An integrated framework for knowledge-based modeling and simulation of natural systems

Jerzy W. Rozenblit and Piotr L. Jankowski
SIMULATION 1991 57: 152
DOI: 10.1177/003754979105700304

TECHNICAL ARTICLE

SIMULATION 57:3, 152 - 165

ISSN 0037-5497/91 $3.00 + .10
Printed in the United States of America

An integrated framework for knowledge-based modeling and simulation of natural systems

Jerzy W. Rozenblit
Department of Electrical and Computer Engineering
The University of Arizona
Tucson, Arizona 85721

Piotr L. Jankowski
Department of Geography
College of Mines and Earth Resources
University of Idaho
Moscow, Idaho 83843

This paper proposes a new approach to simulation modeling of natural systems in the context of water quality modeling in streams affected by point source pollution. The approach has a potential for application to other domains of natural resource modeling. Its conceptual basis is knowledge-based simulation and systems analysis. In the approach presented in this paper, a stream or its section is viewed as a collection of components, i.e. stream segments. The structure of a stream is its segments and their couplings. For each stream segment, a single constituent model describing processes affecting the segment's water quality is defined. Models are coupled in a hierarchical manner. The hierarchical, modular model specification results in a stream model comprised of a finite number of sub-segment models. Fundamental theoretical concepts supporting such a specification are described. A prototype simulation modeling environment to support prediction of water quality in streams has been implemented and tested. The proposed approach differs from other natural systems modeling frameworks in that it offers a) modular model specification facilities, b) high degree of model reusability, and c) support for model selection and coupling.

Keywords: Natural systems modeling, discrete event simulation, knowledge based simulation modeling.

Introduction

Modeling is an inductive process used frequently in a controlled task such as design of artifacts or in explaining the behavior of systems. Both analysis and design facets of the modeling enterprise can be characterized by a hierarchical approach that requires working downwards through the levels of specificity and synthesizing partial results into a coherent structure called a model. A number of methodologies and modeling/simulation systems have been developed to aid the modeling process in engineering domains [1; 10; 12; 13; 28; 29; 31; 32; 43]. However, few frameworks exist to support modeling of natural systems.

The framework used in modeling of one type of natural systems, i.e. water systems can be depicted as a sequence of steps shown in Figure 1. The two parts of Figure 1—a priori and a posteriori—indicate two possible approaches to water quality modeling [26]. The first, identified with a priori structural knowledge, follows a deductive reasoning approach in which one tries to deduce from an existing theory model relationships for a given problem. The second, identified with a posteriori empirical knowledge, follows an inductive approach in which one tries to develop a model from the sampled data. Ideally, these two approaches act as complementary stages of the modeling process. One follows all six steps with model calibration and model validation serving as an empirical test bed for an a priori model. Yet, in some situations characterized by difficulties in obtaining empirical data due to a budget and/or time constraints or preliminary scope of the analysis, the model specification may be reduced to the a priori stage [26].

In the traditional view of modeling the behavior of natural systems these six steps, although logically connected, comprise separate tasks [3 - 6]. Therefore, the development of an operational water quality model is a rather lengthy and expensive undertaking. There is a strong need for a more

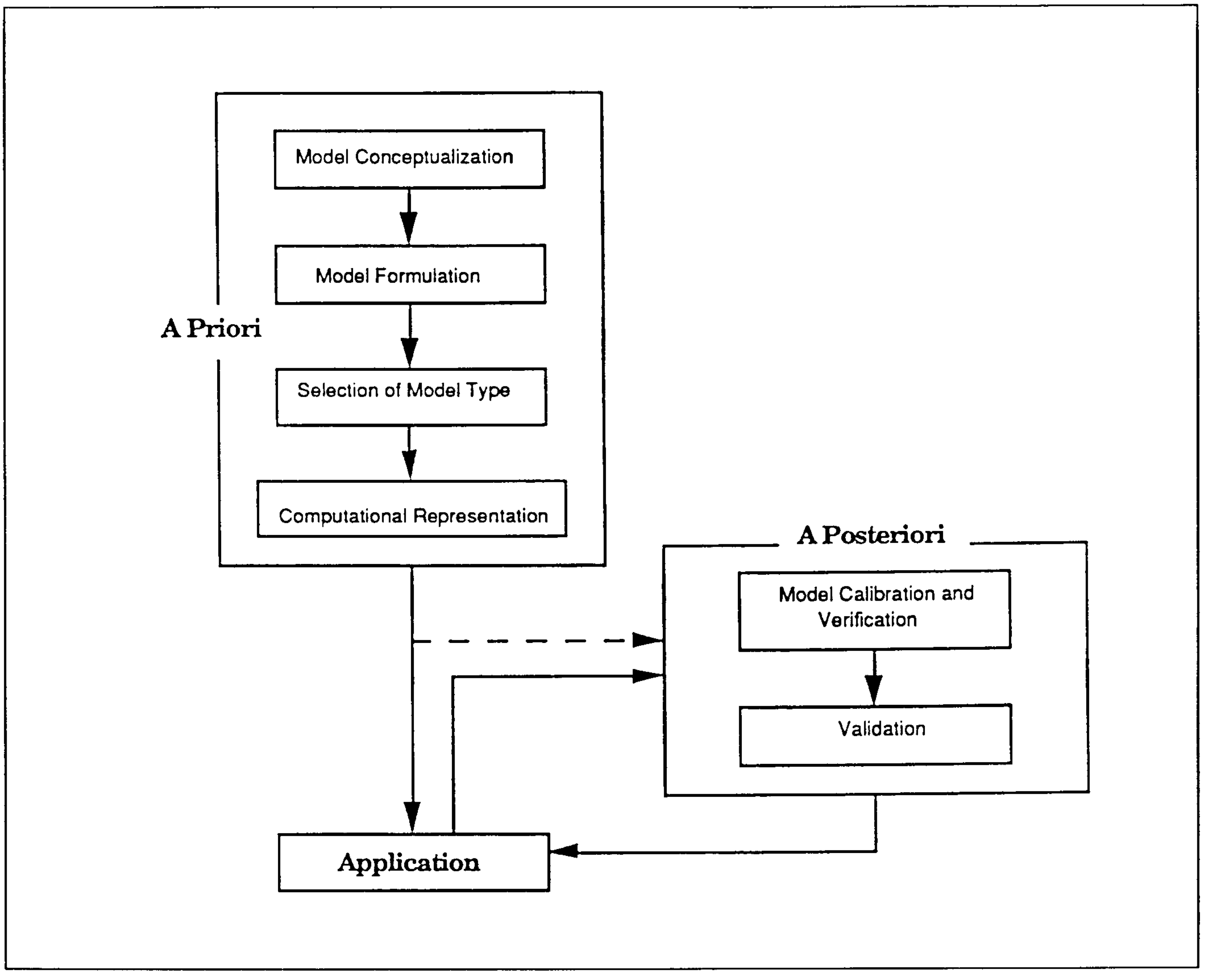


Figure 1. Water quality modeling process (adopted from Orlob *et. al.* 1983, p. 13 [26]).

integrated framework for modeling of water quality and natural systems in general that would better link the steps leading to model development and its implementation [2; 18; 19; 21; 22].

In this article, we demonstrate how Multifaceted Modeling and Knowledge-based Simulation [39] can integrate steps required to model stream water quality. Our work is motivated by the need to provide a more flexible — than the existing approaches — modeling framework for simulating changes in stream water quality in particular, and for predicting the behavior of natural systems in general. For the application domain, simulation of stream water quality has been selected since physical and biochemical processes that occur in streams and determine water quality are relatively well known and knowledge about them is reasonably well structured [7; 8; 18]. Yet, despite the well structured domain, the specification of an operational model for prediction of changes in the water quality of a stream, cannot be considered a simple and quick undertaking [2; 9; 23; 27; 34]. It still requires a considerable amount of human expertise and other resources (time and money). Therefore it is important to provide an integrated modeling framework which would help a modeler reduce time and effort required to specify an operational water quality model pertinent to a problem at hand.

This article is organized as follows: We first summarize the basic tenets of Multifaceted Modeling and Knowledge-based Simulation. Each phase of the approach is then illustrated with an application to stream water quality modeling. Simulation results are compared to the empirical ones. The comparison is favorable and it demonstrates the robustness of our modeling approach.

