

オントロジーを利用したソフトウェアプロセス生成検査システム*

5C-8

 森 かおり 塩澤 朝彦 山口 高平†
 静岡大学‡

1 はじめに

本稿では、知識工学手法に基づくソフトウェアプロセス開発環境として代表的な Articulator^[1] のアプローチをさらに推し進め、プロセスオントロジーとオブジェクトオントロジーの構成方法、ならびにこれらを利用したプロセスモデル作成方法を検討する。

2 オントロジーの構成方法

知識ベースを構成する基本単位をオントロジーと呼ぶが、具体的には概念定義（概念スキーマ）と概念階層木を与えなければならない。また現在、オントロジーを構築するための方法が未整備であるから、順次概念を分化させるための属性（分類属性）を決定する方法を確立することが一つの課題となっている。

本節では、ソフトウェア開発プロセスに対し、様々なソフトウェア開発活動（プロセス）と操作対象物（オブジェクト）に着目して、プロセススキーマとオブジェクトスキーマの表現モデルを検討した後、上位下位階層の概念分化では、根節点の概念を表現する上で重要なスキーマの記述属性を分類属性に選び分化を進める方法を考察し、アクティビティを根節点としたプロセスオントロジーと、オブジェクトを根節点としたオブジェクトオントロジーの構成方法について述べる。プロセスとオブジェクトを個別に整理することにより、プロセスの変化に対応しやすくなり、維持が容易になると考えられる。

2.1 プロセスオントロジー

具体的なプロセスを表現するためのスキーマとその構成要素（プロセスの記述属性）の意味を表1に示す。

表1: プロセススキーマの記述属性

構成	記述属性	記述属性の説明
プロセスで操作されるオブジェクト	input object	プロセスに入力され操作の対象となるオブジェクト
	output object	プロセスから出力されるオブジェクト
	reference object	入力されたオブジェクトを操作するために参照されるオブジェクト
プロセスで実行される時のコンテキスト	tool	プロセスを実行する時に利用するツール等
関連する他のプロセス	agent	プロセスを実行する人
	has - a predecessor process	サブプロセス 一つ前に処理されるプロセス

プロセスオントロジーでは、入力を出力に変換する活動をプロセスと捉え、入出力オブジェクト間の共通部分の存否という関係に注目して概念の分化を進め、それだけでは分化しきれない下位階層においては、プロセススキーマの他の記述属性により分類する。このようにして、M.I.Kellner らによって提案された^[2] ソフトウェアプロセスモデリングのための共通例題^[2] に含まれる具象プロ

セス群を分類した階層木（プロセスオントロジー）の一部を図1に示す。例えば、図中の層1では出力オブジェクトが入力オブジェクトのオブジェクトスキーマの記述属性の1つである has-a に含まれていれば、活動を遂行するための探索空間が狭いということ。Analyze と、そうでなければ広いということ。Synthesize に概念を分化する。層2では入出力オブジェクトの has-a を比較し、一致するものがなければ新しいものを作り出したということ。Make、あれば入力オブジェクトに変更を加えたということ。Modify に分化する。層3では、これ以上入出力オブジェクトの関係により分化を進めることができないため、プロセススキーマの出力オブジェクトの型に注目してさらに分化を進めた。

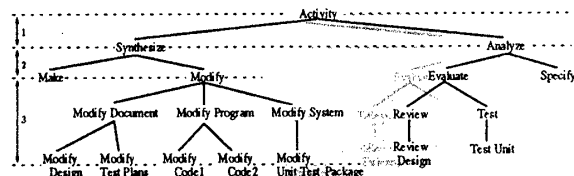


図1: プロセスオントロジーの一部

2.2 オブジェクトオントロジー

プロセスで操作される様々な対象物を表現するためのスキーマとその記述属性の意味を表2に示す。

オブジェクトオントロジーは対象物をソフトウェア開発プロセスに特化した is-a 関係により構成する。そのようにして共通例題に含まれるオブジェクトの階層木（オブジェクトオントロジー）の一部を図2に示す。

表2: オブジェクトスキーマの記述属性

記述属性	記述属性の説明
has - a	所有関係
relevance	オブジェクトにおいて注目されるもの
object type	表現形式

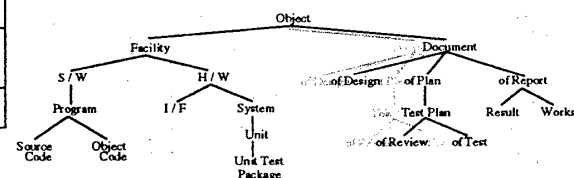


図2: オブジェクトオントロジーの一部

3 ソフトウェア開発プロセス生成検査システム

本節では、2節で構成したプロセスオントロジーとオブジェクトオントロジーを利用して、ユーザからの要求に対するプロセスモデル候補を生成、検査するシステム構成法を基本設計し、机上実験によりシステムの適用可能性について考察する。

*Generating and Testing Software Processes Using Ontologies

†Kaori Mori, Tomohiko Shihozawa and Takahira Yamaguchi

‡Shizuoka University

3.1 システムの基本設計

ソフトウェア開発プロセス生成検査システムは、あるプロダクト(プロセスモデル開始点)から別のプロダクト(プロセスモデル終了点)を生成するための開発手順をユーザとインタラクションをとりながらプロセスモデルとして出力するシステムであり、その概観を図3に示す。

