



*The Society for engineering
in agricultural, food, and
biological systems*

An ASAE Meeting Presentation

Paper Number: 052011

Assessing the Potential of the Object Modeling System (OMS) for Erosion Prediction Modeling

James C. Ascough II, Research Hydrologic Engineer

Olaf David, Research Associate

Lajpat R. Ahuja, Supervisory Soil Scientist

USDA-ARS-NPA, Great Plains Systems Research Unit, 2150 Centre Ave., Bldg. D, Suite 200
Fort Collins, Colorado 80526 Contact E-mail: jim.ascough@ars.usda.gov

Dennis C. Flanagan, Research Hydraulic Engineer

USDA-ARS-MWA, National Soil Erosion Research Laboratory
275 South Russell St., West Lafayette, IN 47907-2077

**Written for presentation at the
2005 ASAE Annual International Meeting
Sponsored by ASAE
Tampa Convention Center
Tampa, Florida
17-20 July 2005**

ABSTRACT. *Current challenges in soil erosion research have created demand for integrated, flexible, and easily parameterized sediment transport models. Most of the existing monolithic erosion models (e.g., WEPP and WEPS) are not modular, thus modifications require considerable time, effort, and expense. In this paper, the feasibility and challenges of using the Object Modeling System (OMS) for soil erosion model development will be explored. The OMS is a Java-based modeling framework that facilitates simulation model development, evaluation, and deployment. We present application of a fully restructured and modularized core WEPP hillslope erosion component functioning within the OMS as a single compartmentalized erosion module. In addition, we discuss specific features of the OMS related to soil erosion modeling including: 1) how to reduce duplication of effort in wind and water erosion modeling; 2) how to make soil erosion models easier to build, apply, and evaluate, 3) how to facilitate long-term maintainability of soil erosion models; and 4) how to improve the quality of soil erosion model code and ensure credibility of model implementations.*

Keywords. *Soil erosion, Object Modeling System, OMS, Modularity, Modeling framework.*

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of the American Society of Agricultural Engineers (ASAE), and its printing and distribution does not constitute an endorsement of views which may be expressed. Technical presentations are not subject to the formal peer review process by ASAE editorial committees; therefore, they are not to be presented as refereed publications. Citation of this work should state that it is from an ASAE meeting paper. EXAMPLE: Author's Last Name, Initials. 2005. Title of Presentation. ASAE Paper No. 05xxxx. St. Joseph, Mich.: ASAE. For information about securing permission to reprint or reproduce a technical presentation, please contact ASAE at hq@asae.org or 269-429-0300 (2950 Niles Road, St. Joseph, MI 49085-9659 USA).

Introduction

The problems facing both developers and users of soil erosion models (and natural resource models in general) are becoming increasingly complex. Tremendous progress has been made in discovering basic principles in different scientific disciplines that created major advances in management and technology for natural resource systems. However, understanding natural resource management issues related to soil erosion (wind and water), hydrology (water management), and farming practices (fertilizer and chemical application) become compounded when viewed within the physical, biological, chemical, and geological responses of the natural world. Computer simulations for prediction and management of soil erosion on agricultural fields, farms, and watersheds have also increased in complexity. The multidisciplinary nature soil erosion assessment usually requires accounting for a significant number of different models (e.g., RUSLE, RUSLE2, WEPP, and WEPS), data sources, management alternatives, and customers/stakeholders.

It can be argued that achieving the goal of sustainable natural resource management should involve consideration of whole system effects. Unfortunately, most natural resource systems involve highly complex interactions of soil-plant-weather-management components that are extremely difficult to quantitatively describe. Thus, state of the art challenges in optimal management of the natural resources have created demand for integrated, flexible and easy to use modeling tools which are able to simulate the quantitative and qualitative aspects of the system (e.g., wind and water erosion) with a sufficient degree of certainty. Although a myriad of soil erosion prediction tools are available, they are typically constrained to the specific scales and purposes they have been developed for and therefore are more robust in some areas than others (depending on the primary goal guiding their development). Furthermore, most of these monolithic models are not modular; are very difficult to update, add to, or connect with other models; have diminishing technical support; and lack the flexibility to meet current needs for more integrated analysis of changing natural resource issues.

