

# CAPIS モデルと品質実現オントロジを用いる クラス抽出思考過程の獲得・伝達法

大山 勝徳<sup>†</sup> 武内 惇<sup>††</sup> 藤本 洋<sup>††</sup>

信頼性や効率性に対する要求が厳しいリアルタイムシステムにおいて、システムの分析・設計者（以降、分析者）にとって様々な品質特性を同時に考慮してクラスを抽出することは難しい。品質特性を実現するための最適なクラス抽出を行うには、オブジェクト指向の一般化された技術だけでなく、技術を繰り返し使用して獲得する熟練技術（経験技術）を駆使することが求められる。本論文では、品質特性を実現する分析者（経験者）の経験技術を他の分析者が利用できるようにするため、設計思考過程を表現する CAPIS モデルをクラス抽出法に適用する。さらに、品質特性を実現するための課題を表すオントロジ（品質実現オントロジ）を用いて経験者の思考過程を獲得し、他の分析者に伝達する CAPIS/OOM（CAPIS model/Ontology based Object Modeling）について述べる。本方式を用いた実験では、品質特性を実現する経験者の思考過程を他の分析者が継承して使用できることと、経験者の思考過程を用いて新たな思考過程を創出することにより複数の品質特性を実現するクラス抽出が行えることを検証する。

## A Thought Process Expression for Finding Classes Using CAPIS Model and Quality Building Ontology

KATSUNORI OYAMA,<sup>†</sup> ATSUSHI TAKEUCHI<sup>††</sup> and HIROSHI FUJIMOTO<sup>††</sup>

In the real-time system development that often require reliability and efficiency, considering quality characteristics is difficult for system analysts/designers (analysts). The analysts need to find classes using experienced techniques, as well as well-known object-oriented techniques. This paper describes the CAPIS/OOM (CAPIS model/Ontology based Object Modeling) for finding classes with considering quality characteristics based on CAPIS model and Quality Building Ontology so that experts' thought processes can be used by the other analysts. Then we experiment to validate which the analysts can inherit a thought process and can generate their new thought process based on the expert's thought process using CAPIS/OOM.

### 1. はじめに

オブジェクト指向分析・設計（以下、分析と呼ぶ）は一般的に使用されるようになってきたが、クラス抽出についてはまだ分析者の経験による部分が多い。

また、リアルタイムシステムの分析では稼働環境やリソース制限を要因とするトレードオフが多く、品質特性（ISO/IEC9126-1<sup>1)</sup>の機能性、信頼性、使用性、効率性、保守性、移植性）を実現することが困難である。MDA 技術<sup>2),3)</sup>が分析作業を幅広く支援する状況の中で、品質特性を実現するクラス抽出作業ははまだ

に分析者の思考活動に依存している<sup>4)</sup>。

一方、筆者らは分解、統合の2段階に分けて要求仕様記述からクラスを抽出する方法<sup>5)</sup>を提案している。この方法においても品質特性を実現する経験者のクラス抽出作業は、他の分析者に示すことができない。

筆者らは、品質特性を効率良く作り込むには、技法やツールの活用経験を持つ分析経験者（経験者）の思考過程を利用できるようにすることが重要だと考える。このため、経験者の思考過程を表現する CAPIS (CAusality of Problem-Issue-Solution) モデル<sup>6),7)</sup>をクラス抽出プロセスに適用することにより、経験者のクラス抽出の思考過程を獲得し、その思考過程を分析者が使用できるようにする方式を検討している。

様々な経験者の思考過程は、それぞれ品質のとらえ方や実現法が異なるために、経験者が思考過程を個別に蓄積しても、利用者がそれらの思考過程の中から最適な思考過程を選択することは難しい。この原因は、

<sup>†</sup> 日本大学大学院工学研究科情報工学専攻  
Graduate School of Computer Science, College of Engineering, Nihon University

<sup>††</sup> 日本大学工学部情報工学科  
Department of Computer Science, College of Engineering, Nihon University

(1) 経験者が品質特性を独自に定義するため複数の経験者が示す品質特性の関連付けが困難である, (2) 複数の品質特性を同時に目標とする課題を経験者が設定していることがあげられる.

この問題を解決するため, 「対象概念の構造」を示すオントロジに着眼し, CAPIS モデルとオントロジに基づくクラス抽出法 CAPIS/OOM (CAPIS model/Ontology based Object Modeling) を提案する. さらに, ライトレールロボットプログラムの開発を例に, 経験者の思考過程を分析者が継承, 創出できることを確認する実験を行い, CAPIS/OOM により複数の品質特性を実現するクラス抽出が行えることを検証する.

本論文では, 2章で CAPIS モデルに基づいて品質特性を実現するクラス抽出法について説明する. 3章で品質実現オントロジの構成について, 4章で CAPIS/OOM の概念成熟プロセスとクラス抽出プロセスを述べる. 5章で思考過程を獲得する実験について, 6章で経験者の思考過程を他の分析者へ伝達する実験について紹介するとともに, 7章で CAPIS モデルの思考過程を用いる分析者のクラス抽出結果を考察する. 8章で関連研究と本方式を比較して位置付けを明らかにした後, 9章で結論と今後の課題を述べる.

## 2. CAPIS モデルに基づくクラス抽出法

### 2.1 CAPIS モデル

設計作業は, 設計要件 (入力仕様) を解釈して具体的な設計結果 (出力仕様) を定義するまでの手順で構成される. 各手順を実行する分析者の思考過程は, 入力仕様を解釈し, 問題を発見し, 次に, その問題を解決するための選択肢 (設計判断) を想起して設計結果 (出力仕様) に具体化するまでの試行錯誤を繰り返す知的活動である.

試行錯誤を繰り返す思考過程は, 複雑性と多様性があり, すべて表現することが困難であることから, 「問題」「課題」「対策」(PIS: Problem-Issue-Solution) の分割表現でモデル化する<sup>6)</sup>. 「問題」と「対策」のみで構成した思考過程は, 「対策」を取捨選択する根拠の表現が難しい. このため, 「問題」と「対策」の因果関係を結びつける「課題」を導入する. 「課題」は, 問題の原因に対する着眼点であり, 「問題」の原因と「対策」のアイデアを結びつけるものである. 「課題」を設定することは, 「問題」を解決する道筋を与えるので「対策」の根拠を示すために重要である. 以上の考え方を基にして, 筆者らは「問題」「課題」「対策」を中心に思考過程をモデル化する. この思考過程のモ

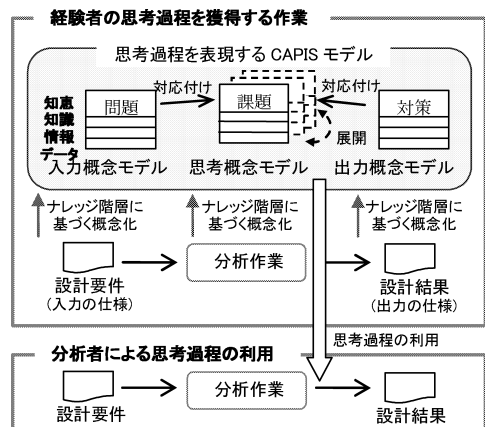


図 1 CAPIS モデル  
Fig. 1 CAPIS model.

