

問題解決オントロジーの構成 -スケジュールリングタスクオントロジーを例にして-

Construction of a Problem Solving Ontology -A Scheduling Task Ontology as an Example-

瀬田 和久*¹ 池田 満*¹ 角所 収*² 溝口 理一郎*¹
Kazuhisa Seta Mitsuru Ikeda Osamu Kakusho Riichiro Mizoguchi

* 1 大阪大学産業科学研究所
ISIR, Osaka University, Ibaraki 567, Japan

* 2 兵庫大学経済情報学部
Faculty of Economics and Information Science, Hyogo University, Kakogawa, 675-01, Japan

1997 年 7 月 14 日 受理

Keywords: problem solving knowledge, task ontology, knowledge sharing

Summary

Recently much attention has been paid to the notion of "ontology" in the expectation that it can serve as the new, strong foundation of knowledge engineering. In the conventional approach to theory of knowledge, to give the operational semantics of knowledge representation has been regarded as of major importance and the analysis of contents of knowledge has been considered to be subordinate to it. To solidify the foundation of knowledge engineering, however, the many researchers, especially in the field of knowledge sharing and reuse, has strongly felt necessity of the change of such a way of thinking. The key to the problem is to understand the essential interaction between "form" and "contents" on equal importance. This implies that deep understanding of "content" will give us new insight into design of knowledge representation. The notion of "ontology" can be key to this issue.

We have investigated the property of problem solving knowledge and tried to design its ontology, that is, task ontology. The main purpose of this paper is to discuss the basic issues concerning the task ontology. It provides three major advantages as follows. (A) It provides human-friendly primitives in terms of which users can easily describe their own problem solving process. (B) The systems with task ontology can simulate the problem solving process at an abstract level in terms of conceptual level primitives. (C) It provides ontology author with an environment for building task ontology so that he/she can build a consistent and useful ontology.

1. はじめに

知識の共有・再利用を実現するために重要な役割を担うものとしてオントロジーという概念に対する期待が国際的に高まり、既に多くの成果が報告されてつつある。しかしながら、オントロジーに関する諸問題を工学的に位置づけて解決するために必要なオントロジーの構築・評価に関する方法論については、それを模索

している段階にある。

オントロジーは、対象世界を構成する概念の意味や、それらの概念の間に成り立つ関係を何らかの形式で体系化し、対象をモデル化する際の規約を与えるものである。さらに、モデル化にあたって簡約化した部分や、利用目的に固有の視点といった前提条件もオントロジーを通じて明らかにされる [溝口 97]。このような、モデル化の基礎としてのオントロジーが適切に構成されれば、モデルを利用するエージェントの間で対象を捉える

概念構成に関する明示的な合意のもとでのモデルの共有が可能になる [堀 94, 溝口 94, 元田 94, Skuce 95]。

オントロジーには主に、利用の範囲を限定しないオントロジーと、利用の範囲を特定の問題解決に限定したオントロジーがある [溝口 97, Heust 97]。本稿では後者のオントロジー、すなわち問題解決の観点から対象世界の概念を組織化するタスクオントロジー (問題解決オントロジー) [Hori 94, Ikeda 97, Mizoguchi 95, Steels 90, Wielinga 92] を対象としている。

知識の共有/再利用の観点からのタスクオントロジーに関する我々の立場については、既に文献 [ティヘリノ 93] において、基本的アイデアについて述べている。そこでは、タスクオントロジーの枠組みを示しその有効性を示しているが、以下の様なオントロジーに関する基礎的課題を残していた。つまり、前提とするエージェントとモデルは何であり、その前提の元で、

- (1) エージェントが共有すべき規約としてのオントロジーをどのように構成すべきか。
- (2) 問題解決過程の記述/実行に必要な概念は何か。
- (3) 問題解決過程を記述したモデルから対応する実行モデルを導く際に考慮すべき概念は何か。

ということがタスクオントロジーとして明らかになっていなかった。

本研究の最終的な目的は、知識ベースシステムの構築/保守の基礎となるタスクオントロジーの構築から利用にいたる方法論を確立することである。本稿では、その第一段階としてタスクオントロジーとそれに関わるエージェントやモデルの間の関係について考察し、我々が設定したタスクオントロジーの基本構成を明らかにする。そして、その基本構成に基づいて構築したタスクオントロジーとタスクオントロジーの構築・利用環境である概念レベルプログラミング環境 CLEPE (Conceptual Level Programming Environment) の基本的枠組みについて述べる。タスクオントロジーの一利用形態としての問題解決知識記述環境の詳細については別項 [瀬田 97] で述べる。

2. 基礎的考察

本章では、タスクオントロジーの構成を考える上で、基礎となる事項を整理する。

2.1 オントロジーの分類 [Heust 97, 溝口 97]

オントロジーは本来「世界を構成する物や概念の間の関係を体系化する学問」という哲学の分野の存在論

(オントロジー) を表す言葉である。知識ベースシステムの分野では、「知識ベースを構成する基礎概念/部品についての体系的記述」という意味で用いられる。ここでは、知識ベースを構成する基礎概念 (内容) を整理する試みである Content オントロジーに関する研究を分類する。

