

GERMによる問題定義からシミュレーション・プログラムを起動するための仕組みについて

02601261 北海道大学経済学部 向原 強

1. はじめに

これまで筆者たちのグループでは、実体-関連モデルを拡張した問題定義の手法(以下, GERM アプローチ)を提案し, その有効性を検討してきた[1,3]. 問題定義とは, 意思決定問題の明確化と構造化をいう。GERMアプローチは問題定義にとどまらず, 数理モデル(Mathematical Model, 以下, MP)との橋渡しや, 解法プログラム(ソルバー)の起動を目指している。これまで, 筆者たちは最適化問題(特に, 整数計画問題)を対象とした事例データベースシステムを開発した[3]。本論では, GERMによる問題定義に関する用語を定義した後, 離散型シミュレーション・モデル(Discrete Event Simulation Model, 以下 DESM)に対しても, 一般性を維持できる MP の一般的な形式について検討し, シミュレーション・プログラムの起動を支援するために必要なモデル変換の仕組みを提案する。

2. GERMによる問題定義

最初に GERM による問題定義に使われる用語を定義する。実体型は型属性とその値の対の集合である。関連型は一組の実体型あるいは他の関連型からなる定義型と, 型属性とその値の対の集合である。システム型は一組の実体型, 関連型, あるいは, 他のシステム型からなる構成型と, 型属性とその値の対の集合である。オブジェクト型とは実体型, 関連型あるいはシステム型のことであり, インスタンスと呼ばれる同類のオブジェクトの集合である。GERMアプローチでは問題型はシステム型として記述され, これを問題定義と呼ぶ。実体型のインスタンスは一組のインスタンス属性とその値の対である。関連型のインスタンスは定義型のインスタンスの組と, インスタンス属性とその値の対の集合である。システム型のインスタンスは構成型のインスタンスの組と, インスタンス属性とその値の対の集合である。インスタンス属性はそれが属するオブジェクト型の型属性のどれかに対応し, その型属性はインスタンス属性が取りうる値の範囲を指定する。インスタンスは実体型あるいはシステム型のばあいには一意識別子(ID)となる1つのインスタンス属性(以下, ID)を持つ。関連型の場合にはその定義型のインスタンスの一意識別子の組が一意識別子となる。関連型は別に, 一意識別子となるインスタンス属性を持ってよい。例えば, 機械型と仕事型を定義型にもつ作業型のインスタンスの一意識別子は, (機械ID, 仕事ID)と表してもよいし, 作業IDというインスタンス属性としても良い。同一のオブジェクト型に属するインスタンスは, そのインスタンス属性の組を見出しとする表形式で記述できる。インスタンス属性に対応する型属性はインスタンス型属性, 対応するインスタンス属性を持たない型属性は非インスタンス型属性と呼ぶ。

問題型を定義するとき, その構成型の一部のオブジェクト型に対してインスタンスの集合を特定する。そのようなオブジェクト型は固定型と呼ばれる。固定型とはならないオブジ

ェクト型は非固定型と呼ぶ。オブジェクト型は実世界における同類の認識対象を記述するのであり, その型属性は対象の定量的特性(あるいは/および定性的特性)を記述する。型属性およびインスタンス属性の値は人手(以下, ユーザ)で入力するか, システムが自動的に算出するかどちらかである。問題型は数理モデルに定式化されるとする。固定型の型属性の定量的特性の内, 問題型の定義の時に値を指定され, 数理モデルに定数として組み込まれるものがある。固定型の型属性の定量的特性の中には, ユーザがその値を, 具体例を入力する時に指定しなければならないものもある。これを特に, 固定型パラメータ属性と呼ぶ。

3. MPの一般的形式

MPは, 記号の定義と, 記号を基本要素とし, その数学的関係を示す一連の数式によって構成される。記号は, 意味的属性(クイディティ[本来の意味], デイメンション, 単位など)に加え, 以下でとりあげるステータス, 形態, ドメイン, 値といった論理的属性を明示する必要がある。

3.1. 記号ステータス

MPの記号には, 変数, 定数, パラメータ, ダミーインデックスがある。これらは記号のステータスと呼ぶ。変数は, 全てを評価する必要はない。評価の対象となる変数は特に評価変数と呼び, 評価変数ではない変数は派生変数と呼ぶ。

3.2. 記号形態, ドメイン, 値

記号は, 単一, 配列, 複合, 集合のように分類できる。これを記号形態と呼ぶ。さらに, 記号は, そのドメインを定義する必要がある。ドメインには, 最大(極大)値, 最小(極小)値に加え, データ型(整数, 実数, 文字列, ブーリアン)も含まれる。従って, 変数が離散であるか連続であるかは, この記号ドメインで指定される。また, 定数は, モデル化段階で特定値を設定することが必要である。このような値を設定するための記号特性を記号値と呼ぶ。記号値は変数の場合は初期値, パラメータの場合はデフォルト値として扱うことができる。

3.3. 数式

MP(最適化モデル, 目標計画モデル, 方程式モデル)に含まれる各数式は, 目標, ゴール, 評価, 制約に区分される。これを数式のステータスと呼ぶ。各数式はテスト条件と呼ばれる成立要件を付記する。最適化モデル, 目標計画モデル, 方程式モデルは, パラメータと評価変数の関係が直接関連しているの, 数式を宣言的に記述することは可能である。各モデル間の相違点は構成する数式のステータスだけである。しかし, DESMは数理モデルとして扱われることが一般的であるのに関わらず, パラメータと評価変数の関係を数式によって直接記述することは難しい。そこで DESMでは, モデル化の対象となるエンティティ・フローと, それによるシステム(エンティティ, リソース)の状態変化を表現しなければならない。状態変化を起こすイベントには, エンティティの生

