APPLICATION OF THE UNIFIED MODELLING LANGUAGE TO A SPATIALLY DISTRIBUTED MICROMETEOROLOGICAL MODEL AT CATCHMENT SCALE

APPLICAZIONE DELL'UNIFIED MODELLING LANGUAGE A UN MODELLO MICROMETEOROLOGICO A SCALA DI BACINO

Rossana Monica Ferrara e Gianfranco Rana*

CRA – Istituto Sperimentale Agronomico, Via C.Ulpiani 5, 70125 Bari, Italy
** Corresponding author: Tel.: +39 080/5475026. Fax: +39 080/5475023 E-mail address: Gianfranco.rana@entecra.it*

Ricevuto 16 dicembre 2005, accettato 23 marzo 2006

Abstract

A micrometeorological model able to produce a spatially distribution at catchment scale of energy fluxes and agro-meteorological variables in a complex terrain has been produced. This model has the advantage to work with enough accuracy, using easily available input variables. It gives an accurate description of the interaction soil-plant-atmosphere in hilly landscapes taking into account the effect of the topographic characteristics influencing the microclimate.
The building of the software system relative to such complex model which contains climatic, topographic, soil and crop inputs, and which gives a spatial distribution of a lot of variables at short time scale can be simplified by adopting the Object Oriented approach. Following this paradigm and meeting the opportunity to produce a model which could work in a stand-alone application or in an integrate environmental model we have adopted the Unified Modelling Language (UML) to support the entire software development lifecycle.
In this paper we discuss how by means of the UML we have produced a clear representation of our micrometeorological model, without the explicit use of any specific formalism or programming language.

Key words: Object Oriented Programming, energy fluxes, agro-meteorological variables, hilly landscape, environment.

Riassunto

È stato realizzato un modello micrometeorologico in grado di fornire la distribuzione spaziale a livello di bacino di flussi di energia e di variabili agro-meteorologiche in terreni dalla topografia complessa. Questo modello ha il vantaggio di lavorare con accuratezza utilizzando variabili di ingresso facilmente reperibili. Esso fornisce una descrizione accurata dell'interazione suolo-pianta-atmosfera in territori collinari, prendendo in considerazione l'effetto delle caratteristiche topografiche che influenzano il microclima.
La costruzione del sistema software di tale modello in cui intervengono inputs climatici, topografici, di suolo e di coltura, e il cui output è la distribuzione spaziale di un gran numero di variabili su breve scala temporale si può semplificare utilizzando l'approccio della programmazione Orientata agli Oggetti. Seguendo questo tipo di orientamento e rispondendo alla necessità di produrre un modello capace di lavorare in una applicazione stand-alone e/o in un modello ambientale integrato si è utilizzato l'Unified Modelling Language (UML) per supportare l'intero ciclo di sviluppo del software.
In questo articolo viene esposto come per mezzo dell'UML abbiamo potuto produrre una chiara rappresentazione del nostro modello micrometeorologico, senza far riferimento ad alcun formalismo specifico o linguaggio di programmazione.

Parole chiave: *Programmazione ad oggetti,flussi di energia,variabili agro-meteorologiche, paesaggi collinari ambiente*

Introduction

The planning of the agricultural management of a region needs the knowledge of some agro-environmental variables spatially distributed in a homogenous area at short time scale (Hobets *et al.*, 1999; Smith, 2000). The catchment scale seems to be the best space unit for this kind of analysis (Jacquemin and Noilhan, 1990; Silberstein *et al.*, 1999; Troch *et al.*, 2003), in fact it is large enough to meet the farm prospective and in the same time small enough to accurately analyse the relationships between the terrain and the micrometeorology of a crop.
To produce a complete agro-environmental model at catchment scale we have to correctly describe the interactions soil-plant-atmosphere and to know the distribution in space of microclimatic variables like air temperature, wind speed, radiation and, for cropped field, evapotranspiration and soil water content. This work concerns the spatial distribution of all these agro-meteorological variables in hilly landscapes.
Interaction between cropped surfaces and atmosphere in complex terrains were largely studied in recent works (i.a. Vázquez and Feyen, 2003; Courault *et al.*, 2003). Despite this effort, in our knowledge, any model can be found in literature giving the spatial distribution of micrometeorological variables in a catchment, accurate enough to be successfully used in practice, but having easily available inputs.
Such a micrometeorological model (MM), to share practical purpose, must be able to determinate all terms of

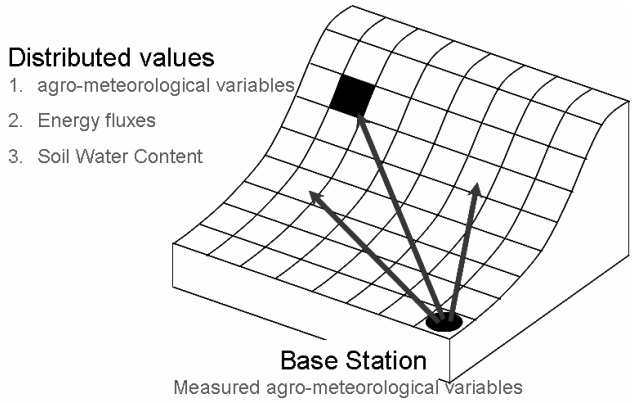

Fig. 1 - Schematic view of the logical path of the Micrometeo-rolo-gical Model.
Fig. 1 - *Vista schematica del percorso logico del Modello Microme-teorologico*

the energy balance (net radiation, latent, sensible and soil heat fluxes) and agro-meteorological variables in all points of a catchment, over a complex terrain, starting from the standard agro-meteorological data acquired in a standard station sited in a single point of the catchment, hereafter called "reference point". In our case, this latter, for simplicity, has been chosen in the bottom of the catchment in a plane place without slope and face exposition. A schematic view of the logical path followed by our model is shown in Fig. 1.

For our purposes, the catchment will be represented as a regular matrix of square cells within which all significant parameters relative to soil and crop are assumed to be homogeneous. For our application, we consider catchment that have a surface of the order of 10 ÷1000 ha. The cell of the hypothetic grid could be of 10×10 m^2 up to 75×75 m^2, following the current Digital Terrain Model (DTM) and/or Digital Elevation Model (DEM) space scale available.

