

# CAPIS モデル方式による設計思考過程の表現法

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

経験者の設計判断を設計者がだれでも共有して使用できるように、経験者の思考過程表現の手法を開発している。設計判断の結果が必ず設計結果に反映されることに着目し、本論文では複雑性や多様性のある思考過程を「問題」「課題」「対策」に分けて表現する CAPIS (CAusality of Problem-Issue-Solution) モデル方式を提案する。CAPIS モデル方式の適用実験として、オブジェクト指向モデリングを経験したことのある経験者 4 人によって表現された思考過程を考察した。その結果、クラス図やメモの記述結果で経験者に言及されなかった常識・専門知識、設計意図等が表現され、経験者の思考過程が表現されたことを確認した。

## The Expression of Design Thought Processes Using CAPIS Model

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

We develop an expression of design thought processes so that designers can share and apply the experts' design decisions. Because it is difficult to expose thought processes with complexity and diversity that includes repetition of trial and error, we focus that all design decisions reflects a design consequence. In this paper, we present the CAPIS (CAusality of Problem-Issue-Solution) model method that illustrates a thought process in parts of "Problems", "Issues", "Solutions". Furthermore, we discuss that an expression of thought processes by four experts that uses Object Oriented Modeling. In this result, we validated this thought process explains common sense, domain knowledge, design rationale, which are not addressed by class diagrams and memo.

### 1. はじめに

ソフトウェア開発は性能と品質の面でますます複雑な条件を満足させることが求められる。多くの機能と高い品質を備えたソフトウェアを短期間で開発するには、設計者が設計技術を効果的に使用する必要がある。筆者らは、設計技術を効果的に使用できる技術者（以降、経験者と呼ぶ）による設計時の判断（設計判断）を表現し、かつ、他の技術者（以降、設計者と呼ぶ）が経験者と同様に設計技術を使用できるようにする方法の研究を進めている。経験者から設計者への設計判断を説明する場合、経験者が持つ常識・専門知識、設計意図（特に設計判断を採用した理由もしくは破棄した理由）は文章表現が難しい暗黙知<sup>1)</sup>であることが多く、設計者は必ずしも経験者の設計判断を理解できる

とは限らない。経験者の設計判断を設計者が同様に使用できるようにするため、本研究では経験者が持つ常識・専門知識、設計意図を明示する思考過程の表現法を開発する。

思考過程表現の難しさの要因は、以下の多様性と複雑性にあると考える。

#### (1) 思考過程の複雑性による表現の難しさ

設計者の思考には、設計対象となる機能やそれを実現する構造を仮決定し、決定内容を評価し、さらに仮決定し、最終的に決定を行うという試行錯誤の工程がある<sup>2)</sup>。設計結果を決定するまでの試行錯誤が増えることにより、思考過程は複雑になる。特に、ソフトウェアには構造の自由度があり、かつ、構成要素（プログラムモジュール）間に関数呼び出し等の依存関係がある。また、設計結果が仕様を満たしていることを確認するには、構成要素がプログラムとして実行可能でなければならない場合が多い。このため、特にソフトウェアの設計では構成要素を決定するまでに試行錯誤が多く、思考過程をすべて表現することは困難である。

<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

## (2) 思考過程の多様性による表現の難しさ

設計作業の進め方は独自の開発環境や経験に基づく知識に依存するため、問題のとらえ方とそれを解決するための設計判断も異なる。設計者がいろいろな設計判断を行うことにより、思考過程には多様性がある。このため、設計判断を単純にドキュメント化して説明するだけでは、設計者が経験者に教えられた設計判断を熟知して設計を進めることはできない。すなわち、本当は有用であるにもかかわらず、経験者が提示した思考過程を設計者は理解して選択できないことがある。

思考過程表現の例として、ソフトウェア開発によく現れる問題とすでに実効性が確認されている解決法を表したソフトウェアパターン<sup>3)</sup>がある。ソフトウェアパターンから類推して分析モデルや設計モデルを構成することが有効な場合もあればそうでない場合もあり、見極めが必要である。問題の背景や適用の意図を理解していないと、実際の設計対象に具体化することが困難になるだけでなく、同じソフトウェアパターンを開発の全過程に適用することによって打ち出の小槌アンチパターン<sup>4)</sup>に設計者が陥る場合もある。

