International Environmental Modelling and Software Society (iEMSs)
2012 International Congress on Environmental Modelling and Software
Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany
R. Seppelt, A.A. Voinov, S. Lange, D. Bankamp (Eds.)
http://www.iemss.org/society/index.php/iemss-2012-proceedings

Semantic Array Programming for Environmental Modelling: Application of the Mastrave Library

Daniele de Rigo [1,2]

[1] *Politecnico di Milano, Dipartimento di Elettronica e Informazione, Via Ponzio 34/5, I-20133 Milano, Italy. derigo@elet.polimi.it*
[2] *Joint Research Centre of the European Commission, Institute for Environment and Sustainability, Via Fermi, 2749, I-21027 Ispra, Italy. daniele.de-rigo@jrc.ec.europa.eu*

Abstract: Environmental datasets grow in size and specialization while models designed for local scale are often unsuitable at regional/continental scale. At regional scale, data are usually available as georeferenced collections of spatially distributed despite semantically atomic information. Complex data intrinsically impose modellers to manipulate nontrivial information structures. For example, multi-dimensional arrays of time series may be composed by slices of raster spatial matrices for each time step, whilst heterogeneous collections of uneven arrays are common when dealing with data analogous to precipitation events, and these structures may ask for integration at several spatial scales, projections and temporal extents. Interestingly, it might be far more difficult to practically implement such a complexity rather than conceptually describe it: a subset of modelling generalizations may deal more with abstraction rather than with the explosion of lines of code. Many environmental modelling algorithms are composed by chains of data-transformations or trees of domain specific sub-algorithms. Concisely expressing them without the need for paying attention to the enormous set of spatio-temporal details is a highly recommendable practice in both mathematical formulation and implementation. The *semantic array programming* paradigm is here exemplified as a powerful conceptual and practical (with the free software library Mastrave) tool for easing scalability and semantic integration in environmental modelling. Array programming (AP) is widely used for its computational effectiveness but often underexploited in reducing the gap between mathematical notation and algorithm implementations, i.e. by promoting arrays (vectors, matrices, tensors) as atomic quantities with extremely compact manipulating operators. Coherent array-based mathematical description of models can simplify complex algorithm prototyping while moving mathematical reasoning directly into the source code – because of its substantial size reduction – where the mathematical description is actually expressed in a completely formalized and reproducible way. The proposed paradigm suggests complementing the characteristic AP weak typing with semantics, both by composing generalized modular sub-models and via array oriented – thus concise – constraints. The Mastrave library use is exemplified with a regional scale benchmark application to *local-average invariant* (LAI) downscaling of climate raster data. Unnecessary errors frequently introduced by non-LAI upsampling are shown to be easily detected and removed when the scientific modelling practice is terse enough to let mathematical reasoning and model coding merge together.

***Keywords**:* environmental datasets; regional scale; data-transformation modelling; semantic array programming; local-average invariant downscaling.

1 INTRODUCTION

Environmental modelling is inherently a multidisciplinary scientific activity which needs the coordination of a wide range of competences. Data modelling and integration in environmental sciences are actively and increasingly investigated within

information and communications technology [Casagrandi and Guariso 2009]. Environmental datasets are growing in size and specialization [Edwards 2004; Hijmans *et al.* 2005; Yesson *et al.* 2007; Scholes *et al.* 2008] while models designed for local scales are often unable to work at regional/continental scale[1] especially when local-scale data requirements exceed or are unsuitable to exploit data availability at larger scale [Merritt *et al.* 2003] – despite large-scale analysis may be essential for assessing, planning and managing natural resources [Loarie *et al.* 2009; Angelis-Dimakis *et al.* 2011; de Rigo, 2012b]. At regional scale, data are usually available as geographically referred collections of information distributed in space and time, such as raster layers of matrices each element of them refers to a given time, spatial location and resolution, possibly along with corresponding spatially distributed reliability information. Complex data intrinsically impose modellers to manipulate non-trivial information structures. For example, multi-dimensional arrays of time series of measures, geographically referred, may be composed by slices of raster matrices for each time step [Haylock *et al.* 2008; van den Besselaar *et al.* 2011], whilst heterogeneous collections of uneven arrays are common when dealing with data analogous to precipitation events (e.g. derived from detailed time series of measured precipitation so to remove low- or zero-intensity periods [de Rigo and Bosco 2011]).

Many environmental modelling algorithms are composed by chains of data-transformation trees of often loosely coupled or orthogonal sub-algorithms, focused on domain-specific aspects whose interaction contributes to better constrain and characterize the modelled quantities [Crowley and Hyde 2008; Lovett *et al.* 2009; Haughton *et al.* 2009; Hashimoto *et al.* 2011]. Concisely expressing them without the need for paying attention on the enormous set of spatial details, is a highly recommendable practice in both mathematical formulation and implementation.

This paper exemplifies the emerging paradigm of *semantic array programming* (SAP) [de Rigo 2012d] as a powerful conceptual and practical tool (with the free software library Mastrave[2]) for easing scalability and semantic integration within environmental modelling. The focus is toward facilitating the implementation and evolution of the mathematics underlying complex models by means of concise, abstract and semantically enhanced notations and tools. The *Mastrave modelling library* implements the SAP paradigm allowing derived domain-specific models to share common semantic notation and existing non-SAP models to be integrated through lightweight semantic enhancement so to strengthen their robustness in an unobtrusive way. An example of the mutual reinforcement between concise array-based notation and semantic constraints will be described, and a regional (continental) scale application to local-average invariant (LAI) downscaling of benchmark climate raster data will serve to further illustrate the approach.

