Object-Oriented Framework for Modelling of Pollutant Transport in River Network

Gennadiy Donchyts and Mark Zheleznyak

Institute of the Mathematical Machines and System Problems,
National Academy of Science of Ukraine,
Prospekt Glushkova, 42, Kiev, 03187, Ukraine
don@env.com.ua

Abstract. The object-oriented framework is developed using C++ language and generic programming for one-dimensional river transport modelling. The set of template classes is introduced for the better software reuse and additional ability for libraries to be extended with new sub-models. The presented model is developed using the new OOP framework coupled with modern third-party linear algebra methods libraries (MTL/ITL). The classes dealing with river topology, time series and other model elements are introduced. The new approach makes it possible to split the task of a complex river network simulation into a set of simple 1D branch simulations. This minimizes the amount of code and makes program more understandable and easy to extend. The template programming preserves high computational performance of the system and this allows program to run as fast as its C/FORTRAN analogs.

1 Introduction

Object Oriented Programming (OOP) is widely used in modelling of complex systems. Due to abstraction introduced by OOP the development process becomes more clear and easy [6,7,8]. At the same time using the OOP in development of numerical models of environment is rather difficult task for modellers, as it requires deep understating of all OO methods. The hardest task in OOP is the class identification for the data domain. But this task becomes even harder during decomposition of the classes which do not represent physical objects, for example boundary conditions, numerical scheme, etc. During model software development one of the most important facts is the computational speed and this should be taken into account when using OOP methods. The virtual inheritance in sensitive parts of an OO program can cause a program work much slower than its FORTRAN/C analogue. In this article the application of the OOP methods for classes' decomposition in considered for establishment the efficient environment for the computational one-dimensional mathematical model of pollutant transport in rivers. The OO model actively uses generic programming, which may be called "programming with concepts". Here a concept is defined as a family of abstractions that are all related by a common set of requirements. In our case it can be used for example in unification of a boundary condition definition, or algorithm for solution of equations by different numerical schemes. Using generic programming any algorithms in the program can

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 35–44, 2003.

be easily replaced by another one and at the same time they will be very fast because of the compile-time polymorphism [10]. Generic algorithms are also actively used in the last standard of the C++ language, in realization of its Standard Template Library STL, [3]. Problem of object-oriented framework development for numerical analysis is discussed in more general way in [1]. MTL library used in the model development is presented in [8] and used as a linear algebra engine. The developed OO model is implemented in the river radionuclide transport model RIVTOX [13] the EU real-time on-line decision support system for offsite nuclear emergency management – RODOS [14].

2 Mathematical Model

The mathematical models of the pollutant transport in rivers use as input data the results of the modelling of the governing hydraulics processes – water surface dynamics and currents and parameters of the suspended sediment transport [9]. For the large scale processes, when contamination is propagated on the distances much larger than river width in the network of the river channels, the one-dimensional models are used. The computational problem for multi-phase pollutant in the large network of the river channels, taking into account the boundary conditions in each junction, can require significant computing time. The development of the efficient algorithms and computer codes for such problem is especially important for the real-time applications. The one-dimensional radionuclide transport model RIVTOX is a part of the real-time on-line decision support system for off-site nuclear emergency management – RODOS [14]. It includes hydraulic submodel describing water dynamics and suspended sediment transport and radionuclide transport submodel. The basic approaches, assumptions and equations of RIVTOX are described in [13-15]. Here we will present only the structure of the basic equations.

RIVTOX includes two submodels for simulation crossectional averaged flow velocity and water elevation in a network of river canals. The first one is based on the hyperbolic system the Saint- Venant equations

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=q_l \tag{1}$$

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)+gA\left(\frac{\partial h}{\partial x}+S_f\right)=0 \tag{2}$$

where Q – water discharge A – water sectional area h – water depth, q_l – water discharge of lateral inflow, distributed along stream.

The friction slope S_f is calculated using one of the empirical resistance laws, such as Chezy's or Manning's:

$$S_f=\frac{Q|Q|}{K^2}, \quad K=C_{Cz}A\sqrt{R} \tag{3}$$

where K – the stream metering characteristics C_{Cz} – the empirical Chezy's friction coefficient .

