Managing Uncertainty in Integrated Environmental Modelling: The UncertWeb framework

Lucy Bastin[a], Dan Cornford[b], Richard Jones[b], Gerard B.M. Heuvelink[c], Edzer Pebesma[d], Christoph Stasch[d], Stefano Nativi[e], Paolo Mazzetti[e], Matthew Williams[b].

[a] Aston University, Birmingham, UK (currently on secondment at Joint Research Centre of the European Commission, Ispra, Italy).
[b] School of Engineering and Applied Science, Aston University, Birmingham.
[c] Wageningen University, Wageningen, The Netherlands.
[d] Institute for Geoinformatics, University of Muenster, Germany.
[e] National Research Council of Italy, Institute of Atmospheric Pollution Research, Rome, Italy.

Abstract:

Web-based distributed modelling architectures are gaining increasing recognition as potentially useful tools to build holistic environmental models, combining individual components in complex workflows. However, existing web-based modelling frameworks currently offer no support for managing uncertainty. On the other hand, the rich array of modelling frameworks and simulation tools which support uncertainty propagation in complex and chained models typically lack the benefits of web based solutions such as ready publication, discoverability and easy access. In this article we describe the developments within the UncertWeb project which are designed to provide uncertainty support in the context of the proposed 'Model Web'. We give an overview of uncertainty in modelling, review uncertainty management in existing modelling frameworks and consider the semantic and interoperability issues raised by integrated modelling. We describe the scope and architecture required to support uncertainty management as developed in UncertWeb. This includes tools which support elicitation, aggregation/disaggregation, visualisation and uncertainty/sensitivity analysis. We conclude by highlighting areas that require further research and development in UncertWeb, such as model calibration and inference within complex environmental models.

1 Introduction

The "Model Web" presents a vision of a future where models are exposed as Web Services in a flexible distributed architecture (Geller and Turner, 2007, Nativi et al., 2011). The principle is that models are exposed on the Web and can be discovered, combined into complex workflows and executed over a distributed architecture. Such a system provides tremendous opportunities to enhance scientific modelling by:

- improving the integration of different models to address practical questions;
- increasing the reproducibility and transparency of research by providing clear and repeatable provenance information for modelling outputs;
- allowing more flexible deployment, for example in cloud architectures;
- facilitating the discovery and reuse of model components and code.

The Model Web is in all important respects a developing realisation of the 'Web Service Modeling Framework' conceptualised by Fensel and Bussler (2002) and the four key elements they identify as essential (*ontologies*, *goal repositories*, *web services descriptions* and *mediators)* map very closely to the tools described later in this article.

A practical implementation of the Model Web concept requires interoperability of models and information models in an open system setting. This raises several important challenges. The first challenge is semantic; for multi-disciplinary models to successfully interact in robust systems modelling, there must be unambiguous definitions of all model inputs and outputs, and the scales on which these are measured, since different science domains may use different terms for the same phenomenon, or the same terms for different phenomena. Villa et al. (2009) describe the consequent need for some form of 'declarative modelling', and review recent responses to this problem in the field of environmental modelling. Such vital semantic mapping issues have been addressed in the context of distributed geospatial modelling by the SWING project[1] and its follow-up, ENVISION[2], which is currently developing semantic annotation, harvesting and ontology management tools to support the adaptive chaining required by the Model Web (Janowicz et al., 2010).

In this paper we focus on another issue that faces all modelling frameworks including the Model Web; that of uncertainty management in an era of increasing access to both data and models. The data are typically Earth observations taken from both satellite and in situ systems, while the models range in complexity from empirical statistical models, through box or lumped conceptual models to fully distributed spatio-temporal simulators such as global climate models. Whatever their complexity, these models have several common features: they all read inputs (for convenience, we consider model parameters here also as model input), carry out computations or other manipulation on those inputs, and produce outputs. Model inputs may be observed or measured values, or they may be outputs from other models. In either case, model inputs are subject to errors, and these errors will contribute to the uncertainty of the model output. Quantifying this error or uncertainty is called *error propagation* (Heuvelink, 1998), as the error in model input is propagated into the error in model output. Additional uncertainty will be contributed by the modelling process itself, and

---

[1] http://138.232.65.156/swing/ last accessed 07/05/2011
[2] http://www.envision-project.eu/ last accessed 07/05/2011

we refer to this henceforth as *model structure uncertainty* (Beck, 2005, Refsgaard et al. 2006a). The components of uncertainty in model results are more fully addressed in later sections – at this stage in the discussion, the important issues are their existence and impact. In the following discussion, we will use the word 'sensor' in a broad sense, to indicate any agent which is capable of recording an observation of the real world.

Model outputs and observational data are increasingly being used in policy and decision making, where the use of incomplete information is a risky undertaking, unless some attempt is made to account for and quantify the impacts of gaps in knowledge (Evans, 2008). Some of the most pressing issues facing society, such as climate change (Stainforth et al., 2006) and its economic impacts (Roughgarden and Schneider, 1999), sustainable development (Levy et al., 2000, DeLara and Marinet, 2009) and future energy supplies (Jebaraj and Iniyan, 2006) are subject to significant uncertainties which seriously affect the development of strategy, as well as causing serious scientific debate as to the value and purpose of modelling (e.g., Dessai et al, 2009). In general where there is a decision with a specific cost on taking some remedial action and a loss associated with taking no action (Berger, 1985), and where the costs and losses have significant non-linear dependency on model outcomes, often with critical thresholds, knowing the uncertainty in the model predictions can change the decision taken. .

There is an increasing recognition of the importance of quantifying uncertainty in modelling (e.g. Geza et al, 2009; Allen et al, 2007; Clancy et al, 2010; Feyen and Caers, 2006; Cheng and Sandu, 2009). However, the treatment of uncertainty within modelling frameworks such as the proposed Model Web is not straightforward: firstly, many frameworks which can currently handle components published as services do not have strong or consistent support for propagating or analysing uncertainties, and secondly, the distributed environment introduces a number of new challenges. In service, grid- and cloud-computing based modelling frameworks, model components may be discovered and composed in flexible and potentially complex workflows. However, if this is done without careful description of uncertainty and attention to the quality of (intermediate) model outputs, then the final model output may be too inaccurate for the intended use; and, more importantly, the user may be unaware of this fact. It is therefore essential that the reliability of intermediate and final results is quantified and communicated to the end user. Extending the Model Web to handle and convey uncertainty information in this way is a great challenge.

Complex environmental and geospatial models have specific issues when it comes to uncertainty handling. These include:
1. large amounts of observational and other data do not currently have reliable uncertainty information associated with them;
2. most existing models used across the geosciences and beyond do not have reliable information about their model uncertainties, or model structure uncertainty, available;
3. many of the phenomena of interest are spatial, temporal or spatio-temporal in nature, are measured and expressed at various spatial and temporal scales and often have strong correlations imposed by the physics and dynamics of the natural systems, all of which cause difficulties when evaluating uncertainty;

4. representing spatially and temporally distributed systems typically requires large numbers of variables, and capturing the uncertainties and correlations in these variables is computationally demanding;
5. most models have non-linear responses to their inputs, and thus can have complex probability distributions over their outputs, even for simple parametric input probability distributions;
6. analytic results will be the exception rather than the rule, and thus Monte Carlo methods, with their associated computational expense, will be the default uncertainty propagation mechanism, implying limitation of the proposed solution to computationally cheap models or situations with large computational resources.

The above issues are challenging, but must be addressed in order to make progress and ensure that the Model Web, or indeed any modelling framework, is of practical use. Additional tools are needed to support the practical usage of uncertainty management, for example to address the current lack of uncertainty information, to reduce the computational demands, to manage the issue of changes in spatial and temporal scale and to communicate the uncertain outputs of the modelling workflows.

In this article we describe a coherent framework for extending the Model Web concept of integrated modelling while also taking into account uncertainties. The framework described will be realised by the UncertWeb project (http://uncertweb.org).

The paper is organised as follows. Section 2 introduces the existing approaches to managing uncertainty in modelling, setting the context for the later work. Section 3 describes the practical issues that arise in quantifying and analysing the propagation of uncertainty. Section 4 reviews existing modelling frameworks with a focus on their ability to support uncertainty management and to interoperate with the Model Web. The solutions proposed within the UncertWeb project are described in Section 5, including the key tools that enable users to exploit the "uncertainty enabled Model Web" effectively. The article concludes with a discussion of the likely impact of the "uncertainty enabled Model Web" on future scientific activities and highlights the areas that require further research.

2 Uncertainty in modelling

"All models are wrong; some are useful." (Box and Draper, 1987). This statement originally referred to statistical models but is equally true of physical-deterministic models of complex environmental systems.

Uncertainty is a challenging notion for scientists who have often been trained following a mechanistic, deterministic modelling paradigm. Yet all models are abstractions and simplifications of the complex reality they aim to represent. In this work we do not discuss the various types of, and basis for, uncertainty identified in more philosophical research (e.g., Smets, 1991; Dawid, 2004; Sigel et al., 2010), but rather focus on an operational / practical approach to uncertainty. Almost all uncertainty we seek to address within this work might be characterised as epistemic uncertainty arising from a lack of knowledge, rather than intrinsic randomness, or lack of precision in semantics.

2.1 Origins of uncertainty

While several mathematical and computational frameworks exist for working under uncertainty or incomplete knowledge, we argue that, practically, a subjective Bayesian approach (Jaynes, 2003) is the most natural choice when working with models, observations and their relation to reality (see also Dawid (2004) for an interesting discussion on this issue). Management of uncertainty is essential when working with models of real systems (Brown, 2010). The main uncertainties arise from uncertainties on model inputs (which are often either direct observations from sensors or data derived from observations using other models), and from model structure uncertainty. Uncertainties on observations or derived data can be identified with:

- measurement uncertainty – the intrinsic uncertainty in a given measurement, due to noise in the electronics of the sensor system (Desenfant and Priel, 2006);
- representativity uncertainty – additional uncertainty arising from the difference between the spatial and temporal sampling footprint of the sensor and the defined spatial and temporal representation of reality (Frehlich, 2011);
- sensor model uncertainty – incomplete knowledge of the sensor, or of the forward observation model which maps the measured quantity to the target variable (Agarwal, 1998);
- transmission uncertainty – possible artefacts and processing errors introduced by the computer systems and electronics that carry and process the sensor observations (Bullen et al, 2003).

This list is not exhaustive and rarely are all sources of uncertainty known. A more complete discussion of observational errors can be found in Hill and Tiedeman (2007; Chapter 3). Often, information about the uncertainty of an observation can only be determined *a posteriori*, using validation campaigns. In such a setting the overall uncertainty with respect to reality is assessed using carefully quality controlled 'reference observations' which are often assumed to have negligible error, or using techniques such as triple collocation (Stoffelen, 1998). Using validation data it is possible to estimate the overall uncertainty. If some data are retained for testing only, the uncertainty judgements made on the observations can also be validated (Gneiting et al., 2007).

Model uncertainty (Allen et al., 2002; Brown and Heuvelink, 2005; Lindenschmidt et al., 2007; Refsgaard et al., 2007; Goldstein and Rougier, 2009; Park et al., 2010; Smith and Marshall, 2010) is even more complex and can arise from a range of causes including:

- mechanism / structural uncertainty – it is impossible to include all mechanisms and physical, chemical, biological or human processes that act on reality in the model – they must be simplified and prioritised (Refsgaard et al., 2006a);
- representation uncertainty – for spatial, temporal and spatio-temporal models it is necessary to map the space, time and space-time fields of the real system to the model variables, typically by discretisation or projection onto some basis such as a grid, set of elements or harmonic expansion. This introduces uncertainty due to the finite dimensional nature of the discrete representation (Frehlich, 2011);

- parameter uncertainty – many inputs to a model cannot be directly observed, and we tend to think about these as being parameters in the models, whose values are often empirically determined but essentially unknown (Aster et al. 2005; Tarantola, 2005; Gallagher and Doherty, 2007);
- numerical uncertainty – non-trivial models will require some sort of solver, often integrating differential or difference equations forward in time, and these together with the finite precision representation on digital computers will introduce additional uncertainty (Ataie-Ashtiani and Hosseini, 2005; Clark and Kavetski, 2010).

As with observational uncertainty, this list is not exhaustive, and is missing a description of the now-notorious 'unknown unknowns' (Meyers, 1969, Jaher, 1970, Kerwin, 1993). Such issues are very challenging to deal with in a quantitative framework, but could be important in some complex models, such as Earth System Models, where human activity for example is very challenging to model. Specification of model structure uncertainty is an open and challenging research problem, and many approaches are being pursued, from the more philosophical reification approach (Goldstein and Rougier, 2009), through approaches based on statistical modelling and inference (e.g. Kuczera et al, 2006) to generative approaches which systematically try to simplify more complex models (Cullen and Frey, 1999).

Figure 1. A schematic representation of the relation between people, reality, models and observations.

As shown in Figure 1, a unified framework is needed to integrate observations, models and reality, with the users (people) constructing both the simulators for the systems (reality), and the sensors that observe the system. The users also play a critical role in the above scheme by selecting and channelling appropriate observations to simulators. As discussed above, the processes of modelling and observation are both subject to uncertainties. Typically the observations will be used in a process of calibration to improve the simulators, so that those simulators produce a better fit to the observations of reality. In this process, it is important to consider the representativity of those observations, and to use sensible data splitting techniques for model validation, in order to identify and avoid over-fitting. However, despite calibration and careful formulation, all simulators of real environmental systems retain non-trivial uncertainties, as discussed above. The aims of modelling can be manifold, from improving understanding of the system to more practical questions of prediction or forecasting. When models (or simulators) are used to inform decisions it is critical that any uncertainties in the model predictions on which the decisions are based are taken into account, because these decisions will affect reality and this in turn will affect people.