Content オントロジーは大きく、(I) 実体やプロセス、時間などのいわゆる「常識」と呼ばれる一般的な概念を定義するオントロジー [Lenat 90, Sowa 95]、(II) 領域に関する概念を明らかにするオントロジー [Gruber 94, Gruninger 95, 来村 97, 笹島 96, 高岡 95]、(III) 問題解決が行われる環境が問題解決に与える影響を規定するオントロジー [Vanwelkenhuysen 95]、(IV) 問題解決の構造をドメイン独立に記述するオントロジーがある。主に (I)(II) のオントロジーが利用方法を限定しないオントロジーでありそれぞれ、(I) 一般/共通オントロジー、(II) ドメインオントロジーと呼ばれる。(III)(IV) のオントロジーは利用の範囲を特定の問題解決に限定するオントロジーであり、それぞれ (III) Workplace オントロジー、(IV) タスクオントロジー (問題解決オントロジー) と呼ばれる。既に述べた様に、本研究ではこのうちタスクオントロジーを対象として議論を行う。

本稿では、タスクオントロジーを「人間が行っている日常的な問題解決に関する操作や操作対象の物の間の関係の体系的記述」という意味で用いる。直感的に、欧州の知識コミュニティで使われている (タスク) モデルに、米国で使われているメソッドとタスクを総称したものに、本稿のタスクオントロジーが対応している。

2.2 オントロジーとモデルの関係

本研究では、オントロジーが定める規約に従ってオーサが問題解決知識を記述した場合に、その知識をモデルと呼び、モデルが「オントロジーに従っている」あるいは「整合している」といった表現を用いることにする。

タスクオントロジーの研究の目的を端的に言えば、問題解決知識のオーサと、その知識を利用する計算機システムの間コミュニケーションをより円滑にすることにある。タスクオントロジーはそのための基礎として「タスクオントロジーに整合したモデル」を規定している。

オブジェクト指向言語とのアナロジーで言えば、モデルオーサ (プログラマ) に対して予め用意された組み込みのクラス定義がオントロジーに対応し、そのクラスを利用して作るインスタンスがモデルに対応する。

厳密には、モデルとオントロジーの関係は絶対的で

表 1 CLEPE におけるエージェントとモデルの関係

	エージェント	オントロジー	モデル	目的
問題解決知識記述環境	エンドユーザ (モデルオーサ) 計算機	タスクオントロジー (クラス)	問題解決知識	オントロジーに基づいて 問題解決知識を記述する エンドユーザが記述した問題解決知識を オントロジーに基づいて実行する
オントロジー構築支援環境	オントロジーオーサ 計算機	コアタスクオントロジー (メタクラス)	タスクオントロジー (クラス)	問題解決知識に対する規約となるタスクオントロジーを オントロジー(メタクラス)に基づいて構築する タスクオントロジーの整合性をオントロジー (メタクラス)に基づいて検証する

はなく、オーサがモデル化する対象に依存する。例えば、クラスを定義するオーサにとっては、そのクラス定義自体がモデルに対応し、クラス定義に際して参照するメタクラスがオントロジーに対応する。既に述べたように、オントロジーをどのようなレベルで設定し、何を定義すべきかという事について考えるときには、モデルオーサの視点とオントロジーの利用目的を注意深く検討しなければならない。すなわち、(モデルオーサを含む) どのようなエージェントが存在し、それぞれのエージェントはオントロジーを使ってどのようなモデルを操作し、どういった目的を達成したいのか？その目的を達成するための規約として何が必要か？ということが重要となる。

このことを我々が開発しているオントロジーワークベンチ CLEPE(Conceptual Level Programming Environment)[Ikeda 97, 瀬田 97, 瀬田 95] の構成(図 1) と対応づけて説明すると以下になる。CLEPE はオントロジーの構築から利用にいたる過程を支援することを目的としており、大きく分けて

- (1) 図 1 の右側面の問題解決知識記述環境と
- (2) 図 1 の上面のオントロジー構築支援環境

の二つの側面を持っている。

- (1) 問題解決知識記述環境では、モデルオーサ(エン

ドユーザ) とシステム(モデル処理系) がエージェントとして存在する。計算機に馴染みのないエンドユーザは、(タスクオントロジーとして定義された) 日頃使っている平易な言葉を使って問題解決知識(以降、問題解決モデルと呼ぶ) を記述して、実行内容を確認する。また、システムは、エンドユーザが記述した問題解決モデルをオントロジーを参照して実行する(表 1)。

(2) オントロジー構築支援環境では、オントロジーオーサ(オントロジーを構築するオーサ) とシステム(言語処理系) がエージェントとして存在する。オントロジーオーサは、エンドユーザが作成する問題解決モデルに対して規約としてはたらくタスクオントロジーを構成することが目的である。また、システムはオントロジーオーサの作業をガイドし、構築されたオントロジーに矛盾がないかどうか(整合性) を検証する。二つのエージェントが協調して整合したオントロジーを構築するために、オントロジーオーサとシステムが参照するタスクオントロジー(メタクラス) が必要である。本研究ではこれをコアタスクオントロジーと呼んでいる。(表 1)。