The Second one is “diffusive wave” simplified form of the Saint-Venant equations [9].

$$\frac{\partial Q}{\partial t}+V_w\frac{\partial Q}{\partial x}-E_{wd}\frac{\partial^2 Q}{\partial x^2}-V_w q_l=0 \tag{4}$$

here V_w is the wave propagation velocity (wave celerity), and E_{wd} is the diffusion coefficient [13].

The transport of crossectionally averaged concentration of the suspended sediments in river channels *S* is described by the 1-D advection -diffusion equation

$$\frac{\partial(AS)}{\partial t}+\frac{\partial(QS)}{\partial x}-\frac{\partial}{\partial x}\left(E_S\frac{\partial(AS)}{\partial x}\right)=\Phi_b+\Phi_l \tag{5}$$

that includes a sink-source term Φ_b describing sedimentation and resuspension rates and lateral distributed inflow of sediments Φ_l. Φ_b is a vertical sediment flux at the bottom, describing sedimentation or resuspension processes in the dependence on the flow dynamical parameters and size of bottom sediments.

The mathematical background of the models of pollutants transport in surface water is also the advection – diffusion equation. Specific for radionuclide transport models – in comparison with other water contamination models – are physical – chemical exchange processes in the system “water – suspended sediments – bottom sediments ” [12,13], The traditional approach in describing and predicting the fate of radionuclides on heterogeneous solids such as soil, suspended and bottom sediments is mainly empirical and is still based on the use of the parameterisation of simplified adsorption- sorption kinetics in particular of the equilibrium distribution coefficients $K_d = C_e^d / C_e$, - where C_e^d is the amount of the contaminant adsorbed at equilibrium on the particles (suspended or bottom sediments) , and C_e is the amount of the contaminant left at equilibrium in solution. On the basis of the adsorption kinetics equations using exchange rate coefficients and *Kd* values the sink-source terms are constructed in RIVTOX, describing the exchange processes between the radionuclide concentration in solute *C*, on suspended sediments *Cs* and in upper bottom layer of the thickness *Z** of the bottom sediments *Cb*. The system of the equation defining these concentrations can be written in the generalized form as

$$\frac{\partial\psi}{\partial t}+\frac{\partial u\psi}{\partial x}-\frac{\partial}{\partial x}\left(E\frac{\partial\left(\frac{\psi}{A}\right)}{\partial x}\right)=R\,, \tag{6}$$

where ψ, *u*, *E*, R are the following vectors

$$\psi=\begin{pmatrix}AC\\ASC^S\\ZC^b\end{pmatrix},\ u=\begin{pmatrix}U\\U\\0\end{pmatrix},\ E=\begin{pmatrix}AE_C\\AE_S\\0\end{pmatrix},\ R=\begin{pmatrix}A\left(f^C+f^{C_l}\right)\\A\left(f^{C^S}+f^{C_l^S}\right)\\Zf^{C^b}\end{pmatrix}. \tag{7}$$

And $U=Q/A$ – the cross-section averaged velocity. The components of the sink – source vector R are described in [13-15].

The system of the equations (5)-(7) describes each branch of the river network. The boundary conditions in the junctions defines conservation of the fluxes of water, sediments and radionuclides or dynamics of water elevation (Saint Venant equations (1-2)).

2.1 Computational Methods

The hyperbolic system of the equations (1–2) are solved numerically using implicit finite- difference scheme [16] with the special algorithmic procedure for the treatment of the boundary conditions on the complicated graphs [9].

The advection-diffusion equations of the water, sediment and radionuclide transport (5)-(7) are solved using a fully conservative finite-difference method with numerical integration on the grid cell [i-1/2, i+1/2], and on the time interval [n, n+1]. The method is based on the second-order, implicit-explicit version of the MPDATA numerical scheme [17], with the optional nonlinear Smolarkiewicz FCT limiter [18]. For all equation mixed implicit-explicit numerical scheme was applied. Boundary condition problem for river network was solved using sweep algorithm for graphs described in [9].