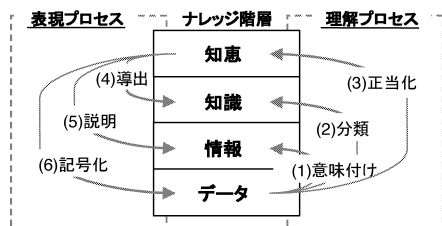


図 2 ナレッジ階層に基づく概念化プロセス  
Fig. 2 The knowledge hierarchical conceptualization process.

デルを CAPIS モデルと呼ぶ.

CAPIS モデル (図 1) は, 「問題」を表す入力概念モデル, 「課題」を表す思考概念モデル, 「対策」を表す出力概念モデルを用いて設計思考過程を表現する. 各概念モデルは, 設計文書に記述されなかった分析者の常識・専門知識や設計意図を表すために, データ層, 情報層, 知識層, 知恵層のナレッジ階層<sup>8),9)</sup>に基づく概念化プロセスに従って記述するモデルである.

ナレッジ階層に基づく概念化プロセス (図 2) は, データ層, 情報層, 知識層, 知恵層の階層関係に従う概念モデルを記述するために「理解プロセス」と「表現プロセス」で進める. 理解プロセスは, 設計文書に記述された主題 (問題, 課題, 対策を表す概念化の対象) を明確にして概念モデルを記述する手順である. 表現プロセスは, 主題を他の分析者へ伝達可能とするために知恵層に示す判断に基づいて概念モデルを記述する手順である.

### 2.2 クラス抽出の思考過程の獲得法

思考過程の獲得は, 経験者の思考過程を他の分析者が理解できる形式で表現することである. 本節においてはクラス抽出の設計意図を伝達するナレッジ階層の考え方と, 思考過程の獲得手順を CAPIS モデルに基

表 1 クラス抽出の思考過程表現に用いるナレッジ階層  
Table 1 Knowledge hierarchy for class finding thought processes.

ナレッジ階層	定義	入力概念モデルと出力概念モデル	思考概念モデル
知恵	問題を解決していく過程で判断した内容と、それを正当化する理由と価値	<ul style="list-style-type: none"> <li>機能や品質に関する問題の状況とそれが発生する原因、分析・設計者や設計対象に与える影響</li> <li>対策となるクラス抽出の方法と、その方法に基づくアイデア、対策が持つ効果</li> </ul>	<ul style="list-style-type: none"> <li>課題(品質実現の課題)を解決するため選択肢とその着眼点、意義</li> <li>特に、クラス抽出においては機能や品質の要求を実現するための選択肢</li> </ul>
知識	常識や専門知識を表す命題(ルールや事実)	<ul style="list-style-type: none"> <li>機能やクラス構成要素に関する問題、または、対策を表す事実</li> <li>問題や対策に関する常識や専門知識</li> </ul>	<ul style="list-style-type: none"> <li>知恵層で示した選択肢を実行するためのクラス抽出プロセス</li> <li>品質特性の定義</li> </ul>
情報	記号や記号列の意味の説明	<ul style="list-style-type: none"> <li>知恵層で示された問題や対策の意味</li> <li>データ層で示された用語の意味</li> </ul>	<ul style="list-style-type: none"> <li>知恵層で示された課題の意味</li> <li>クラス抽出時に有用となった事実</li> </ul>
データ	文章や図で表現された記号や記号列	<ul style="list-style-type: none"> <li>要求仕様書に記述された機能、または、クラス図におけるクラス構成要素</li> </ul>	<ul style="list-style-type: none"> <li>マニュアルや指導書、または、報告書に示すクラス抽出法の事例</li> </ul>

づいて決定する方法について述べる。

### 2.2.1 クラス抽出の思考過程獲得に用いるナレッジ階層の考え方

入力概念モデルにおけるナレッジ階層は、分析作業の入出力となる要求仕様書やクラス図をデータ層の内容とし、データ層を基に経験者があげる機能や品質の問題を情報層、知識層、知恵層で表現する(表1)。

出力概念モデルにおけるナレッジ階層は、クラス抽出結果を表すクラス図をデータ層の内容とし、データ層を基に対策となるクラス抽出の方法を情報層、知識層、知恵層で表現する。

思考概念モデルは、クラス抽出の設計意図の表現を目的としてナレッジ階層を構成する。知恵層の内容は、品質特性を実現するための課題とそれを解決する選択肢である。知識層の内容は、知恵層の選択肢を実行するクラス抽出プロセスや経験者が前提とする品質特性の定義である。情報層の内容は、クラス抽出プロセスにおいて経験者が参考にする事実である。データ層の内容は、知恵層の内容を具体化して説明するクラス抽出法の事例である。

### 2.2.2 思考過程の獲得手順

経験者は、CAPISモデルによる思考過程の表現手順に従い、(1)入力概念モデルの記述、(2)出力概念モデルの記述、(3)思考概念モデルの記述の順に思考過程の獲得を進める<sup>6)</sup>。

#### (1) 入力概念モデルの記述

要求仕様書やクラス図の中から機能や品質に関する「問題」の事例をデータ層に記載する。次に、情報層、知識層、知恵層の順に入力概念モデルを記述する。

#### (2) 出力概念モデルの記述

入力概念モデルと同様に、設計文書の中から「対策」を表すクラス抽出結果の事例をナレッジ階層のデータ層に記載する。次に、情報層、知識層、知恵層の順に概念モデルを記述する。

#### (3) 思考概念モデルの記述

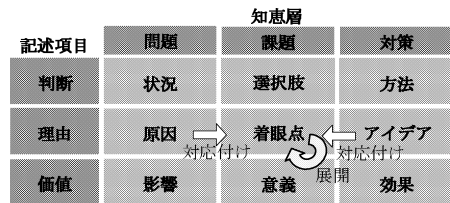


図 3 対応付けと展開

Fig. 3 CAPIS mapping and development.

入力概念モデルと出力概念モデルを関係付ける「課題」を記述するため、対応付けと展開により品質特性の目標を実現するクラス抽出の選択肢を表出し、思考概念モデルの知恵層に記述する。その後、知恵層に基づいて知識層、情報層、データ層の順に思考概念モデルを記述する。

思考過程の対応付けと展開は、課題を経験者に想起させる方法であり、以下の手順で進める。

#### (a) 思考過程の対応付け

問題の「原因」と対策の「アイデア」の因果関係に注目して「着眼点」を記述する(図3)。次に、設計判断として採用したクラス抽出の「選択肢」を記述する。さらに、課題を実現することによる「意義」を記述する。

#### (b) 思考過程の展開

##### (b-1) 設計判断の展開

採用した「選択肢」以外に経験者が検討しているクラス抽出の「選択肢」を記述し、それらと比較することにより、課題を解決するための選択肢が採用される理由、破棄される理由を「意義」に示す。

##### (b-2) 課題の展開

1つの問題を解決するためにすでに設定した課題とは異なる課題を設定してクラス抽出を行っている場合、「選択肢」「着眼点」「意義」を示して新たな課題を記述する。

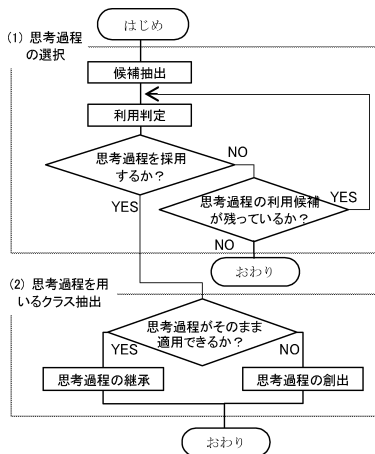


図 4 思考過程を用いるクラス抽出手順

Fig. 4 Procedure of class finding using a thought process.

