Improving Lake Winnipeg Integrated Environment Modelling with OpenMI

W.G. Booty[1], I.W. Wong, R.C. McCrimmon, D.C.-L. Lam & P. Fong
Canadian Centre for Inland Waters, Environment Canada

Abstract
We have applied OpenMI standards for two lake models applied to Lake Winnipeg to improve interactions among models and model calibration. These two models are OneLay and PolTra, which combine to form a 2-D horizontal, vertically mixed lake model. The source codes for the two models were modified for OpenMI to enable data exchange at each time step. Our approach is to calibrate drag coefficient and bottom coefficients in the OneLay model and settling coefficient, resuspension critical shear velocity and resuspension coefficients in the PolTra model for total suspended sediment. Model calibration statistic RMSE is used to measure the effectiveness of the calibration and is updated on an on-going basis as the simulation runs. This allows for the model simulations to be stopped partway through when the statistics are not improving, which greatly reduces the calibration time. A modelling database is developed to store key results of the multiple simulation runs from the calibration process. A decision support system with an implementation of a genetic algorithm is being implemented to demonstrate the system can be further automated.

Keywords: integrated environmental modelling, decision support system, OpenMI, artificial intelligence

1. Introduction

Lake Winnipeg is fed by a vast Lake Winnipeg Watershed (Figure 1) covering 960,000 square kilometres extending over four Canadian provinces and three U.S. States. The problems facing the lake are the result of excessive nutrients, phosphorous and nitrogen, from farms and municipal wastewater ending up in the lake. The excessive nutrients are the main contributors to the growth of blue-green algae, which depletes oxygen in the lake, clogs fishing nets, fouls beaches and produces harmful toxins. To help deal with this problem, the Lake Winnipeg Basin Initiative (LWBI) was established by the Canadian government in 2008.