The agro-meteorological variables and the fluxes we want to determine in each cell to have the complete comprehension of the catchment environment are: net radiation (R_n), surface temperature (T_{surf}), air temperature (T_{air}), wind speed (u), global radiation (R_g), soil heat flux (G), sensible heat flux (H) and latent heat flux (λE). All these variables must be obtained starting from measurements at a reference point of standard agro-meteorological variables: global radiation, air temperature, humidity, speed and direction of wind, rainfall, together with topographic characteristics of the site and soil characteristics. For agricultural catchments also crop characteristics have to be taken into account.

The topographic characteristics influencing the agro-meteorological variables and the energy fluxes in an agricultural field are the slope, the altitude and the aspect (orientation) of the considered plot (Huntingford *et al.*, 1998; Van Wesemael *et al.*, 2003). In particular, the topographic characteristics of a cell affect the solar radiation incoming on the surface, the wind speed field, the energy balance and, as consequence, the surface and air temperature (Wood, 1991; Silberstein *et al.*, 1999; Fu and Rich, 2002; Lookingbill and Urban, 2003).

The building of a software system relative to such a complex model which contains climatic, topographic and crop inputs, and which gives the spatial distribution in a catchment of agro-meteorological variables and fluxes at short time scale, can be simplified by adopting the Object Oriented approach. Recently the software engineering is generating tools to simplify the production of software application, giving techniques to improve the development of well-documented environmental models, constructed in a way to make easy reuse and/or modification of code. Moreover, the traditional programming based on the concept of the single stand-alone application has been exceeded by the component-based approach (Szyperski, 2002; Papagjorji *et al.*, 2004; Donatelli *et al.*, 2004;) which consists in the building of a set of reusable components that can be assembled in different ways in function of the required application. The production of new models starting from existing ones needs clear documentation that helps to evaluate and to select the components in function of the data availability and the needed level of complexity.

With the purpose to produce a software system of our micrometeorological model following the Object-Oriented Paradigm, meeting the opportunity to work in a stand-alone application or in an integrate environmental model we have adopted the Unified Modelling Language (UML) to support the entire software development life-cycle.

In this paper we discuss how the use of a standard methodology based on UML can help the production of a clear representation of a complex micrometeorological environmental model, without the explicit use of any specific formalism or programming language.

Material and method

Outline of the micrometeorological model

It clearly is beyond the scope of this paper to explain the science and algorithms behind our model, which are being detailed in future specific works, nevertheless we need to introduce some fundamental concepts very useful to understand the structure and the functionality of our model for the development of the running software.

We divided our experimental catchment is a grid with n cells and we considered the lowest cell in plan as our reference site.

At first, we suppose that the energy balance (i.e. how the available energy is used by the soil-crop-atmosphere system) and the soil water budget (how the water is used by the crop) in the reference point are modified by the topographic characteristics of each cell "i" in which our catchment is divided.

Using the Activity diagram showed in Fig. 2 we can describe synthetically how our model works. First of all the MM has to load the inputs. Then it starts the computation with the determination of the energy fluxes and the soil water content at the reference site: this is made by implementing algorithms permitting the closure of the energy and of the soil water balances. At this point the model gives the first required outputs and goes to compute the energy and soil water balances at the cell "i", using the topographic characteristics of the cell.

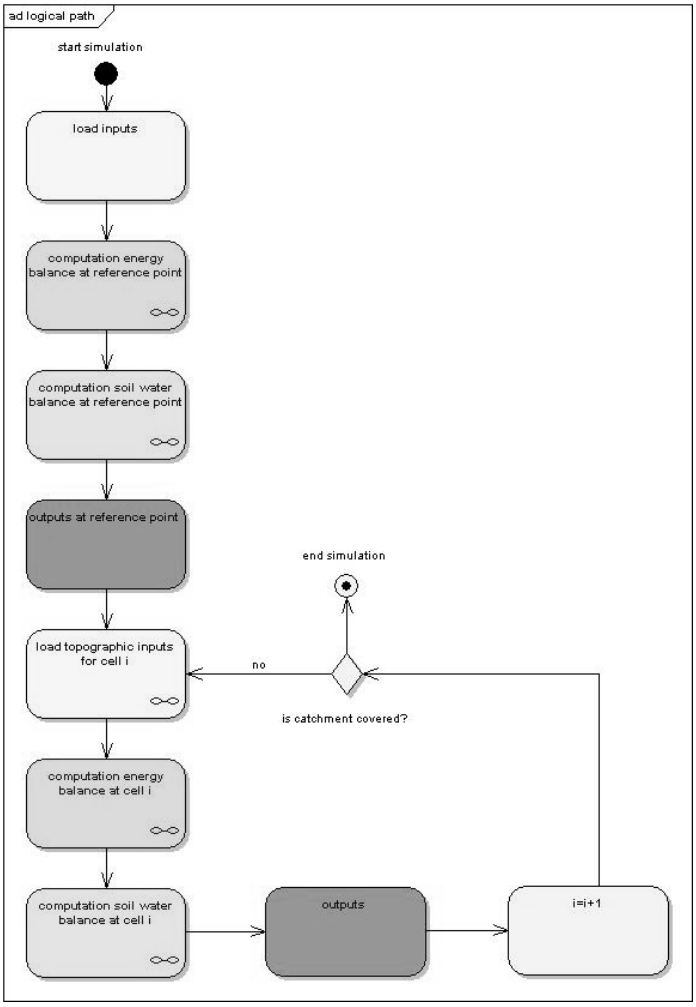

Fig. 2 - Activity diagram for the logical path of the Micrometeorological Model..

Fig. 2 - *Diagramma delle Attività per il percorso logico del Modello Micrometeorologico*

This procedure is repeated, for each time step, over each cell until the catchment is completely covered and the MM saves all required outputs in files of data.

Starting from this general view of the model, now, we give some information relative to the specific approach adopted for modelling the involved variables.