複雑性や多様性を持つ思考過程をすべて表現することが困難であることから、経験者が設計判断を決定する「問題解決の過程」を分割し簡潔に表現できる手法が必要である。筆者らは、思考過程表現の難しさを解決するため、設計者の設計判断を想起する CAPIS (CAusality of Problem-Issue-Solution) モデル方式を提案する。この方式では、設計判断の結果が必ず設計結果に反映されることに着目し、思考過程を「問題」「課題」「対策」に分けてモデル化することによって、試行錯誤のある思考過程を表現する。すなわち、設計判断を顕在化して表現できるようにするため、以下に示す「ナレッジ階層に基づく概念化プロセス」と「PIS (Problem-Issue-Solution) テンプレートを用いた対応付けと展開」で CAPIS モデル方式を定式化する。

## (a) ナレッジ階層に基づく概念化プロセス

CAPIS モデルでは、思考過程を3つの概念モデルで構成する。このための概念モデルをデータ、情報、知識、知恵の階層(ナレッジ階層)に従って記述し、文章上で明確にされなかった設計者の常識・専門知識や設計意図を表す。ナレッジ階層を記述する際、概念化プロセス(意味付け、分類、正当化、導出、説明、記号化)を用いて設計判断をデータから情報、知識、知恵までの階層関係に基づいて理解できるように

し、思考過程の多様性による表現の難しさを解決する。

## (b) 設計判断の想起法

CAPIS モデルを構成する概念モデルの中でも、「思考概念モデル」を記述するためのデータ層となる設計ドキュメントが設計後に残されることは少ない。思考概念モデルを記述する手段として、問題・課題・対策を表すテンプレート(PIS テンプレート)を用いる。PIS テンプレートを用いることにより、入力概念モデルと出力概念モデルの因果関係から設計判断を設計者に想起させて、思考過程の複雑性による表現の難しさを解決する。

本論文では、2章で CAPIS モデルの考え方について、3章でナレッジ階層に基づく概念化プロセスについて、4章で PIS テンプレートを用いた対応付けと展開による思考概念モデルの記述方法について説明する。5章で CAPIS モデル方式を用いて思考過程を表現する手順を示し、さらに、6章でオブジェクト指向分析・設計を事例として本方式を用いて思考過程が表現できることを確認する実験について述べる。7章で思考過程や概念化の関連研究と本手法との比較を行う。8章で結論と今後の課題を述べる。

## 2. 思考過程を表現する CAPIS モデル

## 2.1 考え方

設計作業は、設計要件(入力の仕様)を解釈して具体的な設計結果(出力の仕様)を定義するまでの手順を構成する。各手順において思考過程は、入力の仕様を解釈し、問題を発見し、次に、その問題を解決するための選択肢を想起し、試行錯誤を繰り返す知的活動である。試行錯誤の中で、問題の解決策(対策)を決定するために採用される選択肢を設計判断と呼ぶ(図1)。設計判断を実行した結果は、出力の仕様に具体化される。

試行錯誤のある思考過程には複雑性があり、すべて表現することが困難であることから、「問題解決の過程」を分割表現することで思考過程をモデル化する。思考過程を「問題」と「解決」の2つで構成する場合、設計意図による対策の取捨選択を表現することが難しい。思考過程を表現するには、問題の原因に対する着眼点とそれを解決するために用いるアイデアとの関係から問題を解決する道筋を表す「課題」が重要である。「課題」は問題を解決するために設計者が選んだ部分問題である。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。

