Scholarly Research Exchange

SRX Computer Science • Volume 2010 • Article ID 812789 • doi:10.3814/2010/812789

Research Article

AirTemperature: Extensible Software Library to Generate Air Temperature Data

Marcello Donatelli,[1,2] Gianni Bellocchi,[1] Ephrem Habyarimana,[1] Simone Bregaglio,[1,3] and Bettina Baruth[2]

[1] *Agriculture Research Council, CRA-CIN, 40128 Bologna, Italy*
[2] *European Commission Joint Research Centre, Institute for the Protection and Security of the Citizen, MARS Unit, AGRI4CAST Action, 21027 Ispra (VA), Italy*
[3] *Department of Crop Science, University of Milan, 20133 Milan, Italy*

Correspondence should be addressed to Marcello Donatelli, marcello.donatelli@jrc.ec.europe.eu

Received 10 December 2009; Revised 2 February 2010; Accepted 21 February 2010

The development of a set of reusable libraries to support custom applications has become a goal in biophysical modeling projects. This is true for weather modeling as well. *AirTemperature* is a software component providing a collection of deterministic and stochastic approaches to generate atmospheric temperature data on daily and hourly time steps. Data generated on a daily time step consist of maximum and minimum air temperature and dew point temperature. Hourly estimations include air and dew point temperatures. The software design allows for extension of the models implemented without recompiling the component. The component, inclusive of hypertext help documentation files, is released as compiled .NET2 version, allowing application development in either programming environment. A sample client and a sample extension project using *AirTemperature* are provided as source code. A sample Web service and a Web application are also developed as examples of possible use of the component.

1. Introduction

A large number of existing agricultural and ecological models have been implemented as software that cannot be well maintained or reused, except by their authors, and therefore cannot be easily transported to other platforms (e.g., [1, 2]). In order to possibly include legacy data sources into newly developed systems, object-oriented development has emerged steadily as a paradigm that focuses on granularity, productivity, and low maintenance [3]. Several papers have been recently published in agroecological journals [4–10] targeting at reusable dynamic link libraries either within the Microsoft.NET framework (http://www.microsoft.com/NET/) or using the SUN Java platform (http://java.sun.com/). Such solutions reflect a style of programming referred to as component-oriented programming that has become the leading methodology in developing systems in a variety of domains, including agroecological modeling [2]. Although different definitions of component do exist in the literature [11–13], a component is basically a discrete software unit which makes available specific functionalities, and it can be presented as a black box that provides access to its services through a defined interface. The component development paradigm is to make the construction of a software as plugging together independent components.

In the context of the agricultural and environmental modeling community, alternative frameworks are available to support modular model development through provision of libraries of biophysical modeling modules, as well as reusable tools for data manipulation, analysis, and visualization [14]. Various object- and component-oriented solutions have approached the issue of agricultural and environmental modeling, such as maize irrigation scheduling [15], multiple spatial scales ecosystems [16], greenhouse control systems [17], and households, landscape, and livestock integrated systems [18]. In the same perspective, we have approached the weather generation issue. Long records of weather data are in fact needed for evaluating agricultural management scenarios in natural resource models (e.g., [19, 20]). Weather inputs required by natural resource models include air temperature, precipitation, solar radiation, wind speed, and dew point temperature. Synthetic weather sequences are needed if long-term measured data are not available, measured data contain missing records, or collection of actual data is cost or time prohibitive or when necessary to simulate impacts of future climate scenarios. Weather simulation models (or weather generators) are commonly used to generate synthetic weather records for use in the study of crop growth and development, water availability, soil erosion, climate change, and other domains (e.g. WST, http://www.wcc.nrcs.usda.gov/climate/wst_fact.html). Several weather generators are available in the form of ready-to-use, user-oriented tools, implementing specific solutions to the basic problem of generating one or more weather element. Such an approach is, however, ineffective for developers of custom applications, who have to reimplement the set of equations within modeling applications of various complexity. Moreover, because of either the empiricism or the alternative inputs required by different generation approaches, it may be desirable to compare different methods in case-specific applications in order to provide reliable weather data for case-specific applications.

