CLIMA: a component-based weather generator

Donatelli, M., G. Bellocchi, L. Carlini, and M. Colauzzi

CRA-Research Institute for Industrial Crops, E-Mail: **m..donatelli@isci.it**

Keywords: *weather generation; component-oriented programming; re-usability; extensibility.*

EXTENDED ABSTRACT

Investigating the potential impact of climate on agro-ecosystems using simulation models is underpinned by the availability of climate data at the appropriate temporal scale (daily or higher resolution). The production of artificial series of weather data has traditionally adopted a variety of alternative methods. These range from empirical functions where simple relationships between weather variables are used to estimate missing data from available data, to sophisticated approaches where physically-based models are used. All such approaches illustrate from different perspectives that there is actually a wealth of well developed solutions to the basic problem of estimating or generating weather data, coded in a variety of ways. The weather generation and estimation problem and the frequent need to evaluate alternative approaches in a comparative fashion has suggested that these methods be made available as one, comprehensive set of linkable components which also allow for extension by third parties of models that have already been implemented.

This research describes the development of a software component called *CLIMA*, which provides a structured repository of methods for estimation of weather variables (synthetic production of weather variables from other weather variables) and generation (synthetic production of weather data from site-specific statistical properties). *CLIMA* is a synthetic weather estimator/generator used to characterize study areas and provide daily or sub-daily weather inputs (air temperature, evapotranspiration, precipitation, solar radiation, wind speed) to agro-ecological models. The model component enables users to interrogate the weather database of study areas, to estimate parameters, to produce synthetic weather data and to distribute them for more advanced analysis. There are five main subcomponents in *CLIMA*, including estimation/generation functions for each relevant weather variable. These basic model components are: *AirT*: air temperature generation; *ET*: reference evapotranspiration estimation; *GSRad*: solar radiation estimation/generation; *Rain*: precipitation generation; *Wind*: wind speed generation. Each of the components offers a wide range of alternative methods to perform estimates and data generation. *CLIMA* also includes other components to evaluate pre-conditions and post-conditions, estimate model parameters, check data quality and provide data at run-time for applications in simulation models. All these components, which can also be deployed independently for other uses than in the *CLIMA* framework share a common set of features: 1) full documentation as hypertext help files (one regarding the modelling solutions, and one regarding code, NDoc style, http://ndoc.sourceforge.net), 2) sample clients inclusive of source code (WinForms applications in C#), 3) possibility of extending models by adding new ones without recompiling the component and still using the same call, 4) test of pre- and post-conditions, 5) unit tests for all public methods. The full installation package also includes a client with a graphical user interface, and an application (*MCE*: Model Component Explorer) to discover models and model requirements.

Component-oriented programming, that is encapsulating the solution of modelling problems into discrete, replaceable and interchangeable software unit (components), was used to define the *CLIMA* design. The *CLIMA* software architecture was based on ease of maintenance, reuse, interchangeability and extensibility. *CLIMA* was developed using C# in the .NET platform of Windows, and it adopts a hybrid solution based on a relational database and the file system for data storage and retrieval. Drivers for data export in specific formats can be added without recompiling the CLIMA core component. Some components are also released as Java libraries for use on other operating systems. Sample Windows applications, web services and web applications are also made available as source code to show how to use the subcomponents and extend their capabilities. The goal of *CLIMA* is to extend access to weather modelling to a number of users, and to provide an architecture that ensures the re-use and extension of models. *CLIMA* attempts to overcome some of the technical challenges that have so far limited the development of reusable models for weather generation and estimation. Via their model documentation and the metadata associated with each variable, *CLIMA* model components also make the sharing of modelling knowledge possible in an easy and extensible way.

1. INTRODUCTION

The advent of component-based programming has enabled the development of scalable, robust, large-scale applications in a variety of domains, including agro-ecological modelling (Argent, 2004). The concept of developing modular systems for biophysical simulation has led to the development of several modelling frameworks (e.g. Simile, ModCom, IMA, TIME, OpenMI, SME, OMS, as listed in Argent and Rizzoli, 2004), which make use of models as software components which can be linked together and composed according to the modelling purpose. However, little attention has been paid to the development of components which intrinsically promote re-usability, interchangeability, and extensibility of models (Donatelli et al., 2005). Components with such characteristics could have a different use in different clients and contexts, from end-user applications to web services through modelling frameworks; this would be interesting for cross-cutting communities of model developers.

