56TH EASTERN SNOW CONFERENCE
Fredericton, New Brunswick, Canada, 1999

Implementing the Snowmelt Runoff Model in the USGS Modular Modeling System

N.S. CAJINA,[1] K.L. BRUBAKER,[1] AND A. RANGO[2]

ABSTRACT

The snow hydrology modeling community could benefit from a systems approach, in which the user may exercise judgement and control in selecting and combining model components that best represent the physical processes of interest, in light of available data and desired results. The Modular Modeling System (MMS) developed by the U.S. Geological Survey (USGS) provides a framework for this approach. In this paper, the Martinec/Rango Snowmelt Runoff Model (SRM) is re-programmed into the MMS framework. The implementation process is presented, including conceptual modularization, variable accounting, incorporation of MMS syntax to connect the modules. Advantages of incorporating a stand-alone model into the MMS framework include the easy display of user-defined results, the ability to concentrate on process representation and process connectivity (rather than data input/output or the graphical user interface), and the potential for a deeper insight into both the model and the physical system it represents.

Key words: Snowmelt Runoff Model, Modular Modeling System, simulation, model development.

INTRODUCTION

Simulation models provide an environment to study real-world problems without the time and cost constraints of altering the actual system. In snow hydrology, watershed-scale models simulate the stores and transfers of energy and water mass, the physical processes by which precipitation is ultimately converted to runoff. Current practice in modeling snow hydrology uses a variety of self-contained, stand-alone models of varying complexity. Models differ in their scale and resolution, in their data requirements, and in their ability to simulate different phenomena (for example, point or profile water and energy balance versus streamflow hydrographs). Natural differences among and within watersheds, as well as differences in the available data, mean that two watersheds cannot be identical. Different process models are appropriate for different applications. In the case of off-the-shelf or packaged models, it is often difficult to examine the inner workings of the model, let alone adjust some part of the model procedure for a slightly different application.

Qualitative information explaining the model's conceptualization and quantitative data explaining the steps in validating the model can help ensure proper use of a model. Unfortunately, current packaged models rarely provide explicit information documenting the steps a model builder used to create the model. The model builders' oversight in communicating this important model information has produced models described by Acock and Reynolds (1990) as ''idio-syncratic monoliths'' that are comprehensible only to the model builders. Such models lead to

---

[1] Department of Civil and Environmental Engineering, 1173 Glenn L. Martin Hall, University of Maryland, College Park, Maryland 20742, U.S.A.

[2] Hydrology Laboratory, Building 007, BARC-W, Beltsville Agricultural Research Center, Beltsville, Maryland 20705 2350, U.S.A.

skepticism of their results as modelers find themselves using a black-box approach where they insert data into a model without a full understanding how their data are being processed. At the opposite extreme, modelers and their clients may become overly confident with results from these black-box models (Raymond 1996).

Users who have a thorough understanding of the modeler's strategy in describing the system can then adapt the model for their particular use. For example, in hydrology some watershed processes can be mathematically describing in different ways (for example, snowmelt could be calculated by a temperature index or energy budget method), while others are more generic (for example, adiabatic temperature lapse rate). Hydrologic modeling could be less costly and time consuming if modelers could re-use generic algorithms and also insert others where a different approach is more appropriate. But because current packaged models do not allow users to edit or adapt models to their particular scenario, there is a proliferation of custom-built models that are designed to simulate the same processes. The outcome of this repetitive work is less time available for informative analysis. The systems approach to creating a simulation model addresses problems with insufficient model documentation and the inability to adapt models.

The goal of this research effort is to determine the investment required and benefits gained by implementing an existing stand-alone simulation model into a modeling system framework that facilitates the systems approach to problem solving. This is done by detailing the steps used in implementing the PC-based Martinec/Rango Snowmelt Model (SRM) (Martinec et al. 1994) into the Modular Modeling System (MMS) (Leavesley, et al., 1996; Markstrom 1998a). Before describing the methods used, brief explanations are given on the systems approach to modeling and the MMS tool.

THE SYSTEMS APPROACH TO MODELING

The systems approach, formally known as Systems Analysis (SA), originated in the field of operations research, but has crossed over to disciplines like hydrology that rely on simulation modeling. According to Austin (1998), SA is a systematic method of solving problems and building models that takes a holistic viewpoint of the system. An appropriate SA leaves a trail of the logic and thought process throughout each step of building the model, allowing the model user to re-trace the process used in creating the model. With such documentation, the user does not have to decipher under which conditions a given model will perform best, or which of many models is appropriate for a given situation. This methodical way of producing a model tries to remove the subjectivity in assessing a model and make the task more science than art (Neelamkavil 1987).

By stressing a modular description of the system, SA eases model adaptability. A modular approach allows model users to select from alternative conceptualizations of various processes within the physical system being modeled. Re-using and sharing modules increases the efficiency of model building, and promotes collaborative modeling that can lead to productive discussions of ideas and concerns (Ziegler 1978).

The exact number or structure of the steps in a SA may vary, but a general description of them is as follows:

1. Conceptualizing the system and the processes to be modeled
2. Formulating new or assembling existing mathematical equations to describe the system
3. Characterizing the accuracy of the results
4. Use of the model for the initially intended purpose

THE MODULAR MODELING SYSTEM

The Modular Modeling System (Leavesley et al., 1996; Markstrom 1998a) is a framework that is used to create and apply environmental physical process models. The framework is composed of various tools that facilitate a systems approach to model building.