1.1 The Mastrave Modelling Library

Mastrave is a free software [Stallman 2009] library written to perform SAP and to be as compatible as possible with both GNU Octave [Eaton *et al*. 2008] and MATLAB[3] computing frameworks. The GNU Bash shell [Ramey and Fox 2006] is also transparently integrated[4], to take advantage of some of its relevant stable and almost universally portable features – which have been headed toward the AP paradigm.

Mastrave is mostly oriented to ease complex modelling tasks such as those typically needed within environmental models, even when involving irregular and heterogeneous data series. Since 2005, the Mastrave library supports designing and implementing environmental modelling applications. Examples of applications range from evolutionary techniques for nontrivial parameter training (the SIEVE architecture [*Selective Improvement by Evolutionary Variance Extinction*], applied to [de Rigo *et al.* 2005] approximate dynamic programming in water resources [de Rigo

[1] An opposite phenomenon also affects modelling integration and robustness [Trivedi et al. 2008].
[2] The Mastrave Project, http://mastrave.org/ .
[3] The MathWorks: MATLAB, http://www.mathworks.it/help/techdoc/ .
[4] Other languages – not as concise in supporting AP – could be enriched with minimal SAP features: experimental extensions to the Mastrave core exist to create a common basis for reasoning and encouraging mutual paradigm contamination. Transparent bindings for providing array semantic constraints within Python [Van Rossum and Drake 2011] with NumPy [Oliphant 2006] (http://numpy.scipy.org/) and GNU R [Venables and Smith 2009] (http://www.r-project.org/) are under experimentation. Despite a GNU/Linux like environment is required and supporting free operating systems takes precedence over non-free platforms, Mastrave can run within Cygwin (http://www.cygwin.com/) under Windows.

et al. 2001]), up to on-line policy design for water reservoir networks [Castelletti *et al.* 2008]; from a modelling architecture for evaluating at continental scale potential and actual soil water erosion [de Rigo and Bosco 2011; de Rigo and Bosco (in prep.); Bosco *et al.* (in prep.)], up to several forest resource applications, such as detailed European forest tree species distribution modelling [de Rigo *et al.* (in prep.)], concise graph-based formulation [Estreguil *et al.* 2012] and nonlinear statistical analysis [de Rigo 2012c; de Rigo (submitted)] of heterogeneous spatial pattern indices for characterizing forest habitats [Estreguil *et al.* (in prep.)].

The author explicitly conceived the Mastrave library for supporting nontrivial data-transformation models which typically require the active involvement of experienced modellers. Nontrivial data transformations are usually subjected to scientific peer review as original contributes [Knuth 1974] to environmental modelling. This subset of data-transformation models may easily be suitable and convenient to be reused as precious components of new environmental models. Their role may be to access the information of available datasets by aggregating, filtering, slicing collections of data and by composing multiple datasets for approximating missing information. However, understandability, expressiveness and sustainable maintainability (e.g. abstraction, ease of modification and innovation) of both these models and their libraries may be essential in deciding their long-term survival. Suggestively, Stroustrup [2005] highlights "that on the order of 200 new languages are developed each year and that about 200 languages become unsupported each year".

Besides strictly limiting dependencies to reliable and actively developed free software packages, documentation is also vital. The Mastrave knowledge management policy is to directly update thorough documentation within source code as semantically enhanced structured comments[5] part of a consistent set of coding standards. Online documentation (http://mastrave.org/doc/) only refers to stable modules – for which usually expert users provided feedback. Each module report has a permanent URL safely citable within scientific publications and is automatically updated to persistently be in line with the latest published module version. Examples of usage systematically highlight the abstraction extent of each module.

Interestingly, it might be far more difficult to practically implement several complex models rather than conceptually describe them: a subset of modelling generalizations may deal more with abstraction rather than with the explosion of lines of code [McGregor 2006; Smaalders, 2006; Wilson 2006]. For such a subset of modelling applications, an approach in which "the advantages of executability and universality found in programming languages can be effectively combined, in a single coherent language, with the advantages offered by mathematical notation" [Iverson 1980] might help.

2 SEMANTIC ARRAY PROGRAMMING

Array programming originated [Iverson 1980] to reduce the gap between mathematical notation and algorithm implementations by promoting arrays (vectors, matrices, tensors) as atomic quantities with extremely concise manipulating operators. Coherent array-based mathematical description of models can simplify complex algorithm prototyping while moving mathematical reasoning directly into the source code – because of its substantial size reduction – where the mathematical description is actually expressed in a completely formalized and reproducible way [Eden 2007; Nature 2011]. Two additional design concepts define semantic array programming [de Rigo 2012d] as supported by the Mastrave library:

1. *Modularizing* sub-models and autonomous tasks with a strong effort toward their *most concise generalization* and reusability in other contexts.
2. *Semantically constraining* with array oriented –thus concise– invariants the information entered in and returned by each module instead of relying on external assumptions.

Modularization also expects consistent code self-documentation and uniform predictable conventions for module interfaces (without directly interfering with the preferred module's implementation). Semantic constraints contribute enforcing within each module autonomous distributed consistency checks instead of assuming top-down correctness of input information (e.g. instead of relying on object-oriented "monolithically-designed-to-be-safe" data). This way, even the residual monolithic

[5] Analysing more than 2500 free software projects using MATLAB language, 23% of all lines of code are comments, while the share is over 50% for Mastrave's GNU Octave/MATLAB code (https://www.ohloh.net/languages/matlab , https://www.ohloh.net/p/mastrave/analyses/latest).

management of data structures for exchanging information between modules is minimized up to quite general and *language-neutral* data primitives: multi-dimensional numerical arrays and collections of generic elements (cell-arrays).