Probabilistic uncertainty may be measured or estimated and represented in a number of different ways (O'Hagan, 2011), depending on the nature of the phenomenon and the instrumentation available. UncertWeb attempts to recognise a range of descriptions of probabilistic uncertainty, and to support their practical use. The most complete description of a random variable is the probability density function (e.g. Gelman et al, 2003). In practice, observed patterns are more commonly fitted to a known class of probability distribution functions (e.g., Normal, Poisson) or

summarised using statistics (e.g., moments such as mean, variance and skewness). In other contexts, single or multiple realisations of the variable may be of most value, or may be all that are available. This is further discussed in Section 5.1 in the context of the UncertWeb encoding for uncertainty information, UncertML.

3 Practical uncertainty management in modelling frameworks

To quantify a model's output uncertainties one can either modify the model to allow it to propagate uncertainty itself, or wrap it in an environment that manages uncertainty. Modifying the model, i.e., the computer code that does the numerical calculations, requires that the model be made to understand input uncertainty characterizations, carry out computations taking care of these uncertainties, add uncertainties due to parameter estimation, and write output uncertainty characterization in addition to the model outputs. Although this approach might be computationally the most efficient, especially when all or part of the uncertainty can be propagated analytically, it also raises a number of problems: (i) it requires access to the source code and permission to modify it; (ii) it requires deep knowledge of the model source code, and testing of the modifications, (iii) the modified model may no longer be identified under the same name as the original one.

A more practical approach is to keep the model as it is, and wrap it with an application (environment) that takes care of the uncertainty characterizations. The approach used here a simple Monte Carlo simulation, as follows:
1. an application wrapped around the model reads the uncertainty distribution on an input (and/or model parameter);
2. if this distribution is not characterized as a sample, the application draws a sample of size n from this distribution;
3. The application runs the model n times with each of the sample elements as model input, and collects the n model outputs, or realisations, that characterize the output probability distribution;
4. The application can then convert the model output sample to summary statistics such as the mean, variance, or quantiles to approximate confidence intervals as required for subsequent processing.

The issue of model error, or structure uncertainty, is less easily handled with such a wrapper framework, since this really only allows one to propagate uncertainty on inputs such as model parameters and initial conditions. In theory it could be possible to include model structure uncertainty in the wrapper framework as an additive or multiplicative noise component, which could be simulated and added to the realisations generated at step 3. As noted by a reviewer, the UncertWeb framework, although not specifically designed with multi-model ensembles in mind, could facilitate the creation of such ensembles if a range of competing models for a given system were all deployed within the same framework. Such multi-model ensembles are often used to assess model structural uncertainty, and this could be an additional benefit of exposing models on the web in the manner suggested in this paper.

The Model Web blueprint requires that such a wrapper application, which we could call an 'uncertainty-enabler', needs to be implemented as an interoperable Web Service using open standards. The benefits of this are huge: (i) data sources can directly be retrieved from the data source provider, (ii) data sources, models, and the model wrapper can all run on different platforms, under different operating systems, and may partially run on computer clusters, in the cloud or on mainframes, (iii) Monte Carlo samples can be run in parallel, if the available computing infrastructure allows this, and (iv) data or model resources can be exchanged, or re-implemented on different systems without significant change to the overall setup.

Setting up such a system as a Web Service of course requires that a Web client is available to run it. This client can be an interactive tool such as a workflow modeller that allows orchestration and execution of the workflow and that runs in a Web browser, or it can be another model wrapper that takes the currently modelled workflow as a component (i.e, as a "model") in a larger model composition exercise, to realise a further step of model chaining and integration.

4 Existing frameworks for model-coupling

The Model Web is just one of a number of model-coupling approaches of varying maturity. The differences between a number of these approaches have been summarised in Jagers (2010), who notes that conflicting priorities (e.g., performance, ease of use and generality) have, paradoxically, led to a surprisingly wide variety of alternative solutions to the interoperation challenge. It is particularly useful to note that when choosing a framework for existing models, there is often a trade-off in convenience - for example, the effort required to standardise the interface of legacy code can be substantial, but the resulting usability of the model can be greatly increased, since it may then be easily wrapped and combined with other models.

Elements in the orchestration and composition of environmental models can be broadly classified into:
- standard languages and interfaces;
- workflow and orchestration tools;
- frameworks and framework generators.

The following paragraphs set the scene by describing some commonly-used examples which illustrate the state of the art. It will be seen that a number of these do not fit neatly into the three categories above, and some integrated systems address multiple purposes. The interaction between coupling approaches is very important; since the ultimate aim is often to ensure re-use of models, deciding on an interface or language can be critical for the model developer. For environmental models, spatial data models in particular may impose restrictions on the combination of models and the mapping of outputs to inputs. These 'interoperability' characteristics are further investigated in Table 1. We also consider technologies which handle uncertainty in integrated models. These include packages for model calibration, parameter estimation and sensitivity / uncertainty analysis, and are listed, with application examples, in Table 2.

The languages / interfaces category includes BPEL[3] (Business Process Execution Language), a widely-used programming language focussed on message transmission between systems via Web Services, mediated using Web Services Description Language (WSDL) documents. BPEL includes validation and control flow elements which can be interpreted by a variety of engines to execute a process flow. Other initiatives such as OpenMI[4] concentrate less on the framework within which modules are arranged, and more on standardising the interface which each model presents to the world, so that the requirements and limitations of each are clear. Recent adaptations to the OpenMI standard display particular attention to these common issues of interoperability: for example, allowing more abstract inputs and outputs, and permitting inputs which have no specific time frame, thus opening up the tools for use with non-time stepping models. The Common Component Architecture (CCA) [5] is another component standard which appeals to scientific modellers largely because of its support for multi-dimensional data arrays and parallelisation. In the data mining community, the XML-based PMML (Predictive Model Markup Language) is commonly used to summarise and exchange complete summaries of models complete with defined inputs and outputs. CSIRO's ICMS[6] (Interactive Component Modelling system) is considered under this 'languages' heading, since its primary focus is on allowing the development of executable model components in a system-specific, C-like language called MickL.

Workflow tools such as Taverna[7], Kepler[8], Vis Trails[9] and Trident[10] provide user-friendly GUIs within which modular processing or data entities can be arranged, inputs mapped to outputs and control / break conditions defined. The resulting workflow chains can be stored, published[11], shared and exposed as encapsulated models, while the component models themselves must simply expose a WSDL document describing each process, and its inputs and outputs. Thus these tools can be used as engines for interacting with BPEL workflows, as well as, for example, compiled C code or R scripts. Another, more specific, orchestration tool, the Open Modelling Engine (Rizzoli et al., 1998) can be used to schedule MickL components like those mentioned in the paragraph above.

Finally, there are also a host of more or less discipline-specific frameworks for combining models and controlling their execution, such as Delta Shell[12], FRAMES[13] (Framework for Risk Analysis of Multi-media Environmental Systems), SME (Spatial Modelling Environment)[14], Tarsier[15], ICMS (the Integrated Component Modelling System)[16], Fluid Earth[17], TIME[18] (The Invisible Modelling Environment), MCT

---

[3] http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf   last accessed 07/05/2011
[4] http://www.openmi.org   last accessed 07/05/2011
[5] http://www.cca-forum.org/   last accessed 07/05/2011
[6] http://www.clw.csiro.au/products/icms/   last accessed 09/05/2011
[7] http://www.taverna.org.uk/   last accessed 07/05/2011
[8] https://kepler-project.org/   last accessed 07/05/2011
[9] http://www.vistrails.org   last accessed 03/10/2011
[10] http://tridentworkflow.codeplex.com/   last accessed 07/05/2011
[11] http://www.myexperiment.org/   last accessed 07/05/2011
[12] http://public.deltares.nl/display/DS/Home   last accessed 09/05/2011
[13] http://www.epa.gov/athens/research/modeling/3mra.html   last accessed 07/05/2011
[14] http://www.uvm.edu/giee/IDEAS/sme/docs/SME_guide.html last accessed 07/05/2011
[15] http://ecoviz.csumb.edu/wiki/index.php/Tarsier last accessed 07/05/2011
[16] http://www.clw.csiro.au/products/icms/   last accessed 04/10/2011
[17] http://sourceforge.net/projects/fluidearth/   last accessed 03/10/2011

(Model Coupling Toolkit), ESMF[19] (Earth System Modelling Framework), OASIS[20] (Ocean Atmosphere Sea Ice Soil), CESM[21] (Community Earth System Model) and O-PALM[22]. Many of these frameworks include standard modules for applications such as hydrological or climate modelling. A recent development is the generic Bespoke Framework Generator[23] (Armstrong et al., 2009) which generates wrappers and control code wrappers for model sequences based on standardised model metadata which is collected in XML schemata. The BFG has been used to generate an updated version of the GENIE[24] framework for Earth System Modelling, and is being used for the UK Met Office's FLUME[25] (Flexible Unified Model Environment).

4.1 Interoperability in modelling frameworks

Interoperability between the approaches described above is variable, and some aspects (often strongly influenced by the discipline from which each approach arose) are summarised briefly in Table 1. We give particular attention to the spatial data models employed by each approach, since these are of particular importance for many environmental models and datasets. To quote Mattot et al. (2009) '... the irony in design of both model evaluation tools and integrated modeling systems is that everyone wants to define the 'standard' and be the integrative framework'. However, there have been considerable moves within the modelling community towards interoperability and agreement on common interfaces at least, and Table 1 illustrates how this has expanded the usability of coupling technologies.

In terms of model code, almost all frameworks described here support the use of compiled C, but other languages such as Java and Fortran are less universally supported and often must be wrapped before use. Jagers (2010) presents an extremely useful summary of the capabilities of some of these technologies, particularly with regard to their 'code invasiveness' (i.e., the requirements that each imposes for rewriting legacy code) and their capacity to support high-performance computing. Rahman et al., (2005) describe a typical choice between rewriting and wrapping, where the decision can depend largely on the complexity and current performance of algorithm code. In their example, some elements of the model were rewritten in C#, while others were simply recompiled and wrapped as Windows DLLs. The generation of interoperable, standardised wrappers is a huge design issue for the ModelWeb, where the interfaces to models must be Web Service interfaces.

The welcome move towards standardised model interfaces again raises a paradox: the more widely applicable the interfaces become, allowing new models to easily plug into frameworks, the more abstract the descriptions of model inputs and outputs become, and the more the semantics specific to the discipline from which the model originates are hidden from the outside world. Thus a model with an OpenMI-

---

[18] http://www.toolkit.net.au/TIME last accessed 07/05/2011
[19] http://www.earthsystemmodeling.org/ last accessed 07/05/2011
[20] http://www.cerfacs.fr/3-25801-Technical-Reports.php last accessed 07/05/2011
[21] http://www.cesm.ucar.edu/ last accessed 07/05/2011
[22] http://www.cerfacs.fr/globc/PALM_WEB/ last accessed 07/05/2011
[23] http://intranet.cs.man.ac.uk/cnc/projects/bfg.php last accessed 07/05/2011
[24] http://www.genie.ac.uk/index.htm last accessed 07/05/2011
[25] http://www.quest-esm.ac.uk/ last accessed 07/05/2011

conformant interface may accept an 'object' which might be a multi-dimensional raster grid, a set of point observations or a time series. The nature and appropriateness of the object might be determined only when the data is parsed, so that the responsibility for finding and linking suitable inputs and outputs falls on the user.

Table 1: A variety of coupling technologies of varying granularity, with information on the specificity of their spatial data models, and their capacity for interoperating with other toolkits and technologies.

Approach	Language/s and service interfaces	Spatial data model	Interoperates with...
Kepler	Java, PMML, WSDL, BPEL, wrapped C/Fortran	None: responsibility for assessing the appropriateness of inputs devolves to the model.	R and Matlab, ImageJ, GRASS[26] and GDAL[27] for GIS
Taverna	Java, WSDL, REST, Beanshell, Rshell, Soaplab		R
Trident	C#, Java .NET, WSDL, BPEL		
Vis Trails	Python, WSDL		Quantum GIS
SME	STELLA[28] or SMML[29] (translated to C++ for execution)	Frames of Points which may also represent grids or network graphs	Python and CCA (through a Java-based portal)
ESMF	Fortran, C++	Raster grids	CCA
MCT	Fortran	Raster grids	CCA
Delta Shell	OpenMI	As below – internally, multi-dimensional results can be stored as NetCDF. Spatial vector data model is closely based on OGC features.	GDAL, Google Earth (through KML export)
OpenMI	C# or Java interfaces, wrapped C/Fortran	No explicit description: an input/output 'Object' may represent raster or vector data	Fluid Earth (see below) and Delta Shell (see above)
FRAMES	Native C interface with bindings for Java, .NET, Fortran, VB6 and Python	None: responsibility for assessing the appropriateness of inputs devolves to the model	PEST (as a tightly-coupled module)
Fluid Earth	OpenMI-wrapped models	As above	OpenMI (as a coupling mechanism)
TIME	.NET, wrapped C/Fortran	Raster grids, vector data, networks and time series.	
Tarsier	C++	Raster grids, networks, points and time series	

If information on the nature of model inputs is also published as standardised metadata or as an optional part of the model interface, the task of orchestration is far easier, and may even be automated. This is where the issue of ontology and semantics becomes important to supplement the abstraction enforced by technical interoperability, and to help in achieving context independence without losing access to vital domain knowledge. Rahman et al. (2004) note the importance of metadata which describes the 'properties and capabilities of ..[executable].. components', and specifically chose .NET introspection as the mechanism by which the TIME

[26] http://grass.osgeo.org/   last accessed 03/10/2011
[27] http://www.gdal.org/   last accessed 03/10/2011
[28] http://www.iseesystems.com/softwares/Education/StellaSoftware.aspx   last accessed 07/05/2011
[29] http://www.uvm.edu/~gundiee/AV/Spatial_Modeling_Book/3/smml/ last accessed 07/05/2011

framework would derive this information at runtime – a successful approach which has supported the development of a number of hydrological decision support systems (e.g., Argent et al., 2009), but one which places a language restriction on the developer. The ICMS, by contrast, derives this metadata at the point of model compilation, and stores it in a system-specific form. The XML metadata supplied to the BFG are used in a similar way, and are currently generated by hand, though a GUI to help with this task is planned. In all of these approaches, of course, reflection by the user as to the nature of their model and its requirements is necessary and indeed important; it is simply required at a different stage of the process and the information is encoded in a different way. The FRAMES environment tackles the semantic challenge by imposing a 'design by contract' approach where users subscribe to (or create) a domain ontology containing definitions of what models may produce or consume. Models conformant to the supplied dictionary may then be linked through a 'contract'. A proposal to standardise uncertainty information via similar dictionaries is described in Section 5.1.

