



Towards multi-scale integrated hydrological models using the LIQUID[®] framework. Overview of the concepts and first application examples.

Branger, F.^{(1)*}, Braud, I.⁽¹⁾, Debionne, S.⁽²⁾, Viallet, P.⁽²⁾, Dehotin, J.⁽¹⁾, Henine, H.⁽³⁾, Nedelec, Y.⁽³⁾, Anquetin, S.⁽⁴⁾

(1) Cemagref, UR HHLY, 3bis quai Chauveau CP 220, 69336 Lyon Cedex 09, France

(2) HYDROWIDE, 1025 Rue de la Piscine, 38600 Saint-Martin d'Hères, France

(3) Cemagref, UR HBAN, Parc de Tourvoie BP 44, 92163 Antony Cedex, France

(4) LTHE, Université de Grenoble (CNRS, INPG, IRD, UJF), BP 53, 38400 Grenoble Cedex 9, France

Abstract

Distributed hydrological models are valuable tools that can be used to support water management in catchments. However, the complexity of management issues, the variety of modelling objectives, and the variable availability of data require a flexible way to customize models and adapt them to each individual problem. Environmental modelling frameworks offer such flexibility; they are designed to build and run integrated models on the basis of reusable and exchangeable components. This paper presents the LIQUID[®] framework, developed by Hydrowide since 2005. The purpose of developing LIQUID[®] was to provide both easier integration of hydrological processes and preservation of their characteristic temporal and spatial scales. It suits a wide range of applications, both in terms of spatial scales and of process conceptualisations. LIQUID[®] is able to synchronize different time steps, to handle irregular geometries, and to simulate complex connections between components, in particular involving feedback. The paper presents the concepts of LIQUID[®] and the technical choices made to meet the above requirements, with focuses on the simulation run system and on the spatial discretization of process components. The use of the framework is illustrated by five application cases associated with contrasted spatial and temporal scales.

Keywords

Hydrological models, integrated modelling, framework, LIQUID[®], modularity, spatial discretization, feedback simulation

Software availability

Name of software: LIQUID[®] platform

Contact address: www.hydrowide.com/liquid (Hydrowide – 1025 Rue de la piscine – 38400 Saint Martin d'Hères, France, +33476825284)

Program language/size: C++ / 2Gb

Hardware/Software required: Working station with ANSI compliant C++ compiler; PostgreSQL /PostGIS database

Availability and cost : LIQUID[®] is a proprietary software of Hydrowide. It is available for research through a partnership contract that specifies the rules for intellectual property and

* Corresponding author: Flora Branger, Cemagref, UR Hydrologie-Hydraulique, 3bis Quai Chauveau CP 220, 69336 Lyon Cedex 9, France, Tel: +33 4 72 20 87 87, Fax: +33 4 78 47 78 75, E-mail: flora.branger@cemagref.fr

technical support. The licenses for process modules are set by their owners without restriction on the license type (free software or proprietary/commercial software).

1 Introduction

Hydrological simulation models are valuable tools for addressing challenging issues related to catchment management, such as impact of climate change on water resources, influence of urbanization on floods, droughts, and water quality, or efficiency of mitigation solutions for reducing water pollution. Distributed hydrological models suit particularly well these issues, because of their ability to take explicitly into account landscape heterogeneity and water pathways variety. The choice, development and application of distributed hydrological models are not easy tasks. Indeed, modelling the hydrological behaviour of a catchment implies the integration of multiple processes (infiltration, surface runoff, groundwater flow, river flow...), that have very different spatial and temporal characteristic scales (Blöschl and Sivapalan, 1995). Spatial and temporal scales range from respectively a few square meters and seconds for infiltration, to thousands of square kilometres and several years for groundwater flow. Moreover, quite different formalisms (e.g. simple statistical regression models, conceptual capacity-based models, numerical solutions of partial differential equations...) can be chosen for a given hydrological process, each one being appropriate for one case study while it is not for another. In addition, landscape management practices due to human activity, such as roads, urban areas, sewer networks, agricultural drainage, ditches, induce hydrological discontinuities (Moussa et al., 2002). They play an important role on the hydrological behaviour of a catchment and require specific process conceptualisations and space discretizations. The final modelling strategy, e.g. the choice of a model, is the result of compromises and simplifications taking into account all these elements. It is highly dependent on the size of the catchment, the objectives of the study, the target and / or dominant processes, and the available input and validation data. Therefore it is impossible to find the “perfect” modelling approach that would suit the widest range of cases. To overcome this problem, Leavesley et al. (2002) proposed a new modelling paradigm : “*this concept requires that we change the question of ‘which model is most appropriate for a specific set of criteria?’ to ‘what combination of process conceptualisations is most appropriate?’*”.

In accordance with this paradigm, environmental modelling frameworks offer the possibility to build and run “à la carte” integrated models, using modular combinations of appropriate process conceptualisations. Such software tools have been developed for several years - see for instance the state of the art established by the EU HarmonIT project (Hutchings, 2002), the special issues of *Environment Modelling & Software*, Vol 19, 2004, introduced by Argent (2004a,b) or of *Advances in Geosciences*, Vol. 4, 2005, introduced by Krause et al. (2005), or Voinov et al. (2008). Process components (also called “modules”) are the basic elements for building models using an environmental modelling framework. They correspond to various process conceptualisations; they are reusable and exchangeable. Simulation models are built by connecting these components together and are run through the framework. A communication system allows the components of a model to exchange data during simulation progress. Depending on the framework specifications, complex interactions between components such as feedbacks can be represented. The framework also usually provides facilities for adding new components, generic tools for ancillary tasks such as data input/output, and connections to database and GIS. Developers of process components can therefore concentrate on their field of interest, while taking the best from the advances of other scientific communities. At a broader scale, the use of such a framework as a common

development tool favours team work and is a precious advantage for multidisciplinary projects.

Among the main modelling frameworks, MMS (Modular Modelling System, Leavesley et al., 2002), DIAS (Dynamic Information Architecture System, Sydelko et al., 2001), SME (Spatial Modelling Environment, Voinov et al., 2004), and Catchment Modelling Toolkit projects ICMS (Interactive Component Modelling System, Rahman et al., 2004a) and Tarsier (Watson and Rahman, 2004) were pioneers for environmental and hydrological applications. These frameworks mainly differ by their specifications for adding new components and handling communication between the process components. DIAS focuses on the reusability of pre-existing legacy code. It provides “wrappers” that interface with codes written in various programming languages. The process components correspond to the “wrapped” codes. These components do not interact directly but through objects that represent real-world entities such as “fish” or “river”. Simulations allow run-time feedback between components. In SME, new components representing local processes can be added using a user-friendly graphical language. They are stored in the Library of Hydro-Ecological Modules. In models, these process components are applied and linked spatially on raster grid cells. In ICMS, the addition of new components is also done using a simplified programming language. The components are linked explicitly through a user-friendly graphical interface. On the opposite, components in Tarsier must be coded in raw C++. Communication between modules relies on a passing and receiving message system. It is totally implicit and allows feedback during simulations. Both frameworks were compared by Rahman et al. (2004b). They concluded that ICMS is more suited to users with little programming skills and relatively simple modelling needs, whereas Tarsier allows users to build more complex models, but is more difficult to use. A comparison test between SME, ICMS and Tarsier on a common application case (Argent et al., 2006) confirmed these conclusions. ICMS and Tarsier are now merged into the TIME project (Rahman et al., 2003; 2005) which intends to take the best of both approaches. TIME is a very generic framework which can handle several programming languages and includes (among other functionalities) powerful tools for data processing (statistics, parameter optimization) and visualisation.

The current main frameworks dedicated to hydrology take root from these general developments. The WaterCAST (former E2) simulation tool (Argent, 2007; Cook et al., 2009) is based on the TIME generic framework (Argent et al., 2005; Perraud et al., 2005). The basic principle of WaterCAST is a spatial discretization into sub-catchments, functional units (also known as hydrological response units) and river network. Hydrological processes as well as constituent (sediments and solutes) generation and transport are simulated on these functional units using several conceptualisations that can be selected from a list. A major point is that the process conceptualisations can be different between functional units. This introduces some spatial variability, though WaterCAST does not explicitly consider the geographical locations of functional units in a sub-catchment. The simulation engine orders the component computations according to the topography (Cook et al., 2009). WaterCAST also provides a powerful graphical user interface and parameter calibration tools. The JAMS (Jena Adaptable Modelling System) framework (Kralisch et al., 2005; Kralisch and Krause, 2006) is based on the Object Modelling System (David et al., 2004). Process components implemented in JAMS focus on hydrological and nutrient cycles (Fink et al., 2007; Kralisch et al., 2007). The spatial discretization is more flexible than in WaterCAST and can mix sub-catchments, hydrological response units, and raster cells (Kralisch et al., 2007). In a model, the hydrological process components are ordered in time and space by specific components. Recent developments in JAMS produced calibration tools and coupling to a data management system (Fischer et al., 2009; Kralisch et al., 2009). If WaterCAST and JAMS share the same care for integration of

process components and both propose a wide range of functionalities, they seem mainly convenient for relatively simple process representations. In particular, the downward oriented organization of component computations is not well suited to the simulation of feedbacks, as for example downstream influence (river level) on upstream elements (groundwater flow). Oppositely, the OpenMI (Open Modelling Interface, Gregersen et al., 2005; 2007) was designed to simulate complex interactions and feedbacks between hydrological and /or hydraulic model components, thanks to the implementation of a pull-driven mechanism by which the connected components call each other. It is now applied to an increasing number of practical cases (Devroede et al., 2008; Knapen et al., 2009). However the OpenMI is rather designed for coupling previously existing models. It provides no library of components, no facilities for developing new components from scratch or data input/output facilities.