Investigating the potential impact of climate on agro-ecosystems using simulation models is underpinned by the availability of climate data at the appropriate temporal scale (daily or higher resolution). The production of artificial series of weather data can be achieved by a variety of alternative methods. These range from empirical functions where simple relationships between weather variables are used to estimate missing data from available data (e.g. global solar radiation from air temperature), to sophisticated approaches where physically-based models are used (e.g. combined equations with solar radiation and aerodynamic terms to estimate crop evapotranspiration). Artificial weather series can be generated using the approaches mentioned above, provided that sufficient measured data are available to extract statistical parameters representative of actual weather, and a generation method, preserving such statistical properties in the generated series, is available. Methods of various complexity have been developed, that have been proven able to provide good approximations of the large variability of daily weather patterns. Such models are well suited to ecological and environmental applications, striking a balance between the complexity of the climate system and the goodness-of-fit between observed and simulated data (Hutchinson, 1987). Both estimation and generation methods have been implemented in weather generators and other software tools (e.g., WGEN, Richardson and Wright, 1984; Cligen, Nicks and Gander, 1994; USCLIMATE, Johnson et al., 1996; ClimGen, Stöckle et al., 2001; Climak, Danuso, 2002; RadEst, Donatelli et al., 2003). All such approaches illustrate from different perspectives that there is actually a wealth of well developed solutions to the basic problem of estimating or generating weather data, coded in a variety of ways. The weather generation and estimation problem and the frequent need to evaluate alternative approaches in a comparative fashion has suggested that such methods be made available as one, comprehensive set of linkable components which also allow for extension by third parties of models that have already been implemented.

The objectives of this paper are: 1) to describe the models implemented in a set of software components (namely *CLIMA*) to both estimate and generate site-specific weather variables, and 2) to present the software design chosen to achieve reusability, interchangeability and extensibility of the components developed.

2. WEATHER MODELLING AT EARTH SURFACE LEVEL

A stochastic-deterministic modelling system is included to model a number of approaches approximating the same process. In particular, the system makes it possible, via separate components, to estimate/generate the climate variables required by models assimilating physical, biophysical and physiological processes in terrestrial ecosystems. These model components are: air temperature (*AirT*, Figure 1), evapotranspiration (*ET*, Figure 2), precipitation (*Rain*, Figure 3), solar radiation (*GSRad*, Figure 4), wind speed (*Wind*, Figure 5). These key weather variables are modelled on either daily or sub-daily time steps. Most of the weather models implemented have been extracted from peer-reviewed sources which are the most widely adopted concepts on weather generation and estimation. The relevant information was collected, interlinked and uniformly formatted into navigable structures to grant easy access to readers. Basic literature, not reported in the reference list, can be retrieved from the hypertext help files through the web site http://www.sipeaa.it/tools.

2.1. 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 alternative methods are implemented for generating daily values of T_{max} and T_{min}. The multi-stage generation system is conditioned on the precipitation status with both Richardson-type and

Danuso-type approaches. 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-global solar radiation correlation. A third approach (Remund&Page-type) 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 auto-regressive process from mean air temperatures and solar radiation parameters.

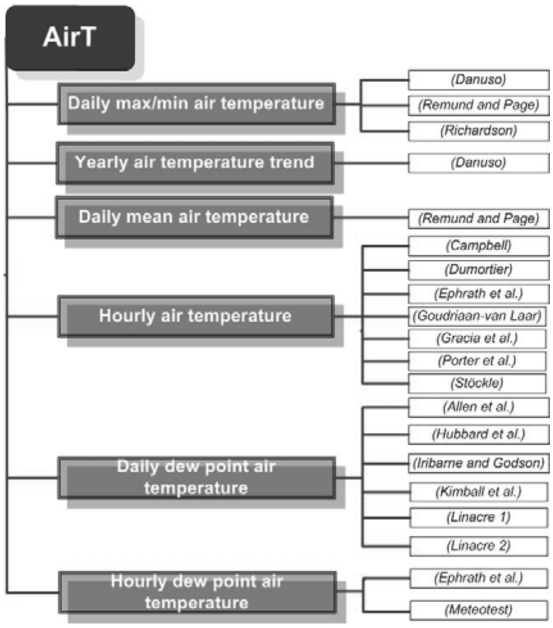

Figure 1. *AirT* models implemented for air temperature modelling.