Of particular interest in the Model Web context are the growing efforts to adapt the above model-coupling tools to comply with or use the OGC Web Processing Service (WPS) standard. (e.g., Guru et al., 2009, Jones et al., 2010, Pratt et al., 2010), which raise many pertinent questions about the abstract nature of OGC service specifications. Essentially, the flexibility of a WPS in accepting or producing any data, in more or less any format, can be problematic when a user who is querying the capabilities and interrogating the processes of that WPS lacks the semantic tools to understand the nature of the inputs and outputs. In these instances, profiling or restriction of the WPS so that it more clearly describes its limitations is extremely helpful in identifying whether that WPS really is a valid candidate for chaining with another. This restriction is most usually applied through reference to XML application schemata, and is fully anticipated in the WPS specification, which states "WPS can be thought of as an abstract model of a Web Service, for which profiles need to be developed to support use, and standardized to support interoperability". While this requirement is logical, it weakens the case for OGC services as 'interoperable' by imposing a requirement on users to develop specific clients to consume or chain these profiled WPS. If a model is to be usable within the Model Web, its interface must either conform to an agreed profile, or must be discovered and consumed by a higher-level 'broker' which has the capacity to translate the published model metadata into a usable format. A proposed solution (the *CaaS*) is described in Section 5.2.

Naturally, there is some metadata about models which can never be used in a fully automatic way. For example, information on the lineage and previous uses of the model, or on the circumstances and contexts to which it is best suited, may in the future be encoded in some sort of trust metric, but currently rely on a textual description and the judgement of the user. However, much of the necessary information on what a model will accept (even complex details such as required data granularity and valid geographical range) can be published using schemata and dictionaries, providing that these are widely accepted and available.

4.2 Exchanging uncertainty information between models

Interface abstraction and the profiling challenge are especially relevant when a model workflow is used to handle and propagate uncertainty. As described in Section 2, there are diverse sources of uncertainty which, even within a probabilistic framework, can be measured and recorded using different numerical summaries and metrics. When working in a multi-disciplinary context, the 'traditional' representations of uncertainty may also vary: for example, the Root-Mean-Squared Error values commonly attached to digital elevation models involve an implicit assumption that the error in elevation is symmetric (typically Normal) and identically distributed in space, while 95% confidence limits given by a sensor manufacturer for a measuring instrument may give no indication of how the expected error is distributed within that range, or whether it is biased or bounded. Some statistical summaries can be easily combined - for example, different probability density functions may be combined hierarchically in models to generate conditional probability density functions, and this approach underpins Bayesian analysis. Often, however, one representation of uncertainty (e.g., a sample) will require explicit transformation in order to be combined with another (e.g., a parameterised probability distribution) in direct computation.

A simple example of such a need for transformation in the Model Web would be as follows: A climate change scenario model is used to generate predictions of temperature for pixels in a geographical area. Each pixel is assigned an expected temperature with a statistical range in the form of a parameterised probability distribution function – this is assumed to be Normal, and so the outputs of the climate model take the form of a mean and variance for each pixel. These two output maps are to be fed to a second model, along with other maps to run 1000 agent-based simulations of animal dispersal. However, the second model requires a static temperature map as the base for each run, so a plausible realisation of temperature, with realistic spatial autocorrelation, must be generated from the statistical summary values, through some intermediate transformation service. Some of the aspects of the first model's outputs are implicitly described within the data (for example, geographical range, projection and resolution are easily extracted from a GML document or netCDF file). If the second model clearly advertises geographical / resolution requirements, this allows a user to assess or even automatically identify the need for resampling, aggregation or reprojection. Other attributes of the outputs, (such as lineage information on the climate model or the nature of the estimated uncertainty), require similar standard encodings. In particular, it must be clear that the type of uncertainty produced by the first model (a probability distribution function) is not the same as that required by the second model (a set of single-valued realisations for each pixel).

It can be seen from the above example that the requirement for models to fully describe themselves is of even more importance when it comes to handling uncertainty and propagating it through a workflow. While many models do not inherently handle uncertainty information on inputs, they can still be 'uncertainty-enabled' within a framework by repeated calls which effectively allow a Monte Carlo simulation or stochastic sensitivity analysis as described in Section 3.

4.3 'Uncertainty-enabled' models – current examples

Many simulation software packages exist which may be used to 'uncertainty-enable' existing models. Mattot et al (2009) present a very useful review of 65 different tools for simulation, calibration, optimisation and model evaluation, and a number of the most widely used or pertinent, with example applications, are listed in Table 2. A further list of freely available software tools for the development of uncertainty management applications is given in Table 3. The most successful and widely-used 'uncertainty-enablers' are model-independent, and sometime platform-independent; this flexibility is generally achieved by a reliance on ASCII-formatted inputs to and outputs from the wrapped models. This tradeoff between flexibility and restriction is an equally important theme for the Model Web, as discussed in section 4.1. The process of 'uncertainty-enabling' can demand significant effort on the part of the user (for example, through the generation of template and instruction files to link and feed models, or through the conversion of binary model outputs to ASCII formats) but in other cases, simulation tools are made easily accessible as modules within existing frameworks. If consideration of uncertainty and validation of linked models is to become routine, especially among non-expert users, this access to powerful simulation tools which can wrap and 'uncertainty-enable' models must be strengthened and improved. Recent moves in this direction include the integration of the PEST parameter estimation toolkit[30] into FRAMES (Castleton and Meyer, 2009). This is an important issue for the Model Web, and a clear opportunity to build on the interest and experience within the wider community of integrated modellers. Visualisation of the spatio-temporal uncertainty of outputs (more fully discussed in Section 5.3.4) is available to varying extents in these solutions, and is also extremely important in the presentation and use of propagated uncertainties.

In many of the above examples, models designed to accept single input values at each observation point are wrapped and run multiple times with stochastically-generated input values derived from the uncertainty specification on these inputs. In other words, though multiple outputs from these models may be summarised to *produce* uncertainty information such as probability distributions, they do not explicitly *accept* such uncertainty information on the inputs. Other models, in contrast, may accept statistical summaries such as standard deviations, ranges or quantiles and use this uncertainty information internally in their calculations. In other words, provided that they can understand the form in which the input uncertainty is encoded, these models are already 'uncertainty-enabled'. One such example is the INTAMAP interpolation Web Service[31], which can consume point observations whose uncertainty is represented as parameter values for well-known distributions, and produce interpolated maps of mean and variance using an algorithm most suited to the nature of the input uncertainty (Pebesma et al., 2011). In an alternative approach which combines numerical and semantic elements of uncertainty, the 'EcoPath' model commonly used in fishery management planning (Pauly et al, 2000) elicits estimates of input uncertainty from users through the assignment of 'pedigrees', recording the lineage of the data (a guess, a global estimate, a measurement) as well as allowing the

---

[30] http://www.pesthomepage.org/home.php   last accessed 03/10/2011
[31] http://www.intamap.org/ last accessed 07/05/2011

user to select confidence intervals. The two elements of the uncertainty are combined to perform potentially complex analyses, for example using the Bayesian 'Ecoranger' module.

Many of the existing frameworks and model implementations tend to address very specific application domains and focus largely on model calibration and parameter estimation; this makes them highly valuable for handling specialised and complex data and algorithms within a research field, but can raise challenges for multidisciplinary model chaining. A number of the existing approaches support uncertainty propagation through Monte Carlo methods, which have proven value for sensitivity and uncertainty analysis. In the following section we develop a generic framework for managing uncertainty in the Model Web context, informed by the lessons learnt in previous work.

Table 2: A selection of existing simulation software packages (or modules within frameworks) which may be used to uncertainty-enable existing models.

Name	Reference(s) or web sites	Comments	Examples of use
PEST (Parameter EStimation Toolkit)	Doherty, (2004)	Powerful calibration, regularization and optimization toolkit. Implements a variety of parameter estimation methods, and null-space Monte-Carlo approaches for linear and non-linear analysis of uncertainty, parameter identifiability and error variance.	1 Castleton and Meyer (2009); integration of PEST into FRAMES. Dausman et al.(2010); testing of alternative hypotheses for the wastewater plume movement, by highly-parallellised calibration of candidate models and generation of a subset of 'superparameters'. 1 Doherty & Hunt (2009); describe statistics (calculated using PEST) to summarise the extent to which each parameter of a model can be identified, and the extent to which the calibration process can improve on the estimate based on prior expert knowledge.
UCODE	Poeter et al. (2005)	Non-linear parameter estimation code – like OSTRICH and PEST, generates confidence intervals and other statistics through model inversion,	1 Kelson et al. (2002); application to a mine hydrological context, resulting in a highly simplified model with equivalent predictive power. 1 Foglia et al., (2009); refinement of parameters from catchment-scale estimates for calibration of distributed hydrological models.
OSTRICH	http://www.civil.uwaterloo.ca/ lsmatott/Ostrich/OstrichMain.html	A versatile tool incorporating a diverse set of algorithms for calibration, optimization and computation of statistics such as parameter correlation / sensitivity, and observation influence.	1Rabideau et al. (2005); calibration of multiple AEM groundwater flow models, with particular attention to effects of model precision and observation location. Mattot & Rabideau (2008); describe a method (in OSTRICH 1.8) for simultaneous calibration of equally plausible models by adaptive weighting and mapping of parameters between reference and surrogate models.
UNCSAM	Janssen et al., (1994)	Can do model emulation; does not cope with spatially and/or temporally correlated variables	1 Bärlund and Tattari (2001); application to the ICECREAM model of field phosphorus loss.
SME (Spatial Modelling Environment)	http://www.uvm.edu/giee/IDEAS/ sme/docs/SME_guide.html		Voinov et al., (1999); ecological-economic spatial process modelling. Villa and Costanza, (2000); spatial agent-based modeling (enabled by linking SME with the SWARM agent-based modeling toolkit [32]) Deal and Schunk (2004); scenario modelling of urban sprawl and its effects; particular attention to the importance of model validation.
GENIE-1	Holden et al., (2010)	Emulation based on ensemble modelling.	1 Holden et al (2010); climate prediction

---

[32] http://www.swarm.org   last accessed 03/10/2011

FRAMES Sensitivity/ Uncertainty module	http://mepas.pnl.gov/framesv1/ sum3ug.stm	Monte Carlo analysis and Latin hypercube sampling. User supplies parametric distributions for input uncertainty (currently uniform, log-uniform, Normal, or log-Normal)	Babendreier and Castleton (2002); parallelised use in the 3MRA pollutant fate model 1 Castleton et al. (2006); linked FRAMES with R to calculate & visualize impacts of input uncertainty
TIME		User supplies parametric distributions as above. Some visualization of uncertainty (e.g., confidence limits on outputs).	Rahman et al. (2005); incorporation of a Stochastic Climate Library (SCL) into TIME
SoftIAM	http://www.tyndall.ac.uk/sites/ default/files/tr51.pdf	Allows Latin hypercube sampling from Normal, log-Normal, uniform, triangular Beta or Davies* distributions (*specifically for risk assessment)	1 Warren et al. (2008); SoftIAM used as an interface to BFG for climate modelling
WADES	http://www.ceh.ac.uk/ sci_programmes/Water/ Wades_Project/index.html	Work in progress - aims to assess the relative costs and benefits of OpenMI wrappers for integrated modelling.	
UNCSIM	Reichert, 2006	Systems analysis toolbox used to link simulators though text input / output files. Supports maximum likelihood parameter estimation & sampling from a variety of multivariate distributions.	Arnold et al., (1998); Soil and Water AssessmentTool (SWAT), watershed-scale hydrological /water quality simulation; Hutson and Wagenet,(1991); simulation of nitrogen dynamics in soil.
DUE (Data Uncertainty Engine)	Brown and Heuvelink, 2007	Quantification of positional and attribute uncertainty in environmental data by probability distributions that take spatial and temporal correlations into account. Can also sample from these distributions for Monte Carlo uncertainty propagation analyses	Refsgaard et al. (2006b); hydrologic river basin modelling, handling changes of scale De Bruin et al. (2008); positional uncertainty in agricultural field boundaries for use in precision farming
Crystal Ball	Oracle, (2011)	Spreadsheet based	1 Dubus et al. (2002); pesticide models
@RISK	Palisade (2011)		1 Dubus et al. (2002); pesticide models

1 Rank input parameter contribution to overall uncertainty

Table 3: Programming tools for development of uncertainty software.

