

タスクオントロジーに基づくエンドユーザプログラミング支援機能

2AE-3

-問題解決過程における因果関係の明示化-

瀬田 和久¹ 池田 満¹ 角所 収² 溝口 理一郎¹

¹大阪大学産業科学研究所 ²兵庫大学経済情報学部

1. はじめに

計算機システムの普及に伴い、計算機に馴染みのないエンドユーザが計算機を操作する機会が増加しており、エンドユーザが容易にプログラミングできるように環境、エンドユーザプログラミング環境が求められてきている。

エンドユーザプログラミング環境を実現するためには、

- (a) エンドユーザが頭の中に持つ知識を容易にかつ円滑に計算機上に表現するための枠組みを備えること、
- (b) エンドユーザが記述したモデルから、エンドユーザの頭の中にある問題解決の思考過程を表現するモデル(概念レベル実行モデル)を捉え、そのモデルに忠実に問題解決を実行すると共に、記述された問題解決知識の修正を支援するための枠組みを備えること。

が重要である。

タスクオントロジーはエンドユーザが認識する対象領域の概念の構造を体系的に定義した物である。タスクオントロジーを基盤にすることで、スケジューリングや給与計算などのエンドユーザが日頃行っているような問題解決において、(a)(b)の能力を

備えた枠組みを実現することができる。

本稿では我々が開発した概念レベルプログラミング環境 CLEPE(Conceptual LLevel Programming Environment)[1]を例にして、主に(b)に関する話題について中心に述べる。

2. エンドユーザプログラミングのシナリオ

CLEPEはタスクオントロジーをシステムの基盤とすることで、計算機に馴染みのないエンドユーザが容易に問題解決知識を記述したり、その実行過程を概念レベルで確認することができる問題解決知識記述環境である。CLEPEにおけるエンドユーザプログラミングの過程は通常のプログラミング言語によるプログラミングと同様に大きく、(1)エンドユーザによる問題解決知識の記述、(2)問題解決知識の実行/デバッグの二つのフェーズに分かれている(図1)。それぞれの段階でエンドユーザはシステムから適切な支援を受けることができる。例えば(1)フェーズでは、(1-a)フローチャートに準じた記述の枠組みがエンドユーザに提供され、エンドユーザはチャート上で、「ジョブ」、「看護婦」、「選ぶ」、「割り付ける」といった日頃使っている言葉を使って問題解決知識を記述することができる。この「ジョブ」、「選ぶ」といった言葉の役割をタスクオントロジーは明確に定義している。さらに、このタスクオントロジーを利用することで、(1-b)エンドユーザが書いた問題解決知識の整合性の検証、(1-c)オブジェクトフローの半自動解釈といった機能が提供される。例えば図1において、Selectプロセスの出力である看護婦と、Assignプロセスの入力である看護婦が「同一の」看護婦であるということ(オブジェクトフロー)はチャート上で明示的に記述されていない。オブジェクトフロー解釈では、この同一性をタスクオントロジーに基づいてシステムが半自動的に解釈する。そして、その結果構成される概念レベル実行モデルでは、それらが「同一の」看護婦として明示的に表現される。

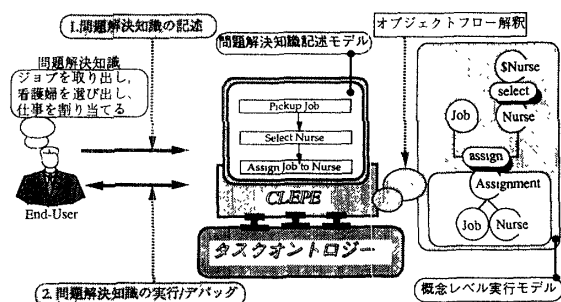


図1 CLEPEにおける
エンドユーザプログラミングのシナリオ

End-user Programming Support Functions Based on Task Ontology
Kazuhisa SETA, Teruyuki SHIMA, Mitsuru IKEDA, Osamu KAKUSHO, and
Riichiro MIZOGUCHI

¹ Institute of Scientific and Industrial Research, Osaka University, 8-1,
Mihogaoka, Ibaraki, Osaka 567 Japan

² Faculty of Economics and Information Science, Hyogo University, Hiraoka-cho,
Kakogawa 675-01 Japan