### 2.3 思考過程を用いるクラス抽出手順

CAPIS モデルの思考過程を用いる最初の手順は、問題、課題、対策の知恵層に記述された内容と分析者自身の考えを比較し、経験者の思考過程を選択する作業である。次に、経験者の思考過程を利用するために継承、または、創出を行う。

思考過程を用いるクラス抽出手順のフローチャートを図 4 に示すとともに、以下、各手順の詳細を説明する。

#### (1) 思考過程の選択

##### (a) 候補抽出

複数の思考過程の中から分析者が必要とする思考過程を絞り込む。思考過程を絞り込む手法は、すでに様々な検索技術があるため CAPIS モデルで特に規定せず、思考過程の利用環境に従う。

##### (b) 利用判定

問題や課題、対策の項目（図 3）で経験者の思考過程と分析者自身が思考している項目との比較を行う。経験者の思考過程に分析者が共感する項目がある場合、思考過程を採用する。

**問題の比較** 経験者の思考過程に示された「状況」と分析者自身が直面する「状況」を比較する。一致する場合、次に、「原因」または「影響」が重要であることを分析者によって判定する。

**課題の比較** 経験者の「選択肢」と分析者が必要とする「選択肢」を比較する。分析者が検討している「選択肢」がある場合、「着眼点」または「意義」が重要であることを判定する。

**対策の比較** 経験者の「方法」が分析者自身のシステム開発で利用可能な「方法」であることを確認する。「方法」が利用できる場合、「アイ

デア」または「効果」から対策の有効性を判定する。

#### (2) 思考過程を用いるクラス抽出

##### (a) 思考過程の継承

分析者が選択した経験者の思考過程を参照し、課題の知恵層を構成するクラス抽出の選択肢や知識層のルールに従いクラス抽出の作業を進める。

##### (b) 思考過程の創出

経験者の思考過程に分析者が必要とする課題が記述されていない場合、経験者の思考過程に新たな課題を展開してクラスを抽出する。すなわち、分析者は、経験者が思考していない新しい課題を解決するクラスを抽出する。その後、分析作業が終わった段階で分析者が創出する思考過程を獲得する。

### 2.4 思考過程を用いるクラス抽出の課題

様々な経験者の思考過程は、それぞれ品質のとらえ方や実現法が異なるために、経験者が思考過程を個別に蓄積しても、利用者がそれらの思考過程の中から最適な思考過程を選択することは難しい。この原因を以下にあげる。

#### (1) 経験者が定義する品質特性の独自性

品質特性は、ISO/IEC9126-1 等の標準規格に従っていても経験者の設計意図によって定義の解釈が異なることがある。経験者が行う定義の解釈を理解していない分析者は、実現すべき品質特性とは異なる目標の思考過程を継承する可能性がある。

#### (2) 複数の品質特性を実現する課題の表現

経験者が行うクラス抽出の思考過程は、機能性や信頼性、使用性等の複数の品質特性を同時に目標とする課題を設定することが多い。1つの課題に複数の品質特性を実現する目標を設定することによって、品質特性を実現する思考過程が複雑となり、対策に記述しきれないことが多い。

本論文ではこれらの原因の解決を目的とし、経験者が目標とする品質特性（品質目標）を分析者が正確に理解し、クラス抽出に利用できるようにする。このため、「概念の構造」を示すオントロジに着眼し、以下に示す思考過程の継承と創出の課題を設定する。

##### (a) オントロジによる品質実現の課題の表現法

##### (b) 思考過程の獲得手順と利用手順の定式化

筆者らは、(a) を解決するために品質実現オントロジを提案する。さらに、(b) を解決するために品質実現オントロジを用いる CAPIS/OOM のプロセスを開発する。

表 2 クラス抽出に用いる品質考慮点表  
Table 2 Quality consideration table for class finding.

品質特性	品質副特性	品質考慮点
機能性	目的性	(1-1) クラスは要求仕様書の内容に沿って識別されている。 (1-2) クラス構造(属性や操作等の構成要素)は、要求される機能を実現できるように構成されている。
	標準適合性	(1-3) 記法や分析法が組織で用いるオブジェクト指向方法論に準じている。
信頼性	成熟性	(2-1) 外部装置から受け取るデータを最適な値に補正する。
	障害許容性	(2-2) 障害時におけるクラスの振舞いが定義されている。
使用性	回復性	(2-3) 障害が起きてから回復するまでの振舞いが定義されている。
	理解性	(3-1) 他の分析者がシステムの仕様や意図をすぐ理解できるように、クラスや属性、操作等に名前付けをしている。 (3-2) クラス構造やその振舞いはシンプルである。
	運用性	(3-3) クラスは各ドメインヘカテゴリー化して配置されている。
効率性	習得性	(3-4) クラスの説明やメソッドの用法がドキュメント化等によって示されている。 (3-5) システムを稼働させるための手順が少ない。 (3-6) クラス(サービスドメインの API)の操作法はルールが少なく、使いやすい。
	時間効率性	(4-1) イベントや状態遷移の図を用いてクラスの振舞いには無駄がないことを説明している。
保守性	資源効率性	(4-2) クラス間の呼びだし関係が複雑ではない。 (4-3) 別のクラスの責務と重複していない (4-4) クラスが持つデータの規模は最小限である。
	変更性	(5-1) 機能やデータが追加・変更されても、より少ない労力でクラスを修正できる。
移植性	安定性	(5-2) クラス構造は凝集度が高く、クラス間の関係は結合度が低いという特徴を持っている。
	環境適合性	(6-1) クラスは下位ドメイン(OS や言語等)の特殊なサービスに依存していない。
	標準適合性	(6-2) 組織内で標準のドメイン、アーキテクチャを利用することを前提としている。
	置換性	(6-3) クラスは修正することなしに別のシステムにも利用できる。

※ 表内の品質考慮点は経験者が分類した結果であり、他の品質副特性についての品質考慮点は今後追加される。

### 3. 品質実現オントロジの構成

品質実現オントロジを基に品質実現の課題を定義する品質考慮点表、および、品質実現オントロジについて述べる。

#### 3.1 品質考慮点表

品質考慮点表は、経験者が各々の品質特性をどのように解釈したかを表すため、経験者の思考過程から抜き出して分類する品質実現の課題である。表 2 において CAPIS モデルの思考概念モデルへのリンク番号(1-1, 1-2 等)を明示することにより、分析者の品質実現の課題に対応する経験者の具体的な思考過程を選択できるようにする。

品質考慮点表は、品質実現オントロジに基づいて経験者の思考過程の知識層に記述される品質考慮点を経験者が追加することで作成する。品質考慮点表の作成手順は(1) クラス抽出、(2) クラス抽出思考過程の獲得、(3) 品質考慮点の追加の順序となる。

