Evaluation and Comparison of Models and Modeling Tools Simulating Nitrogen Processes in Treatment Wetlands

Stina Edelfeldt, Peter Fritzson
PELAB – Programming Environment Lab
Dept. of Computer and Information Science,
Linköping University, Sweden
Email: x04stede@ida.liu.se, petfr@ida.liu.se

Abstract

The global problem of eutrophication, the enrichment of water bodies by nutrients, has in recent years resulted in an increased interest in non-conventional ways to reduce the amount of nutrients discharged into the environment. One commonly proposed and used solution is to append constructed wetlands as a final step in the treatment of wastewater. However, both the construction of the wetlands and the maintenance thereof are time and resource consuming. Due to the high production and maintenance costs, tools for modeling and simulation are often valuable assets in the development of constructed wetlands.

In this paper, two ecological models of nitrogen processes in treatment wetlands have been evaluated and compared. These models were implemented, simulated, and visualized in the Modelica language [5]. The differences and similarities between the MathModelica Model Editor and three ecological modeling tools have also been evaluated.

1. Evaluation and Analysis Methods

The approach to the evaluations and analyses has been the following:

1. A number of important features for each modeling tool/model has been found and evaluated using the McCall method [7].
2. A comparative analysis has been made to determine the differences and similarities between the models or modeling tools and the advantages and disadvantages of different approaches in the models or modeling tools. To ensure that as many differences and similarities as possible are found the comparative analysis has been made in four ways (see Section 1.2).
3. The significance and consequence of the differences in the features has been discussed.

1.1 Evaluation Method

The evaluation method used in this paper is a qualitative method. Each of the features has been determined by observation of the model or modeling tool, and the total quality has been measured on a relative scale, depending on the quality perceived. This increases the risk of a subjective evaluation. To increase the objectivity, the chosen features have been categorized only as present or absent in the models or modeling tools.

In the model evaluation, two models have been simulated and evaluated. These models are described in Edelfeldt & Fritzson [5]. As the models are written in Modelica, the focus of the evaluation is on the functionality and features available in the Modelica implementation of the models. Consideration has been taken to the possibilities and limitations of the programming language when determining features to evaluate.

In the modeling tool evaluation, the MathModelica Model Editor has been compared with tools commonly used for ecological modeling. These tools have been found by searching the internet for modeling tools specifically used for modeling ecological systems. Note that tools not commonly present on the internet may have been overlooked in this search. A prerequisite for the tools has been the existence of a graphical interface.

The focus of the modeling tool evaluation is mainly on the functionality and not on the specific programming languages of the modeling tools. For the common user, this functionality is expressed through the graphical interface, and consequently, it is the functionality available from the graphical interface that is the subject of this evaluation.

For each modeling tool, a model similar to the wetland Nitrification/Denitrification model (see Section 2.2) has been created, although the equations and processes in the model have been simplified. What is important in this evaluation is not that the modeling tools can produce an exact copy of the Nitrification/Denitrification model, but that sufficient equivalents of all the necessary functions and parts of a wetland can be found and used.

A more detailed description of the model and modeling tool evaluation and analysis can be found in Edelfeldt [14].

First, for both evaluations, a number of major factors have been chosen from McCall's quality factors [7]. The factors chosen are:

- *Correctness* - Extent to which a program satisfies its specifications and fulfills the user's mission objectives.
- *Flexibility* - Effort required to modify an operational program.
- *Interoperability* (modeling tool evaluation only) - Effort required to couple one system with another.
- *Maintainability* - Effort required to locate and fix an error in an operational program.
- *Reusability* - Extent to which a program can be used in other applications.
- *Usability* - Effort required to learn, operate, operate, prepare input, and interpret output of a program.

From these factors, a number of criteria equivalent to McCall's quality criteria have been chosen. Some of the criteria influence several factors. Not all criteria listed for each factor are present. This is because no interesting elements have been found for these criteria. The criteria and the factors they influence are shown in Appendix A and B.

To ensure that all features interesting for the model comparison are evaluated, McCall's criteria have been complemented with the model and theory criteria Efficiency, Generality, Model Conditions, Systematics and Validity from "Vetenskapsteori och forskningsmetodik" [8]. The criteria Efficiency and Generality are equivalent to McCall's Operability and Generality criteria, respectively. Model Conditions, Systematics and Validity have been added to the model evaluation, influencing the proper factor (see Appendix A).

