Advances in Geosciences

Using Python as a coupling platform for integrated catchment models

P. Kraft[1], S. Multsch[1], K. B. Vaché[1,*], H.-G. Frede[1], and L. Breuer[1]

[1]Institute for Landscape Ecology and Ressources Management, Gießen, Germany
[*]now at: Biological and Ecological Engineering, Oregon State University, Corvallis, USA

Abstract. Interdisciplinary sharing of knowledge is a key for understanding matter fluxes in landscapes. However, models of transport and reactive fluxes from different disciplines need to work seamlessly together, to capture the tight feedback loops between different compartments and process domains of a landscape. Techniques to facilitate the integration of model codes for integrated catchment modelling exist, but are still scarcely used. In this paper, we are testing a scripting language, Python as a model coupling platform, and demonstrates effects of feedback loops on a virtual agriculturally used hillslope.

1 Introduction

The movement and storage of water in the environment is defined by a series of complex relationships involving atmosphere, biosphere, pedosphere, and hydrosphere. These relationships vary in time and space, and while capturing them within a simulation requires simplification, the process of simplifying the relationships may lead to useful predictions and insights into system function. Solute-based models, which add an additional layer of complexity, are often based upon theory developed in various disciplines. In a watershed model, including nutrient dynamics, the theories from a wide range of disciplines are needed for the model formulation. Disciplines involved range at least from soil and hydrologic sciences, to biogeochemistry and agronomy, not to mention physics and mathematics.

We argue that the degree of success any model has in capturing the key features of such a wide variety of fields, depends on buy-in from disciplinary experts, and further, that this vetting process is facilitated through transparency associated with the model development strategy.

In this paper, we outline the strategies that have been developed to produce such integrated models of environmental process. These include soft coupling (e.g. Cui et al., 2005), redevelopment (e.g. Band et al., 2001), as well as the use of an explicit coupling platform (e.g. Gregersen et al., 2007). The paper outlines some advantages and disadvantages of each of these strategies, and suggests a fourth, less explored but potentially fruitful alternative. Specifically, this alternative involves the development of a high-level, object oriented programming language, accessible through standard scripting tools, and targeted to the development of integrated process models. Ousterhout promoted this use of a scripting language as "glue" between models written in compiled, high performance languages over a decade ago (Ousterhout, 1998). However, the potential of scripting languages to design coupled, yet independent model suites is up to now scarcely used. We suggest that a targeted language possesses a number of significant advantages that have yet to be explored adequately. The system outlined here is proposed as an initial step in the development of an open source standard based code, which focuses on accessibility and portability, as called for by Buytaert et al. (2008).

The processes to be modelled in an integrated matter flux catchment model fall in one of two categories: transport processes and local turn over processes. Transport processes, usually by water, air or management, need to be modelled with spatially explicit models. Turn over models, like plant growth models, biogeochemical process models or local energy budget models are rather plot models without a definite spatial domain. In an integrated catchment approach, a coupling needs to be established between one instance of each type of transport model and many instances of the local models.

This paper focuses on the description of such a strategy, designed to facilitate the integrated simulation of watershed scale hydrology and solute transport.

Correspondence to: P. Kraft
(philipp.kraft@umwelt.uni-giessen.de)

2 Method

To illustrate the simplicity of coupling independent models using Python as a "glue" language, three different models were coupled and applied on a virtual hillslope. A water and solute transport model, a plant growth model and a model of organic matter decomposition.

Turnover of dissolved inorganic nitrogen (DIN) and water into biomass is calculated by the plant growth model, while the decomposition model calculates the turnover from dead biomass to the components DIN dissolved organic carbon (DOC) and gaseous carbon losses (CO_2). Relocation of the two dissolved compounds DOC and DIN is governed by the water and solute transport model. The models used in this exercise were not chosen to fit a specific theory, but for accessibility and simplicity.

2.1 Transport model

The hydrological model framework CMF (Catchment Model Framework) by Kraft et al. (2008, 2010), which is based on the rejectionist framework by Vaché and McDonnell (2006), is an extension to the Python programming language designed to desgin water transport models. A model in CMF is set up as a network of storages and boundary conditions, connected by flux calculating submodels. CMF allows for the development of detailed mechanistic models as well as lumped large scale linear storage based models. In this study, a two dimensional Richards-based hill slope model was setup, whereas in an ungauged artificial catchment study (Holländer et al., 2009) a 2.5 dimensional Green-Ampt/Darcy approach was chosen.