#### 3.2 品質実現オントロジ

品質実現オントロジ(図 5)は、思考概念モデルの記述項目を示すため、データ層、情報層、知識層、知恵層に分けて構成する。品質実現オントロジにおける知恵層は、「品質実現の課題」を「選択肢」「着眼点」「意義」の一組にして構成する。「選択肢」は、品質実

現の課題を解決する分析者の設計意図である。

また、品質実現オントロジは、品質実現の課題とその課題を解決する設計意図を段階的に表現するため、「目標表現」の品質実現オントロジと「設計意図表現」の品質実現オントロジの 2 つに分けて構成する。「目標表現」は、経験者の思考過程を俯瞰し、使用する思考過程を選択可能にすることを目的とする表現である。「設計意図表現」は、経験者の設計意図やルールの表現であり、思考過程の継承と創出を正確に行うために用いる。

#### (1) 「目標表現」の品質実現オントロジ

「目標表現」の品質実現オントロジは、「品質実現の課題」を示す知恵層、知識層、情報層、データ層を構成要素とする(図 5a)。知識層は、知恵層の「品質実現の課題」を説明する知識として「品質考慮点」と「品質定義」を記述項目とする。「品質考慮点」は、思考過程を獲得する経験者が設定する品質実現の課題である。「品質定義」は、経験者の視点でとらえた ISO9126-1 の品質特性の定義である。情報層の「調査事実」は、課題解決のためにデータを分析することによって経験者が得た有用な事実である。

#### (2) 「設計意図表現」の品質実現オントロジ

「設計意図表現」の品質実現オントロジは、品質特性を実現する経験者の設計意図や、常識・専門知識を

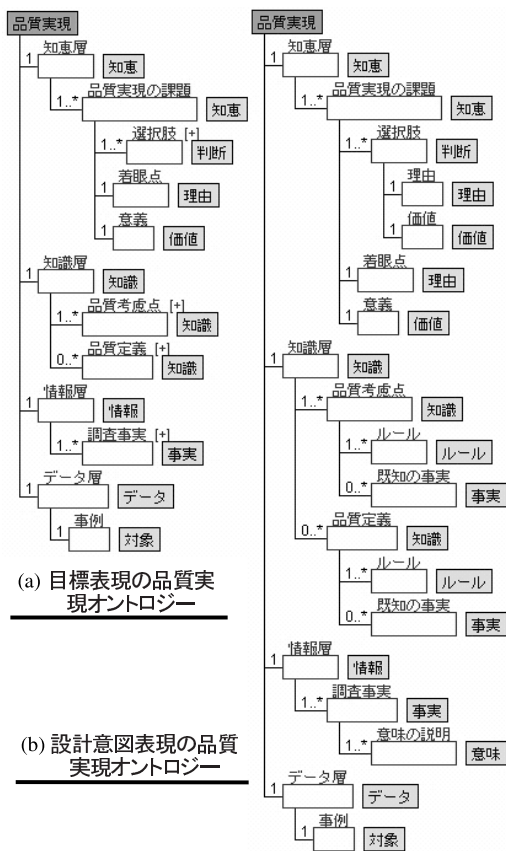


図 5 2 種類の品質実現オントロジ  
Fig. 5 The quality building ontology.

示すために「目標表現」の品質実現オントロジをさらに詳細化する内容を構成要素とする(図 5b). 知恵層は、「品質実現の課題」を表すものであり、選択肢を決定する根拠を明確にするため課題解決の「選択肢」「着眼点」「意義」を記述項目とする。知識層は、「品質考慮点」や「品質定義」に関して経験者が決定する「ルール」や、経験者が理解している「既知の事実」、すなわち、常識や専門知識を記述項目とする。情報層の「意味の説明」は、「調査事実」の記述に用いられたデータの説明を表す。

#### 4. CAPIS/OOM プロセスの定式化

CAPIS/OOM プロセスは、品質実現オントロジを用いて概念モデルを段階的に記述する「概念成熟プロセス」と、経験者の思考過程を用いる「クラス抽出プロセス」で構成する。

##### 4.1 オントロジを用いる概念成熟プロセス

経験者によって最初に記述された思考概念モデルは、データ層、情報層、知識層、知恵層が品質実現の課題を正確に表していないことが多い。このため、概念成

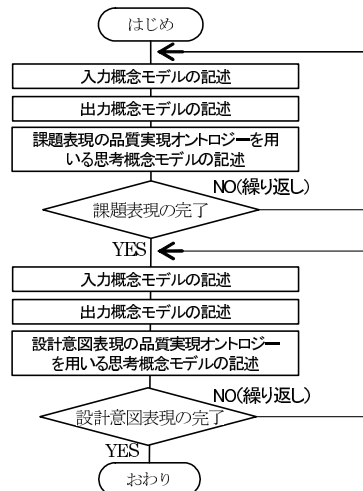


図 6 品質実現オントロジを用いる概念成熟プロセス  
Fig. 6 Concept maturation process using quality building ontology.

熟プロセスは、ナレッジ階層に基づく概念化を理解プロセスと表現プロセスの 1 サイクルのみで完了させるのではなく、課題と設計意図を正しく表現するための品質実現オントロジを用いて 2 段階の概念化を行い、品質実現の課題を正確に表現できるようにする。特に、思考概念モデルの記述では、目標表現の品質実現オントロジを用いて概念化を行った後、設計意図表現の品質実現オントロジを用いて概念化を行う(図 6)。

##### 4.2 オントロジを用いるクラス抽出プロセス

クラス抽出時に実現すべき品質特性は、多くの種類があるにもかかわらず、分析者が同時に検討できる品質特性の数には限りがある。特に分析初心者は、要求される機能を実現するようにクラスを構成する(機能性を実現する)だけで手一杯であり、機能性以外の信頼性、使用性、効率性、保守性、移植性を満たすクラスを構成できないことが多い。このため、クラス抽出プロセスは、以下の機能性と保守性、移植性の特性に基づいて各品質特性を実現する順序を決定する。

(a) 「機能性を実現するクラス抽出」は、ユーザからの要求事項を達成するためのクラスの抽出作業であり、他のどの品質特性との競合があっても優先すべきである。

(b) 「保守性を実現するクラス抽出」と「移植性を実現するクラス抽出」は、ソフトウェア製品の納入後に発生する作業負荷を軽減するためのクラスの抽出作業である。これらのクラス抽出作業は、初期の分析段階でユーザから直接要求される特性ではないが、設計者がクラスを再利用するために分析作業の後工程で重要となる。

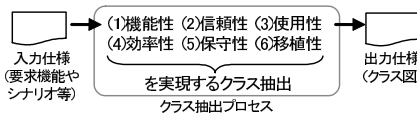


図 7 品質実現オントロジを用いるクラス抽出プロセス  
Fig. 7 Class finding process using quality building ontology.