Tool	Reference / website	Comments	Examples of use / case studies
R	http://www.r-project.org/	Open source application with a wide selection of statistical /modelling libraries including some spatio-temporal functions.	Langford et al. (2009); assessment of susceptibilities of conservation planning algorithms to input uncertainty. Output uncertainties visualised using R plotting functions, as statistical summary plots.
			'Sensitivity' (G. Pujol - http://cran.r-project.org/web/packages/sensitivity/) .A freely-available R package containing a collection of functions for factor screening and global sensitivity analysis of model output.
python	http://www.python.org/	Another library-based language with many mathematical and spatial modules.	ModelBuilder (F. Coelho - http://model-builder.sourceforge.net/) - graphical tool for simulating models based on ordinary differential equations
SimLab	http://simlab.jrc.ec.europa.eu/docs/html/main.html	Development framework specifically for sensitivity/uncertainty analysis - supports global methods only	Le Maire et al,(2011); One of many studies which employ methods implemented in SimLab (in this case, the FAST technique) for Monte Carlo estimation of uncertainties and parameter effects.
DAKOTA	http://dakota.sandia.gov/index.html	Toolkit which implements numerous algorithms for optimisation, experimental design and uncertainty quantification.	Eldred et al (2011); describes methods (incorporated in DAKOTA) for separating and nesting sampling based on epistemic and aleatory uncertainties, combining local and global gradient-based optimisations.
JUPITER API	http://water.usgs.gov/software/JupiterApi/	Platform for developing model analysis applications (e.g., UCODE, MMA) with many built-in algorithms.	Banta et al. (2008) Description of how the API can be used as a platform for fast testing and prototyping, particularly where weighting of prior information and specification of correlated errors are required.

5 A proposal for an Uncertainty-Enabled Model Web

The challenges raised by the transition from isolated data sets and models deployed on individual computers, to Web-deployed data sets and models with well defined and widely understood interfaces and information models cut across a wide range of issues. The management of uncertainty in the Model Web is one of these challenges. Within the UncertWeb project a range of tools are being developed to support the assessment of uncertainty using expert elicitation (Section 5.3.1), the aggregation and disaggregation of spatial and temporal fields (Section 5.3.2), efficient uncertainty and sensitivity analysis methods (Section 5.3.3) and the visualisation of uncertain variables (Section 5.3.4). These tools are key drivers in promoting uncertainty management in the Model Web, and they all use the UncertML standard described below for their communication.

5.1 Representing uncertainty interoperably: UncertML

UncertWeb adopts a probabilistic approach to representing uncertainty. As a conceptual information model for representing probabilistic uncertainty, the UncertML[33] language was devised within the INTAMAP project (Pebesma et al., 2011) to describe random quantities. Version 1.0 of UncertML was a weak- typed design (Williams et al., 2009), with strong dependencies on Geography Markup Language (GML) and extensive use of the Sensor Web Enablement (SWE) standards (OGC 08-094r1, 2004). UncertML 2.0, released in February 2011 and developed within the UncertWeb project, is a simpler hard-typed design. This reduces flexibility but allows more complete interoperability, since software providers can actually claim their software supports UncertML 2.0. Hard-typing also permits the implementation of an Application Programming Interface (API) that supports encoding and decoding of XML and JSON documents.

The conceptual model for UncertML 2.0 is very simple. A basic abstract uncertainty type is specialised to create Distributions (probability distribution functions, including mixture models for multi-modal distributions), Statistics (summary statistics, such as moments), and Samples (realisations of random variables). These types correspond to those described in Section 2.2, and allow UncertML 2.0 to represent uncertainty very flexibly. Where possible the more complete description of a probability distribution is preferred. UncertML consists of a dictionary to precisely define the semantics of the uncertainty elements, and can encode both univariate and multivariate random quantities.

Since the inputs and outputs of complex environmental models are usually spatio-temporal, a standard way to integrate UncertML with spatio-temporal data is required. UncertML separates concerns by focussing purely on uncertainty, and is designed to be used with other standards such as Observations and Measurements (O&M - a common XML encoding for exchanging observations in the Web) (Cox, 2007), which can be used to define the variables being considered, the sampling or model output locations etc. The UncertWeb O&M (O&M-U) profile restricts the O&M specification to permit only certain geometries and time units. This tight profiling

---

[33] http://www.uncertml.org/ last accessed 07/05/2011

solves many of the abstraction problems described in Section 4.1. Uncertainty can be added to an O&M document in two ways: (1) uncertainty can be added as additional quality information to the result, or (2) the result itself can be encoded as an uncertain value. In both cases, UncertML is used to model and encode the uncertainties.

While O&M is well-suited to observations with spatial vector geometries, grid-based observations are better encoded using the Network Common Data Format (NetCDF), an established format for exchanging multi-dimensional gridded environmental data. Thus an uncertainty-enabled NetCDF profile (NetCDF-U) has also been developed within UncertWeb. As a first step, the UncertML dictionary is used to define the variables that contain the uncertainty values. In the longer term, basic data types for uncertainty will be defined in the common data model on which NetCDF is based.

UncertML plays a central role in uncertainty-enabling the Model Web. It is the primary mechanism for communicating uncertainty between Web Services (which, in the Model Web context, act as model interfaces). Existing standards for communicating within Web based systems, (for example the Open Geospatial Consortium series of standards) already have some support for uncertainty in the ISO19139 compliant data quality measures (ISO19139, 2007). However, most of the measures that are defined for such quality indicators (ISO19138, 2006; ISO19157, 2011) are not very generic and relevant only to very specialised domains; for example, there is no method for representing a probability distribution. Another issue with existing encodings is that most models follow the "result" and associated "result quality" pattern. For models with model structure uncertainty there is no notion of a unique result; rather the result itself is uncertain. Thus it would not be very natural to always encode uncertainty in the result quality, because this begs the question of what to put in the result? One option might be the mean, or expected value, but in some situations, (for example where the model predicts the outputs to be in two or more plausible states) the mean can be a very misleading, and indeed, improbable result.

UncertML addresses the deficiencies in data quality standards by providing a standardized way to encode quantified uncertainty such as probability distributions. Using the O&M-U profile, model results with spatial vector geometries can now be provided encoded as uncertainties (rather than as values with associated uncertainty) or with additional metadata about the model result. Similarly, the NetCDF-U profile allows for providing uncertain gridded model results in a standardized and meaningful way.

5.2   The UncertWeb architecture

The UncertWeb framework aims to support uncertainty in the discovery, access and chaining of data sets and models, while keeping in line with the Model Web principles. The design of the UncertWeb framework architecture is based on the following principles:

i) Re-use of existing tools: Several initiatives and projects at national, regional, European and global level already provide resources and tools that could be useful for the development of an UncertWeb framework. The adoption of a Service-Oriented-

Architecture (SOA) style (Erl, 2005) allows us to integrate such heterogeneous components.

ii) Extension of the Service-Oriented-Architecture (SOA): During the last decade, the SOA style, successfully adopted in different contexts such as e-Business and e-Government (IDABC, 2004), has also been adopted in the development of Web-based geospatial resource sharing systems (OGC, 2002). However, in the development of the global System-of-Systems (SoS), SOA has limitations due to the growing complexity of the overall system. Solutions based on the introduction of specific components (brokers) which act as mediators will help to lower the entry barrier for users (Nativi and Bigagli, 2009);

iii) Multiple solutions for uncertainty representation: From a conceptual perspective all data should be treated as uncertain. However, it must be acknowledged that almost all existing data resources are not treated in this way. Most data sets come simply as a series of values, often without any uncertainty information. Therefore the UncertWeb system architecture needs to accommodate both kinds of representation: a) data sets with uncertain values (e.g. expressed as a probability distribution); b) data sets with certain values and associated uncertainty information (e.g. expressed as accuracy metadata).

Figure 2 depicts the UncertWeb architecture in terms of high-level entities, which can be categorised as four high-level packages, and a broker component; package dependency is reported through directed arrows. Packages are as follows:.

- A *GUI* package that includes all the components handling user interaction.
- An *Uncertainty Tools* package that includes all the components and applications for uncertainty management, such as elicitation and visualization.
- An *Available Services* package that collects all the services exposed in the UncertWeb system. It includes the typical geospatial functionalities:
    - *Data View*: for presentation and portrayal of data sets;
    - *Data Access*: for accessing data sets for further evaluation and use;
    - *Data Catalog*: to register and find data sets based on their metadata;
    - *Data Publishing*: to provide a persistence layer for data sets and results.
    - *Data Transformation*: to process and manipulate data sets. For the UncertWeb purposes, general data transformations are further classified as:
        - *Data Processing*: information extraction and processing, by data set aggregation, operation of models on inputs, etc.
        - *Data Conversion*: transformation without information extraction, for example change of format or change of coordinate reference system.
        - *Uncertainty Transformation*: transforming the representation of data set uncertainty.
- A *Data Types* package that includes all the specification and tools for managing uncertainty-enabled data types. These include profiles such as the O&M profile described in Section 5.1.

- A Composition-as-a-Service (*CaaS*) component which is controlled through the GUI and gives access to all the available services, adopting an extended-SOA approach (broker-based mediation).

Figure 2 Dependency view of the main UncertWeb architectural components.
Figure 2 shows that the *Available Services* act upon data sets expressed according to the available *Data Types*. The components in the *GUI* package may also access the *Uncertainty Tools*. On the other hand the GUI needs to access the services in the *Available Services* package. Due to the heterogeneity of the services in terms of interfaces, metadata and data models, a direct link would impose a great complexity on the GUI components, limiting their usability and consequently the scalability of the overall system. Therefore according to the extended SOA approach, a specific service broker component called the CaaS (Composition-as-a-Service) component is introduced to mediate the interactions between the user and the services. The two main tasks of the CaaS are service composition and the publication of workflows as services.

This high-level view helps to highlight some important points:
- UncertWeb provides different resources: Tools (e.g. the *Uncertainty Tools*), Services (i.e. the *Available Services*) and Data (i.e. the *Data Types*).
- The *CaaS* plays a central role, being the component that harmonizes the access to the *Available Services* (and indirectly to the *Data Types*).
The architecture is being further developed and will be more completely described in future publications.

5.3 UncertWeb tools

The UncertWeb framework can only work if the tools needed to analyse how uncertainty propagates through model workflows and to communicate and visualise the resulting uncertainties have been implemented properly. Since UncertWeb currently uses a Monte Carlo simulation approach for uncertainty propagation, all that is required for the actual uncertainty propagation analysis is a computational loop around the models as described in Section 3. The tools described here will support the Model Web's capabilities in the quantification of uncertainties in inputs and models within service chains, the spatio-temporal aggregation and disaggregation of uncertain variables, the analysis of uncertainty and stochastic sensitivity, and the communication of output uncertainty to end users and decision makers. These tools will prove to be critical in achieving impact across the environmental and, more broadly, the applied science user communities, providing a suite of services and applications to make the uncertainty-enabled Model Web easy to use.

5.3.1 Expert Elicitation

Keeping track of uncertainties in service chains implies that the uncertainties about the input data submitted to the chain and the uncertainties associated with the models used in the chain are known. In many cases these uncertainties can be derived - for example, from the precision of measurement devices, goodness-of-fit of regression equations or from statistical sampling error - but sometimes the uncertainty must be

derived from expert judgement. Expert elicitation is a systematic process of formalising and quantifying expert judgements about uncertain quantities, typically in probabilistic terms.

Since the first development of structured expert-opinion elicitation by the RAND Corporation in the 1940s (Cooke, 1991), formal expert elicitation has gradually become a mature research field. Recently, expert elicitation has attracted more attention from statisticians and experts in uncertainty analysis (O'Hagan, 1998; Cooke and Goossens, 2000; Meyer and Booker, 2001; O'Hagan et al., 2006). Uncertainty about quantities elicited from experts is encoded in the form of a probability distribution function. The two statistical frameworks commonly used for this purpose are parametric fitting and nonparametric fitting. The former fits expert judgments to standard parametric families of distributions and is the method used in UncertWeb. In this approach, quantiles of the distribution such as the median and first and third quartiles are elicited from the expert using a formal procedure, after which the most appropriate shape of the probability distribution is selected automatically and the associated parameters are estimated.

When multiple experts are involved in the elicitation process, a combination of expert judgements is needed to utilize knowledge from all experts. Interaction among experts is not needed when using mathematical aggregation (O'Hagan et al. 2006). In contrast, behavioural aggregation requires some degree of interaction amongst experts. UncertWeb approaches experts through the Web, which complicates interaction between experts. Hence, a mathematical aggregation of the experts' opinions is used.

Figure 3. A screen capture of the Elicitator showing the creation of an elicitation problem.

Figure 4. A screen capture of the Elicitator showing the expert elicitation interface.

The implementation of expert elicitation in UncertWeb is provided by the Elicitator[34]. It largely builds on the existing SHELF methodology developed by Oakley and O'Hagan (2010). One major extension is that the entire process is Web-based. It involves a problem owner who defines their problem, provides background information and selects experts. Figure 3 shows an example of the initial Webpage viewed by the problem owner. Once experts are selected, they are notified and can perform the elicitation independently behind their own computer, at any suitable time and at their own pace. Figure 4 shows a typical Web-based form that the expert would need to fill in. Results are communicated to the expert and provisions for reconsidering and changing earlier judgements are provided. Once all experts have submitted their opinion these are aggregated by the tool and the resulting probability distribution is stored in UncertML (Figure 5). All stages of the problem are recorded, so that the lineage of the elicitation is fully accessible to the problem owner, and could potentially also be inserted in a published workflow to support reproducible science.

Figure 5. A screen capture of the Elicitator showing the pooling of expert judgments.