Daily values of T_{max} and T_{min} are used to generate hourly air temperature values, according to alternative methods. Mean daily values of dew point air temperature are estimated via empirical relationships with T_{max} and T_{min} and other variables. A diurnal pattern (hourly time step) of dew point air temperature is also modelled via two alternative methods.

2.2. Evapotranspiration

Evapotranspiration for reference crops (ET_0) is calculated from alternative inputs, conditions and time steps, using one-dimensional equations based on aerodynamic theory and energy balance.

A standardized form of the FAO-56 Penman-Monteith equation is used to estimate daily or hourly ET_0 for two reference surfaces (0.12-m height, cool-season extensive grass such as perennial fescue or ryegrass; 0.50-m height, similar to alfalfa). The Priestley-Taylor equation is useful for the calculation of daily ET_0 for conditions where weather inputs for the aerodynamic term (relative humidity, wind speed) are unavailable. The aerodynamic term of the Penman-Monteith equation is replaced by a dimensionless empirical multiplier (alternative implementations). As an alternative when solar radiation data are missing, daily ET_0 can be estimated using the Hargreaves-Samani equation. The Stanghellini model estimates hourly ET_0 from a multi-layer canopy (well-developed tomato crop) grown in a single glass, Venlo-type greenhouse with hot-water pipe heating. The constituent equations of the Stanghellini model are from the American Society of Agricultural Engineers.

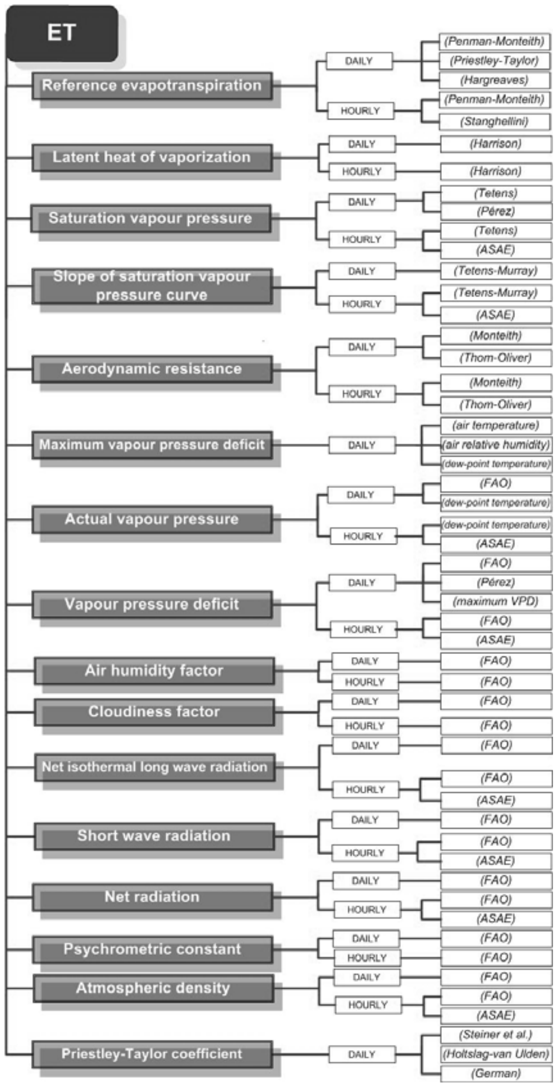

Figure 2. *ET* models implemented for evapotranspiration modelling.