3 Object-Oriented Model

River transport model classes can be divided into three groups: first we should consider *Geographical* part that describes topology of the river network. Next part is *Data* classes used to specify physical parameters in the model together with data units. Then it is necessary to introduce classes to store data as values on the river grid, miscellaneous functions (rating curves, time series, etc.). The last part is *Numerical* classes used to solve numerical equations on the river network grid; these classes mainly represent various numerical algorithms. In [11] it is shown the integration of object-oriented model with GIS system. Here we try to define constraints for different equations via template parameters in order to make possible a replacement of different numerical schemes on the fly, without losing efficiency.

On the class diagrams only the main concept classes of the program are represented. Even without showing attributes and methods explicitly it is easy to catch main ideas.

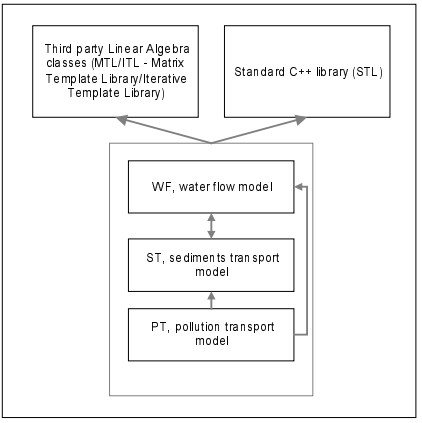

Fig. 1. Structure of the model library

To develop a new sub-model it is necessary just to implement several additional classes that contain numerical equation logic. After that the classes can be utilized as template parameters in other parts of the program. Nevertheless, during development one should always balances between efficiency, code size and understandability. Current version of model consists of three submodels (Fig. 1).

3.1 River Network Topology

The main class in this part is a *River* i.e. a container for all other spatial object such as *Node*, *Crosssection*, etc. On the diagram (Fig. 2) an aggregation link between *River* and *Branch/Node* classes becomes active when we define *BranchModel* that depends on a numerical scheme, equation type, etc.

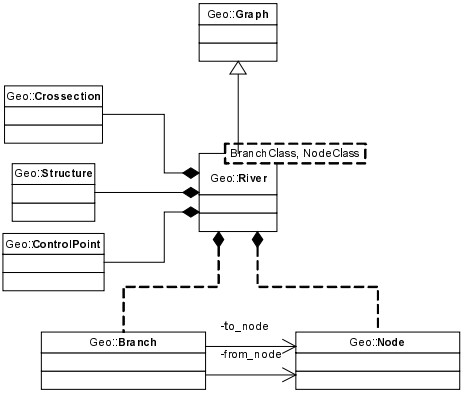

Fig. 2. Classes for management of the river topology and geometry

Crossection class describes a shape of cross-section in a certain location on a river. It would be more correct to define it as a class derived from a *Function* and *Vector* as it stores *W(h),* where *W* is a river width and *h* – its depth, later on in the program this

array is recalculated into *A(h)*. *Structure* shown on a diagram is more complex in reality, because it is necessary to redefine the algorithm of equation solving for different structures and this should be done in a specific boundary equation. The class *Structure* describes the type and parameters of the structure.
River is derived from a more general class *Graph* taken from the BGL (Boost Graph Library); the later one has lots of algorithms for solution of different problems on graphs.

3.2 Model Data Classes

Most of the data in environmental models can be described in a general way as a set of some functional sequences. Model is able to take some parameters as an input, simulate something and produce some results for output. Some data are set as values on some specific model grid (properties of the bottom, cross-sections, equation variables), some – as a common for the whole program (model parameters). Also there can appear more specific data such as boundary conditions (time series or rating curves).