---

[34] http://elicitator.uncertweb.org/ last accessed 8/5/11

The Elicitator facilitates the elicitation of both numerically continuous and categorical variables. In addition, it also supports the elicitation of spatially distributed continuous variables, by providing a tool to estimate the semivariogram.

5.3.2 Spatio-temporal aggregation and disaggregation

Aggregation and disaggregation are common operations or computational components inside environmental models. For instance, hydrological models may aggregate rain when they compute river discharge from spatio-temporally distributed rainfall values. Alternatively, they may predict spatially distributed soil moisture content from catchment average (aggregated) precipitation. Outside the modelling context, aggregation and disaggregation is required when the spatial, temporal or spatio-temporal resolution (or support) of the model input or output does not match the resolution required at the next stage of processing. This sort of functionality is typically found in model couplers, such as the OASIS3 coupler (Valcke, 2006) but the uncertainty introduced by the aggregation/disaggregation process is not estimated by these couplers.

A very common case is that of time series data. When rainfall data are available on a daily basis, but a model requires data on a monthly basis, the time series can be temporally aggregated. By spatio-temporal aggregation we mean the computation of a single value from a set of (spatially or temporally or spatio-temporally) contiguous values, for one or more of these sets. The aggregation involves the application of an aggregation *function*, such as the mean, median, maximum, 95-percentile, or variance. Spatio-temporal disaggregation is the reverse process: from one or more aggregated values, one or more sets of values for smaller spatial, temporal or spatio-temporal units are generated. Other words used for these processes are upscaling (aggregation) or downscaling (disaggregation). Typically, aggregation is a relatively simple activity when a simple function can be applied such as taking the average value over a number of grid cells and will reduce uncertainty. More complex forms can involve techniques such as block kriging. Disaggregation typically involves more modelling, and requires ancillary information about the phenomenon that is not available from the aggregated data alone (Bierkens et al., 2000) and will increase the uncertainty in the smaller scales.

Because spatio-temporal aggregation and disaggregation are commonly-required activities when model chains are formed, the UncertWeb tools will include a generic Web Service for spatial, temporal or spatio-temporal aggregation. It will only work with Monte Carlo samples, and for each sample element will aggregate the values to a new spatio-temporal resolution (Heuvelink and Pebesma, 1999). Disaggregation will be implemented prototypically, for a very limited set of cases, using the area-to-point kriging technique.

5.3.3 Uncertainty and sensitivity analysis

When models are exposed on the Web in a discoverable manner, users will not necessarily be familiar with the detail of the models, and their response to inputs. The problem becomes even worse when models are composed in workflows, where they might have rather unexpected behaviour, due to the interactions of the different components in the workflow, that needs to be characterised and understood by the

users. One way to address this is by undertaking uncertainty or sensitivity analysis (Oakley and O'Hagan, 2004).

Uncertainty analysis involves describing the distribution of the outputs given a particular distribution on the inputs, which might include some of the inputs being fixed, i.e., assumed to be perfectly known. Uncertainty analysis is typically achieved using Monte Carlo techniques although screening methods as proposed in Morris (1991), or other local methods (Hill and Tiedeman, 2007) can also provide useful insights into the model response. Sensitivity analysis involves understanding the model's response to variation in inputs, and can take many forms, including local methods based on derivatives (Hill and Tiedeman, 2007) and global methods, based on variance (Saltelli et al, 2010). Variance-based sensitivity analysis is generally regarded as being more useful, since it allows users to apportion the proportion of variance in the output distribution explained by inputs, and their interactions over the whole of the realistic input space. It is however necessary to acknowledge that local sensitivity analysis methods based on locally linearising the model, while potentially susceptible to errors that arise from strong non-linearities in model response can prove useful in complex models as part of an exploratory analysis (Campolongo et al., 2007) and can be used, for example, to assess the ability of observations to inform parameters (Foglia et al., 2009) and predictions (Tiedeman et al., 2004, Water Resources Research; Moore and Doherty, 2005 WRR; Tonkin et al., 2007, USGS report).

A problem with the Monte Carlo methods used for variance based sensitivity analysis, particularly when applied to models with large run times, is the time required to undertake such an analysis (O'Hagan, 2011). One possible means to address computational costs is to employ emulator technology (Shahsavani and Grimvall, 2011). Emulation involves creating a statistical surrogate model of the underlying model. The surrogate mode is fast to evaluate and can be used in place of the original model as long as the additional 'emulator uncertainty' is accounted for in its usage. The ongoing MUCM project[35] has developed emulation techniques so they can be applied to a wider range of models although limitations remain (see the MUCM toolkit[36] for details, and references therein).

At present emulation methods are most effective when a small number of outputs are being considered, which are real-valued. Very large numbers of outputs require multivariate emulation, which entails describing a complex high dimensional conditional probability distribution. Discrete valued variables require further development of the emulation theory which is mainly based on Gaussian processes (e.g. O'Hagan, 2006) thus assumes a continuously (and often smoothly) varying continuous valued output. Typically one considers emulation for a small number of summary outputs, which might be some combination of all the model outputs, for example the average temperature over a region, or the proportion of a given land use type in a given area over a given time period. If it is necessary then it is possible to build multi-output emulators (Urban and Fricker, 2010; Conti and O'Hagan, 2010) however these are generally rather complex and often it is better to build many very accurate individual emulators – these will capture the joint response of the outputs,

just not the joint emulator uncertainty on the outputs, which will typically be very small by design.

Building an emulator for a computer model, whether exposed as a workflow, a Web Service or a single machine executable is a complex process. There are several steps each involving complex judgements, ideally informed by the model owners / builders. The main steps in constructing an emulator for a given workflow, also depicted in Figure 6, are:

- Elicit input ranges and uncertainties – often uncertainties on model inputs will not be known, making it necessary to elicit expert beliefs about the values of these inputs. This task is supported by elicitation methods discussed in Section **Error! Reference source not found.**.
- (Optionally) find important inputs – using a process known as screening (Morris, 1991), identify inputs that have significant effect on the model output(s) of interest. Identifying inconsequential inputs allows the reduction of the dimension of the input space, thus having a positive effect on emulator complexity and training efficiency.
- Design the training set – with a sampling method such as Latin Hypercube (Santner et al., 2003), a set of points to cover the input space is generated. The model is then run at these points, producing a training set of input-output pairs.
- Train the emulator – an emulator is typically a Gaussian process consisting of a mean function, covariance function, and a set of parameters. Training typically employs Bayesian inference (O'Hagan, 2006). Once an emulator is trained, it can be saved in a portable format such as XML or JSON.
- Validate the emulator – validation is essential to ensure the probabilistic judgements represented are correct (Bastos and O'Hagan, 2008). If validation results are unacceptable, parameters can be adjusted and training can be restarted.
- Use the emulator – the emulator can be used for uncertainty analysis, sensitivity analysis, calibration, forecasting and decision making (O'Hagan, 2006). In the uncertainty-enabled Model Web it will be possible to use this emulator as you would any other model component.

Figure 6. The main steps in constructing an emulator. For each stage, including the optional screening step, the methods supported within UncertWeb are listed.

Several practical implementation issues are addressed when developing a user driven tool to construct an emulator for a Model Web component.

- The tool must be able to read descriptions of model inputs and outputs, and perform runs for training and validation. For this to be possible without requiring specific code, each Web-enabled model must be exposed in a standard way. As the tool will be building service requests, this includes any inputs and outputs. The service and information model profiles developed within UncertWeb aim to facilitate this interoperability. Screening, training and validating an emulator requires several hundred runs of the model. If this model takes minutes or even hours to compute, it is impractical to require a user to keep a web-based program running for this time. Moving this responsibility to the server allows the tasks to be run independently of client

state, but also introduces resource management and reliability issues. Mechanisms for asynchronous execution have been developed to queue tasks if resources are unavailable, and resume tasks in the event of system failure.

- Constructing an emulator requires several choices to be made. Some of these choices can be set to default values, and some of those default values may be changed by expert users. Providing this vast array of options leads to usability challenges. A user could be overwhelmed if they are required to make too many choices, or an expert user may not feel as though they have enough control over the construction process.

In theory it is possible to greatly speed up the computation of uncertainty and sensitivity analysis using emulators, and they are likely to prove particularly effective in distributed modelling frameworks where the emulators can be invoked just as any other model component, but can also be transported easily as JSON or XML, and run almost instantly. Of course not all models will be amenable to emulation, and it is envisaged that often it will be more appropriate to emulate a workflow linking several model components directly, not just single model instances. Several classes of model are currently not amenable to emulation, including models with discrete valued outputs, models with discontinuous outputs (with respect to variation in inputs) and models with very large numbers of inputs and outputs. Such models require very case specific emulation methods which are not currently supported within UncertWeb.

It is not always clear that construction of an emulator can be justified. Extensive examples of the successful utilisation of emulators in modelling studies can be found, for example, in the references in Kennedy et al. (2006). However the construction of an emulator itself is expensive, requiring many model runs to be made. There will be cases where the expense of constructing an emulator cannot be justified, but this will also mean that a formal uncertainty or variance based sensitivity analysis will be rendered impossible (very large / slow to evaluate model, in which case local methods could be considered) or trivial (very small / fast to evaluate model). We do not imagine that emulators will be appropriate or required for all workflows, but we do anticipate that they will in some cases enhance our ability to characterise and use models.

5.3.4 Visualisation

Communication and visualisation of results and associated uncertainties produced by UncertWeb requires a systematic approach that incorporates contributions from cognitive science as well as statistics. A large body of literature and methods are available (e.g., Wittenbrink et al., 1996; Pang et al., 1997; MacEachran et al. 2005; Kardos et al., 2007; van de Kassteele and Velders, 2006; Garlandini and Fabrikant, 2009; Wood et al., 2009) and existing state-of-the-art techniques will therefore be implemented in the UncertWeb visualisation tool.

For visualisation of uncertainty it is sensible to distinguish between uncertain phenomena that are measured on a continuous-numerical scale (e.g. precipitation, concentration of air pollutants, income per head) and those measured on a categorical scale, whose groupings may have no numerical meaning or ranking (e.g. soil type, land cover, age group). Also, it matters greatly whether the phenomenon varies in

space, in time, in both space and time, or is constant in space and time (Heuvelink et al., 2007). Hence, the various uncertainty visualisations can be conveniently presented in a two-dimensional table. The main techniques that will be implemented in UncertWeb are given in Table 4.

For phenomena that vary neither in space nor in time, standard presentation formats such as boxplots, pie charts and graphs of the probability distribution (e.g. the jstat Javascript library[37]) will be used. For spatially distributed phenomena, either static displays with adjacent maps or animations of realisations will be provided. The option to mask or whiten parts of the study area that are too uncertain will also be provided. A very useful option is for Web-based interactive visualisation to allow the user to select locations at which visualisation techniques developed for non-spatial and non-temporal variables can be applied. Uncertainty in dynamic variables can be displayed similarly to uncertainty in spatial variables, with additional possibilities, such as displaying multiple realisations against time in a single figure. This method does not apply to uncertain spatial variables, but in that case multiple realisations can be shown in animation mode. Finally, for space-time phenomena the only feasible options are to let users select locations or time points (slices) to which any of the uncertainty visualisation techniques for spatial or temporal variables will then be applied.

Table 4. Types of uncertainty visualisation tools implemented in UncertWeb for combinations of measurement scale and space-time variability.

	I. continuous numerical	II. categorical
A. non-spatial, non-temporal	1. graph of the probability density or cumulative distribution (e.g., Figure 5) 2. error bar, interquartile range, confidence interval, box plot	1. graph of probability distribution 2. pie chart, stacked bars, bar chart
B. spatial	1. adjacent maps of the mean and standard deviation;adjacent maps of the lower and upper limits of a confidence interval 2. maps of multiple realisations (draws from the probability distribution) in one frame 3. masking or whitening of areas with large uncertainty 4. interactive facility to apply techniques from category A1 at selected point locations in map 5. animations of realisations	1. adjacent maps of the category with maximum probability and the associated probability 2. map of the category with maximum probability but masked, whitened or blinking when the probability is below an (interactive) threshold 3. entropy map 4. interactive facility to apply techniques from category A2 at selected point locations in map
C. temporal	1. graphs of mean, lower and upper limits of confidence interval, or error bars against time 2. multiple realisations plotted in one figure 3. interactive facility to apply techniques from category B1 at selected time points	1. graph of category with maximum probability but masked, whitened or blinking when the probability is below an (interactive) threshold 2. graph of entropy 3. interactive facility to apply techniques from category B2 at selected time points

---

[37] http://www.jstat.org/ last accessed 8/5/11

| D. spatio-temporal | 1. interactive facility to apply techniques from category B1 at time points | 1. interactive facility to apply techniques from category B2 at time points |
| | 2. interactive facility to apply techniques from category C1 at point locations in map | 2. interactive facility to apply techniques from category C2 at point locations in map |

Overall, the set of tools being developed in UncertWeb and explained in Section 5.3 describe a minimal requirement for tools that should be available to a modelling framework which claims to provide practical uncertainty support.

6 Discussion and conclusions

Existing modelling frameworks address uncertainty to variable extents, and there are important lessons for the Model Web: for example, easy-to-use tools which clearly describe their assumptions and requirements can encourage users to assess, record and use uncertainty information at *all* stages of the modelling process. Reliable communication of uncertainty information between diverse models, across disciplines, will avoid the bottlenecks where rich and useful uncertainty information can be lost to the decision maker. Tools to assist with the visualisation of uncertainty can help to communicate this information to a range of different users whose needs may be very different (Davis and Keller, 1997).