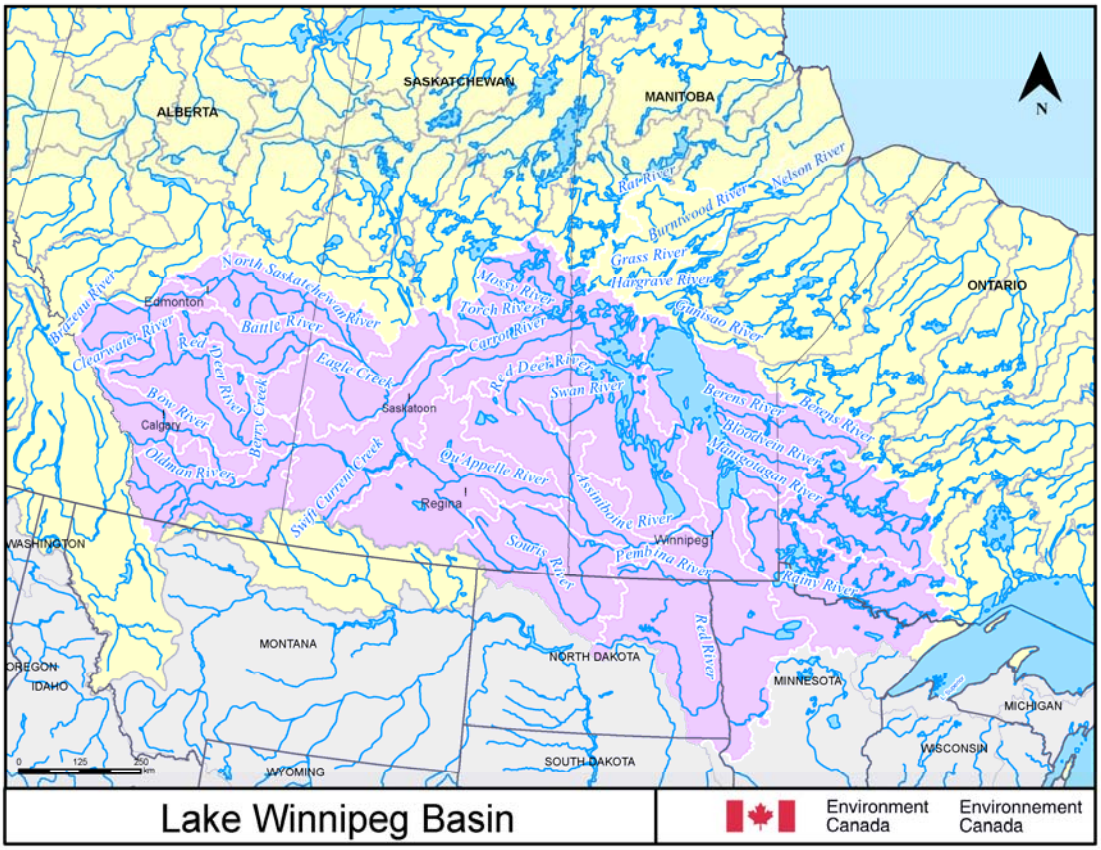

Figure 1 Lake Winnipeg Basin

It is recognized that more research is needed to determine what would constitute sustainable nutrient levels for Lake Winnipeg, the sources of nutrients, and what can be done to manage them throughout the Lake Winnipeg Basin area. The LWBI supports science activities required to understand the relationship between the ecology and nutrient cycle within the lake. Mathematical modelling is a practical and efficient way to reveal critical information about nutrient transport within the basin and the lake.

In order to assess the nutrient loadings or concentrations, we need to understand the lake circulation since it is the driver of the nutrient transport. In our case, a lake circulation model coupled with a lake chemistry model is required. Integrated environmental modelling requires interaction among the models of concern. A typical way to integrate these models is to link them sequentially. Thus, after the completion of the lake circulation model, its results are available entirely and fed into the lake chemistry model. Calibration of the models requires both model runs to be completed and then model results are compared with the observed values. This is a major limitation from the model calibration point of view. If the models can exchange information at each time step, a more desirable approach is to abort the current calibration run once we know that run will not improve the calibration and then revise the coefficients of the calibrating variables for the next iteration. Of course, assigning revisions of coefficients can rely on expert opinions or some artificial intelligence approaches. The advantages of this approach are

to eliminate unnecessary simulation for calibration, to improve the understanding of model interactions and thus, to improve the efficiency of the integrated environmental modelling. Moreover, if we can devise an approach to guide the choice of assigning coefficient values for future iterations, this process can become automated and provide optimal results. In this paper, we propose such an approach.

2. Building Model Interactions using OpenMI

Typically, the traditional approach of running models that interact with each other (i.e., models are linked together so that a model's output is the input of another model) is to run the models independently and in sequence. In this method, for every pair of models that interact, the models' results may have to be reformatted in order for the other model to accept them as input. The nature of the approach also prohibits any two-way interaction as well, that is, the second model cannot be a source of input for the first model (which supplies data to the second model).

Much research has been done to develop techniques that allow model integration to be standardized, tighter and more seamless, which enables the formation of large complex modeling systems from individual models. One such method is the Open Modelling Interface (OpenMI) and is the one used for the modeling work for Lake Winnipeg. It is open source and thus, freely available via the Internet (www.openmi.org). OpenMI (Moore & Tindall, 2005), a linkage mechanism, provides a standard generic interface to allow models of different disciplines and domains to exchange data with each other on a time step by time step basis as they run. Models are able to be run simultaneously (rather than in sequence) and are able to share information as they run (rather than after a model run completes). Specifically, the standard interface's key functions include model component definition to allow others to find out what data this model can provide, configuration to define what exactly gets exchanged when models have been linked and execution to enable a model to accept or provide data during run-time. Metadata plays a role in defining and configuring models. It's used to describe the data that can be exchanged in terms of semantics, units, dimensions and spatial and temporal representations. This information is made available to list what data a particular model can provide and to assist the modeller in creating scientifically valid links between models. OpenMI standardizes the way data exchange is specified and executed.

OpenMI (version 1.4) consists of a suite of software in .NET and Java that implement the standard interface and that help developers in making new and existing models OpenMI-compliant. The steps toward having an OpenMI-compliant model involve modifying the model program to meet the OpenMI specifications and developing a "wrapper" component, which is the communication interface that other components use to access the model. The model program must be first converted into a Dynamic Link Library (DLL) from an ordinary executable to enable other components to access the functions within the model and the model must be changed to allow for a single time step to execute as a separate operation and to allow for data and results to be returned upon request. In addition, an XML document (called an OMI file) must accompany each model

component to define the information needed to initialize the model and to supply the model with the required inputs. Data exchange in OpenMI is based on a request-reply mechanism (i.e., models would “pull” data that they need from other models). The model requiring data makes requests to the model providing the data (via a function call) specifying what data it needs (including the units), for what location and during what time period. The data provider then replies with the appropriate data after performing any necessary spatial, temporal and/or unit conversions.