The energy balance is written as:

$$R_n = \lambda E + H + G$$

The R_n term represents the incoming energy in the soil canopy system and it is calculated by the radiation balance at short and long-wave lengths. From a functionally point of view R_n is expressed as:

$$R_n = f\left(R_g, T_{surf}, T_{air}\right)$$

where the global radiation is given by the sum of the direct and the diffuse solar radiations that, obviously, for a field in hilly terrain are affected by the topography. We assume that the fraction of radiation diffused by the atmosphere and calculated at the reference site is the same in all the cells of the catchment and in particular we use the modelling proposed by Spitters *et al.*, (1986) and Fu and Rich, (2002). For the direct solar radiation we use the Lambert's cosine law including topographic characteristics of the field. So, the short-wave radiation balance is calculated using the information relative to the albedo of the surface, while to produce the long-wave radiation balance we calculate the downward and the upward long-wave radiations given by mean of the black body law in function of the air temperature and of the surface temperature, respectively.

The air temperature can be calculated as a function of the following variables:

$$T_{air} = f(T_{surf}, slope)$$

where we are adopting the standard logarithmic profile corrected by an experimental function to take into account the slope (Personnic, 1999).

Analogously to temperature, the wind speed can be modelled in function of the topographic effect. In particular, we can follow the study by Taylor and Lee (1984) that has been tested in hilly fields in Northern France with very good results (Personnic, 1999). In this approach the wind speed u in the generic cell is given as:

$$u(i) = f(u_{ref}, diru_{ref}, A(i), shape, \Delta z(i))$$

where u_{ref} and $diru_{ref}$ are the speed and the direction of wind measured at the reference site, $A(i)$ is the azimuth of the cell, *shape* is the shape of the hill (asymmetric, step, bi-dimensional) and $\Delta z(i)$ is the difference between the mean height of the cell and the reference site.

The sensible heat flux H is modeled using a flux-gradient approach:

$$H = f(T^*, u^*)$$

where T^* and u^* are the temperature scale and the friction velocity respectively.

Using H and the estimated values of the wind speed we calculate the soil heat flux G:

$$G = f(u, H)$$

using an empirical function (Cellier *et al.*, 1996) chosen as the most appropriate among the available expressions current in literature considering the structure and the approach of our MM.

The last outgoing term of the energy balance is the latent heat flux λE which is the more difficult for modelling. We consider the sum of the evaporation from the soil and the transpiration by the vegetation and in both cases we follow a resistance approach as suggested by Jacquemin and Noilhan, (1990):

$$\lambda E = \lambda E_g + \lambda E_v$$

The evaporation from the soil, E_g, is expressed in function of the fraction of the soil covered by the canopy (f_v), the relative humidity of the surface layer (h_u) calculated in function of the soil water content (w_g) (Noilhan and Planton, 1989), the air and surface temperatures and the aerodynamic resistance (r_a):

$$E_g = f(f_v, h_u, T_{air}, T_{surf}, r_a)$$

The transpiration by the crop, E_v, has the same functional dependence of E_g except the h_u and, furthermore, it depends by the surface canopy resistance, r_s, function

of the soil water content and calculated following the Jarvis model (Jarvis, 1976).
It is clear the correlation between the energy balance and the soil water balance: in fact the actual evapotranspiration is determined by the soil moisture content which, in its turn, is function of the loss of water using the evaporation from the soil and the transpiration by the vegetation.
Our soil water balance model uses the ISBA approach which calculates the actual soil relative volumetric moisture (w_g) and the soil relative volumetric moisture in deep layer (w_{g2}) following the model developed by Noilhan and Planton, (1989). In particular, w_g is calculates as:

$$w_g = f(P_g, E_g, w_{geq})$$

where P_g is the effective rain and w_{geq} is the soil moisture at equilibrium. In the same way w_{g2} is given as:

$$w_g = f(P_g, E_g, E_v)$$

Considering all these functional relathionships to reproduce the behaviour of energy fluxes and agrometeorological variables affected by topography, we produced a model that can estimate the most appropriate values of all considerate unknown variables by using an iterative method. Moreover, one of the peculiarity of our model is the link between the reference point and the cell which is given by the meteorology at the height of the boundary layer. In particular, we consider the atmosphere above a hill divided in two layers: one above the inner boundary layer, which height z_{bl} is calculated using the relationship given by Jackson and Hunt (1975), and another one below this height. The difference between these two layers is the effect of the surface topography on the microclimate: in particular, topography influences only the inner layer's microclimate. So to calculate the air temperature in slope we use the following path:

$$T_{surf}(ref) \rightarrow T_{zBL}(ref) \rightarrow T_{zBL}(i) \rightarrow T_{surf}(i) \rightarrow T_{air}(i)$$

This overview on the theoretical formulation shows the complexity of our model and the strong linking among all its components. Obviously, the translation of a such model in a software application needs a tool capable to give a clear documentation on model's functionality and structure in order to give everyone the possibility to understand what the model does and how it does it.

The UML

Recently the Unified Modelling Language is emerging as a method to design and analyse Object Oriented system: in 1997 it has been adopted by the Object Management Group (OMG) as a standard notation for modelling and for software development. In particular, it is a visual modelling language developed to specify, to visualize, to construct and to document all the artefacts of a software system (Jacobson *et al.*, 1998).
The reasons on the basis of the use of UML for modelling are numerous. First of all, UML gives a common language between analysts and experts of a domain, designers and developers of the relative software, facilitating the collaboration. Moreover, knowing exactly where the adjustments of a code must made, it becomes easy the modification and the maintenance of a model, and, eventually, the incorporation of a model or of parts of models into other larger new applications.
Another important aspect is that using this methodology it is possible to give a global view of the system and of the science behind the model: in particular UML notation allows to identify data input requirements, processes, equations and documentation of the organization of the relative code (Papajorji and Shatar, 2004; Papagjorji *et al.*, 2004). Moreover, the model developed in UML is independent from the language of programming, so the user can choose the more adequate programming language to implement his software system.
UML consists of a set of different basic diagrams and a notation to model in a easily understandable way, providing multiple perspectives of the software system under development. In particular, there are specific diagrams to provide the structural-static viewpoint of the model which gives a description of the entities of the system and the relationships among them, and others relative to the dynamic-behavioural viewpoint to describe the system in time. A schematic classification of the UML diagrams is given in Tab. 1.
The description of this standard notation can be very long and it is not the purpose of this paper, in which we will give only some general information needed to clarify our adopted methodology to produce the UML version of our MM: it is possible to find more information about UML in Booch *et al.*, (1999).
First of all, developers have to select the more adequate UML diagrams to develop the information system relative to their project and purpose. This is what we have done choosing the UML diagrams to clearly represent different aspects of our MM.
From the structural view, a representation of the functionality of the system is given using Use Case diagrams to describe what the system should do, without giving how the system should do it. A Use Case diagram for a system includes *use cases* and *actors*. *Use Cases* represent the functional requirements of the system, that are the services provided by the system in a specific situation; an *actor* represents an abstraction outside the system that interacts with *use cases* expecting to receive their response.
One of the most important transition in the UML notation is from *use cases* to the representation of the system

Tab. 1 – The sub-division of the UML diagrams in function of the adopted viewpoint.