Although the MMS was introduced to the hydrology community nearly a decade ago, the system has not been widely applied in snow hydrology. Multiple explanations for this apparent lack of interest may be proposed. Having invested significant resources in stand-alone models, modelers may be reluctant to take the steps required to migrate their models to the MMS framework. They may be unaware of the actual technical requirements of doing so, or they do not perceive that the benefits of migration would justify the effort. Although it is intended to be a model-independent shell or meta-model in which infinitely many different models could be constructed, the MMS was introduced using the Precipitation-Runoff Modeling System (PRMS) (Leavesley et al. 1983), and many modelers see MMS as synonymous with PRMS. The library of modules is currently fairly small, in comparison to what is possible. Finally, MMS is currently a Unix-based system. The Unix environment is not universally popular, and many existing models have been developed and optimized for other platforms (MS-DOS or Windows, for example), where both model builders and users have developed a certain level of comfort.

The advantages of modeling within a system such as MMS include:

1. Collaborative modeling of complex systems can promote productive discussions of ideas and concerns.
2. The use of a module library promotes re-use of previously debugged and verified source code, and reduces modeling time and cost.
3. Modeling standards provide an environment where existing standards and modeling protocol are used and new ones can be developed.
4. The use of the systems approach to model building emphasizes a modular structure, reduces subjectivity in assessing a model, and generates important model documentation.

These advantages have been well documented for building new models (Maxwell 1999). However, the advantages of re-formatting existing models for use within these modeling systems are less well understood. If the benefits exceed the effort in implementing existing models into a module-based modeling system, packaged stand-alone models can be incorporated into systems like MMS and begin to benefit from the advantages of creating physical process-based models within such systems. The ultimate goal is to create high performance robust models that accurately portray the system.

A major goal of the project described here was to investigate the effort required to migrate an existing MS-DOS based model to the MMS environment, and the benefits of doing so. This section briefly describes the three main components of MMS (the pre-processor, the model builder, and the post-processor); following sections detail the process of model migration.

The pre-processing component of MMS consists of a geographic information system (GIS), termed the GIS Weasel (Viger, 1998), which is used to calculate spatially varying model parameters. It also gives a pictorial representation of the watershed by using user-supplied grid-based digital data. The GIS Weasel is accessed by a user-friendly graphical user interface (GUI) that drives various GIS tools based on the widely-used ARC/INFO system.

The model-building component is made up of a database system accessed through a GUI that organizes modules representing important physical processes. The database acts as a repository for algorithms (modules) pre-programmed in either C or FORTRAN. The modules are stored in file directories organized by process. For example, snowmelt and infiltration would be in separate directories, each of which may contain a number of routines for calculating values. When assembling a model, modelers use their conceptualization of the system to choose the appropriate algorithms (modules) within the directories. Each module has a representative icon that is inserted into the model by clicking and dragging with the mouse. The model-building software uses information about each module's input and output variables to automatically link modules together. If the automatically generated links are not correct, the software allows the user to locate the module in its appropriate position. The software also uses graphical and text-based error messages to alert users to modules that do not have all of their necessary inputs. For example, if a snowmelt module that uses daily temperature in calculating melt has not been linked to a module that produces daily temperature, the snowmelt icon will appear highlighted and attempts to compile the model will result in error messages. Successfully linked modules produce a box and arrow diagram of the model. MMS's strategy of creating an initial schematic of the model is in agreement with the opinions of various authors who suggest that such diagrams are an effective way to understand the system's conceptualization and the strategy for modeling it (Chapra 1997, Grant et al. 1997, Haefner 1996).

The post-processing component uses a GUI environment that has several options for running the model. It allows the modeler to select the variables to track using run time, post-run, 3-D, or animation types of graphical output. As a result, the user is not bound by the model developer's opinion of which variables are important to display. It also provides a built-in optimization and sensitivity analysis feature, either of which can be run simultaneously with the model. This frees the user from performing multiple runs with multiple input files that can be burdensome and monotonous.

IMPLEMENTING THE SRM INTO MMS

The Snowmelt Runoff Model (SRM) (Martinec et al. 1994) is designed to simulate and forecast daily streamflow in basins where snowmelt is a major runoff factor. The microcomputer (DOS) implementation has gone through several versions, now at version 4.01. It is a simple degree-day model that requires remote sensing input in the form of basin or zonal snow cover extent. Both the model and its documentation are available on-line (USDA ARS 1998). A Unix-based version of SRM has been developed and linked with a Relational Database and Geographical Information System for use in streamflow forecast research at the Eidgenössische Technische Hochschule Zürich (Brüsch, 1996). Recently, a Java version has been released at the University of Berne and is available on-line (Kleindeinst, 1999). Several workshops have been held for the international SRM user community (Univ. of Berne, 1998). In addition to the original degree-day version, the SRM has been expanded to a restricted degree-day version that includes net radiation as an additional predictor of snowmelt (Brubaker et al. 1996).

The initial goal of this project was to re-program the SRM for implementation into the MMS framework. Once running in the MMS environment, SRM users can perform many of the steps within a SA without ever leaving the MMS. Furthermore, additional work within the MMS may reveal the strengths and weaknesses of the SRM modules by linking the conceived SRM physical process modules with modules from other models or vice-versa.