2.1 An Example: Semantic Array-based Manipulation of Time Series

Array-oriented semantic constraints show similarities with behavioural subtyping [Liskov and Wing 2001], however without intrinsic links with object oriented programming. An example will highlight the way a SAP module interacts with external information provided by the module input arguments. Within the MATLAB language[6] (which is dynamically and weakly typed) the concepts of vector, matrix and multidimensional array can all be represented by the native type *double* – which can also represent complex numbers.

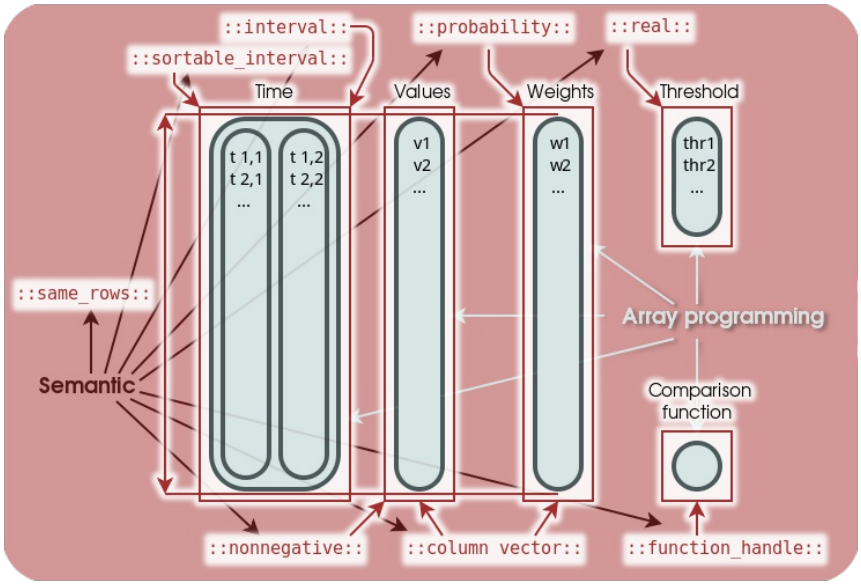

Figure 1. Array-oriented constraints determining at run-time the semantic structure of a time series, from the minimal preconditions (such as **`::same_rows::`**) up to submodel-specific requests (e.g. time interval sortability, values nonnegativity, ...).

A variable of type double can be an arbitrary array $\in \mathbb{C}^{n_1} \times \cdots \times \mathbb{C}^{n_m}, \forall n_1 \cdots n_-$ where $\mathbb{C}$ is the set of 64 bit floating point complex numbers extended with $\pm\infty$ and `nan` (i.e. "not a number" denoting undefined or missing data) partially following the standard IEEE 754 for floating-point arithmetic [IEEE 1985, 2008]. Array operators concisely apply to these variables as if they were atomic, almost completely removing the need for explicit loops. As arrays are considered as a basic type, array variables can legitimately – and silently… – be assigned with other arrays of heterogeneous size. Despite this abstraction offers a powerful and extremely concise notation, complex array-based algorithms and models still may consist of hundreds or thousands of code lines, for which good practices such as assertions and ad-hoc consistency checks may tend to proliferate. This however poses relevant issues on the sustainability and effectiveness of such intensely needed "DIY" ad-hoc solutions. An example will illustrate how array-based, concise and consistent semantic constraints may help to immerse concise pieces of AP code within a robust semantic network of array-concepts.

Manipulating time series of data (Figure 1) is a frequent task to be addressed by specialized sub-models. For example, a rainfall time series may semantically be represented by a vertical set of time intervals, a corresponding set of rainfall values and a third sequence of weights – e.g. from 0 to 1 – providing a reliability indication for each value. Supposing the exemplified module needs to separately sort by date the time-series elements whose weights lie within a certain set of thresholds, then a collection of preconditions (each denoted as ***`::semantic-constraint::`***[7]) should be satisfied:

[6] Descriptions and examples apply to both GNU Octave and MATLAB computing environments.
[7] The report on the **`check_is`** module [de Rigo 2012a] lists some 200 native semantic constraints.

1. the time series elements (time intervals, values and weights) must be composed by the same number of items (rows): `::same_rows::` ;
2. both values and weights arrays must be `::col_vector::`s;
3. values must be of real non-negative numbers (`::nonnegative::`) while weights must be in [0, 1] (`::probability::`);
4. the time-interval array is a two-column matrix whose second-column elements cannot be less than to the corresponding first column ones (`::interval::`);
5. intervals are required to be sortable, so that[8] the intersection of any pair of intervals must be empty (`::sortable_interval::`);
6. thresholds must be real numbers (`::real::`) and a custom comparison function is passed as optional module's input argument (`::function_handle::`).

Complexity of constraints' semantics varies from case to case and several constraints can only be tested at run-time. Constraints apply to vectors, matrices and often to multi-dimensional arrays, easing their concise application to complex data, so that nontrivial networks of semantic constraints (preconditions, invariants and postconditions) could be straightforward to describe. Unsatisfied constraints raise exceptions which can be managed to enable model self-healing, or simply can cause a human-readable and self explaining error to abort the computation. The Mastrave module **`check_is`** implements constraints so to make them available as a rich, consistent, flexible and language-neutral set of semantic assertions systematically used and documented by all Mastrave modules.