2・3 タスクオントロジーの基本構成

本稿の主題であるタスクオントロジーの構成を考えるにあたっては、以下に示す要求を満足するようなタスクオントロジーの構成を明らかにする必要がある。

- I. 問題解決モデルを容易に記述できること。
- II. 問題解決モデルの計算論的な意味が明確になること。
- III. スケジューリングや診断などのタスクタイプ固有性に対応できること。
- IV. 問題解決全般に広く適応できること。

図 2 に (I) ~ (IV) の要求を満足するタスクオントロジーの基本構成を示している。図には、(I)(II) の要求を満足するために、奥行き方向(a 軸) にレキシカルレベルオントロジー(TO/K-L) と概念レベルオントロジー(TO/K-C) を配置し、(III)(IV) の要求を満足するために、垂直方向(b 軸) にタスクタイプ固有(Task-S) オントロジーとコアタスク(C-Task) オントロジーを

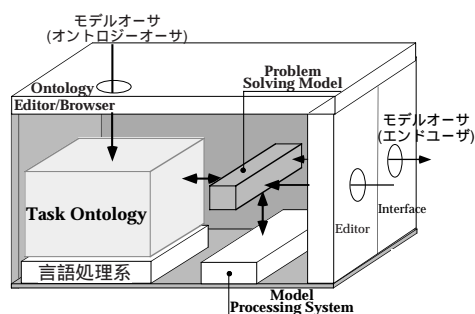


図 1 CLEPE の全体像

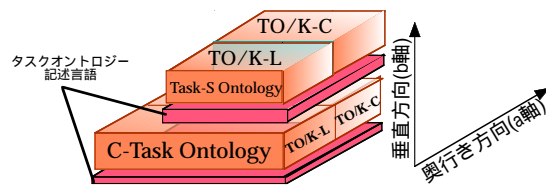


図2 タスクオントロジーの基本構成

配置している。

a 軸のレキシカルレベルオントロジーは (I) の要求を満足するために、問題解決モデルを記述する際に使う単語の品詞的な性質を定めており、概念レベルオントロジーは (II) の要求を満足するために、それらの単語の意味を定めている。また、b 軸のタスクタイプ固有 (Task-S) オントロジーは (III) の要求を満足するために、スケジューリングや診断などのタスクタイプ固有の概念を規定し、コアタスク (C-Task) オントロジーは問題解決に一般の概念を捉えている。

本稿では、3 章、4 章においてタスクオントロジーの奥行き方向 (a 軸) の構成について、主にタスクタイプ固有オントロジーを例にして述べる。垂直方向の構成と言語処理系の詳細については、5 章においてオントロジー構築支援環境を例にして述べる。

3. 問題解決モデルとタスクオントロジー

3・1 問題解決モデルの具体例

2.3 で述べた要求 I,II の間には、人間にとっての記述の容易性を尊重しようとする、その記述の意味が曖昧になり、計算機的なオペレーショナリティが明確にならないというジレンマがある。

本研究では、この要求を両立するために、問題解決モデルを、自然言語の記述に近いレキシカルレベルモデルと、そのモデルの意味内容を表す概念レベルモデルに分けている。通常のプログラミング環境との直感的なアナロジーをとれば、レキシカルレベルモデルがソースに相当し、概念レベルモデルがそのターゲット (内部モデル) に相当する。

問題解決モデルの例として、図3に看護婦の負荷が均一になるようにジョブを割り当てる24時間要員配置問題に対する問題解決モデルを示している。図の左側がレキシカルレベルモデルであり、右側が対応する概念レベルモデル^{*1}を表している。

*1 内部表現を模式的に図示している。

モデルの作成にあたって、エンドユーザは処理概念を名詞と動詞の組み合わせ (以降、汎化プロセスと呼ぶ) を基本として表現し、処理順序を有向リンクで結ぶことによって問題解決のフロー (レキシカルレベルモデル) を作成する。

図3に示したレキシカルレベルモデルが表している処理の大まかな流れは、まず、(日勤、夜勤などの) 勤務単位グループ毎に看護婦に対して勤務を割り当て (図の上部の二重ループの内側のループ)、全ての勤務単位グループについて同じ処理を繰り返す (図の上部の二重ループの外側のループ)。そして、看護婦の負荷が均一になるように割り当てを交換する最適化フェーズ (図の下部のループ) からなっている。

対応する概念レベルモデルでは、汎化プロセスに対応するアクティビティに加えて、オブジェクトフローが補われている。オブジェクトフローによってアクティビティの実行を通じたオブジェクトの変化の過程が表現される。例えば、図の assign アクティビティの働きについて、「Assign アクティビティはループの周回の度に、取り出したジョブ (Picked-up Job) と、そのジョブに対して選択した看護婦 (Selected Nurse) を割り付け (assign)、一つの割り付け (assignment) を生成し、解 (assignment-set) を更新する。」と言ったことが内部表現として表されている。