2.1 The Candidate Models

We identify two candidate models that are suitable to the implementation of OpenMI and to test this concept. These two models are OneLay and PolTra (Simons and Lam, 1986) which combine to form a 2-D horizontal, vertically mixed lake model. The models are based on a rectangular grid representation of a lake. OneLay is the hydrodynamic program that uses lake depths, river inflow/outflow, and wind vector to simulate horizontal currents and water levels. PolTra is the pollutant transport model that uses the bathymetry and water transport computed by OneLay to simulate water and sediment concentrations. Originally all of the hydrodynamic computations were completed before the water quality model started. These programs were written in FORTRAN and designed to run in the MS-DOS environment. These two models were selected because, compared to 3 dimensional models, they are relatively simple and fast running, which is desirable during the preliminary testing stages of the OpenMI wrapper’s development. Also the two models are “in house” models so the source code is readily available and familiar. Figure 2 illustrates the two model system diagram before and after OpenMI implementation.

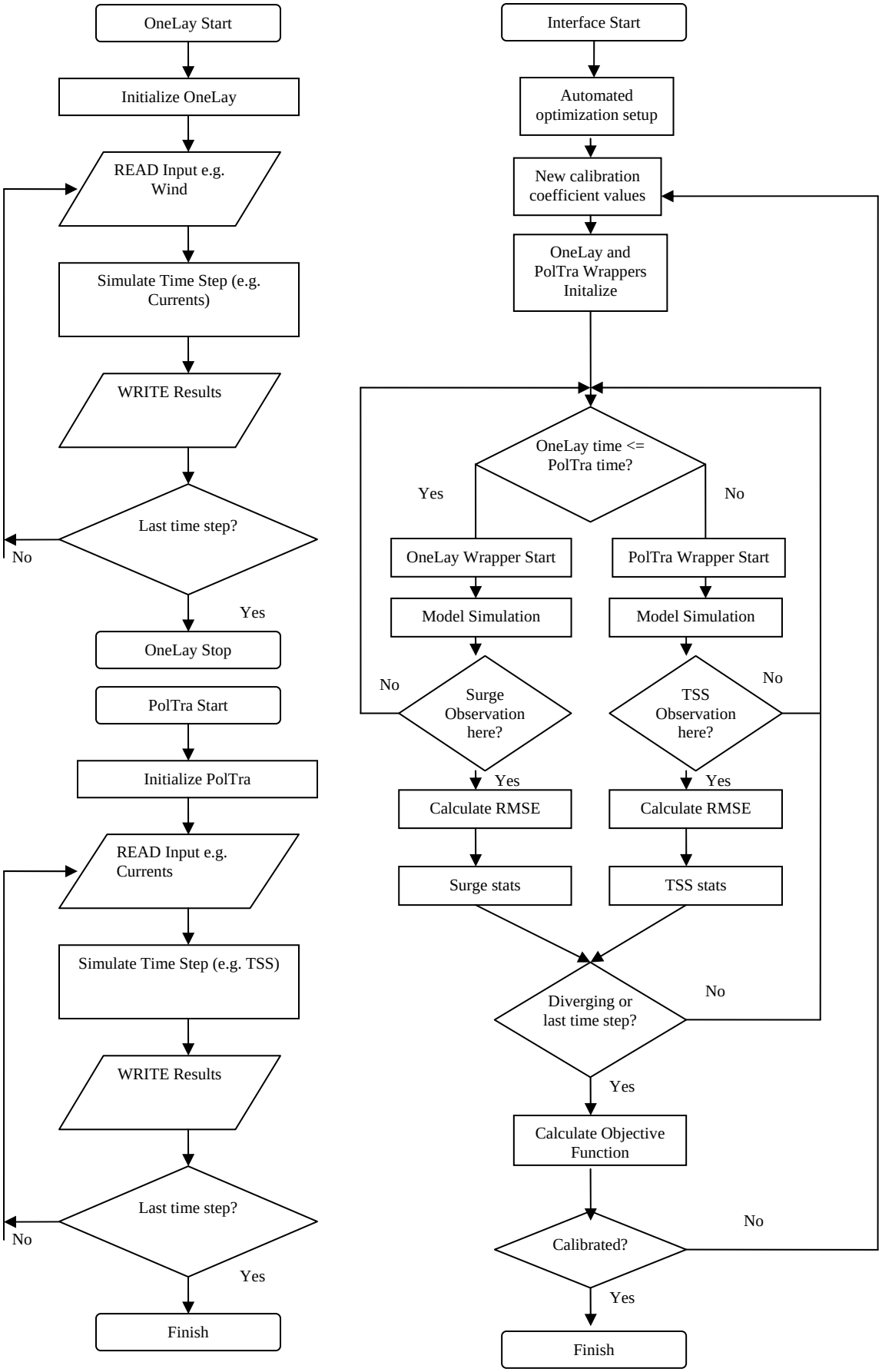
OneLay Start
Initialize OneLay
READ Input e.g. Wind
Simulate Time Step (e.g. Currents)
WRITE Results
Last time step?
No
Yes
OneLay Stop
PolTra Start
Initialize PolTra
READ Input e.g. Currents
Simulate Time Step (e.g. TSS)
WRITE Results
Last time step?
No
Yes
Finish
Interface Start
Automated optimization setup
New calibration coefficient values
OneLay and PolTra Wrappers Initalize
OneLay time <= PolTra time?
Yes
No
OneLay Wrapper Start
PolTra Wrapper Start
Model Simulation
Model Simulation
Surge Observation here?
TSS Observation here?
No
No
Yes
Yes
Calculate RMSE
Calculate RMSE
Surge stats
TSS stats
Diverging or last time step?
No
Yes
Calculate Objective Function
Calibrated?
No
Yes
Finish