Semantic array-programming filters may be used to semantically wrap pre-existing third-party models, since constraint errors can be easily exposed and used for strengthening third-party models' robustness. After this small example, the following section will introduce a wider SAP application to *local-average invariant* (LAI) downscaling of climate raster data.

2 Application To Local-Average Invariant Downscaling

General circulation models are widely used for estimating patterns of temperature and precipitation under climate change scenarios. Their coarse spatial resolution is often unsuitable for direct use within environmental models and can be mitigated by complementing large-scale information with that provided by regional climate models and local-scale data [Bates *et al.* 1998; Patz *et al.* 2005; Fowler *et al.* 2007].

Among the simplest downscaling methods, the *perturbation-method* [Prudhomme et al. 2002] or *delta-change*, *change-factor* method has been proposed for downscaling future climate projections at continental scale, up to a spatial resolution of 30 arc-second (less than 1 km at the Equator) [Ramirez and Jarvis 2010; Tabor and Williams 2010]. The method exploits the availability for the present (baseline) of information at a finer resolution than the one available for future climate scenario projections. Given a coarse-scale projection of climate *changes*, it is first upsampled by applying a smooth interpolation and then added to a higher resolution baseline. The result estimates the climate projection at the finer scale. Despite its simplicity, the change-factor method has been recommended for strategic assessment of multiple climate change scenarios [Wilby *et al.* 2004] and appears as well performing as more sophisticate methods in reproducing mean characteristics [Fowler *et al.* 2007].

An evident weakness of the method may lie in the application of over-simplistic interpolation strategies which may degrade the already coarse original climate change information. For example, smoothing splines are often employed, probably also because of their out-of-box availability. For each coarse-resolution spatial cell, global or regional climate models generally provide a value representing the average of the modelled climatic quantity within that cell. However, downscaling interpolations are often performed by confusing the (known) *average* value of a given cell of the coarse grid to be upsampled with the corresponding (unknown) *punctual* value of the cell centroid[9], which is required by the spline. As a consequence, if the

[8] The definition of intervals' sortability may be customized, depending on the specific task for which an ordering is required. However, an easy and unambiguous sufficient condition holds when all intervals are disjoint (`::sortable_interval::`).

[9] More properly, the "punctual" value is the one of the finer-grid cell in which the centroid of the coarse-grid cell lies. It is obviously *unknown* before the downscaling, and even if it were known, still

downscaling data-transformation model were reversed and from the interpolated fine-resolution values the corresponding coarse-cell averages were re-computed, they *would not respect the original coarse-scale climate change values*.

Several strategies[10] can be followed in order for the downscaling data-transformation modelling to locally preserve as average the values of the original raster data. A downscaling data-transformation fulfilling the constraint to be *local-average invariant* (LAI) is here denoted as LAI-downscaling. Non-LAI spline interpolation is natively supported by the Mastrave function **`upsample`** , which also provides LAI interpolations.

3.1 Importing Remote Data

Several datasets provide raster data tiled per geographical area or time interval [Bossard *et al.* 2000; Farr *et al.* 2007; WorldClim 2012]. Importing these data may require downloading, decompressing and transforming projections and file formats to fit the needs of the computational modelling research. For example, intense array oriented manipulations of data benefit from accessing synchronously a plethora of large files in O(1) time, read/write mode. These activities may imply manipulating a noticeable amount of original, temporary and working files. Mastrave supports importing remote data with a single command. Exemplifying with the WorldClim dataset, the command:

```
uri = @(files)[ 'http://www.worldclim.org/data/v1_4/tiles/cur/' ...
                files '_16_tif.zip'                              ]
mastrave( 'preload', uri( '{t{min,mean,max},alt,prec}' ) )
[filepath, filename, bytes] = fileinfo( '*.bt' )
```

downloads the compressed archives of altitude, precipitation, temperature (monthly minimum, mean and maximum) of the WorldClim tile 16, extracts some 50 geotiff files and converts them in Binary Terrain format[11]. The whole set of tiles might be accessed by replacing `'_16_tif.zip'` with `'_{0..4}{0..11}_tif.zip'` so that in this case almost 3000 geotiff raster data could be pre-loaded with a single command, by default enforcing a (customizable) strategy for avoiding replicated downloads. The **`fileinfo`** module eventually allows generating a list of available data while inspecting their size.

3.2 Interpolation Errors of non-LAI Downscaling

Climate change [Broome 2008; Tol 2009; Jasanoff 2010] involves complex relationships among seasonal and historic variability [Loarie *et al.* 2009]. The qualitative analysis of the interpolation errors due to the use of non LAI downscaling has been performed looking for a simplified benchmark example with monthly information, easy to reproduce and offering a reasonable numerical variety, despite completely decoupled from actual climate change projections. The month-to-month (m-m) differences between the WorldClim 1950-2000 mean temperatures (in North Africa and part of the Europe) of July and June, and of February and January, are shown in Figure 2.