All of the above reasons indicate a need for a new framework of soil erosion model development that can integrate existing and future models (or model components) into a common, collaborative, and flexible system. Such a system will maintain modularity, reusability, and inter-operability or compatibility of both science and auxiliary components. The system will also recognize the fact that different categories of applications may require different levels of scientific detail and comprehensiveness, as driven by problem objectives, scale of application, and data constraints. These functionalities of the system will be obtained by establishing standard libraries of interoperable science and auxiliary components or modules that provide the building blocks for a number of similar applications. Module libraries have been successfully used in several domains, such as the manufacturing, transport, and other systems (Breunese et al., 1998; Praehofer, 1996). One of the earliest modular model developments was done for SHE, the European Hydrologic System Model (Abbot et al., 1986). Leavesley et al. (1996) reported the conversion of the Precipitation Runoff Modeling System (PRMS) to a Unix-based Modular Modeling System (MMS) for hydrologic modeling. Leavesley et al. (2002) presented some successful applications of this concept. Other examples of model integration framework initiatives include the Interactive Component Modelling System (Rahman et al., 2004), Tarsier (Watson and Rahman, 2004), Spatial Modeling Environment (Maxwell and Costanza, 1995, 1997), and HarmonIT (Blind and Gregersen, 2004). To summarize, an approach for erosion prediction modeling is needed that will:

- Reduce duplication of development effort, and improve the quality and currency of model code;
- Make soil erosion models much easier to build, access, understand, and use;

- Facilitate long-term maintainability of existing and new soil erosion models;
- Lead to greater consistency of modeling for particular problems and scales;
- Improve response and delivery times in scientific modeling projects;
- Ensure creditability and security of model implementations; and
- Function on any major computing platform.

The Object Modeling System (OMS) being developed by the USDA-ARS Great Plains Systems Research Unit (Fort Collins CO, USA), USDA-NRCS Information Technology Center (Fort Collins CO, USA), and the USGS (Denver CO, USA) meets the above criteria. The OMS provides a modular modeling framework which allows the implementation of single- or multi-process modules which can be compiled and applied as custom-tailored model assemblies.

Object Modeling System (OMS) Description

Overview

The OMS project was initiated in 1996 at the Friedrich Schiller University of Jena. In October 2000, the OMS evolved into an interagency project between the USDA-ARS, USGS, and USDA-NRCS, with financial support from ARS and in-kind support from the partners. During the past 36 months, the OMS programming team has completed the development of most of the core components of the OMS. The OMS vision, described below for initial ARS implementation, is close to being realized (Ahuja et al., 2002):

“The OMS is a computer framework consisting of: 1) a library of science, control, and database modules; 2) a means to assemble the selected modules into a modeling package customized to the problem, data constraints, and scale of application; 3) automatic generation of a user-friendly interface; and 4) creation of a compiled, ready-to-run, version of the package. The framework is supported by utility modules such as data dictionary, data retrieval, GIS, graphical visualization, and statistical analysis. The framework employs the latest Java-based software technology for all its components. The science modules are also quickly updated or replaced as new knowledge and data become available. The OMS will be supported from a central server for use by all ARS scientists, NRCS specialists, USGS, and other collaborators.”

The OMS is built on top of the NetBeans platform. The NetBeans platform is a framework for building desktop application software in the Java programming language. The OMS leverages NetBeans features such as user interaction components (e.g., menu bars, tool bars, status displays, tabbed-window displays, etc.), storage access components, and help components (e.g., JavaHelp). A schematic of OMS implementation for natural resource modeling in general is presented in Figure 1.

Component Architecture

The general objectives of the OMS project included the development of generic software tools to extract modules from existing non-modular simulation models, and to incorporate them into the OMS framework with standard OMS descriptions. These tools have been developed, but need further testing and improvement. The OMS framework has the following functional components that are currently operational:

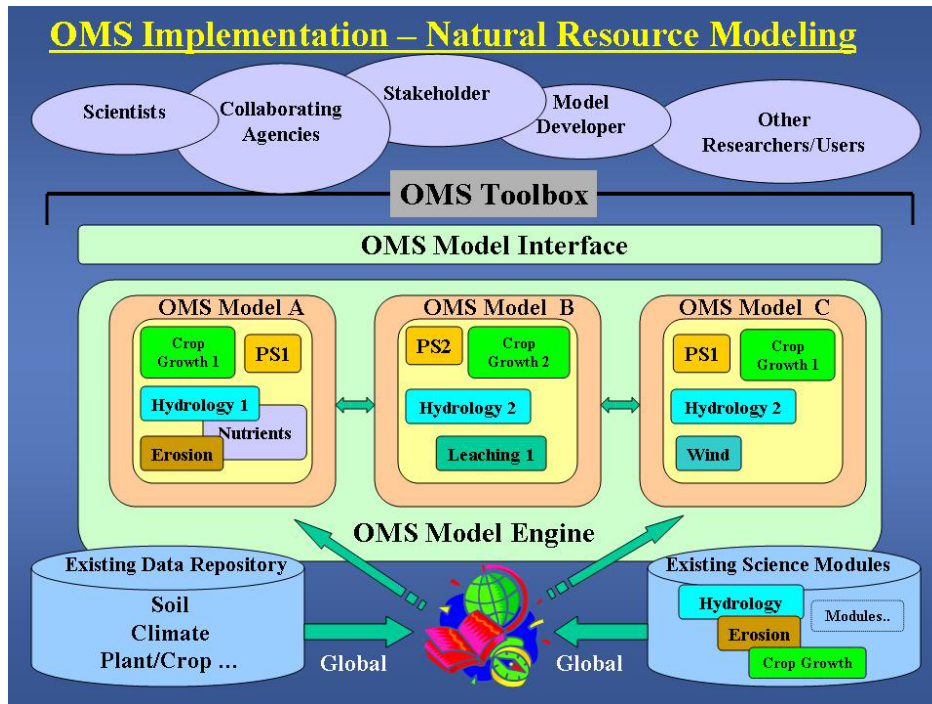


Figure 1. Schematic of OMS implementation for natural resource modeling (including soil erosion) customized for users.

1. A module-building component that facilitates the integration of existing (legacy) code into the framework.
2. A module repository containing modules that can be readily utilized to assemble a working model (types of modules in the library will include science, control, utility, assessment, data access, and system modules).
3. A model builder that assembles modules from the module library into executable models and verifies data connectivity, and compatibility in scale and comprehensiveness.
4. A dictionary framework that manages extended modeling data type information and provides extended semantics checking for module connectivity verification.
5. An extensible user interface that facilitates an appropriate user interaction for general model development and application (it is supported by a number of contributing software packages for database management, visualization, and model deployment).

The components have the following architecture or characteristics:

1. OMS models are treated as hierarchical assembled components representing building blocks. Components are independent and reusable software units implementing processing objects for simulation models. They reside in a model library and are categorized into data access components, science components, control components, utility components, and system components.
2. The OMS is able to integrate legacy code components. By an automated JAVA wrapper generation for legacy code, components written in languages such as Fortran or C++ can be embedded into the OMS at the function level.
3. The “knowledge-backbone” of the OMS is the dictionary framework. It enables the OMS to verify state variables and parameters according to scientific nomenclatures during model development and application. Dictionaries are also used to specify parameter sets, model control information and the component connectivity. They are implemented

in the Extensible Markup Language (XML).

4. The OMS is extensible, i.e., extension packages exist for different aspects in model development and application. Extension packages are used for visual model assembly, model application, an interface to the dictionary framework, output visualization, and GIS integration.
5. The OMS scales from a full-featured, stand-alone development system with tools for model assembly, visualization, and analysis to a runtime Web service environment.

For a more complete explanation of the OMS framework and architecture, the reader is referred to David et al. (2002).

Developing Soil Erosion Models With the OMS: Advantages And Disadvantages

The following section lists some advantages and disadvantages of developing soil erosion models with the OMS.

Advantages and Feasibility

1. *Efficient transfer of technology:* For the transfer of erosion prediction tools to stakeholders, researchers, and other users, the OMS will serve as a multidimensional platform for integration of a variety of software tools. The end users will then develop deployment links to only the OMS, rather than develop links to each of the separate software tools as is current practice. This will result in a faster transfer of appropriate technology. For example, the USDA-NRCS Information Technology Center (ITC) in Fort Collins CO, USA has been a partner in OMS development, and is committed to using the OMS as a means to provide technical tools to 2,500 field offices for natural resource conservation planning. Finally, due to a common model building platform and a common user interface for all models, the OMS will result in reduced start-up time for model development and lower training costs for users.

2. *Cost reduction in maintenance and customer support for software technology:* It is generally agreed that over the long-term, these items cost up to three times as much as for initial development of the software packages. At present, there are numerous small to large erosion prediction software programs that are currently being maintained and supported. Significant time and labor is being spent on this process since monolithic models are becoming very expensive to use, difficult to update, have diminishing technical support, and lack the flexibility to meet today's needs for more integrated analysis of natural resource issues. These problems can be overcome if many of the existing erosion prediction packages were transferred to the OMS and all new packages were developed within the OMS.

3. *Cost reduction in developing new software technology:* In the past, erosion model development efforts have primarily consisted of large teams of scientists. For example, ARS has had a number of individuals and teams build software technology and simulation models, including the erosion models WEPP and WEPS and the water quality models GLEAMS, AnnAGNPS, and SWAT that all contain erosion prediction components. Each of these packages cost many millions of dollars to develop, including scientist and support time. Development costs were also inflated by significant duplication of work. Model developers can now leverage that investment by putting the science in those packages as modules in the OMS to build new customized software packages at a small fraction of the cost.