Figure 2: OneLay/PolTra before (left) and after (right) OPENMI implementation.

New revisions to OneLay and PolTra for this study included the addition of resuspension from the sediments based on computed orbital velocity as well as modifications to implement them for OpenMI. Resuspension was added to the models because it was suspected of being a significant process in Lake Winnipeg. The original OneLay model was a stand alone executable program that would read text input files for all inputs such as time step and duration, initial conditions, daily inputs including wind and river loadings, and would perform the simulations and write the results to text files. PolTra operates the same as OneLay but would read the OneLay output files for currents and water levels. The OpenMI related changes made to the models included converting the models to DLL (dynamic link library), which also avoids the need for writing to text files, and modified code to enable simulation for a supplied time interval. Originally the models would read initial conditions, then run from start to finish with no interaction and could not stop and start because the current state of parameters would be lost. Now they can be used to simulate any time period. Initialization functions were created to start the models; other functions were created to pass in input data; and the current state of parameters is not lost until the new finish functions are called. Output from OneLay can be passed to PolTra immediately so that PolTra does not have to wait for the entire simulation by OneLay to be completed.

Figure 3 illustrates the two model system diagram after the OpenMI implementation. The key difference is that the PolTra model can request information from the OneLay model at any time step and does not have to wait until the completion of the OneLay run.

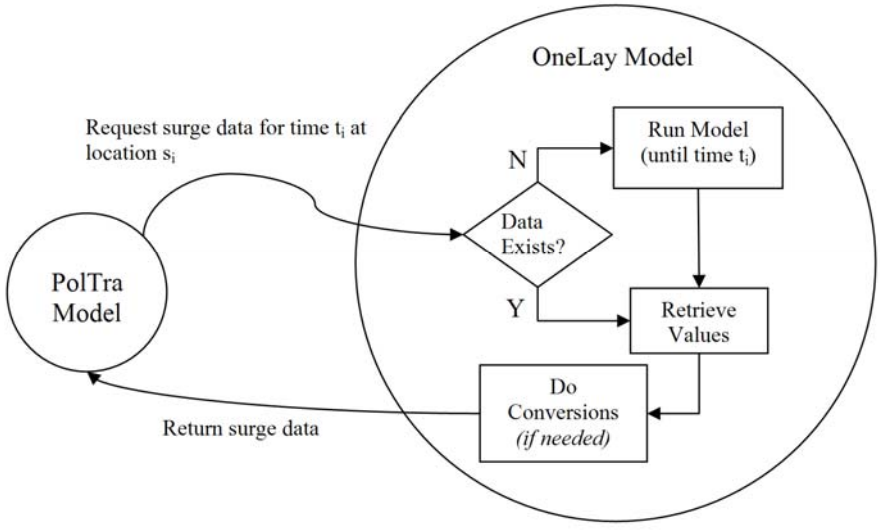

Figure 3: OneLay and PolTra System Diagram after OpenMI Implementation

One of the main objectives of using OpenMI includes simplifying and accelerating the calibration process. Only observed water levels can be used for OneLay calibration since we do not have observed currents. Observed suspended sediment concentrations (or other parameters as available) can be calibrated for PolTra.

An advantage of using the new OpenMI wrapper approach for calibration is that the period can be shortened since the results can be viewed one time step at a time and if divergence is indicated, the run can be stopped.

2.2 Model statistics used in the calibration process

Total suspended sediment (TSS) was chosen as the calibration parameter because it is a relatively simple conservative parameter to model compared to other parameters such as nutrients and has fewer calibration coefficients. It is possible to calibrate for other parameters at the same time, such as water level, but for demonstration we just examine 1 parameter. We use the root mean square error (RMSE) as our modelling statistic:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}$$

where $x_{1,i}$ is observed and $x_{2,i}$ is simulated and i is data point number of TSS, n is the number of data points. RMSE does not have the same problems as coefficient of determination (r^2) and Nash Sutcliffe Efficiency (NSE) coefficient (Nash et al., 1970) and can easily be used for sporadic water quality data.

