

実体 - 関連概念の拡張による スケジューリング問題記述の特徴と応用

向原 強
北海道情報大学

関口 恭毅
北海道大学

鮑 金源
札幌オフィスコンピューター株式会社

(受理 2004年 3月 10日; 再受理 2005年 6月 1日)

和文概要 意思決定支援のための数理モデルの記述方法に関する研究は行われているが、数理モデルによる記述の対象である意思決定問題そのものの操作的な記述方法には適当なものが見あたらない。著者らが意思決定問題の構造化された記述である問題定義の方法として提案している GERM は、実体 - 関連概念をいくつかの点で拡張して利用する。本論文はその拡張の概要を説明し、GERM による問題定義の記述様式を提案し、 $F2//C_{max}$ の問題定義を示すことで具体的に説明する。さらに、実用的な APS ソフトウェアが解く実際のスケジューリング問題を例として、具体的に問題定義を記述することを通して GERM の特徴と有効性を示す。

キーワード: スケジューリング, 問題定義, モデル記述, 実体, 関連, オブジェクト指向

1. はじめに

数理モデルを現実世界の問題解決に適用するためには、それがどのような現実の問題を解決するために開発されたのかを知ることが有益であろう。また、現実世界の新しい課題を解決するための意思決定に直面する場合、類似の課題がどのように整理されて数理モデル化されたかを知ることが、直面する課題を分析する上で良い参考になると期待される。次節で概観するように数理モデル自身、あるいは、その説明を記述する方法に関する先行研究はあるものの、数理モデルの原因である現実世界の状況（以後、意思決定問題）を記述する方法についての研究は少ない。本研究では、意思決定の次のような状況を仮定する。

- 前提 1：意思決定者はその課題の解決のために意思決定支援者の支援を得る。意思決定支援者は意思決定者が解決すべき課題に関する十分な知識を持たない。
- 前提 2：意思決定支援者は数理モデルを利用して支援する。意思決定者は数理モデルやその構築について十分な知識を持たない。
- 前提 3：意思決定者が解決しようとする意思決定問題は類似のものが繰り返し発生するので、意思決定のプロセスを定型化することが望まれる。
- 前提 4：意思決定者と意思決定支援者は、意思決定問題の数理モデルを構築・利用するのに必要な情報交換をする。

これらの仮定の下では、意思決定問題を確定し記述することは意思決定者と意思決定支援者の共同作業である。この共同作業の結果として得る意思決定問題の記述を以下では「問題定義」と呼ぶ。意思決定問題は現実世界にある。数理モデルは数理世界の存在である。数理モデルの数値・記号・式を現実世界の諸特性に対応づけることによって、数理モデルは意

思決定問題の表現として解釈することが可能になる。MS/ORの文献では多くの場合、数理モデルの意味的説明として、意思決定問題が叙述的に説明される。それは多くの場合、問題定義としては不完全であり、しかも、叙述文であるから、再利用には不便である。なぜならば、数理モデルを説明する問題は、数理モデルの原因である意思決定問題とは異なるし、叙述的記述は記述内容や記述様式を一貫させることが困難だからである。

このような観点から、数理モデルそのものではなく、「問題定義」のための記述様式 GERM (Generic Entity-Relationship Model) を具体的なスケジューリング問題を利用して明らかにし、その特徴と有効性を示すことが本論の目的である。

GERM は、論議領域（意思決定者が解決したい課題の存在する現実世界の領域、world of discourse）が相互に関連しあう個々のオブジェクトの集合から成るとの、基本的認識に基づいて実体-関連概念を利用する。その意味ではオブジェクト指向の一種であるが、情報システムの設計開発におけるオブジェクト指向アプローチ（例えば、Frankel *et al.*[9]）とは異なる。情報システムの設計開発のためのオブジェクト指向アプローチでは、記述の対象はデータの入力、処理、出力という情報システムの機能である。これに対して、GERM は論議領域にどのようなオブジェクトが存在し、それらがどのような相互関係を持っているかを記述の対象とする。

GERM による問題定義は、論議領域を構成する個々のオブジェクトやオブジェクトの集合に関する記述から成る。それは数理モデルに含まれる変数や定数となるオブジェクトの数量特性を記述するばかりでなく、数量間の関係を表す定性特性も記述する。本研究はスケジューリング問題の例として、APS (advanced planning and scheduling) ソフトウェアが解く実際的なスケジューリング問題も取り上げる。

本論文の構成は以下の通りである。次節では実体-関連概念を応用する数理モデル記述法を中心に先行研究を概観する。第3節では GERM による問題定義の従来の実体-関連モデルとの違いを概説し、第4節で GERM の記述様式を提案・説明する。第5節で、ある APS ソフトウェアのスケジューリング問題を GERM で記述することを通して、GERM の特徴と有効性を示す。第6節はまとめである。

2. 実体-関連概念を利用するモデル記述

数理モデル記述の方法は意思決定支援のためのモデル管理との関連で研究されている。そこで提案されている様々なモデル記述方式は、代数的モデル化言語 (algebraic modeling languages)、データベース指向 (database-oriented)、グラフ基盤 (graph-based)、知識基盤 (knowledge-based) の4つに分類される (Bharadwaj *et al.*[4])。

実体-関連概念を利用するモデル記述はデータベース指向に分類される。Müller-Merbach ([24], [25], [26]) はモデル作成に關係データモデルを応用し、システムズ・アプローチを援用する立場から5段階のモデル化手順を提案する。Blanning [6] は数式モデル(の処理)における変数(データ)の入出力関係に基づく記述に実体-関連データモデルを利用する。構造化モデリング(以後 SM; Geoffrion[11])は、その基本実体(primitive entity)が実体に、複合実体(compound entity)が関連に相当するので、実体-関連概念を基盤とするモデル化の枠組みである。Geoffrion[13],[14]は SM のモデル記述のために SML (Structured Modeling Language)を開発する。SM はモデル記述に現れる要素(変数や定数)の間の定義上の依存関係(definitional dependency)の概念に基づくものであって、プロトタイプ・システムにインプレメントされている(Geoffrion[12])。Ramirez *et al.*[27]はモデル/データ独立、モデ

ル/ソルバー独立の観点からモデルの多様な活用を実現するために SM に基づいたモデル化言語 SM/DB を開発する。事例ベース推論をモデル化支援に利用する試みの中で Liang[21] は問題 (の数理的構造) を実体グラフ (entity graph) と属性階層 (attribute hierarchy) の組合せによって記述することを提案する。

実体 - 関連概念を応用するに止まらず、記述内容を追加することも提案された Müller-Merbach [24] は体系的な記号命名法、オブジェクトの識別子として数理モデルに使われる添字 (index) を導入し、また、オブジェクトの諸属性の関数を記述する方法を提案する。Lazimy [20] は行動 / 決定 (action/decision) 属性、変換 (transformation-based) 属性、目的 / 評価 (objective/performance) という 3 種類の属性を導入することを提案する。Choobineh[7] は実体 - 関連ダイアグラムに演算記号を導入すること、また、関連の集合の最小要素数、最大要素数を記述することを提案する。最小要素数、最大要素数は SM にも見られる行列の疎 (sparse) 構造記述の特殊な形である。Amatürket *et al.*[1] は線形計画法のソルバーへの入力データを SQL を利用して生成する際の添え字付け (indexing) の方法を提案する。

数理モデルに関する付加的情報を記述する提案もある。Bhargava and Kimbrough[5] は記号の説明、出典、作成者、作成日というような数理モデルに関する情報を (モデル記述の中のコメントとしてではなく) 実行型の形式で記述する埋め込み法 (embedding approach) を提案する。Mannino *et al.*[22] や Hong *et al.*[15], [16] は問題の存在する領域に関する数理的情報を領域世界 (domain world) として記述し、これを数理モデルに結合することを提案する。Banerjee and Basu[2] は問題に適したモデルやソルバーを選択可能にするために、数式モデルを 4 階層に分けて記述し、その各階層で数式モデルと意思決定の論議領域に関する情報を関連づけることを提案する。