The objective of this paper is to present a modelling platform named LIQUID[®], which has been developed since 2005 (Viallet et al., 2006). It is mostly dedicated to the modelling of hydrological processes. It aims at combining the ability to simulate complex interactions between components, like OpenMI, and an integration similar to that provided by WaterCAST and JAMS. The main characteristics of LIQUID[®] derive from the following considerations. Firstly, it should be able to represent the intrinsic space and time scales of different hydrological processes, and to include the representation of hydrological discontinuities due to human activity as mentioned above. Another important point is that the addition of new components should be easy. At last, LIQUID[®] should be adapted to a wide range of applications, ranging from local fields and small watersheds (~1 km²) to large catchments (up to ~10000 km²), and from the simplest rainfall-runoff empirical model to more refined and complicated approaches.

The general concepts and characteristics of LIQUID[®] are presented in Section 2. Section 3 highlights the most distinctive features of the platform that are the simulation run system and the spatial discretization strategy. Finally, Section 4 provides five contrasted examples of LIQUID[®] applications.

2 General description of LIQUID[®]

2.1 Components and technical specifications

The concepts of LIQUID[®] framework and LIQUID[®] platform are distinguished. The framework consists of all the core components and generic utilities, whereas the platform also includes the modelling components and their associated documentations. The main features of LIQUID[®] are shown in Fig. 1. The process components are called modules. A model is an assembly of modules that can be applied to a range of hydrological cases. LIQUID[®] users can be separated in three categories:

- developers who use the platform to create and test new modules, for example as results of their research work. They need guidelines and module templates as well as facilities for data input/output processing and module testing
- model designers who build and run new models on the basis of existing modules. They may be non-expert programmers; therefore the platform provides tools for easy model assembly
- model end-users who apply ready-made models to a given case. Their major need is an up-to-date documentation explaining how to set parameter values and run the model

LIQUID[®] is coded in C++, taking all the advantages of template and meta-template programming. The Standard Template Library (STL) and the Boost library¹ underpin most of the core components. The major core component is the Scheduler, which handles the interactions between modules and manages the time course of simulations. Its principles are detailed in Section 3. Another core component is the build system that builds the models. This component uses the Boost.Build v2 tool, that enables independence of code compilation, creation of executables etc from the operating system (Windows, Linux) and from the C++ compiler. This guarantees portability. Other third party libraries are included and can be used by module developers, such as ATLAS and LAPACK for numerical analysis, CGAL for geometry computations, OTL and SOCI for database connections. A test framework allows the user to build, run and capitalize on module tests, so that verifications of modules after after upgrades of the framework are easier. An automatic code documentation generation system is also available. The standards are ANSI C++, OpenGIS for GIS related libraries, Open DataBase Connectivity (ODBC) for database connectivity, DocBook for the documentation.

The module library contains code for the simulation of hydrological processes, with various levels of complexity. Modules are mainly based on numerical solutions of partial differential equations or on capacity-based approaches. Empirical or semi-empirical implementations are also available. Examples of modules and models that are currently used in LIQUID[®] are presented in Section 4. The modules are mostly implemented in C++, but since recently the integration of code in other languages (e.g. Fortran) is also possible and is currently being tested. More detail about LIQUID[®] concepts and implementation can be found in the online documentation at www.hydrowide.com/liquid. Documentation on the modules is also available at this address.

A collaborative development environment, through a website (<http://forge.hydrowide.com/>) enables the remote management of source code, bug tracking, and release of versions. Mailing lists and forums are used for communication between geographically scattered teams.

2.2 Modules

A module simulates one or several hydrological processes, on a simulation domain associated to of one or several spatial entities. The architecture of a module is represented in Fig. 2. It consists of five elements: a data scheme, a pre-processor, a solver, a set of unit and functional test cases, and the module documentation.

The data scheme and pre-processor provide the solver with data. In LIQUID[®], time-independent input data are managed at the module level rather than the model level. Each module comes with its own data scheme. This data scheme describes and organizes the time-independent data (parameters, initial and boundary conditions, spatial extension of the simulation domain) that the module requires. This particular feature ensures a maximum flexibility to LIQUID[®] models: when a module is removed or added, the associated data are removed or added automatically, without additional work. Another particular feature is that the interface with users for the module data is made through a database rather than input text files. According to the module's data scheme, the framework creates empty tables in a PostgreSQL/PostGIS database that is connected to LIQUID[®] through an ODBC connection. The user completes the tables with the appropriate values for the parameters, initial conditions and spatial entities the module applies on (as PostGIS geometrical objects). The pre-processor reads the data in the tables, creates, and initializes a solver instance for each spatial entity. If required, mesh generation is also performed at this step (for instance the division of a soil

¹ <http://www.boost.org/>

column into layers for the resolution of Richards equation). The pre-processor is able to process vector and raster geographical data.

The simulation is being executed in the solver. It relies on an appropriate set of state variables and governing equations. The module developer can choose freely the most convenient equations and solving methods. Each solver computes its own time step. During a simulation, this time step is estimated again at each solver execution, according to criteria defined by the module developer. These criteria may be value conditions for a given state variable (e.g. a maximal variation of 5% in one time step), or stability conditions for the numerical scheme. The solver is also able to interrupt the computation and shorten its time step in order to take into account immediately a new input value. As a result, the simulation time step can be irregular. This reproduces at best the intrinsic dynamics of the simulated physical processes.

The solver connections to the “rest of the world” are separated into signals (outputs) and slots (inputs) (Fig. 2), following a classical callback mechanism for communication between objects which is implemented in several C++ programming libraries. In LIQUID[®], the solver sends a signal to indicate a new value of a given output. This output can be any scalar, vector, matrix, function, object or structure defined by the developer. A slot is a method called when the solver receives a new input value. The slot controls the response of the solver to new input values. For example, possible changes in rainfall rate justify a slot in a soil infiltration module: the developer should implement a corresponding “OnRainChange” method and define what should be the solver’s response to changes in rainfall input. Through the use of such a push-driven mechanism, connections between modules follow the natural organization of hydrological processes, where each soil component adapts to the forcings it receives (for instance upper soil layers adapt to rainfall, groundwater adapts to recharge etc). This enables data exchange between the solvers and thus module coupling, as detailed in Section 3.

LIQUID[®] provides templates for the development of new C++ modules: the module developer only needs to complete the code of the pre-processor and the solver. The module data scheme and the solver signals and slots must be defined first. The pre-processing of the data scheme and the solver slots are then implemented, as well as the solver main computational method, that is called “OnTimer” and handles time advance in the module. LIQUID[®] provides examples of basic modules, so that the development of new modules with simple processes can be achieved with little programming skills. There are no particular rules for the names of a module’s state variables, slots and signals. However, good practices recommend that detailed metadata are given in the module documentation for each signal and slot: in particular description of the variable, reference for altitude, direction of positive values (z axis), and unit. The convention for units is the use of the SI base units for all modules (for example ms^{-1} instead of mmh^{-1}). This avoids the need for unit conversion routines in LIQUID[®] and thus potential errors in module linking. The model designer is in charge of the good agreement of variables and units when linking modules together. Therefore it is particularly important to have complete and up-to-date module documentations.

2.3 Models

A model is simply built by selecting appropriate modules and connecting their slots and signals. For this purpose, the model designer must prepare an xml file with *.model* extension, as represented in Fig. 3. The LIQUID[®] build system reads the *.model* file, translates it automatically into C++ code, compiles the appropriate modules and generates a model executable file. As shown in Fig. 3, the syntax for the *.model* file is very simple. Thus the file preparation can be done manually by non-expert programmers. It should be noticed

that the file contains only the list of the modules used in the model and the names of the connected slots and signals. The effective connections of the slots / signals for the different spatial entities are made automatically by LIQUID[®].

The LIQUID[®] module library provides Input and Output modules for the model inputs and outputs. These modules have the same structure as physical process modules with a data scheme and slots / signals. The Input module is for time series data. The model designer must use one Input module per input variable (e.g. rainfall, temperature...). The module solver reads the time series data in the user's database and sends a signal to the connected modules each time the input value changes. The data can have fixed or variable time steps. These time steps can be different for each input variable. The Output module has a generic slot to receive the outputs from the connected process modules, and prints the corresponding values in ASCII files. Several options are available for time series aggregation: variable time step (actual simulation time step), sampling of the model output at a constant time step, average value over a user-defined constant time step. The last two options are useful for long-term simulations that generate big amounts of output data, and to spare computing time. The output files can be processed by any convenient data processing software.