Second, a number of important major features have been recognized. The model features are specific for the purpose of simulating nitrogen loss in a wetland and constitute the Traceability criteria. They have been found by listing the necessary attributes for simulation and representation of the model. The features are:

- Prediction of the total nitrogen decrease.
- Predictions of the ammonium and nitrate nitrogen decrease – i.e. the nitrification and denitrification processes.
- Possibility to change parameters in the simulations.
- Possibility to plot chosen parameters in the simulations.
- Changes in concentration described as derivates with respect to a factor (time in wetland, distance in wetland, fraction of wetland etc).

The modeling tool features have been found by listing the necessary attributes for simulation and representation of the wetland model and are all part of the Completeness criteria. The features are:

- Representations of the physical parts of a system.
- Flows/connections between the different parts of the system.
- Equations specifying system reactions.
- Variables specifying different values that can be measured or calculated in the system.
- Diagrams showing variables changes over time.

Third, for each of the criteria, a number of elements have been chosen (Appendix A and B). These are features that are important for the quality of the wetland model or modeling tool. Two or three elements have been chosen for each criterion except for the model Traceability criterion and the modeling tool Completeness criterion. These criteria have more elements since elements corresponding to and evaluating the attributes of the model or modeling tool are all put in these criteria.

Some of the elements might be placed in more than one criterion. To avoid imbalance in the evaluation, the elements have only been placed in the criteria they are considered most suited for.

The McCall factors Efficiency, Integrity, Interoperability (for the model evaluation), Portability, Reliability and Testability have not been considered in this evaluation. The reason is that these factors do not address the proper issues for this evaluation or that the models used in the evaluations are too small to be proper study objects for the factors.

1.2 Comparative Analysis Method

The comparative analyses in this paper have been made in four ways:

1. By adding the features together and analyzing them using McCall's method [7] to determine the quality ratings of each model or modeling tool.
2. By using a correlation analysis to see if there are some mathematical similarities or differences between the models or modeling tools.

3. By analyzing the data from the evaluation with a purely qualitative analysis, a simple variant of the Constant Comparative method [3].
4. Finally, by specifying differences and similarities noted while analyzing the data, especially if they are between entire criteria.

The methodology of adding quality features is the same as the methodology used by McCall [7]. McCall uses the defined set of metrics to develop expressions for each of the factors according to:

$$Fq = c_1 \text{ x } m_1 + c_2 \text{ x } m_2 + \ldots + c_n \text{ x } m_n$$

where Fq is a software quality factor, c_n are regression coefficients and m_n are the metrics that affect the quality factor.

The equation for the correlation coefficient used is:

$$Correl(X,Y) = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

where x and y are the sample means of the two arrays.

The Constant Comparative method is a method used to classify different phenomena into different categories and from these categories find a theory that states the major qualitative features of the studied objects. The Constant Comparative method consists of four stages [3].

In the first stage the analyst starts by coding each incident in the data into as many categories as possible, as categories emerge or as data emerge that fit into an existing category. While coding an incident for a category it is important to compare it with the previous incidents in the same and different groups of the same category.

In the second stage, the method changes from comparison of incident with incident to comparison of incident with properties of the category that resulted from initial comparison of incidents.

In the third stage, the theory begins to develop, in that major modifications become fewer and fewer as the analyst compares the incidents of a category to its properties. Non-relevant properties are removed, modifications to clarify the logic are made, and details are integrated into the categories. Reduction is important in this stage. In this context, reduction means that the analyst may discover underlying uniformities in the original set of categories or their properties, and then formalizes a theory with a smaller set of higher properties [3].

In the fourth stage, the analyst produces, writes and formalizes the theory.

1.2.1 Realization of the Comparative Analysis Method

First, the quality scores have been calculated for the ecological models and modeling tools, according to McCall's methods [7]. The calculations for each factor in this paper are:

Criterion = Number of yes for criterion / Number of elements in criterion

Factor = Criterion$_1$ + Criterion$_2$ + … + Criterion$_n$ / Number of criteria in factor

In this paper, the same method has also been used to calculate the total quality.

Total quality = Factor$_1$ + Factor$_2$ + … + Factor$_n$ / Number of Factors

As these calculations weigh each criterion equally, the results may be imbalanced. In this paper, weighted values have been calculated for the Correctness factor and the Traceability (models) or Completeness (modeling tools) criterion as they contain many of the total elements. The factor and the criteria have been weighted to get the same relative importance as the other elements (Table 3). To further guarantee that the wrong conclusions are not drawn from imbalanced results the total quality has also been calculated without consideration to the criteria or factors, as follows:

Total quality = Number of yes for each modeling tool or model / Number of total elements

Second, the correlation analysis has been made by calculating the correlation coefficient between all evaluation objects, using all elements as measurement basis for the calculations.