The task of re-formatting SRM for use within MMS consisted of (1) modularizing the existing SRM model code, which was originally composed of two subroutines, (2) re-coding the modularized model from QuickBasic (QBasic) into the C programming language, and finally (3) revising the modularized C program to contain the MMS specific functions that handle memory allocation, communication of variables between modules, and input - output of data and results.

Modularizing

The strategy in modularizing the original SRM model code into modules of major physical processes was to ensure that each module contained at least one parameter. Wilby (1997) describes parameters as indices that are used in determining the behavior of the system being modeled. In the absence of actual measurements or underlying theory, parameters are often used to quantify distinct physical processes. Therefore, the SRM parameters were a natural starting point for breaking the model into its constituent physical process representations.

The temperature index version of SRM uses eight parameters:. Lapse Rate, Critical Temperature, Rainfall Contributing Area, Rain and Snow Runoff Coefficients, Recession Parameters (x and y), and Lag. Following the strategy of modularizing SRM using its parameters, the SRM parameters and their corresponding MMS modules are summarized in Table 1. Additional modules that do not correspond to particular SRM parameters are the starting point (OBSB.C) and the flow summation step (TOT_FLOW.C), for a total of ten modules.

Translation to C

Modules for MMS can be written in either the FORTRAN or C programming language (Markstrom, 1998b). The C programming language was chosen for this application based on

Table 1. SRM Physical Process Modules.

SRM Parameter and Description	MMS Module Name and Description
[none]	`OBSB.C` Reads in the values of any user-defined variables and parameters for each time step
Lapse Rate Describes the decrease in temperature with elevation; indirectly provides temperature measurements for elevation zones where temperature is not physically measured	`LAPSE.C` If the temperature values are not given for each zone, the daily basin-wide temperature is adjusted using the lapse rate to calculate the daily temperature in each zone
Critical Temperature Identifies the daily precipitation as rain or snow, based on daily average temperature	`PRECIP_TYPE.C` Determines the form of precipitation using the critical temperature parameter
Rainfall Contributing Area (RCA) A binary value that changes from 0 to 1 at a user-defined time during the melt season. When RCA is 0, rainfall is absorbed by the snowpack; when RCA is 1, rainfall becomes potential runoff immediately.	`RCAF.C` The binary RCA parameter is used to tell whether any precipitation that falls as rain is immediately available as potential runoff.
Degree-Day Factor An integrated measure of all the processes that melt snow; converts zone temperature into zone snowmelt depth.	`SNOW_MELT.C` The amount of snowmelt available as potential runoff is computed using the degree-day factor. The precipitation that falls as snow is accumulated in a reservoir separate form the existing snow pack. The potential melt is taken from the reservoir of new fallen snow and snowfall already on the ground.
Rain runoff coefficient Accounts for losses from rain and late snowfall, including evapotranspiration, sublimation, and blowing snow; varies through the melt season.	`RAIN_RUNOFF.C` The amount of rainfall runoff is computed using the rainfall runoff coefficient. The daily rainfall and new fallen snow are reduced by the rainfall-runoff coefficient parameter.
Snow runoff coefficient Accounts for losses from the snowpack, including evaporation, sublimation, and blowing snow; varies through the melt season.	`SNOW_RUNOFF.C` The amount of snowmelt runoff is computed using the snow-runoff coefficient parameter. The amount of melt from the existing snowpack in each zone is reduced by the snow-runoff coefficient parameter
Lag Represents the time lag between snowmelt or rainfall water input in the watershed and the appearance of that water at the basin outlet. Encompasses the physical processes controlling overland and subsurface event flow and channel routing.	`ZONLAG.C` The total water output is computed by adding the total water from each zone, and the time lag of the daily discharge is computed using the time lag parameter.
Recession parameters x and y Control SRM's time-varying exponential decay in baseflow during periods without water input. Held constant throughout the simulation.	`RECESS.C` The recession coefficient is computed using the x and y parameters.
[none]	`TOT_FLOW.C` The daily discharge is computed using the MRSRM model equation.

personal preference. Once the conceptual modularization of the SRM was complete, the original QBasic code was searched for the relevant portion of each module, and combined into the proposed module's C code.

An important aspect in translating the code was to identify the variables that appeared for the first time within a module and therefore must be declared, and the variables that were passed from other modules. This description helped determine which MMS functions would be included within each module. The original QBasic SRM model code consisted of two subroutines, which were modularized into ten functions in the C version. The resulting stand-alone C program was debugged and verified using a series of hand calculations of the modularized algorithm; Finally, the daily discharge computed by the modularized C program was verified against the stand-alone packaged PC version of the SRM. The original QBasic version of SRM contained numerous lines of code required to operate its GUI. Because the MMS would eventually provide the GUI, the C code contained only the model algorithm and was about one-fourth the size of the original QBasic code

Adding MMS Specific Functions

The appropriate MMS functions were added to the modularized C program to convert each of the ten functions to a module. The MMS manual (Leavesley et al., 1996) details the structure each module must take and where to include MMS specific functions; sample modules provided with the software provided guidance. Each MMS module must be composed of four functions (Declare, Initialize, Run and Main), each function of the modularized C program became the Run function of the corresponding module within MMS. The other MMS module functions were used for communicating variables between modules. For example, the DECLARE function was used to place module variables into the model's database. Adding the MMS functions to the C version of SRM only added about ten lines of code to each function. The modules were then selected and joined into a model in the SRM model builder (Figure 1). Debugging the final MMS model consisted of assuring that the daily discharge computations matched those of the original stand-alone model for a test case.