2.3. Precipitation

The occurrence of wet or dry days is represented by a first-order Markov chain. The transition from one state to the other is governed by transition probabilities. According to multi-transition model, the daily precipitation amounts are divided into a number of up to seven states - dry, or wet from 1 (lowest level of rainfall) to 6 (highest level of rainfall). On days when precipitation is determined to occur the precipitation amount is generated by sampling from alternative probability density functions Most approaches are based on the two-state transition for dry/wet days. The Gamma distribution is even used to model precipitation amounts for the highest level of rainfall in the multi-state transition probability matrix, while linear distribution is applied for the other states. The pattern of the Gamma distribution, plus linear distribution across various occurrence states, exhibits a combined J shaped function. Short-time rainfall data are generated by disaggregating daily rainfall into a number of events, then deriving the characteristics (amount, duration and starting time) for each event.

Four approaches have been implemented to disaggregate daily amounts into 6-hour or shorter resolution (as small as 10 minutes) amounts. The method described by Arnold and Williams targets at 0.5-hour time resolution and assumes that daily rainfall falls in only one event. The peak location is generated first according to a broken linear distribution. The other 0.5-hourly amounts are generated from an exponential distribution and located on both sides of the peak.

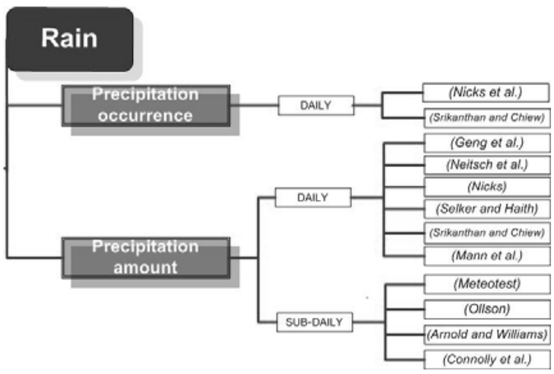

Figure 3. *Rain* models implemented for precipitation modelling.

An autoregressive process and a Gaussian daily profile model are combined in the Meteotest approach to simulate the possibility of precipitation at any hour. Two options are available to generate sub-daily precipitation events at varying time steps. A cascade-based disaggregation brings about the break-up of a time interval into two equally sized sub-intervals. The total amount is redistributed into two quantities according to two multiplicative weights from a uniform distribution: 24-hour rain into two 12-hour amounts, 12-hour amount into 6-hour amounts and so on until a 1.5-hour resolution is achieved. A further approach allows disaggregation of daily rainfall into multiple events on one day, and the simulation of time-varying intensity within each event: (1) distinct storms are assumed to be independent random variables from a Poisson distribution, (2) the storm origins arrive according to a beta distribution, (3) storms terminate after a time that is simulated by a simplified gamma distribution, (4) each storm intensity is a random value exponentially distributed, (5) the time from the beginning of the event to peak intensity is given by an exponential function, (6) peak storm intensity for each event is determined from an exponential function, (7) internal storm intensities are represented with a double exponential function.

2.4. Solar radiation

Solar radiation outside the earth's atmosphere is calculated at any hour based on routines derived from the solar geometry. Daily values are an integration of hourly values from sunrise to sunset. The upper bound for the transmission of global radiation through the earth's atmosphere (i.e. under conditions of cloudless sky) can be set to a site-specific constant or estimated daily by diverse methods. Broadband global solar radiation (about 0.3-3.0 μm wave-band) striking horizontal surfaces of the earth daily is estimated from alternative sets of weather inputs according to strategies based on either physical relationships or stochastic procedures. A sine-curve assumption is used to prescribe the hourly distribution of solar radiation from its daily value, assuming changes with solar elevation angle.

The most simplified models relate diurnal temperature range to solar energy transmission through the earth's atmosphere. As one of the most important phenomena limiting solar radiation at the earth's surface is clouds, a cloud cover measure is incorporated in one of the models supplied to estimate transmissivity. The Winslow-type radiation model uses saturation vapour pressures as a measure of the atmospheric transmission of incident solar radiation. The Ångström-Prescott model is the most common choice to estimate global solar radiation when sunshine measurements are available. As an alternative, an implementation of the Johnson-Woodward model is given. Stochastic generation is based on the dependence structure of daily maximum and minimum temperature, and solar radiation (Richardson-type). Such variables are reduced to time series of normally distributed residual elements with mean zero and variance of one. An autoregressive, weakly stationary multivariate process is used to generate the residual series. Daily values of global solar radiation are generated for dry and wet days as daily deviations above and below the monthly average value. An alternative method is also given.