Multifaceted Modeling and Knowledge-based Simulation

Multifacetted methodology denotes a modeling approach which recognizes the existence of multiplicities of objectives and models in any simulation project. It provides formal representation schemes that support the modeller in organizing the model construction process, in aggregating partial models, and in specifying simulation experiments [39; 42]. Modeling objectives drive three fundamental processes in the methodology; they facilitate the representation of models' structures, retrieval, and manipulation of structures, the specification of models' behavior, and the specification of experimental conditions under which models are evaluated by a simulation study.

Overview

The model construction process begins with developing a representation of system components and their variants. To appropriately represent possible configurations of model components, we have proposed a representation scheme called the *system entity structure* (SES) [20; 39; 40]. The scheme captures the following three relationships: decomposition, taxonomy, and coupling. Decomposition knowledge means that the structure has schemes for representing the manner in which an object is decomposed into components. Taxonomic knowledge is a representation for the kinds of variants that are possible for an object, i.e., how it can be categorized and subclassified. The synthesis (coupling) constraints impose a manner in which components identified in decompositions can be connected together. The selection constraints limit choices of variants of objects determined by the taxonomic relations.

Beyond this, procedural knowledge is available in the form of production rules. They can be used to manipulate the elements in the system domain by appropriately selecting and synthesizing the domain's components. This selection and synthesis process is called *pruning* [17; 29; 30]. Pruning results in a recommendation for a *model composition tree*, i.e. the set of hierarchically arranged entities corresponding to model components. A composition tree is generated from the system entity structure by selecting a unique component for specializations and a unique decomposition for an entity with several decompositions.

The final step in the framework is the evaluation of models derived from composition trees. Model behaviors can be expressed in special formalisms depending on the problem at hand. Typical specifications include differential equations, finite state machine, or discrete event. Each formal model description specifies a system and selects a class of subsystems by placing constraints on the possible static and dynamic structures it encompasses. A characterization of such constraints is given in [39; 40]. The model construction process involves the specification of the static and dynamic structure. Discrete Event System Specification (DEVS) [39 - 42] is a modeling formalism used for model specification in our approach. DEVS provides a formal representation of discrete event systems. It is closed under coupling. This property facilitates the construction of hierarchical DEVS network specifications.

Performance of models is evaluated through computer simulation in the DEVS-Scheme environment [20; 31; 41; 42]. DEVS-Scheme is an object-oriented simulation shell for modeling and design that facilitates construction of families of models specified in the DEVS formalism. Alternative models are evaluated with respect to experimental frames that reflect model performance questions. Results are compared and traded off in the presence of conflicting criteria. This results in a ranking of models and supports choices of alternatives best satisfying the set of modeling objectives.

Model Structure Representation—System Entity Structure

As a step toward a complete knowledge representation scheme for modeling support, we have combined the decomposition, taxonomic, and coupling relationships in a knowledge representation scheme called the *system entity structure* (SES). Knowledge representation is generally accepted to be the key ingredient in artificial intelligence software. Previous work [39; 40; 41] identified the need for representing the structure and behavior of systems, in a declarative scheme related to frame-theoretic and object-based formalisms [42]. The elements represented are motivated, on the one hand, by systems theory [16; 24; 25; 38] concepts of decomposition (i.e. how a system is hierarchically broken down into components) and coupling (i.e. how these components may be interconnected to reconstitute the original system). On the other hand, systems theory has not focused on taxonomic relations, as represented for example in frame-hierarchy knowledge representation schemes. In the SES scheme, such representation concerns the admissible variants of components in decompositions and the further specializations of such variants.

A system entity structure is a labeled tree. Nodes of the tree are classified as entities, aspects, specializations, and multiple decompositions. Variables can be attached to nodes. They are called *attached variables types*. An *entity* signifies a conceptual part of the system being represented by the entity structure. An *aspect* is a mode of decomposing an entity. A *specialization* is a mode of classifying an entity. An entity may have several specializations (and/or decompositions); each specialization (decomposition) may have several entities. The original entity is called a *general type* relative to the entities of a specialization. The entities of a specialization are called specialized types. Since each entity may have several specializations, a hierarchical structure called taxonomy results. A *multiple decomposition* is a means of representing varying number of entities. An *attached variable type* is an attribute of an object represented by the entity with which the variable type is associated.

Figure 2 depicts a high level view of the entity structure for stream water quality modeling. The root entity, named "Stream Water Quality Model", denotes the model of a river basin or a section of it. It has one attached variable "model constituent" whose legal values are: algal, BOD-DO (biochemical oxygen demand—dissolved oxygen deficit), nitrogen, phosphorous, and thermal. Each value of the "model constituent" variable acts as a pointer to one of the five specialized entities: "Algal Constituent Model", "BOD-DO Constituent Model", "Nitrogen Constituent Model", "Phosphorus Constituent Model", and "Thermal Constituent Model". The specialized entities are in turn decomposed along a segmentation aspect into k-entities corresponding to abstract atomic segment-models. The segmentation aspect represents topological, hydrological, geomorphic, and biological criteria upon which a river or its section is divided into segments. The number of entities representing abstract

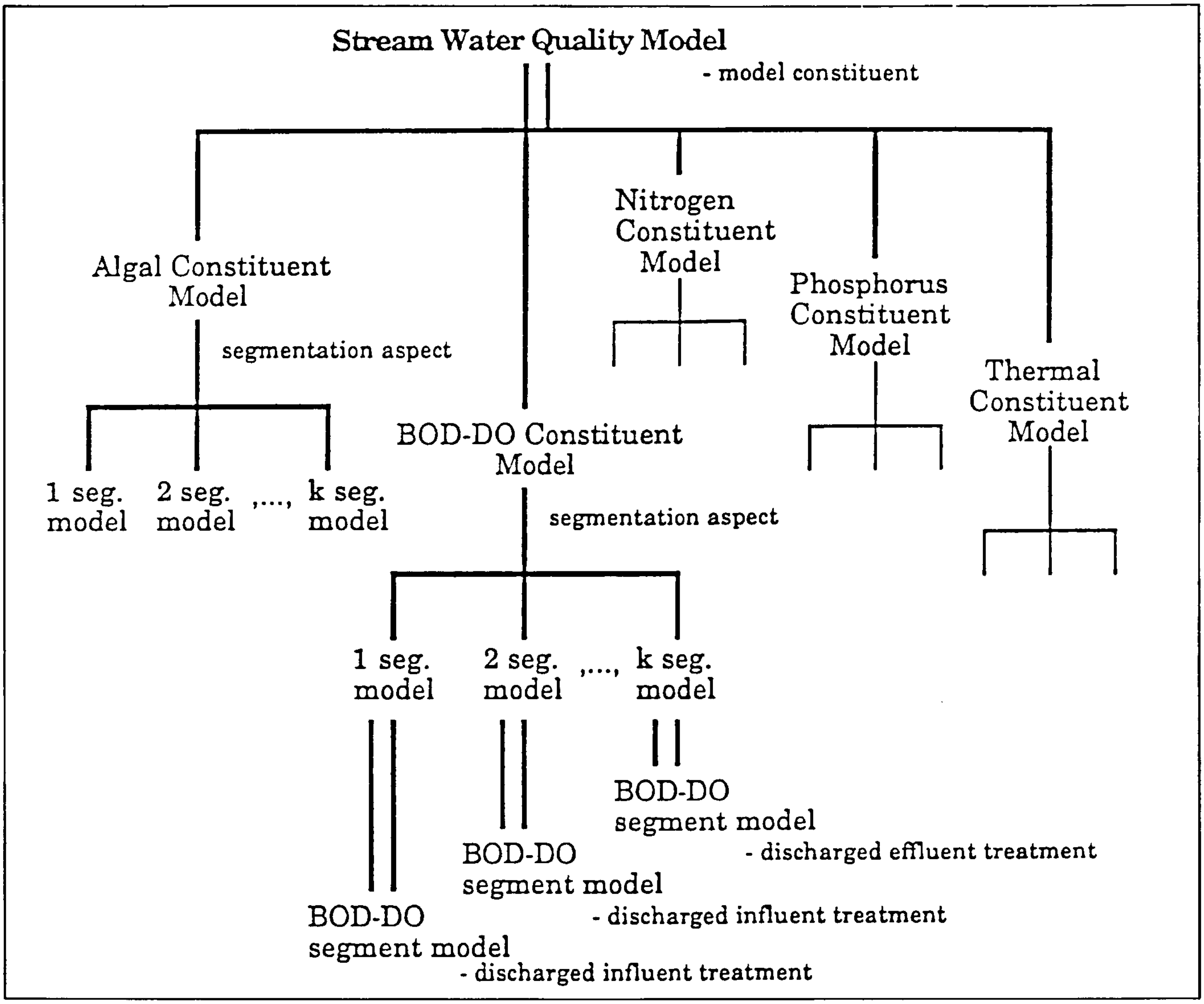


Figure 2. Abstract model of stream segment.

segment-models is a variable element in the presented entity structure. The number (k) has to be assigned by a user for each particular case, that is, for each modeled river or its section. The segment assignment is based upon segmentation criteria [18].