3・2 問題解決モデルに対する規約としてのタスクオントロジー

図3に示された二つのモデルの比較において注目すべき点は、レキシカルレベルモデルにおいて明らかにされていないオブジェクトの同一性が概念レベルモデルにおいて明示的に表されている事である。

例えば、Pickup プロセスにおいて「取り出されたジョブ」と「Assign プロセスの入力となるジョブ」は、レキシカルレベルモデルにおいては単語としてエンティティが異なっているが、概念レベルモデルとしては同一のオブジェクトであり、同一の「ジョブ」として扱うようになっている。また、図の上部の二重ループの内側のループの出力である「部分解」と外側のループの出力である「暫定解」、そして、図の問題解決モデル全体の出力である「最適解」は、同一のオブジェクトであり、問題解決の過程を通じてその状態が変化していくものと考えられる。このように、レキシカルレベルモデルに対応する概念レベルモデルでは、実行モデルのセマンティクスを定義するために不可欠なオブジェクトフローが明示的に表現されている。

この二つの知識レベルのモデル (レキシカルレベルモ

図 3 レキシカルレベルモデルと概念レベルモデル
(最適化フェーズを省略)

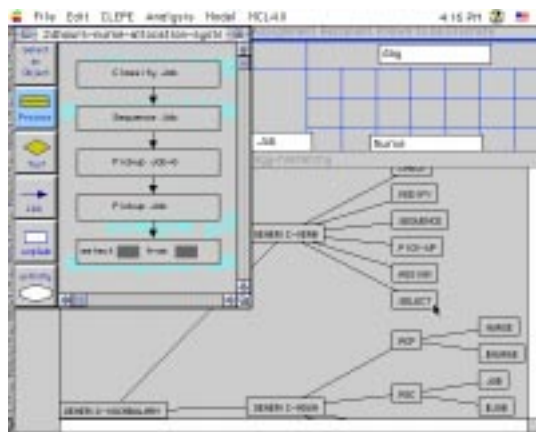


図 4 CLEPE のインタフェース

デルと概念レベルモデル) を規定するオントロジーを知識レベルオントロジー (TO/K) とよんでおり、各々のモデルを規定するオントロジーがレキシカルレベルオントロジー (TO/K-L) と概念レベルオントロジー (TO/K-C) である。

レキシカルレベルオントロジーと概念レベルオントロジーの特性は語彙の階層構造を考えるとわかりやすい。図 4 と図 5 にレキシカルレベルオントロジーと概念

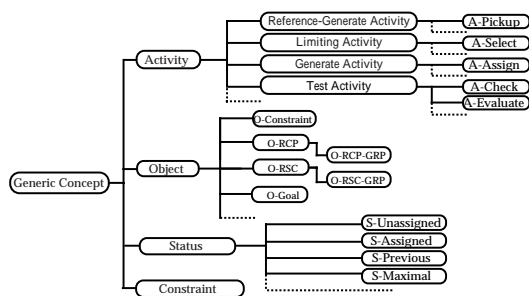


図 5 概念レベルオントロジー

レベルオントロジーの階層図を表している。TO/K-L はモデルオーサ (エンドユーザ) が直接利用するものであり、図 4(下部) の画面に見られるように、単語の品詞を最上位においたオントロジーである。単語の統語的な役割を定義する最下位の階層にはモデル構築者が日頃使っている対象の世界の単語が配置されており、全体として単語の問題解決の観点から見た意味的階層が表現される。一方、TO/K-C は統語的な制約から離れ、レキシカルレベルモデルの意味内容を表現するための概念の構造を組織化している。

図 6 動詞「Select」の定義

という統語形態をとり得ることを定義している。ここで () 内の文構成要素はいずれかが省略可能であることを表している。フィールド N1,N2 には汎化動詞の対応アクティビティの作用対象を表す汎化名詞が入りうることを表している。さらに N1 の実体は出力, N2 の実体は入力オブジェクトであることを規定している。入力オブジェクト, 出力オブジェクトに対する意味制約は TO/K-C (汎化動詞 select に対しては対応するアクティビティ (cor-activity) A-select (後述)) を通して規定する。また, あるインスタンスが動詞「select」のインスタンスであるかどうかを判定する所属性定義 (membership) では, それを外延的 (extensional) に判定することを定義している。

エンドユーザによる問題解決モデルの記述は動詞が定めるこのような統語則に従ってなされる。例えば図 4 の画面で、エンドユーザが TO/K-L の語の階層からマウス操作により動詞 select をクリックした際には、選択可能な統語則がブラウザ上で表示される。この中からいずれかを選択することによって、統語則に適合したフィールドの列が汎化プロセス内に表示される。そしてユーザによってフィールド内に埋め込まれた単語がオントロジー (統語論的公理) に対して整合するかどうか確認された後で、TO/K-C の概念レベルモデルと対応づけられる。

4・2 概念レベルオントロジー:TO/K-C

-概念レベル公理-

TO/K-C では問題解決の過程を概念レベルで捉えるために必要な公理が定義される。TO/K-C に定義される概念は、オブジェクト概念、叙述概念、アクティビティ概念に分類される。直感的にはそれぞれ、汎化名詞、汎化形容詞、汎化動詞の意味内容に相当する。