Third, a very simple variant of the Constant Comparative method has been used. The analysis has compared the results of the evaluation, i.e. which elements are present or absent. The elements listed in Appendix A and B are considered equivalent to the incidents of the Constant Comparative method. In this simple analysis, the first two stages of the Constant Comparative method have been done in one step, as have stages three and four.

Fourth, differences and similarities noted while analyzing the data have been specified, especially if the differences/similarities can be attributed to entire criteria.

To increase the objectivity and accuracy of the evaluation, the modeling tools have not only been compared to Model Editor, but also to each other.

2. Evaluated Models

Detailed descriptions of the models including the Modelica code can be found in Edelfeldt [14].

2.1 Total Nitrogen Model

The model and equations for simulating total nitrogen is from Arheimer and Wittgren [1]. In this paper, nitrogen removal in the wetland is described with a simple first-order model. The wetland is viewed as a completely mixed batch reactor. The model assumes a constant flow of water and is therefore not suitable when the wetland receives a natural and variable water flow. As treatment wetlands receive a relatively constant flow, this is not a problem for the simulations of this contribution.

The Total Nitrogen Model has no graphical representation. This is because the entire model would only consist of one single component with no flows to other components.

2.2 Nitrification/Denitrification Model

In the Nitrification/Denitrification Model, the wetland has been divided into several layers, each constituting a plug flow reactor (PFR). Flows of nitrogen go between the different unmixed layers, simulating the flow of nitrogen within the wetland. The layers consist of a water body, an aerobic sediment layer and several anaerobic sediment layers. The purpose of this division is to simulate the different rates of nitrogen processes within the wetland depending on the oxygen level and other factors.

The Nitrification/Denitrification model is modeled after Kadlec & Knight [4] with some influences from Martin & Reddy [9]. A graphical representation of the model is shown in Figure 5.

3. Evaluated Modeling Tools

3.1 PowerSim Studio 2003

PowerSim is an integrated environment for creating and running simulation models [10]. It uses the graphical modeling language known from the system dynamics method to model a system. The tool uses presentation objects like graphs and tables, and has linking capabilities.

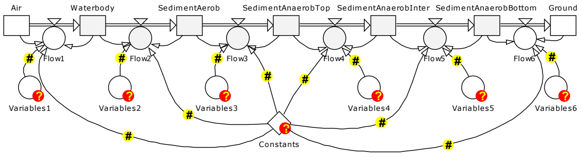

Figure 1. The wetland model modeled in PowerSim, including variable and constant examples.

3.2 Simile

"Semantic Interoperability of Metadata and Information in unLike Environments" (Simile) is a software tool for computer simulation of dynamic systems in the earth, environmental and life sciences [11]. It uses a logic-based declarative modeling to represent the interactions in these systems in a structured, visually intuitive way. It also uses the graphical modeling language known from the system dynamics method.

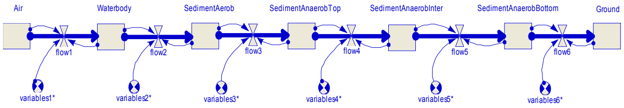

Figure 2. The wetland model modeled in Simile, including variable examples.

3.3 Stella

"Structural Thinking, Experiential Learning Laboratory with Animation" (Stella) was developed for general modeling education. It builds and simulates models of dynamic systems and processes [12]. The graphical interface allows the user to build mathematical relationships without any knowledge of programming. Using the graphical modeling language known from the system dynamics method, the user can construct a map of a process or system.

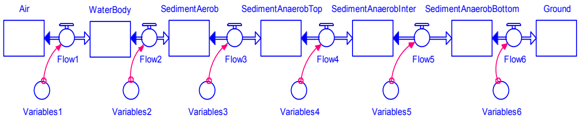

Figure 3. The wetland model modeled in Stella, including variable examples.

3.4 WEST

"World wide Engine for Simulation, Training and Automation" (WEST) is a modeling and simulation environment for any kind of process that can be described as a structured collection of Differential Algebraic Equations (DAEs) [13]. It allows for graphical component-based modeling and offers an environment for the modeling and simulation of different processes such as wastewater treatment plants, rivers, sewers and other water management systems.

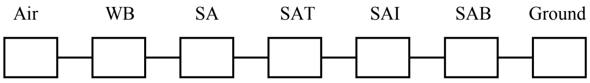

Figure 4. A simplified example of the wetland model modeled in WEST.