Table 1 shows the parameters that can be adjusted for calibrations. The drag and bottom friction coefficients in OneLay will affect the currents and water levels, which will in turn affect TSS transport and therefore require calibration. The PolTra calibration coefficients directly affect the TSS concentration. The sediment is suspected to be mostly clay but the exact composition is unknown, which makes the calibration of Wset, Ucr and Kr a necessity. If different chemicals were being modelled then there would be other calibration parameters to consider.

Table 1 Potential TSS Calibration Coefficients

	Parameter	Range
OneLay	Drag (drag coefficient for wind stress)	-50% to +100% default 0.0025
	Bottom friction	0.01 – 0.05 cm/s default 0.05
PolTra	Wset (settling coefficient)	0 – 1 cm/s
	Ucr (critical shear velocity for resuspension)	0.4 – 1.5 cm/s
	Kr (resuspension coefficient)	0 – 1e-18 default 6.4e-14

As seen in Figure 4, four inflow rivers were modelled and include the Dauphin River, Red River, Saskatchewan River, and Winnipeg River. These rivers make up 80 to 90% of the inflow to the Lake. The Winnipeg River contributes approximately 45% of the inflow but the Red River tends to be the dominant contributor of water quality loads such as nutrients. The Nelson River is the outflow river for the Lake and is also modelled. Also shown in Figure 4 are the TSS observations stations for 2002.

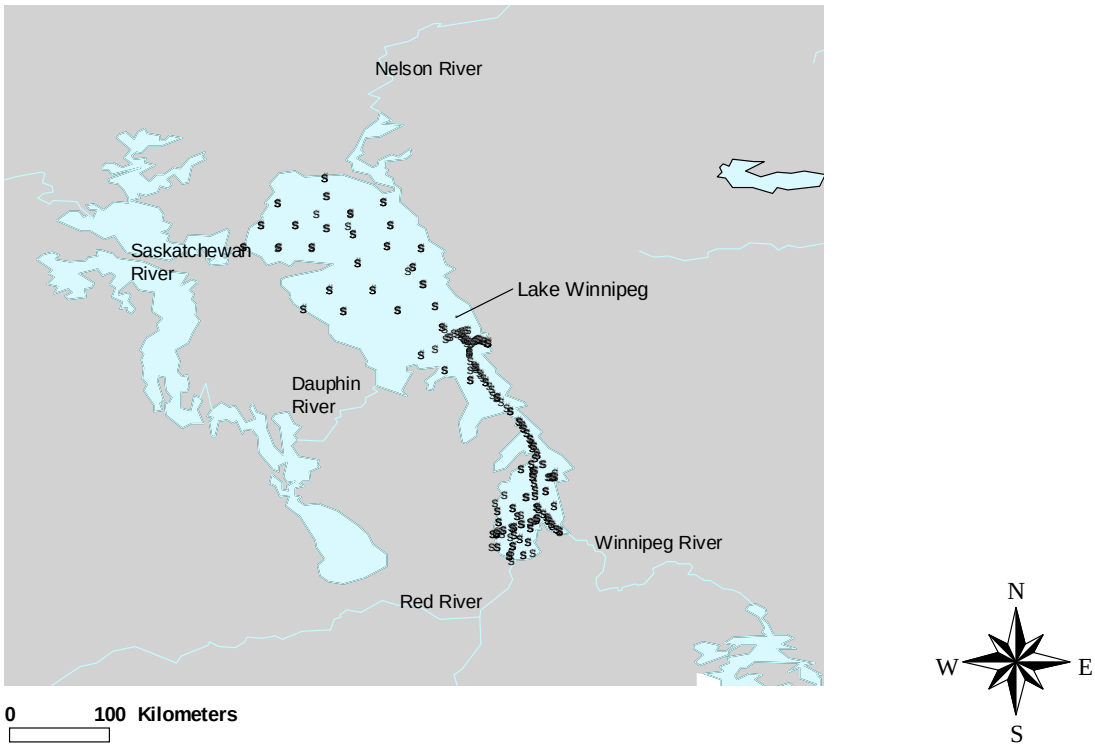

Figure 4: Lake Winnipeg with 2002 TSS observation stations

2.3 Advantages of OpenMI implementation in model calibration

The OpenMI environment provides many benefits to the modeller that may not be present or are difficult to do in traditional approaches of model integration. Models which are OpenMI-compliant can be linked together fairly easily to build larger model systems out of models from different providers. As a result, models can be swapped in and out quickly to test different configurations. OpenMI contains facilities to handle the exchange of data that have different spatial and/or temporal scales, thus addressing one of the problems of model incompatibility. In addition, OpenMI supports two-way interaction of models where models mutually depend on the results from each other allowing for integrated systems with feedback loops to be created.

