

Communicating Uncertainty in Spatial Decision Support Systems: a Case Study of Bioenergy-Crop Potentials in Mozambique

Judith Anne Verstegen^a, Floor van der Hilst^b, Derek Karssenberga, André Faaij^b

^aDepartment of Physical Geography, Faculty of Geosciences, Utrecht University, The Netherlands

^bCopernicus Institute for Sustainable Development and Innovation, Utrecht University, The Netherlands

INTRODUCTION

Spatio-temporal simulation models of environmental processes have been constructed for a wide range of application domains, including hydrology (e.g., Gosling and Arnell, 2010), geomorphology (e.g., Baas, 2002; Jerolmack and Paola, 2007), atmospheric transport (e.g., Brandt et al., 2000), erosion (e.g., D'Ambrosio et al., 2007; Nearing et al., 2005), and land use change (e.g., Verburg and Overmars, 2009). These models have in common that they simulate change over time of a spatial phenomenon or, more likely, several spatial phenomena that interact with each other. Those dynamic processes and their interactions tend to be complex and are rarely fully understood. As a result errors are inherent in simulation models through the debatable choice and conceptualization of relevant sub-processes, the discretization of information and the quantification of input variables and model parameters that are hard or even impossible to measure. These errors propagate through the model because the state of the modelled system at a certain moment in time is a function of its state in the past. This generates uncertainty in model outputs. Methods of uncertainty analysis for spatio-temporal models, sometimes referred to as error propagation modelling, have been discussed in a large number of studies (e.g., Aerts et al., 2003; Brown and Heuvelink, 2007; Eckhardt et al., 2003; Heuvelink, 1998).

Spatio-temporal simulation models are widely used outside scientific communities (Manson, 2007). Especially models constructed to assist policy making are abundant (Bradshaw and Borchers, 2000). Such models can be grouped under the term Planning Support Systems (PSSs), or more generally Spatial Decision Support Systems (SDSSs) (Geertman and Stillwell, 2004). SDSSs are interactive, computer-based systems that include simulation models and visualization tools designed to assess the impact of possible decisions. Users of SDSSs demand practical models with clear and unambiguous results to facilitate decision making. The result of this desire for clarity and simplicity is that SDSSs tend to underestimate, if not ignore, uncertainty (Foody, 2003). They thus yield clear, but therefore sometimes deceptively precise outputs.

Modellers are continuously improving the structure of their models and developing approaches for better estimation of input variables and model parameters. However, construction of completely error-free models is simply impossible, because models remain simplifications of open systems (Manson, 2007). Uncertainty thus needs to be communicated and dealt with. Therefore we claim, together with others (e.g., Foody, 2003; Ivanovic and Freer, 2009; Oreskes et al., 1994), that instead of obscuring uncertainty policy makers should be made more aware of uncertainty in SDSSs. Calculation and communication of model uncertainty and its distribution in space and time makes the model more transparent and the output more informative, which gives users a better basis for decision making (Bradshaw and Borchers, 2000; Geertman, 2006). As it is unlikely that the average user of an SDSS will be inclined or able to develop skills in handling uncertainty at a comparable level as the modellers themselves (Foody, 2003), modellers should aid their end users by providing intuitive and insightful indicators of uncertainty and straightforward tools to visualize those. Ideally, this should be possible without too much additional work on the modeller's side.

A framework that offers a combined interface for spatio-temporal modelling and geospatial analysis is the PCRaster model construction framework (Karssenberga et al., 2010; PCRaster, 2010). The aim of this paper is to show how a modeller can use the PCRaster Python framework to construct an SDSS that integrates simulation, uncertainty analysis and visualization in order to provide policy makers with the means to take uncertainty into account in decision making. This is illustrated by a case study of bioenergy-crop potential in Mozambique. Although some studies have been conducted to assess the area of potentially available land for bioenergy crops in Mozambique (Batidzirai et al., 2006; Watson, 2010), none of these studies was carried out in both a temporally dynamic and spatially explicit way. A land use change model is developed with the PCRaster Python framework to evaluate where bioenergy crops can be cultivated without endangering food production now and in

the near future when population and food intake per capita and thus arable land and pasture areas will increase. It is aimed to show that the PCRaster Python framework allows stochastic model inputs and produces easily understandable outputs of the resulting model uncertainty with respect to possible locations for bioenergy crops.