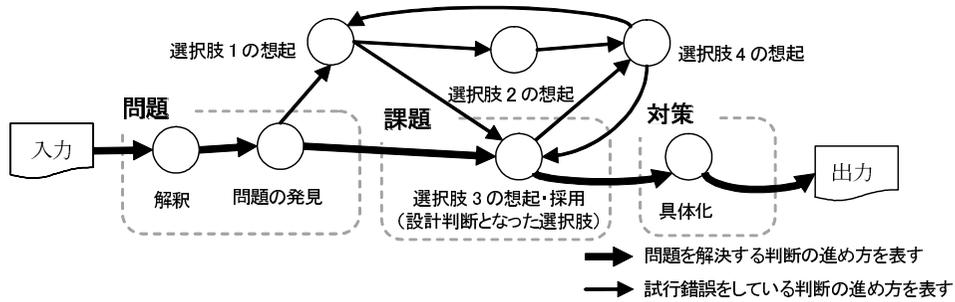


図 1 設計作業における複雑な思考過程

Fig.1 Complex thought processes on design task.

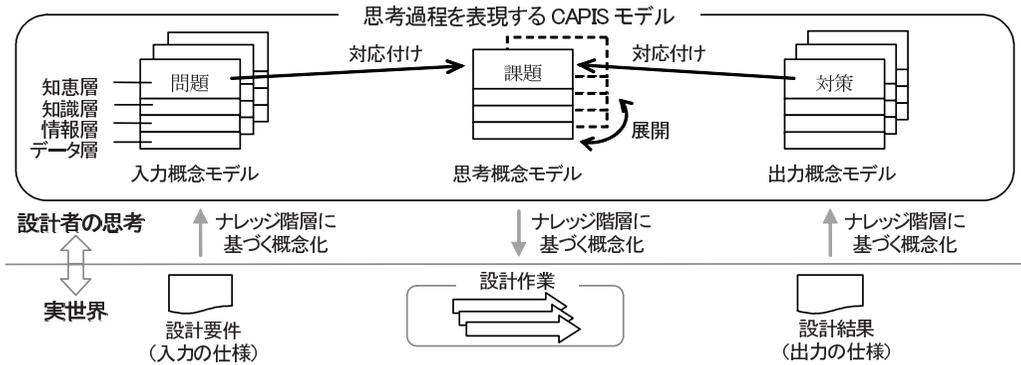


図 2 CAPIS モデル

Fig.2 CAPIS Model.

2.2 問題・課題・対策の因果関係

CAPIS モデル(図 2)は、問題(入力概念モデル)、課題(思考概念モデル)、対策(出力概念モデル)で構成される思考過程の表現である。

設計作業の入出力情報は、作業後に設計ドキュメントとして実在しているため、後述のナレッジ階層に基づく概念化を用いて入力概念モデルと出力概念モデル(以降、両者をあわせて入出力概念モデルと呼ぶ)を記述することができる。すなわち、設計ドキュメントの内容を入出力概念モデルのデータ層として記述できる。

しかしながら、設計判断は設計ドキュメントとして残されることが少ないため、思考概念モデルのデータ層が十分に記述できない。そのことから、思考概念モデルをナレッジ階層に基づく概念化を用いて記述することが難しい。設計判断が設計ドキュメントとして残されることが少ない理由は、(1)設計時にメモやスケッチをしながら思考過程を記録した場合に作業量が増えるだけでなく思考が中断すること、また、(2)設計作業後の設計判断の記述に必要とされるデータが不十分であることにある。

思考概念モデルを記述できるようにするため、問題を「思考の原因」とし、対策を「思考の結果」として

設計判断を設計者に想起させる方法を「対応付け」と「展開」の操作で定式化する。CAPIS モデル上の思考概念モデルを記述する段階で、図 2 における問題と対策との関係から設計判断を表出する操作を対応付けと呼ぶ。また、設計判断が採用された理由、または、破棄された理由を示すために、設計判断を決定するまでの選択肢を比較することにより別の選択肢も表出して課題を構成する操作を展開と呼ぶ。

PIS テンプレートは、対応付けと展開が行えるようにモデル化した問題・課題・対策の因果関係を表す。PIS テンプレートを用いて設計判断の「対応付け」と「展開」を行う方法を 4 章で述べる。

2.3 ナレッジ階層に基づく概念化による思考過程表現

2.3.1 ナレッジ階層

人間が持つ思考の内容を分類するために Ackoff<sup>5)</sup> はデータ、情報、知識、理解、知恵を定義した。その中で「理解」は、知識の内容を知恵へと昇華させることだけでなく、データの内容から意味を判別して情報を得る行為でもある。その点で Bellinger ら<sup>6)</sup> は、データから情報、知識、知恵への遷移が「理解」であると考え、データ、情報、知識、知恵からなる階層(ナレッジ階

表 1 設計思考過程を表現対象とするナレッジ階層  
Table 1 The knowledge hierarchy for design thought processes.