4. *Applying the most suitable science for specific problems:* The OMS will allow the selection of the best evaluated and most appropriate science modules currently available depending upon the nature of the problem and required answers, availability of input data,

and scale of application. The OMS library may have different modules for a research model versus a management decision tool. Similarly, a watershed-scale management model may possibly require different (i.e., less complex) science modules than a field-scale model. Issues of dimensionality (e.g., 1-D surface vs. 2-D groundwater), scale, or dynamic interactions (e.g., feedback mechanisms) between modules can be easily handled in the OMS as long as the required module structures and data dependencies are adequately defined.

5. *Assure reliability in results from software tools for similar applications:* Erosion prediction software application users like the USDA-NRCS often report that different software tools or simulation models (e.g., RUSLE2 vs. WEPP) give vastly different results because the tools used different science approaches in key process areas. The OMS will significantly reduce this problem by utilizing evaluated, documented, and standardized modules for the basic science components for a given category of applications.

6. *The OMS library as a reference and coordination tool for future research and development:* The OMS library will be a repository of current, quantitative knowledge in different areas of natural resource system science. Future scientists could look to this library to help determine where further research and development are needed.

7. *Integrated analysis of natural resource system production and conservation issues:* Effective analysis and management of natural resource systems and the environment requires integration of tools and data types that now exist in an array of individual disparate models. The OMS will provide customized, whole-system tools for the analysis of production and conservation issues (e.g., wind and water erosion, environmental quality, and global climate change management) in natural resource systems.

8. *OMS certification mechanism for approved “science building blocks”:* The OMS supports the technical certification of library components based on X.509 Certificates and the validation of such certificates. This will allow agencies or other entities to certify approved modeling components and models. In addition, license protection of modules or components is allowed through signatures within the Netbeans platform.

9. *Enhance productivity of scientists and researchers:* The customized, best-quality, software tools developed through the OMS will help field scientists quantify their results and transfer them to other soils and climates very rapidly. The gaps identified in the process will make future research more focused. Overall, the productivity of scientists and the quality of science should increase as focus centers on science module implementation rather than Graphical User Interface (GUI) design, software deployment, packaging and maintenance.

10. *International coordination in new science module development and publication:* Through the Internet, the OMS will serve as a common platform for international scientists and researchers to contribute their findings as modules to the OMS library. A supervisory group or organization (e.g., International Association of Hydrological Sciences) could coordinate this development, and provide a mechanism for peer review and quality control. The module contribution to the library will be considered a publication by scientists that could have world-wide impact.

Challenges and Difficulties

Challenges to using the OMS for erosion prediction model development stem from the following problems:

- *Lack of motivation to share model code* – in order to fulfill the intended purpose of the OMS, model developers must build a repository of modules through contribution to the OMS module library.

- *Acceptance of a modular coding structure* – model developers must spend more “up-front” time in module development in terms of module structure, I/O requirements, metadata description, etc.
- *Willingness to share data sets for a range of erosion processes covering different scales and climatic/physiographic regions across the world* – application of erosion prediction models developed under the OMS will be difficult without data sets for comparison and evaluation.
- *Loss of model name recognition* – this can be overcome, however, by the development of a mechanism for peer review and quality control of individual modules.

OMS Integration Example: WEPP Hillslope Component

The OMS utilizes Component Oriented Programming (COP) for the creation and execution of simulation models. Components in general take the idea of object-orientation to a next level. While object-oriented design methods support abstraction and localization of data and methods, they can also lead to simulation systems where objects are co-dependent. To remove that limitation, a more rigid idea was introduced in the OMS: the component. The key difference between a standard object and a component in the OMS is that a component is completely replaceable and provides its behavior by having a passive structure. This means that the simulation component is always called from the system, as opposed to the component calling some system functionality. This approach provides the highest degree of flexibility from a modeling framework point of view. This section presents a brief example of integrating the WEPP hillslope erosion model (Flanagan et al., 2001), comprised of small well-defined components, within the OMS.

WEPP Hillslope Erosion Model Overview

In WEPP hillslope profile simulations, the model separates the upland erosion processes into those caused by excess flow shear detachment in rill channels, and interrill detachment caused by raindrop impact and shallow rain-impacted flows. The soil erosion computations in the WEPP hillslope profile erosion component use solutions to a steady state sediment continuity equation to estimate values of sediment load and net detachment or deposition at points down a profile. The governing equation is:

$$dG/dx = D_f + D_i \quad (1)$$

where G is sediment load in $\text{kg}\cdot\text{s}^{-1}\cdot\text{m}^{-1}$, x is distance downslope in m , D_f is rill erosion rate in $\text{kg}\cdot\text{s}^{-1}\cdot\text{m}^{-2}$, and D_i is interrill sediment delivery rate in $\text{kg}\cdot\text{s}^{-1}\cdot\text{m}^{-2}$ (Foster et al., 1995).