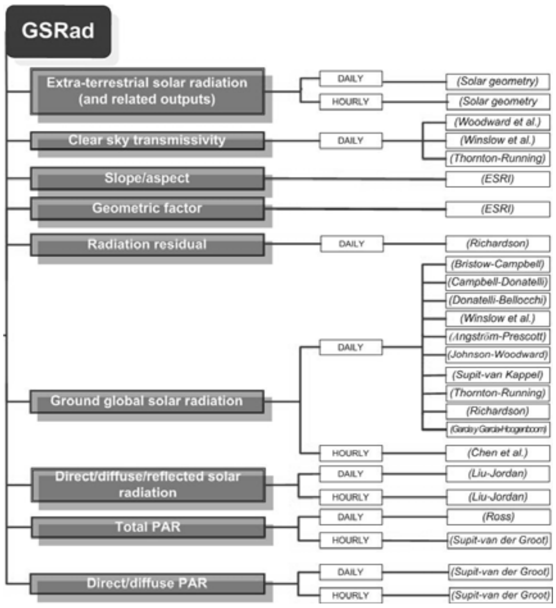

Figure 4. *GSRad* models implemented for solar radiation modelling.

The current implementation splits flux density at the earth's surface into direct, diffuse and reflected fractions (Liu-Jordan concept). The estimation of diffuse radiation on a horizontal surface depends on the extra-terrestrial irradiance and a transmission function. Hourly transmission relies on the anisotropic assumption for estimation on inclined surfaces and is further divided into the isotropic, circumsolar and horizontal ribbon sub-fractions. These sub-fractions are calculated separately and then summed to provide the diffuse irradiance. Ground reflected irradiance is estimated from a slope-dependent factor. Direct fraction of solar radiation is the complement to global solar radiation. The visible band (0.38-0.71 µm wavelength) is estimated daily by the diffuse/direct radiation ratio, and hourly by the solar elevation course. *PAR* amount can also be disaggregated into direct and diffuse fractions. Slope and aspect angles, required to compute geometric factors that convert radiation estimates from horizontal to non-horizontal surfaces, can be derived from digital elevation data grids (http://www.esri.com).

2.5. Wind speed

Daily mean wind speed is generated by sampling from alternative probability distribution functions. Following generation of daily mean wind speed, alternative approaches are available to estimate the maximum and minimum wind speeds for the day. Probability density functions are used to distribute daily mean wind speed within the day. Alternatively, wave functions are used to describe diurnal wind speed variations.

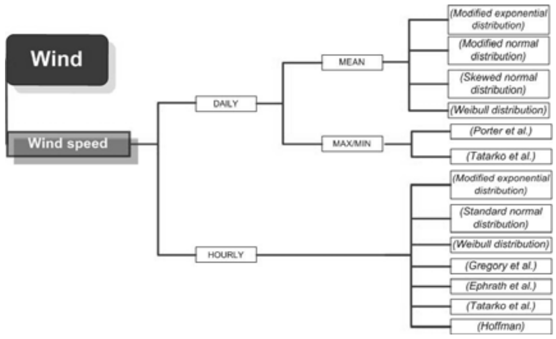

Figure 5. *Wind* models implemented for wind speed modelling.

3. SOFTWARE DESIGN

CLIMA is a set of components written in C# for .NET, most of them deployable and re-usable outside the *CLIMA* framework. *CLIMA* software (Figure 6) is provided with a sample client with a graphical user interface, including an application (*MCE*: Model Component Explorer) which makes it possible to discover inputs, parameters and outputs of alternative models (i.e. “strategies”, section 3.2) in each of the five model components. The component *ClimaCore* includes the functionalities of the component as described in section 3.1. Besides the five model components described in the previous sections, *CLIMA* also includes components to:

- estimate parameters
- evaluate/estimate missing data in time series
- provide data at run-time for use in simulation frameworks

Common features of model components are:

- full documentation as hypertext help files (regarding the modelling solutions and regarding code, NDoc style, http://ndoc.sourceforge.net)
- sample clients inclusive of source code (WinForms applications in C#)
- possibility of extending models by adding new ones without recompiling the component and still using the same signature of the call
- test of pre- and post-conditions
- unit tests for all public methods