3.5 MathModelica Model Editor

The 2004 version of MathModelica Model Editor is a graphical user interface for model diagram construction by “drag-and-drop” of ready made components from model libraries graphically represented by Visio stencils [6]. These libraries correspond to the physical domains represented in the Modelica Standard Library or components from user defined component libraries.

The basic functions of Model Editor are the selection of components from existing libraries, to connect components in model diagrams, and to enter parameter values for different components [6].

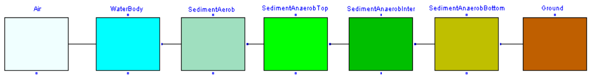

Figure 5. An example of the wetland model modeled in Model Editor.

4. Analysis Results

4.1 Quality Scores

Quality scores have been calculated using McCall’s method as described in Section 1.2.1. In Table 2, the number of elements present and their relative frequencies in the criteria are listed. In Table 3, the relative frequencies of the criteria have been used to calculate the quality of each factor. The factors have then been used to calculate the total quality of the modeling tool. In Table 4, the total numbers of elements present for each model or modeling tool are listed. From these elements, a value for total quality has been calculated.

Note that it is the quality values in relation to each other that are important to study, not the exact quality values of each model or modeling tool.

It seems the Nitrification/Denitrification model has a somewhat higher total quality, regardless of whether a weighted value is used or not. The total quality shown in Table 4 also shows a higher value for the Nitrification/Denitrification model.

For the modeling tools, the use of a weighted value has some significance. The order between the modeling tools are similar no matter how the quality is measured, in that WEST always has the highest value, followed by Model Editor and the tools using the system dynamics method. However, the internal order between the system dynamics tools varies somewhat. These changes are probably due to the differences in the Completeness criteria between the modeling tools.

The result for the modeling tools in Table 4 seems to be more or less consistent with the result from Table 3. WEST seems to have the highest total quality, followed by Modelica Editor and the system dynamics tools. The total quality is lower in all cases except the total quality of Model Editor compared with the total quality when a weighted value is used.

4.2 Correlation Analysis

The correlation analysis has been made by calculating the correlation coefficient for the models or modeling tools.

The correlation between the Total Nitrogen model and the Nitrification/Denitrification model is 0,194. This means that is there no significant major difference between the two. There seems to be little correlation at all. This indicates that some model features are similar and some features are different, thus balancing out the score.

As shown in Table 1, there is a relatively high correlation between PowerSim, Simile and Stella, and between WEST and Model Editor. In all other cases the correlation is low or very low. There is some indication of a significant difference, i.e. negative correlation, between WEST and Stella.

The results indicate that Model Editor and WEST have many similarities, as have PowerSim, Simile and Stella.

Table 1. Results of correlation analysis between the different modeling tools. The values are listed with three decimals.

	PowerSim	Simile	Stella	WEST
Simile	0.732	-	-	-
Stella	0.600	0.600	-	-
WEST	0.040	0.073	-0.339	-
Model Editor	0.047	0.047	-0.144	0.693

4.3 Differences and Similarities Using the Constant Comparative Method

Differences and similarities between the models have been found by using the Constant Comparative method.

The Nitrification/Denitrification model is more complicated than the Total Nitrogen model to learn in that not all variables are the same as in the original, and that it is possible to confuse or connect one part of the model with another. Both models are valid, consequent and can only be used in related systems, and neither of them has any useful error messages except for abnormal variables. Two different categories emerge from this; a complex, reusable, expandable model, and a

Table 2. The number of elements present for each of the criteria in the models/modeling tools and their relative frequencies. From the results in Appendix A and B.

