

“Computer-based modeling environments”

Arthur M. GEOFFRION

European Journal of Operational Research 41 (1989) 33-43

この論文では、MS/ORの分野における、コンピュータを用いたモデリング環境の性質およびこれからの研究課題について考察している。論文の冒頭で著者は、コンピュータを用いたモデリング環境の構築が、MS/ORの分野において今後10年間で最も重要な課題の1つであるとしている。その理由として、モデリング環境はモデルにもとづくタスクにおいて、その生産性を高める手段であり、モデルの質を保証するツールであることをあげている。また、よいモデリング環境の整備により、MS/ORが広く利用される可能性が増大することもあげている。この論文では、モデリング環境において必要とされる5つの性質と3つの重要な研究課題について議論している。以下、論文にしたがって要約する。

モデリング環境において必要とされる5つの性質

1. ライフサイクル支援

モデルを用いるシステムには、概念化、モデルの開発、モデルの使用および廃棄といったライフサイクルがある。このようなライフサイクルは、いくつかのフェーズに分けられているので、各フェーズにまたがった処理を統合していることが望ましい。よって、モデリング環境において、高度なソフトウェア統合が必要となる。さらに、モデリング環境は、データ、モデル、問題を解くプログラムなどの資源を結合できるようなライブラリを提供すべきである。

2. 意思決定者に親切であること

モデリングの専門家以外のユーザーに親切(hospitable)な環境である必要がある。たとえば、モデルの表現が明確で、モデル構造が直観的に理解でき、ユーザーが学習し、使用することが容易で、初心者向きの処理方法を持つなどの点があげられる。

この問題を重視する理由はいくつかある。パーソナル・コンピュータの発展により、モデリングの専門家以外の人が自らモデリングを行なう可能性が増えつつある。モデリングの専門家を通さない、ユーザーの直接操作が望ましい場合が多い。

このような環境を実現する方法の1つとして、実行可能(executable)なモデリング言語があげられる。

3. 発展の柔軟性

モデルを簡単に修正することが非常に重要である。一般に、はじめに作成したモデルは、100%正しいモデルであるとは限らない。また、問題自体も変化しているので、その問題に対応するモデルに対する修正が絶えず必要である。柔軟性を高めるためには、ある種の実行可能なモデリング言語が必要となる。また、モデルに対する修正は、宣言的仕様で行なえる方が望ましい。

4. 自然なモデル定義言語

モデリング環境において、特別な問題領域およびモデル化のパラダイム、あるいは問題解決のテクニックなどにかかわらず、広い領域にわたるようなモデル定義に適用できる汎用言語をそなえることが必要である。

現在、多くの特殊なモデル表現のパラダイムがある(たとえば、決定木、マルコフ連鎖、数理計画問題、待ち行列問題など)。それらの表現の中で、明確性や効率を考慮して提案されたものはほんの一部であって、ほとんどのものは単に標準の表現がないので勝手に作成されたものである。また、表現の多様化はモデリングの専門家とユーザーとの交流も妨げている。ゆえに、モデリング環境にとって、共通の概念上の枠組みが必要である。

5. 重要資源の管理

モデリング環境は、モデル、データ、問題を解くプログラムなどの資源に対する蓄積、共有、再使用などの機

能を提供すべきである。そのためには、広範なデータ管理システム、モデル管理システムが必要である。また、各資源を必要に応じて簡単に結合でき、表現上の混乱を避けるために、それらに対する統一した表現形式が必要とされる。

上述のような性質を持つモデリング環境を実現するために、次の3つの研究課題があげられる。

課題1. 概念モデリングの枠組み

概念モデリングの枠組みはモデリング環境を設計するための賢明な出発点である。

この枠組みは次の性質を持つべきである。

- (a) 汎用性.
- (b) 厳密に定式化できること.
- (c) モデリングの各フェーズに関わる人々にとって、わかりやすく自然であること.
- (d) モデリングやモデル操作におけるどのパラダイムにも整合するようなパラダイム中立性.
- (e) “良い”モデリングスタイルとの整合性.
- (f) 実行可能なモデリング言語の設計の基礎となること.

概念モデルの枠組みは、MS/ORでは重要性があまり認識されていないが、近隣の分野では注目を集めている。データベース理論ではデータモデルとして、人工知能では知識表現として、高級プログラミング言語設計論では抽象データタイプとして研究されている。

MS/ORにおける概念モデリング枠組みの設計には、実体、属性、関係および集合にもとづいたアプローチ、モジュールのネットワークにもとづいたアプローチ、属

性グラフにもとづいたアプローチ、定義システムにもとづいたアプローチがある。

課題2. 実行可能なモデリング言語

どのアプローチの枠組みを使っても、実行可能なモデリング言語が概念上のモデリングに必要である。実行可能なモデリング言語も、上述の(a)から(e)の性質を持たなければならない。ただし、ここではこれらが枠組みの性質ではなく、言語の性質なので違った解釈をしなければならない。たとえば、(c)のわかりやすさに関して、操作的ではなく宣言的で、暗号のような文ではなく覚えやすい文である。そして、実行可能なモデリング言語は広範な整合性のチェックが行なえなければならない。

‘実行可能’という言葉はモデリング環境内のプログラムがこの言語で書かれたモデルを受け取ることによって、実行が可能であることを意味している。

課題3. ソフトウェアの統合

上述のように、モデリング環境は、データ、モデル、問題を解くプログラム等が結合できるライブラリ、‘実行可能性’に必要な機能を提供するプログラム、モデリングの各フェーズを支援するツールおよびユーティリティなどのソフトウェアを含んでいて、これらのソフトウェアを概念的に結合していなければならない。

このように統合できるモデリング環境を実現するための1つのアプローチは、概念モデリング枠組みを利用して、その環境の構成部分をも‘モデリング’することである。

(東京工業大学 李東, W. パークブーム)

× × × × ×