Reusability in weather generation can be efficiently achieved by capturing the domain knowledge currently available (i.e., weather models already developed and tested) and making it available in software components. This is the reason why component-oriented tools have been recently developed to fit this need, that is, *ET* for calculating evapotranspiration and related variables [5], *GSRad* for estimating synthetic values of solar radiation [6], *Rain* [4] for generating precipitation data, and *Wind* [9] for generating wind speed data. The components mentioned provide a set of alternate models to estimate variables specific for the domain targeted and are implemented using a software architecture which promotes reusability [21]. The present study focuses on the modeling of air temperature that, to the best of the authors' knowledge, has not yet encapsulated into component-based solutions. Air temperature values are essential to plant growth and the development of organisms. One problem in simulating air temperature is that measured daily maximum and minimum air temperatures are often slightly skewed and not normally distributed in each month. So, generating air temperature from the normal distribution may result in physically improbable values (especially extreme hot temperatures). Although the assumption of normality is often contradicted [22], the normal distribution (variously interpreted and corrected) is the reference distribution of all the approaches currently used to generate air temperature data. Weather generators (including Cligen [23], WGEN [24], USCLIMATE [25], LARS-WG [26], ClimGen [27], and CLIMAK [28]) are commonly used to generate daily maximum and minimum air temperatures in agroecological projects. Generation of maximum and minimum air temperatures is also useful for modeling applications that require estimates of hourly temperature throughout a day. A best guess is made by assuming that minimum air temperatures normally occur close to sunup and maximum air temperatures a few hours after solar noon (e.g., [29]). Moreover, relationships between air relative humidity and air temperature can be rearranged as an association of the dew point temperature with the two daily extremes (e.g., [30]). Disaggregation from daily to hourly records and estimation of dew point air temperatures are both largely based on empirical relationships.

This paper illustrates how well-known air temperature generation approaches have been implemented into a software component (namely, *AirTemperature*). The procedures implemented in the component, the scientific background, some principles of usage, and source code are extensively documented in hypertext help files. The paper describes the implementation features that guided the development of *AirTemperature*, followed by a discussion on the main component features.

2. Background

The modeling background implemented in *AirTemperature*, fully documented in the online help file, is not reproduced hereafter. The main features are only briefly summarized. All these models are published in peer-reviewed journals; details about their development and the applications in case studies are reported in the referenced papers.

2.1. Daily Generation of Air Temperature. The generation of daily maximum ($T_{\max}$, °C) and minimum ($T_{\min}$, °C) air temperatures is considered to be a continuous stochastic process, possibly conditioned by the precipitation status of the day. Three methods are implemented for generating daily values of $T_{\max}$ and $T_{\min}$. The multistage generation system is conditioned on the precipitation status with both approaches from Richardson [31] Danuso and [28]. Residuals for $T_{\max}$ and $T_{\min}$ are computed first, then daily values are generated independently (Richardson-type) or with dependence of $T_{\max}$ on $T_{\min}$ (Danuso-type). A third stage, that adds an annual trend calculated from the Fourier series, is included in Danuso-type generation. The Richardson-type approach accounts for air temperature—solar radiation correlation. A third approach [32] generates $T_{\max}$ and $T_{\min}$ independently in two stages (daily mean air temperature generation first, $T_{\max}$ and $T_{\min}$ next), making use of an autoregressive process from mean air temperatures and solar radiation parameters.

2.2. Hourly Generation of Air Temperature. Daily values of $T_{\max}$ and $T_{\min}$ are used to generate hourly air temperature values, according to alternative methods. Sinusoidal functions are largely used to represent the daily pattern of air temperature. Six approaches, by Campbell [33], Goudriaan and van Laar [34], Ephrath et al. [35], Porter et al. [36], Stöckle [29], and Gracia et al. [37] are used to generate hourly values from daily maximum and minimum temperatures. A further approach, proposed by Dumortier [38], derives hourly air temperatures from the daily solar radiation profile. Mean daily values of dew point air temperature are estimated via empirical relationships between $T_{\max}$ and $T_{\min}$ and other

Table 1: List of all the inputs and outputs of the models implemented into *AirTemperature* component. Outputs are arranged by an identification number (ID) assigned to input variables and parameters used to calculate each output.