Interrill sediment delivery to rills is predicted in WEPP using the equation:

$$D_i = K_{iadj} I_e \sigma_{ir} \text{SDR}_{RR} F_{nozzle} (R_s/w_{rill}) \quad (2)$$

where K_{iadj} is the adjusted interrill erodibility factor in $\text{kg}\cdot\text{s}\cdot\text{m}^{-4}$, I_e is effective rainfall intensity in $\text{m}\cdot\text{s}^{-1}$, σ_{ir} is the interrill runoff rate in $\text{m}\cdot\text{s}^{-1}$, SDR_{RR} is a sediment delivery ratio that is a function of random roughness, row side-slope and the interrill particle size distribution, F_{nozzle} is an adjustment factor to account for sprinkler irrigation nozzle impact energy variation (value of 1.0 for natural rainfall conditions), R_s and w_{rill} are the respective rill spacing and rill width in m .

Effective rainfall intensity is estimated during the infiltration computations using the equation:

$$I_e = (1/t_e) \int I dt \quad (3)$$

where I_e is the effective rainfall intensity in $\text{m}\cdot\text{s}^{-1}$, I is breakpoint rainfall intensity in $\text{m}\cdot\text{s}^{-1}$, t is time in seconds, t_e is the total time over which rainfall rate exceeds infiltration rate, and the integral is evaluated over time t_e .

The interrill sediment delivery ratio, SDR_{RR} , is determined through three steps. First, an interrill roughness factor is calculated based upon a functional representation of Table 8.4 in Foster (1982) using the equation:

$$RIF = -23 RR + 1.14 \quad (4)$$

where RIF is the interrill roughness factor and RR is random roughness in m of the current overland flow element. Random roughness values used in this function are limited between 0.0061 and 0.0496 m (RIF values of 1.0 to 0.0, respectively).

The second step of the procedure is to compute a delivery ratio (DR_i) for each of the WEPP particle size classes using the interrill roughness factor and the fall velocity of each size class. The five WEPP particle size classes, predicted using relationships from Foster et al. (1985b) are primary clay, primary silt, primary sand, small aggregate and large aggregate. For particles with fall velocities (v_{fi}) less than $0.01 \text{ m}\cdot\text{s}^{-1}$ the relationship used is:

$$DR_i = az (RIF)^{bz} \quad (5)$$

where

$$az = e^{(0.0672 + 6.59 v_{fi})} \quad (6)$$

$$bz = 0.1286 + 2209 v_{fi} \quad (7)$$

and for particles with fall velocities greater than or equal to $0.01 \text{ m}\cdot\text{s}^{-1}$ the relationship used is:

$$DR_i = 2.5RIF - 1.5 \quad (8)$$

The subscript i represents the individual particle size class, and delivery ratio values are constrained to between 0 to 1 within the model.

The final step to determine the interrill sediment delivery ratio is to take a weighted average of the sediment delivery ratio for each particle size class, weighted by the mass fraction of sediment in each class:

$$SDR_{RR} = \sum f_{deti} (DR_i) \quad (9)$$

from $i=1$ to 5, where f_{deti} is the fraction of each size class predicted with the Foster et al. (1985b) equations. Additionally, the fraction of sediment in each size class that is delivered from the interrill areas to the rills ($fideli$) is calculated as:

$$fideli = f_{deti} (DR_i / SDR_{RR}) \quad (10)$$

These values are used to update the flow sediment size classes at the end of each detachment region and start of each deposition region in the rills.

Rill erosion rate may be either positive in the case of detachment or negative in the case of deposition. Rill detachment in WEPP is predicted when the flow sediment load is below transport capacity, and flow shear stress acting on the soil exceeds critical shear stress. In that case, D_f is predicted with:

$$D_f = K_{radj} (\tau - \tau_{cadj}) [1 - (G/T_c)] \quad (11)$$

where K_{radj} is the adjusted rill erodibility factor in $\text{s}\cdot\text{m}^{-1}$, τ is flow shear stress in Pa, τ_{cadj} is adjusted critical shear stress of the soil in Pa, G is sediment load in the flow ($\text{kg}\cdot\text{s}^{-1}\cdot\text{m}^{-1}$), and T_c is flow sediment transport capacity in $\text{kg}\cdot\text{s}^{-1}\cdot\text{m}^{-1}$. Sediment transport capacity is computed using a simplified function of shear stress raised to the 3/2 power, times a coefficient that is determined through application of the Yalin (1963) equation at the end of the slope profile (Finkner et al., 1989).