これらの先行研究は数理モデル、あるいは、数理モデルに対する付加的な情報の記述を扱っている。これに対して、本論文は数理モデル構築の対象である意思決定問題そのものの記述 (問題定義) を対象とする。問題定義には、数理モデルを構築するために必要な情報は全て含まれていなければならない。また、問題定義は非数理的な記述が基本となるが、数式が含まれても構わない。

順序づけ問題の分枝限定法を開発する研究者を支援するシステムにおける問題記述の方法として項目 - 詳細 (items-details) アプローチ (関口 [28]) があるが、その項目は限定され、記述可能な問題は限られている。

関口・鮑・向原 [29] は、実体 - 関連概念を利用した問題定義の記述アプローチ (GERM) によって、項目 - 詳細アプローチでは実現することが困難であるような、幾つかの現実的なスケジューリング問題を記述できることを示す。そこでは、幾つかの APS ソフトウェアが対象とする問題の GERM アプローチによる記述可能性が主に検討される。しかし、GERM の一般的な記述様式は明示されていない。

3. GERM による問題定義の概要

3.1. 問題定義とモデル・クラス

Geoffrion[10] は数理モデルの具体例 (= 数値例) とモデル・クラスを区別することを提案する。モデル・クラスは「あらゆる類似の具体例のコレクション」であり、いくつかのパラメータを含む 1 つの数理モデルに対応すると考えられる。具体例は数理モデルの全てのパラメータに対するデータである。SM では、具体例のデータは一連の要素詳細表 (elemental detail table) に記載され、モデル・クラスは要素グラフ、属 (genus) グラフ、モジュール木

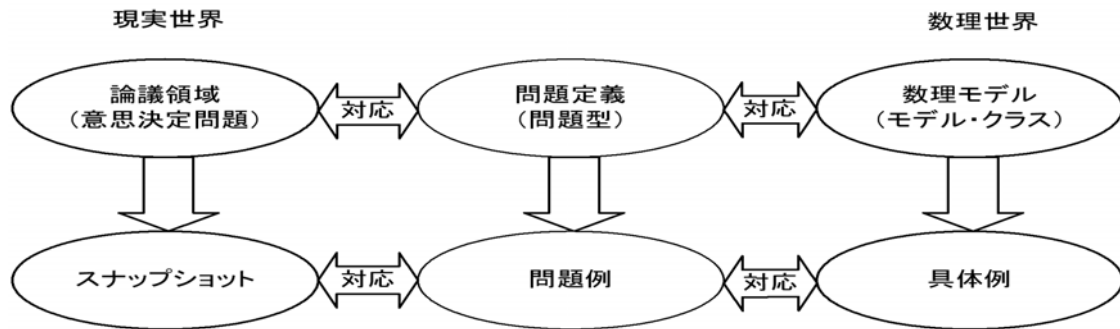


図 1: 現実世界と数理世界の対応

(modular tree), スキーマなどによって記述される。しかし, モデル・クラスの記述では関数や式が直接記述され, 関数や式によって記述される論議領域の構造的特性を記述する非数理的方法がSMにはない。つまり, モデル・クラスの記述はパラメータを含む数理モデルに関する記述であって論議領域の直接的記述ではない。

モデル・クラスや具体例は, それが現実世界の意思決定問題を表現するものである限り, 対応する論議領域を持つ(図1)。この対応を明らかにするには, 意思決定問題を構成する諸特性量とモデルのパラメータおよび関数や式を関係づける必要がある。意思決定問題を記述したものが問題定義であり, 前提3により対象とする意思決定問題は複数の具体例に対応する数理モデルを必要とするから, 問題定義はモデル・クラスに対応する。

3.2. オブジェクト型の概念

GERMは, データベース設計分野で利用される実体-関連モデル(Edmond[8])の拡張である。この実体-関連モデルに基づき, 本研究で利用される諸概念を次のように定義する。実体(entity)を他のオブジェクトとは独立に識別できるオブジェクトと定義する。実体でないオブジェクトは他のオブジェクトに依存して識別される。このようなオブジェクトを関連(relationship)という。つまり, オブジェクトは実体か関連のどちらかに分類される。オブジェクトの属性を属性名とその値(属性値)の対と定義する。関連の属性は他の複数のオブジェクトの組合せに対して定義される。つまり, 関連は他の複数のオブジェクトの相互関係を記述する。

属性名と同じ集合に値を与えることによって記述できるオブジェクトは互いに類似(similar)であるという。論議領域に存在する類似なオブジェクトの集合をオブジェクト型(type)という。オブジェクト型には, 実体の集合である実体型と, 関連の集合である関連型がある。オブジェクト型の要素である個々のオブジェクトをその型のインスタンスという。例としてジョブショップ問題を考える。ジョブショップ問題の論議領域には, ジョブ, 機械, 作業などのオブジェクトが観察される。個々のジョブや機械は実体である。ジョブを構成する作業は, ジョブと機械の組合せ(相互関係)に対応すると考えられるので関連である*。

*実体型, 関連型の区別, オブジェクトとその属性の区別は必ずしも唯一でない。この点についてはEdmond[8]を参照されたい。従って, GERMアプローチによる問題定義は, 一つの論議領域に対して唯一ではない。また, 常識的には同じ種類と考えられるオブジェクトでもそれが存在する論議領域によって実体とすべきか関連とすべきかの判断が異なる場合がある。例えば, 作業は5.2.では実体として記述される。この場合は, 作業型のインスタンスを表1-(a), (b)の機械やジョブのように通し番号等の識別名によって識別することになる。しかし, ある作業が, どの機械のものに対応しているのか, もしくは, どのジョブに対応したものであるのかについて別途明示的に記述する必要があり, 表1-(c)と比べて煩雑になる。実体として扱うか, 関連として扱うかの選択は, 記述の柔軟性と簡便性の両面からよく検討する必要がある。

ある時点に，この論議領域に存在するすべてのジョブの集合は一つの実体型である．ジョブ（実体）型と呼ぶ．すべての機械の集合はもう一つの実体型（機械型）である．ある時点の作業は，その時点のジョブ型と機械型の要素の対に対応し，作業型を構成する．

論議領域のすべてのオブジェクト型（実体型および関連型）が，考察の対象となる期間にわたって不変と考えられるなら，意思決定問題に関するすべてのデータは変化しない．この時，この論議領域に対応する問題定義はただ1つの問題例に対応する．本研究では，モデル・クラスに対応する問題定義を対象とするから，論議領域には時間と共に変化するオブジェクト型が存在する．そのようなオブジェクト型の中でインスタンス自体が時間と共に別のインスタンスに置き換わるものを未定（open）であるという．これに対して，インスタンス自体は時間が変わっても置き換わらないオブジェクト型を固定（fixed）であるという．本研究では，未定なオブジェクト型を未定オブジェクト型，固定なオブジェクト型を固定オブジェクト型と呼ぶことがある．ジョブショップ問題の論議領域では，通常，機械型は固定であり，ジョブ型や作業型は未定である．しかし，現実世界では変化しないオブジェクト型は存在しないと言って良いから，固定・未定の区別は便宜上のものである．例えば，いくつかの機械が使用不能に陥る場合があり，原則として異なる時点では負荷の対象となる機械の集合が異なる場合を考える．機械型を固定と考え，各時点ではその状態に応じて作業を割り付けるものと考えることができる．また，機械型を未定であるとし，各時点においてそのインスタンスを記述し直すとも考えることもできる．前者の場合，問題定義における機械型に含まれる機械の集合は不変であるが，各機械にはその状態を記述する属性が必要になる．後者の場合には機械型に含まれる機械の集合が各時点で異なる．

GERMによる問題定義では，論議領域を構成する実体型と関連型，並びに，固定オブジェクト型のすべてのインスタンスを記述する[†]．これに対して一般の実体 - 関連アプローチでは，個々の実体と関連を記述するが，実体型や関連型の記述はほとんどない．このことを次に説明する．

3.3. オブジェクト型の記述

モデル・クラスとして Johnson[18] の古典的な 2 機械フローショップ問題 $F2//C_{max}$ を取り上げて，GERM の問題定義とモデル・クラスの具体例の記述の相違を述べる．3 個のジョブからなる $F2//C_{max}$ の具体例は，表 1 のように記述できる．