	Variables/parameters	Unit	Output ID
Output variables	$T_{d(hr)}$: hourly dew point air temperature	°C	1
	T_d: daily dew point air temperature	°C	2
	$T_{\max}$: daily maximum air temperature	°C	3
	$T_{\min}$: daily minimum air temperature	°C	4
	T_T^k: yearly trend of daily maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperatures on dry ($k = 0$) or wet ($k = 1$) days	°C	5
	$\chi_{(d)}^{k_0,k}(j)$: daily standardized residual of maximum ($j = 1$) and minimum ($j = 2$) air temperatures on dry ($k = 0$) or wet ($k = 1$) days	—	6
	T_{hr}: hourly air temperature	°C	7
Input variables	$RH_{\max}$: daily maximum relative humidity	%	2
	$RH_{\min}$: daily minimum relative humidity	%	2
	dT: day-to-day difference of mean air temperatures	°C	3, 4
	dT_{sd}: standard deviation of day-to-day difference of mean air temperatures	°C	3, 4
	$\overline{G}$: monthly average of daily global solar radiation on a given surface	MJ $\mathrm{m^{-2}d^{-1}}$	3, 4, 6
	$G_{x(d)}$: daily global solar radiation at ground level	MJ $\mathrm{m^{-2}d^{-1}}$	3, 4, 6
	Z: site elevation above sea level	m	2
	$T_m(c)$: mean air temperature of the coolest month	°C	2
	$T_m(d)$: mean air temperature of the warmest month	°C	2
	$G_{h(hr)}$: hourly global solar radiation on a horizontal surface	MJ $\mathrm{m^{-2}}$ $\mathrm{d^{-1}}$	7
	d: day number into year	—	3, 4, 5
	S_i: precipitation occurrence of current day	—	3, 4, 5, 6
	$DL_{(d)}$: day length	h	7
	$sr_{(d)}$: time of sunrise	h	1, 7
	$T_{\max}$: daily maximum air temperature	°C	2, 7
	$T_{\min}$: daily minimum air temperature	°C	2, 7
	$\overline{T}_T^{k_0,k}$: monthly average of daily maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	°C	3, 4, 6
	$\sigma_T^{k_0,k}$: monthly standard deviation of daily maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	°C	3, 4, 6
Input parameters	A_T^k: annual mean maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	°C	3, 4, 5
	B_T^k: semiamplitude of the first harmonic for yearly trends of maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	—	3, 4, 5
	C_T^k: phase shift of the first harmonic for yearly trends of maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	—	3, 4, 5
	D_T^k: semiamplitude of the second harmonic for yearly trends of maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	—	3, 4, 5
	E_T^k: phase shift of the second harmonic for yearly trends of maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) air temperature on dry ($k = 0$) or wet ($k = 1$) days	—	3, 4, 5
	RR_{nn}: monthly autocorrelation coefficient for minimum air temperature residuals, with time lag of 1 day	—	3, 4
	RR_{nx}: monthly correlation coefficient between minimum and maximum air temperature residuals	—	3, 4

Table 1: Continued.