Each of the k-entities representing abstract atomic segment-models can be specialized into an entity denoting a constituent-specific type of atomic segment-model, for example, algal segment-model, BOD-DO segment model, etc. A variable "discharged effluent treatment" attached to the "BOD-DO segment model" entity can receive the values: none, mechanical, biological.

The system entity structure organizes a variety of system decompositions and, consequently, a variety of model constructions. Its generative capability facilitates convenient definition and representation of models and their attributes at multiple levels of aggregation and abstraction. More complete discussions of the system entity structure and its associated structure transformations are presented in [28; 30; 31; 32; 39].

Rule-based System Entity Structure Pruning

In Multifacetted Modeling, a model is synthesized from components stored in the model base. A synthesis specification is the result of *pruning* a substructure from the system entity structure. Pruning results in a model structure candidate for a best match to the set of modeling objectives. It can be viewed as a search through the space of candidate solutions to the problem. Production rules represent the knowledge consisting of modeling objectives, coupling constraints, user's requirements and performance expectations. The aim of pruning is to recommend plausible candidates for an optimal solution to the problem (with respect to the requirements and constraints). More detail can be found in [17; 20; 30].

The following steps are required to provide the rules that guide pruning of the system entity structure: 1) for each specialization, specify a set of rules for selecting an entity; 2) for an entity with several aspects, specify rules for selecting a unique aspect; 3) for each aspect specify rules that ensure that the entities selected from specializations are configurable, i.e. the components they represent can be validly coupled. The above rule sets constitute a knowledge base for the inference engine that prunes a system entity structure for a particular application domain. Pruning generates a model composition tree, i.e. a structure that contains all the information needed to synthesize a model in a hierarchical fashion from its atomic model components. To support the model construction process, we have available a set of software tools that are currently being integrated on AI workstations and PCs. An expert system shell MODSYN (MODel SYNthesizer) [17; 30] to generate model structures was developed and implemented.

DEVS-Scheme Modeling and Simulation Environment

DEVS-Scheme is a simulation environment that synthesizes simulation models from a composition tree specification [39]. It thus serves as the modeling and simulation layer underpinning the multifacetted methodology. In this article, we can only provide a brief review of the DEVS (Discrete Event System Specification Formalism) and its implementation in DEVS-Scheme. More detail is available in [20; 39; 42].

The Discrete Event System Specification (DEVS) formalism introduced by Zeigler [39] provides a means of specifying a mathematical object called a system. Basically, a system has a time base, inputs, states, outputs, and functions for determining next states and outputs given current states and inputs. The insight provided by the DEVS formalism is in the simple way that characterizes how discrete event simulation languages specify discrete event system parameters. Having this abstraction, it is possible to design new simulation languages with sound semantics that is easier to understand.

DEVS-Scheme, an implementation of the DEVS formalism in Scheme (a Lisp dialect), supports building models in a hierarchical, modular manner. This is a systems oriented approach not possible in popular commercial simulation languages such as Simscript, Simula, GASP, SLAM and Siman (all of which are discrete event based) or CSMP and ACSL (which are for continuous models).

Basic Models

In the DEVS formalism, one must specify: 1) basic models from which larger ones are built, and 2) how these models are connected together in hierarchical fashion. In this formalism basic models are defined by the structure:

$$M = < X, S, Y, \ \delta, \lambda, ta >$$

where: X is the set of external input event types, S is the sequential state set, Y is the set of external event types generated as output, δ_{int} (δ_{ext}) is the internal (external) transition function dictating state transitions due to internal (external input) events, λ is the output function generating external events at the output, and ta is the time-advance function. Rather than reproduce the full mathematical definition here [39], we proceed to describe how it is realized in DEVS-Scheme.

To specify modular discrete event models requires that we adopt a different view than that fostered by traditional simulation languages. As with modular specification in general, we must view a model as possessing input and output ports through which all interaction with the environment is mediated. In the discrete event case, events determine values appearing on such ports. More specifically, when external events, arising outside the model, are received on its input ports, the model description must determine how it responds to them. Also, internal events arising within the model change its state, as well as manifest themselves as events on the output ports to be transmitted to other model components.

A basic model contains the following information:

- the set of input ports through which external events are received
- the set of output ports through which external events are sent
- the set of state variables and parameters
- the time advance function which controls the timing of internal transitions
- the internal transition function which specifies to which next state the system will transit after the time given by the time advance function has elapsed
- the external transition function which specifies how the system changes state when an input is received; the next state is computed on the basis of the present state, the input port and value of the external event, and the time that had elapsed in the current state.
- the output function which generates an external output just before an internal transition takes place.

Coupled Models

Basic models may be coupled in the DEVS formalism to form a multi-component model which is defined by the structure:

$$DN = < D, M_i, I_i, Z_{i,j}, SELECT >$$

where:

D: is a set of component names;

for each i in D,

M_i: is a component basic model

I_i: is a set, the influencees of i

and for each j in I_i,

$Z_{i,j}$: is a function, the i–to–j output translation

$SELECT$: is a function, the tie-breaking selector.

Multi-component models are implemented in DEVS-Scheme as coupled models. A coupled model, tells how to couple (connect) several component models together to form a new model. This latter model can itself be employed as a component in a larger coupled model, thus giving rise to hierarchical construction. A coupled model contains the following information:

- the set of components
- for each components, its influencees
- the set of input ports through which external events are received

- the set of output ports through which external events are sent

The coupling specification consists of:

- the external input coupling which connects the input ports of the coupled model to one or more of the input ports of the components
- the external output coupling which connects output ports of components to output ports of the coupled model
- the internal coupling which connects output ports of components to input ports of other components
- the select function which embodies the rules employed to choose which of the imminent components (those having the minimum time of next event) is allowed to carry out its next event.

A multi-component model *DN* can be expressed as an equivalent basic model in the DEVS formalism [39]. Such a basic model can itself be employed in a larger multi-component model. This shows that the formalism is closed under coupling as required for hierarchical model construction.

Model Construction—Summary

We begin model construction process by conceptualizing decompositions and specializations of components of the system being modeled. To do this, we use system entity structuring tools. We utilize the system entity structure base as a repository of previous modeling experience. Thus, we may retrieve an entity structure from this base which is applicable to the modeling domain at hand. Such an entity structure is modified and enhanced with entities required in the new project. Models associated with new atomic entities must be developed and placed in the model base.

A rule base is developed and used in the pruning process. The pruning engine generates model composition trees. For each component atomic model represented by the tip node of the composition tree, DEVS specification is defined.

To carry out a simulation experiment, we specify an experimental frame, i.e. the set of circumstances under which we observe the behavior of the model. This is accomplished by defining DEVS-Scheme components that: a.) generate input stimuli to the model, i.e., discrete event input segments; b.) observe model output; and c.) control the simulation experiment by observing model variables. A detailed presentation of the experimental frame concepts is given in [28; 29; 39; 42].

The above framework is now illustrated in a natural system modeling example, namely, in simulation modeling of stream water quality.

System—Analytical Specification of Stream for Discrete Event Simulation Modeling

In systems analysis terms, the system under study, a stream or a section of a stream, can be viewed as a collection of component stream segments [35; 36]. Such a collection can be abstracted as a set *S* and its components as $\{s_1, s_2, s_3, \ldots, s_k\}$, where $s_i \neq s_k$ } and $i, j \in (1, 2, 3, \ldots, k)$. A structure of the studied system can be represented by the set *S* and a coupling of its components [35; 36]. A coupling of the system components can be defined as a structure:

$$N = < S, I_l, M_l >$$

where in the case of a non-branching, single stream:

I_l is a set of influencees of s_i components, $i \in (1, 2, 3, \ldots, k)$, $l \in (1, 2, 3, \ldots, k)$, M_l is a mapping function that connects the s_{i-1} component with the s_i component.