ナレッジ階層	定義	入出力概念モデル	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>問題の状況<sup>*</sup>とそれが発生する原因<sup>*</sup>、設計者や設計対象に与える影響<sup>*</sup></li> <li>対策を実現する方法<sup>*</sup>と、その方法に基づくアイデア<sup>*</sup>、対策が持つ効果<sup>*</sup></li> </ul>	<ul style="list-style-type: none"> <li>課題を解決するため選択肢<sup>*</sup>とその着眼点<sup>*</sup>、課題を解決する意義<sup>*</sup></li> <li>課題解決のために選択された設計判断と、それに関わる設計意図</li> </ul>
知識	常識や専門知識を表す命題（ルールと事実で表される）	<ul style="list-style-type: none"> <li>システム構成要素に関する問題もしくは対策を表す事実</li> <li>システム構成要素を補足するための常識や専門知識</li> </ul>	<ul style="list-style-type: none"> <li>設計手順</li> <li>設計技術の使い方</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>データ層で示された用語の意味</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>設計ドキュメントに記述されたシステム構成要素</li> </ul>	<ul style="list-style-type: none"> <li>マニュアルや指導書として表現された設計技術</li> </ul>

※4.2.2に詳述

層)で思考の内容を分類した。概念化を行うにあたっては、設計判断を明らかにする以前にデータ層を出発点とする知恵層までの各階層に対して「理解」を行う必要があると考える。このことから、筆者らは Bellinger のナレッジ階層に基づき、表 1 のように CAPIS モデルの概念モデルをデータ、情報、知識、知恵の階層に分けて記述する。

多様性を持つ設計判断を設計者が容易に理解できるようにするには、設計結果だけでなくそれを作成するための設計意図、常識・専門知識、用語（記号）の意味まで明らかである必要がある。これらの項目は設計者から見て階層関係を構成しており、概念化に利用することができる。すなわち、データ層には設計ドキュメント上に記述されたシステム構成要素を表す記号を示す。情報層には、データ層で記述された記号の意味を説明する。知識層には、情報層における意味の説明で用いられた設計者の常識・専門知識を示す。知恵層には、知識層における常識・専門知識を適用する設計意図となる設計者の判断を示す。

特に、知恵層には知識層で説明されている命題の良否、または、真偽を決定するまでに行われた 1 つ 1 つの判断とその過程を示す。CAPIS モデルでは、経験者が正当化した設計判断を提示することを目的とし、知恵層を「判断」「理由」「価値」の項目で表現する。「判断」が判断の過程とその結果を表すのに対し、「理由」は判断が思考された原因や判断の根拠である。「判断」とその「理由」の 2 項目で経験者の設計判断を提示した場合、判断の正しさは主観的な事実であり形式的には表現しきれない。設計の思考過程は判断の選択肢を比較説明することが重要であることから、「判断」「理由」に加えて「価値」を併記する。このときの「価値」は判断の評価とその比較結果、または、設計者の

主観的な基準を表す。

### 2.3.2 ナレッジ階層に基づく概念化と概念化プロセス

筆者らはデータ、情報、知識、知恵からなる階層（ナレッジ階層）に注目し、ナレッジ階層に基づく概念化を考案する<sup>7)</sup>。すなわち、設計ドキュメント上の記号が表す意味の構造を示すこと<sup>8)</sup>を概念化とするだけでなく、意味の構造をデータ、情報、知識、知恵からなるナレッジ階層として記述する。ナレッジ階層は一度に詳細まで概念化の作業によって表現できるものではないため、設計ドキュメントから設計者の知識や知恵を段階的に記述できる概念化プロセスが必要となる。

