

A hydrological programming language extension for integrated catchment models

Philipp Kraft¹, Kellie B. Vaché^{1,2}, Hans-Georg Frede¹, Lutz Breuer¹

NOTICE: this is the author's version of a work that was accepted for publication in Environmental Modelling and Software. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in EnvSoft, [VOL#, ISSUE#, (DATE)] doi:10.1016/j.envsoft.2010.12.009

ABSTRACT

Hydrological models are created for a wide range of scales and intents. The Catchment Modelling Framework (CMF) extends the Python programming language with hydrology specific language elements, to setup specific hydrological models adapted to the scientific problems and the dominant flow processes of a particular study area. CMF provides a straightforward method to test hydrological theories and serve as a transport module in integrated, interdisciplinary catchment model approaches.

Keywords: Hydrological model, solute transport, model coupling, object orientated programming, catchment models

SOFTWARE AVAILABILITY

Software name	Catchment Modelling Framework – CMF
Software homepage	http://www.uni-giessen.de/cms/faculties/f09/institutes/ilr/ilr-frede/download
Developers	Philipp Kraft, Kellie B. Vaché
Contact	Philipp.kraft@umwelt.uni-giessen.de
Year first available	2010
Availability	free download, GNU public license, v2
Software required	Python 2.6 with NumPy (>1.3) and Shapely (>1.2.4) on Linux and Windows
Hardware required	Multi-core CPU
Programming language	C++, Python
Program size	2.2 MB source code, 4MB documentation

¹ Institute for Landscape Ecology and Resources Management (ILR), Justus-Liebig-Universität Gießen, Germany

² now at: Biological and Ecological Engineering, Oregon State University, USA

INTRODUCTION

In the last 40 years, a multitude of more or less physically-based, deterministic hydrological models have been developed for various scopes, scales and regions. Despite the existence of broadly accepted governing equations in hydrology, including the Richards equation describing flow in porous media and the St. Venant equations defining open channel flows, modelling landscape-scale water movement remains hampered by heterogeneity, parameter uncertainty and lacking computational power.

In response to this challenge, Buytaert et al. (2008) suggest that model codes should be fully accessible, modular and portable. The incorporation of these qualities into the code is a mechanism to allow the model developer to aid model users in testing different hypotheses about flow regimes, and, ultimately, in applying the most appropriate model structure to different questions of interest. To these essential characteristics we add the concept of connectivity, i.e. that model codes also should focus on between-model connections, particularly in the case of water quality models, to allow integration into a multi-objective landscape model approach.

SOFTWARE DESIGN CONCEPT

Our approach presented here, originally based on the rejectionist framework approach by Vaché and McDonnell (2006), represents a model abstraction concept similar to the finite volume method (FVM) approach by Qu and Duffy (2007). The Catchment Modelling Framework (CMF) generalizes the FVM discretization scheme to let the user attach the finite volumes (water storages in CMF) with a variety of flow accounting equations (flux connections in CMF). The compounds of the model are assembled using the well known scripting language Python. The advantages of scripting computer languages for scientific modelling are described by Ousterhout (1998) and Karssenberget al. (2007), namely rapid application development using very high level instructions and a dynamic typing system. To benefit from the simplicity of scripting languages on the one hand, and from the reduced computation effort of compiled languages, the core components of the framework are written in C++ as an extension for Python, using the “Simplified Wrapper Interface Generator” (SWIG) by Beazley and Lomdahl (1996).

A model built on CMF is composed of a network, a spatial context and a solver (Figure 1). The nodes of the network correspond to the water storages and boundary conditions, and the network edges to the governing equations, like Richard’s or Manning’s equation, amongst others. The differential equation representing the water volume (V) in one storage-object i is then derived from the network, and is defined as:

$$\frac{dV_i}{dt} = \sum_{j=1}^{N_i} -q_{i,j}(V_i, V_j, t) \quad (1)$$

The flux function q is calculated by the flux connection objects, the edges of the network. N is the number of connected nodes of the actual node i and j denotes a node connected to i . Sources and sinks of water are realized by boundary objects. Solute storages are associated with each water-storage-object. The resulting system of ordinary differential equations for water volume and matter content is in most cases stiff and only integratable with an implicit solver. The CVODE solver (Hindmarsh et al., 2005) has been included in CMF for that purpose, along with other integration methods for the few cases of non-stiff setups. Any solver in CMF supports shared memory parallelism using OpenMP. Each node of the network is part of a spatial context, either the whole study area (the project), like reaches and big aquifers, or of a horizontal discretization unit, the cell.