In a more general case of a branching stream channel, M_l is a mapping function that connects multiple s_j influencees, where $j < i$, with the s_i component.

A coupling scheme can be defined by a pair of sets $< I_l, M_l >$. The scheme for coupling system components can also be perceived as a non-branching network representing interconnections of stand alone elements to form a higher level system. A non-branching network is an appropriate model for the coupling scheme of stream segments.

A single component of the system under study — a segment s_i — can be analyzed in a similar way as the stream; its structure can be considered as consisting of sub-segment-components and their coupling scheme. Such a system decomposition (coupling scheme and component specification) can continue to some finite level as determined by the scale of analysis. Consequently, one creates a hierarchical system specification. Consider a hierarchical specification of a stream system with one decomposition. In such case, the segment component s_i is the terminal component of the system under study. A convenient way of representing a stream segment is through its model. The concept of model is understood here as a mathematical representation of a system (or system's component) intended to emulate its responses to input stimuli. The mathematical representation of a system is also its abstraction intended to provide a less complex representation of the system structure.

For our analysis, a sufficiently general model of a stream segment can be defined as the following structure:

$$A = < X, Y, S, \delta, \lambda >$$

where:

X is the set of inputs,
Y is the set of outputs,
S is the set of internal states,
δ is the transition function, and
λ is the output function.

A model described by the structure "A" represents an open system since it maintains flows (inputs and outputs) across its boundary as depicted in Figure 3. The input set *X*, where $X = (x_1, x_2, x_3, \ldots, x_n)$ and $x_i \in R$, represents the part of the interface through which the environment communicates with the system. The x_i elements of *X* are the river characteristics downstream from a given segment. Conversely, the output set *Y*, where $Y = (y_1, y_2, y_3, \ldots, y_n)$ and $y_i \in R$, represents the part of the interface through which the system communicates with the environment.

The inputs are mapped into system internal states represented by state variables, by the transition function:

$$\delta : X \to S$$

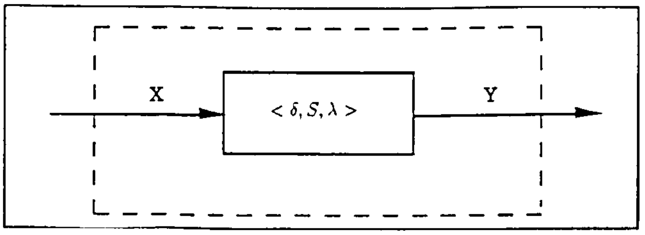


Figure 3. Abstract model of stream segment.

and transferred outside the model by an output function:

$\lambda : S \rightarrow Y$

The model "A" can be called *atomic* to indicate that it is not further decomposed. In the system under study, the elements of X and Y can be, for example, river flow volume, tributary flow volume, sewage inflow volume, values of water quality indicators, and parameters describing the level of biochemical reactions in the stream water. Transition of X into S can be expressed in the form of a differential equation which returns the output Y.

At the higher level of system specification one can consider a number of atomic models described by the A structure, coupled into the non-branching network of stream segment models as illustrated in Figure 4. Such a network forms a coupled model C which consists of atomic models. Linking of atomic models is realized by the interface mapping function. At the next higher level of system specification, one can consider a number of coupled models, where each model represents a stream component of a river basin, aggregated into a hierarchically coupled model. Such a model represents a branching network of a river basin (Figure 5).

The coupled model of a system can be defined as a structure:

$C = < \{\mathbf{X}, \mathbf{Y}, \mathbf{M}\}, \phi >$

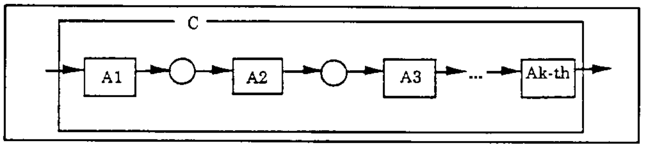


Figure 4. Network structure of the coupled model C representing a single, non-branching stream (circles symbolize the interface mappings).

where:

$\mathbf{X}$ is the set of input sets, $X \in \mathbf{X}$,
$\mathbf{Y}$ is the set of output sets, $Y \in \mathbf{Y}$,
$\mathbf{M}$ is a set of models, where models can be of atomic type, $A, A \in \mathbf{A}$,
and of coupled type $C, C \in \mathbf{C}$,
ϕ is the interface mapping function.

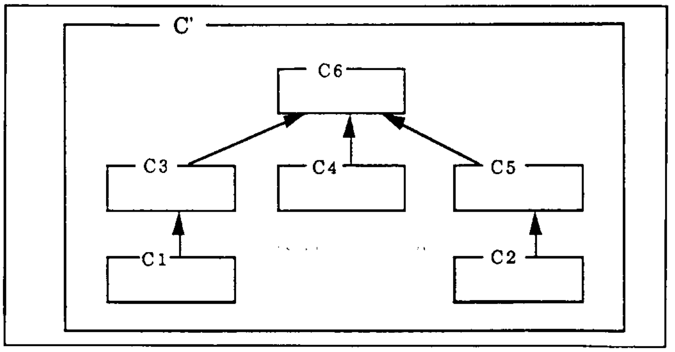


Figure 5. Coupled model C representing a tree structure of the branching network of a river basin.

The role of the mapping function ϕ is to translate outputs into inputs (interpreted as flows from one segment to the next).

System Entity Structure Specification of Stream Water Quality Models

The entity structure for a single-constituent water quality model represents a model space from which alternative configurations of a single-constituent stream water quality model can be derived. A prototype of the entity structure tree for the stream water quality model is presented in Figure 6. The root entity—named "Stream Water Quality Model"—denotes a single-constituent model of a river basin or a section of it. It is specialized into five entities: "Algal Constituent Model", "DOD-DO Constituent Model", "Nitrogen Constituent Model", "Phosphorus Constituent Model", and "Thermal Constituent Model".

Each specialized entity has the attached variable *number of segment models* which, together with the multiple specialization, represent the number of segments partitioning upon the segmentation criteria (topological, hydrological, geomorphic, and biological) for a given stream, and the number of single-segment, single-constituent models.

Each of the five entities can be specialized into subentities denoting a constituent-specific type of atomic segment-model. In this paper, only the subentities resulting from the specialization of the "BOD-DO Constituent Model" entity are further specialized. This is sufficient to demonstrate the validity of the proposed entity structure for the entire modeling domain. The entity "BOD-DO Constituent Model" representing a BOD-DO model type is specialized into two entities: "carbonaceous BOD-DO models" denoting those BOD-DO models which take into account only the carbonaceous oxidation of the organic matter contained in the discharged sewage, and "carbonaceous and nitrogenous BOD-DO models" representing those BOD-DO models which account for the carbonaceous and nitrogenous oxidation of organic waste components. A variable "discharged effluent treatment" attached to the "BOD-DO Constituent Model" entity can receive the values: none, mechanical, biological. These values play the role of pointers to two specialized entities "carbonaceous BOD-DO models" and "carbonaceous and nitrogenous BOD-DO models". The "carbonaceous BOD-DO models" entity is further specialized into a segment-specific atomic model. Similarly, the "carbonaceous and