表 1: 2 機械フローショップ問題 $F2//C_{max}$ の具体例

| (a) 機械 機械識別名 | (b) ジョブ ジョブ識別名 | (c) 作業 | | |
|-----------------|-------------------|--------|--------|------|
| | | 機械識別名 | ジョブ識別名 | 処理時間 |
| 1 | 1 | 1 | 1 | 5 |
| 2 | 2 | 1 | 2 | 10 |
| | 3 | 2 | 1 | 3 |
| | | 2 | 2 | 2 |
| | | 3 | 1 | 7 |
| | | 3 | 2 | 8 |

[†]インスタンスには実体型のものと同関連型のもの双方がありうる．未定オブジェクト型のインスタンス（例えば表 1-(c)）は，その定義から時間とともに変化するデータを含むため，問題定義には含まれない．

これらは各オブジェクト型のインスタンスの属性値を一覧表にしたもので、SMの要素詳細表に相当し、一般の実体-関連アプローチはこれを記述する。慣例的には作業の表(表1-(c))だけが示されるが、実体と関連を記述するということを明示するために機械表(表1-(a))、ジョブ表(表1-(b))も付した。しかし、これらの表はF2//Cmaxの具体例データであって、数理モデルを作成するのに必要な多くの情報を含んでいない。そうした情報(の一部)を表2に示す。これらはF2//Cmaxに対応する問題定義に記載されなければならない。

表2はオブジェクト型の特性を記述していることに注意されたい。また、機械型は固定であるため、ここにも機械表(表2-(d))が示されている。表2-(d)には、表1-(a)にある識別名だけでなく、モデル化に必要な情報として、各機械のサービス規則も記載されている。表2の情報に基づいて数理モデルが作成される。その具体例データが表1である。表2で太字の型属性は表2-(d)、および表1-(b)、(c)の属性となっている。表2-(c)の型属性名に「識別名」が現れないのは、「定義型」で代用できるためである。表1-(c)で明らかのように、特定の作業(作業型のインスタンス)は「ジョブ-識別名」と「機械-識別名」の組で識別可能である。作業型における「定義型」の属性値を「ジョブ型、機械型」とすることで、これを記述している。

F2//Cmaxの問題定義[‡]は、その論議領域が表2に示した3つのオブジェクト型で構成され、意思決定問題が総所要時間を最小にする作業の処理スケジュールを求めることであること、を記述することで完成する。

表2、表1および意思決定問題の記述から以下のことが読み取れる。このショップの機械は2台であり通し番号で呼ばれる。同時には多くても一つの作業しか処理できず、機械1はスケジュールの順に、機械2は到着順にジョブ(作業)を処理する。ジョブは n 個で時によって変化し、どれも分割せずに機械1で処理した後、機械2で処理される。ジョブを構成する作業は正整数で表される処理時間を持ち、処理の中断は許されない。総所要時間を最小化することがスケジューリングの目的である。具体例はジョブが3個の場合で、作業の表が各ジョブに対して機械ごとに1個、合わせて2個の作業を含むのは偶然ではなくこのショップの特性である。

これは、Johnson[18]にある次のような問題説明に含まれる情報を含み、かつ、定式化に必要なその他の情報(各機械のサービス規則は順列スケジュールを求めることを示す)も含んでいる。

「 n 個の品物があり、それぞれ第1の加工段階(あるいは機械)をってから第2の加工段階を通らなければならない。各加工段階には1台ずつしか機械がない。また、1つの品物は、一度に高々1台の機械にしかかけられない。」 $2n$ 個の任意の正の定数 $A_i, B_i, i = 1, 2, \dots, n$ を考える。 A_i は、品物 i を最初の機械にかけるための段取り時間プラス加工時間とし B_i は2番目の機械の同様の時間とする。そこで、この n 個の品物を生産するのに要する総所要時間を最小にする最適なスケジュールを求めよ。」

以上から、意思決定問題を記述するには、オブジェクト型と固定オブジェクト型の各インスタンスを記述することが必要なことが分かる。オブジェクト型の記述には、そのインスタ

[‡]「F2//Cmaxの問題定義」という表現はF2//Cmaxが数理モデルであることから、本論文の立場(図1)に矛盾する。本来であればナマの現実世界を論議領域としてその問題定義を行うべきである。しかし、F2//Cmaxは良く知られた数理モデルで、その「仮想的な」現実世界のイメージも多くの文献に解説されている。こうして読者に共有されていると考えられる「仮想的」論議領域を前提として、以下ではF2//Cmaxの問題定義を行う。このような便法を使う理由は、ナマの現実世界を読者と共有することは困難だと考えるからである。第5節においてAPSソフトウェアが解くスケジューリング問題を使うのも同じ理由による。

表 2: F2//Cmax の論議領域を構成するオブジェクト型とオブジェクト

| (a) 機械型 | | (b) ジョブ型 | |
|----------------|----------------------|----------|---------|
| 型属性名 | 型属性値 | 型属性名 | 型属性値 |
| 識別名 | 通し番号 | 識別名 | 通し番号 |
| 要素数 | 2 | 要素数 | n |
| サービス規則 | (スケジュール順, FIFO) | 技術的順序 | 1→2 |
| 同時処理作業数 | 1 | ジョブ分割 | 不可 |
| (c) 作業型 | | (d) 機械 | |
| 型属性名 | 型属性値 | 識別名 | サービス規則 |
| 定義型 | ジョブ型, 機械型 | 1 | スケジュール順 |
| 最小最大出現数 | 2 | 2 | FIFO |
| 要素数 | 2×n | | |
| 処理時間 | 正整数 | | |
| 中断 | 不可 | | |
| (e) F2//Cmax 型 | | | |
| 型属性名 | 型属性値 | | |
| 構成型 | 機械型 (固定), ジョブ型, 作業型 | | |
| 最大完了時刻 | これを最小にするジョブの処理順を決めたい | | |

ンスの属性の範囲(例: 処理時間)に加えて, 他の多くの属性の記述が必要である. そこで GERM では, オブジェクト型の記述に必要な属性を型属性 (type-attribute) と呼ぶ. インスタンスの記述に必要な属性は, これと区別してインスタンス属性と呼ぶ. また, 表 1 の具体例は, 各オブジェクト型のある時点におけるインスタンスの記述であり, 論議領域のある時点での状況であるスナップショット (snapshot) に対応する (図 1).

4. GERM による問題定義の様式

4.1. 実体 - 関連ダイアグラム風記述の併用

図 2 は表 2 の F2//Cmax を実体 - 関連ダイアグラム風に視覚化したものである. このような図はオブジェクト (型) の相互関係を把握しやすいので併用するが, GERM による問題定義にとって本質的なものではない. 二重の長方形と菱形は実体型と関連型を示し, 単線の長方形は (固定オブジェクト型の) 個々のインスタンスを示す. 二重の形状を使うのは, GERM ではオブジェクト型 (集合) の記述が中心であるためである. これに対して一般の実体 - 関連ダイアグラムでは単線の長方形や菱形を使い, 個々のインスタンスの情報を記述する. 図 2 で機械 1 と機械 2 を単線で示したのは, その様式を踏襲したものである. なお, 機械型のインスタンスが記述されるのは, 機械型が固定だからである. また, GERM ではこれら一連のオブジェクト (型) は一つのシステムを構成すると考え, これをシステム型 (4.3. 参照) と呼ぶ. 図 2 の F2//Cmax はシステム型の一例である. GERM による問題定義は, このような一つ以上のシステム型から構成される. システム型も他のオブジェクト型と同様に型属性を持つ. 「最大完了時刻」はその一例である. 数理モデルとしての F2//Cmax の目的関数 (総所要時間) はこのシステム型の型属性「最大完了時刻」に対応づけられる.