The next section of this paper first describes the concepts and methods of the PCRaster Python framework and why this framework is expected to be suitable for constructing an uncertainty-inclusive SDSS. After that it is outlined how the PCRaster Python framework is applied to construct the land use change model for Mozambique, functioning as an SDSS that explicitly takes input errors into account. The results section shows uncertainty in model output due to uncertainty in population growth and diet change in Mozambique between 2005 and 2030. The final section discusses the usability of the PCRaster Python framework to aid modellers constructing an SDSS that informs policy makers by explicitly indicating the quantity, location and time step of uncertainty in model outputs.

METHODOLOGY

Software framework

A method that provides a relatively easily understandable representation of uncertainty is Monte Carlo simulation. The Monte Carlo method is attractive because of its general applicability and ease of implementation (Aerts et al., 2003). An error in an input variable or model parameter is represented by a probability distribution that embodies all possible values and their likelihood of occurrence. Monte Carlo simulation involves running the model a large number of times, each time drawing a realization from the input probability distribution(s). For spatial models this results in different spatial patterns for the different runs. With GIS these results can be compiled into one map that indicates, for example, the average output value or the probability on a certain state. However, functionality to accomplish this is not always present in software packages. Most packages are either dedicated to model development, e.g. Stella (Stella, 2010) and NetLogo (NetLogo, 2010), or to uncertainty analysis (for a package overview see Goovaerts, 2010). The developer of the SDSS sometimes does undertake an evaluation of uncertainty with such an external package (e.g., Brown et al., 2005; Chang et al., 2008) and present the results to the users, but static uncertainty information for specific model settings are not of much help to the end user, because uncertainty can change with different input data or parameter settings. Much more convenient for SDSS users (and for the modeller as well) would be if both the option to define probability distributions for stochastic inputs, the possibility to run a Monte Carlo simulation and the representation of the resulting uncertainty are integrated in the SDSS, so that only a single software package is needed.

The PCRaster model construction framework (Karssenberg et al., 2010; PCRaster, 2010) facilitates this integration of spatio-temporal modelling and uncertainty analysis through the PCRaster Python library (Karssenberg et al., 2007). This library provides a large set of spatial and spatio-temporal functions on raster maps, embedded in the Python language (Python, 2010). Both a spatio-temporal modelling framework and a Monte Carlo analysis framework are present as a Python class. These classes include methods to write the simulation results and uncertainty analysis to disk as maps, which can be visualized with the Aguila software (Pebesma et al., 2007), included in the PCRaster Python distribution package. The advantage of working with the generic programming language Python is that other Python libraries can be embedded to support model construction and even complete existing models can be imported into the framework for uncertainty analyses (Karssenberg et al., 2010). The combination of a flexible and extendible spatio-temporal modelling framework, uncertainty analysis options and visualization routines makes the PCRaster Python framework suitable for the construction of uncertainty-inclusive SDSS.

To allow construction of a spatio-temporal model that permits stochastic inputs and assessment of the resulting uncertainty three main methods are provided by the framework that together form the scheme in Table 1 (Karssenberg and De Jong, 2006). Firstly, it provides a method for evaluation of the spatio-temporal process itself (line 2). Herein, the modeller can program the equations that represent the change of the system state within a time step (line 3). The framework provides a number of functions particularly designed for spatial and stochastic operations for this purpose. Secondly, a loop over this spatio-temporal model is performed to generate the Monte Carlo samples (line 1), for which the stochastic variables and parameters differ according to their probability distribution. Finally, summary statistics, representing uncertainty, are computed (line 4), using all Monte Carlo realizations from the previous methods.

Table 1: Modelling scheme of the PCRaster Python framework

```

1  for each MC sample:
2    for each time step:
3      solve system state equation
4    compute summary statistics

```

The individual Monte Carlo sample results and summary statistics can be written to disk with a function from the PCRaster Python library. This function uses rules for file names defined by the modelling framework. The files can directly be visualized with the Aguila software that recognizes these name conventions. Temporal data can be animated through time and stochastic data can be visualized as maps or plots by means of user defined percentile values, mean values, standard deviations, cumulative probability distributions, or exceedance probabilities (Karssenberget al., 2010; Pebesma et al., 2007).