The run of a model takes three steps. First, a set of empty tables is generated in the user's PostgreSQL/ PostGIS database, according to the data schemes of the modules involved in the model. Once all these tables are completed by the user, the second step is the pre-processing of the modules and the initialization of the modules solvers. The last step is the model run itself, according to the start and end dates indicated by the user.

3 Particular features of LIQUID[®]

3.1 Discrete event simulator

The Scheduler (Viallet et al., 2006) is the key component of LIQUID[®] for the progress of simulations. It contains the time reference as a Gregorian date with a default time resolution of 1 nanosecond. This time reference is accessible to all the modules like a common-shared calendar. As stated before, a module solver computes its own time step. At each time step, it schedules the date of its next execution as a differed call in the Scheduler. This pair (solver, date) is called an action. At the scheduled date, the Scheduler acts as an alarm-clock and executes the action by calling the solver main computing method "OnTimer". This system is combined with the slots and signals so that modules can communicate during a simulation.

Fig. 4 illustrates the course of a simulation involving a simple one-way coupling between two modules. Module 1 can be for instance an Input module for rainfall rate time series sending this variable as a signal. Module 2 can be a soil infiltration module or any production function module receiving rainfall rate as a slot. At $t=t_0$ (Fig. 4a), Module 1 and Module 2 main methods "OnTimer" are executed. The modules schedule their next actions independently, according to their internal conditions: at t_1 for Module 2 (for example evolution of the soil moisture profile) and at t_2 for Module 1 (for example new rainfall rate value). At $t=t_1$, the Scheduler calls Module 2: its method "OnTimer" is executed and Module 2 sets its next execution time at t_3 (Fig. 4b). At $t=t_2$, "OnTimer" method for Module 1 is executed. Module 1 sends a signal indicating that the value of its output variable (e.g. Rainfall) has changed. This information is received by the corresponding slot of Module 2. This slot is implemented to take into account immediately the new value. Module 2 consequently cancels the action scheduled at t_3 and re-schedules it at the current time (t_2), so that the Scheduler calls Module 2 at t_2 instead of t_3 (Fig. 4c).

The slot implementations are crucial for the synchronization of modules and the overall simulation progress. They induce a more or less strong dependence between modules, so that all sorts of couplings can be simulated. A slot may specify that the newly received value should be used only at the next solver execution time, so that the current computation remains undisturbed (loose coupling). On the opposite, another specification may be that the new value should be used immediately and induce a new execution of the solver (tight coupling, as illustrated in Fig. 4). That way, bi-directional couplings with feedbacks can be handled by appropriate definitions of the slots, for instance through a tight coupling one-way and a loose coupling the other way. Other solutions may use iterative processes. The slots definitions should then be compatible with the stability and robustness of numerical schemes.

3.2 Space discretization and spatial coupling

A great variety of spatial representations can be used in LIQUID[®] models, from traditional regular grids, or Triangulation Irregular Networks (TINs), to more unusual spatial discretizations. In most of the current LIQUID[®] models, the catchment is discretized into hydro-landscapes representing hydrological functional units (Dehotin and Braud, 2008). The hydro-landscapes are defined according to the modelling objectives and the data availability of a given case study. They are usually obtained by overlaying several GIS vector layers. Thus they may have very irregular geometries mixing polygon, line or point elements. However, the exact method for delineation of hydro-landscapes depends on the application case and on the model, as shown in Fig. 5 for the BVFT, CVN and BALANCE models that are presented in Section 4.

In order to be used in a simulation, the hydro-landscape map must be put by the user in the PostgreSQL / PostGIS database. The user must define which module should be applied on which hydro-landscape. As mentioned in Sec. 2.2, during the pre-processing of each module, LIQUID[®] creates a solver instance for each hydro-landscape the module is applied on. Therefore there is a direct correspondence between the hydro-landscapes and the spatial modelling units of a model. For example, in the case of the hydro-landscapes of the BVFT model represented in Fig 5a, there are four tile-drained fields in the catchment. At the model pre-processing step, LIQUID[®] should create four solver instances of the module dedicated to the simulation of tile-drainage (see Sec. 4.3).

The process interactions between spatial units (e.g. routing of overland flow, groundwater flow...) are accounted for through the slots / signals connections. During the simulation, each individual solver is calculated on its hydro-landscape and communicates with other solvers. LIQUID[®] connects automatically the slots and signals to those of the appropriate solvers, according to the information given by the user in the PostgreSQL / PostGIS database. Typically, LIQUID[®] relies on the Id numbers provided by the user (subcatchment n°1 to river reach n°1 etc). This strategy for spatial coupling is illustrated in more details for several models and application cases in Section 4.

4 Application cases using LIQUID[®]

4.1 PESTDRAIN: a model for pesticide transport in tile-drained fields

PESTDRAIN (Branger et al., 2009) is one of the first models developed within LIQUID[®]. It is a simplified, conceptual model for simulation of pesticide transport at the local scale of a tile-drained agricultural field. It consists of three modules: SIDRA is physically-based and simulates the water flow in the saturated zone; SIRUP is capacity-based and simulates the water flow in the unsaturated zone and the surface runoff; SILASOL is transfer-function based and simulates pesticide transport in both the saturated and unsaturated zones.

More detail about the equations implemented in these modules can be found in Branger et al. (2009). SIDRA, SIRUP and SILASOL are coupled in LIQUID[®] as described in Fig. 6. The coupling between SIDRA and SIRUP is bi-directional; whereas it is only one-way for SILASOL, which uses data from SIDRA and SIRUP as inputs. The bi-directional coupling between SIDRA and SIRUP is enabled through the definition of the slots. SIDRA has a slot named “OnRechargeChange”, which is the recharge rate to the water table (Fig. 3). When this slot is called, the member variable “CurrentRecharge” is updated and the next scheduled action is cancelled and re-scheduled at current time, inducing a new execution of the SIDRA solver. SIRUP has a slot named “OnWaterLevelChange” (Fig. 3). When this slot is called, values are updated but no timer interruption is required. The new value for the water-table level is used with a delay at the next solver execution. This way the feedback of SIDRA on SIRUP can be handled without causing endless loops.

The resulting PESTDRAIN model has an irregular time step, which is very short (3 minutes or less) during rainfall events, and can be as long as one week during dry periods. This allows the model to have a good accuracy during transient periods, without losing any computational time during steady periods. PESTDRAIN is therefore able to simulate pesticide concentrations at the event scale, which is crucial for the assessment of acute contamination on surface water bodies. The model was tested on field data sets with encouraging results, and appears as a promising tool for agricultural water management (Branger et al., 2009).

4.2 ELIXIR-D2D: model coupling for detailed simulation of pipe pressurization in agricultural drained fields

The ELIXIR-D2D model was developed to better understand and predict the influence of temporary drainage pipe pressurization on agricultural drainage discharge during heavy rainfall events (Henine et al., 2010). The coupled model consists of a 1D Saint-Venant module (ELIXIR) computing flow and hydraulic head in a network of pipes and channels, and a 2D shallow water table module based on Boussinesq approximations and computing water table elevation and discharge into buried drains (D2D). Two main issues are addressed here. The first one is the simulation of pressurized conditions for both buried pipes and water table above drain. The second issue is the overall effect of topography and network organization on drainage pipe pressurization, as well as on discharge in watercourses.

Coupling between the modules is bi-directional; a new computation cycle starts at each change of boundary condition related to a result from either model. D2D and ELIXIR are linked as follows (Henine and Nedelec, 2009): ELIXIR takes as slot the time derivative of drainage outflow computed by D2D at the beginning of its time step, so that input for ELIXIR can be treated as a continuous piecewise linear function; on the other hand, D2D takes as slots the water level or equivalent pressure head inside drain pipes computed by ELIXIR at the end of its time step. The error resulting from the difference between a D2D boundary condition assumed constant after a present scheduled action and a new head in pipe computed at the next scheduled action for ELIXIR is reduced by choosing short time steps for the latter. The coupled model is applied at the scale of a tile drained field. Drainage pipes are represented explicitly with ELIXIR. D2D spatial discretization is a rectangular grid with nodes located along drain axis. Spatial data are adjusted according to field topography and slopes of pipe network.

The model was tested on data from an experimentation carried out in a 1.7 ha field of the Orgeval catchment (East of Paris, France). Its good agreement with observations helped to explain how drainage pipe pressurization can lead to a decrease and delay of drainage discharge, as well as a temporary storage of rain water in the soil. The effect of arterial

drainage channel discharge capacity on drainage flow rates could also be investigated using this model.