The framework was designed to be connectible with other models through the implementation of a clear application programming interface (API). To demonstrate this functionality, a virtual hillslope using a variable saturated, continuous model was set up, using a discretized form of the two dimensional Richards equation. The hillslope is divided laterally into cells and each cell is divided vertically into layers. The flux between the layers of one cell (percolation and capillary rise) and between the layers of adjacent cells (lateral flow) is calculated using the wetness-based form of the Richards equation, spatially discretized using a finite volume approach. Surface flow is routed to the bottom of the hillslope using a kinematic wave approach. At the bottom of the slope, a constant head boundary condition of 25 cm below ground is imposed to simulate a downslope ditch with constant head. A soil depth of 3 m with impervious bedrock is assumed, whereas evapotranspiration is simulated by the plant growth model, through the API. However, for applications not involving a plant growth model, methods of calculating evapotranspiration are implemented in CMF. Solute transport is modelled using a simple advective transport scheme.

2.2 Turnover model 1: plant growth

Plant growth is determined with the Plant growth Modeling Framework (PMF) (Multsch, 2010). The model divides the plant into its physical components root, shoot, leaf, stem and storage organs. In the physical structure the growth processes are calculated on an abstract level. These components are related to process modules, which hold numerical solutions for the growth processes. The model can be adjusted to agriculture crops without changing the fundamental structure. Two interfaces handle the data transfer between PMF and other models or databases.
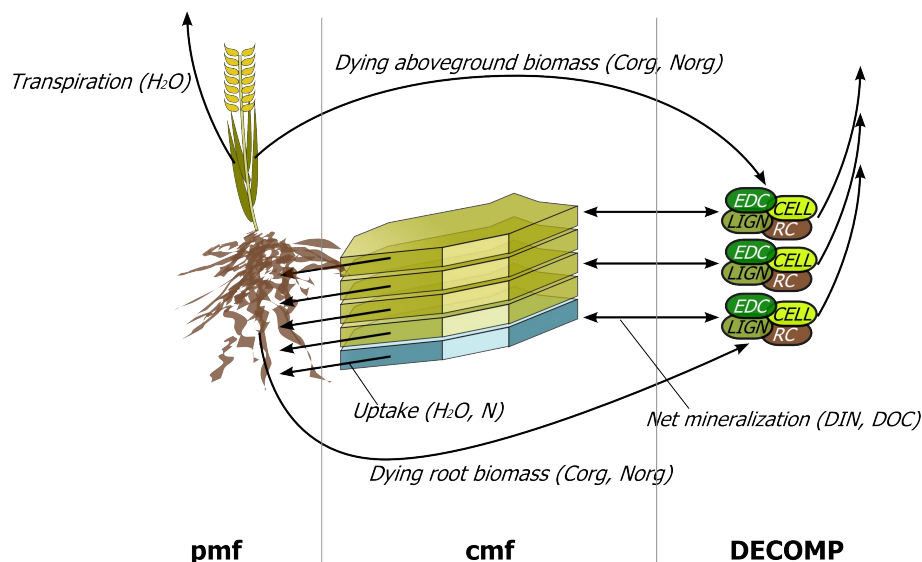
In this study, PMF is parameterized to represent summer wheat. Daily biomass accumulation is calculated with the radiation use efficiency and solar radiation (Acevedo et al., 2002). The biomass is allocated at the plant organs in relation to the development phase, which is determined using the thermal time concept (Monteith and Moss, 1977; Miller et al., 2001). Drought and nitrogen stress limit growth, but the plant can adapt to these stresses by varying the root biomass distribution. Stress is defined in PMF as $1-\alpha$, where α is the ratio between actual uptake of water or nutrients and potential uptake.