Existing modelling frameworks also have varying levels of interoperability, with some being strongly specialised to specific domains, or specific types of problem, for example time-stepping models in the OpenMI framework. Interoperability itself is a complex topic and can in any case be achieved at a number of methods. We would argue that one might consider the following levels of interoperability:

1. Machine encoding interoperability – a common underlying representation of basic data values, e.g. big-endian or byte order assumptions, often IEEE standards based.
2. Format encoding interoperability – use of a common data format which specifies, for example, header structure or the order of elements, delimiters and tags. Examples are NetCDF, GML and O&M application schema, shapefile, etc.
3. Semantic dictionary interoperability - understanding of the meaning of the data values, based on semantics / ontologies, for example RDF / OWL. This is 'hardwired' semantics via a dictionary.
4. Semantic machine interoperability - the real goal of semantic integration where machines can 'understand' concepts and reason with them, typically without resorting to a central controlled vocabulary.
5. Information interoperability – here, the relation of the data to reality is quantified so that the data can be used appropriately in a given application. At present this is little addressed.

To achieve information interoperability it is necessary to quantify the information (or uncertainty) in all aspects of the modelling operation. No existing modelling frameworks provide a complete solution to managing uncertainty. We contend that information interoperability, i.e., the ability not just to share data and models, but to actually base rational decisions and policy on the outcomes from these integrated

modelling frameworks, requires a rigorous and consistent definition of uncertainty and a framework that can manage this from end to end.

When addressing uncertainty, a probabilistic approach seems most natural (Dawid, 2004; O'Hagan, 2011) although other approaches such as fuzzy set theory and Generalised Likelihood Uncertainty Estimation (Beven and Freer, 2001) are also applied. Other coherent frameworks for managing uncertainty, for example Bayes Linear (Goldstein and Wooff, 2007), and imprecise probability (Reichert, 1997) also deserve attention. These frameworks have attractive features, in that they require fewer assumptions to be used (for example Bayes Linear methods work with expectation, not full probability distributions), but then enable one to make weaker statements as a result (since one only has expectations, including some higher order judgements).

All quantitative approaches also have limitations; 'unknown unknowns' will always require a qualitative treatment, and in simulations of social systems the issue of human choice and free will make modelling particularly challenging, and uncertainties still more challenging to quantify. Even for environmental models of systems that are reasonably well understood, for example the Earth's atmosphere, obtaining reliable uncertainty estimates for inputs and model structure uncertainty is an open research problem. Expert elicitation can assist in the determination of subjective uncertainty on unobserved model inputs and more rigorous and precise uncertainty estimation for observations can assist in characterising the uncertainty on other inputs. As discussed in Section 2.2, careful validation of uncertainty judgements should be undertaken whenever possible using appropriate probabilistic methods (Gneiting et al., 2007).

In order for a modelling framework to support probabilistic uncertainty it is necessary that:
- a model for probabilistic uncertainty be defined for communication between all components including model and data resources (in this work, this model is UncertML)
- uncertainty should be propagated through model components by an appropriate mechanism (typically Monte Carlo) with minimal change to the model component;
- where necessary, conversions between different representations of probabilistic uncertainty (e.g. a probability distribution to samples) should be automated;
- changes of spatial, temporal and spatio-temporal support should be provided which also propagate uncertainty;

To make the framework accessible to a variety of users tools which permit the following operations would be beneficial:
- expert elicitation of uncertain inputs;
- automated method to assess uncertain inputs where observations exist, based on statistical inference;
- uncertainty and sensitivity analysis;
- visualisation of uncertain variables across space, time and space-time;
- probabilistic validation of the outputs of the chains when observations are available.

To address the computational issues it will be necessary to consider parallelism and cloud based deployment, and also the use of emulators, or statistical surrogate models,

which can be deployed easily on Web Services in a semi-automated manner, and can be built either for model components or sections of the complete workflow.

Many computational, theoretical, architectural and user interaction issues remain to be addressed before a comprehensive framework for managing uncertainty can become a reality. The UncertWeb modelling framework represents an attempt to address many of these issues and to push the boundary of what can practically be achieved closer to a complete uncertainty management system. Further development of the UncertWeb framework could be envisaged to develop tools to assist with inferential (or estimation) problems such as data assimilation and model calibration (parameter estimation). Many of the generic calibration and uncertainty evaluation tools listed in Table 2 provide useful and tested methodologies to enhance uncertainty management, and a concerted effort to integrate the most widely used methods from these tools into an interoperable architecture such as that of UncertWeb would be most beneficial to all.

Ultimately it seems natural that we should be considering computers whose basic types include not only floats, integers etc., but also the equivalent continuous and discrete random variables, in their many representations. Ruckdeschel et al. (2006) provide an implementation of this idea in the R environment. This would represent an ambitious but important paradigm shift, from viewing uncertainty information as metadata which is attached to a quantised or estimated value and may be ignored or discarded, to using uncertainty itself as the fundamental element for computation and modelling.

Acknowledgements

Bibliography

Agarwal, M., 1998. Practical issues in designing estimators under realistic model and sensor errors. Control Engineering Practice 6 (2), 197-208.

Allen, M. R., Kettleborough, J., Stainforth, D., 2002. Model Error in Weather and Climate Forecasting, from the Proceedings of the 2002 ECMWF Predictability Seminar, European Centre for Medium Range Weather Forecasting, Reading, UK, 275-294.

Allen, J. I., Somerfield, P.J., Gilbert, F.J., 2007. Quantifying uncertainty in high-resolution coupled hydrodynamic-ecosystem models, Journal of Marine Systems 64 (1-4), 3-14.

Argent, R.M., Perraud, J.-M., Rahman, J.M., Grayson, R.B., Podger, G.M., 2009. A new approach to water quality modelling and environmental decision support systems. Environmental Modelling & Software 24, 809-818.

Armstrong, C.W., Ford, R.W., Riley, G.D., 2009. Coupling integrated Earth System Model components with BFG2. Concurrency and Computation: Practice and Experience 21, 767-791.

Arnold, J.G., Srinivasan R., Muttiah R.S., Williams J.R. 1998. Large area hydrologic modeling and assessment - Part 1: Model development. Journal of the American Water Resources Association 34, 73-89.

Aster, R. C., Borchers, B. and Thurber, C. H., 2005. Parameter Estimation and Inverse Problems, Elsevier, Amsterdam.

Ataie-Ashtiani, B., Hosseini, S. A., 2005. Numerical errors of explicit finite difference approximation for two-dimensional solute transport equation with linear sorption. Environmental Modelling & Software 20 (7), 817-826.

Babendreier, J.E., Castleton, K.J., 2002. Investigating Uncertainty and Sensitivity in Integrated, Multimedia, Environmental Models: Tools for FRAMES-3MRA. Environmental Modelling & Software 20, 1043–1055.

Banta,E.R., Hill, M.C., Poeter, E., Doherty, J.E., Babendreier, J. 2008. Building model analysis applications with the Joint Universal Parameter IdenTification and Evaluation of Reliability (JUPITER) API, Computers and Geosciences 34, 310-319.

Bärlund, I., Tattari., S., 2001. Ranking of parameters on the basis of their contribution to model uncertainty. Ecological Modelling 142, 11-23.

Bastos, L. S., O'Hagan, A., 2008. Diagnostics for Gaussian process emulators. Technometrics 51, 425-438.

Beck, M.B., 2005. Environmental foresight and structural change. Environmental Modelling & Software 20, 651-670.

Berger, J.O., 1985. Statistical Decision Theory and Bayesian Analysis. Springer-Verlag.

Beven, K.J., Freer, J., 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems, Journal of Hydrology, 249, 11–29.

Bierkens, M.F.P., Finke, P.A. , de Willigen, P., 2000. Upscaling and Downscaling Methods for Environmental Research. Kluwer Academic Publishers.

Box, G. E. P., Draper, N.R., 1987. Empirical Model-Building and Response Surfaces. Wiley.

Brown, J.D., Heuvelink, G.B.M., 2005. Assessing uncertainty propagation through physically based models of soil water flow and solute transport, in: Anderson, M.G. (Ed.), Encyclopaedia of Hydrological Sciences. Chicester, UK: Wiley, pp. 1181-1195.

Brown, J.D. and Heuvelink, G.B.M. 2007. The Data Uncertainty Engine (DUE): A software tool for assessing and simulating uncertain environmental variables. Computers & Geosciences 33, 172-190

Brown, J.D., 2010. Prospects for the open treatment of uncertainty in environmental research. Progress in Physical Geography 34, 75-100.

Bullen, R.J., Cornford, D., Nabney, I.T., 2003. Outlier detection in scatterometer data: neural network approaches. Neural Networks 16 (3-4), 419-426.

Campolongo, F., Cariboni, J., Saltelli, A., 2007. An effective screening design for sensitivity analysis of large models, Environmental Modelling and Software, 22 (10), 1509-1518.

Castleton, K.J., Hoopes, B. J., Whelan, G., 2006. FRAMES Software System: Linking to the Statistical Package R. Technical report available online at http://www.pnl.gov/main/publications/external/technical_reports/PNNL-16781.pdf (last accessed 7/5/2011)

Castleton, K.J. and Meyer, P.D., 2009. FRAMES-2.0 Software System: Frames 2.0 Pest Integration (F2PEST). PNNL-18395, Pacific Northwest National Laboratory, Richland, WA., USA.

Cheng, H., Sandu, A., 2009. Uncertainty quantification and apportionment in air quality models using the polynomial chaos method. Environmental Modelling & Software 24 (8), 917-925.

Clark, M. P., & Kavetski, D., 2010. Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes. Water Resources Research, 46 (10), W10510.

Conti, S., O'Hagan, A., 2010. Bayesian emulation of complex multi-output and dynamic computer models. Journal of Statistical Planning and Inference 140 (3), 640-651.

Cooke, R.M., 1991. Experts in uncertainty: opinion and subjective probability in science. Oxford University Press.

Cooke, R M., Goossens, L.H.J., 2000. Procedures guide for structured expert judgement in accident consequence modelling. Radiation Protection Dosimetry 90, 303-309.

Cox, S., 2007. Observations and Measurements—Part 1—Observation schema. OpenGIS Standard 07-022r1, Open Geospatial Consortium Inc. Available online at: http://www.opengeospatial.org/standards/om - last accessed 07/05/2011.

Clancy, D., Tanner, J.E., McWilliam, S., Spencer, M., 2010. Quantifying parameter uncertainty in a coral reef model using Metropolis-Coupled Markov Chain Monte Carlo. Ecological Modelling 221 (10), 1337-1347.

Cullen, A. C., Frey, H. C., 1999. Probabilistic Techniques in Exposure Assessment: A Handbook for Dealing with Variability and Uncertainty in Models and Inputs, Plenum Press, New York.

Davis, T. J., Keller, C.P., 1997. Modelling and visualizing multiple spatial uncertainties. Computers and Geosciences 23, 397-408.

Dausman, A.M, Doherty, J., Langevin, C.D., Dixon, J., 2010. Hypothesis testing of buoyant plume migration using a highly parameterized variable-density groundwater model. Hydrogeology Journal 18, 147-160.

Dawid, A. P., 2004. Probability, Causality and the Empirical World: A Bayes-de Finetti-Popper-Borel Synthesis. Statistical Science 19 (1), 44-57.

de Bruin, S., Heuvelink, G.B.M., Brown, J.D., 2008. Propagation of positional measurement errors to agricultural field boundaries and associated costs. Computers and Electronics in Agriculture, 63, 245-256.

De Lara, M., Marinet, V., 2009. Multi-criteria dynamic decision under uncertainty: a stochastic viability analysis and an application to sustainable fishery management, Mathematical Biosciences 217, 118-124.

Deal, B., Schunk, D., 2004. Spatial dynamic modeling and urban land use transformation: a simulation approach to assessing the costs of urban sprawl. Ecological Economics 51, 79-95.

Desenfant, M., Priel, M., 2006. Road map for measurement uncertainty evaluation. Measurement 39 (9), 841-848.

Dessai,S., Hulme,M., Lempert,R., Pielke,R. jr., 2009. Climate prediction: a limit to adaptation? Chapter 5 in, Adapting to climate change: thresholds, values, governance (eds.) Adger,W.N., Lorenzoni,I. and O'Brien,K. Cambridge University Press.

Doherty, J. 2004. PEST, Model-Independent Parameter Estimation, User's Manual: 5th Edition, Watermark Numerical Computing, Australia.

Doherty, J., Hunt, R.J., 2009. Two statistics for evaluating parameter identifiability and error reduction, Journal of Hydrology, 366, 119-127.

Dubus, I.G., Brown, C.D., Beulke S., Turner, N.L., 2002. Uncertainty and probabilistic approaches to pesticide fate modelling. Cranfield University Report to DEFRA for Project PL0548. Available online at: http://www.pfmodels.org/downloads/PL0548.pdf - last accessed 7/5/2011.

Eldred, M.S., Swiler, L.P., Tang, G., 2011. Mixed aleatory-epistemic uncertainty quantification with stochastic expansions and optimization-based interval estimation. Reliability Engineering and System Safety 96, 1092-1113.

Erl, T., 2005. Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Prentice-Hall.

Evans , S. A., 2008 . A new look at the interaction of scientific models and policymaking. Workshop report, Policy Foresight Programme, James Martin Institute, Oxford University, Oxford. Available online at: http://www.modeling.uga.edu/tauc/Oxford/Uncertainty_Models_Policy_Record.pdf - last accessed 6/5/2011.

Fensel, D., Bussler, C., 2002. The Web Service Modeling Framework WSMF. In Proceedings of Electronic Commerce Research and Applications 2002, 113-137.

Feyen, L., Caers, J., 2006. Quantifying geological uncertainty for flow and transport modeling in multi-modal heterogeneous formations. Advances in Water Resources 29 (6), 912-929.