4.3 BVFT: a model for the assessment of landscape management influence on water fluxes in small agricultural catchments

The BVFT model was developed for assessing the influence of landscape management practices on the hydrology of small agricultural catchment, namely the Fontaine du Theil catchment (1.28 km²) located in Brittany, France (Branger, 2007). Practices such as subsurface drainage, ditches or hedgerows affect considerably the different water pathways, with possible consequences on river regimes or water quality. For example, tile drainage lowers the water table and reduces surface runoff, but is also responsible for fast water flow in the soil with a direct connection to the hydrographic network; ditches intercept surface and subsurface flows with also a fast connection to the river. In BVFT, the hydro-landscapes are directly derived from the land-use map (Fig. 5a). The model consists of five modules as represented in Fig. 7. Each module simulates the hydrological processes governing a particular hydro-landscape type. The SIDRA and SIRUP modules are for tile-drained zones. The FRER1D module is for non-drained agricultural zones. It computes vertical infiltration and evapo-transpiration using a 1D resolution of the Richards equation (Ross, 2003; Varado et al., 2006). The HEDGE module is for hedgerows. It is a capacity-based conceptual module which simulates the water table dynamics and evapo-transpiration processes, following the approach of Viaud et al. (2005). At last, the RIVER1D module is a simplified flow routing module for the ditch and river network (Branger et al., 2008). It is based on the 1D kinematic wave approximation (Moussa and Bocquillon, 1996). Flow velocity is computed using the Manning-Strickler equation. The river network is divided into several reaches with trapezoidal cross-sections.

The WTI (Water Table Interface) and WTRI (Water Table River Interface) modules are for the computation of lateral subsurface flow between the FRER1D, HEDGE and RIVER1D modules. Each WTI or WTRI solver instance applies on the contact zone between two adjacent hydro-landscapes. This contact zone and the Ids of the corresponding hydro-landscapes must be indicated by the user in the database. The WTI / WTRI solver takes as slots the water table levels of these spatial units (or the water level in the river reach) and sends back to them a lateral subsurface flow in m³s⁻¹. This lateral flow is calculated according to Darcy law for agricultural fields and/or hedgerows (WTI module). The Miles (1985) approach is used for saturated lateral flow between fields / hedgerows and river reaches (WTRI module). The FRER1D, HEDGE and RIVER1D modules all have a slot "OnSubsurfaceFlowChange" and incorporate the corresponding lateral flow into their water balance. Thus the couplings between the modules are all bi-directional: thanks to these interface modules, the groundwater flow from upstream to downstream as well as the feedback influence of the river water level on groundwater flow could be simulated.

A first application was done performing a six-month simulation during a winter period. The results show a good general behaviour of the model. In particular, the interfaces calculate adequately flow exchanges between the modules. Without any calibration, BVFT is able to reproduce the base flow at the catchment outlet with a good order of magnitude and reacts to the rainfall events with a correct timing (Branger et al., 2008). Additional validation tests are currently in progress.

4.4 CVN: a model for flash flood simulation and understanding

The CVN model was developed for the analysis of hydrological responses in the context of flash flood events in the Cévennes region. This region of about 20000 km² in south-eastern France is prone to such events due to specific climate and topography

conditions which lead to particularly heavy rainfall events mostly in autumn. The CVN model is designed to assess the possible role of the conceptualisation of physical processes and the specifications of inputs and parameters on runoff generation and catchment response. In particular, studies on the sensitivity to rainfall and the description of soil properties variability can help in locating and designing future experimental field campaigns (Anquetin et al., in press; Braud et al., in press).

The CVN model has a simple and robust design. The hydro-landscapes are the combination of a discretization in subcatchments and the cartographic soil units taken from the soil properties database (Bdsol Languedoc-Roussillon), as represented in Fig. 5b for the Alzon catchment (86 km²). The subcatchments cover an area of about 1 km² to be consistent with the radar rainfall data resolution and to simplify the assumptions on surface runoff transfer. Three modules are considered (see Fig. 8). Infiltration is simulated on each hydro-landscape by a FRER1D solver instance. The PEF module converts the ponding over soil surface simulated by FRER1D (due to the exceedance of soil infiltration capacity or to the saturation of soil profiles) into runoff flow. This runoff flow is connected to the river reach corresponding to the subcatchment the hydro-landscape belongs to, according to the Ids specified by the user in the database. The RIVER1D module simulates water flow throughout the hydrographic network. Unlike in the BVFT model, lateral flow exchanges in the saturated zone are presently neglected in CVN. As the rainfall intensities that occur during such extreme events are particularly high and the average soil depth in the region does not exceed 0.5 m (Manus et al., 2009), the water table contribution to the flow was shown to be of minor importance as compared to surface runoff. However, it should be considered for a proper simulation of interstorm periods (Braud et al., in press).

The model was applied to 32 catchments ranging from 2.5 to 100 km² in the Cévennes region for the September 8-9th, 2002 event. During this event, the raingauge network locally recorded 24h cumulated rainfall amounts greater than 600 mm, leading to a catastrophic flood with 23 casualties and damage over 1.2 billion Euros. The model was driven by high resolution radar data. Parameters were specified according to the available soil properties data, without any calibration. The simulation results on the different subcatchments were compared to estimated maximum peak discharges that were provided by a post-flood survey. They appeared to be in good agreement with the observations (Manus et al., 2009; Anquetin et al., in press). Based on the model results, the possible dominant hydrological processes were analysed: areas prone to saturation excess and areas prone to infiltration excess were distinguished (Manus et al., 2009). The soil saturation patterns that were obtained give interesting perspectives for further field surveys (Braud et al., in press). The next steps for this case study are the application of the CVN model to other catchments of the region, in particular larger catchments (up to 1000 km²) and the inclusion of evapo-transpiration processes, in order to study the influence of initial conditions.

4.5 BALANCE: a model for water balance on large-scale catchments

The BALANCE model was designed to study the long-term water balance on a large catchment and its sensitivity to modifications of land-use. The target catchment was the Saône river upper catchment (11700 km²), located in the North-East of France. This size is much larger than the previous case studies.

The important role of groundwater on this catchment, as highlighted by Engeland et al. (2006), and the necessity to represent evapo-transpiration processes, were used as guidelines for the model design. Another guideline was also the spatial discretization. Dehotin and Braud (2008) proposed and used a discretization approach based on landscape classification in order to deal with the high variability of resolution, accuracy and significance of the available

geographical data for such a large catchment. The resulting modelling units are represented in Fig. 5c.

The model design includes the FRER1D module for infiltration, modules for evapotranspiration processes according to Varado et al. (2006), the RIVER1D module for water routing, and BOUSS2D module for groundwater flow (Fig. 9). BOUSS2D was developed specifically for this application. It solves the 2D Boussinesq equation for water flow within the saturated zone, and is adapted to unstructured and irregular meshes, provided they fulfil convexity constraints. BOUSS2D applies on a set of hydro-landscapes, each hydro-landscape being a cell of the unstructured mesh. A FRER1D solver instance also applies on each hydro-landscape. The coupling between FRER1D and BOUSS2D is bi-directional: on each hydro-landscape, BOUSS2D takes as slot the recharge from the corresponding FRER1D solver instance; and FRER1D takes as slot the groundwater level from BOUSS2D. The WTRI interface module is used to calculate the lateral exchange flow between BOUSS2D and RIVER1D as for the BVFT model presented in Sec. 4.3. The feasibility of the coupling between FRER1D and BOUSS2D was demonstrated on simple test cases by Dehotin (2007). These tests should be complemented and real case studies should be simulated to confirm these preliminary results.

5 Discussion

LIQUID[®] was originally designed for research in hydrology, and, as such, has been used only by researchers for research applications thus far. These researchers, from Cemagref and Grenoble University, belong to the three user categories presented in Sec. 2. Among the co-authors, some of us were successively module developers, model designers and end-users, whereas some of us used only pre-built models.

The examples presented in Sec. 4 show that LIQUID[®] can be used for very contrasted hydrological situations and scales. The flexibility to build and apply such different models in the same environment is one of the major advantages of modelling frameworks, including LIQUID[®], in comparison to fully integrated hard-coded approaches. Moreover, some modules could be reused in several applications: SIDRA and SIRUP modules were used in both PESTDRAIN and BVFT models; FRER1D and RIVER1D modules were used in BVFT, CVN and BALANCE models. For the users, LIQUID allows more control on the models in terms of hydrological processes and articulation between these processes, as well as an appreciable gain of time for model development and setup. In addition, LIQUID[®] makes it possible to build models where modules with extremely different concepts can be used next to, and communicate with, one another. In the example of the BVFT model (Sec. 4.3), the HEDGE module is conceptual and capacity-based, whereas the FRER1D module is physically-based with a numerical resolution of the Richards equation. Both modules communicate and exchange water during simulations. This capability is a distinctive feature of LIQUID[®]. LIQUID[®] is particularly promising for the representation of hydrological discontinuities in distributed hydrological models.