Root water uptake is calculated from potential transpiration and a crop specific response function relating uptake and soil matrix potential. The water uptake is represented as sink term in the water flux equation. This concept is similar to the macroscopic water uptake approach type II (Hopmans and Bristow, 2002; Feddes et al., 2001). Nitrogen uptake is divided into an active and a passive component following Simunek and Hopmans (2009). Passive uptake is the product from the dissolved nitrogen concentration and water uptake. Active uptake is determined from the residual nitrogen demand after passive uptake assuming Michaelis-Menten type kinetics (Simunek and Hopmans, 2009).

2.3 Turnover model 2: decomposition of organic matter

The DECOMP model (Wallman et al., 2006) is a semi-deterministic model of decomposition of organic matter. It was developed as a part of the integrated plot scale forest biogeochemical model ForSAFE. It includes four carbon pools representing decomposable components, cellulose-like material, lignin-like material and recalcitrant material. Each pool has a potential transformation rate, and parameters to describe the reaction of the decomposition rate to environmental conditions, like wetness, soil temperature and soil acidity. The products of decomposition are distributed between dissolved organic carbon (DOC) and carbon dioxide (CO_2).

For simplicity, only one Nitrogen pool exists for the four carbon pools. The gross mineralisation rate of Nitrogen is calculated from the mass of carbon decomposed to DOC and CO_2. The mineralised N is partly released to the soil solution and partly immobilized, depending on the N content in the

Fig. 1. Matter fluxes in one cell of the integrated model. Matter fluxes across model domain boundaries are addressed integrated setup script, while matter fluxes between cells are calculated by the transport model CMF. EDC (easily decomposing compounds), CELL (cellulose like compounds), LIGN (lignin like compounds) and RC (recalcitrant compounds) denote the four organic matter compounds in DECOMP.

organic matter. To extract the model from the integrated For-SAFE code, a minimal model version was reimplemented using C++ as a Python extension, similar to CMF. The slightly different curve shape to calculate the immobilization/mineralization ratio does not affect the usability of the model to be used as a demonstration code example in this model coupling exercise. The parameterisation of the model was taken from Wallman et al. (2005). The parameters were determined to represent the behaviour of forest soils. Agricultural soils may behave differently, but as stated earlier, the focus of this paper is rather the feasibility of integrated models and the coupling approach, than on the parameterization of the models.

2.4 Integration

The three models were coupled with each other using a common setup script in Python. Since CMF and DECOMP are Python extensions, written in C++ and PMF is entirely written in Python, the integration is straightforward. Existing legacy model codes can be wrapped as a Python extension using tools like SWIG (Simple Wrapper Interface Generator) (Beazley and Lomdahl, 1996) for model codes written in C or C++ or F2PY (Peterson, 2009) for model codes written in FORTRAN. The level required to wrap an existing model code depends primarily on the modularity of the existing code.
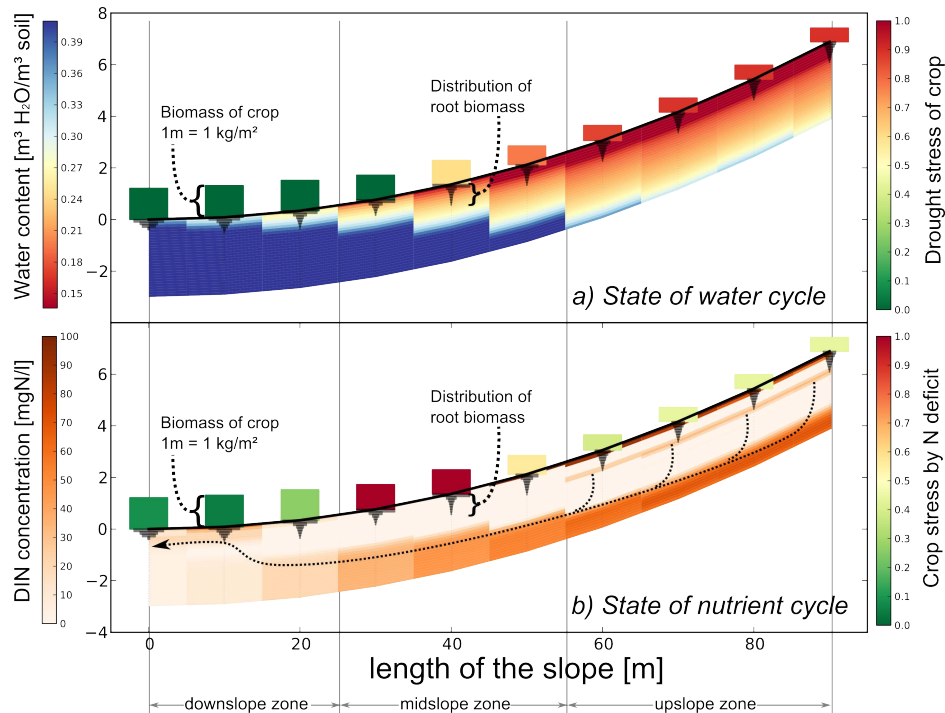
For our study, CMF was set up as a fully connected two dimensional hillslope model. For each lateral unit (cell in CMF) an instance of the plant growth model was created, and for each vertical unit (layer) in each cell an instance of DECOMP setup.