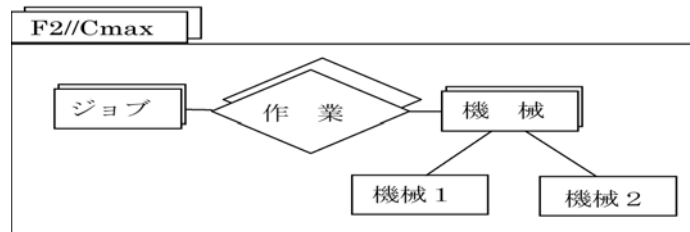


図 2: F2//Cmax の実体 - 関連ダイアグラム風記述

4.2. 型属性の記述様式

GERM における記述の基本要素は，次のような記述様式に従う型属性の記述である．

<型属性名> (<型属性の実質的意味の説明>): <インスタンス属性かどうか>, <データ形式> <データ領域>, <次元> [<単位>], <性質>, “<データ値の説明やデータ値が満たすべき条件の記述>”;

ここで <> は欄の値, (), [], および “ ” は欄の値の区切り, コロン (:) とコンマ (,) は欄区切り, セミコロン (;) は 1 つの型属性の記述の終わりを各々示す．各欄の記述の意味を表 3 に示す．

表 3: 型属性の記述様式における各欄の意味

| 項目 | 内容 |
|-----------------------------|--|
| <型属性名> (<型属性の実質的意味の説明>) | 属性の識別名．かつこ内は必要に応じて記入する型属性の実質的意味の説明である． |
| <インスタンス属性かどうか> | この欄の値は‘インスタンス属性’または‘ ’．‘インスタンス属性’はこの属性がこの型の各インスタンスの属性と成ること, ‘ ’はこの属性が集合的特性であってインスタンス属性とはならないことを表す． |
| <データ形式> <データ領域> | <データ形式> はこの属性の値がどのように表記されるかを表す．例：文字列, 実数, 整数, 集合, 年月日, 時分秒． <データ領域> はインスタンス属性の場合は各インスタンスが取りうるこの属性の値の範囲, そうでない場合は集合的特性の値． |
| <次元> [<単位>] | この属性の次元と単位．例：なし[なし], なし[個], 長さ[Km], 距離[マイル], 時刻[年月日], 時刻[時分秒], 金額[円]． |
| <性質> | 論議領域におけるこの属性の特性．変数：他の属性の値から計算される, 固定値：不変の値, パラメータ：スナップショット毎に変化, 決定変数：意思決定者が求めたい値, 識別子：各インスタンスを識別するのに使う属性, のどれかとする． |
| “<データ値の説明やデータ値が満たすべき条件の記述>” | この属性値が満たすべき条件, 型属性値やインスタンス属性値の説明． |

4.3. オブジェクト型とシステム型の記述様式

GERM による問題定義における個々のインスタンスの記述は，良く知られている実体や関連の表である．論議領域の意思決定問題を記述するのに，GERM では実体型，関連型およ

<オブジェクト型名> (<オブジェクト型の実質的意味の説明>);
 識別名 (): インスタンス属性, <データ形式> <値の領域>, なし [なし], 識別子,
 “<識別名の値に関する説明>”;
 要素数 (): , 正整数 <値の領域>, なし[<要素数の数え方>], <性質>, “<データ値
 の説明やデータ値が満たすべき条件の記述>”;

(a) 実体型の場合

<オブジェクト型名> (<オブジェクト型の実質的意味の説明>);
 定義型 (): , 集合 <‘オブジェクト型名 (関係の説明句)’ のリスト>, なし [なし],
 固定値, “<データ値の説明やデータ値が満たすべき条件の記述>”;
 最小最大出現数 (): , 集合 <‘オブジェクト型名 (m: n)’ のリスト>, なし [なし],
 固定値, “<データ値の説明やデータ値が満たすべき条件の記述>”;
 要素数 (): , 正整数 <値の領域>, なし[<要素数の数え方>], <性質>, “<データ値
 の説明やデータ値が満たすべき条件の記述>”;

(b) 関連型の場合

<システム型名> (<システム型の実質的意味の説明>);
 構成型 (): , 集合 <‘オブジェクト型’ のリスト>, なし [なし],
 固定値, “<データ値の説明やデータ値が満たすべき条件の記述>”;

(c) システム型 (問題型)

図 3: GERM による問題定義記述の定型的部分

びシステム型を記述する。その記述様式は、図 3 に示す定型的部分に、オブジェクト型に固有の型属性の記述を付加したものである。図中、ゴシックで示した部分是不変である。システム型とは、その構成型に列挙されるオブジェクト型が連結性と自己完結性を満たすものをいう。連結性とは、すべての実体型が関連型の定義を通して相互に連結していることである。自己完結性とは、各関連型の定義型が構成型に含まれることである。システム型の構成型に含まれるオブジェクト型の中に、決定変数となる型属性を含むものがあるとき、このシステム型は意思決定性を満たすという。問題型とは意思決定性を満たすシステム型をいう。なお、意思決定の善し悪しを評価する属性はシステム型の型属性として、オブジェクト型の型属性と同じ形式で記述する。

F2//Cmax の論議領域が図 2 のオブジェクト型で構成されとした場合、GERM による問題定義は図 4 のようになる。F2//Cmax には、Johnson 以後の研究によって様々な属性が追加されてきた。例えば、図 2 には記述してないが、遅れ時間（機械 1 における開始時刻、完了時刻と機械 2 における開始時刻・完了時刻の相互関係に関する制約）はそうした属性の一つである。すなわち、ある意思決定問題を定義するのに明示的に指定すべき型属性は、意思決定者や意思決定支援者の論議領域に関連する知識にも依存する。例えば、APS ソフトウェアが解く一般的なジョブショップ問題の特殊形として F2//Cmax の問題定義をする場合、はるかに多くの属性を記述することが必要になる（5.6. 参照）。

5. GERM による問題定義の特徴と有効性

5.1. 問題定義と APS ソフトウェアへのデータ入力様式の相違

APS ソフトウェアは実際の工場における生産計画・スケジューリング問題に対する実用的なソルバーであるとされる。APS ソフトウェアへの入力データには、製造に関するマスターデータ、スケジューリングロジックのオプション、スケジューリングの結果に対する修正、

機械型 (ショップ内の機械集合);

- 識別名 (機械に固有の名称): インスタンス属性, 文字列 { 1, 2 }, なし [なし], 識別子, “”;
- 要素数 (スナップショットにおける機械台数): , 正整数 { 2 }, なし [台], 固定値, “”;
- サービス規則 (到着ジョブの着手順): インスタンス属性, 文字列 { スケジュール順, FIFO }, なし [なし], 固定値, “スケジュール順: 決定する順番, FIFO: 到着順”;
- 同時処理作業数(): , 正整数 { 1 }, なし [個], 固定値, “”;

ジョブ型 (本ショップで処理するジョブ集合);

- 識別名 (ジョブに固有): インスタンス属性, 文字列, なし [なし], 識別子, “”;
- 要素数 (スナップショットにおける作業数): , 正整数, なし [個], パラメータ, “”;
- 技術的順序 (ジョブを構成する作業の処理順序): , 文字列 { 1→2 }, なし [なし], 固定値, “1→2: 機械 1 での作業の次に機械 2 での作業を行う”;
- ジョブ分割 (ジョブを分割できるかどうか): , 文字列 { 分割不可 }, なし [なし], 固定値, “”;
- 処理順序 (ジョブを処理する順番): インスタンス属性, 正整数 { 1 ~ ジョブ要素数 }, なし [なし], 決定変数, “ジョブ全体の中で何番目に機械 1 の作業を開始するかを記述する”

作業型 (ジョブを構成する作業集合);

- 定義型(), 集合 { ジョブ型 (含む), 機械型 (処理する) }, なし [なし], 固定値, “”;
- 最小最大出現数(): , 集合 { ジョブ型 (2:2), 機械型 (ジョブ要素数: ジョブ要素数) }, なし [なし], 固定値, “ジョブは機械 1 と機械 2 の 2 つの作業から成る”;
- 要素数 (スナップショットにおける関連数): , 正整数 { 2 × ジョブ要素数 }, なし [個], パラメータ, “”;
- 処理時間 (処理に要する時間): インスタンス属性, 非負整数, なし [U-time], パラメータ, “”;
- 中断 (着手後の中断の可否): , 文字列 { PROHIB }, なし [なし], 固定値, “PROHIB: 中断不可”;
- 開始時刻 (処理に着手する時刻): インスタンス属性, 非負整数, 時間 [U-time], 変数, “条件: 機械 1 での先頭作業の開始時刻=0”;
- 完了時刻 (処理を完了する時刻): インスタンス属性, 非負整数, 時間 [U-time], 変数, “条件: 開始時刻 + 実処理時間 = 完成時刻”;