The push-driven coupling system that was chosen for LIQUID[®] appears to be relevant and efficient. Simulation performance is good with quite short computing times compatible with use on desktop computers. To provide examples, the 6-month simulation of BVFT model on the Fontaine du Theil catchment (with about 200 spatial units) with daily climatic data was performed in 10 minutes on a regular laptop. The simulations of CVN model for a 48-hour period with rainfall forcing of 5 minutes and 1 km² of temporal and spatial resolution, respectively, were performed in a period of about 2 to about 10 minutes, depending on the

vertical discretization of soil layers and on the size and complexity of the catchments (from one dozen to more than 400 spatial units).

As far as practical aspects are concerned, the facilities for module development and model design are used successfully by the users. Beginner C++ programmers manage to develop modules thanks to the templates provided by the framework. However, it is necessary to ensure that the implementations are time step independent, so that the solver can be run with a variable time step. Therefore, in comparison with more classical approaches, the modelling concepts and formulations sometimes have to be questioned more thoroughly. For model design, although manual, the preparation of the *.model* file is fairly straightforward thanks to the simple syntax. Even for the more complex models presented in Sec. 4 (e.g. BVFT model, Fig. 7), the syntax of the *.model* file remains as simple as that represented in Fig. 3.

For model application, the use of a PostgreSQL /PostGIS database for input data appears to be a little challenging at first for users not familiar with databases. Adaptation time is sometimes required to get used to all the tools. Yet the fact that LIQUID[®] takes input time series with independent and variable time steps is highly appreciated because it avoids the cumbersome task of interpolating / disaggregating time series from various sources. The most demanding and time-consuming task for model setup is the preparation of GIS data in order to define the topology of the connections between the hydro-landscapes, especially for complex models (Secs. 4.3 to 4.5). Operations such as delineation of subcatchments, correction of the topology of vector layers, ordering of river reaches, identification of neighbours etc are necessary to complete the PostgreSQL / PostGIS database with adequate data for the spatial coupling. The use of GIS topological softwares can be a great help for the automation of these operations. For the post-processing of model results, the LIQUID[®] users have been developing a collection of R routines (<http://www.r-project.org/>), suited in particular to outputs with irregular time steps.

Since LIQUID[®] was originally designed for researchers, user-friendliness and development of graphical user interfaces were not the first priorities in the framework development, and are thus quite limited at the moment. However, the technical characteristics of LIQUID[®] and the possibilities offered for research in hydrological modelling compensate for this relative user-unfriendliness. The use of LIQUID[®] also enhanced communication between developers and has been a good vector for scientific collaboration. Therefore we can conclude that LIQUID[®] is an interesting tool for research applications. Providing tools for operational users (other than researchers) is a goal for LIQUID[®], but it is more a long-term objective at the moment. The operational users will likely be more interested in building models from existing models and / or use of pre-built models and will ask for more developed graphical interfaces.

6 Conclusions and perspectives

LIQUID[®] is an environmental modelling software tool developed since 2005. It is dedicated to hydrological modelling and allows the user to build custom-made models on the basis of hydrological process modules available from a module library. It also provides templates for an easy development of new modules. Most of the features of LIQUID[®] were dictated by the concern of integration of hydrological processes. The modules are directly connected to database and GIS and can be applied on irregular geometries. A discrete event simulator called Scheduler and a callback “slot and signal” mechanism enable simulations with variable time steps that respect the characteristic time scales of hydrological processes.

Modules with different process conceptualisations can be run together and exchange data. The couplings between the modules can include feedbacks.

LIQUID[®] has been used for research applications so far. The five examples provided in this paper illustrate the capacity of LIQUID[®] to deal with a wide range of spatial scales (from a 1ha field to a more than 10000 km² catchment), landscape discretization strategies and process representations.

After the excitement of the early stages, LIQUID[®] has now entered a more mature development phase. For the research audience of the framework, next developments will be oriented towards higher computational efficiency and parallel computing. Another challenging issue will be the inclusion of parameter optimization mechanisms. Further in depth studies are also required to adapt traditional numerical methods to complex geometric configurations and enhance the stability of numerical schemes. For potential operational users, user-friendliness is the main direction for future developments.

Acknowledgements

The work presented in this paper has been supported by Hydrowide, Cemagref and CNRS fundings. The CVN model set-up was done through the HYDRATE project funded by the European Commission FP6 project under the n°GOCE 37024. Part of the work was also funded by the French National Research Agency (ANR) under contract ANR-07-VULN-01.

References

Anquetin, S., Braud, I., Vannier, O., Viallet, P., Boudevillain, B., Creutin, J.-D., Manus, C., in press. Sensitivity of the hydrological response to rainfall fields estimation and soil variabilities in the context of flash floods. *Journal of Hydrology*
doi:10.1016/j.jhydrol.2010.07.002

Argent, R.M., 2004a. Concepts, methods and applications in environmental model integration. *Environmental Modelling & Software* 19 (3), 217.

Argent, R., 2004b. An overview of model integration for environmental applications - components, frameworks and semantics. *Environmental Modelling & Software* 19 (3), 219-234 .

Argent, R., Grayson, R., Podger, G., Rahman, J., Seaton, S., Perraud, J.-M., 2005. E2 - a Flexible Framework for Catchment Modelling. In Zenger, A. and Argent, R. (Eds), *Proceedings of the MODSIM 2005 International Congress on Modelling and Simulation*, Melbourne, Australia, 594-600.

Argent, R., Voinov, A., Maxwell, T., Cuddy, S., Rahman, J., Seaton, S., Vertessy, R.A., Braddock, R., 2006. Comparing modelling frameworks - A workshop approach. *Environmental Modelling & Software* 21 (7), 895-910.

Argent, R., 2007. E2 - Past, Present and Future. In Oxley, L. and Kulasiri, D. (Eds), *Proceedings of the MODSIM 2007 International Congress on Modelling and Simulation*, Christchurch, New-Zealand, 860-866.

Blöschl, G., Sivapalan, M., 1995. Scale issues in hydrological modelling: a review. *Hydrological Processes* 9, 251-290.

Branger, F., 2007. Using an environmental modelling framework for representing the influence of landscape management practices on water quality. Application to the Fontaine du Theil experimental catchment (France), PhD Thesis, Grenoble, Université Joseph Fourier, 220 pp (in French).

Branger, F., Braud, I., Viallet, P., Debionne, S., 2008. Modelling the influence of landscape management practices on the hydrology of a small agricultural catchment. In Wang, S.S.Y., Kawahara, M., Holtz, K.P., Tsujimoto, T., and Toda, Y. (Eds), *Proceedings of the 8th International Conference on HydroScience and Engineering*, Nagoya, Japan, 586-594.

Branger, F., Tournebize, J., Carluier, N., Kao, C., Braud, I., Vauclin, M., 2009. A simplified modelling approach for pesticide transport in a tile-drained field: the PESTDRAIN model. *Agricultural Water Management* 96, 415-428.

Braud, I., Roux, H., Anquetin, S., Maubourguet, M.-M., Manus, C., Viallet, P., Dartus, D., 2010. The use of distributed hydrological models for the Gard 2002 Flash-Flood event. Analysis of associated hydrological processes. *Journal of Hydrology*
doi:10.1016/j.jhydrol.2010.03.033.

Cook, F.J., Jordan, P.W., Waters, D.K., Rahman, J., 2009. WaterCAST - Whole of Catchment Hydrology Model: An Overview. In Anderssen, R.S., Braddock, R.D., and Newham, L.T.H (Eds), *Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation*, Cairns, Australia, 3492-3498.

David, O., Schneider, I., Leavesley, G., 2004. Metadata and Modeling Frameworks: the Object Modeling System Example. *Proceedings of the iEMSs 2d Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2004)*, Osnabrück, Germany.

Dehotin, J., 2007 *Prise en compte de l'hétérogénéité des surfaces continentales dans la modélisation hydrologique distribuée. Exemple d'application avec des données du haut-bassin de la Saône*. PhD Thesis, Institut National Polytechnique de Grenoble (in French).

Dehotin, J., Braud, I., 2008. Which spatial discretization for distributed hydrological models? Proposition of a methodology and illustration for medium to large scale catchments. *Hydrology and Earth System Sciences* 12, 769-796.

Devroede, N., Ronse, Y., Van Assel, J., Vereecken, H., 2008. Demonstration of integrated modelling using OpenMI in the Schelt River Basin. In Sánchez-Marrè, M., Béjar, J., Comas, J., Rizzoli, A., and Guariso, G. (Eds), *Proceedings of the iEMSs Fourth Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2008)*, Barcelona, Spain, 1094-1101.

Engeland, K., Braud, I., Gottschalk, L., Leblois, E., 2006. Multi-objective regional modelling. *Journal of Hydrology* 327(3-4), 339-351.

Fink, M., Krause, P., Kralisch, S., Bende-Michl, U., Flügel, W., 2007. Development and application of the modelling system J2000-S for the EU-water framework directive. *Advances in Geosciences* 11, 123-130.

