

II-212

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

京都大学大学院 学生員 鈴木俊朗 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

1 はじめに 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。モデルの構築作業・修正作業・シミュレーションの実行において、type 2 は type 1 に比べ以下の点で優れている。

1. 部分モデルのメニューが充実していれば、モデルの構築作業は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。
2. モデルの修正作業は部分モデルの交換作業になる。
3. 降雨分布・水文特性の分布を表現できる。

現在用いられているコンピュータモデルのうち type 2 に属するものには、HEC-1, HSPF, TR-20, SSARR, SWMM などがある。これらの多くは約 30 年前に開発されたモデルであるが、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てモデルの適用範囲を拡大してきたので、現在に到りなお用いられている。こうして改良を重ね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。ところがモデルを修正する場合、この膨大なソースコードを詳細に理解する必要があるため、開発機関の手によらなければ事実上不可能である。しかし、流域の局所的变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修正できるのが望ましい。そこで本研究では、構築・修正・追加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 Type 2 の既存モデルは、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。この部分流域モデルの内部には、更にいくつかの流域要素(流域斜面・河道・貯水池など)が含まれている。従って、部分流域内の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。これでは type 2 のモデルの利点(左記 2.)が満足されて

いない。こ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、要素モデルを組み立てて全体モデルを構築する方法をとる。更に、構築・修正・追加の容易なモデルの開発には、

- 要素モデル自体の構築・修正が容易であること
- 要素モデルの構成による全体モデルの構築が容易であること

の2点を満たさなければならない。そのために、要素モデルに次の条件を与えた。

条件 1 各要素モデルについて共通機能(動的機能・入出力機能)を表す部分と、独自機能(数理モデル)を表す部分とを分離する。

条件 2 各要素モデル間の入出力方式を統一する。

条件 3 その要素モデルに関係する数値(状態量・パラメータ)とそれを操作するのに必要な全ての関数を、その要素モデルの内部に備える。

条件 1 により新しい要素モデルは、独自機能を表す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。また、条件 2 により要素モデルの自由な結合が可能となり、更に条件 3 により要素モデルが単独で機能的に完結するので、要素モデルを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。

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構造化・部品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ソースコードを簡素化・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

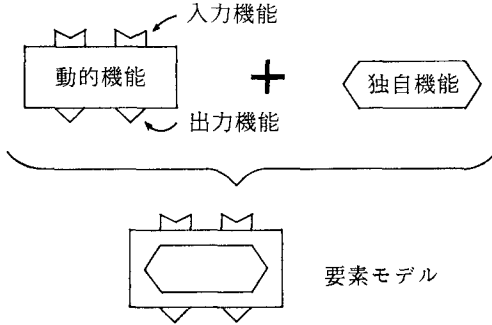


図1 要素モデルの構築

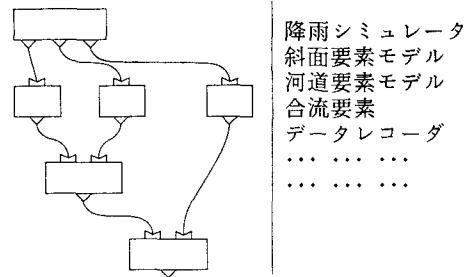


図2 全体モデルの構成

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3を実現できる。

次に継承の説明のため A, B という2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・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2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. 要素モデルの構築・全体モデルの構成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・降雨シミュレータ機能を実現できる。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、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。