Variables/parameters	Unit	Output ID
SR_T: monthly standard deviation of the residuals from the trends of maximum ($T = T_{\max}$) and minimum ($T = T_{\min}$) temperature both on dry and wet days	°C	3, 4
R_{1n}: standardized residual of minimum air temperature of the previous day	—	3, 4
$T_{\max(d-1)}$: maximum air temperature of the previous day	°C	3, 4, 7
$T_{\min(d-1)}$: minimum air temperature of the previous day	°C	3, 4, 7
N: number of days in a month	—	3, 4
$N(d)$: number of dry days in a month	—	3, 4
$N(w)$: number of wet days in a month	—	3, 4
b_T: scaling factor	—	3, 4
k_0: dry/wet days separation option	—	3, 4, 6
$\mathbf{A}^{\mathbf{k_0,k}}$: 3×3 matrix function of the lag-0 serial- and cross-correlation coefficients of the residuals on dry ($k = 0$) or wet ($k = 1$) days	—	3, 4, 6
$\mathbf{B}^{\mathbf{k_0,k}}$: 3×3 matrix function of the lag-1 serial- and cross-correlation coefficients of the residuals on dry ($k = 0$) or wet ($k = 1$) days	—	3, 4, 6
$\chi_{(d-1)}^{k_0,k}(j)$: standardized residual of maximum ($j = 1$) and minimum ($j = 2$) air temperatures and global solar radiation ($j = 3$) on dry ($k = 0$) or wet ($k = 1$) days for the previous day	—	3, 4
K_0: daily dew point air temperature correction factor	°C	2
a, b, c, d: empirical parameters of model Hubbard for daily dew point air temperature	–, –, –, °C	2
A, B, C, D, E, F, G: empirical parameters of model of Kimball et al. for daily dew point air temperature	–, –, –, –, –, –, °C	2
EF: evaporative demand	—	2
$A1$, $B1$, $C1$, $D1$, $E1$: empirical parameters of model Linacre 1 for daily dew point air temperature	m^{-1}, –, –, –, °C	2
$A2$, $B2$, $C2$, $D2$: empirical parameters of model Linacre 2 for daily dew point air temperature	–, $°\mathrm{C}^{-1}$, –, °C	2
hr_{dv}: hour of the day for maximum air temperature to occur	h	7
$T_{\min(d+1)}$: minimum air temperature of the next day	°C	7
$sr_{(d+1)}$: time of sunrise of the next day	h	7
$ss_{(d+1)}$: time of sunset of the next day	h	7
$T(ss_{(d-1)})$: air temperature at sunset of the previous day	°C	7
a_{slp}: slope coefficient	—	7
LSH: hour of maximum solar height	h	7
p: delay of the maximum air temperature	h	7
T_k: air temperature increment	°C	7
TC: nocturnal time coefficient	h	7
k: shift factor	—	7
A_P, B_P, C_P: empirical parameters of model of Porter et al. for hourly air temperature	h, h, –	7
A_S, B_S, C_S, D_S, E_S: empirical parameters of model Stöckle for hourly air temperature	–, –, –, –, –	7
$T_{d(d+1)}$: dew point air temperature of the next day	°C	1
$T_d(\max)$: maximum dew point air temperature	°C	1

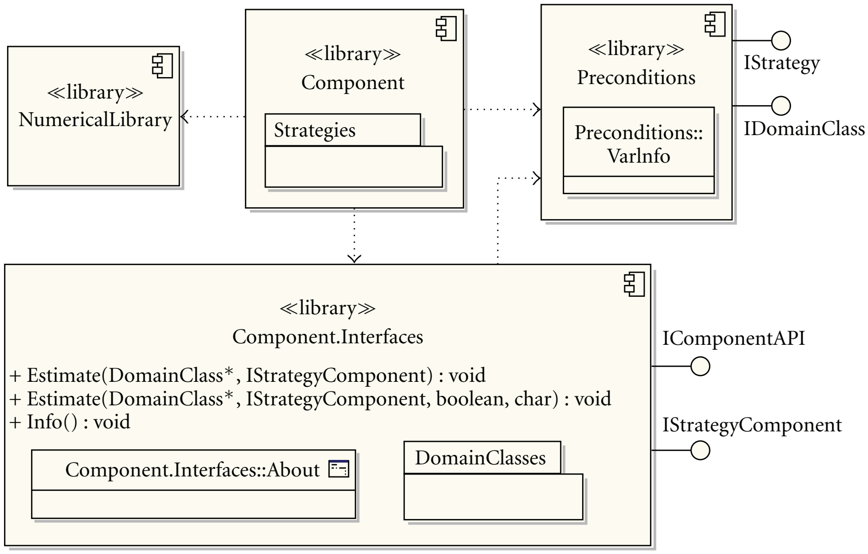

Figure 1: Generic component model used for *AirTemperature*. The Preconditions component allows the implementation of the design-by-contract approach and provides the base classes to build and make accessible the component ontology. The separation of data types (domain classes) and interfaces from models (strategies) in two discrete units allows the implementation in clients of the design pattern *Bridge*, which facilitates replacement of model components.

variables [30, 39–43]. A diurnal pattern (hourly time step) of dew point air temperature is also modeled via two alternative methods [35, 44].

3. Software Features

3.1. Input and Outputs. The outputs produced by *AirTemperature* and the inputs (variables and parameters) required by the models implemented are listed in Table 1.