品質特性を実現するクラス抽出プロセスは、クラスの追加や再構成にともなう各品質特性の影響を最小限に抑えるため、品質特性を実現する優先順序に基づき 6 段階のプロセス（機能性の実現、信頼性の実現、使用性の実現、効率性の実現、保守性の実現、移植性の実現）で構成する（図 7）。

各段階のクラス抽出プロセスは、以下の手順に従う。

#### (1) 品質実現の課題設定

経験者が作成する品質考慮点表から品質実現の課題と品質考慮点を選択する。また、このとき品質考慮点表で示すリンク番号に対応する思考過程を選択する。

#### (2) 思考過程の選択

思考過程の比較（問題の比較、課題の比較、対策の比較）を行うことにより、利用すべき思考過程を選択する。

#### (3) 思考過程の利用

##### (a) 思考過程の継承

思考過程を利用する場合、設計意図表現の品質実現オントロジに従って思考概念モデルの「品質実現の課題」「品質考慮点」「品質定義」「調査事実」「事例」を参照する。さらに、品質実現オントロジの詳細レベルである「選択肢」や「ルール」「意味の説明」に記述された内容を用いてクラス抽出を行う。

##### (b) 思考過程の創出

経験者の思考過程を分析者のシステム分析に適用することにより新たな問題が生じる場合、経験者の思考過程を展開し、クラス抽出を行う。

創出された思考過程の獲得は、分析者自身が入力概念モデルの記述と出力概念モデルの記述を行い、次に、品質実現オントロジを用いる概念成熟プロセスに従って思考概念モデルを記述する順に進める。

### 5. 思考過程の獲得実験

本実験においては、CAPIS/OOM の有効性を確認するため品質特性を実現する経験者 A の思考過程を獲得する。実験に参加した経験者 A は、UML を用い

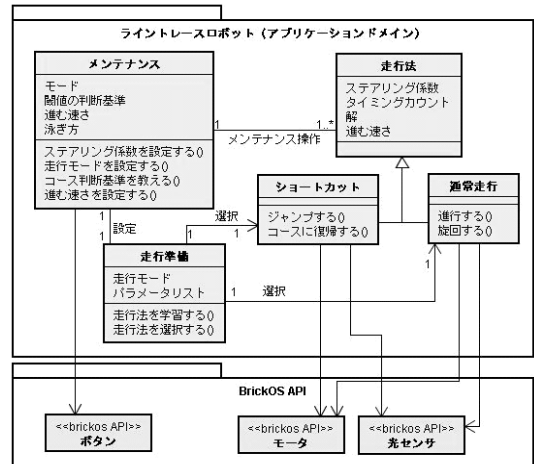


図 8 メンテナンスクラスに関連する部分のクラス図  
Fig. 8 The part of class diagrams concerning maintenance.

るオブジェクト指向モデリングを過去 5 年行い、ET ロボットコンテスト<sup>10)</sup> のライトレースロボットプログラム開発を 2 度行った経験を有する。

経験者 A は、ライトレースロボットの分析を行い、次に、思考過程の獲得を行う。

#### 5.1 経験者の分析結果

経験者 A は、ライトレースロボットの仕様を調査し、クラス図を作成する過程で最初に機能性、信頼性、使用性を考慮してクラス抽出を行った後、保守性を実現するクラス抽出を行った。

分析結果（図 8）は、ライトレースロボットの走行を制御するとき用いられるパラメータ（ステアリング制御の特性関数に用いる値、走行方式、ライトレースロボットが走行する速さ、ライン識別の色判断境界等）を調整・把握できるように、「メンテナンスクラス」を抽出したことを示している。パラメータの設定は、ユーザのボタン操作によって行う。

#### 5.2 思考過程の獲得結果

経験者 A は、ET ロボットコンテストの走行環境で効率的にプログラムのテストを行うための保守性が重要だと考え、保守性を実現するクラス抽出の思考過程（図 9）を獲得した。

##### (1) 入力概念モデルの記述結果

入力概念モデルは、「走行法」クラスをデータ層として「走行テストの作業に時間がかかる」ことを問題としている（図 9a）。この「走行法」クラスは、ステアリング制御のためのパラメータが多数あり、修正したプログラムをコンパイルして対象機器に書き込む手間がテスト効率の低下を生じさせている。

##### (2) 出力概念モデルの記述結果

<p><b>問題(知恵層)</b> (状況) ・パラメータの調整や把握が困難となり、走行テストの作業に時間がかかる。 (原因) ・開発対象はコンパイル後にプログラムを転送する組み込み機器である。 (影響) ・プログラムにアルゴリズムを追加することによりプログラムの分岐条件が複雑化し、テスト効率が低下する。</p>	<p><b>課題(知恵層)</b> (選択肢) ・プログラム実行中にパラメータを調整する。 (着眼点) ・クラスの内容を修正せずに、パラメータを修正できるようにする。 (品質考慮点5-1) (意圖) テスト作業の効率が向上する。</p>	<p><b>対策(知恵層)</b> (方法) ・メンテナンス操作を行うクラスを抽出する。 (アイデア) ・パラメータをLCD上で調節する。 ・不揮発メモリを利用して設定値を保存する。 (効果) ・実行中にパラメータ調整ができる。 ・起動時の操作が複雑になる。</p>
(a) 問題	(b) 課題	(c) 対策

図 9 経験者 A の思考過程 (知恵層)

Fig. 9 The thought process (wisdom layer) of expert A.

出力概念モデルは、クラス図における「メンテナンスクラス」をデータ層として「メンテナンス操作を行うクラスを抽出する」という対策を示している (図 9c) . この「メンテナンスクラス」は、「走行」クラスのパラメータをプログラム実行中に調整することができる . ただしこの対策には使用性を低下させるデメリットもあり、調整するパラメータ数を増やすたびに操作が複雑となる .

### (3) 思考概念モデルの記述結果

#### (a) 目標表現の思考概念モデルの記述結果

思考概念モデルは、「保守性を実現するクラス抽出」を行うために経験者 A が選択した品質考慮点表 (表 2) の 5-1 を着眼点としている .

経験者 A は、目標表現の品質実現オントロジを用いて対応付けを行った後、知恵層、知識層、情報層、データ層の順に思考概念モデルを記述した . 最初の対応付けの作業において経験者 A は、「パラメータを LCD パネル上で調節する」というアイデアを利用して「コンパイル後にプログラムを転送する組み込み機器である」という原因の解消を着眼点とした . この着眼点に該当する品質考慮点表の 5-1 を基に「保守性を実現するクラス抽出」を品質実現の課題として設定した (図 9b) . 次に、知識層に経験者 A が定義した品質考慮点と品質定義を記述し、情報層に「調査事実」を記述した .

#### (b) 設計意図表現の思考概念モデルの記述結果

設計意図表現の品質実現オントロジを用いて記述した思考概念モデルは、設計判断の展開による「メンテナンスのクラスを作る」と「パラメータのリアルタイム計算」の比較結果と、「メンテナンスのクラスを作る」ことを判断するためのルールを知恵層と知識層で表現している .