Model calibration, whether done manually or done through an automatic procedure, is an iterative process requiring repetitive runs of the model simulations with varying model coefficients. This is a time consuming process and using OpenMI can make it perform more efficiently by taking advantage of OpenMI's abilities to run models on a single time

step basis and to query for results at any given time step. Model calibration statistics such as root mean square error or Nash Sutcliffe Efficiency coefficient can now be calculated and updated on an on-going basis as the simulation runs and this would allow for model simulations to be stopped partway through when the statistics are not improving. Hence, simulation run times are possibly reduced.

2.4 Streamlining the calibration process

The model calibration process, which consists of a great many runs of the models, can potentially generate large quantities of results that need to be kept track of and organized. For example, the model inputs should be recorded along with the model results. It is important to retain results from all model runs during the calibration and not just the one determined to have produced the best results because they show how the calibration progressed and are useful afterwards for sensitivity analyses of the model coefficients. Figure 5 is a diagram of the data model for the database used to store the results from the model calibration. The types of information that should be saved include the model results, model coefficients and inputs, calibration statistics to evaluate the quality of the calibration, time step when the model run stopped, and any other associated metadata. This information is stored across seven different tables: Models, ModelStatistics, Calibrations, ModelRuns, ModelInputs, ModelOutputs, and ModelRunStats. A calibration is related to one or more ModelRuns records and each model run consists of multiple records in the ModelInputs, ModelOutputs and ModelRunStats tables. The Models and ModelStatistics tables store general information about the models and statistics and are independent of any one calibration or model run. For simplicity, the actual model outputs are not saved within the database itself, but are kept separate in their native format and the database records the location where they can be found.