Criteria (number of Model/Modeling tool elements)	Total Nitrogen Retention	Nitrification/ Denitrification	PowerSim	Simile	Stella	WEST	Model Editor
Completeness (3/10)	1 (1/3)	3 (1.0)	3 (0.3)	4 (0.4)	5 (0.5)	7 (0.7)	8 (0.8)
Consistency (2/2)	2 (1.0)	1 (0.5)	2 (1.0)	2 (1.0)	2 (1.0)	2 (1.0)	2 (1.0)
Data and communication commonality (-/2)	-	-	1 (0.5)	1 (0.5)	0 (0.0)	2 (1.0)	1 (0.5)
Expandability (2/2)	0 (0.0)	2 (1.0)	0 (0.0)	0 (0.0)	1 (0.5)	2 (1.0)	2 (1.0)
Generality (2/2)	1 (0.5)	1 (0.5)	2 (1.0)	1 (0.5)	2 (1.0)	1 (0.5)	1 (0.5)
Instrumentation (2/2)	1 (0.5)	1 (0.5)	2 (1.0)	1 (0.5)	2 (1.0)	0 (0.0)	0 (0.0)
Model conditions (2/-)	2 (1.0)	1 (0.5)	-	-	-	-	-
Modularity (3/3)	1 (1/3)	3 (1.0)	2 (2/3)	2 (2/3)	1 (1/3)	3 (1.0)	3 (1.0)
Operability (3/3)	3 (1.0)	2 (2/3)	2 (2/3)	2 (2/3)	2 (2/3)	2 (2/3)	2 (2/3)
Simplicity (2/2)	1 (0.5)	1 (0.5)	1 (0.5)	2 (1.0)	2 (1.0)	1 (0.5)	1 (0.5)
Systematics (3/-)	3 (1.0)	3 (1.0)	-	-	-	-	-
Traceability (5/-)	4 (0.8)	5 (1.0)	-	-	-	-	-
Training (-/2)	-	-	1 (0.5)	1 (0.5)	1 (0.5)	2 (1.0)	0 (0.0)
Validity (3/-)	3 (1.0)	3 (1.0)	-	-	-	-	-

Table 3. The relative frequencies of the criteria in the models and the calculated value for total quality. The values are listed with three decimals. Based on the results from Appendix A and B.

Factor (influencing Model/Modeling tool criteria)	Total Nitrogen Retention	Nitrification/ Denitrification	Power-Sim	Simile	Stella	WEST	Model Editor
Correctness (Completeness, Consistency, Model conditions, 2*Traceability)/(4*Completeness, Consistency)	0.787	0.800	0.440	0.520	0.600	0.760	0.840
Flexibility (Expandability, Generality, Model conditions, Modularity, Simplicity)/(Simplicity, Expandability, Generality, Modularity)	0.467	0.700	0.542	0.542	0.708	0.750	0.750
Interoperability(-)/(Modularity, Data and communication commonality)	-	-	0.583	0.583	0.167	1.000	0.750
Maintainability (Instrumentation, Modularity, Simplicity, Validity) /(Simplicity, Instrumentation, Modularity)	0.533	0.750	0.722	0.722	0.778	0.500	0.500
Reusability (Generality, Modularity, Simplicity, Systematics) /(Simplicity, Generality, Modularity)	0.583	0.750	0.722	0.722	0.778	0.667	0.667
Usability (Operability, Systematics) /(Operability, Training)	1.000	0.833	0.583	0.583	0.583	0.833	0.333
Total quality weighted (Correctness*2.4) / (Correctness*10/3)	0.699	0.774	0.554	0.586	0.602	0.754	0.696
Total quality	0.674	0.767	0.599	0.612	0.602	0.752	0.640

Table 4. The total number of elements present for each model and the calculated value for total quality. The values are listed with three decimals.

	Total Nitrogen Retention	Nitrification/ Denitrification	Power-Sim	Simile	Stella	WEST	Model Editor
All elements	22	26	16	16	18	22	20
Total quality	0.688	0.813	0.533	0.533	0.6	0.733	0.667

simple model with fewer possibilities of reuse and expandability. The only thing that contradicts this is that both models have complexity in that both cannot be understood without programming skill. However, the Total Nitrogen model is still easier to understand than the Nitrification/Denitrification model. To simplify the names of the categories they can be called Complex Models and Simple Models.

The theory that results from this is that the major difference between these categories lies in the number of functions and in the possibility of reuse and expansion. Another theory is that the similarities between the models are that they are all consequent, logical, valid, specialized, and easy to use if the user has programming skill.

Differences and similarities between the modeling tools have also been found using the Constant Comparative method.

All the tools are fairly easy to use and overview. However, WEST and Model Editor have more functionality and allow for more complex modeling and the creation/reuse of separate components. PowerSim, Simile and Stella have features that make it easier for users with little programming experience, like built-in mathematical functions and useful error messages. The major difference between the two groups is in the possibility to create and reuse separate components and all the complexity in these components. To simplify the names of these categories, they are called Complex Components and Simple Components. The tools in the Simple Components category often have a high uniformity, simplicity, generality and error handling. The users of these tools cannot create and define their own components. The Complex Components have a high completeness, expandability and modularity. They have more functionality, allow for more complex modeling and the creation and reuse of separate components.

The theory that results from this is that the major difference between these categories lies in the possibility to create and reuse separate components and the complexity in these components. Another theory is that the similarities between the categories are that they are all consistent, easy to overview and use, if no new components are to be created.