経験者 A は、設計意図表現の品質実現オントロジを用いて展開を行った後、目標表現の思考概念モデルの各階層を詳細に記述した . 最初に経験者 A は、問題の原因から「パラメータ調整の労力を抑える」に着眼点において、「パラメータを自動

計算するクラスを作る」という選択肢をあげた . この選択肢には、「走行テストの大部分を省略できる」というメリットがある代わりに、「走行中の処理効率を下げる」という特徴がある . このため、品質実現の課題の意義に、この選択肢との比較結果を記述している . 知恵層の記述結果を基に、メンテナンスを作るためのルールや事実を知識層に示した . さらに知恵層の記述結果を基にして情報層で経験者 A が得た調査事実とそれぞれの意味を記述し、データ層でクラス抽出の作業過程を記載した (図 10) .

## 6. 思考過程を用いるクラス抽出実験

### 6.1 目的

CAPIS/OOM のクラス抽出プロセスを用いて経験者 A の思考過程を別の対象システム分析者が利用することにより、思考過程の継承と創出が行えることを以下の順に検証する .

#### (1) 思考過程の継承実験

UML を用いるオブジェクト指向モデリングを 1 年間経験した 3 人の分析者 A, B, C が暖房システム (家庭にある各部屋の温度を集中管理するリアルタイムシステム) の分析において経験者 A の思考過程を用いて保守性を実現するクラス抽出が行えること、すなわち思考過程の継承を以下の手順で確認する .

##### (a) 思考過程を継承するクラス抽出

分析者 A, B, C は、経験者 A の思考過程を用いて保守性を実現するクラス抽出を行う .

##### (b) 分析者に対するインタビュー

各分析者がクラス抽出を行った後、インタビューを行い、経験者 A の思考過程を選択した理由から思考過程が各分析者に伝達していることを確認する .

#### (2) 思考過程の創出実験

経験者 A の思考過程を用いて新たに品質実現の課題を設定したクラス抽出を行い、各分析者が思考過程を創出できることを以下の手順で確認する .



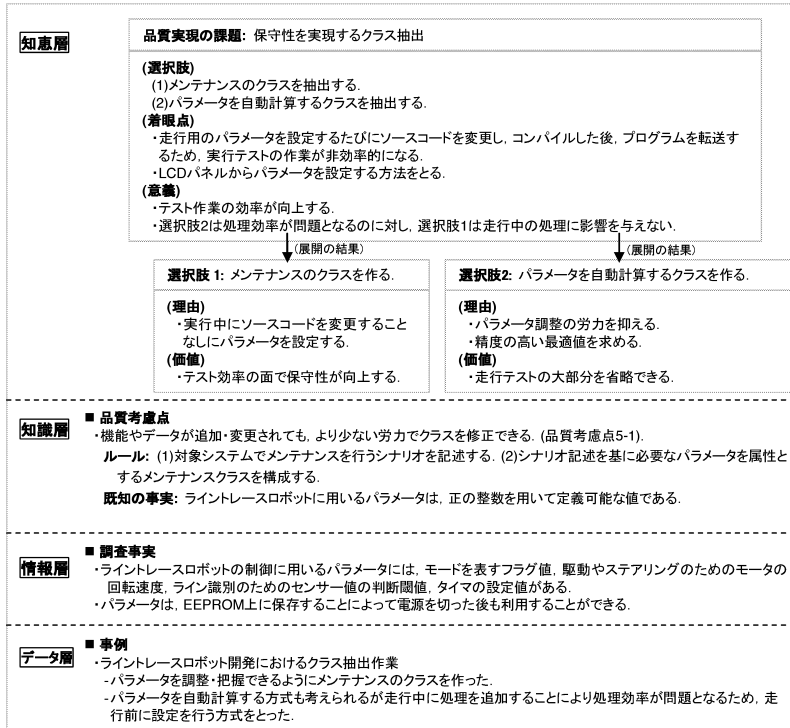


図 10 設計意図表現の品質実現オントロジを用いて記述した思考概念モデル

Fig. 10 The conceptual thinking model described by quality building ontology of design rationale view.

(a) 思考過程を創出するクラス抽出

経験者 A の思考過程を利用した分析者は、思考過程に品質実現の課題を新たに追加し、クラス抽出を行う。

(b) 分析者の思考過程の獲得

分析者は品質実現の課題と分析結果を基に創出した思考過程を獲得する。

6.2 思考過程の継承実験の結果

(1) 実験の結果

分析者 B と分析者 C は、経験者 A の思考過程を選択し、メンテナンスクラスを抽出した。

分析者 A は、保守性を実現するクラス抽出を行う前にメンテナンスの操作の一部を別のクラスで定義していたため、メンテナンスのクラスを作る経験者 A の思考過程を選択しなかった。

(2) 分析者に対するインタビューの結果

各分析者が思考過程を選択する理由や未選択の理由は、表 3 に示すとおりである。

また、品質実現オントロジを用いてクラス抽出プロセスを進める分析者 B と分析者 C の手順は、以下の点で共通する。

(a) 問題、課題、対策 (図 9) を参照することで思

表 3 思考過程の選択状況

Table 3 The cases of selection with thought processes.

分析者	選択状況	選択・未選択の理由
A	未選択	現在、直面している問題を解決できるような対策を含んでいるが、対策の内容には事前に作成したモデルと競合する部分があるため
B	選択	メンテナンスクラスは導入が簡単であること、テスト効率を上げることによる保守性が実現できるため
C	選択	既存のモデルを基に実装したプログラムは、様々な変数を変更する必要があり、デバッグに多くの時間を費やしてしまうため

考過程の利用を検討した。

(b) 思考過程を利用する場合、思考概念モデルで示されている選択肢、ルール、または、事例に従ってメンテナンスクラスを抽出した。

6.3 思考過程の創出実験の結果

(1) 実験の結果

経験者 A の思考過程を利用した分析者 B と分析者 C は、経験者 A が設定する「保守性」の課題に加え、「移植性」を実現するクラス抽出を行った。

分析者 B は、経験者 A の思考過程を用いて抽出したメンテナンスクラスの属性や操作がアプリケーションのデータ構造に依存しているため、対象システム

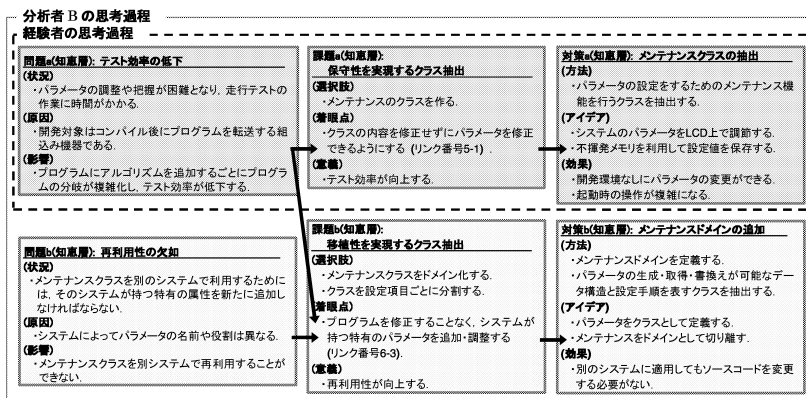


図 12 分析者 B による思考過程の創出結果 (知恵層)

Fig. 12 The result (wisdom layer) of thought process generation by system analyst B.