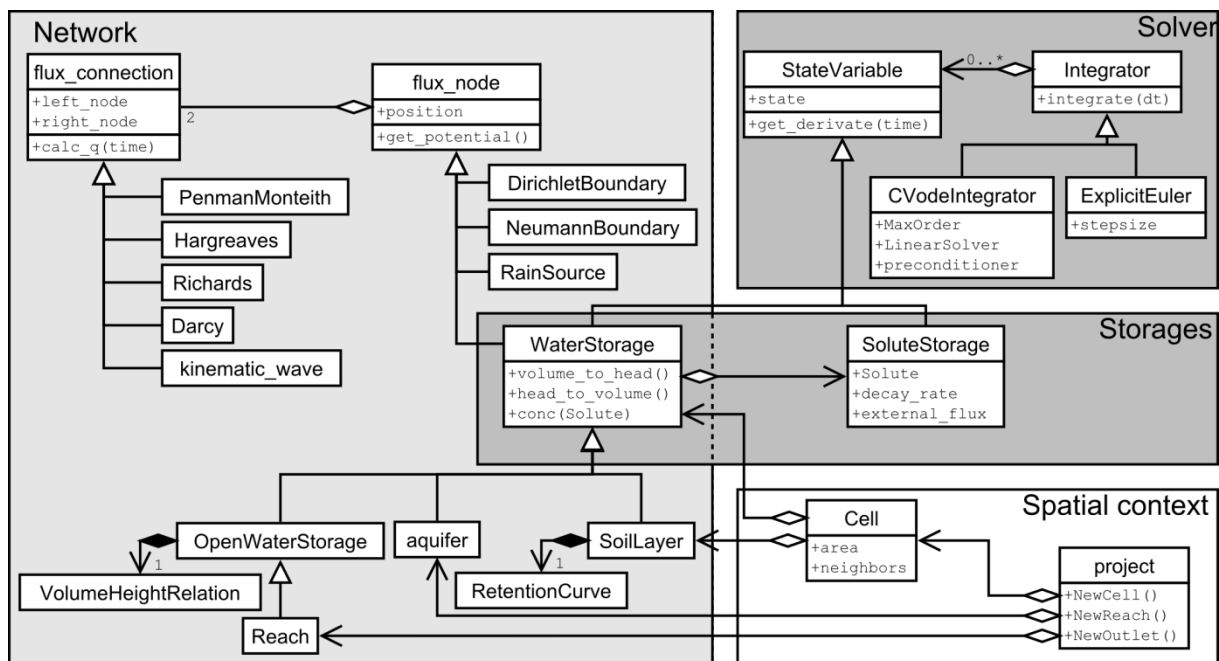


Figure 1: Simplified UML class diagram of CMF. The left section (light gray) includes the components for defining the network of water fluxes, the upper right part (dark gray) the classes to create and solve the resulting equation system, and the lower right (white) part shows classes for the spatial context

Using these components a wide range of hydrological catchment models can be build: Lumped conceptual models with physically based boundary conditions, semidistributed models as well as highly detailed physically based fully distributed models. Other modular framework approaches limit the range of models to be built to a specific model type. FUSE (Clark et al., 2008) and FLEX (Fenicia et al., 2006) for example are restricted to lumped and semidistributed approaches and the rejectionist framework by Vaché and McDonnell (2006) constrained to distributed topographic gradient driven approaches. The MIKE SHE model (Refsgaard and Storm 1995) covers a broader range of possible model setups. However due to the closed nature of the source code, the user cannot extend the system by his or herself.

Apart from the catchment scale, hillslope models, as well as three dimensional representations of fluxes in a lysimeters can be set up using CMF. A feasibility study concerning the coupling capabilities with biogeochemical models of the CMF approach has recently been published (Kraft et al., 2010).

REFERENCE LIST

- Beazley, D.M. Lomdahl, P.S., 1996. Lightweight computational steering of very large scale molecular dynamics simulations. 96, 50-61. 1996. Proceedings of the 1996 ACM/IEEE conference on Supercomputing.
- Buytaert, W., Reusser, D., Krause, S. Renaud, J.P. 2008. Why can't we do better than Topmodel? Hydrological Processes 22, 4175-4179.
- Clark, M.P., Slater, A.G., Rupp, D.E., Woods, R.A., Vrugt, J.A., Gupta, H.V., Wagener, T. Hay, L.E. 2008. Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose

differences between hydrological models. *Water Resources Research* W00B02, doi:10.1029/2007WR006735.

Fenicia, F., Savenije, H.H.G., Matgen, P., Pfister, L. 2006. Is the groundwater reservoir linear? Learning from data in hydrological modelling. *Hydrology and Earth System Sciences* 10, 139-151.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E. Woodward, C.S. 2005. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software* 31, 363-396.

Karssenbergh, D., de Jong, K. van der Kwast, J. 2007. Modelling landscape dynamics with Python. *International Journal of Geographical Information Science* 21, 483-495.

Kraft, P., Multsch, S., Vaché, K.B., Frede, H.-G. Breuer, L. 2010. Using Python as a coupling platform for integrated catchment models. *Advances in Geosciences* 27, 51-56.

Oosterhout, J.K. 1998. Scripting: Higher-level programming for the 21st century. *IEEE Computer* 31, 23-30.

Qu, Y.Z. Duffy, C.J. 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. *Water Resources Research* 43, W08419, doi:10.1029/2006WR005752.

Refsgaard, J.C. & Storm, B. 1995. MIKE SHE. In: *Computer Models of Watershed Hydrology* (ed V.P. Singh), pp. 809-846. Water Resources Publications, Highlands Ranch, Colorado.

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