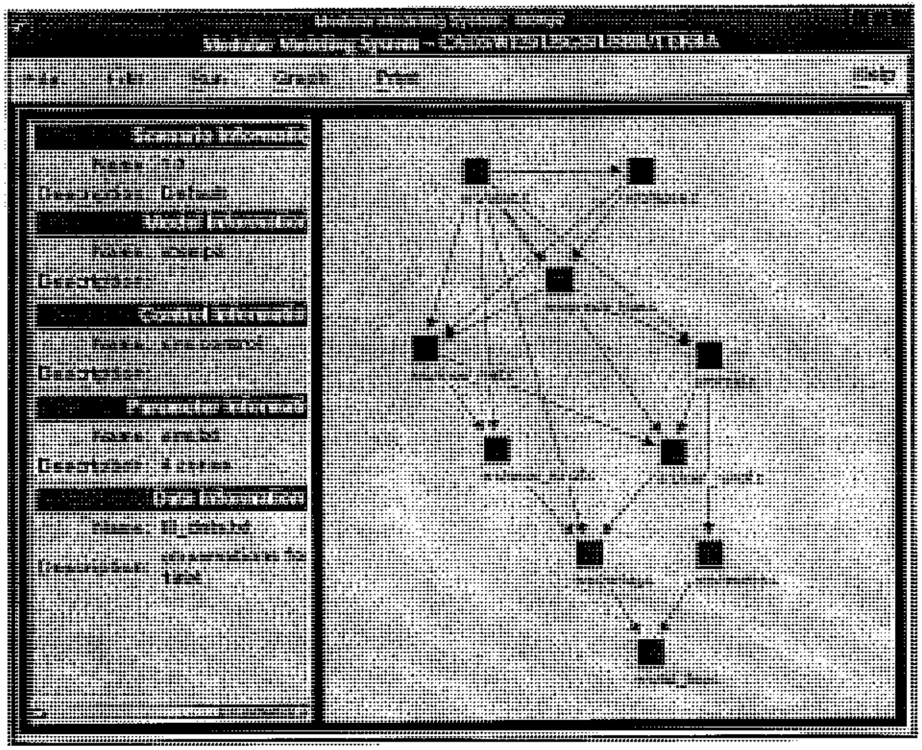


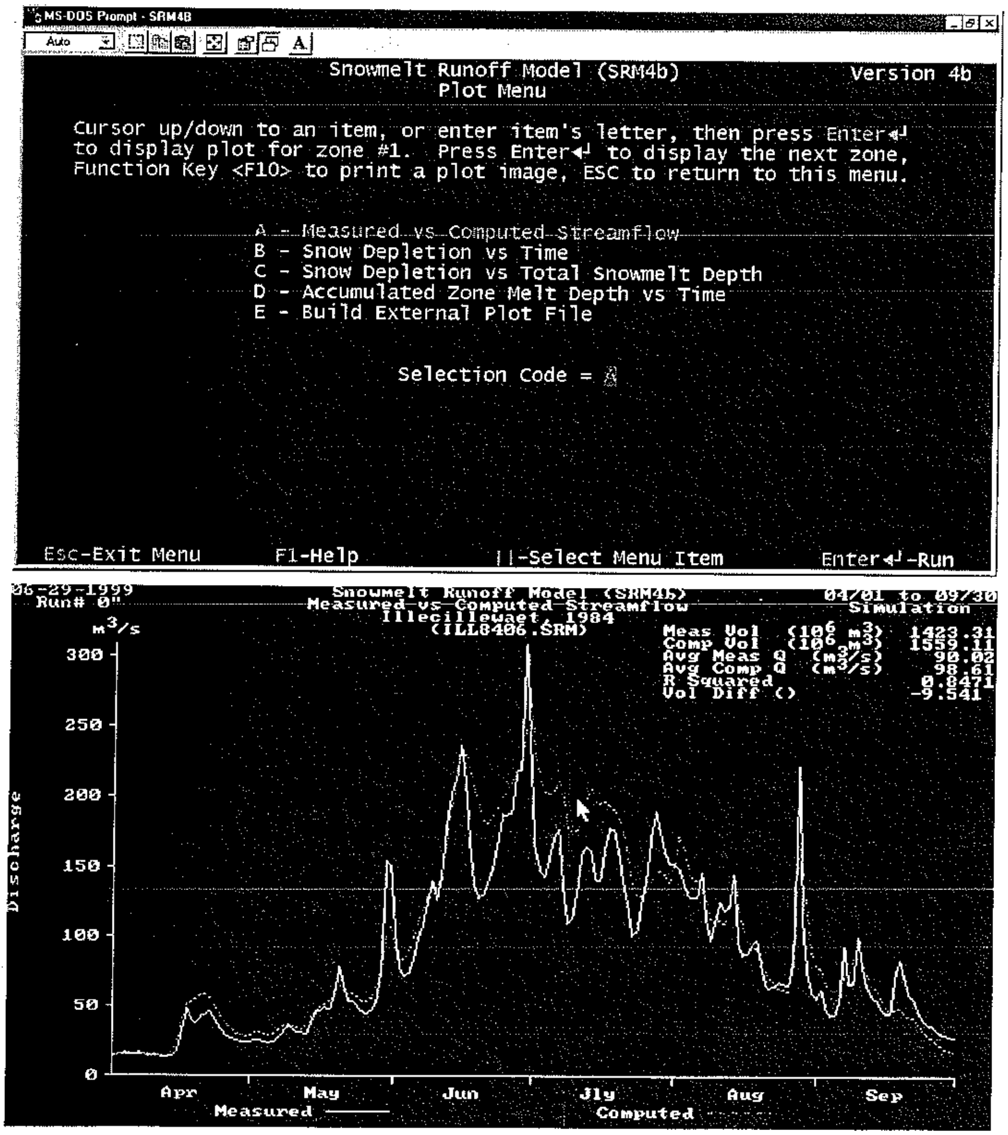
Figure 1. MMS schematic of the modularized SRM. Module names correspond to descriptions given in Table 1 (/src is the directory where modules are located). Arrows show the flow of information between modules.