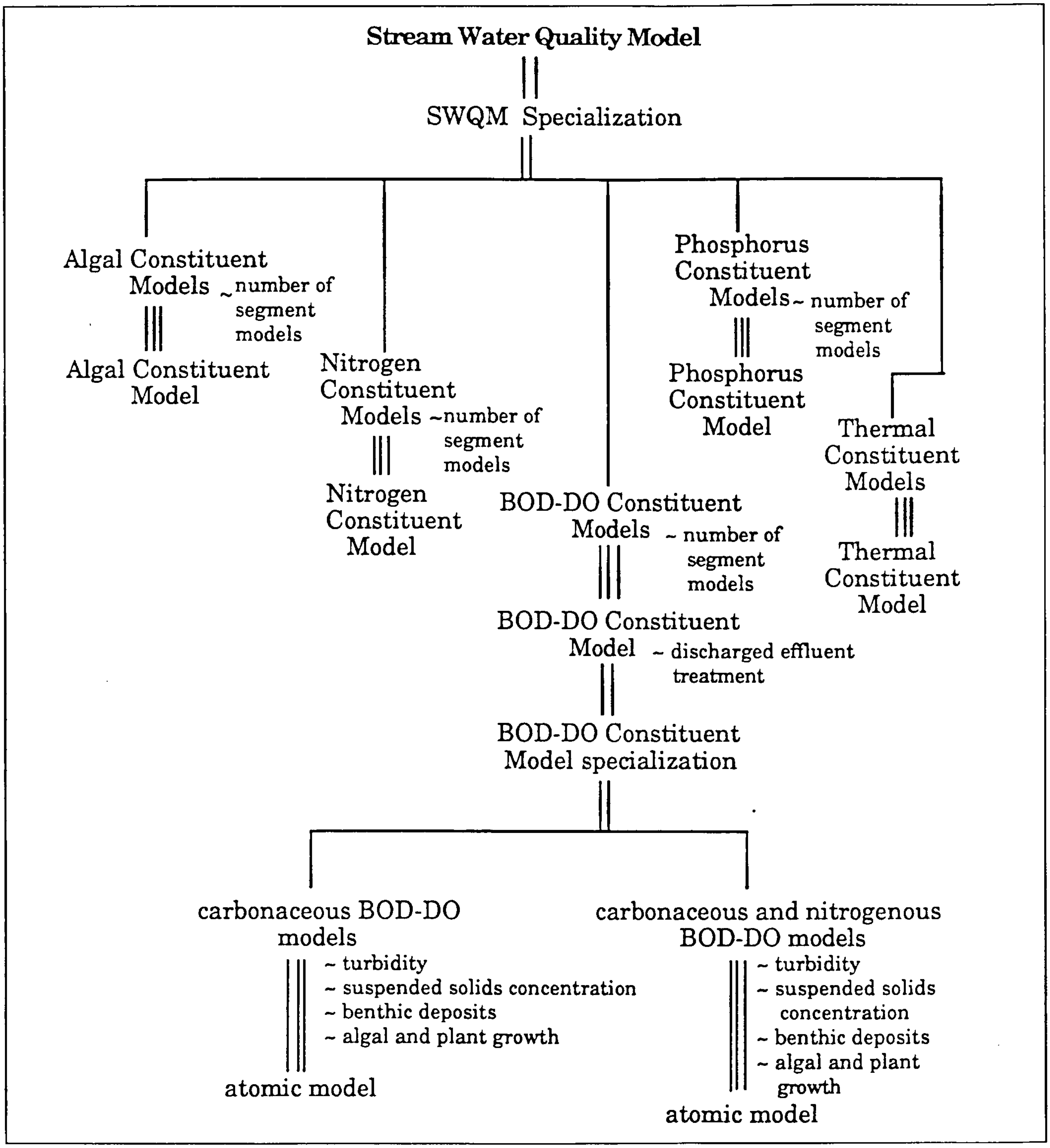


Figure 6. System entity structure for single constituent stream water quality models.

nitrogenous BOD-DO models" entity is specialized into a segment-specific atomic model representing carbonaceous-nitrogenous BOD-DO atomic segment-model.

The entities depicted in the entity structure tree represent component modeling knowledge at the conceptual level. The equivalents of the leaf entities (lower-level entities) at the computer implementation level are stored in the model base and are called *atomic models*. They are the primitives from which the model description of a system is assembled. The atomic models can be expressed in a special formalism depending on the problem at hand. Typical specifications include differential equations, finite difference equations, or discrete event system specification. In the proposed approach to stream water quality modeling, we employ the discrete event specification of atomic models based on DEVS [39]. The DEVS-formulated atomic models are retrieved from the model base as the leaf entities of the composition tree. Linking of the retrieved atomic models into a system model is accomplished through coupling and is guided by constraints.

Pruning of the System Entity Structure for Stream Water Quality Modeling

In order to choose an appropriate atomic segment-model for each given segment, we employ rule based pruning. The following attributes expressed by attached variables (see Figure 6) are used in this process:

- discharged effluent treatment,
- turbidity,
- suspended solids concentration,
- benthic deposits,
- algal and plant growth.

The five attached variables can take on the following values:

- discharged effluent treatment: < none, mechanical, biological >,
- turbidity: < low, high >,
- suspended solids concentration: < low, high >,
- benthic deposits: < low, medium, high >,
- algal and plant growth: < low, medium, high >.

The set of categorical values assigned to each attached variable consists of collapsed — classified continuous measures. These categorical values are a special case of ordinal values. The number of allowed variable values is kept intentionally low in order to reduce the overlapping character of qualitative values. The price paid for it is a partial coverage of the observable level range of the attached variables.

The heuristics involving the combinations of values received by the attached variables can be conveniently expressed in *production rules*. Production rules comprising the rule base of the model management system prototype are used for pruning of the entity structure tree. Typical pruning rules can be specified as follows:

```
Rule #1:
IF discharged effluent treatment = none OR
  discharged effluent treatment = mechanical
THEN BOD-DO segment-model = carbonaceous BOD-DO model

Rule #2:
IF BOD-DO segment model = carbonaceous BOD-DO model AND
  turbidity = low AND
  concentration of suspended solids = high AND
  benthic deposits = high AND
  algal and plant growth = high
THEN recommended atomic segment-model = "2-SP"
```

An example of the result of such pruning – a composition tree for the entity "1 seg. model" – is depicted in Figure 7.

Pruning of the entity structure tree with respect to "1 seg. model" entity is presented in Figure 7 in two consecutive steps. In the first step, depicted on the left side of the figure, specializations remain with singular entities in each specialization. In the second step, (the right side of the figure) specializations are removed resulting in the composition tree. In removing specializations, the terminal entity "2-SP" (a reference to 2-SP model) replaces first the "carbonaceous BOD-DO model" entity and then the "BOD-DO segment model" entity. The "BOD-DO constituent model" replaces "Stream Water Quality Model" forming the "BOD-DO Stream Water Quality Model" root entity.

After repeating the pruning operation k-1 times we assign a specific atomic-model for each of the k segments. A model corresponding to the root entity in Figure 7 (the BOD-DO Stream Water Quality Model) is derived by coupling atomic segment-models represented in the composition tree. This step is called model synthesis. In our framework, the model synthesis process is carried out according to the coupling scheme presented in Figure 8.

Entities "1 seg. mode", "2 seg. model",...,"k seg. model" serve as aliases which are replaced by specific atomic segment-model templates from the model base. The coupling scheme for the SWQM root entity is consistent with the coupled model specification. Model synthesis results in a coupled model of the given stream (or its section) which is simulated in DEVS-Scheme.

Implementation

The entity structure tree for modeling of stream water quality was implemented in the DEVS-Scheme software environment [18]. The DEVS model definition is compatible with the network specification of the model of a stream or its section presented before. Hence, an atomic segment-model template employed to calculate values of water quality indicators for the given segment is a discrete event model specified as follows:

- X is the set of constituent values and parameters charac terizing the sewage effluent and the quality of upstream water entering a segment (as well as values of biochemical reaction rates pertinent for the segment)
- S is the set consisting of two states: *active* – when the model is engaged in calculations upon receiving an exter nal input x, and *passive* – after the results have been transferred to the next contiguous, as defined by the coupling scheme of Figure 8, atomic segment-model and the next external input has not yet arrived
- δ_{int} changes the active state to the passive state after an output event has occurred
- δ_{ext} calculates constituent values for the set of spatially oriented (along the longitudinal dimension of the water flow) points inside of the segment, upon receiving the initial constituent values
- ta advances the time for which a model is allowed to stay in an active state without an occurrence of an external input event

DEVS is closed under coupling. Thus, the coupled model of a steam or its section consisting of the DEVS atomic