成、離脱、待ち行列への入出、リソースの占有・解放、プロセスの開始、終了などがあげられる[2]。この状態変化は、通常、定量的なものであるため、数式による表現が可能である。[4]では、これをイベント計算(event calculus)と呼び、代数的表現方法を提案している。こうして記述された数式のステータスは本研究の文脈では評価か制約である。しかしこのアプローチは、(1)非専門家が理解することは容易でない、(2)専門家にとっても、シミュレーション言語によるモデル化と比べると複雑で直感的理解が難しい、(3)モデル管理(モデルの蓄積、取り出し、再利用)が難しい、などの欠点がある。このうち、(1)、(3)は一般的なモデリング言語としての短所であり、(2)は代数的表記に変換することによる短所である。

そこで本研究では、[4]と同様な方法で DESM を代数的表現として扱った上で、問題定義から出発する GERM アプローチに活用する方法を検討する。GERM アプローチは上記の短所に対する一つの解決法である。

4. モデルの変換メカニズム

GERM アプローチで扱う MP には、問題定義における型属性と MP の記号との対応が、その一部において一対一として示すことのできるユーザ定義モデル(User-Defined Mode, 以下、UDM)と、ソルバーを開発するために構築される標準型モデル(Standard Model, 以下、SM)がある。

4.1. 問題定義から UDM へ

問題定義と UDM の対応は結合条件が担う。結合条件を利用すると、非固定型の型属性と固定パラメータ属性のインスタンスはパラメータ記号の具体値として UDM へ渡すことが可能になり、UDM の評価変数の値はシステム型属性として、問題定義へ戻すことが可能である。結合条件は次を満足するように設定する必要がある。

- ✓ 問題定義の型属性は、UDM の記号とそのステータスが一致しなければならない。
- ✓ UDM 記号のインデックス次元数と型属性が属するオブジェクト型の定義型要素数は一致しなければならない。また、各インデックスは定義型要素のいずれかと対応づけられる。
- ✓ 問題定義の型属性のうち、そのステータスがパラメータとなるものは、UDM パラメータのいずれかと対応づけられていなければならない。それは非固定型の型属性か固定パラメータ属性の何れかである。
- ✓ UDM の評価変数は問題定義のシステム型属性のいずれかと対応づけられていなければならない。

これらを満足すれば、結合条件によって、UDM パラメータの具体値を獲得することができる。

ところで、UDM として、[4]のように時系列をインデックスとした記号を利用するとき、それらは問題定義の型属性と対応づけることができないので、そのような型属性はパラメータや評価変数にはできないことに留意する。

4.2. UDM から SM へ

UDM は SM に変換した後、ソルバー起動を実現する。結合情報がそれを担う。ソルバー起動に必要な情報は、ソルバーをチューンナップするための情報(ソルバー・オプション)

とターゲット・モデル(SM)のパラメータに分類される。このとき、SM のパラメータは、UDM のパラメータか定数から得られるが、その関係は次のどれかを満足しなければならない。[3]ではモデル間の関係として記述したが、柔軟性や正確性を期するため、ここでは記号間の関係として整理する。

- ✓ ある SM パラメータは、一つの UDM のパラメータと、構造(形態、インデックス構造、およびドメイン)が同じである。このとき、互いの記号は等価であるという。DESM ではシミュレーション言語の仕様に従って UDM をモデル化するので、多くの記号関係は等価である。
- ✓ ある SM パラメータは、UDM にとって定数と扱うことができる。このような関係は特殊型という。
- ✓ ある SM パラメータは、一つもしくは複数の UDM パラメータとドメインが同じであり、それらを適当に結合しインデックスを揃えると、その記号が SM パラメータと等価になる。これは LP の標準型を構築する手続きと類似であり、このような関係を標準型という。
- ✓ 一つもしくは複数の UDM パラメータを既知の関数で合成すると、その記号がある SM パラメータと等価になる。このとき SM パラメータは UDM の派生型という。

全ての SM パラメータが、上の条件を満たすとき、UDM から SM のパラメータを全て獲得することが可能である。したがって、ソルバーの起動に必要なデータのうち、ソルバー・オプションを除く必要データを揃えることができる。よって、これらをソルバー形式に変換することによってソルバー起動に必要な入力データを形成できる。

4.3. 数理モデルとしての DESM

紙面の都合上、DESM とそのモデル変換のサンプル事例を記述することができなかったが、数理モデルを『パラメータと評価変数の数学的關係』と一般化して定義しているため、この変換メカニズムの枠組みの中では DESM を、最適化モデルや目標計画モデルなどと同様に扱っている。各パラダイムで異なるのは使用される数式のステータスにすぎない。それが提案システムの特徴であり、重要な点である。

参考文献

- [1] Bao, J. and Y. Sekiguchi (1999), "Developing User Interface for Input of Pre-Mathematical Problems Specification in Scheduling," *Discussion Paper Series A*, No.61, Faculty of Economics, Hokkaido University.
- [2] Kelton W.D. 他, 高桑宗右エ門訳(1999), 『シミュレーション-Arena を活用した総合的アプローチ』, コロナ社
- [3] Mukohara, T. and Y. Sekiguchi (1999), "Method for Invoking Solvers from Non-Mathematical Problems Specifications," *Discussion Paper Series A*, No.63, Faculty of Economics, Hokkaido University.
- [4]. Pollatschek, M.A., "Algebraic Description of Discrete Event Simulation Models and its Implications for a Graphical Interface", <http://catt.okstate.edu/itorms/bani/alshome.htm>