

## 河川流域の流出シミュレーションモデルのシステム化に関する研究

京都大学工学部 正員 高棹琢馬 京都大学工学部 正員 椎葉充晴  
 京都大学工学部 正員 堀智晴 京都大学工学部 正員 立川康人  
 京都大学大学院 学生員 ○鈴木俊朗

1. はじめに 現在用いられている HEC-1, HSPF, Sacramento Model, SSARR, Stanford Watershed Model, SWMM, TR-20 などのコンピュータモデルの多くは約 30 年前に開発されたモデルである。これらのモデルが現在なお用いられているのは、モデルの開発機関が水文学の成果を次々にモデルに取り入れ、モデルの適用範囲を拡大する努力を重ねてきたからである。しかし修正を繰り返して適用範囲を拡大した反面、結果としてこれらのモデルのソースコードは膨大なものとなっている。モデルを修正する場合、ソースコードを詳細に理解する必要があるため、これらのモデルの修正は開発機関の手による場合を除いて、事実上不可能である。しかし、流域の局所的変化や水文学理論の進歩にモデルを柔軟に対応させていくには、ユーザーが必要に応じて容易に構築・修正できるのが望ましい。そこで本研究では、構築・修正の容易な新しいモデリングシステムを提案する。

2. 要素モデルによる流出モデルの構成 流出モデルは、その構造により、

Type 1 流域全体を単一のモデルで表現するモデル  
 Type 2 複数の部分モデルで全体を表現するモデル

に分類できる。Type 1 のモデルの修正作業はモデル全体を作りなおす作業となり、修正作業というより再構築作業となる。一方 Type 2 の修正作業は、修正すべき部分に対応する部分モデルのみの修正作業に限定できる。また、部分モデルのメニューが充実していれば修正作業は部分モデルの交換作業に簡素化でき、モデルの構築作業もメニューの中から部分モデルを選択して組み立てる作業となる。更に、Type 2 のモデルでは降雨分布や水文特性の分布も表現できるのでより現実に即したシミュレーションが出来る利点もある。以上の理由で、本研究では Type 2 のモデルの開発に取り組んだ。

Type 2 の既存モデルには、HEC-1, HSPF, TR-20 などがあり、これらのモデルは部分モデルとして部分流

域モデルという概念を取り入れている。しかし、この部分流域モデルも以下の理由で、Type 2 のモデルの利点を生かしていない。

1. 部分流域モデルの内部には更にいくつかの機能が融合されている。例えば、部分流域内部には、流域斜面・河道など複数の要素が含まれているので、ある流域要素のみが局所的に変化した場合でも部分流域モデル全体を修正しなければならない。
2. 部分流域モデル間の結合様式（入出力値の受渡し方法）が統一されていない。結合関係の仕様をそれぞれの部分流域モデルごとに把握していなければ、部分流域モデルを自在に組み合わせて全体モデルを組み立てるのは困難である。また、ユーザーが独自の部分流域モデルを構築する場合も他のモデルとの結合関係を把握するために多大な労力を必要とする。
3. 各部分流域モデルに共通する部分（動特性・結合関係）と共通でない部分（数理モデル）が融合している。例えば、部分流域モデルを構築する場合に共通部分をそのまま利用することができない。

このように既存のモデルは、ユーザーが独自に部分流域モデルを構築したり修正する作業が困難なものとなっている。

そこで上記 1 の問題点を解決するため、流域要素に対応する部分モデルとして要素モデルという概念を取り入れ、要素モデルの組み立てにより流域全体を構築する方法をとる。また、上記 2, 3 の問題点を解決した上で構築や修正の容易なモデルを開発するには、この要素モデルが更に次の条件を備える必要がある。

- 条件 1. 要素モデルの結合様式が統一されている。  
 条件 2. 要素モデルに関係する数値とそれを操作するのに必要な全ての関数は、その要素モデルの内部に備えている。