オブジェクト概念定義は、汎化名詞の対応概念として、それがタスクにおいてどういう性質をもった「も

図 10 名詞「暫定解」の定義

るクラス制約や、オブジェクトの生滅に関わる作用を定義する。例えば、入力オブジェクトの集合から一つのオブジェクトを取り出すアクティビティ概念「A-Select」の定義では、入力オブジェクトが集合であり、出力オブジェクトは個体でなくてはならないことを定義している。また、オブジェクトの構成や生滅に関わる作用として、「A-Select」は入力オブジェクトからオブジェクトを限定する作用を持つことが定義されている。

4・3 TO/K の公理：語用論的公理

語用論的公理は 3.2 で述べたような問題解決コンテキストに依存したオブジェクトの同一性について、レキシカルレベルモデルと概念レベルモデルの間の整合した対応関係を定める。

例として、図 10 に汎化名詞「暫定解」の定義を示している。汎化名詞のクラス定義には、クラス階層 (:class-hierarchy) と対応オブジェクトに対する制約を記述する。対応オブジェクトに対する制約としてここでは、汎化名詞「暫定解」(図 10) では、対応オブジェクトとして「解 (O-assignment-set)」を持ち、そのうち状態制約として「暫定 (S-temporary)」状態にあるものと対応することを定義している。

TO/K-L と TO/K-C の対応関係を定めるこのような TO/K の公理 (図 10 の例では、汎化名詞「暫定解」と暫定状態にある「解オブジェクト」の対応関係を定める cor-object スロットが TO/K の公理に対応する)、及び、オブジェクトがとり得る状態の順序に関する整合した対応関係を定める公理に基づいて、(1) オブジェクトとその状態の組み合わせに関する整合性を検証することができる。また、(2) ユーザが書いた問題解決

ラスが持つべき必須のスロットやスロットの役割などのメタな構成を規定する。タスクタイプ固有オントロジーではそのインスタンスとして、Select などの概念が定義される。本研究では、この作業を行うコアタスクオントロジーオーサを、タスクオントロジーの構築に際して工学的に最も重要な役割を担うオーサとして、オントロジー工学者を想定している。

図 11 に言語処理系を組み込んだ CLEPE の全体像を示している。言語処理系は、モデルのオントロジーに対する整合性を検証する役割を持っている。タスクオントロジーはその役割に応じて複数のレイヤに分けられている。これらの間の関係は、下位から上位と進むにつれて、相対的に下位のオントロジーで定義される語彙が、オントロジー記述言語を段階的に拡張し、相対的に上位のオントロジーの記述を支える形態になっている [瀬田 96]。例えば、コアタスクオントロジーはタスクタイプ固有オントロジーの意味を支える役割を担っている。

二つのタイプのオントロジーオーサと言語処理系の関係は次のようになっている。コアタスクオントロジーオーサは TOL/0 を用いてコアタスクオントロジーを定義する。言語処理系は、コアタスクオントロジーに基づいて言語を拡張し、タスクタイプ固有オントロジーを定義するための言語 TOL/S をタスクタイプ固有オントロジーオーサに対して提供する。タスクタイプ固有オントロジーオーサは、TOL/S が提供する概念プリミティブを用いてモデルとしてのタスクタイプ固有オントロジーを定義する。

以下では、言語処理系がオントロジーとモデルの階層関係を捉えることによってオントロジーオーサに対してどのような環境を提供することが出来るかという事について、CLEPE におけるオントロジー構築支援環境を例にしてタスクタイプ固有オントロジーオーサの視点から述べる。

5・2 オントロジー構築支援機能

オントロジー構築支援環境 (図 12) では、定義対象とする、TO/K-L と TO/K-C をオントロジーオーサが同時に定義できるような環境を提供している。

大まかに図の左側が TO/K-L、右側が TO/K-C を定義するための画面に対応し、上部に既に定義された概念の階層が表示されている。図の中央に位置する、「verb」、「noun」、「activity」、「object」などのボタンはコアタスクオントロジーで定義された概念に対応している。このボタンのいずれかをマウスでクリックするかもしくは、上部のブラウザ上でクラスを表す新し

図 11 言語処理系を組み込んだ CLEPE の全体像

図 12 オントロジーエディタの画面ダンプ

いノードを作り，既存のノードと結ぶことで新しいクラスを定義する。その際，下部の画面に概念を定義するための概念プリミティブと各概念定義に応じたスロット，およびスロットの内容を定義するためのフィールドがルートクラス（後述）で定義された定義内容と共に表示される。

タスクタイプ固有オントロジーオーサが定義対象のクラスとして Activity を選択した際には，アクティビティを定義するための概念プリミティブ (define-task-s-activity) とアクティビティを定義するために必要なスロット，およびアクティビティのルートクラスの定義内容（後述する #Activity，図 13 のベースパートの内容）がフィールド内に示される。タスクタイプ固有オントロジーオーサはこれを参照しながらタスクタイプ固有のアクティビティの意味内容を定義し，プルダウンメニューから「check」を選択することにより，定義した内容のコアタスクオントロジーが定める TO/K-L, TO/K-C, TO/K の三つの公理に対する整合性の検証を行う。例えば，タスクタイプ固有オントロジーオーサがタスクタイプに固有の動詞 (e.g. Select) を定義する