Tab. 1 *– Suddivisione dei diagrammi UML in funzione del punto di vista adottato.*

Structural-static viewpoint	**Dynamic-behavioural viewpoint**
Use case diagrams	Activity diagrams
Component diagrams	State-chart diagrams
Package diagrams	Collaboration diagrams
Class and Object diagrams	Sequence diagrams
Deployment diagrams	

in terms of *classes*: in particular it is possible to identified *classes* from *use cases*' aims. Then, the Class diagrams are used to describe the type of objects of the system and their static relationships: they consist of *classes*, defined in terms of attributes and methods, linked by means of relationships. In detail, attributes correspond to the input data needed by the model in development, methods represent functions to calculate or manipulate data, while relationships allow the transfer of data among *classes* and they indicate the sharing of resources. To produce a good model it is also necessary that in each *class* the relative roles are indicated. In the Class diagram there is also the possibility to indicate the needed pre-conditions and post-conditions allowing the use of the *design by contract* approach for designing computer software: by means of these obligations the limits of the applications are clearly indicated.

A set of *classes* with similar functionality can be grouped to create a UML *package*: the identification of *packages* is a practical and recommended way to identify the main modules of the information system, giving a first map of the principal elements of the system and their interactions. Then, for each *package* a Class diagram is constructed to have a visual summary of the content of the *package*.

Another UML tool relative to the static view of a system is the Component diagram: it allows to capture the architecture of the system which has to implement *use cases*, to organize source code and to explain organization and relationships among software modules.

Considering the dynamic viewpoint of an information system, UML offers modelling techniques that help to describe processes and behaviours: in particular, there are Sequence diagrams and the Activity diagrams.

The Sequence diagrams describe the flow of messages between objects and focus on the order in which the messages are sent over time: they give a representation of the event flow, called "workflow", and of how elements in general cooperate over time to achieve a specific result.

The "workflow" is also described in terms of the Activity diagrams: they are used to represent the sequence of steps and conditions in a computational process (*actions*) by modelling the flow of control from *activity* to *activity*. This type of diagrams is very important to understand the code of an information system in which there are complex algorithms like is our model; moreover the description of the full "workflow" can require many diagrams, so it can be divided in "sub-workflows" relative to *sub-activities*.

There are many CASE tools available for UML diagramming within development process: two examples are Enterprise Architect (EA) by Sparx Systems and Together Architect by Borland. Particularly, in Borland enterprise development environments the UML modelling is standard tool.

Both EA and Together support the process "round trip engineering" for Object Oriented Programming. This process converts a UML class model into source code via *forward* engineering and back again via *reverse* engineering. The *forward* engineering process translates the structure of each class into code in which the attributes are named using the names of the class diagram, and empty skeletons for each method are created: it will be the user who needs to introduce in each method the algorithm relative to the chosen approach. Moreover, the reverse engineering is very useful to create UML models starting from an existing source code, helping to identify logical errors and giving the possibility to reuse and improve source code developed for other applications. Round trip engineering requires specific languages like VB6, C# and VB.NET.

Moreover, the more advanced technology available in the CASE tools based on UML allows to continuously update the model from the source code and vice versa. In particular, working on the code the UML, diagrams change in real time, while corrections into the source code are directly produced after modifying the UML diagrams.

After this overview on the UML tools selected to produce the UML version of our MM, in the following section we report examples of the developed UML diagrams. All the UML documentation about our model can be found on http://www.inea.it

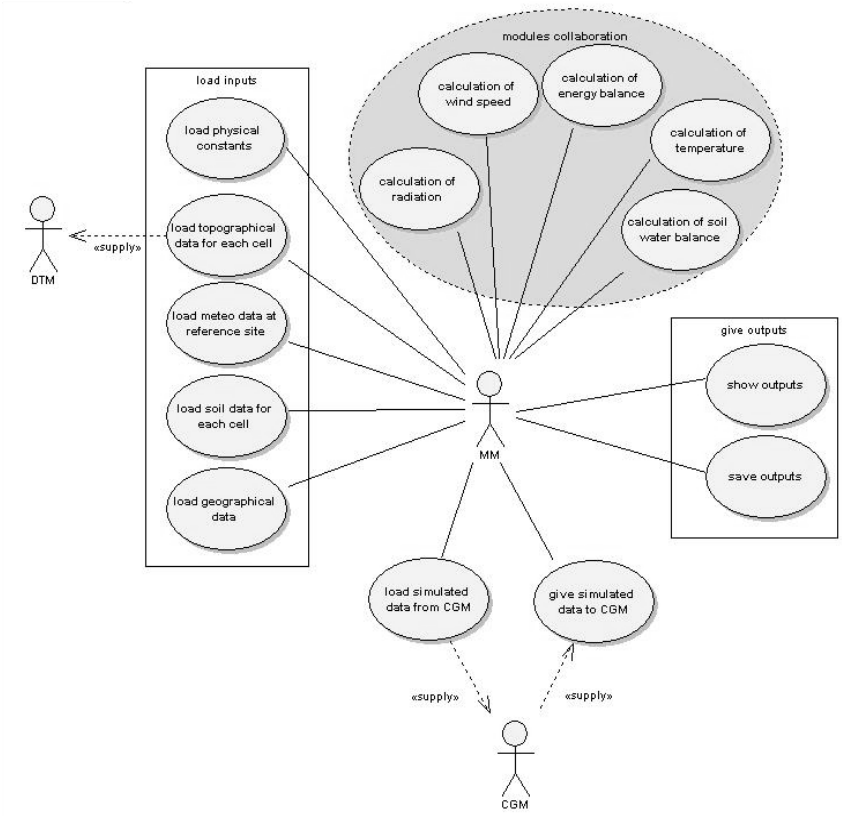

Fig. 3 - The Use Case diagram for general view of the Micrometeorological Model.