Figure 1 depicts the communication between the models. To keep the figure clear, only matter fluxes between model domains are shown.

Two different interfacing strategies has been chosen. Exchange of data between DECOMP and CMF is implemented using a method to be called by the main time loop, copying explicitly the data between the models. This simple strategy of data exchange between independent models is facilitated using a scripting language as interface, since necessary conversion or interpolation of data can be accomplished using the built-in features or existing mathematical libraries of the scripting language. To illustrate the implementation of this strategy, the data exchange between CMF and DECOMP at the layer scale is shown in the Appendix A.

PMF, on the other hand, is specifically designed for different types of input of boundary conditions, namely nitrate concentration and soil moisture. PMF expects at setup time an interface providing the required data. These interfaces can route the data queries to a database with measured data, or to a model providing the requested data. A class wrapping a single cell of the CMF based transport model can be constructed, implementing the interface of PMF. Using this strategy, no direct copying of data between the models is required. The synchronization of water and solute fluxes between PMF, CMF and DECOMP, as shown in Fig. 1 is rather implemented using the first strategy.

Input data, like meteorological time series, are imported from the setup script, using the advanced syntax of Python for text analyzes and partly hard coded into the setup script. The main time loop of the integrated model is part of the common setup script. The turnover models in this application run with a daily time step. However, the transport model

Fig. 2. State of crops and soil water modelled by the integrated PMF-CMF-DECOMP system on 28 June 1992 after 12.5 years of integrated model run time. Each model cell has a size of $10\,\mathrm{m} \times 10\,\mathrm{m}$. Coloured rectangular bars indicate the crop state. The upper figure shows the spatially distribution of soil water content θ (ranging from 0.15 to $0.42\,\mathrm{m}^3/\mathrm{m}^3$, red to blue), and crop stress (colour of rectangles), ranging from 0 (no stress, green) to 1 (no growth due to draught, dark red). The soil cross-section in the lower figure is coloured according to the calculated DIN concentration in soil solution, ranging from 0 (white) to 100 (brown) mgN/l. The dashed arrow shows the long term transport path for excessive DIN from the upslope zone to the ditch.

uses a variable time step, which might be in the range of seconds during rainfall events. Therefore, the setup runs each of the models for one day, taking one model time step for the turnover models while CMF is taking internally many time steps.

3   Results

To demonstrate the feedback loops, the model suite is run for 12 years using climatic data from the German meteorological service station "Giessen". Apart from the climatic data, the application of fertilizer is considered as an external driver. The slope is fertilized three times a year. During sowing in early spring (1 March), 20 kg N/ha manure is applied, and shortly before germination (mid April) mineral fertilizer with 80 kg N/ha is given. A third application of 80 kg N/ha mineral fertilizer is carried out during the shooting phase of the crops (end of May). The soil properties are assumed to be constant throughout the hillslope profile, and resemble a sandy soil.

Despite the simple model setup, the data produced leaves wide space for different interpretations. Due to the limitations of the chosen approach, as discussed below, it is not the objective of this study to show realistic model behaviour, but

rather the potential how lateral transport influences model results. A single model time step, 28 June 1992 is chosen out of the simulated time series to explain possible effects of lateral nutrient transport on simulated crop growth in different parts of a virtual hill slope. At this time step, the shooting phase of the summer wheat is finished and nutrient storages in the rooting zone depleted. The date is located at the end of a four week period with less than 20 mm rainfall.