Land use change model

The potential to employ the PCRaster Python framework for an SDSS including uncertainty is illustrated by the construction of a land use change model of Mozambique. This model has the purpose to aid policy makers to evaluate where bioenergy crop plantations can be allocated. The case is relevant, because Mozambique is considered promising for bioenergy crop production by its vast amounts of available land (Smeets et al., 2004), favourable environmental conditions for cultivation (Batidzirai et al., 2006), and relatively low productivity of current agriculture, which offers potential for improvement. The population is expected to increase and its diet is expected to change as well, which induces shifts in land use. This information is needed by policy makers because cultivation of bioenergy crops should not endanger food production now or in the future. However, the various population growth and diet change projections differ significantly (Arndt et al., 2010; FAO, 2003; UNDP, 2008), so applying those drivers deterministically ignores a large input error. Our land use change model, or SDSS, takes this error and its resulting uncertainty into account. The model is used here for illustration of the possibilities of the PCRaster Python framework, so only a brief outline of the implemented processes is given. For a more elaborate explanation the reader is referred to van der Hilst et al. (forthcoming).

The main procedure of the model is the state transition function, the spatially explicit change in land use. This change is steered by two factors: the demand of the population for food and wood, and the maximum potential yield of the land, defined by the country's technological state of art in agriculture. The actual location of the expansion or contraction of the land use types is determined by suitability factors, like distance to cities and transport networks, current land use in the neighbourhood and location-specific yield due to characteristics of the soil and climate. Areas not occupied by food production or protected land uses, like national parks, are available for bioenergy crops.

The model uses a time step (Δt) of one year; time step $t = 1, 2, \dots, T$. Most equations are evaluated separately for each of the N land use types, with $n = 1, 2, \dots, N$. Population growth and diet change are assumed to be the main drivers of land use change. The demand $d_{n,t}$ (kg) for products of a land use type is:

$$d_{n,t} = p_t * i_{n,t} * r_t, \text{ for each } n \text{ in each } t \quad (1)$$

In Equation 1, p_t denotes the number of inhabitants, intake $i_{n,t}$ (kg / caput) specifies the demand per capita of products of land use type n , and the self-sufficiency ratio r_t (-) is the extent to which the food demands of the country's population are fulfilled by products from the country itself.

Potential yield $P_{n,t}$ (kg) is the possible product yield of a land use type at a certain location:

$$P_{n,t} = m_{n,t} * f, \text{ for each } n \text{ in each } t \quad (2)$$

Note that a bold font indicates that a variable is a spatial field. In equation 2, $m_{n,t}$ is the maximum possible product yield (kg / cell) of products from a land use type, which can increase through time due to technological improvements in the agricultural sector (FAO, 2003). The location-specific variable f is the actual fraction of this yield that can be reached in a cell, depending on factors like soil type, climate, and water availability.

The total product yield of a land use type is calculated using the current land use map as follows. First a spatial field of the current product yield $\mathbf{c}_{n,t}$ (kg) of land use type n is constructed that contains the value of $\mathbf{P}_{n,t}$ for cells that are currently occupied by type n and zero for cells occupied by other land use types. The total product yield $\mathcal{Y}_{n,t}$ (kg) of this land use type is:

$$\mathcal{Y}_{n,t} = \sum \mathbf{[(c)]}_{n,t} \quad (3)$$

, for each n in each t

In Equation 3, the summation sign indicates summation over the whole spatial field. To determine where a certain land use expands or contracts every land use type is assigned a number of suitability factors. The number and sort of suitability factors differ per land use type. For an overview of these factors, the methods to compute them and their parameters see van der Hilst et al. (forthcoming). For every land use type a total suitability map is computed:

$$\mathbf{s}_{n,t} = \sum_{i=1}^{X_n} \mathbf{[(W)]}_{i,n} * \mathbf{p}_{i,n,t} \quad (4)$$

, for each n in each t

Equation 4 sums the X individual suitability factor maps $\mathbf{P}_{i,n,t}$ of land use type n , with weights $\mathbf{W}_{i,n}$ that designate their relative importance, to form suitability $\mathbf{s}_{n,t}$ (-). This suitability indicates the total appropriateness of a given area for land use n at time step t by a value between 0 and 1. Now, all information is available to start the allocation procedure. The allocation schema can be written in pseudo-code as in Table 2. The land use types have a certain hierarchy determined mainly by their economic importance. The type with the highest 'rank' starts allocating. When the land use type expands, it allocates new cells of this type at locations with the highest suitability, i.e. $\max(\mathbf{s}_1(n, t))$, and when it contracts it removes cells of this type with the lowest suitability, i.e. $\min(\mathbf{s}_1(n, t))$. Cells are converted to or removed from this land use until the total yield $\mathcal{Y}_{n,t}$ equals the total demand $\mathbf{d}_{n,t}$. In line 7 of Table 2 the number 0 refers to the land use category 'abandoned'.