F2//Cmax 型 (古典的な二機械フローショップ問題)

- 構成型(): , 文字列 { 機械型 (固定), ジョブ型, 作業型 }, なし [なし], 固定値;
- 最大完了時刻 (機械 2 で最後に処理する作業の完了時刻): インスタンス属性, 正整数, 時間 [U-time], 変数, “これを最小にするジョブの処理順を決めたい”;

図 4: F2//Cmax の GERM による問題定義

注文情報などの問題例データ作成に必要な基礎データが含まれる。これらのデータからは、論議領域のスナップショットの記述が生成されると考えられる。しかし、APSソフトウェアは、部品引き当て計画、在庫補充計画、スケジューリングなど多くの機能を持ち、これらの機能の相互関係は一般にAPSソフトウェアごとに異なる。また、実際に解かれるスケジューリング問題がどのようなものになるかは、APSソフトウェアをどんな設定で利用するかによっても異なる。各APSソフトウェアは固有のデータ入力様式を定めており、それはモデル記述方式の一種と考えることができる。しかし、その主要な目的は、スケジューリング問題の具体例データを入力することであり、スケジューリング問題そのものの記述を志向するものではない。以上から分かるように、APSソフトウェアの入力データによって記述される生産計画・スケジューリング問題を特定するのは一般に容易ではない。

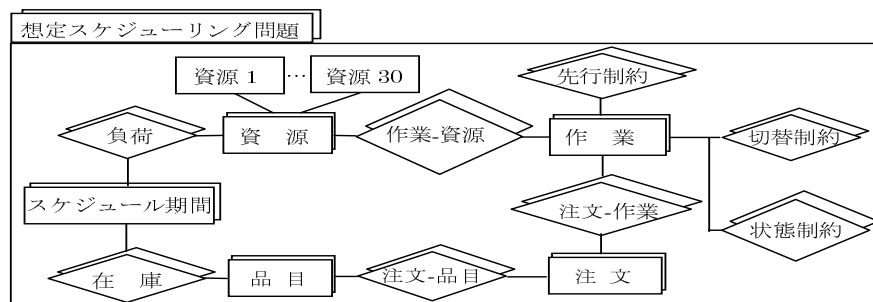


図 5: 想定スケジューリング問題の実体 - 関連ダイヤグラム風記述

本節ではGERMによる問題定義が実際的なスケジューリング問題の記述にも適用できることを示すために、例としてAPSソフトウェアに対する汎用的なモデル記述言語を志向したPSLの仕様書（法政大学情報マネジメント研究室[17]）から想定される1つのスケジューリング問題を使用する。これは図5に示すようなシステム型となる。なお、この問題では工場の資源（設備）はスナップショットを通じて不変であると仮定する。

GERMによる問題定義は、図5の各オブジェクト型に対する図4に示したような記述と、固定オブジェクト型である資源型のインスタンスの表とから成る。しかし、全体を示すことは紙幅を要するので、詳細は関口・鮑・向原[29]を参照されたい。以下では、いくつかのオブジェクト型を取り上げて、スケジューリング問題の記述におけるGERMの特徴や有効性を説明する。

5.2. 型属性と意思決定問題の構造的特性

図6は想定スケジューリング問題の資源型と作業型の記述である。資源型の要素数は固定値で、インスタンス属性となる型属性に第四欄の値が固定値のものがあることから、資源型が固定であることが分かる。固定なオブジェクト型のインスタンスの表では、性質欄の値が固定値とされたインスタンス属性の欄に特定のデータ値を記載する。例えば、資源型では、識別名、負荷上限、負荷下限、優先度がそのような属性であり、そのインスタンスの表は30行から成る。

図6で作業型は実体型であり、図4とは異なっている。APSソフトウェアでは、各品目の生産情報をマスターデータとして保持し、どれかの品目の生産要求である注文が発生すると、このマスターデータから作業データを生成するのが一般的である。各作業には代替資源が指定されることも多い。それをどの資源を使っていつ実施するかを、負荷バランスを取りながら決定するのである。従って、APSソフトウェアを考えるには、作業を実体とする方が都合がよいと考えられる。

作業型は図4の場合に較べて、多数の型属性を持つ。これは、例えば、中断の値の増加と中断時間の指定、優先度の指定、納期の指定などの影響を考えれば、想定スケジューリング問題の複雑性を反映していることが分かる。このように必要に応じて、オブジェクト型や型属性の種類を増減し、型属性値の範囲を拡大・縮小することで、GERMは極めて多様なスケジューリング問題を記述できる。

図6を図4の資源型と比較すれば、同時処理作業数とサービス規則がなく、他方、負荷上限、負荷下限および優先度が追加されている。このことと関連して、作業-資源型（図7）に負荷量なる型属性があり、作業の処理の資源への負荷量を記述する。この負荷量を固定値1とし、資源型の負荷上限を固定値1とすれば、図4で同時処理作業数を1と指定するのと同様の記述ができる。図6の資源型にサービス規則がないのは、資源の負荷量を調整しながら各作業の開始時刻と完了時刻を決めるのが、APSソフトウェアの機能であるからである。そのため資源型と作業型には優先度なる型属性があって、作業をどのような順番で、どの資源に優先的に負荷山積みするかを記述する。これはAPSソフトウェアが必ずしも非遅れ（non-delay）スケジュールを求めないことの反映である。

要素数、中断、最大中断回数などはインスタンス属性とならない型属性である。また、すべての資源の負荷上限が1、すべての作業の負荷量が1となる場合、これらの属性値が、すべてのスナップショットにおいて、不変であるから、固定値として良いことを意味する。つまり、

負荷上限（実行可能な最大負荷）： 負荷上限 ，正整数 { 1 }，なし [台]，固定値， “”；

のように，インスタンス属性とならない型属性として記述できる．これらの型属性が数理モデルの型に重要な影響を与えることは良く知られている．このように，インスタンス属性とは成らない型属性は一般に数理モデルの型を左右するから，意思決定問題の構造的特性を記述すると考えられる．

属性には数量的なもの（＝数量属性）とそうでないもの（＝定性属性）がある．数量属性が問題の構造的特性を大きく左右することもある．例えば資源型の資源数が1，負荷下限が0，負荷上限が1，作業－資源型の負荷量が1で，これらが固定値であれば，これは1機械のジョブショップ問題であることが分かる．このように，固定値である数量属性は構造的特性を記述すると考えられる．しかし，経験的には定性属性が構造的特性を左右する可能性の方がはるかに高い．例えば，作業型の中断は定性属性であるが，図6では図4の場合と較べて，インスタンス属性となる点と値が4種類に増えた点で，意思決定問題の構造を複雑にしている．このことは，中断時間がインスタンス属性として作業毎に指定されることから，さらに増幅される．

なお，型属性における次元 [単位] の記述は多様な記述ができなければならないことから，パラメトリックな記述が許される．例えば在庫量は，セット，重量，体積，台数，個数，ダース数など品目によって異なる次元や単位で管理される．従って，D1 [U1] のようにパラメトリックに記述し，インスタンス毎に次元 [単位] が異なること，かつ，インスタンスごとに一貫して使い分けられなければならないことを示す．

資源型（本工場が所有する生産資源の集合）；

- 識別名（資源に固有の名称）：インスタンス属性，文字列，なし [なし]，識別子，“”；
- 要素数（スナップショットにおける資源数）： 要素数 ，正整数 { 30 }，なし [個]，固定値，“”；
- 負荷上限（実行可能な最大負荷）：インスタンス属性，正整数，なし [台]，固定値，“”；
- 負荷下限（実行可能な最小負荷）：インスタンス属性，正整数，なし [台]，固定値，“”；
- 優先度（負荷決定の優先順位を表現）：インスタンス属性，正整数 or ナル値，なし [なし]，固定値，“”；