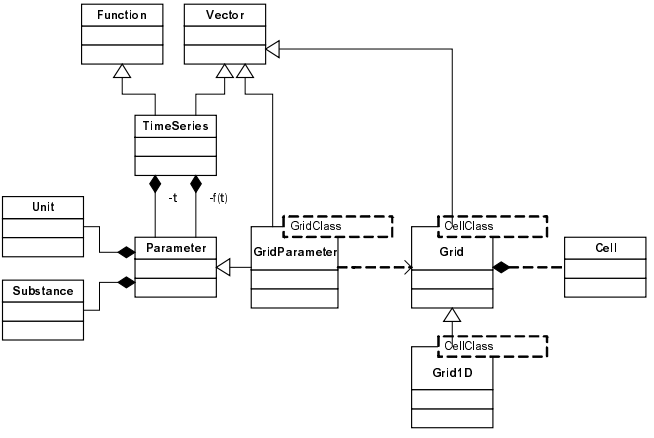

Fig. 3. Model data classes responsible for the information about physical parameters, substances, and state vector of some parameter/variable on the model space grid

In the framework the concept of *Parameter* is introduced as a class that stores information about some physical (but not necessarily) parameter (Fig. 3). Additionally there are *Units* and *Substance* to define data units of parameter. *Substance* used to define for example "concentration of Cs-137 in the water", so in this way it is possible to operate with *Parameter* "concentration in the water" and *Substance* identifying pollutant ("Cs-137" or other), plus *Units* can be set e.g. to Bq/m^3.

The *Grid* class is defined as a class built as a *Vector* of *Cells*. Later cell are defined to store some specific information needed by the model. *GridParameter* class defines value of some *Parameter* on a one-dimensional model grid. It is used to define values of model variables and parameters that depend on spatial location (cell).

Development of classes for more complex grids for multi-dimensional models is discussed in [2].

3.3 OO Numerics

To construct model of the river transport *Model* concept is defined as a basis for all classes that can be started for a simulation. The *RiverModel* class is constructed as a container class for *BranchModel* class, in this way it is possible to divide complex task of solving equation for the whole river network into a set of solving simple one-dimensional tasks on each branch.

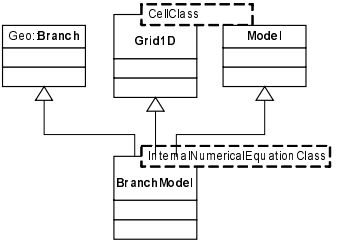

Fig. 4. BranchModel class stores logic that is necessary to run one-dimensional model on one branch of the river network using some numerical algorithm/equation

BranchModel is defined as it is shown on diagram (Fig. 4). At the same time *BranchModel* is a model, a branch on the river and a one-dimensional grid. The class also depends on a type of numerical equation given as a template parameter. *BranchModel* links with numerical equation at the compilation time, so it does not influence on a run speed of the program, but gives an opportunity to realize only general methods/properties which are common for all one-dimensional models.
The model classes are based on geographical units, e.g. *RiverModel* (Fig. 5). The numerical schemes are introduced as the template parameters of these classes. Using this approach it is possible to construct any model that simulates river network in a specific way. The *Simulation* is a container of model classes. It is necessary to create links between different sub-models and develop mechanism of synchronization to exchange data between linked sub-models. Two types of the numerical equations are used.

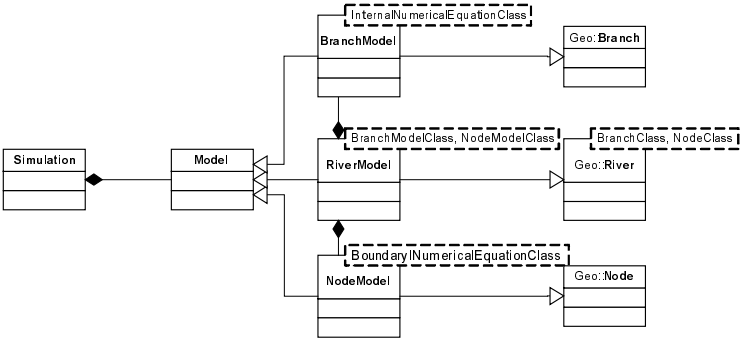

Fig. 5. Design of the *RiverModel* class; program uses this class to make simulation.

One type – for the solution of the boundary problem, for this reason the *BoundaryNumericalEquation* added to the *NodeModel* as a template parameter and is responsible for setting and solving boundary conditions in the point of connection of

several branches. The second class *InternalNumericalEquation* solves algebraic equations on one-dimensional grid on the branch using some numerical scheme (Fig. 6).