3.2. Design. The software design promotes reusability by limiting dependencies and providing a semantically rich, public interface. By allowing extensibility of approaches in a straightforward way, it also allows third parties to add new equations and the comparison of alternate air temperature models. This design [21] combines architectural traits that maximize transparency, extensibility, scalability, traceability, and data quality control. The same design has been already used in the development of several components for agrometeorology, agromanagement, crop and soil water/nitrogen/chemicals simulation, model evaluation, and soil pedotransfer functions (http://www.apesimulator.org/help.aspx). The Unified Modeling Language (UML) component model of *AirTemperature* (Figure 1) shows the discrete units and their dependencies.

3.3. Architecture. *AirTemperature* architecture allows extending data-types and adding new modeling solutions without the need of recompilation of the core component.

The component implements *Strategies*, which are alternative implementation of air temperature models. Each model is computed via one of such discrete units, which encapsulates the algorithm, the test of pre- and postconditions, and parameters declaration (if any). The component implements *Composite Strategies* (built using two or more strategies) and *Context Strategies*, that is, model units which implement logic to select among the strategies associated, for example, based on the inputs available. Extension is made possible by the definition of the common interface `IAirTDataStrategy`, which must be implemented by all strategies. The UML class diagram (Figure 2) shows the classes and interfaces which allow extending the component via the Composite and Strategy design patterns.

The design-by-contract approach [45] is used, requiring pre- and postconditions (e.g., maximum daily air temperature > minimum daily air temperature) to be respected. Any application using *AirTemperature* can hence test inputs for a possible violation of preconditions, and it can check postconditions (http://agsys.cra-cin.it/tools/preconditions/help/). The Model Component Explorer (MCE, http://agsys.cra-cin.it/tools/, page "Applications", then "MCE") is an application to discover parameters, inputs, and outputs of each model and to browse the component ontology by inspecting data-types (called *Domain Classes*) and the component interfaces. *AirTemperature* is one of the core components of the weather generator CLIMA [8].

3.4. Distribution. *AirTemperature* is distributed via a Software Development Kit that includes the source code of Visual Studio.NET projects demonstrating how to extend and reuse the component. Also, hypertext files are made available, documenting the models implemented (http://agsys.cra-cin.it/tools/airtemperature/help/) and the code of the software component (http://agsys.cra-cin.it/tools/airtemperature/codedoc/). The project source code of sample web services (http://agsys.cra-cin.it/webservices/airtemperature/) and web