They have been used as benchmark examples due to their complementary, almost opposite patterns of changes which could allow the numerical effects of downscaling a wide range of possible climate changes to be qualitatively simulated. The average intensities of the two benchmark temperature changes are the two m-m annual local minima with positive value, and lie below almost all the projected temperature anomalies between 2000 and 2100. For upsampling (with both usual non-LAI cubic splines and a LAI interpolator), the Mastrave function **`upsample`** has been used. The function allows a series of raster-data to be processed with a single invocation even if they are multi-dimensional arrays. 30 arc-seconds is the original WorldClim resolution, preserved for the benchmark baseline. In order for the benchmark to

---

the problem of imposing the local average invariance (LAI) of the smoothing spline across each coarse-grid cell would remain.

[10] For example, iterative smoothing of residuals with usual splines with a final LAI interpolation to exactly impose the constraint; or deconvolution of a smoothed non LAI interpolation with high differentiability class. For details, see the **`upsample`** function documentation.

[11] http://vterrain.org/Implementation/Formats/BT.html . This file format is natively supported e.g. by the GDAL library [2012] and is straightforward to be accessed (read/write mode) in constant time.

simulate a coarser change-factor grid[12], the mean of all non-missing data within blocks of 25 x 25 cells is simply computed with the Mastrave function **`mblk_fun`**.

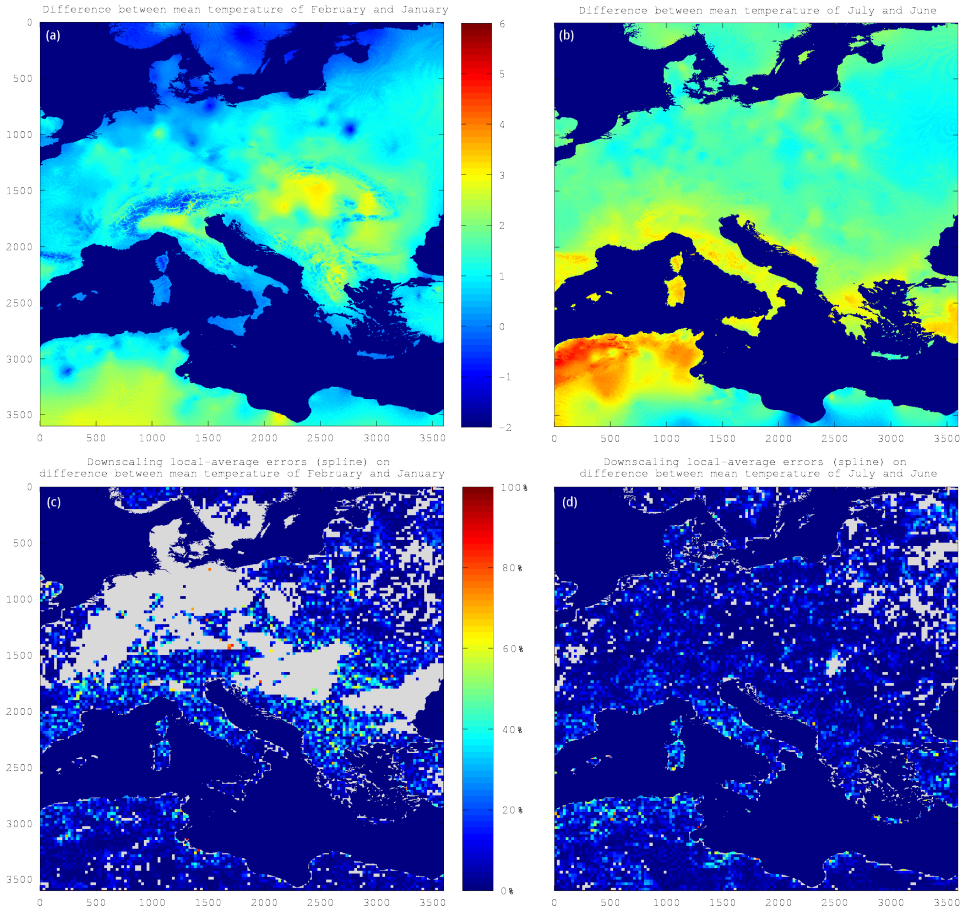

Figure 2. Benchmark difference (month-to-month) in °C/month between mean temperature [WorldClim 2012] of (a) February and January, and (b) July and June. The corresponding dimensionless ratio – (c) and (d) – between non-LAI interpolation local average errors (which using LAI approaches would be zero) and usual difference errors between downscaled (via cubic splines) and "true" future values, is computed on the dense-grid and averaged in the coarse-grid. Grey pixels indicate where both errors are within the dataset uncertainty (±0.05 °C). These errors are unnecessarily introduced with non LAI smoothing: LAI-downscaling mathematically ensures the error ratio maps to be null. Spatial units: 30 arc-seconds lat-lon with translated origin (top left).

It enables separately processing rectangular blocks of arbitrary size (here, 25 x 25 pixels) of a given raster grid, with arbitrary functions (here, the function mean is used) passed as input argument:

```
mblk_fun( raster_data, @(x)mean(x(~isnan(x))), 25 )
```

whose meaning is straightforward when knowing that `~` is the logical negation and `@(x)...` is an anonymous function with `x` as input argument ("mean of `x` without its `nan` values"). The ratio $\varepsilon^{\text{non-LAI}}/\varepsilon^{\text{down}}$, (dimensionless) between non-LAI interpolation errors $\varepsilon^{\text{non-LAI}}$ (absolute difference between each change-factor's coarse-grid pixel and the corresponding average of 25 x 25 non-LAI interpolated pixels of the dense-grid) and the 25 x 25 average of the absolute dense-grid errors between the interpolated and the "true" future values, $\varepsilon^{\text{down}}$, is shown in Figure 2 (c,d). Using LAI interpolation, $\varepsilon^{\text{non LAI}}$ would have been mathematically imposed to be null, so to also force (c,d) to be zero. This highlights the non negligible errors which non-LAI smoothing unnecessarily introduces in addition to the intrinsic downscaling error $\varepsilon^{\text{down}}$. While estimating $\varepsilon^{\text{down}}$ is

[12] This is done while still preserving the possibility of comparing modelled results with the "true" future denser grid: which, outside this benchmark simulation, would typically be impossible.

generally impossible in non-benchmark applications of the change-factor method, $\varepsilon^{non\text{-}LAI}$ is always computable even as a SAP semantic constraint.