3. エンドユーザ支援機能：概念レベル実行

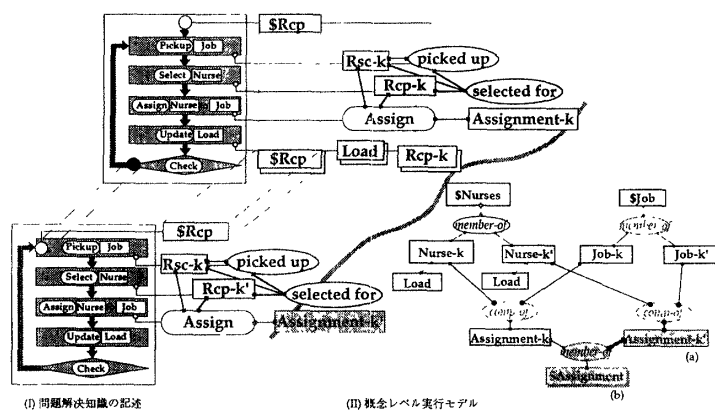


図2 問題解決知識の記述(a)と概念レベル実行モデル(b)

フェーズ(1)においてエンドユーザが「ジョブを看護婦に割り当てる要員配置問題」を対象にして書いた問題解決知識と、それに対応する内部モデル(概念レベル実行モデル)を図2に示している。図の問題解決知識記述モデルは、「ジョブの集合から一つのジョブを取り出し、それを割り当てるべき看護婦を1人選択し、新しい割り付けを生成する。そして、割り付けが行われた看護婦の負荷を更新し、まだ割り付け残るべきジョブが残っていれば同じ処理を繰り返す。」という問題解決知識を表している。

フェーズ(2)では、このようなユーザの意図をシステムが正しく解釈しているかを確認するために、概念レベル実行環境がエンドユーザに対して提供される。ここでは例えば、最終的な成果物となる「解」の変化について、システムは概念レベル実行モデルに基づき、「処理がAssignプロセスを通過するたびに一つの割り付けが生成され、それが割り付け集合(解)の要素となる。処理の過程でそれが中間解、暫定解と変化し、最終的な成果物である最適解になる。」という情報をエンドユーザに対して与える。この概念レベル実行によって、エンドユーザは問題解決知識(プログラム)の振る舞いを概念レベルで確認し、必要に応じてこれを修正することができる。

ここで、図2のモデルを比較して注目していただきたいことは、エンドユーザが明示的に記述していない対象世界の「もの」の変化、例えば、処理がAssignプロセスを通過した際に割り付けオブジェクトを構成するということはチャート上に明示的に記述(Assign Job to Nurse)されているが、^(a)割り付け集合の構成要素が増え、それによって、^(b)割り付け集

```

(define-problem-solving-causality PWM-GC (?p-obj ?w-obj ?act)
  :class-hierarchy (subclass-of PWM-GC
    Part-Whole-Member-of-Causality)
  :participant-constraint (
    (?p-obj :constraints ((class object)))
    (?w-obj :constraints ((class set)))
  )
  :axiom (
    (member-of ?p-obj ?w-obj)
    (propagation-change-from ?p-obj ?w-obj
      :generated ?p-obj)
    (version-change-time= ?p-obj ?w-obj)
  )
)
  
```

図3 問題解決 Causality 「PWM-GC」の定義

合が変化する(図2)ということは明示的に記述されていない。概念レベル実行モデルではこのような対象世界の物の変化がモデルとして明確に表現(図の網掛けされた部分に対応)されている。概念レベル実行では、このようなエンドユーザが重要と捉える暗黙な変化がエンドユーザに対して明示的に提示される。

4. タスクオントロジー：因果関係の明示化

先の例の(a)(b)に対応する部分は、それぞれタスクオントロジーとして定義された作用定義[2](Assigned)と、問題解決 Causality(Part-Whole-Member-of-Generate-Causality:PWM-GC)に基づいて、エンドユーザに対して提示される。Assignedは、Assignプロセスを実行した際の対象世界の物の変化を捉えるための概念である。以下では、PWM-GCについて説明する。

問題解決 Causalityは、エンドユーザが問題解決知識として明示的に記述しないが、暗黙に意識している重要な因果関係を捉えるための概念である。図3に Part-Whole-Member-of-Generate-Causality(PWM-GC)の定義を示している。集合オブジェクトの要素である部分オブジェクトが生成された際には、それによって全体オブジェクトが変化することになる。PWM-GCは、このような部分オブジェクトが生成された際に起こる全体オブジェクトの変化について、部分オブジェクトの生成と全体オブジェクトの変化の間の因果関係を捉えるための概念である。システムは、この定義を参照し、割り付けオブジェクトが生成された際におこる割り付け集合の変化を、重要な因果関係としてエンドユーザに対して提示することができる。

5. まとめ

本稿ではタスクオントロジーをシステムの基盤とすることで、問題解決過程においてエンドユーザが暗黙に意識する重要な因果関係を捉えることができることを、概念レベルプログラミング環境CLEPEを例にして述べた。

参考文献

[1] 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).
 [2] 島輝行, 瀬田和久, 池田満, 角所収, 溝口理一郎: 問題解決環境: CLEPEにおけるエンドユーザ支援機能, 信学技法, (1997-7).(掲載予定)