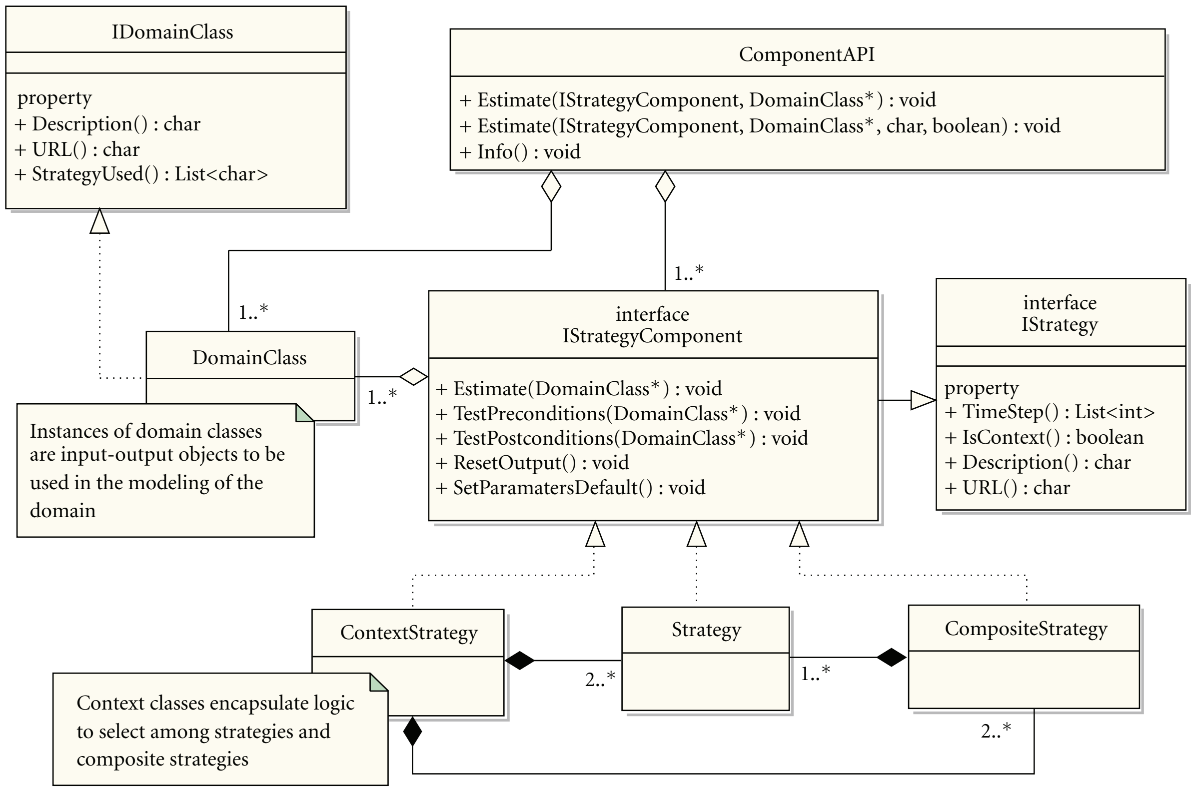

Figure 2: Class diagram illustrating the specific implementation of the design patterns *Composite* and *Strategy*.

application (http://agsys.cra-cin.it/webapplications/airtemperature/) is also provided. The component requires the framework Microsoft.NET 2.0 (or newer) installed.

4. Remarks

It is widely accepted that research in agroecology must be supported by the state-of-the-art modeling. Model development and operational use require, however, the capability of quickly accessing knowledge in different domains, selecting and comparing alternate modeling options, and making use of such knowledge via computer-based tools. The modeling system of *AirTemperature* can be considered a way to share knowledge, making it available in an operational tool. To date, although the use of software model frameworks has improved the maintainability of complex simulation systems, effective reuse of discrete units in the domain of biophysical models is still mostly a goal rather than an achievement. The architecture of *AirTemperature* decouples data from weather models, providing a semantically rich interface in framework-independent implementation, thus facilitating reuse and independent extensibility by third parties.

Acknowledgments

Laura Carlini is acknowledged for her contribution to the initial implementation. This publication has been funded under the SEAMLESS integrated project, EU 6th Framework Programme for Research, Technological Development and Demonstration, and Priority 1.1.6.3. Global Change and Ecosystems (European Commission, DG Research, contract no. 010036-2).

References

[1] J. F. Reynolds and B. Acock, "Modularity and genericness in plant and ecosystem models," *Ecological Modelling*, vol. 94, no. 1, pp. 7–16, 1997.

[2] P. Papajorgji, H. W. Beck, and J. L. Braga, "An architecture for developing service-oriented and component-based environmental models," *Ecological Modelling*, vol. 179, no. 1-2, pp. 61–76, 2004.

[3] B. Timothy, *An Introduction to Object-Oriented Programming*, Addison-Wesley, Reading, Mass, USA, 2nd edition, 1997.

[4] L. Carlini, G. Bellocchi, and M. Donatelli, "A library to generate synthetic precipitation data," *Agronomy Journal*, vol. 98, no. 5, pp. 1312–1317, 2006.

[5] M. Donatelli, G. Bellocchi, and L. Carlini, "Sharing knowledge via software components: models on reference evapotranspiration," *European Journal of Agronomy*, vol. 24, no. 2, pp. 186–192, 2006.

[6] M. Donatelli, L. Carlini, and G. Bellocchi, "A software component for estimating solar radiation," *Environmental Modelling and Software*, vol. 21, no. 3, pp. 411–416, 2006.

[7] R. Confalonieri, G. Bellocchi, and M. Donatelli, "A software component to compute agro-meteorological indicators," *Environmental Modelling and Software*. In press.