作業型（本工場で処理する作業の集合）；

- 識別名（作業に固有）：インスタンス属性，文字列，なし [なし]，ID，“”；
- 要素数（スナップショットにおける作業数）： 要素数 ，正整数，なし [個]，パラメータ，“”；
- 処理時間（作業着手から完成までの所要時間）：インスタンス属性，非負整数，時間 [U-time]，パラメータ，“値0はダミー作業”；
- 優先度（スケジュールする際の優先度）：インスタンス属性，正整数，なし [なし]，パラメータ，“”；
- 中断（着手後の中断の可否，中断タイプ）：インスタンス属性，文字列 { PROHIB, DELAY, INSRT, RCHNG のどれか }，なし [なし]，パラメータ，“PROHIB：中断不可，DELAY：中断中に他作業挿入不許可，INSRT：中断中に他作業挿入可，RCHNG：中断し他資源へ移動可”；
- 中断時間（中断する場合に許容される最大中断時間）：インスタンス属性，正整数かナル値，時間 [U-time]，パラメータ，“条件：中断の値が PROHIB の時はナル値”；
- 最大中断回数（中断の回数の最大許容値）： 最大中断回数 ，非負整数 { 1 }，なし [回]，固定値，“”；
- 最早着手可能時刻（作業の処理に着手できる最も早い時刻）：インスタンス属性，非負整数，時間 [U-time]，パラメータ，“”；
- 納期（作業を完成させるべき時刻）：インスタンス属性，非負整数，時間 [U-time]，パラメータ，“条件：最早着手可能時刻 + 実処理時間 = < 納期”；
- 開始時刻（処理に着手する時刻）：インスタンス属性，非負整数，時間 [U-time]，決定変数，“条件：最早着手可能時刻 = < 開始時刻”；
- 完成時刻（処理を完了する時刻）：インスタンス属性，非負整数，時間 [U-time]，決定変数，“条件：開始時刻 + 実処理時間 + 実中断時間 = 完成時刻”；

図 6: 資源型と作業型の記述

5.3. 関連型の記述

各作業がどの資源を必要とするかは，作業と必要資源を組にすることで記述できる．GERM ではこのような組の特性を関連型の型属性として記述する．図7の作業－資源型はそのよう

な関連型である。

関連型の定義により，そのインスタンス属性の値は，定義型の各オブジェクト型から取ったインスタンスの組合せに対応して決まる．定義型の各インスタンスの識別名をインデックスとすれば，関連型のインスタンス属性値はマトリクスを構成する．数理モデル構築の観点からすれば，このマトリクスの特性は重要である．例えば，表2の作業型の表から表1の作業の表の構造を正確に知ることができなければ，3.3. で取り上げた Johnson[18] の問題説明におけるパラメータ ($A_i, B_i, i = 1, 2, \dots, n$) を作業に割り当てることができない[§]．その意味で最小最大出現数は，意思決定問題の構造的特性を記述する重要な属性である．また，先行関係型の型属性「パターン」は各注文に結びつけられるインスタンスの集合によって構成される技術的順序が非巡回になるような作業の組合せだけを許容することで，間接的に定義型のインスタンスの組合せの範囲を規定している．なお，関連型のインスタンスの識別には（表1でのように）それを定義するオブジェクト型の各インスタンスの識別名の組を使うことができる．しかし，固有の識別名が必要であれば型属性として別に定義しても良い．

最小最大出現数は伝統的な実体 - 関連モデルで記述される「対応関係」を一般化した型属性である．対応関係は定義型の各インスタンスが関連型のインスタンスを規定するとき，1:1，1:多，多:1あるいは多:多のうちどの対応関係を持つかを記述する．作業 - 資源型の最小最大出現数の値に，作業型 (1:n) とあるのは，どのスナップショットでも各作業が少なくとも1個以上の資源と組み合わせられることを示す．他方，資源型 (1:n) とあるから，各資源はすべてのスナップショットにおいて，少なくとも1つ以上の作業の処理に利用できることを示す．

APS ソフトウェアのモデル記述には負荷計算方式や作業効率のデータが現れる．作業と資源の組合せに応じて負荷の計算方式や作業効率が異なるからである．GERM ではこれに対して，注文ごと作業 - 資源の組合せごとに必要な計算をして処理時間を決定する．従って，負荷計算方式や作業効率はマスターデータとしては必要でも，問題定義の記述には現れない．作業を処理できる資源の組合せが複数ある場合には，その作業を必要数複製して各作業を処理できる資源の組合せがただ一つとなるようする．（これは，作業間の相互関係を記述する場合に，作業を処理する資源の組合せが確定することが必要な場合もあり，そのような場合の記述が容易になるようにする GERM に特徴的なテクニックである．）さらに，この複製に則して作業 - 資源型も再構築され，そこでは負荷量も，負荷計算と注文数量の値に応じて実際の値を算出して置き換えるものとする．

先行制約は構造的特性である．図7の先行関係型の型属性のなかで指定時間だけがインスタンス属性となる数量属性であり，他は定性属性である．数理モデルが作られる場合，例えば，タイプは開始あるいは終了時刻の関係を規定する制約不等式として表現される．パターンや範囲は数理モデルの型を特徴づける重要な情報である．例えば，先行関係型の範囲を「同一注文（工程）内」，パターンをフロー（すべての注文の技術的順序が同じで，分枝を含まない），最小最大出現数を作業型 (from) (0:1)，作業型 (to) (0:1) として，工程型に必要な修正を施せば，2工程のフローショップを表現できる（ただし，2機械であるかどうかは資源の記述による）．また，タイプはインスタンス属性となる定性属性である．この工場では先行関係を指定する作業対ごとにタイプを指定するが，いわゆるジョブショップ問題ではESがすべての先行関係に共通して指定されることも多い．その場合にはタイプは定性属性であり，インスタンス属性にならない型属性となる．

[§]Johnson 問題の問題説明は，数理モデルの意味的説明であって，本論文でいう問題定義ではない．

作業 - 資源型 (作業と資源の関係の集合);

定義型 (): , 集合 { 作業型 (使う), 資源 (処理する) }, なし [なし], 固定値, “”;
 要素数 (スナップショットにおける関連数): , 正整数, なし [個], パラメータ, “”;
 最小最大出現数 (): , 集合 { 作業型 (1 : n), 資源型 (1 : m) }, なし [なし], 固定値, “”;
 タイプ (資源が主資源かどうか): インスタンス属性, 非負整数, なし [なし], パラメータ, “ 0 : 主資源, 0 以外 : 副資源, 同じタイプの資源のうち一つだけ選択利用 ”;
 負荷量 (占有する資源量): インスタンス属性, 非負整数, なし [台], パラメータ, “ デフォルト値は 1 ”;

先行関係型 (作業間の先行関係 - 代替的關係も含む - の集合);

定義型 (): , 集合 { 作業型 (from), 作業型 (to) }, なし [なし], 固定値, “”;
 最小最大出現数 (): , 集合 { 作業型 (from) (0 : n), 作業型 (to) (0 : n) }, なし [なし], 固定値, “”;
 要素数 (先行関係を指定する作業対の個数): , 正整数, なし [なし], パラメータ, “”;
 ボタン (含まれる作業の先行関係 - 代替的關係を除く - に関する制約): , 文字列 { アサイクリック }, 固定値, “ 作業をノードとし, 先行関係を矢印とするネットワークが巡回路を持たないものであることだけを要求 ”;
 範囲 (先行関係 - 代替的關係を除く - を指定できる作業の範囲): , 文字列 { 異注文間および同一注文内 }, なし [なし], 固定値, “”;
 タイプ (先行制約の特性): インスタンス属性, 文字列 or ナル値 { ES, SS, EE, ESS, SSS, EES, ESR, SSR, EER, ALT }, なし [なし], パラメータ, “ ナル値はデフォルト値で ES を指定, ES : 先行作業が完了後に後続作業に着手可, SS : 先行作業が開始すれば後続作業も開始可, EE : 先行作業が完了後に後続作業が完了可, ESS : 先行作業が完了した後指定時間経過後に後続作業を開始する, SSS : 先行作業が開始後指定時間後に後続作業を開始する, EES : 先行作業が完了後指定時間後に後続作業が完了する, ESR : 先行作業が完了後指定時間以内に後続作業を開始する, SSR : 先行作業が開始後指定時間内に後続作業を開始する, EER : 先行作業の完了後指定時間以内に後続作業を完了する, ALT : 相互に代替的作業を示す ”;
 指定時間 (前後作業の開始・完了時間に関するタイプに対応する指定時間): インスタンス属性, 非負整数 or ナル値, 時間 [U-time], パラメータ, “ ナル値はデフォルト値として指定なしを意味する, タイプが ALT の時はナル値とする ”;