Table 2: Pseudo-code for land use allocation procedure. Allocation of each land use type n in each time step t proceeds until yield $\mathcal{Y}_{n,t}$ fulfills demand $\mathbf{d}_{n,t}$.

```

1  if  $\mathbf{d}_{n,t} > \mathcal{Y}_{n,t}$  :
2    while  $\mathbf{d}_{n,t} > \mathcal{Y}_{n,t}$  :
3      convert cell with  $\max(\mathbf{s}_1(n, t))$ 
to n
4      update  $\mathcal{Y}_{n,t}$ 
5  else if  $\mathbf{d}_{n,t} < \mathcal{Y}_{n,t}$  :
6    while  $\mathbf{d}_{n,t} < \mathcal{Y}_{n,t}$  :
7      convert cell with  $\min(\mathbf{s}_1(n, t))$ 

```

When allocation of the land use type with the highest rank is finished, allocation of the next type is performed, with the restriction that it cannot convert cells with a land use type that has already been allocated in that time step. After each time step, when the land use map has changed according to the demands of the different land use types, it can be determined which cells are potentially available for bioenergy crops. This is done by excluding all areas occupied by crops, pasture, steep slopes, roads, water, cities and nature reservation areas. This results in a Boolean map $\mathbf{[(b)]}_t$ where cells are available (1) or unavailable (0) for bioenergy crops at time step t . The total possible crop yield of this area can be calculated using Equations 2 and 3.

Implementation

The schedule of the land use change model, implemented in the PCRaster Python framework, is given in Table 3. The frame of the schedule is provided by the PCRaster Python framework and

corresponds to the scheme shown in Table 1. The PCRaster Python framework consists of four main methods: the `premcloop` (line 6 in Table 3) is evaluated only once, the `initial` (line 21) once for each realization, the `dynamic` (line 32) once for each time step in each realization, and the `postmcloop` (line 49) performs the uncertainty analysis. Some variables have the prefix `self`, as they are defined as member variables to allow usage over the four different methods.

Table 3: Main scheme of land use change model. Three dots and a discontinuity in the line numbering indicate omitted sections.

```

1  class LandUseChangeModel(DynamicModel, MonteCarloModel):
2      def __init__(self):
3          DynamicModel.__init__(self)
4          MonteCarloModel.__init__(self)
5          setclone('landuse')

6      def premcloop(self):
7          self.initialEnvironment = self.readmap('landuse')
...
17         self.landUseList = Parameters.getLandUseList()
...

21     def initial(self):
22         self.environment = self.initialEnvironment
23         self.populationStoch = mapuniform()
24         self.landUse = LandUse(self.landUseList, self.environment)
25         self.landUse.createLandUseTypeObjects(self.relatedTypeDict
...
           , self.suitFactorDict, self.weightDict, self.varDict)
...

32     def dynamic(self):
33         timeStep = self.currentTimeStep()
34         maxYield = timeinputscalar('maxY.tss', self.environment)
35         demandUp = timeinputscalar('deUp.tss', self.environment)
36         demandLow = timeinputscalar('deLow.tss', self.environment)
37         demandDiff = (demandUp - demandLow)
38         demand = demandDiff * self.populationStoch + demandLow
39         self.landUse.calculateSuitabilityMaps()
40         self.landUse.allocate(maxYield, demand)
41         self.environment = self.landUse.getEnvironment()
42         biofuelPotential = self.landUse.getBiofuelPotential()
43         biofuelYield = self.landUse.getPotentialBiofuelYield()
44         self.report(self.environment, 'landUse')
...

49     def postmcloop(self):
50         name = ['fuel']
51         sampleNumbers = self.sampleNumbers()
52         timeSteps = self.timeSteps()
53         mcaveragevariance(names, sampleNumbers, timeSteps)
54         name = ['yiel']
55         percentiles = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
56         mcpercentiles(name, percentiles, sampleNumbers, timeSteps)

57     nrOfTimeSteps = 26
58     nrOfSamples = 50
59     myModel = LandUseChangeModel()
60     dynamicModel = DynamicFramework(myModel, nrOfTimeSteps)
61     mcModel = MonteCarloFramework(dynamicModel, nrOfSamples)
62     mcModel.run()

```

When the `LandUseChangeModel` is initiated it calls `PCRaster` Python's `DynamicFramework` (line 3) and `MonteCarloFramework` (line 4). Among other things, these frameworks allow definition of the number of time steps (line 57) and Monte Carlo samples (line 58). Next, the `premcloop` (line 6) imports input maps in lines 7 to 16. A separate file, `Parameters.py`, exhibits all non-spatial inputs. This is done to prevent that the end user has to make changes in the main model scheme. All its variables and parameters are imported in lines 17 to 20.