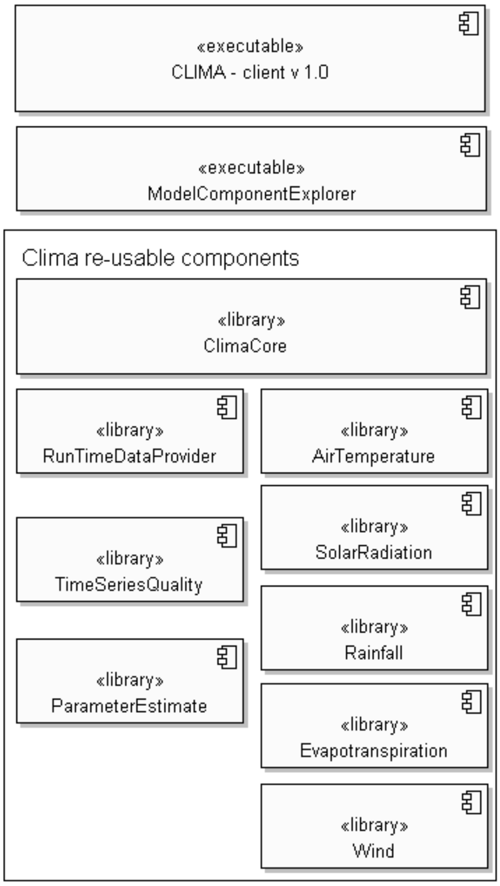

Figure 6. Main *CLIMA* components.

3.1. The *ClimaCore* component

Functionalities for data storing and retrieval are implemented in this component, based on a relational database for location and model parameters data and a file system to store both observational and generated data. The database includes tables to store user access rights. Data can be exported to various formats via specific drivers which can be extended without the need to recompile the component. This component makes it possible to create configurations for data generation (i.e. selection of location, variables and relevant strategies).

3.2. Model component architecture

Model components are implemented separating the model equation component interface and its implementations into different software units as shown in Figure 7. This design feature allows concurrent development of both components and clients, and makes it possible to define units of *reusability* - model component implementations and model component interfaces - and units of *interchangeability* - model component implementations alone (Rizzoli *et al*., 2005).

All models can be called up explicitly with or without the test of pre- and post-conditions via two overloads of the method `Estimate()`. The design-by-contract approach (Meyer, 1997) requires pre-conditions (in this case input values within a given range, and concurrent conditions, e.g. $T_{max} > T_{min}$) to be respected in order to obtain results complying with post-conditions (in these implementations, output values within a given range). Although respecting pre-conditions is a responsibility of the client application, the components implement some options to deal with input data violating pre-conditions, and to check post-conditions. If pre-conditions are violated, exceptions may occur. Exception handling both prevents the application from crashing, and provides the user with information about the type and location of the exception. Implementing the test of pre- and post-conditions also forces developers of biophysical models to state the numerical conditions which need to be respected in order to use the model correctly. All logical components depend on a service component (namely, *Preconditions*) which implements the test of pre- and post-conditions, and makes it possible to obtain the output test on screen, txt or XML files, or custom developed format. The *Preconditions* component is also used for form validation into the *CLIMA* client.

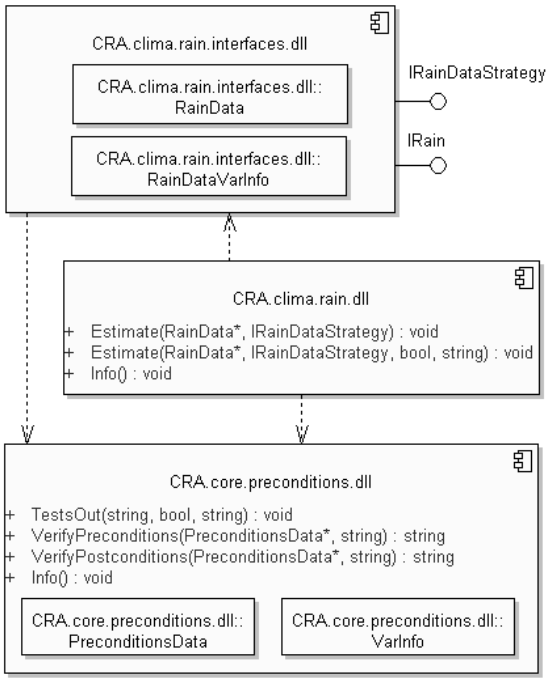

Figure 7. Component diagram of *AirT*, *ET*, *GSRad*, *Rain*, and *Wind* components (the *Rain* diagram is shown)