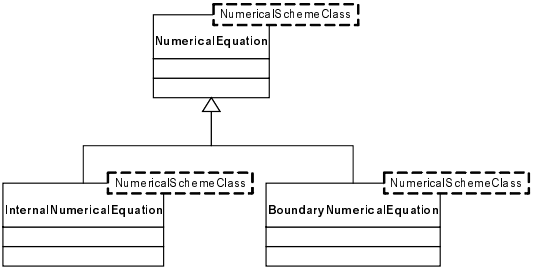

Fig. 6. Classes responsible for numerical equation

Numerical scheme for this purpose is defined separately and is used independently from equation. In fact it is hard (and even unnecessary) to separate these concepts as they are very coupled, but we can define in numerical equation classes general properties of equation, such as parameters, variables, general logic of solution etc.

From boundary numerical equation it is possible to build a hierarchy of all possible combinations of boundary conditions. When we include different structures during model application it is necessary to develop concrete classes for each structure or a group of structures with more specific properties.

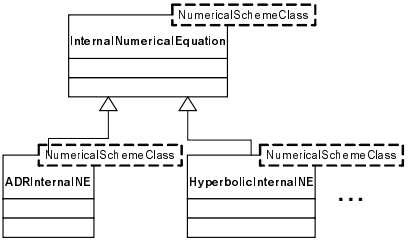

Fig. 7. Classes for internal branch numerical equations

Two types of the equations, – the hyperbolic (for water flow model) and parabolic (advection-diffusion transport with reaction) are separated in two classes due to their different properties (Fig. 7).

For each type of equation it is necessary to develop a numerical scheme used to recalculate model variables on each time step. Then after developing of several numerical schemes for one type of equations it is easy to replace them during constructing of model objects.

4 Implementation into the Modelling System

The presented methodology is used for the development of subsystem for river radionuclide transport modelling (RIVTOX [13]) of the EU real-time on-line decision support system for offsite nuclear emergency management – RODOS [14].

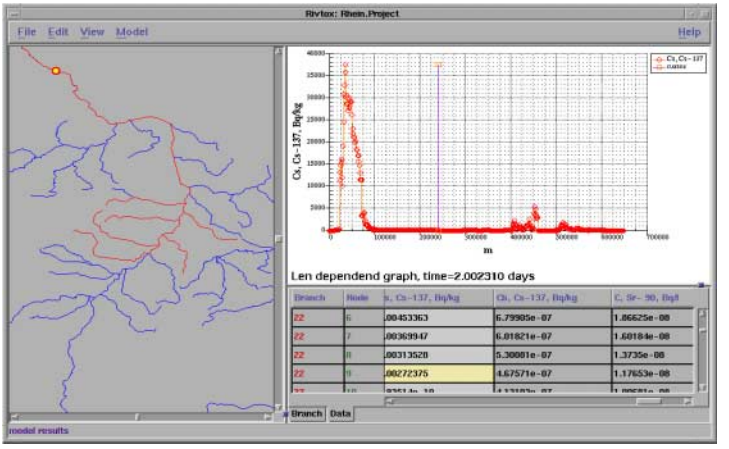

Fig. 8. River network modelling system – RIVTOX

Model code is developed using Microsoft Visual C++ compiler and ported to UNIX platform using GCC compiler. The graphical user interface (Fig. 8) was developed using Motif 1.2 XWindow library and compiled using G++. The implementation of the OO model in the RIVTOX gives possibility to diminish the size of that code and increase it transparency. The flexibility of the approach allows providing efficient extension of the module, e.g., the including of the data assimilation module into the RIVTOX has been done without changing of the main parts of the code.

5 Conclusions

The object-oriented framework is developed using C++ language and generic programming for one-dimensional river transport modelling. Model classes are designed in the way that produces efficient code which updating can be provided quickly due to the clear modular structure of the framework. Model components can be easily replaced and extended independently that gives possibility to apply and analyse different numerical solvers, that is important for the scientific modelling. The template programming preserves high computational performance of the system and this allows program to run as fast as its C/FORTRAN counterparts.

