A COMMON PROGRAMMING FRAMEWORK FOR DISTRIBUTED HYDROLOGIC MODELING RESEARCH: AN OVERVIEW OF THE ARCHITECTURE

Zhengtao Cui, Hydrology Laboratory, Office of Hydrologic Development, National Weather Service, NOAA, Silver Spring, MD, Zhengtao.Cui@noaa.gov; Victor Koren, Hydrology Laboratory, Office of Hydrologic Development, National Weather Service, NOAA, Silver Spring, MD, Victor.Koren@noaa.gov; Fekadu Moreda, Hydrology Laboratory, Office of Hydrologic Development, National Weather Service, NOAA, Silver Spring, MD, Fekadu.Moreda@noaa.gov; Michael Smith, Hydrology Laboratory, Office of Hydrologic Development, National Weather Service, NOAA, Silver Spring, MD, Michael.Smith@noaa.gov;

Abstract

It is a common task for distributed hydrologic modelers to enhance an existing model by adding new components/models or modifying existing components/models. In addition, to test the enhanced model, scientists normally need to 1) run the model with various configurations and, 2) input/output various sets of variables for better understanding. One caution is that models can easily become unmanageable as they are enhanced, leading to solutions that are very expensive to develop and difficult to maintain and extend. This presentation introduces the architectural features of a manageable programming framework designed to facilitate distributed hydrologic modeling research.

A programming framework typically is a hierarchy of node classes and functions that provides services to a theory of the problem domain. It is developed when many applications are going to be developed within a specific problem domain. Application programmers extend the framework by providing leaf classes in the hierarchy and reuse the services provided by the framework. Therefore, both design and code are reused to avoid reinventing the wheel, yet the specific actions should be supplied by the application programmer to solve his/her particular problem. The common programming framework for distributed hydrologic modeling research introduced here is an evolutionary step in the research and development program of the Hydrology Laboratory.

Most commonly available systems use the object-oriented approach to predefine a set of data abstractions (abstract classes). Typically, these systems are then extended via inheritance of these predefined data abstractions. One drawback of this approach is that the extensions are constrained to the predefined set of data abstractions. This paper presents our approach which combines object-oriented design and generic programming techniques to ensure that the framework is modular and scalable. In addition to inheritance, the resulting framework can be extended by templates as well, adding one more degree of freedom when extending the framework. Our approach affords other benefits such as type-safe and performance gains. Other architectural features such as using graph theory to manage distributed data sets, using the object factory design pattern to select simulations at runtime, and the C and FORTRAN programming interface are also discussed. The framework has been implemented in C++ programming language.

The framework is designed to be modular. Each module contains o
The modules are organized into layers. Upper layers depend on lower laye
not know the existence of upper layers. Modules in the same layer are indepe

Among various classes defined in the modules, there are three inter
the *HydroDomain*, the *Model*, and the *ModelObjectFactory*. The class *Hyd*
the physical geographic area on which hydrologic simulation will
HydroDomain is a watershed consisting of cells. There are two kinds
connected and unconnected. In a connected domain, a cell is linked to one
cells by its flow direction. An unconnected *domain* is the same as a conne
that the flow connectivity of the grid cells is either unknown or irrelevan
excess and overland flow cannot be routed from cell to cell in un-connec
quantities are nonetheless informative from a water balance standpoint.
concept has been implemented using graph theory to solve problems such as
a particular order. The *HydroDomain* is also implemented as a class te
template parameters specifies cell properties. Thus, cell properties could
developers according to their model's particular needs. For example, a deve
the framework by defining triangle cells or sub-basins with irregular shape, e
modeling requirement.

The class *Model* is designed as an abstract base class template. It a
interface for various concrete hydrologic models. The class *Model* can be
inheritance and by template. The framework uses the object factory desig
model objects at runtime for user's selection.

The *ModelObjectFactory* class was designed to create instances of va
at run-time without modifying existing code when adding new models or co
programming languages such as C++ are strongly typed, before an object can
has to be known. However, in this design, which *Model* to run is selected b
What users know is only the model name, normally a string that identifies t
the *Model* class itself designed by the programmer. Using '*switch*'-like stater
because the code has to be modified for each newly added model. The *M*
class delays the creation of class instances until run-time by knowing only the
string.

The framework also contains C and FORTRAN programming interf
that many scientists are not proficient in object oriented programming and n
were written in C or FORTRAN. Therefore C and FORTRAN programmers c
framework without switching to C++ and codes written in C or FORTR
incorporated. The investment in C and FORTRAN is protected.

Although the framework was designed for distributed hydrologic r
philosophy and data structures described here could help develop an operatio
example, the *HydroDomain* class template could also be an underlyin
operational model to take advantages of graph theory. The object facto
ModelObjectFactory, can be used to select models at run-time. From the

maintenance, the object-oriented design and generic programming techniques produce a modular and type-safe framework. Modular means errors are localized and type-safe means errors are caught at compilation-time instead of run-time.

When working within this framework, modelers are afforded the following advantages: *a)* new models can be added with minimal effort, *b)* model data are managed by a graph object, therefore various graph algorithms can be applied to solve problems such as visiting cells within a watershed in a particular order, *c)* a subset of models can be chosen to run in a simulation, *d)* a C and FORTRAN programming interface is provided to protect a modeler's investment in these languages, *e)* the resulting model is computationally efficient, and *f)* the framework has predefined simulation algorithms therefore no additional programming is needed to input/output grid and time series data after a new model is added.

U.S. ARMY CORPS OF ENGINEERS UTILIZATION AN
MANAGEMENT OF HYDROLOGIC MODELS

By James D. Barton, P.E., Chief of Columbia Basin Water Management Di
Engineers, Northwestern Division, Portland, Oregon, james.d.barton@usac
Robert A. Bank, P.E., Technology Team Leader, Corps of Engineers He
Washington, DC, robert.a.bank@usace.army.mil

Abstract
The U.S. Army Corps of Engineers (Corps) has recently initiated an effort
management of hydrologic models used throughout the agency. As one of the
water resource agencies, the Corps has numerous offices located throughout the
involved in the management of complex water resource projects and systems.
this work, a wide variety of different hydrologic models have been develope
used over time in order to meet these diverse modeling needs. Corps h
missions include planning, design, and operation of water resources projects f
reduction, coastal protection, navigation, water quality, and water supply. T
resulted in a large inventory of hydrologic models and other related software acr

The Corps has initiated its Science and Engineering Technology (SET) in
Communities of Practice (representatives of practitioners across the Corps), to
tools and practices. As a part of SET, the Corps is conducting an inventory a
their hydrologic tools and models based on criteria such as functionality, suitabi
use, compliance with security requirements, and others. Based on this informa
be made to identify key models and tools that best meet the overall needs of the
focus future investments in these tools to improve efficiency and effectiven
hydrologic models. This effort will also assist the Corps in developing it
Management and Budget (OMB) Form 300 Business Cases. The SET inve
initiated in Fiscal Year 2005, and with continued funding, will assess s
disciplines in the Corps as well.

One of the key goals in this program is to improve the overall value of investme
and other models. This could be accomplished through reduced acquisitio
maintenance, and training costs for models, increased cost effectiveness, red
inventory of models, and better standardization of technology across the ma
performing hydrologic modeling. This paper will describe these efforts to b
Corps software inventory and assess hydrologic models under the SET initiative