[8] M. Donatelli, G. Bellocchi, E. Habyarimana, S. Bregaglio, R. Confalonieri, and B. Baruth, "CLIMA, a modular weather generator," in *Proceedings of the 18th World IMACS Congress and MODSIM International Congress on Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation*, R. S. Anderssen, R. D. Braddock, and L. T. H. Newham, Eds., pp. 852–858, Cairns, Australia, July 2009, http://www.mssanz.org.au/modsim09/C3/donatelli_C3a.pdf.

[9] M. Donatelli, G. Bellocchi, E. Habyarimana, R. Confalonieri, and F. Micale, "An extensible model library for generating wind speed data," *Computers and Electronics in Agriculture*, vol. 69, no. 2, pp. 165–170, 2009.

[10] D. P. Holzworth, N. I. Huth, and P. G. de Voil, "Simplifying environmental model reuse," *Environmental Modelling and Software*, vol. 25, no. 2, pp. 269–275, 2010.

[11] P. A. Bernstein, T. Bergstraesser, J. Carlson, S. Pal, P. Sanders, and D. Shutt, "Microsoft repository version 2 and the open information model," *Information Systems*, vol. 24, no. 2, pp. 71–98, 1999.

[12] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley, Reading, Mass, USA, 1999.

[13] C. Szypersky, D. Gruntz, and S. Murer, *Component Software—Beyond Object-Oriented Programming*, Addison-Wesley, London, UK, 2nd edition, 2002.

[14] R. M. Argent, A. Voinov, T. Maxwell, et al., "Comparing modelling frameworks—a workshop approach," *Environmental Modelling and Software*, vol. 21, no. 7, pp. 895–910, 2006.

[15] J.-E. Bergez, P. Debaeke, J.-M. Deumier, et al., "MODERATO: an object-oriented decision tool for designing maize irrigation schedules," *Ecological Modelling*, vol. 137, no. 1, pp. 43–60, 2001.

[16] P. B. Woodbury, R. M. Beloin, D. P. Swaney, B. E. Gollands, and D. A. Weinstein, "Using the ECLPSS software environment to build a spatially explicit component-based model of ozone effects on forest ecosystems," *Ecological Modelling*, vol. 150, no. 3, pp. 211–238, 2002.

[17] J. M. Aaslyng, J. B. Lund, N. Ehler, and E. Rosenqvist, "IntelliGrow: a greenhouse component-based climate control system," *Environmental Modelling and Software*, vol. 18, no. 7, pp. 657–666, 2003.

[18] R. Matthews, "The People and Landscape Model (PALM): towards full integration of human decision-making and biophysical simulation models," *Ecological Modelling*, vol. 194, no. 4, pp. 329–343, 2006.

[19] T. M. Shenk and A. B. Franklin, *Modeling in Natural Resource Management: Development, Interpretation, and Application*, Edited Island Press, Washington, DC, USA, 2001.

[20] H. S. Mavi, *Agrometeorology: Principles and Applications of Climate Studies in Agriculture*, Haworth Press, Binghamton, NY, USA, 2004.

[21] M. Donatelli and A. E. Rizzoli, "A design for framework-independent model components of biophysical systems," in *Proceedings of the International Congress on Environmental Modelling and Software (IEMS '08)*, M. Sànchez-Marrè, J. Béjar, J. Comas, A. E. Rizzoli, and G. Guariso, Eds., vol. 2, pp. 727–734, Barcelona, Spain, July 2008.

[22] R. D. Harmel, G. Johnson, and C. W. Richardson, "The GEM experience: weather generator technology development in the USDA," *Bulletin of the American Meteorological Society*, vol. 83, no. 7, pp. 954–957, 2002.

[23] A. D. Nicks and G. A. Gander, "Cligen: a weather generator for climate inputs to water resources and other models," in *Proceedings of the 5th International Conference on Computer in Agriculture*, D. G. Watson, F. S. Zanueta, and T. V. Harrison, Eds., pp. 3–94, American Society of Agricultural Engineers, Orlando, Fla, USA, February 1994.

[24] C. W. Richardson and D. A. Wright, "WGEN: a model for generating daily weather variables," Tech. Rep. ARS-8, U.S. Department of Agriculture, Agricultural Research Service, Washington, DC, USA, 1984.