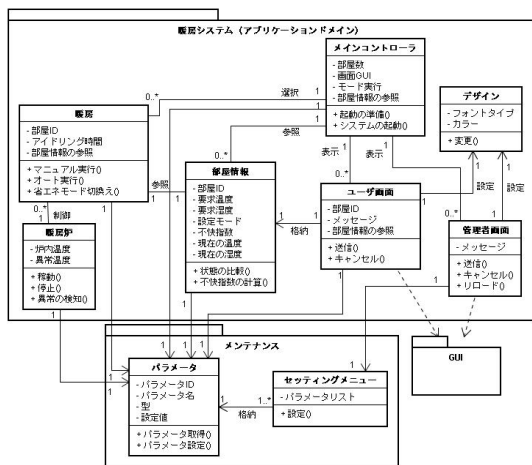


図 11 分析者 B のクラス図 (メンテナンスドメインに関する部分)  
Fig. 11 The class diagram of system analyst B (the part concerning maintenance domain).

においてこのクラスを再利用することが困難であると考えた。分析者 B は、この点を問題として移植性を実現するクラス抽出を行うため、品質考慮点表の 6-3 (クラスは修正することなしに別のシステムにも利用できる) を品質実現の課題とした。クラス抽出の結果、メンテナンスクラスをドメインとして定義 (ドメイン化) して「セッティングメニュー」クラスと「パラメータ」クラスを構成した (図 11)。分析者 C も、抽出したクラス名と属性、関係は異なるが、分析者 B と同様にメンテナンスクラスをドメイン化し、そのドメインの中でパラメータのデータ構造を表すクラスを抽出した。

(2) 思考過程の獲得の結果

経験者 A の思考過程を基に創出した分析者 B の思考過程の獲得を行った結果を以下に示す。

(a) 入力概念モデルの記述結果

入力概念モデルは、データ層にメンテナンスクラスを記載し、知恵層で「再利用性の欠如」の問題を示している (図 12 の問題 b)。

(b) 出力概念モデルの記述結果

出力概念モデルは、メンテナンスドメインを構成する「セッティングメニュー」クラスと「パラメータ」クラスを抽出することを対策としている (図 12 の対策 b)。

(c) 思考概念モデルの記述結果

(a) 対応付け

分析者 B は、「システムによってパラメータの名前や役割は異なる」という原因の解消法を着眼点とした。さらに、経験者 A が示した「開発対象はコンパイル後にプログラムを転送する組み込み機器である」ことも原因とし、問題と課題を関連付けた (図 12)。

(b) 展開

分析者 B は、「テスト効率の低下」と「再利用性の欠如」の問題を解決するために、「移植性を実現するクラス抽出」を課題として記述した。この課題を解決する選択肢として、メンテナンスクラスをドメイン化することを記述した。また、メンテナンスクラスのデータ構造に着目し、「クラスを設定項目ごとに分割する」という選択肢を追加した。この選択肢は、当初考案した方法であるがクラス数とコード量が扱えきれないほどに増加するため最終的に破棄されている。

(c) 設計意図表現の思考概念モデルの記述

分析者 B は、設計意図表現の品質実現オントロジに従い知恵層の選択肢を基に品質考慮点を知識層に記述した。情報層においては「パラメー



図 13 分析者 B が記述した思考概念モデル

Fig. 13 The conceptual thinking model described by system analyst B.

タを追加するたびにメンテナンスクラスの属性や操作を何度も変更した」等のように分析者 B が得た調査事実を記述した。データ層においては、品質実現の事例としてライントレースロボットの開発過程を記載している (図 13)。

## 7. 考 察

### (1) 思考過程の獲得について

保守性を実現する経験者 A の思考過程は、設計意図表現の品質実現オントロジが示す記述項目に従い品質実現の課題、品質考慮点を思考概念モデルで示している。品質実現オントロジに基づく思考概念モデルは、知恵層や知識層が少ない記述量であるが、知恵層の選択肢と知識層の品質考慮点とそのルール等の記述項目間の関係を示していることから、品質実現の課題を正確に記述することに有用であると考えられる。

また、本方式は経験者の思考過程だけでなく分析者の思考過程も獲得できるクラス抽出プロセスだといえる。実験により獲得した思考過程は 4 つであり、その中の 1 番目から 3 番目は経験者 A の思考過程であり、4 番目の思考過程は分析者が創出し、獲得した思考過程である (表 4)。

表 4 実験で獲得した思考過程

Table 4 The thought process captured in this experiments.

番号	品質実現の課題(経験者の定義)	獲得した思考過程
1	機能性: 走行タイムを短縮する機能が備わっている (品質考慮点 1-2)	ライントレースロボットのドルフィンジャンプ(コースショートカット法)クラスの抽出
2	信頼性: コースから脱線した場合、復帰することができる (品質考慮点 2-3)	ライントレースロボットのコース復帰クラスの抽出
3	保守性: ハードウェアからパラメータを変更可能にする (品質考慮点 5)	ライントレースロボットのメンテナンスクラスの抽出
4	移植性: クラスは修正することなしに別のシステムにも利用できる (品質考慮点 6-3)	家庭用暖房システムのメンテナンスクラスの追加

### (2) 思考過程の継承について

各分析者は、思考過程の選択を知恵層の内容で判断してクラス抽出を行い、問題の状況、課題が示す品質特性、対策の効果をあげて選択理由を説明している。このことを確認するために筆者らが思考過程の比較を行った結果から、思考過程の選択は経験者 A の思考過程の知恵層を以下のように比較していることが明らかとなった。

#### (a) 問題に関する思考過程の比較

分析者 A と分析者 C は、クラス図を作成する分析者自身が直面している問題と経験者 A の思考過程における問題との比較結果を理由にあげている。

#### (b) 課題に関する思考過程の比較

分析者 B は、課題の「意義」を用いて思考過程を

利用した理由を示し、「テスト効率を上げることによる保守性の実現」を説明に用いている。

### (c) 対策に関する思考過程の比較

分析者 A は、「既存のモデルと競合する部分がある」のように、対策の「方法」を分析者自身のクラス図への適用を検討した結果を示している。各分析者が思考過程を利用する手順は、目標表現の品質実現オントロジに従って最初に思考過程を選択した後に、設計意図表現の品質実現オントロジに従って各分析者が思考過程の中を参照する順となっている。経験者 A の思考過程を選択した分析者 B と分析者 C は、設計意図表現の品質実現オントロジを用いて思考概念モデルにおける知恵層の選択肢、知識層のルールを参照し、「メンテナンス」クラスを構成している。以上から、本方式のクラス抽出プロセスは 3 人の分析者が思考過程を継承してクラス抽出を行うことに有用であると考えられる。また、思考過程を利用する分析者は、品質考慮点表から品質実現の課題を選択することによって、品質考慮点の番号に該当する思考過程を特定する。本実験の保守性を目標とする思考過程は 1 つであったが、品質考慮点表の品質考慮点に他の経験者の思考過程が加わり、選択すべき思考過程が増える場合がある。その場合、思考過程を利用する分析者は各思考過程の品質実現の課題に記述される選択肢の「理由」「価値」を参照し、それぞれの特徴を理解することにより思考過程の選択することができる。