RESULTS

To migrate an existing model to MMS, or to prepare a new module, requires basic programming skills in either C or FORTRAN. The required programming consists of process equations and well-defined MMS functions to declare variables and pass variables between modules. Our experience shows that being freed from programming the Graphical User Interface (GUI) and file input/output (I/O) reduces the programmer's effort significantly (about three-quarters of the original QBasic SRM code was devoted to GUI and I/O).

An understanding of the steps of Systems Analysis is a major aid to implementing an existing model within the MMS modeling system. A rough schematic diagram of the modularized SRM prepared earlier was helpful in determining which variables belonged within each module. In this project, the task of translating the code was manageable because of the extensive use of comments and consistent formats within the original QBasic SRM code; that observation underscores the importance of good model documentation in collaborative modeling efforts.

A major benefit of having migrated the SRM into MMS is improved flexibility in the graphical display of final and intermediate results. The original SRM was constrained by memory limitations to allow graphical output of a few selected model inputs and outputs (Figure 2); the MMS framework allows the user to graphically inspect any variable or intermediate calculation (Figure 3). This flexibility allows the modeler to evaluate the adequacy of internal or intermediate process representations, not only the final integrated streamflow simulation (which may be right for the wrong reasons).



```
                Snowmelt Runoff Model (SRM4b)                    Version 4b
                           Plot Menu

   Cursor up/down to an item, or enter item's letter, then press Enter↵
   to display plot for zone #1.  Press Enter↵ to display the next zone,
   Function Key <F10> to print a plot image, ESC to return to this menu.

                  A - Measured vs Computed Streamflow
                  B - Snow Depletion vs Time
                  C - Snow Depletion vs Total Snowmelt Depth
                  D - Accumulated Zone Melt Depth vs Time
                  E - Build External Plot File

                           Selection Code = A

 Esc-Exit Menu      F1-Help         ||-Select Menu Item         Enter↵-Run
```

```
06-29-1999                Snowmelt Runoff Model (SRM4b)           04/01 to 09/30
  Run# 0"               Measured vs Computed Streamflow             Simulation
                              Illecillewaet, 1984
                                (ILL8406.SRM)       Meas Vol   (10⁶ m³)   1423.31
                                                    Comp Vol   (10⁶ m³)   1559.11
                                                    Avg Meas Q  (m³/s)      90.02
                                                    Avg Comp Q  (m³/s)      98.61
                                                    R Squared               0.8471
                                                    Vol Diff ()             -9.541
```

Figure 2. Selection screen for graphical output and example of plotted streamflow hydrograph in the original (MS-DOS) SRM.

Again, because of memory limitations in the DOS environment, the original SRM accepted model input by means of screen entry and saved the input data file in a somewhat cryptic text structure designed to optimize storage (Figure 4a). The MMS data input file is an expandable column structure, well suited to exchange with spreadsheet software (Figure 4b).

Although this option was not explored in the present study, MMS can perform automated sensitivity analysis by means of repeated simulations with perturbed parameters. With the original