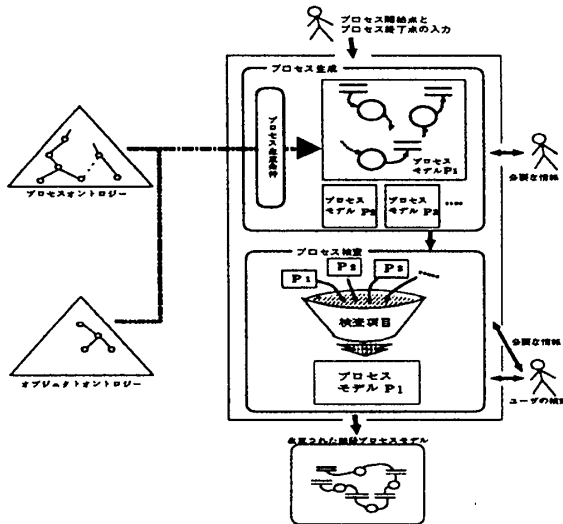


図 3: システムの概観

(1) プロセスモデルの生成

プロセスモデルの開始点と終了点を入力し、オントロジーを利用して、開始点を入力オブジェクトに持つプロセスを検索し、次にそのプロセスの出力オブジェクトを入力オブジェクトに持つプロセスを検索する。同時に終了点からも逆の検索を行ない、2つの検索結果が合流するまで双方向に検索を続けることによってプロセスモデル候補が生成される。プロセスは、スキーマレベルでの属性値の照合により検索し、その際必要な情報やユーザの持つ条件はユーザとインタラクションをとることによって検索条件に加え、属性が最も一致したものを採用する。照合においては属性値はオブジェクトオントロジー上の概念となるため、オブジェクトオントロジー上の包含関係により照合を行なう。

(2) プロセスモデルの検査

まず、生成されたプロセスモデル候補の検査項目について、開発期間や信頼性、および資源の衝突などからユーザの要求に従って選択する。次に、その項目についての検査戦略を利用し、プロセスモデルを各プロセスのスキーマ属性を参照したり、またユーザとインタラクションを通して検査することで、最も適切なプロセスモデルが選択される。

3.2 机上実験

開始プロダクトとして現在のデザイン(設計)、終了プロダクトとしてテスト結果を入力し、プロセスモデルを作成する。まず「現在のデザイン」から前向きに「テスト結果」から後ろ向きに双方向に検索を始め、合流するまで検索を続けた。その結果、2つのモデルが作成された。

次に、この2つのモデルを検査項目に従って検査をする。今回、ユーザから「信頼性の高いプロセスモデル」という要求が与えられ、それをReviewのプロセスをより多く含むモデルを選択するといったような戦略に変換し、

それらを利用して検査した結果、プロセスモデル1が選択された。この処理過程を示したものが図4である。

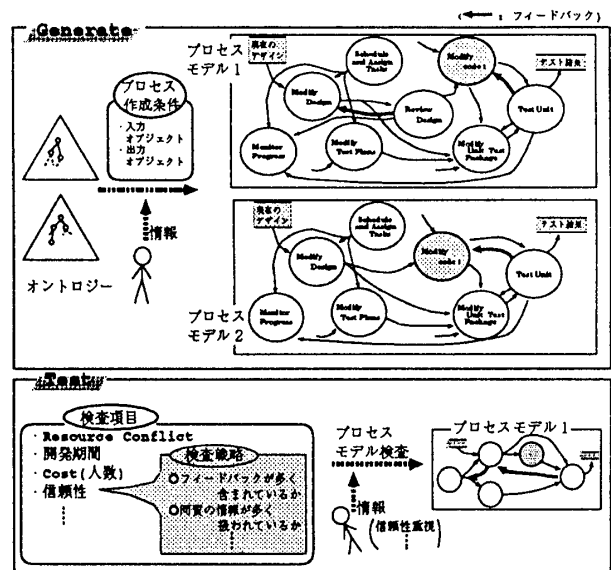


図 4: 机上実験

3.3 考察

本机上実験では、比較的限定された開発手順に関するプロセスモデルを生成したため、2つのプロセスモデルが生成され、検査の結果、一意に決定できたが、大規模なプロセスモデルを生成する際、生成されるモデルが多すぎたり、逆に前後から検索していった結果モデルができない可能性がある。多くのモデルが生成される場合は、ユーザが明確な条件を持っていなかったことが考えられ、ユーザが生成されたモデルを基にして条件を付加して再び生成を行なうことにより適切なモデルを生成することができると考えている。またモデルが生成されない場合は、オントロジーの記述が不十分であったり、ユーザの要求に誤りがあったことが考えられる。その場合はオントロジーの精練や、ユーザに入力の再考を求める機構を考察することにより解決できると考えている。

4 おわりに

本稿では、プロセスオントロジーとオブジェクトオントロジーの構成方法を示すとともに、ユーザの要求からプロセスモデルを生成・検査するシステムを基本設計し、簡単な机上実験からその有効性を確認した。現在、システム開発取り引きの共通フレーム^[5]などからさらにオントロジーの整備を進め、SICStusPrologにより実装を始めている。

参考文献

- [1] Peiwei Mi and Walt Scacchi. "A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes". IEEE Transactions on Knowledge and Data Engineering, Vol.2, No.3, pp.283-294 (1990).
- [2] 落水 浩一郎."ソフトウェアプロセスに関する研究の概要". 情報処理, Vol.36, No.5, pp379-385 (May.1995).
- [3] 大野 豊 監修, 共通フレーム検討委員会編集: システム開発取り引きの共通フレーム SLCP-JCF 94, 通産資料調査会 (1994).