Fig. 3 - *Il diagramma dei Casi d'Uso relativo al Modello Micrometeorologico visto in generale.*

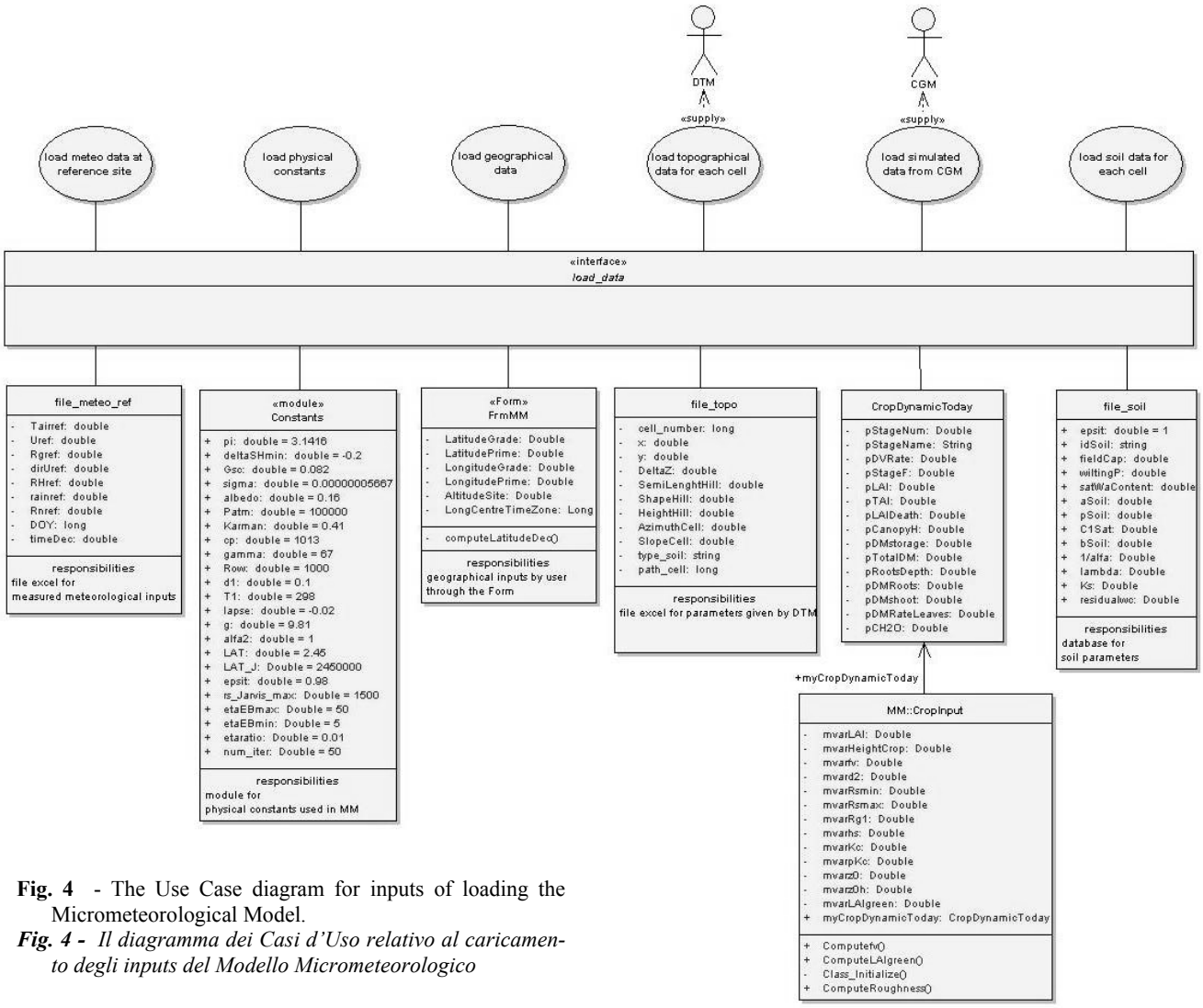

Fig. 4 - The Use Case diagram for inputs of loading the Micrometeorological Model.

Fig. 4 - *Il diagramma dei Casi d'Uso relativo al caricamento degli inputs del Modello Micrometeorologico*

/isa/stamina/umlwebpage/index.htm. In the following we show and describe the most general UML diagrams produced for the realization of the MM.

To produce the UML version of our MM we have used the Enterprise Architect 4.0 software (http//:www.sparxsystems.com).

Results: UML diagrams for the micrometeorological model

The starting point in the development of a complete model of a software system using the UML methodology is the production of the Use Case diagrams. Fig. 3 gives a synthetic visual map of our MM how it is seen by external users. From this it is clear what our model should do: first of all it has to load needed input data i.e. physical constants, soil parameters, meteorological and geographical data; moreover, it has to load data from others models: a DTM model relative to the required topographical data and a Crop Growth Model at which at the same time it can give simulated data. Moreover, users can easily understand that our model executes calculations of radiation, wind speed, energy balance, temperature, soil water balance and lastly it manages outputs. Nevertheless, in this diagram there are not information about how the MM does all these things, but it is evident the possibility of using our model as a component of a most complex environmental model, even if in this stage it is not explicated how this will do.

Starting from this diagram we have developed Use Case diagrams most detailed: for example it can be useful to show the Use Case diagram produced to describe in detail the loading of input data. Fig. 4 shows that the MM loads, using an interface, the required inputs from the relative databases: the name and the type of the required variables are also indicated as well as the type of file. This simple scheme can help the user to understand what are the needed inputs to run the model and the structure of needed databases.