Computing less trivial statistics such as the velocity of the temperature change [Loarie *et al.* 2009] (km month^{-1}) with the benchmark data, would just need a few additional lines of code (i.e. a compact data-transformation codelet):

```
check_is( {T_Jan T_Feb lat lon}, 'same_size' )% semantic check: minimal example
noise = rand( size( T_Jan ) )*0.1-0.05;        % WorldClim T uncertainty: 0.1 °C
k = [1;2;1]*[-1:1]                             % slope kernel
[dx,dy] = deal( mfilter( {k, k'}, T_Jan + noise, [] ){:} );
lat2km = lat * 111.325                         % conversion latitude-degree → km
lon2km = cos( pi/180 * lon ) * 111.325         % conversion longit.-degree → km
T_speed = ( T_Feb - T_Jan ) ./ hypot( dx.*lat2km, dy.*lon2km )
```

where `T_Jan, T_Feb, lat, lon` are the mean WorldClim temperatures of January and February, and the latitude and longitude raster data in degrees. The velocity of change is defined as the ratio between the temporal gradient (here `T_Feb - T_Jan` ºC month^{-1}) and the spatial gradient (the module of the vector with components `dx` and `dy`, ºC km^{-1}). Each pixel is added with uniformly distributed random noise in ±0.05 °C for mitigating – within the dataset uncertainty which for WorldClim temperature is 0.1 °C – "the incidence of flat spatial gradients that cause infinite speeds", as suggested by Loarie *et al.* [2009].

Comparing this code (also the Mastrave **`mfilter`**) with the mathematical description in Loarie *et al.* [2009] easily shows that conciseness could also enhance model understandability. Outside benchmark examples, computing the velocity of change for a given quantity between current and future scenarios could typically use coarser change-factors for estimating fine-scale future information: in this case, climate change upsampling should be addressed with LAI downscaling.

Conclusions And Perspectives

Semantic array programming (SAP) is proposed as a powerful conceptual and practical tool (supported by the Mastrave library) for easing scalability and semantic integration in environmental modelling, by complementing extremely concise manipulating operators with modularization and compact semantic constraints. Mastrave usage has been exemplified with a regional scale benchmark application to *local-average invariant* (LAI) downscaling of climate raster data, whose straightforward use in Mastrave could help to promote LAI downscaling as a more correct approach for upsampling grids of climatic spatial averages. SAP concise data-transformation codelets, made available as free software, can naturally contribute supporting a smooth transition toward reproducible research [Morin *et* al. 2012; Peng 2011; Stodden 2012, 2011; YLSRDCS 2010; De Leeuw 2001].

From a general perspective and revisiting a classic framework for Information Technology benefits [Maggiolini, 2011], SAP conciseness – as implemented in Mastrave – may positively affect several environmental-modelling aspects: form reducing new models' *production* costs (because of its reusable modules and the drastic decrease of code lines inherent to adopting the AP paradigm), to mitigating *coordination* costs among data and sub-models (because of coherent, compact abstraction and the ability to transparently import large remote data) finally also enabling *communication* economies (achieved through extremely compact, reliably tested semantics, also suitable to *semantically wrap* legacy models). These technical benefits should not shadow the strategic goal of resisting "pressure to privatize science", for "knowledge contributes to society when it can be shared and developed by communities" [Stallman 2005].

Acknowledgments

Mastrave is hosted by the GNU Savannah[13] free software forge and reviewed by the Free Software Directory[14]: the required code review improved the library structure and licensing. SAP was conceived by also considering the epistemological implications[15] of free software [Stallman 2005, 2009]. Many students, computational and

[13] http://savannah.nongnu.org/projects/mastrave
[14] http://directory.fsf.org/wiki/Mastrave

environmental modelling scientists provided valuable user's feedbacks along with the Maieutike Research Initiative volunteers. A metadata analysis on semantic data-transformation models is actively investigated with G. Guariso, while research is on-going on applications of LAI downscaling with C. Bosco, G. Caudullo and J.I. Barredo.

REFERENCES

Angelis-Dimakis, A., M. Biberacher, J. Dominguez, G. Fiorese, S. Gadocha, E. Gnansounou, E., G. Guariso, *et al.*, Methods and tools to evaluate the availability of renewable energy sources, *Renew. and Sust. Energy Reviews, 15*(2), 1182–1200, 2011.

Bates, B.C., S.P. Charles, and J.P. Hughes, Stochastic downscaling of numerical climate model simulations, *Environmental Modelling & Software, 13*(3–4), 325–331, 1998.

Bosco, C., D. de Rigo, J. Poesen, O. Dewitte and P. Panagos, Modelling soil erosion at European scale: harmonisation and reproducibility. In prep.