Figure 2 shows the state of the hillslope near to the end of the simulation. The upper figure shows the distribution of soil moisture and plant water stress, while the lower figure displays the distribution of DIN and resulting plant stress. Bar length in both figures indicates the produced biomass in $\mathrm{kg/m}^2$ and bar colours indicate the stress state of the plant: a green bar shows a plant where the growth is not hampered by lack of water (upper figure) or nitrogen in the soil solution, while dark red shows drastic reduced growth.

DIN is added to the transport system by application of fluid fertilizer and by mineralization of plant residuals (DE-COMP) in the first layer and removed by plant uptake in the layers containing fine roots (PMF). The rooting depth changes over the vegetation period (PMF) from germination (uptake from first layer only) to a maximum of ca. 1 m (uptake from the upper 10 layers).

As shown in Fig. 2, the production rate in the upslope zone of the hillslope (50–90 m from ditch) is limited due to drought. As a result, the fertilizer applied to this area is only partly taken up by the growing crops, while excessive DIN is leached from the rooting zone. The excessive DIN is then transported at the soil bedrock interface by lateral flows towards the saturated zone in the midslope zone of the hillslope (30–50 m from ditch). Here, the higher DIN concentrations in the saturated zone are not accessible for the plant roots. Therefore plant growth is limited by DIN deficit, despite the high concentrations below the rooting zone. In the downslope zone (0–30 m from ditch), the saturated zone is accessible for the plants and the excessive DIN from the upslope zone can be taken up. The dashed arrow shows the long term transport of DIN through the hillslope. Since water from the saturated zone is ascending by capillary rise and accessible for the crop roots in the downslope zone of the hillslope, excessive DIN transported from the upper region becomes available to the crops of the lower region.

4 Discussion

The exemplary model setup shown above is not designed to be competitive for "real" problems. For operational use, the parameterization of DECOMP has to be validated for agricultural soils and a spin-up time of at least 100 years should be used. However, the goal of the application presented here is to show the effect of connecting typical plot scale models like plant growth models or biogeochemical soil models with a transport scheme. For "real" applications, sensitivity and uncertainty measures need to be taken into account.

In the exemplary setup shown above, the nutritional demand of a crop under water limited conditions affects the nutritional supply of crops in the riparian zone. This kind of spatial relationship cannot be captured with a classical one dimensional plant growth and nutrient turnover model, including a percolation model. Examples for this approach are PnET (Aber et al., 1997), EPIC (Williams et al., 1984), CENTURY (Parton et al., 1983) and WOFOST (Diepen et al., 2007). On the other hand, two dimensional Richards equation based models, such as CATFLOW (Maurer, 1997; Zehe et al., 2001) or HYDRUS 2D (Simunek et al., 1999) do neither include submodels to calculate nutritional uptake by plants, nor models for decomposition of the crop residuals after harvest. Although the combination of an existing crop growth model with an existing transport model in a single code base is feasible, the resulting integrated model might suffer a specific functionality for the next arisen use case. The most prominent example of a statically coupled distributed nutrient flux model is RHESSys (Band et al., 2001) and influenced the framework approach presented. However, since the models integrated into RHESSys were completely redeveloped, using alternative models is a time consuming task.

The second option for a tight coupling, the use of standardized model coupling interfaces (e.g. Gregersen et al., 2007; Lagarde et al., 2001) avoids this problem. Model codes to be used in this kind of coupling environment need to implement the interfaces of the coupling platforms, and can be coupled with each other. However, building an interface for a coupling platform is not trivial and therefore better suited for model codes unlikely to be changed. With a scripting language as coupling infrastructure, as promoted by Ousterhout (1998) and implemented in this study, a lightweight, flexible and less formalized interface can be used for exchanging data between models. Converting an existing model, like one of the mentioned well known plant growth models into a module of a scripting language is relatively simple, given automation tools such as SWIG (Beazley and Lomdahl, 1996) for model codes in C or C++ or F2PY (Peterson, 2009) for model codes in FORTRAN. In difference to coupling platforms, the user of a "pure" submodel integrated into a scripting language yields benefits, such as simplified testing or model run batching. However, the effort to implement an existing model as an extension to the Python language depends mainly on the quality and modularity of the code. Well structured and documented model codes might be wrapped in a few days. Other codes might not be suitable for integration into the Python language, due to their unstructured design or by inaccessibility of the source code. The main challenge for wrapping is the often limited possibility to disable specific process descriptions in models, covered by another coupled model already. For example most plant growth models include a more or less sophisticated model of percolation, which has to be disabled when coupled with a transport model like CMF.