Fischer, C., Kralisch, S., Krause, P., Fink, M., Flügel, W., 2009. Calibration of hydrological model parameters with the JAMS framework. In Anderssen, R.S., Braddock, R.D., and Newham, L.T.H (Eds). Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Cairns, Australia, 866-872.

Gregersen, J., Gijbers, P., Westen, S., Blind, M., 2005. OpenMI: the essential concepts and their implications for legacy software. *Advances in Geosciences* 4, 37-44.

Gregersen, J.B., Gijbers, P.J., Westen, S., 2007. OpenMI: Open Modelling Interface. *Journal of Hydroinformatics* 9, 175-191.

Henine, H., Nédélec, Y., 2009. Effect of Overpressure on the Discharge of a Tile Drainage System: An Integrated Modeling Strategy. Proceedings of the 33rd Congress of IAHR - Water engineering for a sustainable environment, Vancouver, Canada, 4078-4085.

Henine, H., Nédélec, Y., Augeard, B., Birgand, F., Chaumont, C., Ribstein, P., Kao, C., 2010. Effect of pipe pressurization on the discharge of a tile drainage system. *Vadose Zone Journal*, 9, 36-42.

Hutchings, C., 2002. State of the art review, HarmonIT project report for WP1, HR Wallingford, Report SR 598, September 2002, 81 pp. Available at <http://www.harmonit.org/documents/pubview.php?view=1&doctype=all>

Knapen, M.J.R., Verweij, P., Wien, J.E., Hummel, S., 2009. OpenMI - The universal glue for integrated modelling? In Anderssen, R.S., Braddock, R.D., and Newham, L.T.H (Eds) Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Cairns, Australia, 895-901.

Kralisch, S., Krause, P., David, O., 2005. Using the object modelling system for hydrological model development and application. *Advances in Geosciences* 4, 75-81.

Kralisch, S., Krause, P., 2006. JAMS - a framework for natural resource model development and application. In Gourbesville, P., Cunge, J., Guinot, V. and Liong, S.Y. (Eds), Proceedings of the 7th International Conference on Hydroinformatics, Nice, France, 3, 2356-2363.

Kralisch, S., Krause, P., Fink, M., Fischer, C., Flügel, W., 2007. Component based environmental modelling using the JAMS framework. In Oxley, L. and Kulasiri, D. (Eds), Proceedings of the MODSIM 2007 International Congress on Modelling and Simulation, Christchurch, New-Zealand, 812-818.

Kralisch, S., Zander, F., Krause, P., 2009. Coupling the RBIS Environmental Information System and the JAMS Modelling Framework. In Anderssen, R.S., Braddock, R.D., and Newham, L.T.H (Eds), Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Cairns, Australia, 902-908.

Krause, P., Kralisch, S., Flügel, W.A., 2005. Preface: Model integration and development of modular modelling systems. *Advances in Geosciences* 4, 1-2.

Leavesley, G., Markstrom, S., Restrepo, P., Viger, R., 2002. A modular approach to addressing model design, scale, and parameter estimation issues in distributed hydrological modelling. *Hydrological Processes* 16, 173-187.

Manus, C., Anquetin, S., Braud, I., Vandervaere, J.-P., Creutin, J.-D., Viallet, P., Gaume, E., 2009. A modeling approach to assess the hydrological response of small mediterranean catchments to the variability of soil characteristics in a context of extreme events. *Hydrology and Earth System Sciences* 13, 79-97.

Miles, J., 1985. The representation of flows to partially penetrating rivers using groundwater flow models, *Journal of Hydrology* 82, 341-355.

Moussa, R., Bocquillon, C., 1996. Criteria for the choice of flood-routing methods in natural channels. *Journal of Hydrology* 186, 1-30.

Moussa, R., Voltz, M., Andrieux, P., 2002. Effects of the spatial organization of agricultural management on the hydrological behaviour of a farmed catchment during flood events. *Hydrological Processes* 16, 393-412.

Perraud, J.-M., Seaton, S., Rahman, J., Davis, G., Argent, R., Podger, G., 2005. The architecture of the E2 catchment modelling framework. In Zerger, A. and Argent, R. (Eds), *Proceedings of the MODSIM 2005 International Congress on Modelling and Simulation*, Melbourne, Australia, 690-696.

Rahman, J., Seaton, S., Perraud, J., Hotham, H., Verrelli, D., Coleman, J., 2003. It's TIME for a new environmental modelling framework. In Post, D. (Ed), *Proceedings of the MODSIM 2003 International Congress on Modelling and Simulation*, Townsville, Australia, 4, 1727-1732.

Rahman, J., Seaton, S., Cuddy, S., 2004a. Making frameworks more usable: using model introspection and metadata to develop model processing tools. *Environmental Modelling & Software* 19, 275-284.

Rahman, J., Cuddy, S., Watson, F., 2004b. Tarsier and ICMS: two approaches to framework development, *Mathematics and Computers in Simulation* 64, 339-350.

Rahman, J., Perraud, J.-M., Seaton, S., Hotham, H., Murray, N., Leighton, B., Freebairn, A., Davis, G., Bridgart, R., 2005. Evolution of TIME. In Zerger, A. and Argent, R. (Eds), *Proceedings of the MODSIM 2005 International Congress on Modelling and Simulation*, Melbourne, Australia, 697-703.

Ross, P.J., 2003. Modeling soil water and solute transport. Fast, simplified numerical solution. *Agronomy Journal* 95(6), 1352-1361.

Sydelko, P., Hlohowskyj, I., Majerus, K., Christiansen, J., Dolph, J., 2001. An object-oriented framework for dynamic ecosystem modeling: application for integrated risk assessment. *The Science of the Total Environment* 274, 271-281.

Varado, N., Braud, I., Ross, P.J., 2006. Development and assessment of an efficient numerical solution of the Richards' equation including root extraction by plants. *Journal of Hydrology* 323(1-4), 258-275.

Viallet, P., Debionne, S. Braud, I., Dehotin, J., Haverkamp, R., Saâdi, Z., Anquetin, S., Branger, F., Varado, N., 2006. Towards multi-scale integrated hydrological models using the LIQUID® framework. In Gourbesville, P., Cunge, J., Guinot, V. and Liong, S.Y. (Eds), *Proceedings of the 7th International Conference on Hydroinformatics, Nice, France, 1*, 542-549.

Viaud, V., Durand, P., Merot, P., Sauboua, E., Saadi, Z., 2005. Modeling the impact of spatial structure of a hedge network on the hydrology of a small catchment in temperate climate. *Agricultural Water Management* 74(2), 135-163.

Voinov, A., Fitz, C., Boumans, R., Costanza, R., 2004. Modular ecosystem modelling. *Environmental Modelling & Software* 19(3), 285-304.

Voinov, A., Arctur, D., Zaslavskiy, I., Ali, S., 2008. Community-based software tools to support part modelling: a vision. In Sánchez-Marrè, M., Béjar, J., Comas, J., Rizzoli, A., and Guariso, G. (Eds), *Proceedings of the iEMSs Fourth Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2008)*, Barcelona, Spain, 766-773.

Watson, F., Rahman, J., 2004. Tarsier: a practical software framework for model development, testing and deployment. *Environmental Modelling & Software* 19, 245-260.

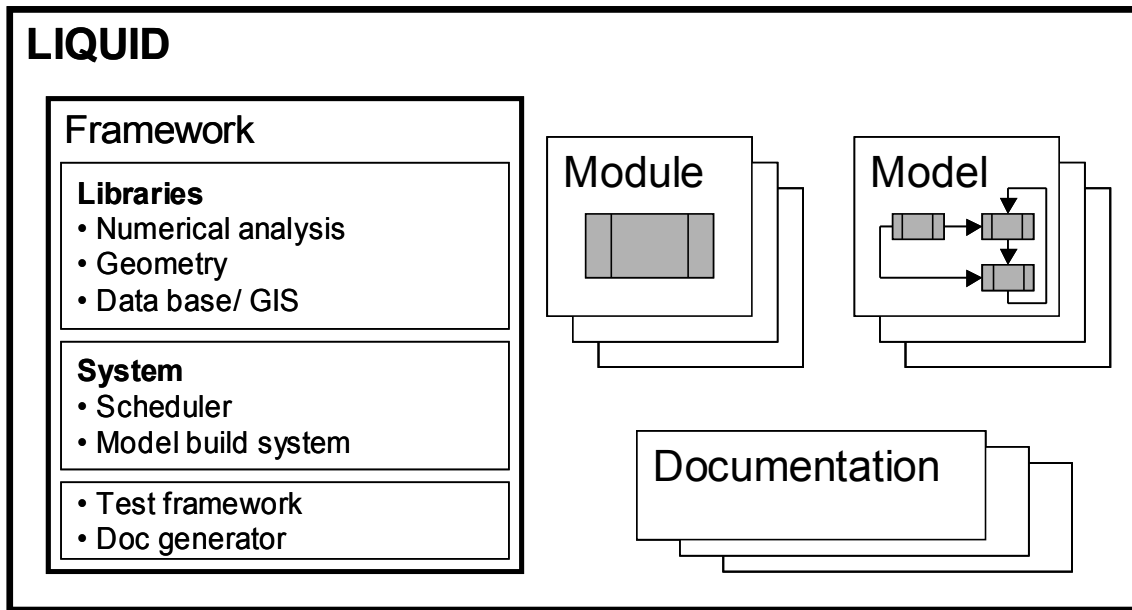


Figure 1: main components of the LIQUID[®] modelling platform.