[25] G. L. Johnson, C. L. Hanson, S. P. Hardegree, and E. B. Ballard, "Stochastic weather simulation: overview and analysis of two commonly used models," *Journal of Applied Meteorology*, vol. 35, no. 10, pp. 1878–1896, 1996.

[26] M. A. Semenov, R. J. Brooks, E. M. Barrow, and C. W. Richardson, "Comparison of the WGEN and LARS-WG stochastic weather generators for diverse climates," *Climate Research*, vol. 10, no. 2, pp. 95–107, 1998.

[27] C. O. Stöckle, R. L. Nelson, M. Donatelli, and F. Castellvì, "ClimGen: a flexible weather generation program," in *Proceedings of the 2nd International Symposium Modelling Cropping Systems*, M. Bindi, M. Donatelli, J. Porter, and M. K. van Ittersum, Eds., pp. 229–230, Florence, Italy, July 2001.

[28] F. Danuso, "Climak: a stochastic model for weather data generation," *Italian Journal of Agronomy*, vol. 6, pp. 57–71, 2002.

[29] C. O. Stöckle, *Atmosphere: Derived Climate Variables and Time Interpolation*, BSysE 562 lecture, Pullman, Wash, USA, 2002.

[30] E. Linacre, *Climate Data and Resources: A Reference and Guide*, Routledge, London, UK, 1992.

[31] C. W. Richardson, "Stochastic simulation of daily precipitation, temperature, and solar radiation," *Water Resources Research*, vol. 17, no. 1, pp. 182–190, 1981.

[32] J. Remund and J. Page, "Chain of algorithms: short- and longwave radiation with associated temperature prediction resources," SoDa Deliverable D5-2-2/3. Internal document, 2002.

[33] G. S. Campbell, *Soil Physics with BASIC: Transport Models for Soil-Plant Systems*, Elsevier, Amsterdam, The Netherlands, 1985.

[34] J. Goudriaan and H. H. van Laar, *Modelling Potential Crop Growth Processes*, Kluwer Academic Publishers, London, UK, 1994.

[35] J. E. Ephrath, J. Goudriaan, and A. Marani, "Modelling diurnal patterns of air temperature, radiation wind speed and relative humidity by equations from daily characteristics," *Agricultural Systems*, vol. 51, no. 4, pp. 377–393, 1996.

[36] C. H. Porter, N. B. Pickering, J. W. Jones, and G. Hoogenboom, "Weather module in DSSAT v. 4.0 documentation and source code listing," Agricultural and Biological Engineering Department Research Report 2000-1203, University of Florida, Gainesville, Fla, USA, 2000.

[37] C. Gracia, S. Sabaté, and A. Sánchez, *GOTILWA+: A Forest Growth Simulation Model. Model Documentation and User's Guide*, Center for Ecological Research and Forestry Applications, Barcelona, Spain, 2003.

[38] D. Dumortier, "Prediction of air temperatures from solar radiation," Tech. Rep. SoDa-5-2-4, CNRS-ENTPE, 2002.

[39] E. T. Linacre, "A simple formula for estimating evaporation rates in various climates, using temperature data alone," *Agricultural Meteorology*, vol. 18, no. 6, pp. 409–424, 1977.

[40] J. V. Iribarne and W. L. Godson, *Atmospheric Thermodynamics*, D. Reidel, 1981.

[41] J. S. Kimball, S. W. Running, and R. Nemani, "An improved method for estimating surface humidity from daily minimum temperature," *Agricultural and Forest Meteorology*, vol. 85, no. 1-2, pp. 87–98, 1997.

[42] R. G. Allen, L. S. Pereira, D. Raes, and M. Smith, "Crop evapotranspiration: guidelines for computing crop water requirements," Irrigation & Drainage Paper 56. UN-FAO, Rome, Italy, 1998.

[43] K. G. Hubbard, R. Mahmood, and C. Carlson, "Estimating daily dew point temperature for the northern Great Plains using maximum and minimum temperature," *Agronomy Journal*, vol. 95, no. 2, pp. 323–328, 2003.

[44] Meteotest, *Meteonorm version 5.0. The global meteorological database for engineers, planners and education. Software and data on CD-ROM*, James and James, London, UK, 2003.

[45] B. Meyer, *Object-Oriented Software Construction*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 1997.