However, since benefits exist in having a model as a Python extension, even for stand-alone applications and development, the authors wish to encourage model developers to create Python wrappers for their own models, and release the model codes to the public using an open source license. A multitude of models with a built-in facility of coupling even during model development might establish a fast knowledge exchange path between different disciplines.

Appendix A

Data exchange between CMF and DECOMP

```
def RunDECOMP(dt, cmf_layer, DECOMP_SOM):
    # parameter exchange DECOMP queries
    # the wetness and DOC concentration from cmf
    DECOMP_SOM.wetness = cmf_layer.wetness
    DECOMP_SOM[DOC] = cmf_layer.conc(DOC)
    # run DECOMP for timestep and
    # return N and DOC release rate
    Nmin, newDOC, newCO2 = DECOMP_SOM.run(dt)
    # Convert kg/(ha,day) to g/day
    Nmin_g_day = Nmin * cmf_layer.cell.Area/1e4
    newDOC_g_day = newDOC * cmf_layer.cell.Area/1e4
    # Add N and DOC release as tracer source in cmf
    cmf_layer.Solute(N).source += Nmin_g_day
    cmf_layer.Solute(N).source += newDOC_g_day
```

Edited by: A. Wahren, F. Tavares Wahren, and K.-H. Feger
Reviewed by: M. Wegehenkel and T. aus der Beek

References

Aber, J. D., Ollinger, S. V., and Driscoll, C. T.: Modeling nitrogen saturation in forest ecosystems in response to land use and atmospheric deposition, Ecol. Model., 101, 61–78, 1997.

Acevedo, E., Silva, P., and Silva, H.: Wheat growth and physiology. Bread wheat: improvement and production, Food and Agriculture Organization (FAO), Rome, 39–70, 2002.

Band, L. E., Tague, C. L., Brun, S. E., Tenenbaum, D. E., and Fernandes, R. A.: Modeling watersheds as spatial object hierarchies: structure and dynamics, Trans. GIS, 4, 181–196, 2001.

Beazley, D. M. and Lomdahl, P. S.: Lightweight computational steering of very large scale molecular dynamics simulations, in: Proceedings of the 1996 ACM/IEEE conference on Supercomputing, IEEE Computer Society Washington, DC, USA, 96, 50–61, 1996.

Buytaert, W., Reusser, D., Krause, S., and Renaud, J. P.: Why can't we do better than Topmodel?, Hydrol. Process., 22, 4175–4179, 2008.

Cui, J. B., Li, C. S., Sun, G., and Trettin, C.: Linkage of MIKE SHE to Wetland-DNDC for carbon budgeting and anaerobic biogeochemistry simulation, Biogeochemistry, 72, 147–167, 2005.

Diepen, C. A., Wolf, J., Keulen, H., and Rappoldt, C.: WOFOST: a simulation model of crop production, Soil Use Manage., 5, 16–24, 2007.

Feddes, R. A., Hoff, H., Bruen, M., Dawson, T., de Rosnay, P., Dirmeyer, P., Jackson, R. B., Kabat, P., Kleidon, A., and Lilly, A.: Modeling root water uptake in hydrological and climate models, B. Am. Meteorol. Soc., 82, 2797–2809, 2001.

Gregersen, J. B., Gijsbers, P. J. A., and Westen, S. J. P.: OpenMI : Open Modelling Interface, J. Hydroinform., 9, 175–191, 2007.

Holländer, H. M., Blume, T., Bormann, H., Buytaert, W., Chirico, G. B., Exbrayat, J.-F., Gustafsson, D., Hölzel, H., Kraft, P., Stamm, C., Stoll, S., Blöschl, G., and Flühler, H.: Comparative predictions of discharge from an artificial catchment (Chicken Creek) using sparse data, Hydrol. Earth Syst. Sci., 13, 2069–2094, doi:10.5194/hess-13-2069-2009, 2009.