For the static view of the MM also Component diagrams are useful: in Fig. 5 there is the general one corresponding to our model. It is clear that our model consists of two *components* one relative to the energy balance and the other one relative to the soil water balance. The former, in its turn, consists of four *components* to calculate the radiation, the wind speed, the energy balance and the temperatures, where each *component* is a representation of a source code file. In this diagram relations of composition (full diamond) indicate that the *components* of the system have the same lifecycle of the all. Moreover, the energy balance consists of two *components*, one to implement the procedure relative to the reference site and

the other one to make calculations for each cell of the catchment: the relationship used in this case is an aggregation (empty diamond) that is less strong than composition; in fact the model can work only at the reference site and/or can compute also at the generic cell. Moreover, the exchanged data among the *components* are indicated on the dependency lines: for example, it is clear the link between the soil water balance *component* and the energy balance *component*: the former needs the evapotranspiration calculated by the latter which in its turn uses the soil water content in deep layer to compute the evapotranspiration.

To complete the structural view of our model we have developed the Package and the Class diagrams. In Fig. 6 there is the wide-ranging Package diagram which reproduces the component subdivision as well as the indication of *classes* relative to each *package*.

Taking into consideration the behavioural view of the MM the development of the Activity and the Sequence diagrams are needed for our project. Fig. 7 illustrates the "workflow" of the MM by modelling the flow of *activities*. Activity *partitions* (open green rectangle) specify the names of *classes* involved in the *activities* and are used to logically organize an Activity diagram. Moreover, for complex *activities* the symbol to indicate a *sub-activity* (∞) is used and the Enterprise Architect 4.0 software makes possible to see the relative "workflow" with a double click of the mouse on the icon of the *sub-activity*.

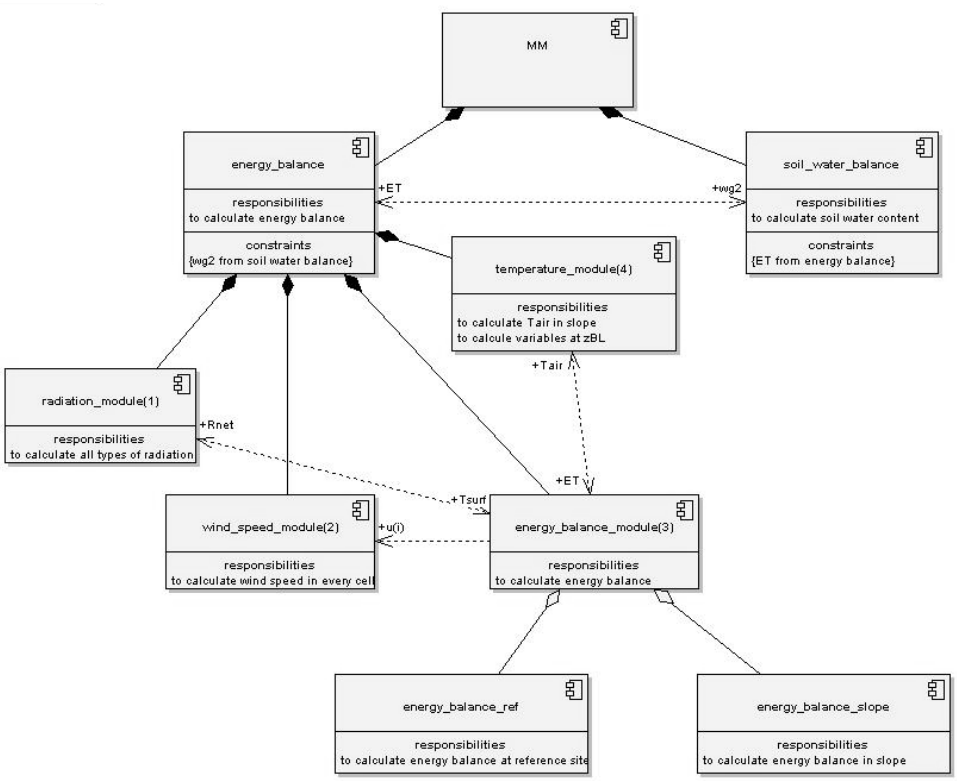

Fig. 5 - The Component diagram for the general view of the Micrometeorological Model.
Fig. 5 - *Il diagramma dei Componenti del Modello Micrometeorologico*