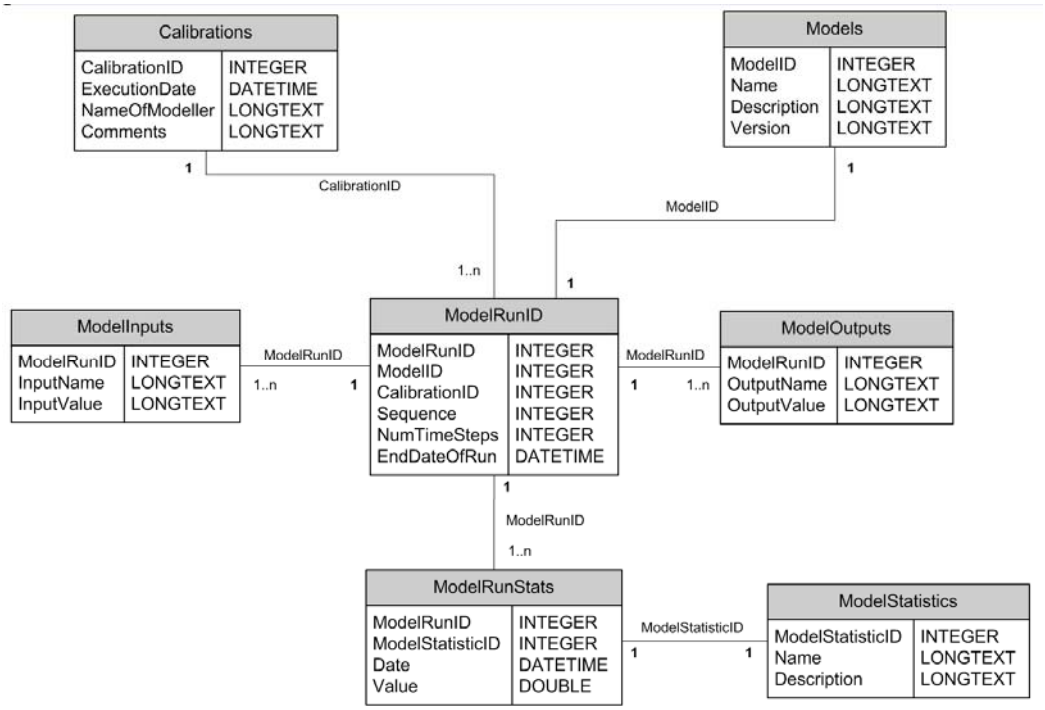

Figure 5: diagram of the modelling database

2.5 Use of artificial intelligence techniques in the modelling calibration process within a decision support system

Techniques from the field of artificial intelligence are well suited for automating and/or optimizing the model calibration process. Global optimization algorithms such as Shuffled Complex Evolution (SCE) (Duan et al., 1993), Dynamically Dimensioned Search (DDS) (Tolson et al., 2007) and Genetic Algorithms (GA) (Goldberg, 1989) can be applied to search the multi-dimensional model coefficient space in an efficient manner to try to find a near optimal set of coefficients, which results in accurate predictions of observed data by the model. These algorithms begin with an initial population of candidate or potential solutions, which are usually randomly generated. The individuals or members of the population are then evolved to form a new population for the next generation. The evolution proceeds in a manner that attempts to minimize or maximize the objective (optimization) function. In model calibration, functions that assess the prediction accuracy such as root mean square error (RMSE) or Nash Sutcliffe Efficiency coefficient are suitable objective functions. This evolutionary process repeats until a terminating condition has been achieved such as a maximum number of population generations or objective function values not improving.

To test the proof of concept, a simple decision support system (DSS) can be built that integrates an artificial intelligence technique that is used to calibrate the models for total suspended sediment (TSS). Calibration involves comparing the simulated to observed values and adjusting model parameters to get the simulated values closer to the observed. The objective function was computed as follows:

$$\text{Objective Function} = 1 / (1 + \text{RMSE})$$

The closer the objective is to 1 the better the agreement between simulated and observed. This particular objective function was used in order to meet the requirements of the automated calibration routine: objective function must be a real number from 0 to 1 and a higher value is better fit. Since the observed values are scattered around the Lake by location and by time, the RMSE was computed as an accumulated value for the whole lake. After each computation time step, the system would check for observed values for that time period. If there were observed values, then the simulated value for the matching grid cell would be retrieved and included in the RMSE calculation. At this point the RMSE would be based on all observations since the start of the simulation.

Automating calibration using artificial intelligence in the DSS has many advantages including the following:

- Manually comparing simulated and observed values is extremely tedious. One would have to extract values for specific grid cells at the same time of the observed from text files. Using an automated process that makes several runs is more efficient, and can be used with OpenMI components.

- Normally the comparison to observed values is done via statistics or visually with contoured maps after each calibration run. Each model run would have to be set up then run, and after the run the statistics and/or maps produced and compared to previous runs, then calibration parameters adjusted and another simulation would be prepared. The automated calibration would allow one to "walk away" while calibration is performed with no interaction required. The calibration would run for a set number of runs or until some criteria is reached, i.e. until the model is found to have been calibrated satisfactory.

3. Discussion

The proof-of-concept DSS is currently being implemented. At present, we have successfully implemented the OpenMI for the OneLay and PolTra models. Each model calibration run is stored in the modelling database. Figure 6 shows results of a typical calibration run with observed versus model TSS results and the RMSE value at the end of the simulation.

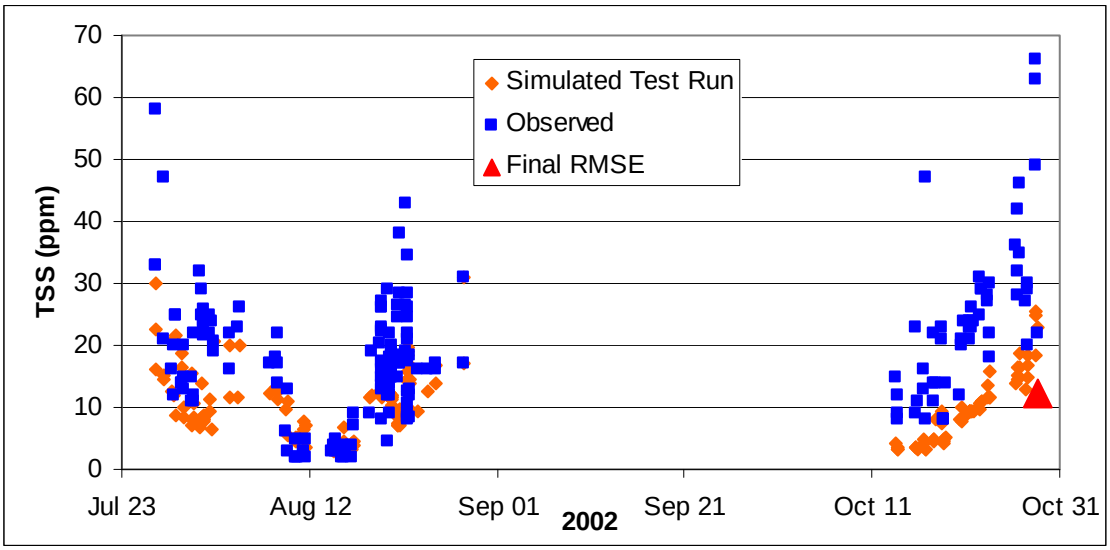

Figure 6: Results of a typical complete calibration run with observed versus model TSS results and the RMSE value

The OpenMI implementation proves to be fairly effective in our study. Because of the short time of this project, we are currently implementing a simple GA algorithm (Yang et al., 1998) to automate the modelling calibration process. Early results indicate that in some generations, the calibration run terminates during the iteration due to worsening RMSE comparison between the current run and the existing best run.