際に，その意味内容を表す概念として，アクティビティ以外の概念 (e.g. オブジェクト概念 O-Assignment-set) を指定した際には，コアタスクオントロジーとして定義されている「動詞 (TO/K-L) の意味内容是对應するアクティビティ概念 (TO/K-C) によって定義されなければならない」という TO/K の公理に違反しているといったことが，タスクタイプ固有オントロジーオーサに対して示される。図には「A-Select」の定義内容についてのコアタスクオントロジーに対する整合性を検証した際の様子が示されており，タスクタイプ固有オントロジーオーサが定義した入力オブジェクトスロットのクラス制約がコアタスクオントロジー (TO/K-C) が定める規約「入力オブジェクトスロットに関するクラス制約はメタクラスで定義される #object のインスタンス (タスクタイプ固有オントロジーにおけるオブジェクトクラス) でなければならない。」ということに違反しているというメッセージが示されている。タスクタイプ固有オントロジーオーサはこのメッセージをもとに定義内容を修正し，コアタスクオントロジーが定める概念化の規約に整合したタスクタイプ固有オン

図 13 コアタスクオントロジー「#Activity」の定義

うになっている。

本研究で構成したタスクオントロジーの新規性は、問題解決モデルの記述/モデルの整合性や振る舞いをタスクオントロジーに基づいて概念レベル(計算機固有の概念を隠蔽した形)で検証したり実行したりすることができることにある。この能力は、特に計算機に不慣れたエンドユーザが自らモデル化を行う際に有用である [Seta 97]。

Protege-II や KSM などの他の問題解決メソッドに関する研究(以下 PSSM 研究)では、問題解決モデルと実行コードの概念的結びつきが弱く、問題解決モデルの妥当性の検証は、実行コードの振る舞いと問題解決モデルを対応づける開発者の洞察力に委ねられている。

Common KADS は問題解決システムを構成するための一般的な方法論を提供しており、実際にこの方法論に基づいた数多くのシステムが構築されている。これは、KADS が提供する方法論、つまり、知識ベースシステムを構成するために、どのような概念をどういう順序で捉えたらよいかという事のガイドラインが、知識ベース構築者にとって受け入れられやすい形で明確になっていることから生まれる効果である。本研究では、この点に関して現段階で明確な指針を持っていない。今後、本論文で述べた、タスクオントロジーの構成を元にして知識ベース開発のためのガイドラインを与えるような方法論の確立を目指していく予定である。

一方で、元田 [元田 94] が指摘しているように、KADS 方法論では、(a) モデル化に際して提供されるプリミティブが一般的でありすぎるために、モデルの表現力が不十分、(b) モデル化の過程を支援する枠組みがない、(c) 概念モデルの整合性を検証する枠組みがなく、概念モデルを実行することができない、などの

問題点が残されている。これに対して本研究では (a) の問題を扱うための枠組みとして、問題解決全般を扱えるような枠組みを保ちながら、問題解決タイプに固有の概念を捉えることができるようなタスクオントロジーの枠組みを示した。(b) に関しては、本稿で明らかにしたタスクオントロジーの三つの公理に基づいて、問題解決過程の記述/実行の各段階において、タスクオントロジーが与える問題解決タイプ固有の概念化の規約に基づいてモデル化の支援を行うことができる。(c) については、上述のPSM研究との比較で述べた様に、概念レベルモデルの整合性を検証したり実行したりすることが可能になっている。

もちろん、KADS 方法論が提供する知識をモデル化するための汎用的な方法論を提供を目指すアプローチと、一般的な枠組みの実現を視野に入れながら対象の世界の固有の原理/原則を明らかにし、それを積み上げていく我々のアプローチが将来的には一つの方法論として体系化されることが理想である。

問題解決のオントロジーを対象とした研究として、本研究と最も関連が強いものは、Mark S. Fox らが進めている TOVE(TORonto Virtual Enterprise) オントロジー [Fox 93, Gruninger 95] である。企業内のアクティビティを表現する語彙を集積し、その意味を様相論理を用いて公理化する枠組みを考えている。TOVE では、企業内の様々なアクティビティに対して抽象的な見通しをたてることをオントロジーの能力に求めている。これは、本研究が TO/K-C に求めている概念レベルでの実行能力に相当している。オントロジーに対する考え方は、TOVE と本研究で概ね一致しているが、問題解決モデルを記述する枠組みとして見たときの本研究の特徴は知識レベルを「レキシカルレベル」と「概念レベル」に分離し、レキシカルレベルオントロジーにエンドユーザとコンピュータの概念ギャップを埋めるうえで仲介的な役割を設定したところにある。他の関連プロジェクトでは、概念レベルオントロジーに相当する問題解決コンポーネントを直接的、あるいは知識工学者の仲介を前提として間接的にエンドユーザに開放しており、知識レベルを提供しているとはいいたいがたいものが多い。また、オントロジーとモデルの階層関係を明確にし、オントロジーに対するモデルの整合性を検証する枠組みを備えている点も本研究の特徴である。