The `initial` method (line 21) is used to define initial or temporally constant variables that can vary in the different Monte Carlo realizations. The class `LandUse`, defined by the model builder, is called here (line 24). It is used to keep track of the changing land use map. This class in its turn has a method to instantiate N objects of the class `LandUseType` (line 25) that handle the type-specific tasks, like computing a suitability map and allocating land. These methods are implemented with functions from the `PCRaster` Python library, including point, neighbourhood, and global operations.

In the `dynamic` method (line 32), the temporal component, a time series of maximum yield ($m_{n,t}$) per land use type is imported (line 34). Demand is defined as a stochastic input variable by two time series per land use type. One includes the upper bound of the predicted population P_t and intake $i_{n,t}$ (line 35), and one lower bound (line 36) to form the range of all possible input values. In each Monte Carlo sample a value from a uniform distribution between 0 and 1 is drawn with the `PCRaster` Python `mapuniform()` function (line 23), which is used as the position between the upper and lower bound to determine the value (realization) for $d_{n,t}$ (lines 37 and 38).

Next, the total suitability maps are calculated for all land use types (line 39) and the allocation procedure, explained in Table 2, is called (line 40 in Table 3). The land use map is updated (line 41) and it is checked which cells are left over for bioenergy crops (line 42) and what their total potential yield is (line 43). Output maps are saved to disk with the `PCRaster` Python function `self.report()` (lines 45 to 48) that creates file extensions recognizable for the `Aguila` software.

In the `postcloop`, the `PCRaster` Python function `mcaveragevariance()` (line 53) calculates mean and variance of the file defined in line 50. The function `mcpercentiles()` (line 56) computes the percentiles specified in the list in line 55. All these estimators of uncertainty are automatically saved to disk, so that they can be viewed with `Aguila`.

RESULTS

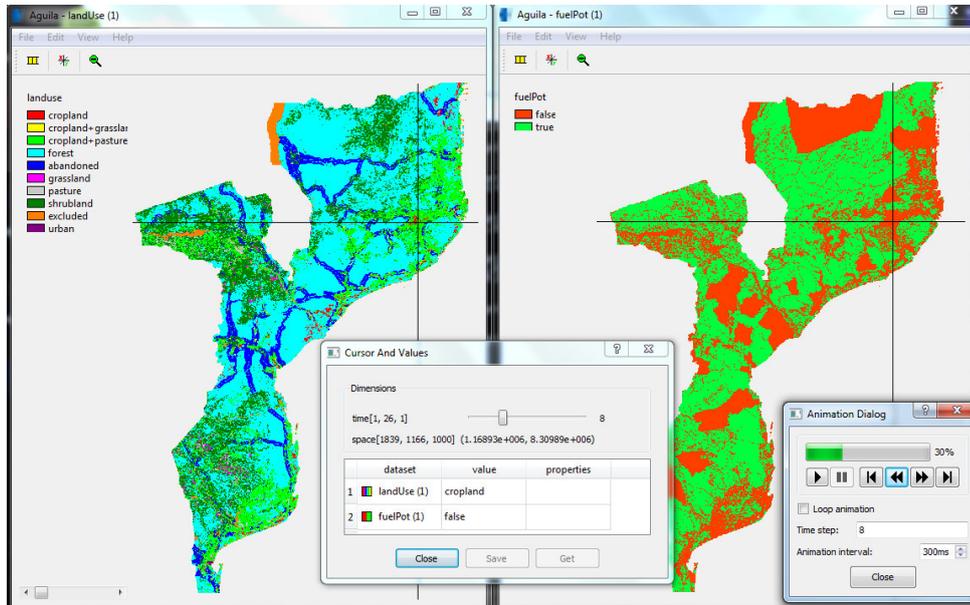


Figure 1: Screenshot of `Aguila` showing land use (left map) and bioenergy crop potential b_t (right map), in which true is available and false is unavailable, in Mozambique of one Monte Carlo sample in 2012 ($t = 8$) and the values of the cell that is selected with the cursor (centre window).