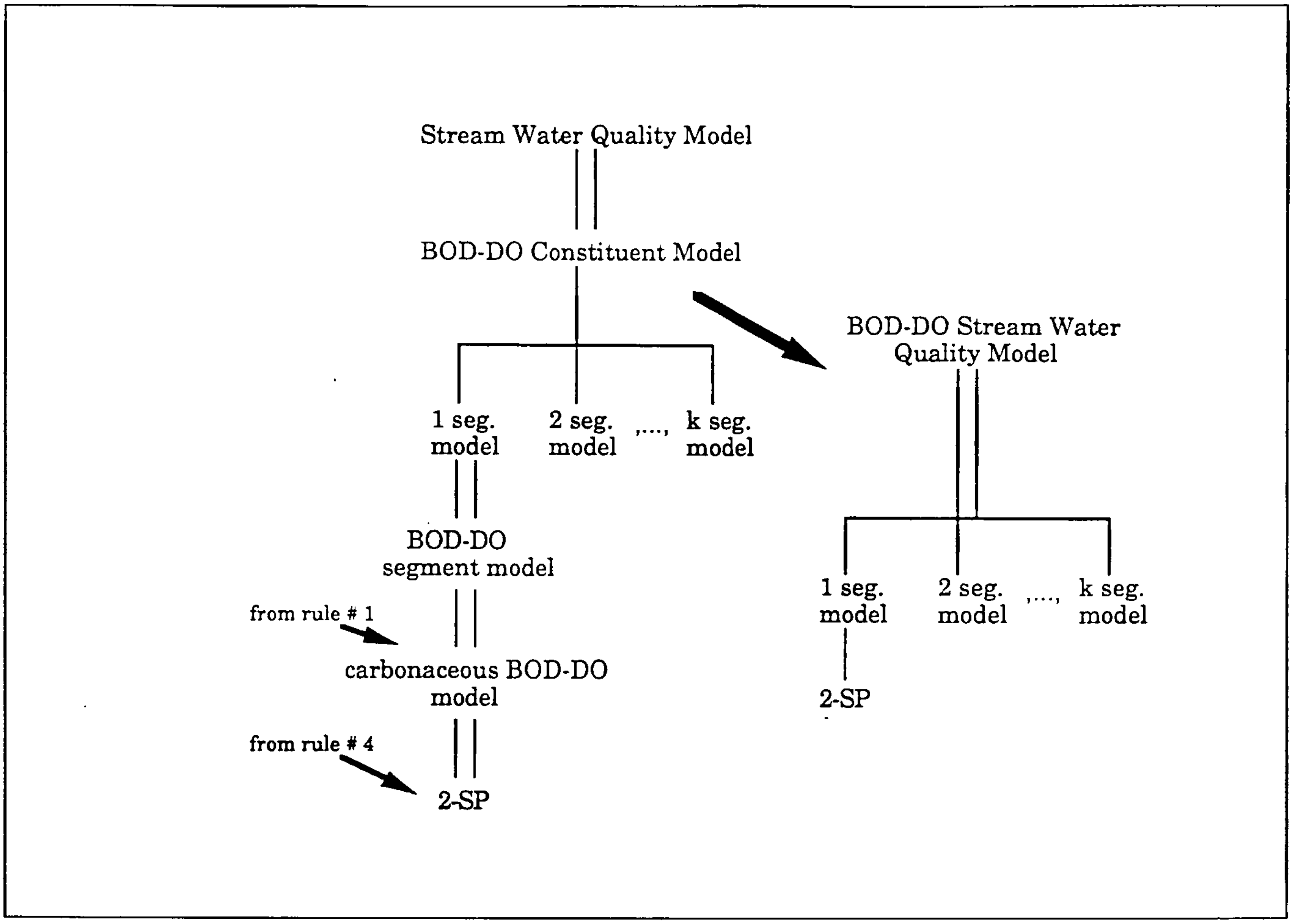


Figure 7. Pruning stream water quality model entity structure with respect to "1 seg. model" entity.

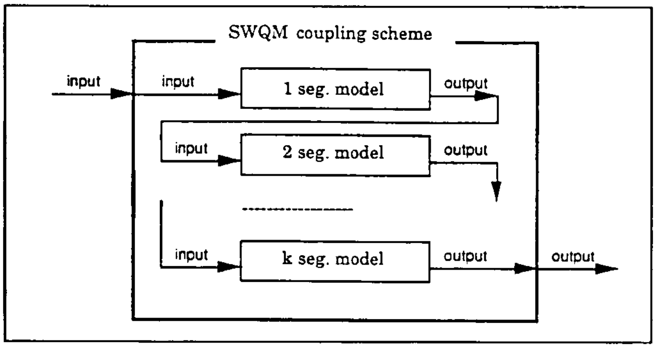


Figure 8. Coupling scheme for stream water quality models (SWQM).

segment-models is also a DEVS model. Figure 9 presents the DEVS-Scheme implementation of the entity structure tree depicted in Figure 7.

Pruning was performed with respect to the BOD-DO Constituent Model. Comparing Figure 9 with Figure 7 one can notice some differences between the conceptual representation (Figure 7) and its implementation (Figure 9). In the DEVS-Scheme implementation of the pruned entity structure tree the root entity "SWQM Simulator" is decomposed into "BOD-DO constituent-model" and "experimental frame" entities. The "BOD-DO constituent model" entity is further decomposed into two entities representing atomic segment-models: "carbonaceous BOD-DO models", "carbonaceous and nitrogenous BOD-DO models", and a model coordinator labeled as "coordinator". The multiple decomposition of the entities carbonaceous and nitrogenous BOD-DO models represents the multiple instances of both models. The experimental frame consists of an input segment generator and output transducer.

The entity "coordinator" in Figure 9 represents an atomic model whose function is to coordinate a sequence of inputs and outputs of the atomic segment-models during the simulation. The function of the model coordinator is depicted in Figure 10.

The coordinator receives input values from the generator. The input values represent various conditions of water quality, upstream from the first segment of the given stream or its section, expressed by dissolved oxygen deficit (DOD) and bio-chemical oxygen demand (BOD). The input value pairs (DOD, BOD) are transmitted from the coordinator through the output port X1 to the first segment-model's input port "in". Values of DOD and BOD, for the endpoint of the first segment, are then calculated by an atomic segment-model selected for the "1 seg. model". Subsequently, the

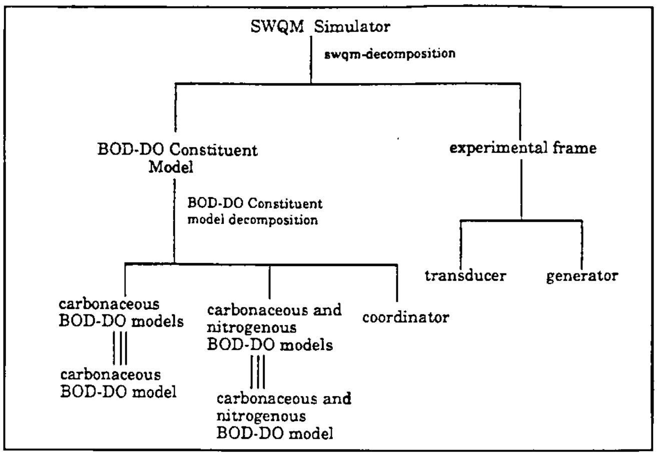


Figure 9. Composition tree underlying DEVS-scheme simulator of SWQM.

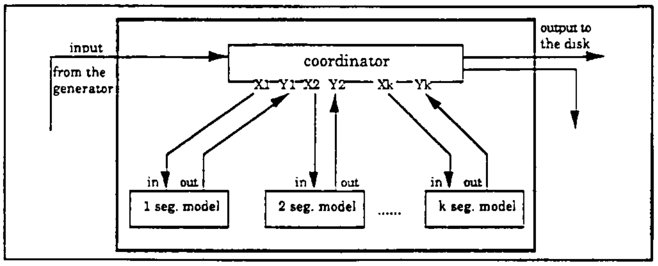


Figure 10. Specification of model coordinator.

calculated (DOD,BOD) pair is sent out of the"1 seg. model" through the port "out" and received by the coordinator on the input port "Y1". The coordinator transmits the pair of values (DOD,BOD) to the next segment-model, "2 seg. model", through the output port "X2" and simultaneously sends it to a log-file on the disk. The process continues until the last pair (DOD,BOD) is calculated for the endpoint of the last k-th segment. The last value pair (DOD,BOD) is sent to a file and also to the transducer - a component of the experimental frame.

Simulation Results

Preliminary simulation tests were run to assess the performance of a DEVS-Scheme based specification for stream water quality models. A 222 km long section of the Warta River, the main tributary of Odra River and the third largest river in Poland, was used for the test. The river section was divided into six segments ranging in length from 22.5 km to 68 km [18]. This division was based on hydrologic, geomorphic, and biologic criteria such as point of waste discharge, point of abrupt change in flow conditions, area of noticeable change in biological conditions (e.g. water plant or algal growth, benthic deposits).

The simulation was run for the initial value pair $BOD = 17$ $[g/m^3]$ and $DOD = 8$ $[g/m^3]$, reflecting the average water quality conditions for the summer period, upstream from the first segment of the section. The initial BOD and DOD values reflect the high level of water pollution in the river. The results of the simulation runs were then compared with the observed values of water quality indicators and values obtained by simulating a traditionally specified single-constituent model [18]. The results are presented in Table 1 and are graphically illustrated in Figure 11. The maximum error for a stream segment for BOD was slightly over −8%, while the cumulative error across all segments for BOD was −4.6%. Respective values for DOD were −2.4% and +1%. These results, however preliminary, suggest that a DEVS-Scheme hierarchical and modular model specification is a robust approach to modeling and simulation of stream water quality.