Deposition in rills is predicted when flow sediment load exceeds sediment transport capacity. In this case the model predicts the rill erosion rate using:

$$D_f = (\beta v_{\text{eff}} / q) [T_c - G] \quad (12)$$

where β is a raindrop-induced turbulence factor, v_{eff} is an effective fall velocity for the sediment in $\text{m}\cdot\text{s}^{-1}$, q is flow discharge per unit width in $\text{m}^2\cdot\text{s}^{-1}$, and T_c and G are as previously defined. Currently, β is assigned a value of 0.5 for rain-impacted flows, and a value of 1.0 for other cases such as snow melt or furrow irrigation erosion.

WEPP solves the sediment continuity equation for total sediment load using normalized equations at 100 points down each overland flow element. In cases in which sediment load is below transport capacity, a Runge-Kutta numerical method is used to solve for sediment load at progressive points down an OFE using Equations 1, 2, and 11. When sediment load exceeds transport capacity, a closed-form solution is used with Equations 1, 2, and 12.

The erosion component of WEPP also contains equations that are used to estimate the sediment particle sorting in rill channels due to selective deposition. An analytic solution of the normalized sediment continuity equation is used to estimate the particle size distribution reaching the end of a deposition region, assuming that the total sediment load computed previously with Equation (12) is correct. See Foster et al. (1995) and Flanagan and Nearing (2000) for details on the procedure.

Integration Process

As previously stated, creating highly integrated natural resource simulation models using traditional programming methods can be expensive and time-consuming. This section shows a simple example of using the OMS to create a WEPP-based hillslope erosion prediction module utilizing component-oriented methods coupled with process-based sub-modules. That is, the WEPP V. 2004.7 hillslope erosion component was disaggregated into smaller sub-components which were then reassembled into a functionally identical model within the OMS. A challenge of monolithic model disaggregation is deciding at which point in the original code it is reasonable to extract a new section of code into a singular executable component. This extraction process is typically iterative with the intended result being a module containing a level of complexity that makes sense for the intended use. In other words, module size and complexity should be determined by functional use, e.g., modules consisting of single algorithms may be appropriate or an entire monolithic model may function as a module.

Figure 2 shows the OMS interface with open windows of the various components required to create an OMS-based WEPP hillslope erosion model. These include the project window, component library window, declared variable attribute window, and the model output window. The basic steps for creating the new erosion model, i.e., model integration, in OMS are:

1. Create the component(s) (as defined above) using the “Make/Build” button in the “Modeling Projects” Project Manager.
2. Publish the components into the Component Library by pressing the “Add to Component Library” button.
3. Create a new model within a workspace using the “Templates | Object Modeling System | Model” template.
4. Drag and drop a component from the Component Library into the model.
5. Compile the component or group of components.
6. Run the model.