Hopmans, J. W. and Bristow, K. L.: Current capabilities and future needs of root water and nutrient uptake modeling, Adv. Agron., 77, 103–183, 2002.

Kraft, P., Vaché, K. B., Breuer, L., and Frede, H.-G.: A solute and water flux library for catchment models, in: iEMSs 2008: International Congress on Environmental Modelling and Software Integrating Sciences and Information Technology for Environmental Assessment and Decision Making 4th Biennial Meeting of iEMSs, edited by: Sànchez-Marrè, M., Béjar, J., Comas, J., Rizzoli, A., and Guariso, G., International Environmental Modelling and Software Society (iEMSs), 2, 2053–2056, available at: http://www.iemss.org/iemss2008/index.php?n=Main.Proceedings (last access: 16 August 2010), 2008

Kraft, P., Vaché, K. B., Frede, H.-G., and Breuer, L.: A hydrological programming language extension for integrated catchment models, Environ. Modell. Softw., submitted, ENVSOFT-S-10-00017-1, 2010.

Lagarde, T., Piacentini, A., and Thual, O.: A new representation of data-assimilation methods: The PALM flow-charting approach, Q. J. Roy. Meteor. Soc., 127, 189–207, 2001.

Maurer, T.: Physikalisch begründete, zeitkontinuierliche Modellierung des Wassertransports in kleinen ländlichen Einzugsgebieten, Ph.D. thesis, Universität Karlsruhe, Germany, 1997.

Miller, P., Lanier, W., and Brandt, S.: Using Growing Degree Days to Predict Plant Stages, Montguide, 2001.

Monteith, J. L. and Moss, C. J.: Climate and the Efficiency of Crop Production in Britain, Philos. T. Roy. Soc. B, 281, 277–294, 1977.

Multsch, S.: Plant growth Modeling Framework (PMF), Entwicklung eines dynamischen Pflanzenwachstums – Modells, Msc. thesis, Justus-Liebig-Universität Giessen, 2010.

Ousterhout, J. K.: Scripting: Higher-level programming for the 21st century, IEEE computer, 31, 23–30, 1998.

Parton, W. J., Anderson, D. W., Cole, C. V., and Stewart, J. W. B.: Simulation of soil organic matter formation and mineralization in semiarid agroecosystems, The Univ. of Georgia, College of Agriculture Experiment Stations, Athens, Georgia, 1983.

Peterson, P.: F2PY: a tool for connecting Fortran and Python programs, International Journal of Computational Science and Engineering (IJCSE), 4, 296–305, 2009.

Simunek, J. and Hopmans, J.: Modeling compensated root water and nutrient uptake, Ecol. Modell., 220, 505–521, 2009.

Simunek, J., Sejna, M., and van Genuchten, M. T.: The HYDRUS-2D software package for simulating two-dimensional movement of water, heat, and multiple solutes in variably saturated media, Version 2.0, IGWMC – TPS – 53, International Ground Water Modeling Center, Colorado School of Mines, Golden, Colorado, 1999.

Vache, K. B. and McDonnell, J. J.: A process-based rejectionist framework for evaluating catchment runoff model structure, Water Resour. Res., 42, W02409, doi:10.1029/2005WR004247, 2006.

Wallman, P., Belyazid, S., Svensson, M. G. E., and Sverdrup, H.: DECOMP – a semi-mechanistic model of litter decomposition, Environ. Modell. Softw., 21, 33–44, 2006.

Wallman, P., Svensson, M. G. E., Sverdrup, H., and Belyazid, S.: FORSAFE – an integrated process-oriented forest model for long-term sustainability assessments, Forest Ecol. Manag., 207, 19–36, 2005.

Williams, J. R., Jones, C. A., and Dyke, P. T.: A modeling approach to determining the relationship between erosion and soil productivity, T. Am. Soc. Agr. Eng., 27, 129–144, 1984.

Zehe, E., Maurer, T., Ihringer, J., and Plate, E.: Modeling water flow and mass transport in a loess catchment, Phys. Chem. Earth Pt. B, 26, 487–507, 2001.