4.4 Noted Differences and Similarities

There seem to be more criteria that are very similar than are very different between the two models. The Generality, the Instrumentation, the Simplicity, the Systematics and the Validity criteria all show the same results. Many of these features are general features that have to do with error handling, if there is consequence in the model and how easily it is handled. There is also a high similarity between the models in the Operability criteria.

There are differences between the models in the Completeness and Modularity criteria, although the difference is not total. There is, however, total difference in the Expandability criteria. What these criteria have in common is that they favor a model separated into several independent parts with different equations kept separate in independent components as much as possible. This is difficult in a simple model as the Total Nitrogen model, which gives the Nitrification/Denitrification model an advantage over the Total Nitrogen model. The Total Nitrogen model seems to have an advantage where Consistency and Model conditions are concerned, though, which probably is a result of it being a small simple model.

The noted differences and similarities in the models are consistent with the results from the Constant Comparative method.

WEST and Model Editor have much in common. Only four features of the 30 listed in Appendix B differ. It is interesting that both features in the Training criteria differ. A major difference between the two modeling tools is the help file and manual – i.e. the help the tool provides the user to learn the functionality of the tool. Help features are virtually non-existent in Model Editor.

The system dynamics modeling tools also have much in common, especially PowerSim and Simile. The differences are more diverse, however, so no overall conclusions can be drawn. It is more interesting to note the differences between the systems dynamics group and the group consisting of WEST and Model Editor. To begin with, many of the Completeness features seem to be different. The flexibility of WEST and Model Editor that allows for modification of existing components is higher than for the system dynamics tools. The Expandability criteria should also be noted, as this criterion is higher in WEST and Model Editor. This is also connected to the components, as the feature concerns the flexibility and the creation of the components. Another difference between the groups is in the Instrumentation criteria, where neither WEST nor Model Editor has any useful automatic error handling, while the system dynamics tools are useful in this aspect.

Two interesting similarities should be noted. The values of the features for the Operability and the Consistency criteria are identical. There seem to be little difference in the quality of the Consistency and Operability of the tools.

The noted differences and similarities in the modeling tools are consistent with the results from the Constant Comparative method.

5. Concluding Discussion

The modeling tools and the models can both be separated into two categories: Simple Components and Complex Components for the modeling tools, and Simple Models and Complex Models for the models. The similarities between the types of these categories are apparent. These similarities are probably not coincidental. When modeling, there is often a choice between complexity and simplicity. A complex model or tool may provide more possibilities to detail a simulation of a process or a system. However, the price of this complexity and flexibility is lack of simplicity. It may not be necessary or even possible to describe a system detailing many parameters, and a simpler model may often be enough. This consideration must always be taken in account when modeling. Most (if not all) models are simplifications of real life, and it is only the level of simplification that has to be decided.

To conclude this paper, the nitrogen decrease in a constructed treatment wetland should be simulated using the Nitrification/Denitrification model as this model has the highest overall quality score and provides a more variable environment. The model can well be simulated in the MathModelica Model Editor, as the Model Editor has an equal and often higher overall quality score compared with the other modeling tools – only WEST has a higher score. However, some changes to the Model Editor are recommended to make the creation of the model easier. These changes include the addition of a tutorial and the addition of useful error handling and messages.

References

[1] Arheimer,B. & Wittgren, H.B. (2002) Modeling Nitrogen retention in potential wetlands at the catchment scale. *Ecological Engineering 19*: 63-80.

[2] Fritzson, P. (2004). Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, 939 pages, Wiley-IEEE Press, ISBN 0-471-471631.

[3] Glaser, Barney G. & Strauss, Anselm L. (1999). *The discovery of grounded theory: strategies for qualitative research.* 2nd ed. Aldinne de Gruyter, New York.

[4] Kadlec, R. H. & Knight, R.L. (1996*). Treatment Wetlands*. Lewis, Boca Raton, Florida.

[5] Edelfeldt, S. & Fritzson, P. (2005). Evaluation and Comparison of Ecological Models Simulating Nitrogen Processes in Treatment Wetlands, Modeled in Modelica. In *BioMedSim 2005 - Proceedings of the Conference on Modeling and Simulation in Biology, Medicine and Biomedical Engineering.* Linköping, Sweden, May 26-27 2005. UniTryck.

[6] Fritzson, P. & Bunus, P. (2002). Modelica – A General Object-Oriented Language for Continous and Discrete-Event System Modeling and Simulation. In *Proceedings of the 35th Annual Simulation Symposium*. San Diego, California, April 14-18 2002. IEEE Press.