### (3) 思考過程の創出について

思考過程の課題を展開してメンテナンスクラスをドメイン化したことは、経験者 A の思考過程が複数の品質実現の課題を解決するクラス抽出に有用であると考えられる。すなわち経験者 A の思考過程を基に創出した分析者 B の思考過程は、分析者 B が直面する問題が保守性を実現する思考過程で解決しきれない問題を新たにあげ、「移植性を実現するクラス抽出」にそれぞれの問題を関連付けている。

## 8. 関連研究

オブジェクト指向分析で行われるクラス抽出法には、要求仕様書やユースケース記述から名詞を抜き出す方法（名詞抽出）や CRC カード法<sup>11)</sup>、構造化手法と組み合わせる機能分割を行う FCD<sup>12)</sup> がある。また、有形物、役割、出来事、仕様、相互作用の分類<sup>3)</sup> や、インタフェース、エンティティ、コントロールの分類<sup>13)</sup> を用いてユースケース記述からクラスを識別する方法もある。これらのクラス抽出法は機能性を実現するこ

とに注目する方法であり、機能性以外の信頼性や使用性等を考慮してクラスを抽出する手段を有していない。筆者らが提案した分解、統合の 2 段階に分けて品質特性を考慮するクラス抽出法<sup>5)</sup> においても品質特性を実現する思考過程の内容が分析者の設計意図に依存するという問題がある。このため筆者らは、品質特性を効率良く作り込むには経験者の思考過程を利用できるようにすることが重要だと考え、CAPIS モデルに基づくクラス抽出方式を開発している。本方式は、(1) 品質特性を満たす条件を機能性、信頼性、使用性、効率性、保守性、移植性までの順序で実現できるようにクラスを抽出している点、(2) 経験者のクラス抽出思考過程を利用している点で CRC カード法や FCD 等上記のクラス抽出手法と異なる。

來村の機能オントロジ<sup>14)</sup> は、大規模システムの機能分解においてクラス候補の識別を容易にする知識を提供するオントロジである。これに対し、本方式の品質実現オントロジは品質特性を考慮するクラス抽出法の知識を提供し、加えて、知識を形成する土台としての知恵を表している。すなわち、品質実現オントロジはナレッジ階層の構造に基づいて構築するため、知識を形成する経験者の意図や視点を示すことができるとともに、知識の必要性を利用者が理解することに有用であると考えられる。

## 9. おわりに

本論文では、CAPIS モデルと品質実現オントロジを用いて思考過程を継承し、創出してクラスを抽出する CAPIS/OOM について述べた。本方式により品質特性を実現する経験者の思考過程が獲得できること、他の分析者が思考過程を継承できることを実験で確認した。さらに、経験者の思考過程を用いて新たな思考過程を創出する分析者の思考過程を獲得できることを明らかにした。これらの結果から、CAPIS/OOM により分析者が経験者の思考過程を用いて複数の品質特性を実現するクラス抽出が行えることを検証した。

今後の課題として、獲得した思考過程をインターネット上で参照できるようにするため、オントロジ技術に基づいてクラス抽出の思考過程獲得を支援するシステムを開発する。

謝辞 本研究は、日本大学学術研究助成金を受けて実施しました。

## 参考文献

- 1) 日本規格協会：JIS X 0129-1 (ISO/IEC9126-1), JIS ハンドブック 66-1 ソフトウェア (2005).

- 2) Selic, B.: The Pragmatics of Model-Driven Development, *IEEE Software*, Vol.20, No.5, pp.19–25 (2003).
- 3) Mellor, S.J. and Balcer, M.J.: *Executable UML*, Addison-Wesley (2002). 二上貴夫, 長瀬嘉秀 (監訳): Executable UML, 翔泳社 (2003).
- 4) Wirfs-Brock, R.: Looking for Powerful Abstractions, *IEEE Software*, Vol.23, No.1, pp.13–15 (2006).
- 5) 大山勝徳, 武内 惇, 藤本 洋: オブジェクト指向分析における品質特性を考慮したクラス抽出法, 日本大学工学部紀要, Vol.47, No.2, pp.103–111 (2006).
- 6) 大山勝徳, 武内 惇, 藤本 洋: CAPIS モデル方式による設計思考過程の表現法, 情報処理学会論文誌, Vol.47, No.8, pp.136–139 (2006).
- 7) Oyama, K., Takeuchi, A. and Fujimoto, H.: CAPIS Model Based Software Design Method for Sharing Experts' Thought Processes, *Proc. 30th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC 2006)*, pp.307–314 (2006).
- 8) Ackoff, R.L.: From Data to Wisdom, *Journal of Applied Systems Analysis*, Vol.16, pp.3–9 (1989).
- 9) Bellinger, G., Castro, D. and Mills, A.: Data, Information, Knowledge & Wisdom. <http://www.systems-thinking.org/dikw/dikw.htm>
- 10) ET ソフトウェアデザインロボットコンテスト. <http://www.etrobo.jp/>
- 11) Wirfs-Brock, R.: Designing Objects and Their Interactions: A Brief Look at Responsibility-Driven Design, *Scenario-Based Design: Envisioning the Work and Technology in System Development*, Carroll, J.M. (Ed.), John Wiley & Sons (1995).
- 12) Chang, C., Cleland-Huang, J., Hua, S. and Kuntzmann-Combelles, A.: Function-Class Decomposition: A Hybrid Software Engineering Method, *IEEE Software*, Vol.34, No.12, pp.87–93 (2001).
- 13) Jacobson, I., Christerson, M., Jonsson, P. and Overgaard, G.: *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley (1992). 西岡利博ほか (訳): オブジェクト指向ソフトウェア工学 OOSE, トッパン (1995).
- 14) 來村徳信, 笠井俊信, 吉川真理子, 高橋 賢, 古崎晃司, 溝口理一郎: 機能オントロジーに基づく機能的知識の体系的記述とその機能構造設計における利用, 人工知能学会論文誌, Vol.17, No.1, pp.73–84 (2002).

(平成 18 年 7 月 31 日受付)

(平成 19 年 5 月 9 日採録)



大山 勝徳 (正会員)

昭和 53 年生。平成 15 年日本大学大学院工学研究科情報工学専攻博士前期課程修了。現在、日本大学大学院工学研究科情報工学専攻博士後期課程在学中。思考過程支援, MDA, Semantic Web に興味を持つ。



武内 惇 (正会員)

昭和 21 年生。昭和 45 年日本大学理工学部電気工学科卒業。昭和 47 年北海道大学大学院電子工学専攻修了。同年より沖電気工業 (株) に入社。以来、ソフトウェア生産技術の開発に従事。平成 4 年より日本大学工学部情報工学科助教授。平成 18 年より日本大学工学部情報工学科教授。博士 (工学)。要求工学, オブジェクト指向技術, 概念処理に興味を持つ。電子情報通信学会, 人工知能学会, IEEE 各会員。



藤本 洋 (正会員)

昭和 16 年生。昭和 40 年日本大学理工学部電気工学科卒業。同年より東京大学生産技術研究所に勤務。昭和 44 年富士通 (株) に入社。以来、通信システム (特に電子交換システム) のソフトウェア開発に従事。平成 7 年より日本大学工学部情報工学科教授。博士 (工学)。組織体モデル, マルチエージェントシステム, オブジェクト指向技術, コデザイン技術, および概念処理に興味を持つ。電子情報通信学会, IEEE 各会員。