However, the OpenMI technique does have its share of drawbacks. The model program itself must be modified, which means that access to the source code is required. Thus, models whose source code cannot be obtained (e.g., proprietary or legacy models) are unable be used within the OpenMI environment. Programming ability is needed to

modify the model program and to develop the wrapper and knowledge of the inner workings of the model program is required, which can be difficult to understand if one is not the author. Also, passing a large scalar set between components was found to slow down the simulation. The flexibility of OpenMI with different units seems to slow it down. Conversely, there is an advantage to editing the source code in this example for OneLay and PolTra: the original models were run sequentially and PolTra would read the OneLay output text file to get currents but due to the large size of the output file, it was only possible to save the OneLay output every six hours of the simulation so the currents used by PolTra would only change every 6 hours. The OpenMI implementation is able to pull the currents from OneLay every time step, pass to PolTra, and bypass the need for reading input from text files.

4. Recommendations

There are a few areas that we hope to improve in the future. Although a robust modelling database is built to handle the modelling calibration process, little ontology is considered at this time. We need to adopt an approach based on the concept of ontologies to represent the relationships among data and models. The use of ontology coupled with modelling meta-data standards not only improves the quality of the modelling calibration, but also reduces the complexity of handling data and model results. Building a semantic enriched framework based on ontology produces an intelligent agent utilizing the knowledge base to detect any discrepancy or problem within the integrated modelling environment.

We demonstrate the use of a typical genetic algorithm in this paper. However, the process is fairly slow and requires a large amount of iterations to converge. In reality, we should select a more efficient approach such as SCE and DDS for the optimization of model coefficient calibration. Moreover, new emerging techniques in parallel computing should be considered.

The OpenMI allows us to further expand to integrated environmental modelling to both watershed and lake models, and beyond. The results of the watershed models will affect the calibration of the lake models, and vice versa. Two-way calibration within this type of model integration is required for overall optimal model calibration.

Finally, some desirable functionality in the DSS should be built to make integrated modelling more user-friendly. The functionality includes integration of datasets with ontologies to ensure consistency of data flow, handling iterations of modelling calibration intelligently and a mapping platform such as MapWindow (Watry & Ames, 2008) to review all the model results and overall model statistics.

Acknowledgements

The authors would like to thank Dr. L. Leon, Dr. R. Yerubandi, M. Sloboda and O. Resler of Environment Canada and Dr. D.A. Swayne of University of Guelph who provided their useful insights. Special thanks to C. Richard of the University of Waterloo for his invaluable input in the OpenMI implementation.

References

Duan, Q., Gupta, V. K., and Sorooshian, S. 1993. Shuffled complex evolution approach for effective and efficient global minimization, Journal of Optimization Theory and Applications. 76 (3), 501–521.

Goldberg, D. E. 1989. Genetic algorithms in search, optimization and machine learning. Kluwer Academic Publishers, Boston, MA.

Moore, R. V. and Tindall, C. I. 2005. An overview of the open modelling interface and environment (the OpenMI). Environmental Science and Policy. 8, 279-286.

Nash, J. E. and Sutcliffe, J. V. 1970. River flow forecasting through conceptual models part I — A discussion of principles, Journal of Hydrology, 10 (3), 282–290.

Simons, T.J. and Lam, D.C.L. 1986. Documentation of a Two-Dimensional X-Y Model Package For Computing Lake Circulations and Pollutant Transports in Physics-based modeling of lakes, reservoirs, and impoundments: a report / prepared under the auspices of the Committee on Environmental Effects of the Energy Division of the American Society of Civil Engineers under a grant from the U.S. Environmental Protection Agency through the Oak Ridge National Laboratory; edited by William G. Gray.

Tolson, B. A. and Shoemaker, C. A. 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration, Water Resources Research. 43, W01413, doi:10.1029/2005WR004723.

Watry, G. and Ames, D. P. 2008. A practical look at MapWindow GIS.

Yang, G., Reinstein, L.E., Pai, S., Xu, Z. and Carroll, D. L. 1998. A new genetic algorithm technique in optimization of permanent 125-I prostate implants, Medical Physics, Vol. 25, No. 12, December 1998, pp. 2308-2315.