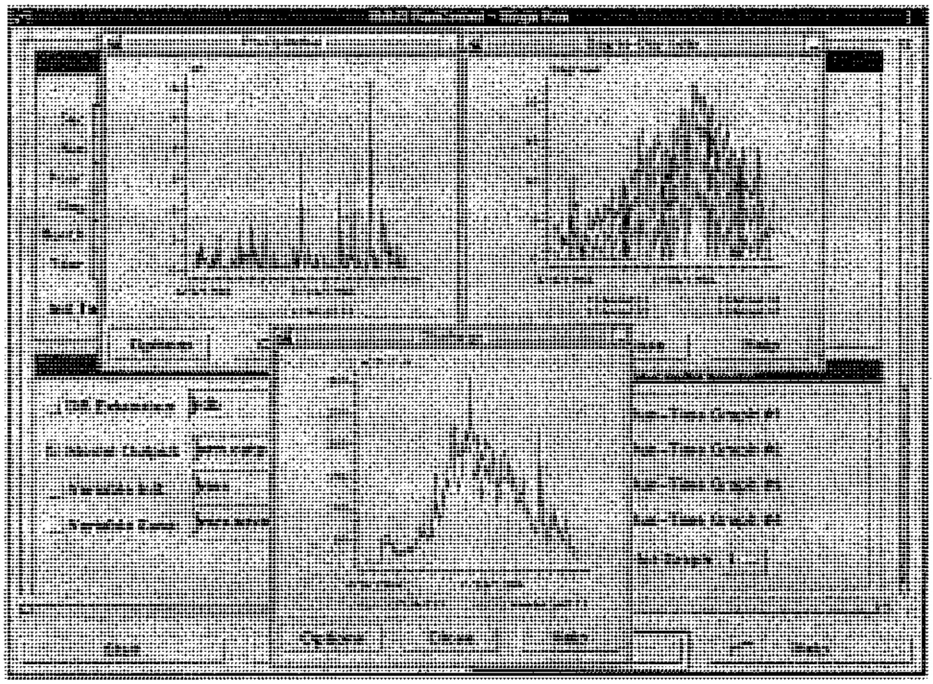
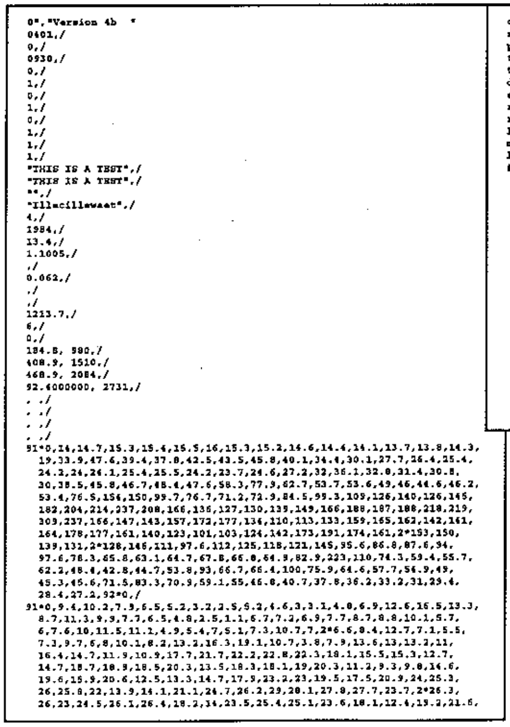
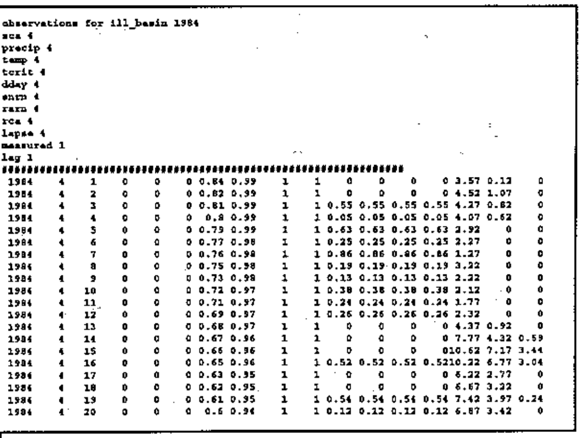