[7] McCall, Jim. A., Richards, Paul. K., & Walters, Gene F. (1977). *Factors in software quality*. RADC-TR; 77-369. 3 bind. Griffiss, N.Y. : Griffiss Air Force Base.

[8] Wallén, G. (1996). *Vetenskapsteori och forskningsmetodik.* 2nd ed. Studentlitteratur, Lund.

[9] Martin, J.F & Reddy, K.R. (1997). Interaction and spatial distribution of wetland nitrogen processes. *Ecological Modelling*. 105: 1-21.

[10] *Powersim Studio 2003*. Powersim Software AS. From <http://www.powersim.com/products/studio.asp> January 3 2005.

[11] *Welcome to Simulistics*. Simulistics Ltd. From <http://simulistics.com/index.htm> January 3 2005.

[12] *Stella – Systems Thinking for Education and Research.* Isee systems, inc. From <http://www.iseesystems.com/softwares/Education/StellaSoftware.aspx> January 3 2005.

[13] *West - Worldwide Engine for Simulation, Training and Automation.* Hemmis.com. From <http://www.hemmis.com/products/WEST/default_WEST.htm> January 3 2005.

[14] Edelfeldt, S. (2005). *Evaluation and Comparison of Ecological Models Simulating Nitrogen Processes in Treatment Wetlands, Implemented in Modelica*. C-uppsats LIU-ITN-C—05/004—SE. Department of Science and Technology, Linköpings universitet, Electronic version: <http://www.ep.liu.se/exjobb/itn/2005/asp/004>.

Appendix A. Comparison between the simulated models. Specific elements are listed for each criterion.

	Total Nitrogen Retention	**Nitrification/ Denitrification**
Completeness (Correctness) - Attributes of the software that provide full implementation of the functions required.		
Different concentrations can be calculated in different parts of the wetland	No	Yes
The model takes flows within the wetland into consideration	No	Yes
The model takes outside variables like temperature into consideration	Yes	Yes
Consistency (Correctness, Reliability) - Attributes of the software that provide uniform design and implementation techniques and notation.		
Use of the same symbols for the variables in the simulated model as in the original model	Yes	No
Use for similar variable names for similar equations throughout the whole model	Yes	Yes
Expandability (Flexibility) - Attributes of the software that provide for expansion of data storage requirements or computational functions.		
Possibility to easily add chemical processes to the model	No	Yes
Possibility to easily add processes other than chemical to the model	No	Yes
Generality (Flexibility, Reusability) - Attributes of the software that provide breadth of the functions performed.		
Possibility to use the model as a part of or in related systems	Yes	Yes
Possibility to use the model as a part of or in unrelated systems	No	No
Instrumentation (Maintainability, Testability) - Attributes of the software that provide for the measurement of usage or identification or errors.		
Useful error messages when debugging (using debug function or when simulating)	No	No
Warnings when a variable value is not normal	Yes	Yes
Model conditions (Correctness, Flexibility) - Simplifications, assumptions, valid domains, supplementary conditions to determine empirical consequences.		
Simulated models equations the same simplification level as the original models	Yes	Yes
Consideration taken in the model for situations when the model equations may be invalid	Yes	No
Modularity (Flexibility, Interoperability, Maintainability, Reusability, Testability) - Attributes of the software that provide a structure of highly independent modules.		
Possibility to use the model separately and incorporate it into another model or system	Yes	Yes
Possibility to use a part of the model separately and incorporate it into another model or system	No	Yes
Possibility to reuse the code in the model when adding new processes	No	Yes
Operability (Usability) - Attributes of the software that determine operation and procedures concerned with the operation of the software.		
Does the same variable only have to be changed in one place in the model?	Yes	Yes
One part of the model impossible to confuse or connect with another when creating the whole wetland model	Yes	No
Relevant values easily found for all variables and constants in the model from literature	Yes	Yes
Simplicity (Flexibility, Maintainability, Portability, Reusability, Testability) - Attributes of the software that provide implementation of functions in the most understandable manner.		
Variable names and labels easy to understand	Yes	Yes
Possibility to understand the model without programming and ecological knowledge	No	No
Systematics (Usability, Reusability) - Inner consistency, absence of contradictions, logical context.		
Each step of the model consequently implemented	Yes	Yes
Logical relationship between all parts of the model	Yes	Yes
Model free from contradictions in the equations	Yes	Yes
Traceability (Correctness) - Attributes of the software that provide a thread from the requirements to the implementation with respect to the specific development and operational environment.		
Prediction of total decrease in nitrogen in a wetland	Yes	Yes
Predictions of the decrease in ammonium nitrogen and nitrate nitrogen – i.e. the nitrification and denitrification processes in a wetland	No	Yes
Possibility to change parameters in the simulations	Yes	Yes
Possibility to plot chosen parameters in the simulations	Yes	Yes
Changes in concentration described as derivates with respect to a factor (time in wetland, distance in wetland, fraction of wetland etc)	Yes	Yes
Validity (Maintainability) - No or few systematic errors in the model. This could be theoretical validity (relevant variables and parameters etc) or empirical validity (make adequate prognoses etc).		
All variables used relevant for the application	Yes	Yes
All processes, reactions and relationships well-defined and well-used	Yes	Yes
Does the model produce the expected result compared with pre-calculated values when tested?	Yes	Yes