条件 3. 各要素モデルに共通している動特性・結合関係などを表す部分と要素モデルごとに異なる数理モデルを表す部分とが分離されている。

条件 1 により要素モデルの自由な構成が可能となり、条件 2 により要素モデルは単独で機能的に完結する。従って、要素モデルは全流域モデルを構成する一つの部品と見なせる。このことを要素モデルの部品化という。また条件 3 により、新しい要素モデルは数理モデルなどを表す非共通部分のみを構築して共通部分と組み合わせることで完成することができる。このようにモデルの機能を分離して、その内部構造を明確化することを要素モデルの構造化という。

3. オブジェクト指向言語による要素モデルの実現  
手続き型言語で要素モデルの部品化・構造化を実現するとソースコードが複雑になることは避けられない。しかし、C++、Smalltalk などのオブジェクト指向言語を用いるとプログラムコードを簡素化・明確化することが出来る。これは、オブジェクト指向言語がクラス、継承という概念を備えているからである。

C++ の用語を使って説明すると、クラスとは、あるデータ群とそれを操作する関数群をパッケージとしてまとめたものである。このデータ群をデータメンバ、関数群をメンバ関数と呼ぶ。ここで、状態量、パラメータなど各要素モデル固有の値をデータメンバとし、それを操作するのに必要な全ての関数をメンバ関数としてひとつのクラスにまとめれば、要素モデルをクラスで表現できる。またこのようにすれば要素モデルを全体のモデルから分離できるので、条件 2 を実現できる。

次に継承の説明のため A、B というふたつのクラスを考える。クラス B が、クラス A と同じデータメンバ、メンバ関数に加え B 独自のデータメンバやメンバ関数を持っているとする。このとき、クラス B をクラス A と独立して作るのではなく、クラス A から B を派生させて作ることができる。B が A から派生していることを宣言すれば、B のソースコードで A のデータメンバやメンバ関数に関して再び記述する必要はない。B のソースコードでは、A からの派生宣言と B 独自のデータメンバやメンバ関数について記述する。このとき A をベースクラス、B を派生クラスという。このように派生宣言を行うだけで B が A の機

能を引き継ぐことを継承という。

本研究では、流域斜面・河道・貯水池などの要素モデルのもつ機能のうち、共通機能(動特性や結合関係に関する機能)のみを持つ抽象的なクラス Inout を定義した。クラス Inout の派生クラスとして要素モデルを定義すれば、要素モデルは共通機能を Inout から継承できる。従って要素モデルを構築するときは、Inout からの派生宣言とその要素モデル独自の機能を記述すればよい。また、共通部分に結合関係の機能を定義しておけば各要素モデルの結合関係の機能を統一できる。これにより、条件 1、条件 3 を実現できる。

4. 流域モデルの試作 クラス Inout を継承して要素モデルを定義すれば本研究の目指す流出モデルが構築できるようになった。そこで、流域を構成する要素モデルとして、

- 流域に降雨強度を与えるクラス : Sky
- 線型貯水値モデルのクラス : L\_reservoir
- 流域からの流出量を記録するクラス : Observer

を定義し、これらを直列に結合した簡単な流出モデルを試作し、これを用いて、

Point 1. クラス Inout に定義した共通機能が正常に動作するか。

Point 2. モデル化の作業が円滑に進められるか。

Point 3. 更に改善すべき点はないか。

を検討した。

その結果、Inout で定義した共通機能は問題なく動作することが分かった。また、次のような改善が必要であることが分かった。

1. 要素モデルのメニューを充実させ、流出モデルの構築を容易にする。
2. 要素モデルや流域モデルの構築を対話形式により行えるようにする。
3. クラス Observer, Sky は他の要素モデルと異なり、一つのモデルで流域内の全ての記録機能、降雨シミュレート機能を実現できる。これらのモデルは機能自体は単純だが、多くのモデルと結合するので例えば結合関係の初期化作業などは単純作業の繰返しとなる。そこでこうした作業を簡素化する方法を開発する。