Figure 3. MMS post-processor screen showing a number of possible choices for graphical display of SRM intermediate and final results.



```
0","Version 4b  "
0401,/
0,/
0930,/
0,/
1,/
0,/
1,/
0,/
1,/
1,/
1,/
"THIS IS A TEST",/
"THIS IS A TEST",/
"",/
"Illecillewaet",/
4,/
1984,/
13.4,/
1.1005,/
,/
0.062,/
,/
,/
1213.7,/
6,/
0,/
184.8, 980,/
408.9, 1510,/
468.9, 2084,/
92.4000000, 2731,/
, ,/
, ,/
, ,/
, ,/
91*0,14,14.7,15.3,15.4,15.5,16,15.3,15.2,14.6,14.4,14.1,13.7,13.8,14.3,
  19,33.9,47.6,39.4,37.8,42.5,43.5,45.8,40.1,34.4,30.1,27.7,26.4,25.4,
  24.3,24,24.1,25.4,25.5,24.2,23.7,24.6,27.2,32,36.1,32.8,31.4,30.8,
  30,38.5,45.8,46.7,48.4,47.6,58.3,77.9,62.7,53.7,53.6,49,46,44.6,46.2,
  53.4,76.5,154,150,99.7,76.7,71.2,72.9,84.5,95.3,109,126,140,126,145,
  182,204,214,237,208,166,136,127,130,135,149,166,188,187,188,218,219,
  309,237,166,147,143,157,172,177,134,110,113,133,159,165,162,142,161,
  164,178,177,161,140,123,101,103,124,142,173,191,174,161,2*153,150,
  139,131,2*128,146,111,97.6,112,125,118,121,145,95.6,86.8,87.6,94,
  97.6,78.3,65.8,63.1,64.7,67.8,66.8,64.9,82.9,223,110,74.3,59.4,55.7,
  62.2,48.4,42.8,44.7,53.8,93,66.7,66.4,100,75.9,64.6,57.7,54.9,49,
  45.3,45.6,71.5,83.3,70.9,59.1,55,46.8,40.7,37.8,36.2,33.2,31,29.4,
  28.4,27.2,92*0,/
91*0,9.4,10.2,7.9,6.5,5.2,3.2,2.5,5.2,4.6,3,3.1,4.8,6.9,12.6,16.5,13.3,
  8.7,11,3,9,9,7.7,6.5,4.8,2.5,1.1,6.7,7.2,6.9,7.7,8.7,8.8,10.1,5.7,
  6,7.6,10,11.5,11.2,4.9,5.4,7,5.1,7.3,10.7,7,2*6.6,8.4,12.7,7.1,5.5,
  7.3,9.7,6,8,10.1,8.2,13.2,16.3,19.1,10.7,3.8,7.9,13.6,13,13.2,11,
  16.4,14.7,11.9,10.9,17.7,21.7,22.2,22.8,22.3,18.1,15.5,15.3,12.7,
  14.7,18.7,18.9,18.5,20.3,13.5,18.3,18.1,19,20.3,11.2,9.3,9.8,14.6,
  19.6,15.9,20.6,12.5,13.3,14.7,17.9,23.2,23,19.5,17.5,20.9,24,25.3,
  26,25.8,22,13.9,14.1,21.1,24.7,26.2,29,28.1,27.8,27.7,23.7,2*26.3,
  26,23,24.5,26.1,26.4,18.2,34,23.5,25.4,25.1,23.6,18.1,12.4,19.2,21.6,
```



```
observations for ill_basin 1984
sca 4
precip 4
temp 4
tcrit 4
dday 4
entn 4
rarn 4
rca 4
lapse 4
measured 1
lag 1
##################################################################
 1984    4    1    0    0    0 0.84 0.99    1    1    0    0    0    0 3.57 0.12    0
 1984    4    2    0    0    0 0.82 0.99    1    1    0    0    0    0 4.52 1.07    0
 1984    4    3    0    0    0 0.81 0.99    1    1 0.55 0.55 0.55 0.55 4.27 0.82    0
 1984    4    4    0    0    0  0.8 0.99    1    1 0.05 0.05 0.05 0.05 4.07 0.62    0
 1984    4    5    0    0    0 0.79 0.99    1    1 0.63 0.63 0.63 0.63 2.92    0    0
 1984    4    6    0    0    0 0.77 0.98    1    1 0.25 0.25 0.25 0.25 2.27    0    0
 1984    4    7    0    0    0 0.76 0.98    1    1 0.86 0.86 0.86 0.86 1.27    0    0
 1984    4    8    0    0    0 0.75 0.98    1    1 0.19 0.19 0.19 0.19 3.22    0    0
 1984    4    9    0    0    0 0.73 0.98    1    1 0.13 0.13 0.13 0.13 2.22    0    0
 1984    4   10    0    0    0 0.72 0.97    1    1 0.38 0.38 0.38 0.38 2.12    0    0
 1984    4   11    0    0    0 0.71 0.97    1    1 0.24 0.24 0.24 0.24 1.77    0    0
 1984    4   12    0    0    0 0.69 0.97    1    1 0.26 0.26 0.26 0.26 2.32    0    0
 1984    4   13    0    0    0 0.68 0.97    1    1    0    0    0    0 4.37 0.92    0
 1984    4   14    0    0    0 0.67 0.96    1    1    0    0    0    0 7.77 4.32 0.59
 1984    4   15    0    0    0 0.66 0.96    1    1    0    0    0    010.62 7.17 3.44
 1984    4   16    0    0    0 0.65 0.96    1    1 0.52 0.52 0.52 0.5210.22 6.77 3.04
 1984    4   17    0    0    0 0.63 0.95    1    1    0    0    0    0 6.22 2.77    0
 1984    4   18    0    0    0 0.62 0.95    1    1    0    0    0    0 6.67 3.22    0
 1984    4   19    0    0    0 0.61 0.95    1    1 0.54 0.54 0.54 0.54 7.42 3.97 0.24
 1984    4   20    0    0    0  0.6 0.94    1    1 0.12 0.12 0.12 0.12 6.87 3.42    0
```

Figure 4. (a) Example of input file for the original (MS-DOS) SRM. Memory limitations dictated a somewhat cryptic structure, best entered and edited within the SRM program. (b) Example of input file for SRM as implemented in MMS. The expandable column format allows file exchange with spreadsheet programs.

stand-alone SRM, such analysis would require repeated manual changes to on-screen input and manual processing of multiple results.

Modularizing SRM on the basis of parameters may eventually help reduce model error. Within the MMS, replacing a parameterized module with an alternative non-parameterized approach will reduce the number of model parameters, which can result in a reduction in model error (Jeffers 1979).

Migrating the SRM into the MMS explored many of the steps within the first phase of a SA (Model conceptualization/Modularization). Because this project consisted of analyzing an existing model, many of the steps within the second phase of a SA (Quantitative model building) had already been completed (process equations, parameter estimation). At this stage of the project, Phases 3 and 4 must be completed. Future work will involve:

1. Model analysis at the sub-module scale that will show the relationships between and among variables and parameters.
2. Parameter estimation using GIS tools within the MMS.
3. Sensitivity analysis of the SRM modules using the MMS tools.
4. Model use, applications, and testing.

Eventually, our goal is to identify the strongest components of the SRM and similar existing models and judge whether they can be blended within the systems framework to build more accurate and useful models.