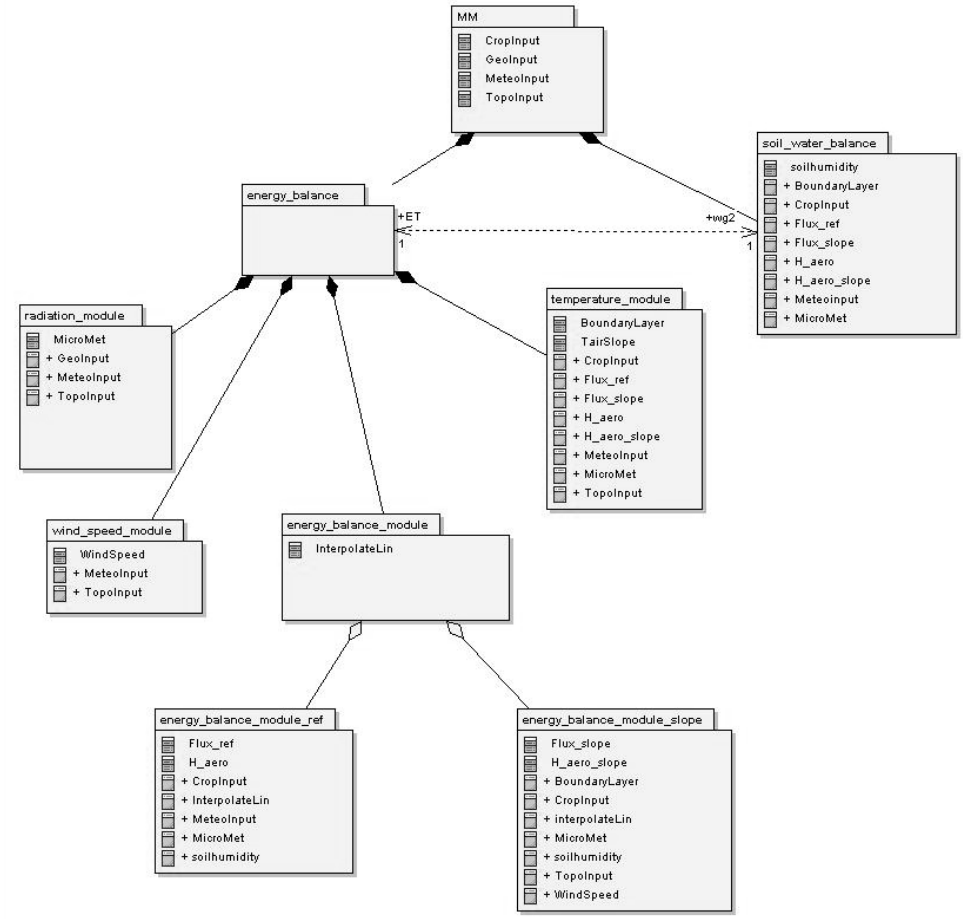

Fig. 6 - The sub-division in packages of the Micrometeorological Model.
Fig. 6 - *La suddivisione in pacchetti del Modello Micrometeorologico*

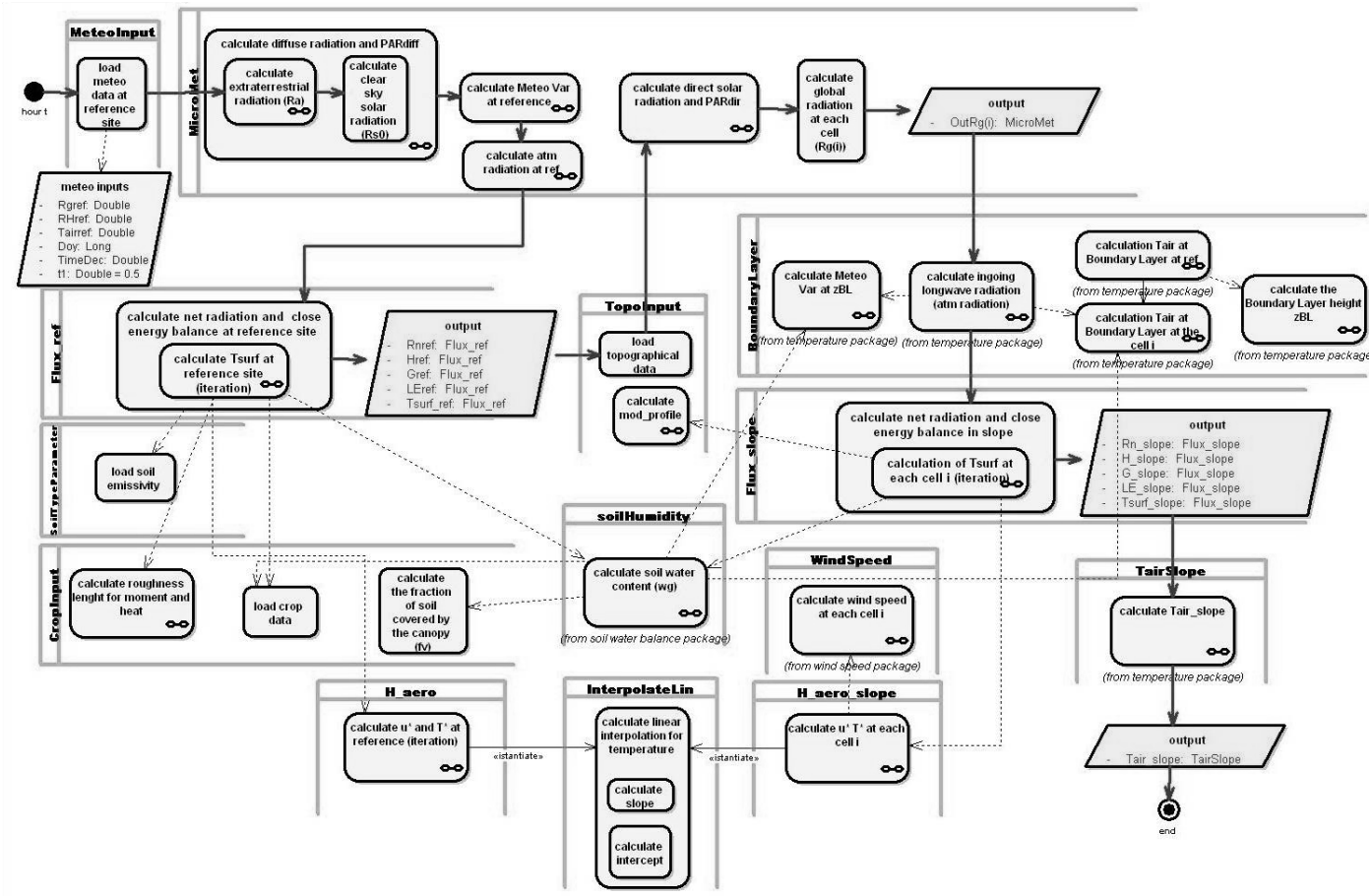

Fig. 7 - The general "workflow" of the Micrometeorological Model by means of the Activity diagram.
Fig. 7 - *Diagramma delle Attività relativo al "workflow" generale del Modello Micrometeorologico*

In Fig. 7 the path indicates that the MM at the generic step time, *hour t*, loads the meteorological data by means of the *class MeteoInput* and then it gives the control to the class *MicroMet* to calculate, at the reference site, the diffuse radiation, key meteorological variables and the atmospheric radiation. Then, the *Flux_ref class* tries to close the energy balance at the reference site using an iterative process which is explained in another Activity diagram. At this point topographical inputs are loaded and the flow continues with the computations relative to the generic cell. In particular, the control turns back to the *MicroMet class* to calculate the global radiation at the generic cell, and then to the *BoundaryLayer class* that is the link between the reference site and the generic cell of the catchment. Now, the model tries to close the energy balance at the generic cell by means of the *class Flux_slope*, giving the required outputs except the air temperature calculated by means of another *class*, *TairSlope*. The dependences among the *activities* are also indicated using lines with arrowheads that show the direction of the relationships. The arrows point to the *activity* from which information can be obtained: so, for example, to close the energy balance at the cell the calculations of the soil water content, of a "topographic stability function" (*mod_profile*) and of the scale factors u^* and T^* made by the *classes soilhumidity*, *TopoInput* and *H_aero_slope* respectively are needed.

