, and that software would access various type of services. During the HYDROSYS research project (funded under the 7th framework program of EU) we identified six end-user groups in relation to our scenario, which was peri-urban and suburban storm water management. These include local habitants with general interest in water environment, municipal environmental authorities with responsibilities in environmental quality, and contractors with specific needs.

4.2 Experiments with web services

Within this line of research we are at an early prototype stage. In the HYDROSYS project we carried out an experiment to publish a simple hydraulic model as an OGC web processing service. This is explained in some detail by Ferencik et al (2010).

In another project we work with data about threatened species and habitats along the coast of the Gulf of Finland. This data is served as OGC web services within the project to support modeling. The data is attribute-rich, i.e., there are many characteristics that describe both species and habitats. It is very easy to create a web map service out of the data, but the list of possible map types easily gets long. To support more interactive mapping we customized a web map service and developed a system to dynamically create the code for the associated JavaScript-based client (based on OpenLayers). Figure 2 illustrates a web mapping page where the JavaScript code is generated by a CGI program based on the selections of the user.

![Figure 2. A screenshot of the interactive mapping client for studying the spatial distribution of threatened species along the Finnish coast of the Gulf of Finland. The Javascript mapping tool is controlled by the selections made in the HTML form. The controls are for filtering groups of species and for visualization options. Below of the map (not shown) there is a list of the species depicted on the map.](image-url)

Another example of a system that exploits web services to deliver modeling support is a web feature service, which automatically serves all geospatial data

...
layers that are available in a database. This is possible by exploiting the meta data of a database and using corresponding SQL statements when serving the data. We use PostGIS, Perl, and GDAL software. This set up is enhanced with tools that compute new data sets into the database, which are then immediately available as new data layers in the feature service. It is also possible to create more specialized tools for the modelers by adding code (either SQL or Perl in our case) into the server script itself.

4.3 Modeling service

We have been studying the modeling service concept by developing a prototype service using REST architecture. The modeling service will be described in more detail in forthcoming papers but we will give a short introduction here. The service was first designed to support three modeling steps simplified from the ten steps of Jakeman et al (2006). The steps and the initial implementation results are the following.

*Definition of the modeling context:* within this step the modeler develops a conceptual model using ontologies. For an ontology service we are currently testing the SEDRIS service\(^2\), which is a reference implementation of the ISO/IEC standard 18025:2005 developed by the SEDRIS Organization. The modeling context service uses the SEDRIS service to deliver the required functionality.

*Modeling:* within this step the modeler creates an executable model based on the conceptual model. The idea is to develop the model as a code resource in the modeling service. In the current implementation the model is an object that belongs to a subclass of the Model class in Django framework\(^3\). Django is a Python library for web programming and thus the model is Python code.

*Post-processing:* within this step the modeler calibrates and validates the model and visualizes the model results. The prototype service that supports this step is developed using R wrapped to Python using rpy2 code\(^4\).

5 DISCUSSION AND CONCLUSION

The overall goal in distributed environmental modeling is to develop a comprehensive suite of web services and services in the web for modeling needs and useful client applications for modelers. In this paper we have presented and discussed the topic and identified two main use cases. In the first use case several modelers work in collaboration and develop a shared modeling object using a service in the web. In the second use case a single modeler uses a modeling client software, which uses modeling services in the web.

Our observation is that in many cases it is a benefit to be able to work with text files and/or formal structured descriptions of the objects and tasks. Data of this kind lends itself well to be used with existing tools like databases and version control systems. Object-orientation, as for example in our IBAM case supported with HUGIN, allows the division of the overall task into parts, which can be integrated in the distributed system level. Describing modeling workflows using workflow languages with varying detail can be instrumental in designing both modeling clients that use web services and those web services.

The use of modeling languages is at least in principle similar to using general purpose programming language and thus all the tools for distributed software

\(^2\) [http://sedrisreg.sedris.org/](http://sedrisreg.sedris.org/)
\(^3\) [https://www.djangoproject.com/](https://www.djangoproject.com/)
\(^4\) [http://rpy.sourceforge.net/rpy2.html](http://rpy.sourceforge.net/rpy2.html)
A. Jolma et al. / Distributed environmental modeling

engineering should be applicable. The value of environmental modeling languages has been studied before by Karssenberg (2002) who concluded that their strong point is ease of use in constructing and modifying models. We believe that standardization efforts regarding high-level environmental modeling languages would be useful from the viewpoint of distributed modeling.

Technology is developing and it is getting increasingly easier to develop web services for applications and for humans. Another side of distributed modeling is the change in modeling paradigm, which may require changes in mindsets. A central requirement for advances is to identify standards for model descriptions and modeling services.

ACKNOWLEDGMENTS

Funding support from Maa- ja vesitekniikan tuki ry for the project “Interoperable and standards-based web information system for water resources and environmental data and models” is acknowledged.

REFERENCES

Scharl, A., (Editor), Environmental online communication. Springer. 314 pp., 2004.