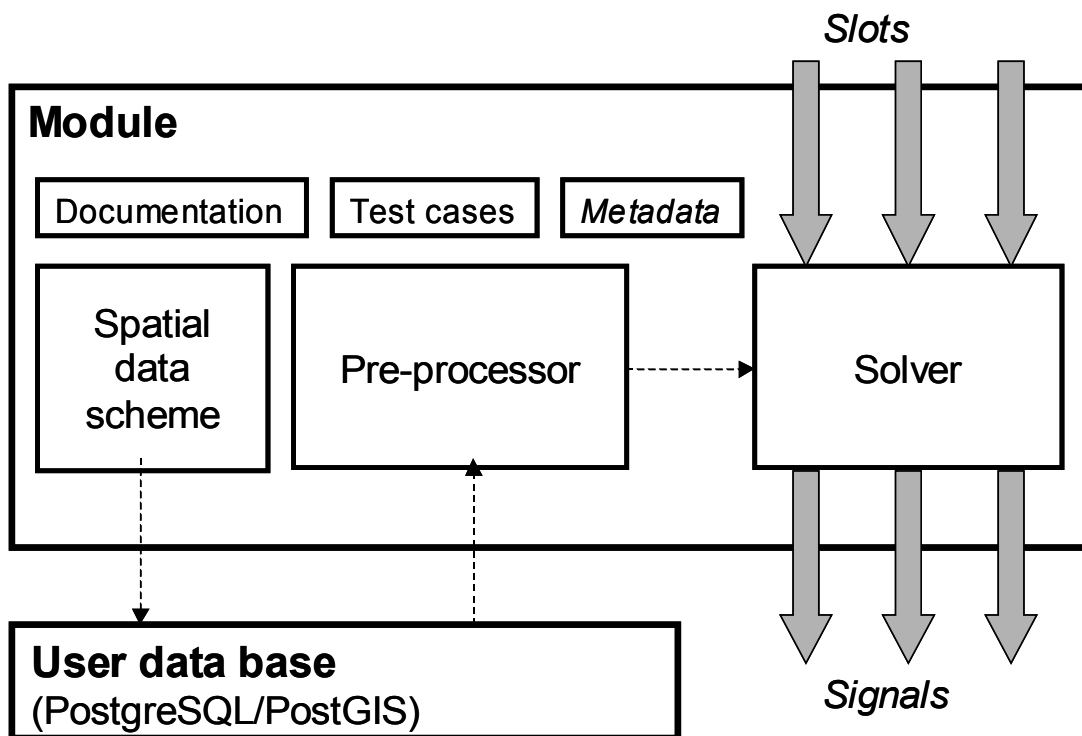


Figure 2: structure of a module.

hal-00540923, version 1 - 29 Nov 2010

```
<module>
  <type>sidra</type>
  <instance>Sidra</instance>
  <params>" "</params>
</module>
<module>
  <type>sirup</type>
  <instance>Sirup</instance>
  <params>" "</params>
</module>
<connection>
  <name> </name>
  <from>
    <instance>Sirup</instance>
    <signal>m_sigRecharge</signal>
  </from>
  <to>
    <instance>Sidra</instance>
    <slot>OnRechargeChange</slot>
  </to>
</connection>
```

Figure 3: extract from the *.model* file of the PESTDRAIN model (see 4.1). The SIRUP module computes water infiltration in the unsaturated zone. The SIDRA module simulates water flow in the saturated zone. The signal “m_sigRecharge” of SIRUP corresponds to the recharge to the water table. It is connected to the slot “OnRechargeChange” of SIDRA.

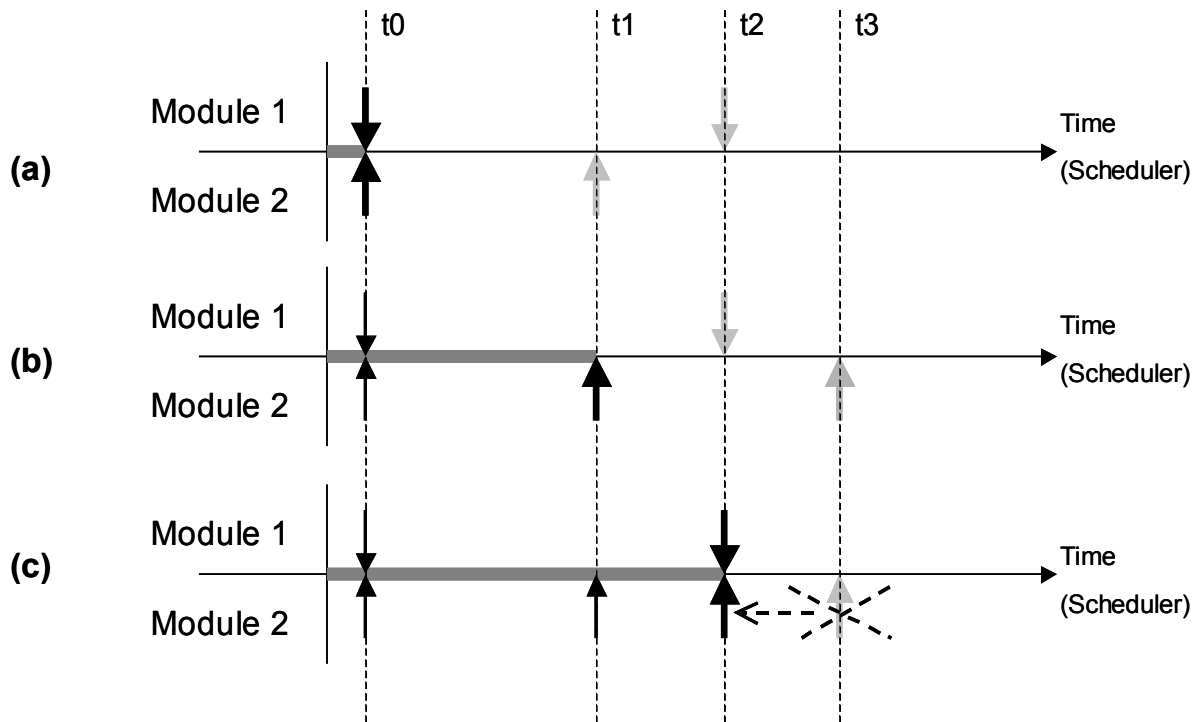


Figure 4: Example of communication between two modules (Module 1 and Module 2) during a simulation. (a): initial situation at $t=t_0$; (b): situation at $t=t_1$; (c): situation at $t=t_2$. Effective time steps are represented with black arrows and planned time steps with gray arrows.