Foglia, L., Hill, M. C., Mehl, S. W., Burlando, P., 2009. Sensitivity analysis, calibration, and testing of a distributed hydrological model using error-based weighting and one objective function, Water Resources Research, 45, W06427.

Frehlich, R., 2011. The definition of 'truth' for Numerical Weather Prediction error statistics. Quarterly Journal of the Royal Meteorological Society 137, 84–98.

Gallagher, M., Doherty, J., 2007. Parameter estimation and uncertainty analysis for a watershed model, Environmental Modelling and Software 22 (7), 1000-1020.

Garlandini, S., Fabrikant, S., 2009. Evaluating the effectiveness and efficiency of visual variables for geographic information visualization. Spatial Information

Theory 195-211

Geller, G.N, and Turner, W., 2007. The model web: a concept for ecological forecasting, Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2007). IEEE International, 2469 – 2472.

Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B., 2003. Bayesian Data Analysis, Chapman and Hall/CRC.

Geza, M., Poeter, E. P., McCray, J.E., 2009. Quantifying predictive uncertainty for a mountain-watershed model. Journal of Hydrology 376 (1-2), 170-181.

Goldstein, M., Rougier, J., 2009. Reified Bayesian modelling and inference for physical systems. Journal of Statistical Planning and Inference 139 (3), 1221-1239.

Goldstein, M., Wooff, D., 2007. Bayes Linear Statistics, Theory and Methods. Wiley.

Gneiting, T., Balabdaoui, F., Raftery, A. E., 2007. Probabilistic forecasts, calibration and sharpness. Journal of the Royal Statistical Society Series B: Statistical Methodology 69, 243-268.

Guru, S.M., Kearney, M., Fitch, P., Peters, C., 2009. Challenges in Using Scientific Workflow Tools in the Hydrology Domain. In Anderssen, R.S., R.D. Braddock and L.T.H. Newham (eds) 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, 3514–3520.

Heuvelink, G.B.M., 1998. Error propagation in environmental modelling with GIS. Taylor & Francis Ltd.

Heuvelink, G.B.M., Pebesma, E.J., 1999. Spatial aggregation and soil process modelling. Geoderma 89, 47-65.

Heuvelink, G.B.M., Brown, J.D., van Loon, E.E. (2007). A probabilistic framework for representing and simulating uncertain environmental variables. International Journal of Geographical Information Science 21, 497-513.

Hill, M. C. and Tiedeman, C. R., 2007. Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions, and Uncertainty, Wiley-Blackwell, New Jersey.

Holden, P.B., Edwards, N. R., Oliver, K. I. C., Lenton, T. M., Wilkinson, R. D., 2010. A probabilistic calibration of climate sensitivity and terrestrial carbon change in GENIE-1. Climate Dynamics 35, 785-806.

IDABC (Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens), 2004. European Interoperability Framework for pan-European eGovernment Services version 1.0. Office for Official Publications of the European Communities.

ISO19138, 2006. Geographic information -- Data quality measures. International Standards Organisation.

ISO19139, 2007. Geographic information -- Metadata -- XML schema implementation. International Standards Organisation.

ISO19157, 2011. Geographic information -- Data quality. Under Development, International Standards Organisation.

Jagers, H.R.A., 2010. Linking data, models and tools: an overview. Proceedings of iEMSs (International Environmental Modelling and Software Society) 2010, Ottawa, Canada.

Jaher, F.C., 1970. Aerospace: End of the Gravy Years. Time (March 9th 1970) 63. Available online at: http://www.time.com/time/magazine/article/0,9171,878813,00.html (last accessed 25/11/2011)

Janowicz, K., Schade, S., Bröring, A., Keßler, C., Maué, P., Stasch, C., 2010. Semantic enablement for spatial data infrastructures. Transactions in GIS 14, 111-129.

Janssen, P.H.M., Heuberger, P.S.C., Klepper, O., 1994. UNCSAM: a tool for automating sensitivity and uncertainty analysis. Environmental Software 9, 1–11.

Jaynes, E.T., 2003. Probability Theory: The Logic of Science: Principles and Elementary Applications, Cambridge University Press.

Jebaraj, S., Iniyan, S., 2006. A review of energy models. Renewable and Sustainable Energy Reviews 10 (4), 281-311.

Jones, R., Bastin, L., Cornford, D., Williams, M., 2010. Handling and communicating uncertainty in chained geospatial Web Services. Proceedings of Spatial Accuracy 2010, Leicester, UK.

Kardos, J., Benwell, G L., Moore, A.B., 2007. Assessing different approaches to visualise spatial and attribute uncertainty in socioeconomic data using the Hexagonal or Rhombus (HoR) Trustree. Computers, Environment and Urban Systems 31, 91-106.

Kelson, V.A., Hunt, R.J., Haitjema, H.M., 2002. Improving a regional model using reduced complexity and parameter estimation. Ground Water 40,132–143.

Kerwin, A., 1993. None too solid: medical ignorance. Science Communication 15, 166-185.

Kennedy, M.C., Anderson, C.W., Conti, S., O'Hagan, A., 2006. Case studies in Gaussian process modelling of computer codes. Reliability Engineering and System Safety 91, 1301–1309.

Kuczera, G., Kavetski, D., Franks, S., Thyer, M., 2006. Towards a Bayesian total error analysis of conceptual rainfall-runoff models: Characterising model error using storm-dependent parameters. Journal of Hydrology 331 (1-2), 161-177.

Langford, W., Gordon, A., Bastin, L. (2009). When do conservation planning methods deliver? Quantifying the consequences of uncertainty. Ecological Informatics 4, 123-135.

le Maire, G., Marsden, C., Verhoef, W., Ponzoni, F., Lo Seen, D., Bégué, A., Stape, J-L., Nouvellon, Y., 2011. Leaf area index estimation with MODIS reflectance time series and model inversion during full rotations of Eucalyptus plantations. Remote Sensing of Environment 115, 586-599.

Levy, J.K, Hipel, K.W., Kilgour, D.M., 2000. Using environmental indicators to quantify the robustness of policy alternatives to uncertainty. Ecological Modelling 130, 79-86.
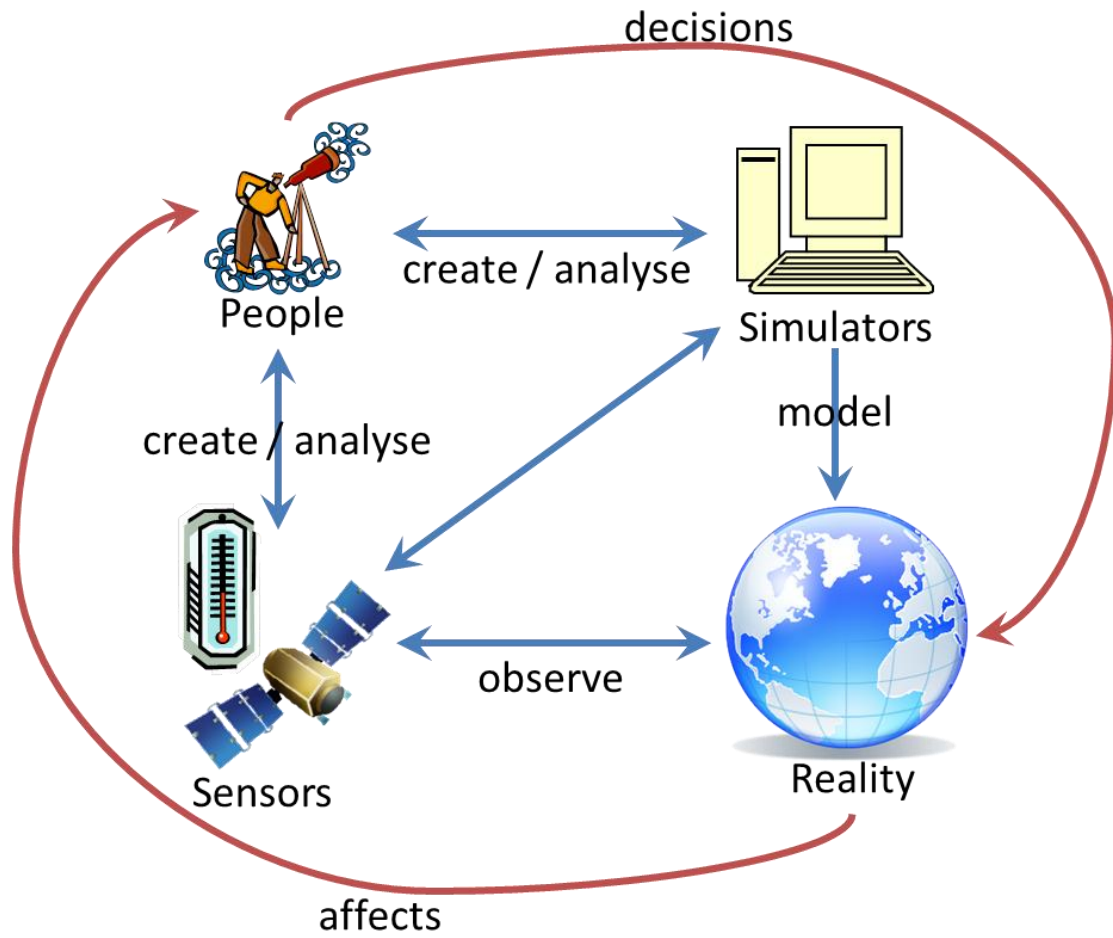
Lindenschmidt, K-E., Fleischbein, K., Baborowski, M., 2007. Structural uncertainty in a river water quality modelling system. Ecological Modelling 204 (3-4), 289-300.

MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., Hetzler, E., 2005. Visualizing geospatial information uncertainty: What we know and what we need to know. Cartography and Geographic Information Science 32, 139-160.

Matott, L.S., Rabideau, A.J., 2008. Calibration of complex subsurface reaction models using a surrogate-model approach, Advances in Water Resources, 31, 1697-1707.

Matott, L. S., Babendreier, J. E, Purucker S. T., 2009. Evaluating uncertainty in integrated environmental models: A review of concepts and tools, Water Resources Research, 45, doi:10.1029/2008WR007301.

Meyer, M.A., Booker, J. M., 2001. Eliciting and Analyzing Expert Judgment: A Practical Guide. Society for Industrial Mathematics.

Meyers, H.B., 1969. For Lockheed Everything's Coming Up Unk-Unks. Fortune 80 (August 1969) 77- 81.

Moore, C.R., and Doherty, J.E., 2005, Role of the calibration process in reducing model predictive error.Water Resources Research 41, doi:10.1029/2004WR003501

Morris, M. D., 1991. Factorial sampling plans for preliminary computational experiments. Technometrics, 33 (2), 161–174.

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

Nativi, S., Mazzetti, P., Geller, G., 2011. Environmental Model Access and Interoperability: the GEO Model Web Initiative, in this special issue.

Oakley, J., O'Hagan, A., 2004. Probabilistic sensitivity analysis of complex models: a Bayesian approach. Journal of the Royal Statistical Society Series B: Statistical Methodology 66, 751–769.

Oakley J. E., O'Hagan, A., 2010. SHELF: the Sheffield Elicitation Framework (version 2.0), School of Mathematics and Statistics, University of Sheffield, UK. (http://tonyohagan.co.uk/shelf)

OGC 02-112, 2002. The OpenGIS Abstract Specification Topic 12: OpenGIS Service Architecture -version 4.3. Available at http://portal.opengeospatial.org/files/?artifact_id=1221

OGC 08-094r1, 2004. SWE Common Data Model Encoding Standard. Available online at: http://www.opengis.net/doc/IS/SWE/2.0. - last accessed 07/05/2011.

O'Hagan, A., 1998. Eliciting expert beliefs in substantial practical applications. The Statistician 47, 21-35.

O'Hagan, A., 2006. Bayesian analysis of computer code outputs: a tutorial. Reliability Engineering and System Safety 91, 1290-1300.

O'Hagan, A., 2011. Probabilistic uncertainty specification: Overview, elaboration techniques and their application to a mechanistic model of carbon flux. Environmental Modelling and Software, In Press, available online 31 March 2011.
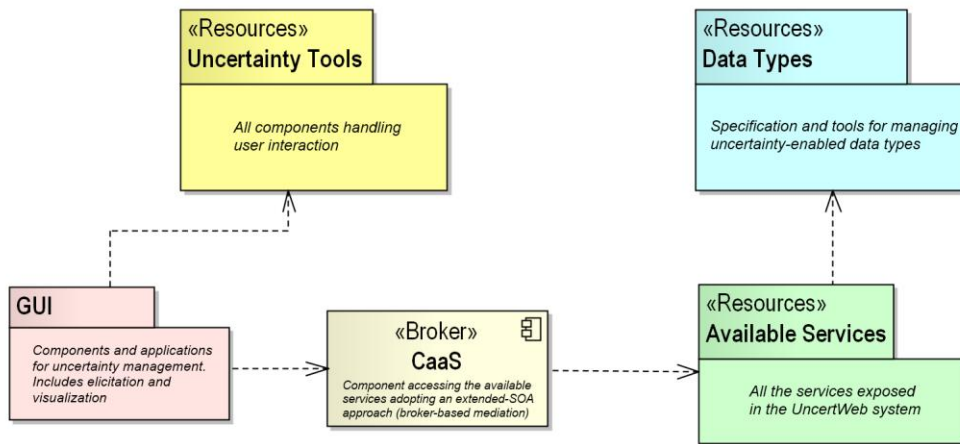
O'Hagan A., Buck, C., Daneshkhah, A., Eiser, J., Garthwaite, P., Jenkinson, D., Oakley, J., Rakow, T., 2006. Uncertain Judgements : Eliciting Experts' Probabilities (John Wiley & Sons, Ltd., Chichester).