Table 1. Calculated and Observed Values of BOD and DOD for the Set of Calibrated Rates.

distance from river mouth [km]	seg. No.	simulation results		observed values	
		BOD	DOD	BOD	DOD
456	1	13.4	4.0	13.7	4.1
418.5	2	13.4	4.0	13.6	4.0
389.5	3	12.1	3.6	12.3	3.5
348.5	4	11.4	4.1	11.4	4.2
280.5	5	13.5	3.4	13.9	3.3
258	6	14.5	3.9	14.0	3.9

Conclusions

The focus of this paper has been a framework for modeling support of stream water quality prediction. Pivotal to the approach presented here are the concepts of hierarchically specified models. Models may have several submodels represented and managed by the system entity structure. The entity structure in which all components are encoded in the form of discrete event models constitutes a modeling knowledge representation scheme. The goal-driven pruning of the structure is the basis for model selection and composition.

The approach described in this paper differs from other techniques presented in the literature [2 – 6], [11; 14; 26]. Conventional stream water quality modeling focuses on the development of mathematical models. Models are usually large, comprehensive, multiconstituent and multiple equation mathematical structures accounting for detailed processes influencing surface water quality. If properly calibrated, they can produce very accurate predictions of changes in water quality, and serve as short–, medium–, and long–term planning and management tools. However, such models require much input data which can be time-consuming and expensive to gather. Moreover, the data input into to the model and the interpretation of the output results can be cumbersome and difficult for a novice user. Our premise is that instead of centering a computer-aided modeling system around one large, multiconstituent water quality model, it is advantageous to create a model base containing atomic models reflecting basic components of the domain. The base,

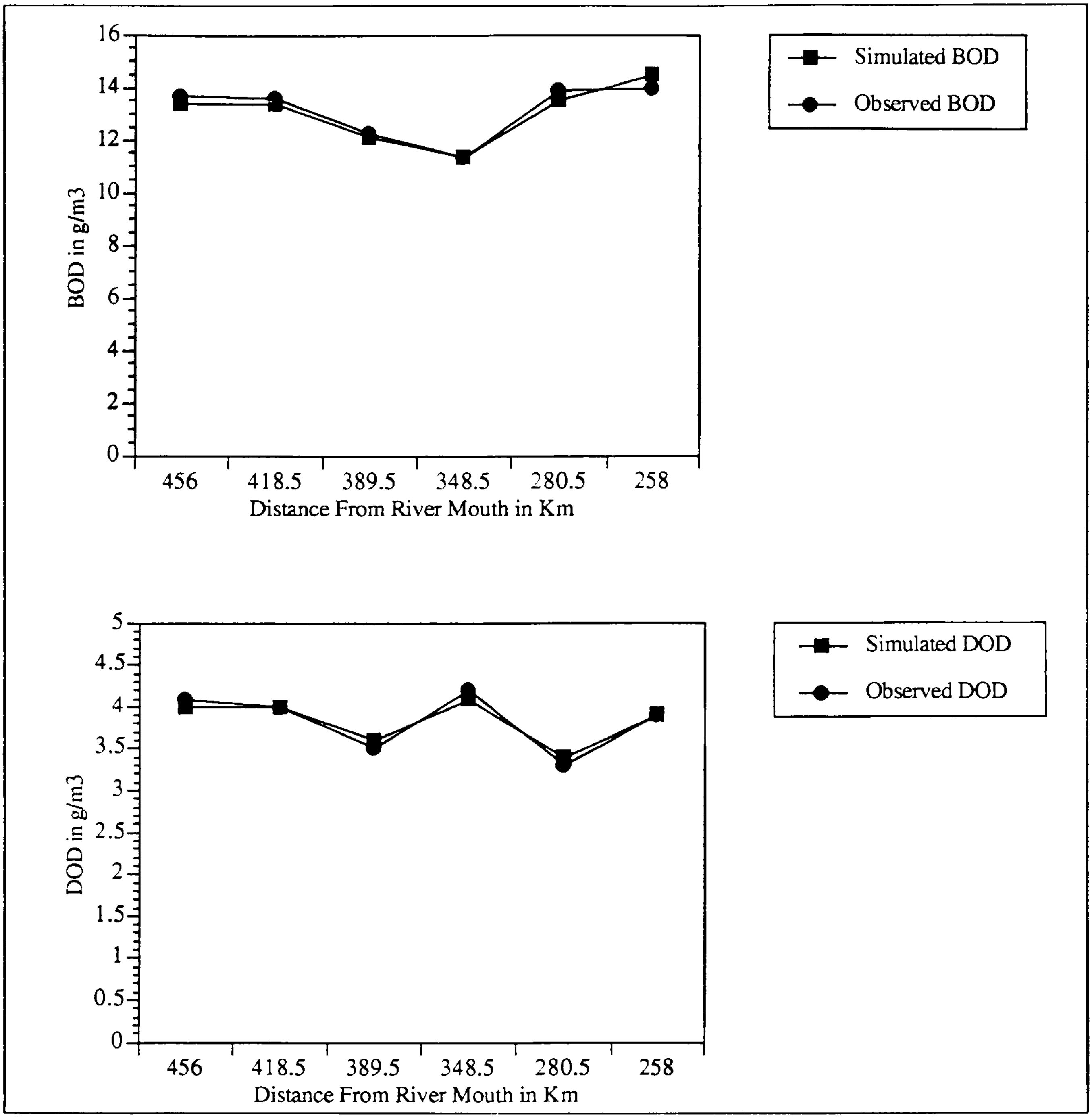


Figure 11. Comparison of simulated and observed results.

rules for model selection, and procedures for model coupling constitute the core of a rapid modeling environment in which a case-specific model can be promptly built.

The potential application areas of our framework include: a) water quality planning at the regional scale where a large number of streams modeled would make the use of input demanding models very expensive; and, b) long-term regional and local water quality policy making where the charting of trends in water quality does not require highly accurate predictions.

Acknowledgment

To the anonymous reviewer who took the time and trouble to do a thorough statistical analysis of the results presented in Table 1, our special thanks.

References

[1] Agogino, A.M. et. al., 1989. AI/OR Computational Model for Integrating Qualitative and Quantitative Design Methods, *Proc. of NSF Engineering Design Research Conference*, Amherst, pp. 97-112.

[2] Antunes, M.P., Seixas, M.J., Camara, A.S., Pinheiro, M., 1987. Logic dynamic simulation for water quality management. In: *System analysis in water quality management*, ed. M.B. Beck, pp. 401-410. New York: Pergamon Press.

[3] Barnwell, T.O., Brown, L.C., Whittemore, R.C., 1987. QUAL IIE - a case study in water quality management. In: *System analysis in water quality management*, ed. M.B. Beck, pp. 377-387. New York: Pergamon Press.

[4] Brinkman, J.J., Griffioen, P.S., Groot, S., Los, F.J., 1987. A water quantity and quality model for the evaluation of water management strategies in the Netherlands - application to the province of Friesland. In: *System analysis in water quality management*, ed. M.B. Beck, New York: Pergamon Press, pp. 23-30.

[5] Camara, A.S., Pinheiro, M., Antunes, M.P., Seixas, M.J., 1987a. A new method for qualitative simulation of water resources systems.1. Theory. *Water Resources Research*, Vol. 23, pp. 2015-2022.

[6] Camara, A.S., Ponherio, M., Antunes, M.P., Seixas, M.J., 1987b. A new method for qualitative simulation of water resources systems. 2 Applications. *Water Resources Research*, Vol. 23, pp.2019-2022.

[7] Chen, W.J. 1986. An expert system for the estimation of parameter values of a water quality model. In: *Proc. of the 1986 IEEE International Conference on Systems, Man, and Cybernetics*, Atlanta, pp. 236-240.

[8] Chorley, R.J., Haggett, P., 1967. Physical and information models in geography. London: Methuen.

[9] Da Costa, J.R., Loucks, D.P., 1987. Water quality management in the Ave River: from research to practice. In: *System analysis in water quality management*, ed. M.B. Beck, pp. 389-400, New York: Pergamon Press.