Bossard, M., J. Feranec, and J. Otahel, CORINE land cover technical guide - Addendum 2000, *Technical report No 40*, *European Environment Agency*, 2000.

Broome, J., The ethics of climate change. Scientific American 298 (6), 96-102, 2008.

Casagrandi, R., and G. Guariso, Impact of ICT in Environmental Sciences: A citation analysis 1990-2007, *Environmental Modelling & Software, 24*(7), 865–871, 2009.

Castelletti, A., D. de Rigo, L. Tepsich, R. Soncini-Sessa, and E. Weber, On-Line Design of Water Reservoir Policies Based on Inflow Prediction, *Proc. of the 17th IFAC World Congress*, 14540–14545, 2008.

Crowley, T.J. and W.T. Hyde, Transient nature of late Pleistocene climate variability, *Nature, 465*(7219), 226–230, 2008.

De Leeuw, J., Reproducible research: the bottom line, 2001.

de Rigo, D., Applying semantic constraints to array programming: the module "check_is" of the Mastrave modelling library. *Mastrave project tech. report*. 2012.

de Rigo, D., Behind the horizon of reproducible integrated environmental modelling at European-scale: ethics and practice of scientific knowledge freedom. In prep.

de Rigo, D., Detecting general multi-dimensional nonlinear correlations: the module "dist_corr" of the Mastrave modelling library. *Mastrave project tech. report*. 2012.

de Rigo, D., Semantic Array Programming with Mastrave - Introduction to Semantic Computational Modelling. The Mastrave project, to appear in 2012.

de Rigo, D., Multidimensional Distance Correlation Analysis with User-defined Metrics. Submitted to: *Free Software and Semantic Array Programming Research*.

de Rigo, D. and C. Bosco, Architecture of a framework for integrated natural resources modelling at regional scale: application to pan-European soil water erosion assessment. In prep.

de Rigo, D. and C. Bosco, Architecture of a Pan-European Framework for Integrated Soil Water Erosion Assessment, *Environ. Software Systems. Frameworks of eEnvironment, IFIP Advances in Information and Commun. Technology, 359*, Chapter 34, 2011.

de Rigo, D., A. Castelletti, A.E. Rizzoli, R. Soncini-Sessa, and E. Weber, A selective improvement technique for fastening Neuro-Dynamic Programming in Water Resources Network Management. *Proc. of the 16th IFAC World Congress*, 7–12, 2005.

de Rigo, D., A.E. Rizzoli, R. Soncini-Sessa, E. Weber, and P. Zenesi, Neuro-dynamic programming for the efficient management of reservoir networks. Proc. of MODSIM 2001, *International Congress on Modelling and Simulation, 4*, 1949-1954, 2001.

de Rigo, D., G. Caudullo, G. Amatulli, P. Strobl and J. San-Miguel-Ayanz, Modelling tree species distribution in Europe with constrained spatial multi-frequency analysis. In prep.

Eaton, J.W., D. Bateman, and S. Hauberg, GNU Octave Manual, *Network Theory Ltd.*, 2008.

Eden, A. H., Three paradigms of computer science, Minds and Machines 17 (2), 135-167, 2007.

Edwards, J.L., Research and Societal Benefits of the Global Biodiversity Information Facility. *BioScience, 54*(6), 486–487, 2004.

Estreguil, C., D. de Rigo, and G. Caudullo, Towards an integrated and reproducible characterisation of habitat patterns. Submitted to: *Environmental Modelling & Software*.

Estreguil, C., G. Caudullo, D. de Rigo, C. Whitmore, and J. San-Miguel-Ayanz, Reporting on European forest fragmentation: Standardized indices and web map services. *IEEE Earthzine*, IEEE Committee on Earth Observation (ICEO), 3rd Quarter Theme 2012.

Farr, T., P. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, et al., The Shuttle Radar Topography Mission. *Reviews of Geophysics, 45* RG2004, 2007.

Fowler, H.J., S. Blenkinsop, and C. Tebaldi, Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling, *International Journal of Climatology, 27*(12), 1547–1578, 2007.

Geospatial Data Abstraction Library, version 1.8, Open Source Geospatial Foundation, http://gdal.osgeo.org (accessed 2012-02-26).

Hashimoto, S., Morishita, T., Sakata, T., Ishizuka, S., Increasing trends of soil greenhouse gas fluxes in Japanese forests from 1980 to 2009, *Scientific Reports, 1, n.116*, 2011.

[15] http://mastrave.org/research.html provides information on volunteering within research collaborations and on expected scientific code review process and knowledge freedom.

Haughton, A.J., Bond, A.J., Lovett, A.A., Dockerty, T., Sünnenberg, G., Clark, S.J., Bohan, et al., A novel, integrated approach to assessing social, economic and environmental implications of changing rural land-use: a case study of perennial biomass crops, *J. of Applied Ecology, 46*(2), 315–322, 2009.

Haylock, M.R., N. Hofstra, A.M.G. Klein Tank, E.J. Klok, P.D. Jones, and M. New., A European daily high-resolution gridded dataset of surface temperature and precipitation, *J. Geophys. Res (Atmospheres), 113*, D20119, 2008.

Hijmans, R.J., S.E. Cameron, J.L. Parra, P.G. Jones, and A. Jarvis, Very high resolution interpolated climate surfaces for global land areas, *Int. J. Climat., 25(*15), 1965–1978, 2005.

IEEE, IEEE standard for binary floating-point arithmetic. *Std. 754-1985*, 1985.