Each model is implemented as a class. Models can be simple or composite, in the latter case an association of classes of the same assembly is implemented. Both simple and composite classes are used as strategies, and they encapsulate parameter declaration and tests of pre- and post-conditions.

The developer of a software application which uses the *CLIMA*'s components may wish to make available to end users approaches other than those currently implemented. This can be done by using the design pattern "strategy" (Gamma et al., 2004; Mesketer, 2004, C# implementations). This software design makes it possible to encapsulate new models without recompiling the component. Sample applications are provided to show how to extend component models.

CLIMA (full package) and its components (independently packaged) are freely distributed through the web site http://www.sipeaa.it/tools. The installation includes compiled subcomponents, and the source code of sample applications.

4. CONCLUSIONS

The goal of *CLIMA* development is to extend access to weather modelling features to multiple users, and to provide an architecture to ensure re-use and extension of coded models. *CLIMA* attempts to overcome some of the technical challenges that to date have limited the development of reusable weather generation and estimation capabilities. Via their model documentation and the metadata associated with each variable, *CLIMA* model components are also a way to share modelling knowledge in an extensible way.

5. ACKNOWLEDGMENTS

This publication has been partly funded by the SEAMLESS integrated project, EU 6th Framework Programme for Research, Technological Development and Demonstration, Priority 1.1.6.3. Global Change and Ecosystems (European Commission, DG Research, contract no. 010036-2) and by the Italian Ministry of Agricultural and Forestry Policies, project SIPEAA (http://www.sipeaa.it), paper no. 32.

6. REFERENCES

Argent, R.M., 2004. An overview of model integration for environmental applications - components, frameworks and semantics. Environmental Modelling and Software 19: 219 - 234.

Argent, R.M. and A.E. Rizzoli, 2004. Development of multi-framework model components, iEMSs 2004 Conference, Osnabrück, Germany, 1, 365-370.

Donatelli, M., G. Bellocchi, and L. Carlini, 2005. A software component for estimating solar radiation. Environmental Modelling and Software. (in press).

Donatelli, M., G. Bellocchi, and F. Fontana, 2003. RadEst3.00: Software to estimate daily radiation data from commonly available meteorological variables. European Journal of Agronomy, 18: 363 - 367.

Gamma, E. R. Helm, R. Johnson, and J. Vlissides, 1994. Design patterns: elements of reusable object-oriented software. Addison-Wesley, Boston, MA, USA

Hutchinson, M.F., 1987. Methods for generation of weather sequences. In: Bunting A.H., Agricultural environments: characterisation, classification and mapping. CAB International, Wallingford, United Kingdom, 149 – 157.

Johnson, G.L., C.L. Hanson, S.P. Hardegree, and E.B. Ballard, 1996. Stochastic weather simulation: overview and analysis of two commonly used models. Journal of Applied Meteorology, 35: 1878 - 1896.

Mesketer, S.J., 2004. Design patterns in C#. Addison-Wesley, Boston, MT, USA.

Meyer, B., 1997. Object-oriented software construction, 2nd edition. Prentice Hall, Upper Saddle River, NJ, USA.

Nicks, A.D., and G.A. Gander, 1994. Cligen: a weather generator for climate inputs to water resources and other models. 5th International Conference on Computer in Agriculture, Orlando, FL, USA, 3 - 94.

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

Rizzoli, A.E, M. Donatelli, I. Athanasiadis, F. Villa, R. Muetzelfeldt, D. Huber, 2005. Semantic links in integrated modelling frameworks. Proc. of the MODSIM conference, 12-15 December, Melburne, Australia.

Stöckle, C.O., R.L. Nelson, M. Donatelli, and F. Castellvì, 2001. ClimGen: a flexible weather generation program. 2nd ISMCS, Florence, Italy, 229-230.