References

1. Semenov V.A., Morozov S.V., Tarlapan O.A., Shiryaeva, E.Yu.: An object-oriented framework for development of numerical simulation systems. Cybernetics issues. System programming applications, ed. Ivannikov V.P., Moscow (1997) 205–226
2. Guntram Berti: Generic Software Components for Scientific Computing, Dissertation, http://www.dissertation.de, ISBN 3-89825-169-1 , BTU Cottbus (2000)
3. Bjarne Stroustrup: The C++ Programming Language, 3rd Ed., Addison-Wesley (1997)

4. David Musser, Sibylle Schupp, Rüdiger Loos: Requirement Oriented Programming, Concepts, Implications, and Algorithms, Selected Papers. Lecture Notes in Computer Science 1766 Springer (2000), 12–24
5. Daene C. McKinney a, Ximing Cai b: Linking GIS and water resources management models: an object-oriented method, a Department of Civil Engineering, University of Texas, Austin, ELSEVIER, (2001)
6. Todd L. Veldhuizen. Techniques for scientific C++, http://osl.iu.edu/~tveldhui/papers/techniques/, (1999)
7. John J. Barton and Lee R. Nackman. Scientific and engineering C++, Addison-Wesley, (1995)
8. Andrew Lumsdaine and Jeremy Siek. The Matrix Template Library (MTL), http://www.lsc.nd.edu/research/mtl/, (1999)
9. Cunge, J.A., Holly, F.M., Jr., and Verwey, A., 1980, Practical aspects of computational river hydraulics: London, Pitman Publishing Limited , 420 p.
10. David Musser, Sibylle Schupp, Rüdiger Loos: Requirement Oriented Programming, Concepts, Implications, and Algorithms, Selected Papers. Lecture Notes in Computer Science 1766 Springer (2000), 12–24
11. Daene C. McKinney a, Ximing Cai b: Linking GIS and water resources management models: an object-oriented method, a Department of Civil Engineering, University of Texas, Austin, ELSEVIER, (2001)
12. Onishi Y., Serne J., Arnold E., Cowan C., Thompson F., Critical review: radionuclide transport, sediment transport, water quality, mathematical modeling and radionuclide adsorption/desorption mechanism (NUREG/CR-1322, Pacific Northwest Laboratory, Richland, 1981) 512 p.
13. Zheleznyak, M, Donchyts, G., Hygynyak, V., Marinetz, A., Lyashenko, G., Tkalich, P., 1997, RIVTOX – one dimensional model for the simulation of the transport of radionuclides in a network of river channels, RODOS Report WG4-TN(97)05, Forschungszentrum Karlsruhe, 2000 , 48 p.
14. Zheleznyak M.J., Tkalich P.V., Lyashenko G.B., Marinets A.V. Radionuclide aquatic dispersion model-first approach to integration into the EC decision support system on a basis of Post-Chernobyl experience. – *Radiation Protection Dosimetry*, N6, 1993, pp. 37–43.
15. Slavik O., Zheleznyak M., Dzuba N., Marinets A., Lyashenko G., Papush L., Shepeleva T., Mihaly B. Implementation of the decision support system for the river-reservoir network affected by releases from the Bohunice NPP, Slovakia – *Radiation Protection Dosimetry*, 1997, v.73, No.1–4, pp. 171–175
16. Holly, F.M., Yang, J.C., Schwarz, P., Schaefer, J., Hsu, S.H., Einhellig, R.: CHARIMA: numerical simulation on unsteady water and sediment movement in multiply connected networcs of mobile-bed channels. //IIHR Report No. 343, Iowa Institute of Hydraulic Research, University of Iowa, Iowa City, Iowa. 1990. 327 p.
17. Smolarkiewicz, P.K., Margolin, L.G.: MPDATA: A Positive Definite Solver for Geophysical Flows, J. Comp. Phys, (1996)
18. Oran, E. S. and Boris, J. P.: Numerical Simulation of Reactive Flow, Elsevier (1987)