This described template has been developed with high level of detail and all the produced UML diagrams relative to our micrometeorological model is being published on the Web page http://www.inea.it/isa/stamina/umlwebpage/index.htm, allowing the user to see directly the features of this information system.

Conclusions

Now, using the UML version of the MM, the structure and the functionality of our simulation model are well documented showing clearly what the model does and how it does it.

Moreover, the developed template could be used as a starting point to expand our model in a new larger applications. This eventually extension or modification in function of new aims, could be possible thanks the existence of a well structured documentation, produced by means of the standard UML. This is a methodology for presenting our MM in a easily comprehensible way, without using any specific formalism or programming language: in our case we have used Visual Basic 6.0, nevertheless, it could be possible to implement the model

in another specific programming language allowing programmers to decide the best environment according their needs and resources.

Acknowledgements

This work has entirely funded by the European project *STAMINA*. Moreover we thank Dr. Patrizia Trevisiol from University of Milan (Italy) for her technical and indispensable support during our work.

References

Booch, G., Rumbaugh, J., Jacobson, I., 1999. The Unified Modelling Language User Guide. Addison-Wesley, Longman, Reading, MA.

Cellier, P., Richard, G., Robin, P., 1996. Partition of sensible heat flux into bare soil and the atmosphere. Agric. For. Meteorol., 82, 245-265.

Courault, D., Lacarrère, P., Clastre, P., Lecharpentier, P., Jacob, F., Morloie, O., Prévot, L., Olioso, A., 2003. Estimation of surface fluxes in a small agricultural area using the three-dimensional atmospheric model Meso-NH and remote sensing data. Can. J. Rem. Sens., 29 (6), 741-754.

Donatelli, M., Carlini, L., Bellocchi, G., 2004. GSRAD, a software component to estimate Solar Radiation. Rivista italiana di Agrometeorologia. 1, 24-30.

Fu, P., Rich, P., 2002. A geometric solar radiation model with applications in agriculture and forestry. Comp. Elect. Agric., 30, 25-35.

Hobets, F., Noilhan, J., Golaz., C., Goutorbe, J.P., Lacarrère, P., Leblois, E., Martin E., Ohlè, C., Vidal-Madjer, D., 1999. The ISBA surface scheme in a macroscale hydrological model applied to the Hapex-Mobilhy area. Part I: Model and database. J. Hydrol., 217, 75-96.

Huntingford, C., Blyth, E. M., Wood, N., Hewer, F.E., Grant, A., 1998. The effect of orography on evaporation. Boundary-Layer Meteorol., 86, 487-504.

Jackson, P.S., Hunt, J.C.R., 1975. Turbulent wind flow over a low hill. Q. J. R. Meteorol. Soc, 101, 929-955.

Jacobson, I., Booch, G., Rumbaugh, J., 1998. The Unified Software Development Process. Addison-Wesley, Reading, MA.

Jacquemin, B., Noilhan, J., 1990. Sensitivity study and validation of a land surface parameterization using the HAPEX-MOBILHY data set. Boundary-Layer Meteorol., 52, 93-134.

Jarvis, P.G., 1976. The interpretation of the variations in leaf water potential and stomatal conductance found in canopies in the field. Phil. Trans. Roy. Soc. London Series B, 273, 593-610.

Loookingbill, T.R., Urban, D.L., 2003. Spatial estimation of air temperature differences for landscape-scale studies in montane environments. Agr. For. Meteor., 114, 141-151.

Noilhan, J., Planton, S., 1989. A simple parameterization of land surface processes for meteorological models. Mon. Wea. Rev., 117, 536-549.

Papagjorji, P., Shatar, T.,2004. Using the Unified Modeling Language to develop soil water balance and irrigation-scheduling models. Env. Mod. Soft., 19, 451-459.

Papagjorji, P., Beck, H., Braga , J., 2004. An architecture for developing service-orinted and component-based environmental models. Ecol. Mod., 179, 61-76.

Personnic, P., 1999. Mesure et modélisation de la variabilité microclimatique induite par le relief dans le vignoble champenois: incidence sur la phénologie de la vigne. PhD thesis, INA-PG, 245 pp.

Silberstein, R. P., Sivalpolan, M., Wyllie, A., 1999. On the validation of a coupled water and energy balance model at small catchment scales. J. Hydrol., 220, 149-168.

Smith, M., 2000. The application of climatic data for planning and management of sustainable rainfed and irrigated crop production. Agric. For. Meteorol., 103 (1), 99-108.

Spitters, C.J.T., Toussaint, H.A.J.M., Goudriaan, J., 1986. Separating the diffuse and direct component of global radiation and its implications for modelling canopy photosynthesis. Part I, components of incoming radiation. Agric. For. Meteorol., 38, 217-229.

Szyperski, C., 2002. component Software (2nd ed.). Addison-Wesley, UK, 589 pp.

Taylor, P.A., Lee, R.J., 1984. Simple guidelines for estimating wind speed variations due to small topographic features. Climatol. Bull., 18(2), 3-32.

Troch, P.A., Perriconi, C., Mclonghin, D., 2003. Catchment scale hydrological modelling and data assimilation. Adv. Nat. Res., 26, 131-135.

Van Wesemael, B., Commerant, E., Mulligan, M., Burke, S., 2003. The impact of soil properties and topography on drought vulnerability of rainfed cropping system in southern Spain. Agr. Ecos. Env., 94, 1-15.

Vázquez, R.F., Feyen, J., 2003. Effect of potential evapotranspiration estimates on effective parameters and performance of the MIKE SHE-code applied to a medium-size catchment. J. Hydrol., 270, 309-327.

Wood, E.F. (Ed.), 1991. Land surface-atmosphere interactions for climate modelling. Observations. Models and analysis. Kluwer Academic press, Dordrecht reprinted from Surv. Geophys., 12, 1-3.