Communicating Uncertainty in Spatial Decision Support Systems: a Case Study of Bioenergy-Crop Potentials in Mozambique

A Monte Carlo run of 50 samples was carried out to determine land use change in Mozambique from 2005 to 2030. A raster land use map with cells of 1 x 1 km was used. Ten land use types were specified of which five were assigned product demands and accordingly expanded or contracted through time. The resulting change of land use and the area that is left for bioenergy crops in a single sample can be animated over time with Aguila. An example of this is given in Figure 1.

The maps in Figure 1 are examples of what is usually the output of an SDSS. They only hold information of a single realization. If a bioenergy crop plantation is planned on a currently available spot, the availability of that location after some years can be checked with the right map in Figure 1, which is based on that single realization. But consequently it does not provide any information on the certainty of the output; in the next realization both location of availability and the total area of available land can be quite different. In contrast, the PCRaster Python framework allows automatic calculation and easy visualization of uncertainty. As a result, Figure 2 presents the *probability* that a location is still not occupied by food crops or pasture, based on averaging over all 50 realizations, so that future competition with those land uses can be certainly avoided or reduced to a minimum probability. The purple (not available) and red (available) areas are certain, but in between there is uncertainty in the availability of land for bio energy crops. Outputs of different realizations can also be viewed separately. Figure 3 shows all 50 results for the total potential yield of bioenergy crops (ton). As can be seen, the yield that can be reached becomes substantially more uncertain over time. All this extra information is obtained by addition of only a few lines of code to the land use change model.

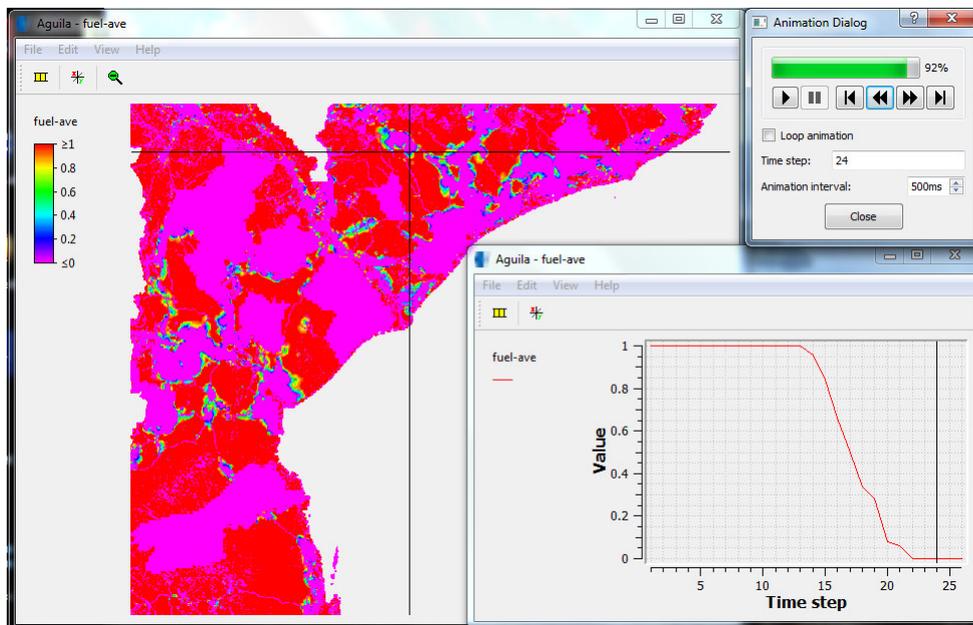


Figure 2: Screenshot of Aguila showing the probability on land availability for bioenergy crops in the central part of Mozambique in 2028 (time step 24) (map), and development of this probability through time of the cell that is selected with the cursor (graph in the lower right corner).