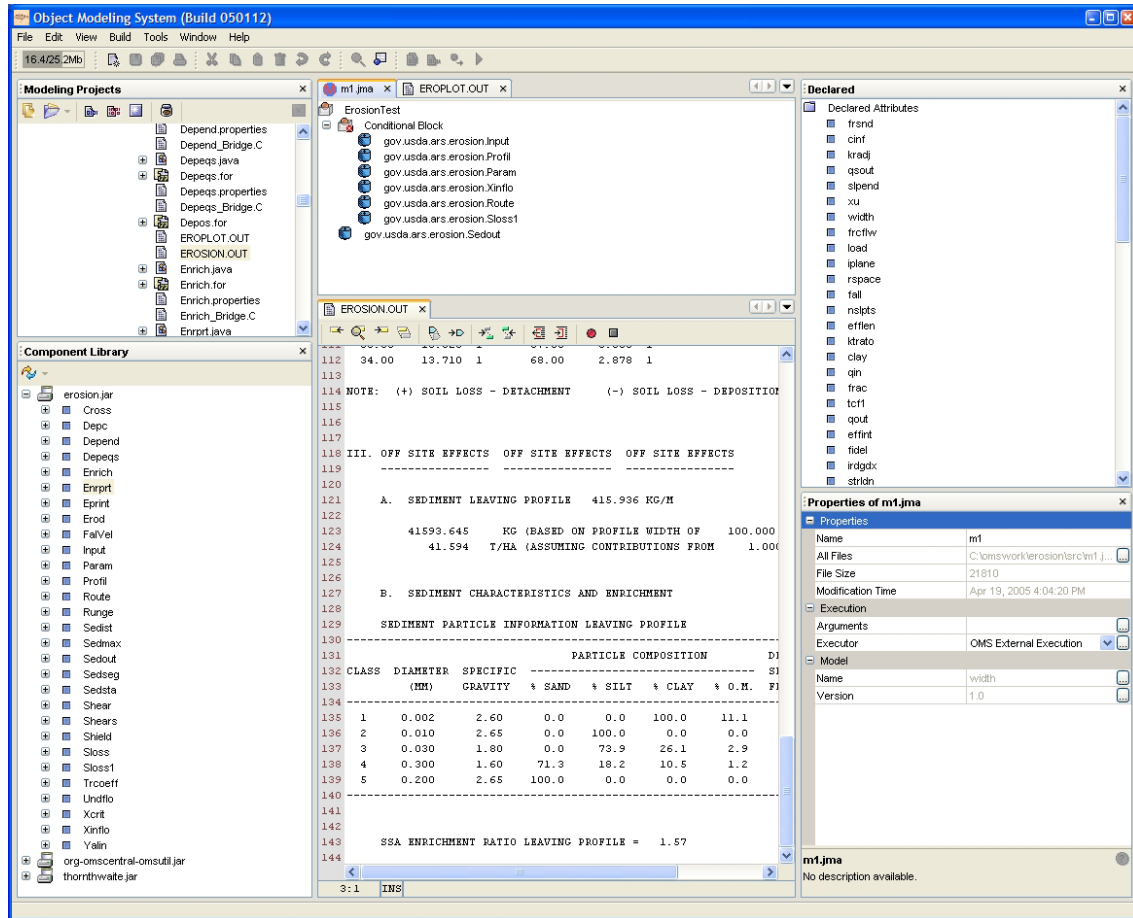


Figure 2. Object Modeling System (OMS) Interface

Conclusion

A large number of complex erosion prediction models are currently in use worldwide. Most of these models are based on sound science, but have specialized data requirements and significant duplication exists in many areas. The current OMS development tool will leverage these sizeable investments of developer time and money to: 1) facilitate an interdisciplinary effort extracting the best scientific routines of existing models; and 2) provide integration and interoperability of existing and new scientific modules and modern data resources. To summarize, principal advantages of the OMS include:

1. The OMS will increase the probability of using the best science available in various combinations for the given conditions and problem.
2. The OMS will be easier to maintain and update as new knowledge, data and technology become available. The OMS will allow a “select, plug, and play” mechanism for modules consisting of sub-models, equations, graphics, statistics, risk analysis, parameter estimation, standard data sources and various reporting formats.
3. New knowledge expressed in the form of modules will be relatively easy to verify and evaluate (and possibly lead to scientific peer review).
4. The OMS will help to eliminate duplicate functionality across erosion prediction models. The OMS library of modules will serve as a reference and a coordination mechanism for

future improvements. The OMS will facilitate communication between model developers by providing a common standard for development and implementation.

5. The OMS should significantly reduce the problem of different erosion prediction models giving different results by utilizing a library of evaluated, documented, and standardized modules and integrated output options.
6. The OMS will provide a consistent interface for model creation and evaluation, and will reduce startup time for scientific users and developers.
7. The OMS offers support-ready compliance with the distinct advantage of having a set of application packages (e.g., data input, parameterization, visualization) under a single user interface or a set of consistent user interfaces.
8. The OMS will allow flexibility to choose scientific modules most appropriate for the scale or region of interest, or to respond to other unique influencing factors under consideration.

In conclusion, the component-oriented and modular approach of the OMS and the modules/models implemented in it will provide the basis for more efficient and internationally collaborative model development in the future. This type of integrative and open-source approach is desperately needed in order to solve global challenges impacting natural resource systems such as sustainable management and the impact of global climate change. For more details on the OMS project mission, project documentation, or to download the entire application or individual modules, visit the OMS web site at <http://oms.ars.usda.gov/>.

Acknowledgements

The authors wish to thank Ian Schneider for his programming support on OMS core model development. We also wish to thank Ken Rojas and Frank Geter of the USDA-NRCS for continued testing of the OMS core model. Finally, we deeply appreciate the continued support of the OMS project by Jack Carlson of the USDA-NRCS Information Technology Center, Fort Collins CO, USA.