7. む す び

本論文では、タスクオントロジーの構築に関する設

計指針を提案し、問題解決に必要な概念を定式化することを目的として、タスクオントロジーの構成に関する包括的な議論を進めてきた。そして、タスクオントロジーが定める三つの公理によって (A) エンドユーザにとってのモデルの記述の容易性と (B) 計算機にとってのモデルのオペレーショナリティを両立することが可能になることを示した。このような特長を備えたタスクオントロジーを利用することによって、エンドユーザに対してヒューマンフレンドリな問題解決知識記述環境を実現することが可能になる。タスクオントロジーの利用については、別稿 [瀬田 97] で報告している。

問題解決エンジンの構成に対するタスクオントロジーの有効性については、本研究の前身である Multis プロジェクトを通じて、スケジューリング問題解決タイプにおける 13 の実例 (生産計画 4, 工程計画 2, 配置 1, 配車/配送 3, 人員割り付け 2, その他 1) について有効性を検証している [ティヘリノ 93]。

また、オーサの視点に応じたオントロジーとモデルの階層性を、タスクオントロジーとして明確にすることの有効性について、オントロジー構築支援環境を例にして述べた。問題解決知識記述環境の側面からのコアタスクオントロジーの有効性については今後の課題としたい。コアタスクオントロジーのもとで用意された様々な対象タスクに固有のタスクタイプ固有オントロジーから、エンドユーザが対象とするタスクに対応するタスクタイプ固有オントロジーを選択する (ontology swapping [Seta 97]) ことによって、様々なタスクタイプを扱うことが出来る柔軟なシステムを構築できると考えられる。今後、この枠組みの有効性を様々なタスクで検証していく予定である [Ikeda 97]。

参 考 文 献

- [Fox 93] Fox, M.S., Chionglo, J., Fadel, F.: A Common-Sense Model of the Enterprise, Proc. of the Industrial Engineering Research Conference (1993)
- [Gruber 94] Gruber, T.: An ontology for engineering mathematics, Proc. of Comparison of implemented ontology, ECAI'94 Workshop, W13, pp.93-104, 1994.
- [Gruninger 95] Gruninger, M. and Mark Fox, M.S.: Methodology for the Design and Evaluation of Ontologies, Proc. of IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing (1995).
- [Heust 97] Heust, G.V., Schreiber, A.T., Wielinga, B.J.: Using explicit ontologies in KBS development, Journal of Human-Computer Studies, Vol.45, pp.183-292, (1997).
- [堀 94] 堀雅洋: 知識ベース共有へのアプローチ, 人工知能学会誌, Vol.9, No.1, pp. 17-22 (1994).
- [Hori 94] Hori, M., and Nakamura, Y.: Reformulation of problem-solving knowledge via a task-generic level, Proc. of Third Japanese Knowledge Acquisition Work-