[10] Dixon, J.R., et. al., 1989. Computer-Based Models of Design Processes: The Evaluation of Designs for Redesign, *Proc. of NSF Engineering Design Research Conference*, Amherst, pp. 491-506.

[11] Dorfman, R., Jacoby, H.D., Thomas, H.A., 1972. *Models for managing regional water quality.* Cambridge: Harvard University Press.

[12] Elzas, M.S., Oren, T.I., Zeigler, B.T., eds., 1986. *Modeling and simulation methodology in the artificial intelligence era.* Amsterdam: North- Holland.

[13] Gero, John S. et. al., 1989. An Approach to Knowledge Based Creative Design, *Proc. of NSF Engineering Design Research Conference*, Amherst, pp. 333-346.

[14] Grimsrud, G.P., Finnemore, E.J., Owen, H.J.,1976. Evaluation of water quality models: a management guide for planners. *Socio-Economic Environmental Studies Series.* Washington, D.C., Office of Air, Land, and Water Use, U.S. Environmental Protection Agency.

[15] Gruber, T. and Cohen, P., 1987. Principles of Design for Knowledge Acquisition. *Proc. of IEEE 1987 Third Conference on Artificial Intelligence Application*, February, pp. 9-15

[16] Ho, Y., 1989. Editors Introduction, Special Issue on Dynamics of Discrete Event Systems, *Proceedings of the IEEE*, Vol. 77, No.1.

[17] Huang, Y. M., 1987. Building An Expert System Shell for Design Model Synthesis in Logic Programming, Master Thesis, University of Arizona, Tucson, Arizona.

[18] Jankowski, P., 1989. Knowledge-based structured modeling; an application to stream water quality management. Ph.D Dissertation, University of Washington, Seattle.

[19] Johnson, L.E., 1988. Water resource management decision support systems. *Journal of Water Resources Planning and Management*, Vol. 112, No. 3, pp. 308-325.

[20] Kim, Tag Gon and B.P. Zeigler, 1990. The DEVS-Scheme Simulation and Modelling Environment, in Knowledge Based Simulation: Methodology and Application (eds: Paul A. Fishwick and Richard B. Modjeski), Springer Verlag Inc. (to appear).

[21] Loucks, D.P., Stedinger, J.R., Haith, A.D., 1981. *Water Resources Systems Planning and Analysis.* Englewood Cliffs: Prentice-Hall.

[22] Loucks, D.P., Kindler, J., Fedra, K., 1985. Interactive water resources modeling and modell use: an overview. *Water Resources Research*, Vol. 21, No. 2, pp. 95-102.

[23] Lam, D.C., Fraser, A.S., Bobba, A.G., 1987. Simulation and analysis of watershed acidification. In: *System analysis in water quality management*, ed.M.B. Beck, pp. 85-96. New York: Pergamon Press.

[24] Mesarovic, M. D. and Y. Takahara, 1975. *General Systems Theory: Mathematical Foundations*, Academic Press.

[25] Oren, T. I., 1984. GEST—A Modelling and Simulation Language based on System Theoretic Concepts, in: *Simulation and Model—Based Methodologies: An Integrative View*, (eds: Oren, T.I., Zeigler, B.P and M. S. Elzas), North Holland, pp. 3-40.

[26] Orlob, G.T., ed. 1983. *Mathematical modeling of water quality: streams, lakes, and reservoirs.* Chichester: John Wiley & Sons.

[27] Rinaldi, S., Soncini-Sessa, R., Stehfest, H., Tamura, H., 1979. *Modeling and control of river quality.* London: McGraw-Hill.

[28] Rozenblit, J.W., 1985. A Conceptual Basis for Integrated, Model-Based System Design, Ph.D. Dissertation, Department of Computer Science, Wayne State University, Detroit, Michigan.

[29] Rozenblit, J. W. and Zeigler, B. P. Design and Modelling Concepts, 1988. In: *International Encyclopedia of Robotics, Applications and Automation*, (ed. Dorf, R.) John Wiley and Sons, New York, pp. 308-322.

[30] Rozenblit, J.W. and Y. Huang, 1987. Constraint-Driven Generation of Model Structures, In: *Proc. of 1987 Winter Simulation Conf.*, Atlanta, GA, pp. 604-611.

[31] Rozenblit, J.W. et.al., 1990. Knowledge-Based Design and Simulation Environment (KBDSE): Foundational Concepts and Implementation, *Journal of Operational Research Society*, Vol.41, No. 6, pp. 475-489.

[32] Rozenblit, J.W., Sevinc, S., Zeigler, B.P., 1986. Knowledge-based design of LAN's using system entity structure concepts. In: *Proc. of the 1986 Winter Simulation Conference.* Washington, D.C., pp. 604-611.

[33] Sage, A. P., 1977. *Methodology for Large Scale Systems*, Mc-Graw Hill, New York.

[34] Whitehead, P.G., 1983. Modeling and forecasting water quality in nontidal rivers: The Bedford Ouse study. In: *Uncertainty and forecasting of water quality*, eds. M.B. Beck, G. Van Straten, Berlin: Springer-Verlag, pp. 321-337.

[35] Wilson, A.G., 1981. *Geography and environment: system analytical methods.* Chichester: John Wiley.

[36] Wilson, A.G., Bennett, R.J., 1985. *Mathematical methods in human geography and planning.* Chichester: John Wiley.

[37] Wymore, A. W., 1967. *A Mathematical Theory of Systems Engineering: The Elements*, John Wiley.

[38] Zadeh, L.A. and C.A. Desoer, 1963. *Linear Systems Theory, The State Space Approach*, McGraw Hill.

[39] Zeigler, B.P., 1984. *Multifacetted Modelling and Discrete Event Simulation*, Academic Press.

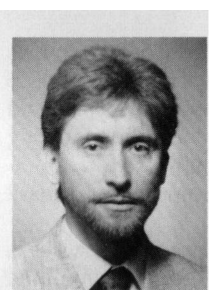
[40] Zeigler, B.P., 1984. System—Theoretic Representation of Simulation Models, *IIE Trans.*, 16, pp. 10-27.

[41] Zeigler, B. P., 1987. Hierarchical, Modular Discrete-Event Models in an Object-Oriented Environment, *SIMULATION*, Vol. 50, No. 5, pp. 219-230.

[42] Zeigler, B.P. 1990, *Object—Oriented Simulation with Hierarchical Modular Models: Intelligent Agents and Endomorphic Systems*, Academic Press.

[43] Yang, J. and J.W. Rozenblit, Case Studies of Design Methodologies: A Survey 1990, *Proc. of the International Conference on AI, Simulation and Planning in High Autonomy Systems*, IEEE Computer Press, pp. 136-141.

JERZY W. ROZENBLIT is an Assistant Professor of Electrical and Computer Engineering at The University of Arizona, Tucson. He received his Ph.D. and M.S. degrees in Computer Science from Wayne State University, Detroit, in 1985 and 1983, respectively, and his M.S. degree in Control Engineering from the Technical University of Wroclaw, Poland, in 1980. He specializes in modeling and computer simulation, knowledge-based system design, and artificial intelligence. His principal research activities focus on the development of expert, computer-aided environments for engineering design support. Dr. Rozenblit has consulted for Martin Marietta, Semiconductor Research Corporation (SRC), and Siemens, and has been a reviewer for the National Science Foundation, Research Council of Canada, and a number of archival journals. He also serves as an Associate Editor of ACM Transactions on Modeling and Computer Simulation. His research in system design has been supported by NASA, Siemens, McDonnell Douglas, SRC, and the National Science Foundation.



PIOTR L. JANOWSKI is an Assistant Professor in the Department of Geography, College of Mines and Earth Resources, University of Idaho. His current professional activities include teaching and externally founded research in the area of decision making support for natural resources management and planning. Dr. Jankowski has five years of computer consulting experience in the U.S. in the areas of mathematical programming, natural systems simulation modeling, statistical analysis, computer mapping and GIS applications. He initiated research on the application of structured modeling to natural systems, has worked on developing expert system applications to computer mapping, and engaged in research on evaluation techniques for environmental planning and natural resource development. A native of Poland, Dr. Jankowski was a recipient of a Fulbright Scholarship to study at the University of Washington where he received a Ph.D. degree in Geography (1989) with the award for Distinguished Achievement as a Ph.D. Student (1989). Prior to coming to the United States, he earned a M.S. degree in Operations Research and Econometrics with honors from the Poznan School of Economics in Poland.