References

- Ahuja, L.R., David, O., Blackburn, W., Amerman, R., Werner, J., Carlson, J., Knighton, R., and Leavesley, G. 2002. The Object Modeling System (OMS): An advanced object-oriented, modular modeling computer technology for agricultural production systems. USDA-ARS Working White Paper, Vers. 1, Fort Collins, CO.
- Abbot, M.B., Bathurst, J.C., Cunge, J.A., O'Connell, P.E., and Rasmussen, J. 1986. An introduction to the European Hydrological System-Systeme Hydrologique Europeen, "SHE", 2: structure of a physically-based, distributed modelling system. *J. Hydrol.* 87:61-77.
- Blind, M. and Gregersen, J.B. 2004. Towards an Open Modelling Interface (OpenMI) The HarmonIT project. In: Complexity and Integrated Resources Management, (Eds) Pahl-Wostl, C., Schmidt, S., Rizzoli, A.E., and Jakeman, A.J., Transactions of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs). Manno, Switzerland, ISBN 88-900787-1-5, pp. 346-351.
- Breunese, A.P.J., Top, J.L., Broenink, J.F., and Akkermans, J.M. 1998. Libraries of reusable submodels: theory and application. *Simulation* 77(1) (July): 7-22.
- David, O., Markstrom, S.L., Rojas, K.W., Ahuja, L.R., and Schneider, I.W. 2002. The Object Modeling System. In: Agricultural System Models in Field Research and Technology Transfer, (Eds) Ahuja, L.R., Ma, L., and Howell, T.A., Lewis Publishers, pp 317-330.

- Finkner, S.C., Nearing, M.A., Foster, G.R., and Gilley, J.E. 1989. A simplified equation for modeling sediment transport capacity. *Trans. Am. Soc. Agric. Eng.* (32):1545-1550.
- Flanagan, D.C. and Nearing, M.A. 2000. Sediment particle sorting on hillslope profiles in the WEPP model. *Trans. Am. Soc. Agric. Eng.* (43): 573-583.
- Flanagan, D.C., Ascough II, J.C., Nearing, M.A., and Laflen, J.M. 2001. Chapter 7: The Water Erosion Prediction Project (WEPP) model. In: R.S. Harmon and W.W. Doe III (Eds.), *Landscape Erosion and Evolution Modeling*. Kluwer Academic/Plenum Publishers, New York, NY. p. 145-199.
- Foster, G.R. 1982. Chapter 8: Modeling the erosion process. In: *Hydrologic Modeling of Small Watersheds*, C.T. Haan (Ed.), ASAE Monograph No. 5, American Society of Agricultural Engineers, St. Joseph, MI, pp. 295-380.
- Foster, G.R., Young, R.A., and Neibling, W.H. 1985. Sediment composition for nonpoint source pollution analyses. *Trans. Am. Soc. Agric. Eng.* (28):133-139.
- Foster, G.R., Flanagan, D.C., Nearing, M.A., Lane, L.J., Risse, L.M., and Finkner, S.C. 1995. Chapter 11: Hillslope erosion component. In: *USDA-Water Erosion Prediction Project: Hillslope Profile and Watershed Model Documentation*. D.C. Flanagan and M.A. Nearing, Eds.), NSERL Report No. 10, USDA-ARS National Soil Erosion Research Laboratory, West Lafayette, IN.
- Leavesley, G.H., Restrepo, P.J., Markstrom, S.L., Dixon, M., and Stannard, L.G. 1996. *The Modular Modeling System (MMS): User's Manual*, Open File Report 96-151, USGS, Denver, Colorado.
- Leavesley, G.H., Markstrom, S.L., Restrepo, P.J., and Viger, R.J. 2002. A modular approach to addressing model design, scale and parameter estimation issues in distributed hydrological modeling. *J. Hydrol. Proc.* (16):173-187.
- Maxwell, T. and Costanza, R. 1995. Distributed modular spatial ecosystem modeling. *Int. J. Comp. Simul.: Special Issue on Advanced Simulation Methodologies* 5(3):247-262.
- Maxwell, T. and Costanza, R. 1997. A language for modular spatio-temporal simulation. *Ecol. Modell.* 103(2,3):105-114.
- Praehofer, H. 1996. Object oriented, modular hierarchical simulation modeling: towards reuse of simulation code. *Simul. Pr. Theory* 4(4):221-234.
- Rahman, J.M., Seaton, S.P., and Cuddy, S.M. 2004. Making frameworks more useable: using model introspection and metadata to develop model processing tools. *Env. Modell. & Soft.* 19(2004):275-284.
- Watson, F.G.R and Rahman, J.M. 2004. Tarsier: a practical software framework for model development, testing, and deployment. *Env. Modell. & Soft.* 19(2004):245-260.
- Yalin, Y.S. 1963. An expression for bedload transportation. *J. Hydraul. Div., Proc. ASCE*, 89(HY3): 221-250.