CONCLUDING REMARKS

The goal of this project was to implement the widely used stand-along SRM within MMS, which facilitates a systems approach to model building within a single software environment. The advantages of model building using a system like MMS include a deeper insight into both the model and the physical system that it is meant to represent. MMS allows model developers to concentrate on the modeled physical processes while not having to worry about re-inventing the "GUI wheel." Through its emphasis on modularization, modeling within MMS promotes collaborative model building and reduces the time and money spent building models by allowing developers to interchange new or pre-conceived process algorithms within their own models. Unlike users of stand-alone packaged models, the MMS modeler is not tied to another model builder's conceptualization of the system dynamics.

The SA approach supported by MMS ensures a systematic approach to model building and applies the scientific rigor that simulation models should go through before being used. Although many current or older packaged models probably go through a SA type investigation, the process is often poorly documented. The results of this project show that the resources invested in implementing the SRM within the MMS were well spent. This type of project can only help move us towards our common goal of a more profound understanding of simulation models and the underlying physics they represent.

ACKNOWLEDGEMENTS

This research is supported by the National Science Foundation under Grant No. CMS9733925. Mr. Ralph Roberts at the ARS Hydrology Laboratory kindly provided the original SRM source code and advice.

REFERENCES

Acock, B. and Reynolds, J.F. 1997. "Introduction: Modularity in Plant Models," *Ecological Modelling*, **94:(1)**, 1-6.

Austin, M.A. and Frankpitt, B.A. 1998. *Part 1 of Lecture notes for ENSE 621*, Institute for Systems Research, University of Maryland. 174 pp.

Brubaker, K.L., A. Rango and W. Kustas, 1996. "Incorporating Radiation Inputs into the Snowmelt Runoff Model," *Hydrological Processes* 10:1329-1343.

Brüsch, W., 1996. *SRM-ETH User's Manual, Technical Report No. 160*, Computer Vision Laboratory – Institute for Communication Technology, Eidgenössische Technische Hochschule Zürich (ETHZ); http://www.vision.ee.ethz.ch/cgi-bin/create_abshtml.pl?93 (accessed June 1999)

Chapra, S.C. 1997. *Surface-Water-Quality Modeling*, McGraw-Hill, New York. 844 pp.

Grant, W.E. 1998. *Ecology and Natural Resource Management: Reflections from a Systems Perspective*, Wiley, New York. 115 pp.

Haefner, J.W. 1996. *Modeling Biological Systems Principles and Applications*, Chapman and Hall, New York. 473 pp.

Jeffers, J.N.R. 1979. *An Introduction of Systems Analysis with Ecological Applications,* Edward Arnold Limited, Maryland. 198 pp.

Kleindienst, H., 1999, "SRM-Java Documentation," http://saturn.unibe.ch/rsbern/research/topic/srm/srmjava12/index.html (accessed June 1999)

Leavesley, G.H., Lichty, R.W., Troutman, B.M., and Saindon, L.G., 1983, *Precipitation- runoff modeling system: User's manual: U.S. Geological Survey Water Resources Investigations Report 83-4238*, 207 pp.

Leavesley, G.H., Restrepo, P.J., Markstrom, S.L., Dixon, M., and Stannard, L.G., 1996, *The Modular Modeling System (MMS): User's Manual: U.S. Geological Survey Open File Report 96-151*, 142 pp.

Markstrom, S., 1998(a). Modular Modeling System (MMS), U.S. Geological Survey, http://wwwbrr.cr.usgs.gov/mms/ (accessed Feb.1999)

Markstrom, S., 1998(b). "The Modular Modeling System (MMS) – A Modeling Framework for Multidisciplinary Research and Operational Applications," http://wwwbrr.cr.usgs.gov/mms/html/mms_paper.html (accessed June 1999).

Martinec, J., Rango, A., and Roberts, R. 1994. *Snowmelt Runoff Model (SRM) User's Manual.* Geographica Berensia, P29, Department of Geography, University of Bern. 65 pp.

Maxwell T. "A Meta-Model Approach to Modular Simulation", http://kabir.cbl.umces.edu/SME3/MetaModelsF.html (accessed Feb. 1999)

Neelamkavil, F. 1987. *Computer Simulation and Modeling*, John Wiley & Sons, Chichester. 77 pp.

Raymond, E., ed., 1996. "The On-line Hacker Jargon File, version 4.0.0", http://www.tuxedo.org/~esr/jargon/jargon-toc.html (accessed Feb. 1999)

University of Berne (Switzerland), 1998, "4th Snowmelt Runoff Model (SRM) Workshop," http://saturn.unibe.ch/remsen/event/srm98_final.html (accessed June 1999).

USDA-ARS Hydrology Laboratory, 1998, "SRM Home Page," http://hydrolab.arsusda.gov/cgi-bin/srm-home

Viger, Roland, 1999, "The GIS Weasel: An Interface for the Treatment of Spatial Information in Modeling," http://wwwbrr.cr.usgs.gov/weasel/ (accessed October 1999).

Wilby, R.L., ed., 1997. *Contemporary Hydrology*, John Wiley & Sons, West Sussex. 354 pp.

Ziegler, B.P. 1978. 'What is modelling and simulation methodology?', in Ziegler, Elzas, Klir, Oren, Eds., *Methodology in Systems Modelling and Simulation*, North - Holland Pub., Amsterdam. 537 pp.