図 7: 作業 - 資源型ならびに先行制約型

5.4. 時間の扱い

時間をどのように取り扱うかは、スケジューリング問題を記述する際に基本的な考慮点である。順序づけ問題では、時間は抽象的な連続数として扱えるが、APS ソフトウェアでは在庫や負荷を時点ごとに管理することが必要なので、適当な単位時間で測った各時刻をインスタンスとするオブジェクト型 (図 8) を用意する。インスタンス属性となる型属性は識別名 (スケジュール時刻) だけである。

スケジュール期間型 (スケジュール期間の各時点の集合);

識別名 (0 からスケジュール期間の終期までの各時刻): インスタンス属性, 正整数 { 0, 1, ..., スケジュールの終期 }, 識別子, “ この値はスケジュール時刻になる ”;
 要素数 (単位時間でカウントしたスケジュール期間の長さ): , 正整数, なし [なし], パラメータ, “”;
 単位時間 (スケジュール時刻の一単位が実時間の何分かを示す): , 正整数, 時間 [分], 固定値, “ U-Time はこの単位時間を一単位とする時間単位 ”;
 始期 (スケジュール期間の開始時刻): , 正整数, 時間 [年月日時分], パラメータ, “ 条件 : 始期 < 終期, あらかじめ指定された様式で記述した年月日時分 ”;
 終期 (スケジュール期間の最終時刻): , 正整数, 時間 [年月日時分], パラメータ, “ 条件 : 始期 < 終期, あらかじめ指定された様式で記述した年月日時分 ”;

図 8: スケジュール期間型の記述

5.5. GERM による問題定義の DSS への応用

問題定義に対する数理モデルが構築され、その解法が開発されれば、それら間の対応関係が明確になる。この対応関係を体系的に蓄積すれば、問題定義からソルバープログラムを起動し、問題を解くことが可能になる。

また、現実世界では全く異なると認識される問題の問題定義が同じ一つの数理モデルに対応づけられることがある。例えば、順序に依存する段取り時間の総和を最小にする処理順序を求める問題 (段取り時間問題) は、平面上に多数の穴を 1 スピンドルの穴あけ機械で加工する際の総移動距離を最小にする穴あけ順序を求める問題 (穴あけ順序問題) と、適当な条

件の下では同じ数理モデルに定式化される．また，異なる問題定義に対応する数理モデル間の数学的関係が明らかになる場合がある．それは等価関係，一般 - 特殊の関係などである．これらの事実を利用することによって，物理的には全く異なる問題やそれらの数理モデルを相互に関連づけることも原理的には可能である．

しかし，段取り時間問題に直面した意思決定者に対し，穴あけ順序問題の表現様式に従って，データを収集することは容易ではない（数理モデルの構造とは関係なく，）意思決定者の表現様式に従い，情報交換をすることで，意思決定者からの情報を正確に把握することが可能となる．

更に穴あけ順序問題との論理的関係が明らかであり，穴あけ順序問題用のソルバーが利用可能である場合，段取り時間問題として収集されたデータとの対応関係を整理することで，ソルバーを起動することが可能となる．ソルバーを対応可能にするために問題定義を再構築する必要がなくなり，結果として，前提3にあるような意思決定プロセスの定型化が促進されると期待される．

こうした面からの応用については文献（鮑 [3], Mukohara-Sekiguchi[23]）を参照されたい．

5.6. 問題定義の作成支援への応用

意思決定支援者は，数理モデルを構築するのに必要な情報を意思決定者から聞き出し，問題定義に反映させる．意思決定支援者が目的を達成した場合，問題定義に記述される型属性およびインスタンス属性には，数理モデルの具体例に対応する属性ばかりでなく，数理モデルの構造を規定する属性がすべて含まれる．また，問題定義に含まれる実体型や関連型の組合せ，型属性の組合せ，インスタンス属性となる型属性，インスタンス属性の領域等の変化に応じて，数理モデルの構造は変化する．このことから，問題定義に含まれるオブジェクト型に関する情報を体系的に蓄積すれば，問題定義を作成する際の参考情報として有効なことが分かる．

例えば，GERMによる問題定義におけるオブジェクト型やオブジェクトの記述の一部を変更すれば，異なる意思決定問題の問題定義となることは既に例示した（5.2. 参照）．また，図5の問題定義からF2//Cmaxの図4とは異なる問題定義を作るのも容易である（詳細は関口・鮑・向原 [29] を参照）．もちろん，任意の変更が現実的に意味のある問題定義を生成することにはならない．しかし，直面する論議領域に対して問題定義を作成しようとする意思決定者や意思決定支援者に，それに似た論議領域における問題定義に利用されたオブジェクト型や型属性を例示することは有効な支援となると期待される．

二つのオブジェクト型が類似であるとは，それらを定義する型属性名の集合が同じであることと定義する．二つの関連型が類似であるとは，それらを定義する型属性名の集合が同じで，定義型に含まれるオブジェクト型が一致することと定義する．これらの類似性の定義はオブジェクトの類似性（3.2. 参照）の定義と同様である．二つのシステム型（問題型）が類似であるとは，それらの構成型に含まれる各オブジェクト型が互いに類似であることと定義する．類似なシステム型（問題型）は似通った現実世界の状況を記述したものと考えられる．

互いに類似なオブジェクト型の型属性値を各型属性ごとにまとめた集合を，汎型ドメイン（generic-type domain）と呼ぶ．互いに類似なオブジェクト型の型属性名とその汎型ドメインの対の集合を汎オブジェクト型（generic object type）と呼ぶ．類似なシステム型の構成型に含まれる各オブジェクト型に対して，汎オブジェクト型を作成する．類似なシステム型の構成型に含まれる各オブジェクト型を汎オブジェクト型で置き換えた記述を汎システム

型と呼ぶ。汎オブジェクト型の各型属性名に対して汎型ドメインの要素の1つを値として割り当てると、1つのオブジェクト型が定義される。そこで、汎システム型の構成型に含まれる各汎オブジェクト型に対して同様の手続きを繰り返すことにより類似のシステム型（問題型）が定義される。従って、汎システム型は問題定義のための参考例、ないしは、問題定義の事例集として利用できる。

上田, 関口 [30] は、このような事例集の作成を目的として4つのAPSソフトウェアが解くスケジューリング問題のGERMによる問題定義から、ある汎用的な問題定義を生成した。更にAPSソフトウェアの利用経験がない工程担当者に、この汎用的問題定義に基づいてスケジューリング問題の説明をしている。

6. まとめ

数理モデルの対象となる意思決定問題の構造化された記述である問題定義の方法として、実体-関連概念をいくつかの点で拡張して利用するGERMの記述様式を提案した。GERMによる問題定義は、問題型、その構成型に含まれる各オブジェクト型、その中の各固定オブジェクト型のインスタンス集合の3種類の記述から構成される。その例として2機械フローショップ問題の問題定義を示した。また、あるAPSソフトウェアのモデル記述言語が記述するスケジューリング問題を例として用いて、GERMによる問題定義の特徴と有効性を示した。さらに、意思決定支援システムや問題定義作成支援への応用を論じた。

現実世界における意思決定問題を文書化して再利用するという本研究の方法を実際に利用するには、問題定義を作成し、かつ、作成した問題定義を数理モデル及びソルバーと関連づけるための作業が必要である。GERMの実用化には、こうした準備作業に必要な労力の軽減が必要であろう。