Oracle, 2011. Documentation for Crystal Ball, version 11.2.1 http://www.oracle.com/technetwork/middleware/crystalball/documentation/

Palisade, 2011. Documentation for @Risk version 5.7. http://www.palisade.com/risk/

Pang, A.T., Wittenbrink, C.M., Lodha, S.K.,1997. Approaches to uncertainty visualization. Visual Computer 13, 370-90.

Park, I., Amarchinta, H.K., Grandhi, R.V., 2010. A Bayesian approach for quantification of model uncertainty. Reliability Engineering and System Safety 95 (7), 777-785.

Pauly, D., Christensen, V., Walters, C., 2000. Ecopath, Ecosim, and Ecospace as tools for evaluating ecosystem impact of fisheries. ICES Journal of Marine Science 57, 697-706.

Pebesma, E., Cornford, D., Dubois, G., Heuvelink, G.B.M., Hristopoulos, D., Pilz, J., Stohlker, U., Morin, G., Skoien, J.O., 2011. INTAMAP: the design and implementation of an interoperable automated interpolation web service. Computers and Geosciences 37, 343-352.

Poeter, E.P., Hill, M.C., Banta, E.R., Mehl, S. and Christensen, S, 2005, UCODE_2005 and six other computer codes for universal sensitivity analysis, calibration, and uncertainty evaluation: U.S. Geological Survey Techniques and Methods 6-A11. http://typhoon.mines.edu/freeware/ucode/

Pratt, A., Peters, C., Siddeswara, G., Lee, B., Terhorst, A., 2010. Exposing the Kepler Scientific Workflow System as an OGC Web Processing Service. Proceedings of iEMSs (International Environmental Modelling and Software Society) 2010, Ottawa, Canada.

Rabideau, A. J., Matott, L. S., Jankovic, I., Craig, J. R., Becker, M. W., 2005. Influence of numerical precision on the calibration of AEM-based groundwater flow models. Environmental Geology, 48, 57-67.

Rahman, J.M., Seaton, S.P., Cuddy, S.M., 2004. Making frameworks more useable: using model introspection and metadata to develop model processing tools. Environmental Modelling and Software 19, 275-284.

Rahman, J. M., Perraud, S.P., Hotham, H., Murray, N., Leighton, B., Freebairn, A., Davis, G., Bridgart, R., 2005. Evolution of TIME. In: MODSIM2005, International Congress on Modelling and Simulation, pp. 697–703, December.

Refsgaard, J.C., van der Sluijs, J.P., Brown, J., van der Keur, P., 2006a. A framework for dealing with uncertainty due to model structure error. Advances in Water Resources 29 (11), 1586-1597.

Refsgaard, J.C., Van Der Keur, P., Nilsson, B., Müller-Wohlfeil, D., Brown, J., 2006b. Uncertainties in river basin data at various support scales - example from Odense pilot river basin. Hydrology and Earth System Sciences Discussions 3, 1943-1985.

Refsgaard, J.C., van der Sluijs, J. P., Hojberg, A. L., Vanrolleghem, P.A., 2007. Uncertainty in the environmental modelling process - A framework and guidance. Environmental Modelling and Software 22 (11), 1543-1556.

Reichert, P., 1997. On the necessity of using imprecise probabilities for modelling environmental systems. Water Science and Technology 36, 149-156.

Reichert, P., 2006. A standard interface between simulation programs and systems analysis software. Water Science and Technology 53, 267-75.

Rizzoli, A.E., Davis, J.R., Abel, D.J., 1998. Model and data integration and re-use in environmental decision support systems. Decision Support Systems 24, 127–144.

Roughgarden, T., Schneider, S.H., 1999. Climate Change Policy: Quantifying Uncertainties for Damages and Optimal Carbon Taxes. Energy Policy 27, 415-429.

Ruckdeschel, P., Kohl, M., Stabla, T., Camphausen, F., 2006. S4 classes for distributions. R News 6 (2), 2-6.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S., 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Computer Physics Communications 181, 259-270.

Santner, T. J., Williams, B.J., Notz, W., 2003. The design and analysis of computer experiments. Springer, New York.

Shahsavani, D., Grimvall, A., 2011. Variance-based sensitivity analysis of model outputs using surrogate models. Environmental Modelling & Software 26 (6), 723-730.

Sigel, K., Klauer, B., Pahl-Wostl, C., 2010. Conceptualising uncertainty in environmental decision making: the example of the EU Water Framework Directive. Ecological Economics, 69 592–510.

Smets, P., 1991.. Varieties of ignorance and the need for well-founded theories. Information Sciences 57-58, 135-144

Smith, T.J., Marshall, L.A., 2010. Exploring uncertainty and model predictive performance concepts via a modular snowmelt-runoff modeling framework. Environmental Modelling and Software 25 (6), 691-701.

Stainforth, D. A., Aina, T., Christensen, C., Collins, M., Faull, N., Frame, D. J., Kettleborough, J. A., Knight, S., Martin, A., Murphy, J. M., Piani, C., Sexton, D., Smith, L. A., Spicer, R. A., Thorpe, A. J., Allen, M. R., 2006. Uncertainty in predictions of the climate response to rising levels of greenhouse gases. Nature, 433 (7024), 403-406.

Stoffelen, A., 1998. Toward the true near-surface wind speed: Error modeling and calibration using triple collocation. Journal of Geophysical Research 103, 7755–7766.

Tarantola, A., 2005. Inverse problem theory and methods for model parameter estimation. Society for Industrial and Applied Mathematics, Philadelphia, USA.

Tiedeman, C., Ely, D., Hill, M. and O'Brien, G., 2004. A method for evaluating the importance of system state observations to model predictions, with application to the Death Valley regional groundwater flow system. Water Resources Research 40: doi: 10.1029/2004WR003313.

Tonkin, M.J., Papadopoulos, S.S, Tiedeman, C.R., Ely, D.M. and Hill, M.C., 2007, OPR-PPR, a Computer Program for Assessing Data Importance to Model Predictions Using Linear Statistics. USGS Report TM –6E2, http://pubs.usgs.gov/tm/2007/tm6e2/

Urban, N. M., Fricker, T.E., 2010. A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model. Computers and Geosciences 36 (6), 746-755.

Valcke, S., 2006. OASIS3 User Guide (prism_2-5). PRISM Support Initiative Report No 3.

Van de Kassteele, J., Velders, G.J.M., 2006. Uncertainty assessment of local NO_2 concentrations derived from error-in-variable external drift kriging and its relationship to the 2010 air quality standard. Atmospheric Environment 40, 2583-2595.

Villa, F., Costanza. R., 2000. Design of multi-paradigm integrating modeling tools for ecological research. Environmental Modelling and Software 15, 169-177.

Villa, F., Athanasiadis, I.N., Rizzoli, A.E., 2009. Modelling with knowledge: A review of emerging semantic approaches to environmental modelling. Environmental Modelling and Software 24, 577-587

Voinov, A., Costanza, R., Wainger, L., Boumans, R., Villa, F., Maxwell, T., Voinov, H., 1999. Patuxent landscape model: integrated ecological economic modeling of a watershed. Environmental Modelling and Software 14, 473-491.

Warren, R., de la Nava Santos, S., Arnell, N.W., Bane, M., Barker, T., Barton, C., Ford, R., Füssel, H.-M., Hankin, R.K.S., Hinkel, J., Klein, R., Linstead, C., Kohler, J., Mitchell, T.D., Osborn, T.J., Pan, H., Raper, S.C.B., Riley, G., Schellnhüber, H.J., Winne, S., Anderson, D., 2008.. Development and illustrative outputs of the Community Integrated Assessment System (CIAS), a multi-institutional modular integrated assessment approach for modelling climate change. Environmental Modelling and Software 23, 592-610.

Williams, M., Cornford, D., Bastin, L., Pebesma, E., 2009. Uncertainty markup language (UncertML). OpenGIS Discussion Paper 08-122r2, Open Geospatial Consortium Inc. Available online at: http://portal.opengeospatial.org/files/?artifact_id=33234 - last accessed 07/05/2011.

Wittenbrink, C.M., Pang, A.T., Lodha, S.K., 1996. Glyphs for visualizing uncertainty in vector fields. IEEE Transactions on Visualization and Computer Graphics 2, 266-279.

Wood, J., Slingsby, A., Khalili-Shavarini, N., Dykes, J., Mountain, D., 2009. Visualization of uncertainty and analysis of geographical data, In: Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology, pp. 261-262.

decisions

create / analyse

create / analyse

model

observe

Sensors

Reality

affects

«Resources»
Uncertainty Tools

All components handling user interaction

«Resources»
Data Types

Specification and tools for managing uncertainty-enabled data types

GUI

Components and applications for uncertainty management. Includes elicitation and visualization

«Broker»
CaaS

Component accessing the available services adopting an extended-SOA approach (broker-based mediation)

«Resources»
Available Services

All the services exposed in the UncertWeb system

Domain experts
Manage your network of domain experts.

Elicit judgements
Handle discrete, continuous and spatial variables.

Monitor progress
View the current status of all your variables

Password

Sign in ☐ remember me?
Forgot your password?

What is it?

Expert elicitation is the process of synthesising expert opinions on a subject where there is uncertainty due to insufficient data, when such data is unattainable because of physical constraints or lack of resources.

How it works

The Elicitator allows you to create 'elicitation problems' with many uncertain variables. You can invite a number of domain experts to make judgements about the uncertainty within these variables, culminating in a probability distribution.

Register today

Registration is free and should only take 60 seconds. Once you are registered, create an elicitation problem and begin understanding the uncertainties about the data you work with.

create an account

Create an elicitation problem

Name * Western (Basalt) Plains Natural Ter

A short name to describe the problem.

B *I* U ≡ ≡ ≡ ≡ ≔ ≔ ≔ ≔ ✂ ✂ x₂ x² Ω

The native Western (Basalt) Plains Natural Temperate Grassland to the west of Melbourne is one of Australia's most endangered vegetation types with over 99.5% of its original distribution having been lost or substantially altered. It supports a diversity of animal species, including eight nationally threatened species. The remaining remnants predominantly occur on private land (DNRE, 1997) and continue to face a number of threats including inappropriate grazing or fire regimes, weed invasion, inappropriate herbicide use and the application of fertilizers. The condition of native grassland is expected to change depending on its management status. This elicitation exercise aims at discovering your beliefs about the rate of change for grassland in two different categories: managed and unmanaged. Land is considered to be managed when it is within a public reserve or private land incentive program. Unmanaged land is grassland on private land subject to entitled uses (such as grazing) and uncontrolled threats (such as perennial weed invasion). There will be variation among the grasslands in the two categories, and so it is important that you include this uncertainty in the values you submit. You can do this by specifying ranges and quartiles.

Description

A brief summary describing the problem.

Save elicitation problem

Briefing document

Research objectives

Unmanaged grasslands have been observed to degrade over time, while managed grasslands generally improve in quality. However, once a patch of grassland falls below a certain condition threshold it is difficult to full restore it to a state where it can support many of the original species. This part of the elicitation is about the level of condition at which you believe this threshold lies.

Please submit:

- the median (the level at which you believe this threshold is most likely to lie)
- the minimum and maximum (the condition levels outside which you are sure it does not lie)
- the lower and upper quartiles (the levels between which you are 50% sure the value lies.

Irrecoverable condition threshold

Minimum	0
Maximum	1.00
Lower quartile	0.2
Median	0.35
Upper quartile	0.44

Task	Status
1. Read briefing document	✔ complete
2. Set minimum value	✔ complete
3. Set maximum value	✔ complete
4. Set lower quartile	✔ complete
5. Set upper quartile	✔ complete
6. Set median value	✔ complete

Results:– Normal (0.33, 0.17)

CDF | PDF | Both
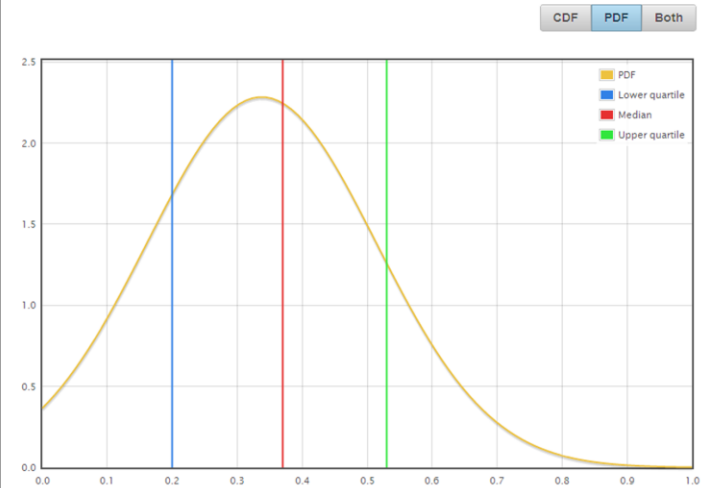
Western (Basalt) Plains Natural Temperate Grassland / Irrecoverable condition threshold

Pooled probability distribution


```
{
  - GammaDistribution: {
    - shape: [
        3.33
      ],
    - scale: [
        0.1
      ]
    }
}
```

View details
Export as UncertML

Elicit input ranges
- Use the Elicitator
- UncertML

Find important inputs
- Morris method based screening
- User diagnostics

Design the training set
- Latin hypercube sampling

Train the emulator
- Bayesian plugin inference
- Maximum likelihood

Validate the emulator
- Plots (standardised errors, Q-Q plots)
- Metrics

Use the emulator
- Uncertainty analysis
- Sensitivity analysis