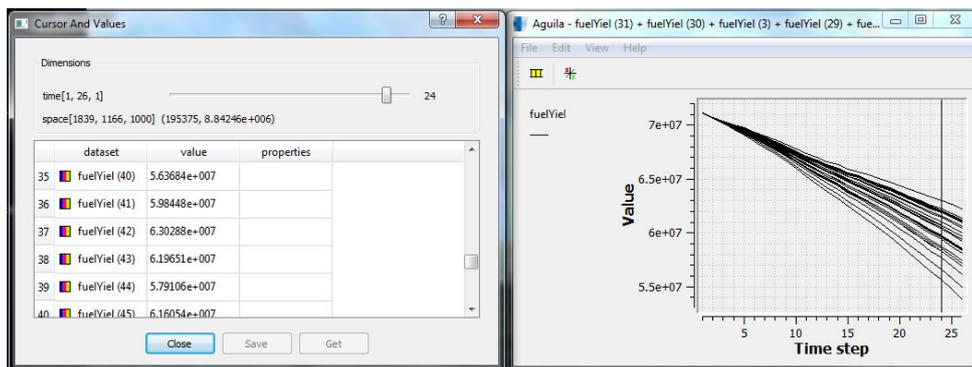


Figure 3: Screenshot of AguilA showing all 50 realizations of the total possible bioenergy crop yield (right window) and the corresponding yield values (ton) for 2028 (time step 24) (left window).

DISCUSSION AND CONCLUSIONS

We have shown how a modeller can use the PCRaster Python framework to construct a Spatial Decision Support System that integrates simulation, uncertainty analysis and visualization. This is considered very useful in the light that current SDSSs tend to ignore uncertainty (Foody, 2003). We claim that the output maps and graphs of uncertainty and its distribution in space and time provide an intuitive way for policy makers to take uncertainty into account in their decisions. However, in order to verify this claim, the policy makers themselves have to be asked for their opinions. Nevertheless, it is important that uncertainty in simulation models, which grow ever more complex, is somehow evaluated and communicated. The PCRaster Python framework facilitates this almost without any additional work on the modellers side, which is a major step forward in the exposure of uncertainty in SDSSs.

REFERENCES

- Aerts, J.C.J.H., Goodchild, M.F. and Heuvelink, G.B.M., 2003. Accounting for Spatial Uncertainty in Optimization with Spatial Decision Support Systems. *Transactions in GIS*, 7(2): 211-230.
- Arndt, C., Benfica, R., Tarp, F., Thurlow, J. and Uaiene, R., 2010. Biofuels, poverty, and growth: a computable general equilibrium analysis of Mozambique. *Environment and Development Economics*, 15: 81-105.
- Baas, A.C.W., 2002. Chaos, fractals and self-organization in coastal geomorphology: Simulating dune landscapes in vegetated environments. *Geomorphology*, 48(1-3): 309-328.
- Batidzirai, B., Faaij, A.P.C. and Smeets, E.M.W., 2006. Biomass and bioenergy supply from Mozambique. *Energy for sustainable development*, X(1): 28.
- Bradshaw, G.A. and Borchers, J.G., 2000. Uncertainty as information: narrowing the science-policy gap. *Conservation Ecology*, 4(1): [online] URL: <http://www.consecol.org/vol4/iss1/art7/>
- Brandt, J., Christensen, J.H., Frohn, L.M. and Zlatev, Z., 2000. Numerical modelling of transport, dispersion, and deposition - Validation against ETEX-1, ETEX-2 and chernobyl. *Environmental Modelling and Software*, 15(6-7 SPEC. ISS): 521-531.
- Brown, D.G., Page, S., Riolo, R., Zellner, M. and Rand, W., 2005. Path dependence and the validation of agent-based spatial models of land use. *International Journal of Geographical Information Science*, 19(2): 153-174.
- 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 and Geosciences*, 33(2): 172-190.

Communicating Uncertainty in Spatial Decision Support Systems: a Case Study of Bioenergy-Crop Potentials in Mozambique