GERMアプローチによる問題定義を読み、その記述内容を再構成するには、実体-関連概念に基づく記述に関する知識や経験が必要である。また、問題定義に記載される豊富な情報の中で数理モデルの構造的特徴を決めるような重要部分はどこかを発見するのは簡単ではない。本論文では実体-関連ダイアグラム風の記述を問題定義の記述内容の全体像を把握する一助としたが、この点の改善も残された課題である。

謝辞

PSLが記述する内容をGERMで記述してスケジューリング問題を明示的に示す作業の一部は北海道大学大学院経済学研究科博士後期課程上田雅幸氏による。著者らの一人は本研究に際して部分的に日本学術振興会科学研究費補助金奨励研究(A) No. 13780347 および日本学術振興会平成14-15年度科学研究費補助金(基盤研究(B)(1), No. 14380185)の助成を受けた。他の一人は部分的に文部科学省科学研究費補助金基盤研究(C)(2) No. 09680412, 科学研究費補助金若手研究(B) No. 16710123の支援を受けた。

最後に、本論文を改善するにあたり貴重な示唆をいただいた査読者に記して謝意を表す。

参考文献

- [1] A. Amatiürk, E.L. Johnson, J.T. Linderoth M.W.P. Savelsbergh: A relational modeling system for linear and integer programming. *Operations Research*, **48**(2000), 846-857.
- [2] S. Banerjee and A. Basu: Model type selection in an integrated DSS environment. *Decision Support Systems*, **9-1** (1993), 75-89.

- [3] 鮑 金源: 問題定義 / モデル構築支援のための事例ベースシステムの開発. オフィス・オートメーション, **18-5** (1998), 118-126.
- [4] A. Bharadwaj, J. Choobineh, A. Lo, and B. Shetty: Model management systems: A survey. *Annals of Operations Research*, **38-1** (1992), 17-67.
- [5] H.K. Bhargava and S.O. Kimbrough: Model management: An embedded languages approach. *Decision Support Systems*, **10-3** (1993), 277-299.
- [6] R.W. Blanning: An entity-relationship approach to model management. *Decision Support Systems*, **2** (1986), 65-72.
- [7] J. Choobineh: A diagramming technique for representation of linear models. *OMEGA International Journal of Management Science*, **19-1** (1991), 43-51.
- [8] D. Edmond: *Information Modeling*. Chapter 11 (Prentice Hall of Australia Pty Ltd., 1992).
- [9] D.S. Frankel, P. Harmon, J. Mukerji, J. Odell, M. Owen, P. Rivitt, M. Rosen, R.M. Soley: The Zachman Framework and the OMG's Model Driven Architecture, A Business Process Trends White Paper, <http://www.omg.org/mda/presentations.htm> , (2003).
- [10] A.M. Geoffrion: Integrated modeling systems. Working Paper No. 343, (Western Management Science Institute, University of California , Los Angeles, 1986).
- [11] A.M. Geoffrion: An introduction to structured modeling. *Management Science*, **33- 5** (1987), 547-588.
- [12] A.M. Geoffrion: FW/SM: A prototype structured modeling environment. *Management Science*, **37-12** (1991), 1513-1538.
- [13] A.M. Geoffrion: The SML language for structured modeling: Levels 1 and 2. *Operations Research*, **40-1** (1992), 38-57.
- [14] A.M. Geoffrion: The SML language for structured modeling: Levels 3 and 4. *Operations Research*, **40-1** (1992), 58-75.
- [15] S.N. Hong, M.V. Mannino, and B. Greenberg: Measurement theoretic representation of large, diverse model bases – The unified modeling language Lu. *Decision Support Systems*, **10-3** (1993), 319-340.
- [16] S.N. Hong and M.V. Mannino: Formal semantics of the unified modeling language Lu. *Decision Support Systems*, **13-3/4** (1995), 263-294.
- [17] 法政大学情報マネジメント研究室: 生産スケジューリング問題記述言語 P S L仕様書バージョン 2.1.0. <http://www.img.k.housei.ac.jp/pslib/> (2000).
- [18] S.M. Johnson: Optimal two- and three-stage production schedules with set-up time included. *Naval Research Logistics Quarterly*, **1**(1954), 61-68.
- [19] J.F. ミュース, G.L. トンプソン (編): インダストリアルスケジューリング. (竹内書店, 1966) , 35-45.
- [20] R. Lazimy: Knowledge representation and modeling support. In March, S.T. (ed.): Knowledge-based systems. *Entity-Relationship Approach* (Elsevier Science Publishers B.V., North-Holland, 1988), 133-161.
- [21] T.P. Liang: Analogical reasoning and case-based learning in model management system. *Decision Support Systems*, **10-2** (1993), 137-160.

- [22] M. Mannino, B. Greenberg, and S.N. Hong: Model libraries: Knowledge representation and reasoning. *ORSA Journal of Computing*, **2-3** (1990), 287-301.
- [23] T. Mukohara and Y. Sekiguchi: The DSS architecture based on non-mathematical problems specification and model/solver independence. In E. Kozan and A. Ohuchi (eds.): *Operations Research/Management Science at Work: Applying Theory in the Area Pacific Region*, (Kluwer Academic Publishers, 2002), 281-298.
- [24] H. Müller-Merbach: Model design based on the systems approach. *Journal of Operational Research Society*, **34-8** (1983), 739-751.
- [25] H. Müller-Merbach: Database-oriented design of planning models. *IMA Journal of Mathematics Applied in Business & Industry*, **2** (1990), 141-155.
- [26] H. Müller-Merbach: Design of planning models supported by relational databases. *The Journal of the Operational Research Society*, **42-6** (1991), 522-523.
- [27] R.G. Ramirez, C. Ching, and R.D. St. Louis: Independence and mappings in model-based decision support systems. *Decision Support Systems*, **10-3** (1993), 341-358.
- [28] 関口恭毅: CAMP: 順序づけ分枝限定アルゴリズム設計支援システム. *オペレーションズ・リサーチ*, **37-11** (1992), 550-554.
- [29] 関口恭毅, 鮑 金源, 向原 強: GERMによるスケジューリング問題記述の特徴と応用あるAPSシステムの問題記述を例として, ディスカッションペーパーシリーズB, 北海道大学大学院経済学研究科, **39** (2002).
- [30] 上田雅幸, 関口恭毅: 問題定義を活用するAPSソフトウェア選択手順の有効性に関する研究, *生産管理学会論文誌*, **11-2** (2005), 57-66.

向原 強
北海道情報大学経営情報学部
〒060-8585 北海道江別市西野幌 59 番 2
E-mail: muko@do-johodai.ac.jp

ABSTRACT

**SCHEDULING PROBLEMS SPECIFICATION BY EXTENDING
ENTITY-RELATIONSHIP CONCEPTS:
CHARACTERISTICS AND APPLICATIONS**

Tsuyoshi Mukohara Yasuki Sekiguchi Jinyuan Bao
Hokkaido Information University Hokkaido University Sapporo Office Computer Co.,Ltd.

Problems and mathematical models are two essential elements in OR/MS. Particularly, there is some accumulation of research on mathematical model building for decision support problems. By contrast, very few researches have focused on the methodology for structured and operational description of problems. Such description is called problem specification and should be constructed prior to mathematical model building. From this point of view, the authors have proposed GERM as problem specification method. GERM is an extension of the traditional Entity-Relationship Model approach used in designing databases.

First, this paper explains the outlines of this extension. In GERM approach, a problem specification is represented as a system type, which consists of a set of object types (entity type and relationship type) and a set of objects (instances of fixed object types and those of open object types). These object types are defined as a set of user defined type attributes. Thus, GERM vocabulary is expandable and flexible. In addition, because type attributes and their values can be some natural terms in a real world of the problem being specified, they are useful for decision makers and model builders to communicate each other.

Then, a general form of problem specification by GERM is proposed and it is explained in detail by using the classical F2//Cmax type problem. A description of the problem by the classical entity-relationship approach is compared to a GERM description.

Finally, this paper examines and demonstrates the characteristics and the effectiveness of GERM. Because of its features, GERM can deal with not only simple problem like F2//Cmax, but also practical and complex scheduling problems. This is exemplified by describing job shop problems dealt with by a powerful APS. In conclusion, GERM is useful as a constituent for a new architecture of decision support systems which start supporting from problem specification process.