Appendix B. Comparison between evaluated tools. Specific elements are listed for each criterion.

	Power-Sim	Simile	Stella	WEST	Model Editor
Completeness (Correctness) - Attributes of the software that provide full implementation of the functions required.					
Representations of system parts, flows/connections, equations, variables and diagrams all present	Yes	Yes	Yes	Yes	Yes
Representing element can be other than the equivalent of a variable	No	No	No	Yes	Yes
Flows and connections can be integrated	No	No	No	Yes	Yes
Flows can go in both directions	No	No	Yes	No	Yes
Possibility of several variables in one flow	No	No	No	Yes	Yes
Possible to change and write new equations in components without programming skill	No	Yes	Yes	No	No
Built-in mathematical functions.	Yes	Yes	Yes	No	No
Variables can be defined in the elements representing a physical part of the system	No	No	No	Yes	Yes
Variables can be represented by other than separate components	No	No	No	Yes	Yes
Automatically generated graphs and tables	Yes	Yes	Yes	Yes	Yes
Consistency (Correctness, Reliability) - Attributes of the software that provide uniform design and implementation techniques and notation.					
Uniform design	Yes	Yes	Yes	Yes	Yes
Names and labels consistent in the entire environment	Yes	Yes	Yes	Yes	Yes
Data and communication commonality (Interoperability) - Attributes of the software that provide the use of standard data, protocols and interface representation.					
Possibility for easy import/export from related systems (not counting text and pictures)	Yes	Yes	No	Yes	Yes
Possibility for easy import/export from un-related systems (not counting text and pictures)	No	No	No	Yes	No
Expandability (Flexibility) - Attributes of the software that provide for expansion of data storage requirements or computational functions.					
Possibility to define and create new components	No	No	No	Yes	Yes
Several types of the different components/elements exists	No	No	Yes	Yes	Yes
Generality (Flexibility, Reusability) - Attributes of the software that provide breadth of the functions performed.					
Can be used in related systems (in present state)	Yes	Yes	Yes	Yes	Yes
Can be used in unrelated systems (in present state)	Yes	No	Yes	No	No
Instrumentation (Maintainability, Testability) - Attributes of the software that provide for the measurement of usage or identification or errors.					
Automatically showing modeling errors in model	Yes	No	Yes	No	No
Useful error messages when debugging (using debug function or when running)	Yes	Yes	Yes	No	No
Modularity (Flexibility, Interoperability, Maintainability, Reusability, Testability) - Attributes of the software that provide a structure of highly independent modules.					
Possibility to save separate and specific components of models	No	No	No	Yes	Yes
Uses submodels or groups of components that can be saved and used separately	Yes	Yes	No	Yes	Yes
Possibility to save models	Yes	Yes	Yes	Yes	Yes
Operability (Usability) - Attributes of the software that determine operation and procedures concerned with the operation of the software.					
Easy to use the different tools to create a model	Yes	Yes	Yes	Yes	Yes
Possibility to create simple models without programming skill (by using existing components)	Yes	Yes	Yes	Yes	Yes
Possible to create new components without programming skill (by using existing components)	No	No	No	No	No
Simplicity (Flexibility, Maintainability, Portability, Reusability, Testability) - Attributes of the software that provide implementation of functions in the most understandable manner.					
Easily overviewed modeling and simulation environment	Yes	Yes	Yes	Yes	Yes
Possible to use without looking in the manual	No	Yes	Yes	No	No
Training (Usability) - Attributes of the software that provide transition from current operation or initial familiarization.					
Useful help file	Yes	Yes	Yes	Yes	No
Useful manual (including links to easily accessed useful manual)	No	No	No	Yes	No