IEEE, IEEE standard for floating-point arithmetic. *Std. 754-2008*, 2008.

Iverson, K. E., Notation as a tool of thought, *Commun. ACM, 23*, 444-465, 1980.

Jasanoff, S., Testing time for climate science. Science 328 (5979), 695-696, 2010.

Knuth, D. E., Computer programming as an art, Commun. ACM 17 (12), 667-673, 1974.

Liskov, B.H. and J.M. Wing, Behavioral Subtyping Using Invariants and Constraints, *Formal methods for distributed processing*, 254–280, 2001.

Loarie, S.R., P.B. Duffy, H. Hamilton, G.P. Asner, C.B. Field, and D.D. Ackerly, The velocity of climate change, *Nature, 462*(7276), 1052–1055, 2009.

Lovett, A., G. Sünnenberg, G. Richter, A. Dailey, A. Riche, and A. Karp, Land Use Implications of Increased Biomass Production Identified by GIS-Based Suitability and Yield Mapping for Miscanthus in England, *BioEnergy Research, 2*(1), 17–28, 2009.

Maggiolini, P., Information Technology Benefits: A Framework, *Emerging Themes in Information Systems and Organization Studies*, 281-292, 2011.

McGregor, J. D. 2006. Complexity, its in the mind of the beholder. J of Obj. Tech. 5(1): 31-37.

Merritt, W.S., R.A. Letcher, and A.J. Jakeman, A review of erosion and sediment transport models, *Environmental Modelling & Software, 18*(8-9) 761–799, 2003.

Morin, A., J. Urban, P. D. Adams, I. Foster, A. Sali, D. Baker and P. Sliz , Shining Light into Black Boxes, *Science, 336*(6078), 159-160, 2012.

Nature, Devil in the details, *Nature 470* (7334), 305–306, 2011.

Oliphant , T., E., Guide to NumPy, *Trelgol Publishing,* 2006.

Patz, J.A., D. Campbell-Lendrum, T. Holloway, and J.A. Foley, Impact of regional climate change on human health, *Nature, 438*(7066), 310-317, 2005.

Peng, R. D., Reproducible Research in Computational Science, *Science, 334*(6060), 1226-1227, 2011.

Prudhomme C., N. Reynard, and S. Crooks, Downscaling of global climate models for flood frequency analysis: Where are we now?, *Hydrological Processes, 16*, 1137–1150, 2002.

Ramey, C., and B. Fox, The GNU Bash Reference Manual, *Network Theory Ltd*, 2006.

Ramirez, J., and A. Jarvis, Downscaling Global Circulation Model Outputs: The Delta Method, *Decision and Policy Analysis Working Paper No. 1*, CIAT, 2010.

Scholes, R.J., G.M. Mace, W. Turner, G.N. Geller, N. Jürgens, A. Larigauderie, *et al.*, Toward a Global Biodiversity Observing System, *Science, 321*(5892), 1044-1045, 2008.

Smaalders, B., Performance Anti-Patterns. *Queue 4* (1), 44-50, 2006.

Stallman, R., Free Community Science and the Free Development of Science, *PLoS Med 2*(2), 2005.

Stallman, R., Viewpoint: Why "open source" misses the point of free software, *Commun. ACM, 52*(6), 31–33, 2009.

Stodden, V., Reproducible Research: Tools and Strategies for Scientific Computing, *Computing in Science & Engineering , 14*(4), 11-12, 2012.

Stodden, V., Trust Your Science? Open Your Data and Code , *Amstat News* , 2011.

Stroustrup, B. A rationale for semantically enhanced library languages, *ACM LCSD05*, 2005

Tabor, K. and Williams, J.W., Globally downscaled climate projections for assessing the conservation impacts of climate change, *Ecological Applications, 20*(2), 554–565, 2010.

Tol, R. S. J., The economic effects of climate change. *J. Econ. Perspect. 23* (2), 29-51, 2009.

Trivedi, M.R., P.M. Berry, M.D. Morecroft, and T.P. Dawson, Spatial scale affects bioclimate model projections of climate change impacts on mountain plants, *Global Change Biology, 14*(5), 1089–1103, 2008.

van den Besselaar, E.J.M., M.R. Haylock, G. van der Schrier, and A.M.G. Klein Tank, A European Daily High-resolution Observational Gridded Data set of Sea Level Pressure, *J. Geophys. Res., 116*, D11110, 2011.

van den Besselaar, E.J.M., M.R. Haylock, G. van der Schrier, and A.M.G. Klein Tank, A European Daily High-resolution Observational Gridded Data set of Sea Level Pressure, *J. Geophys. Res., 116*, D11110, 2011.

Van Rossum, G. and F.J. Drake, Python Language Ref. Manual, *Network Theory Ltd.*, 2011.

Venables, W.N. and D.M. Smith, An Introduction to R, 2nd ed., *Network Theory Ltd.*, 2009.

Wilson, G., Where's the real bottleneck in scientific computing? *American Scientist 94*(1), 2006.

WorldClim, Data Format, http://www.worldclim.org/formats, (accessed 2012-02-26).

Yale Law School Roundtable on Data and Code Sharing (YLSRDCS), Reproducible Research, *Computing in Science & Engineering , 12*(5), 8-13, 2010.

Yesson, C., P.W. Brewer, T. Sutton, N. Caithness, J.S. Pahwa, M. Burgess, W.A. Gray, *et al*., How Global Is the Global Biodiversity Information Facility?, *PLoS ONE 2*(11), 2007.