- Chang, N.B., Parvathinathan, G. and Breeden, J.B., 2008. Combining GIS with fuzzy multicriteria decision-making for landfill siting in a fast-growing urban region. *Journal of Environmental Management*, 87(1): 139-153.
- D'Ambrosio, D., Iovine, G., Spataro, W. and Miyamoto, H., 2007. A macroscopic collisional model for debris-flows simulation. *Environmental Modelling and Software*, 22(10): 1417-1436.
- Eckhardt, K., Breuer, L. and Frede, H.-G., 2003. Parameter uncertainty and the significance of simulated land use change effects. *Journal of Hydrology*, 273: 164-176.
- FAO, 2003. World agriculture towards 2015/2030 an FAO perspective, Food and Agriculture Organisation, Rome.
- Foody, G.M., 2003. Uncertainty, knowledge discovery and data mining in GIS. *Progress in Physical Geography*, 27(1): 113-121.
- Geertman, S., 2006. Potentials for planning support: a planning-conceptual approach. *Environment and Planning B: Planning and Design*, 33: 863-880.
- Geertman, S. and Stillwell, J., 2004. Planning support systems: An inventory of current practice. *Computers, Environment and Urban Systems*, 28(4): 291-310.
- Goovaerts, P., 2010. Geostatistical Software. In: M.M. Fischer and A. Getis (Editors), *Handbook of Applied Spatial Analysis*. Springer, Heidelberg, pp. 129-138.
- Gosling, S.N. and Arnell, N.W., 2010. Simulating current global river runoff with a global hydrological model: model revisions, validation, and sensitivity analysis. *Hydrological Processes*: n/a.
- Heuvelink, G.B.M., 1998. Error propagation in environmental modelling with GIS. *Error propagation in environmental modelling with GIS*.
- Ivanovic, R.F. and Freer, J.E., 2009. Science versus politics: truth and uncertainty in predictive modelling. *Hydrological Processes*, 23: 2549-2554.
- Jerolmack, D.J. and Paola, C., 2007. Complexity in a cellular model of river avulsion. *Geomorphology*, 91(3-4): 259-270.
- Karssenber, D. and De Jong, K., 2006. Towards improved solution schemes for Monte Carlo simulation in environmental modeling languages. In: P.J.M. Oosterom (Editor), *Geo-Information and computational geometry*. NGC (Nederlandse Commissie voor Geodesie, in English: Netherlands Geodetic Commission), Delft.
- Karssenber, D., de Jong, K. and van der Kwast, J., 2007. Modelling landscape dynamics with Python. *International Journal of Geographical Information Science*, 21(5): 483-495.
- Karssenber, D., Schmitz, O., Salamon, P., de Jong, K. and Bierkens, M.F.P., 2010. A software framework for construction of process-based stochastic spatio-temporal models and data assimilation. *Environmental Modelling & Software*, 25: 489-502.
- Manson, S.M., 2007. Challenges in evaluating models of geographic complexity. *Environment and Planning B: Planning and Design*, 34: 245-260.
- Nearing, M.A. et al., 2005. Modeling response of soil erosion and runoff to changes in precipitation and cover. *Catena*, 61(2-3 SPEC. ISS.): 131-154.
- NetLogo, 2010. NetLogo website, Available online at: <http://ccl.northwestern.edu/netlogo/>.
- Oreskes, N., Shrader-Frechette, K. and Belitz, K., 1994. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263(5147): 641-646.
- PCRaster, 2010. PCRaster internet site, Available online at: <http://pcraster.geo.uu.nl>.

AGILE 2011, April 18-22: Judith Anne Verstegen, Floor van der Hilst, Derek Karssenberg, André Faaij

- Pebesma, E.J., de Jong, K. and Briggs, D., 2007. Interactive visualization of uncertain spatial and spatio-temporal data under different scenarios: An air quality example. *International Journal of Geographical Information Science*, 21(5): 515-527.
- Python, 2010. Python programming language, Available online at: <http://www.python.org>.
- Smeets, E.M.W., Faaij, A.P.C. and Lewandowski, I.M., 2004. A quickscan of global bio-energy potentials to 2050. An analysis of the regional availability of biomass resources for export in relation to the underlying factors. Report NWS-E-2004-109, ISBN 90-393-3909-0, Copernicus Institute, Utrecht University, Utrecht.
- Stella, 2010. Stella website, Available online at: <http://www.iseesystems.com>.
- UNDP, 2008. World Population Prospects: The 2008 Revision. Population Division of the Department of Economic and Social Affairs of the United Nations Secretariat.
- van der Hilst, F., Faaij, A., Karssenberg, D. and Verstegen, J.A., forthcoming. Land use change modeling for the assessment of land availability for energy crops: the case of Mozambique. In prep.
- Verburg, P.H. and Overmars, K.P., 2009. Combining top-down and bottom-up dynamics in land use modeling: Exploring the future of abandoned farmlands in Europe with the Dyna-CLUE model. *Landscape Ecology*, 24(9): 1167-1181.
- Watson, H., 2010. Potential to expand sustainable bioenergy from sugarcane in southern Africa. submitted.