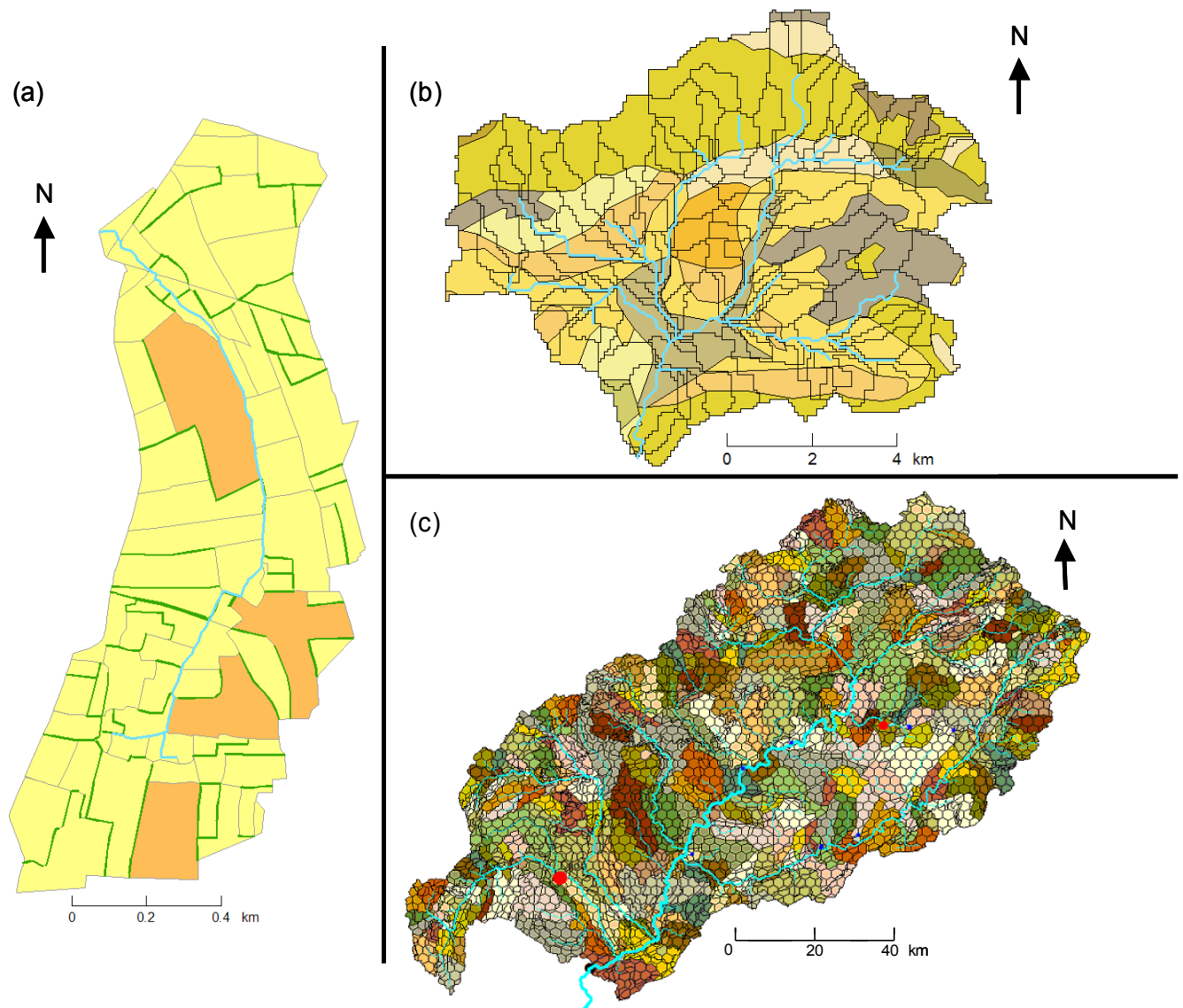


Figure 5: Examples of irregular spatial discretizations in hydro-landscapes. (a) Case of BVFT model on the Fontaine du Theil catchment (see 4.2): hydro-landscapes follow the limits of the agricultural fields, according to the land-use map. (b) Case of CVN model on the Alzon catchment (see 4.3): hydro-landscapes are a combination of subcatchments and pedological units from the Bdsol Languedoc-Roussillon. (c) Case of BALANCE model on the Saône catchment (see 4.4): hydro-landscapes result from a landscape classification approach and were slightly modified to fulfil convexity constraints.

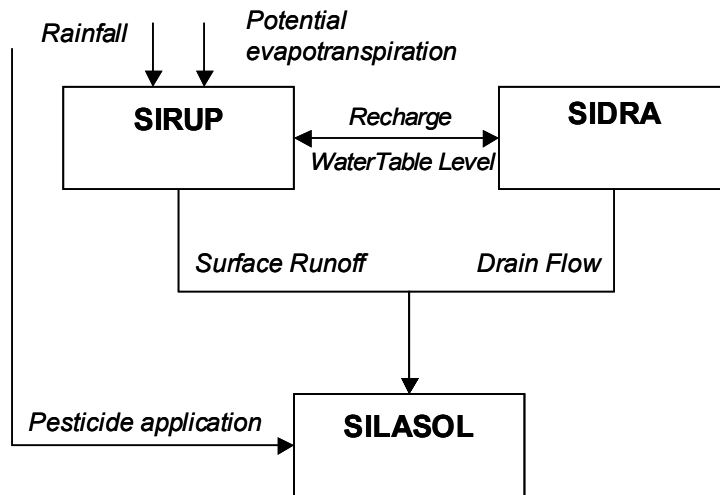


Figure 6: Structure of the PESTDRAIN model and couplings between the three modules SIDRA, SIRUP and SILASOL.

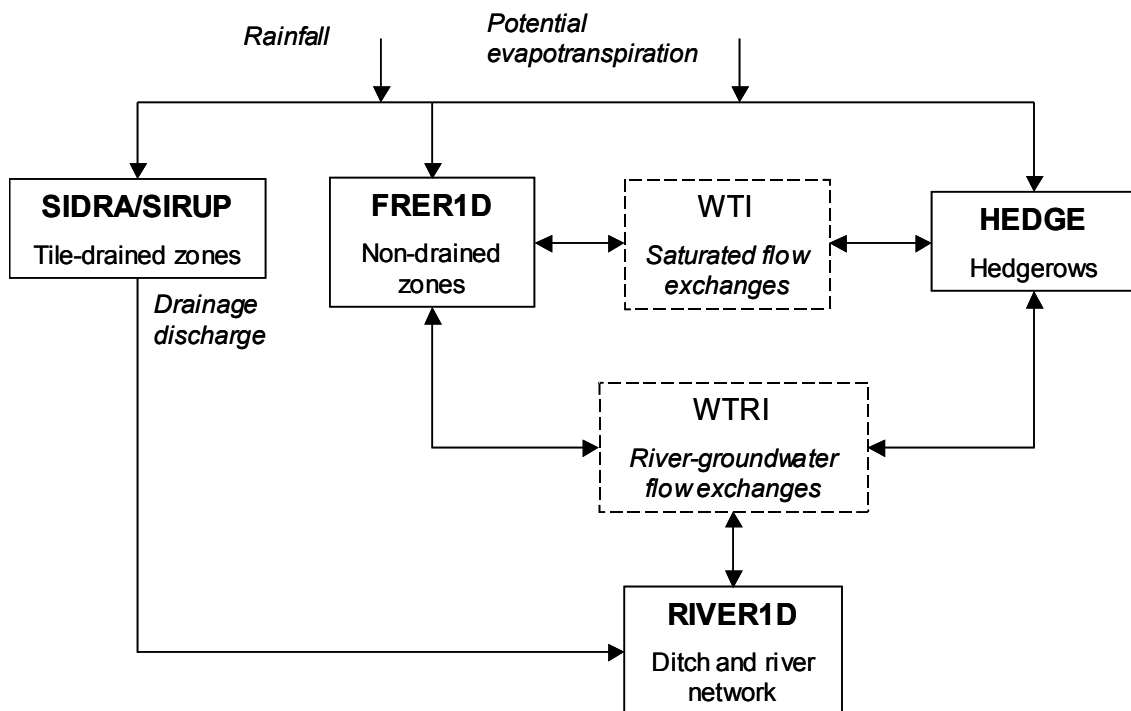


Figure 7: Structure of the BVFT model and couplings between the modules.

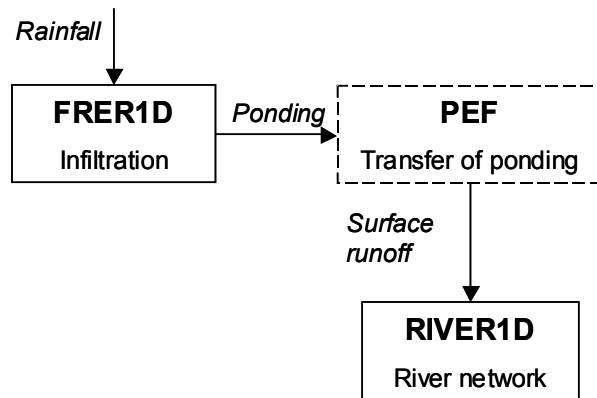


Figure 8: Structure of the CVN model and couplings between the modules.

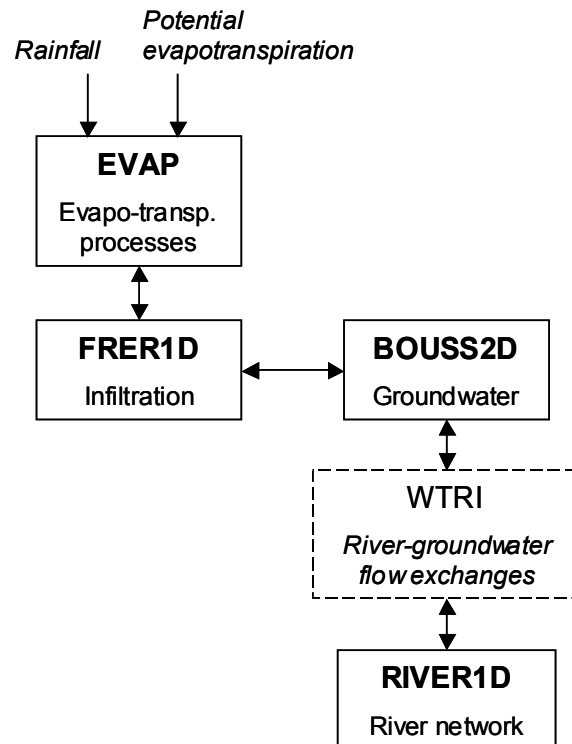


Figure 9: Design of the BALANCE model.