- shop for Knowledge-Based Systems: JKAW'94, pp.3-15, (1994).
- [Ikeda 97] Ikeda, M., Seta, K., Mizoguchi, R.: Task Ontology Makes It Easier To Use Authoring Tools, Proc of IJCAI-97, pp.342-347, Nagoya, Japan (1997).
- [来村 97] 来村徳信, 笹島宗彦, 池田満, 他: モデルに基づく問題解決のための流体と時間のオントロジーの構築とその評価, 人工知能学会誌, Vol.12, No.1, pp.132-143, (1997).
- [Lenat 90] Lenat, D. and Guha, R.V.: Building Large Knowledge-Based Systems, Addison-Wesley, Reading, MA (1990).
- [溝口 94] 溝口理一郎: 知識の共有と再利用の現状と動向, 人工知能学会誌, Vol.9, No.1, pp.3-9 (1994).
- [Mizoguchi 95] Mizoguchi, R., Ikeda, M., Seta, K., Vanwelkenhuysen, J.: Ontology for Modeling the World from Problem Solving Perspectives, Proc. of IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, (1995).
- [溝口 97] 溝口理一郎, 池田満: オントロジー工学序説, 人工知能学会誌, Vol.12, No.4, pp.559-569 (1997).
- [Molina 96] Molina, M., Shahar Y., Cuenca, J., Musen, M.A.: Structure of Problem-Solving Methods for Real-time Decision Support Solving Methods: Modeling Approaches Using Protege-II and KSM, Proc. of KAW-96, (1996).
- [元田 94] 元田浩: 知識ベースの再利用へのアプローチ, 人工知能学会誌, Vol.9, No.1, pp.10-16 (1994).
- [笹島 96] 笹島宗彦, 来村徳信, 池田満, 溝口理一郎: 振る舞いと機能のオントロジーに基づく機能モデル表現言語 FBRL の開発, 人工知能学会誌, Vol.11, No.3, pp.420-431, (1996).
- [瀬田 95] 瀬田和久, 池田満, 角所収, 溝口理一郎: タスクオントロジーに基づく概念レベルプログラミング環境の設計, SIG-KBS-9503-9, pp.51-58, (1995).
- [瀬田 96] 瀬田和久, 池田満, 角所収, 溝口理一郎: 概念レベルプログラミング環境のためのオントロジー記述言語の開発, 人工知能学会全国大会(第十回)論文集, pp.207-210, (1996).
- [瀬田 97] 瀬田和久, 島輝行, 池田満, 角所収, 溝口理一郎: 問題解決オントロジーに基づく概念レベルプログラミング環境 CLEPE, 信学論(D-II), (投稿中).
- [Seta 97] Seta, K., Ikeda, M., Kakusho, O., Mizoguchi, R.: Capturing a Conceptual Model of Problem Solving for End-user Programming: Task Ontology as a Static User Model, Proc. of the Sixth International Conference on User Modeling, Sardinia, Italy, pp.203-214 (1997).
- [Skuce 95] Skuce, D. ed.: Proc. of IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing (1995).
- [Sowa 95] Sowa, J.: Distinctions, combinations, and constraints: Proc. of IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, (1995).
- [Steels 90] Steels, L.: Components of Expertise, AI Magazine, Vol.11, No.2, pp. 30-49 (1990).
- [高岡 95] 高岡良行, 広部健治, 溝口理一郎: 再利用可能知識ベースの構築—変電所事故復旧問題を例にして—, 人工知能学会誌, Vol.10, No.5, pp.786-797, 1995.
- [ティヘリノ 93] ティヘリノ・ジュリ・A, 池田満, 北橋忠宏, 溝口理一郎: タスクオントロジーと知識再利用に基づくエキスパートシステム構築方法論—タスク解析インタビューシステム MULTIS の基本思想, 人工知能学会誌, Vol.8, No.4, pp.476-487(1993).
- [Vanwelkenhuysen 95] Vanwelkenhuysen, J. and R. Mizoguchi: Workplace-Adapted Behaviors: Lessons Learned for Knowledge Reuse, Proc. of KB&KS '95, pp.270-280, 1995.
- [Wielinga 92] Wielinga, B., Schreiber, A. T., and Breuker, J. A.: KADS: A modeling approach to knowledge engineering, Knowledge acquisition, 4(1), pp.5-

53(1992)

〔担当編集委員: ×, 査読者: ××〕

——著者紹介——

瀬田 和久(学生会員)

1993年龍谷大学理工学部電子情報学科卒業, 1995年同大学院修士課程修了, 1998年, 大阪大学大学院工学研究科博士課程修了, 同年, 大阪大学産業科学研究科助手, 現在に至る。博士(工学)。オントロジー工学に興味を持つ。電子情報通信学会, 情報処理学会会員 seta@ei.sanken.osaka-u.ac.jp

池田 満(正会員)

1984年宇都宮大学工学部情報工学科卒業, 1986年同大学院修士課程修了, 1989年大阪大学大学院博士課程修了, 同年, 宇都宮大学助手, 1991年大阪大学産業科学研究科助手, 1997年同助教授, 現在に至る, 工学博士, 形式言語の構文解析, 仮説推論, 帰納的推論, 知的教育システム, オントロジー工学の研究に従事, 1996年人工知能学会創立10周年優秀論文賞受賞, 電子情報通信学会, 情報処理学会, 教育システム情報学会会員 ikeda@ei.sanken.osaka-u.ac.jp

角所 収(正会員)

1950年大阪大学工学部通信工学科卒業, 1957年同大学産業科学研究科勤務, 1989年同大学名誉教授, 龍谷大学・理工学部教授, 1995年兵庫大学・経済情報学部教授, 現在に至る。工学博士。超音波, 電子応用計測, 医用電子装置, 音声パターン認識, 心理音響, ネットワーク理論, 信号処理および知的情報システムの研究に従事。1985年度 Pattern Recognition Society 論文賞, 1987年度電子情報通信学会論文賞受賞, 電子情報通信学会, 情報処理学会, 日本音響学会, 日本認知科学会, IEEE 各会員 kakusho@humans-kc.hyogo-dai.ac.jp

溝口 理一郎(正会員)

1977年大阪大学大学院基礎工学研究科博士課程修了, 1978年同大学産業科学研究科助手, 1987年同研究所助教授, 1990年同教授, 現在に至る。工学博士。パターン認識関数の学習, 音声対話処理, 知識工学, 知的教育システム, オントロジー工学の研究に従事, 1985年 Pattern Recognition Society 論文賞, 1987年度電子情報通信学会論文賞受賞, 1996年人工知能学会創立10周年記念論文賞受賞, IEEE Expert, J. of AI in Education, J. of AI in Engineering 編集委員, 教育システム情報学会誌編集委員長, 同理事, 人工知能学会理事, 電子情報通信学会, 情報処理学会, 日本認知科学会, 日本音響学会, IEEE, AAAI, 各会員 miz@ei.sanken.osaka-u.ac.jp