## 3. ナレッジ階層に基づく概念化プロセス

### 3.1 CAPIS モデル方式における概念化

設計作業の入出力となった仕様をデータ層とし、ナレッジ階層に従って入出力概念モデルを記述する。データ層を基にして知恵層までの階層関係を持った概念モデルを効率良く記述することができるように、CAPIS モデル方式では概念化プロセスを定義する。入出力概念モデルの記述では、次節のように意味付けから概念化プロセスを進める。

### 3.2 概念化プロセス

概念化プロセスは、ナレッジ階層に基づいて理解のプロセス（意味付け、分類、正当化）と表現のプロセス（導入、説明、記号化）で構成する（図 3）。

設計ドキュメントに記述された主題（問題、課題、対策を表す概念化の対象）を明確にする意味付け、分類、正当化を理解のプロセスと呼ぶ。記号に対する意味付けによって、知識層で用いる概念を明らかにする。次に、概念を分類することで、知恵層で正当化すべき命題を明らかにする。

表 2 概念化のプロセス

Table 2 Conceptualization processes.

概念化プロセス	定義	目標	
理解	(1)意味付け	ミーニングトライアングル (記号, 事物, 概念) に基づいて文章中に出てきた記号に対し, 対応する事物を示す. 設計者が意味を一意に解釈できるように, 予め決められた規則 (専門分野によって異なる記述項目, 説明方法など) を用いて説明する.	記号が表す意味を理解し, かつ, その意味について設計者自身は説明できる.
	(2)分類	意味付けされた記号が表す概念を主体とする命題を列挙する, その命題に関連する常識や専門知識を記述する. このときの命題とは, 真偽が問える事実またはルールである.	列挙された命題に, 概念化の対象 (主題) が含まれている. また, それに関連した常識, 専門知識を記述している.
	(3)正当化	文章中から概念化の対象 (主題) を見つける. その内容を判断, 理由, 価値として抽出するとともに主題の正しさを説明する.	判断の正しさを説明できるように, 理由と価値を明らかにしている.
表現	(4)導出	正当化の結果, 真だとされた判断の内容 (命題) を結論として記述する. さらに, 結論に関連した項目を説明する. 各命題を詳細化した結果, 矛盾や不足があったときは修正を行う.	結論に関連した各命題は, 第三者が利用できるように詳細となっている.
	(5)説明	正当化の過程で新たに記述された用語に対応する事物を示す.	正当化に用いられた記号が表す意味を設計者が説明できる.
	(6)記号化	概念化によって新たに必要となった記号を追加し, 正当化された主題の文章表現を改善する. それと同時に, 伝える対象 (設計者), 手段 (言語, メディア) を決定する.	記述された内容が設計者にとって可読な形式である.

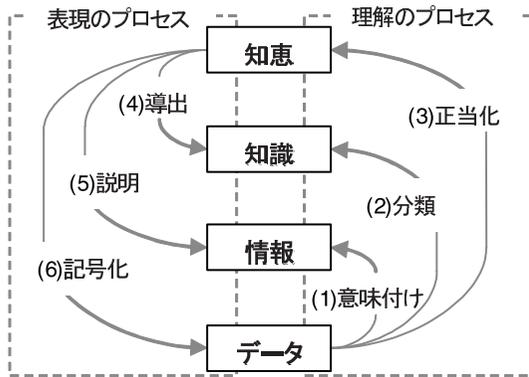


図 3 ナレッジ階層に基づく概念化プロセス

Fig. 3 Knowledge hierarchy based conceptualization processes.

また, 知恵層にある判断に基づいた導出, 説明, 記号化を表現のプロセスと呼ぶ. 主題の「表現」を効率良く行うために, まず結論となった判断の結果を導出する. 次に結論を正確に表すために意味付けがなされた用語を用いて説明を行い, それを設計者にとって可読な形式に記号化する.

データを構成する記号は, 情報層においては意味付けに利用される. 意味付けされた記号は, 知識層においては命題の分類に利用される. 各階層が階層関係を持った概念モデルを設計者が効率良く記述することができるように, 概念化のプロセスを定義する (表 2).

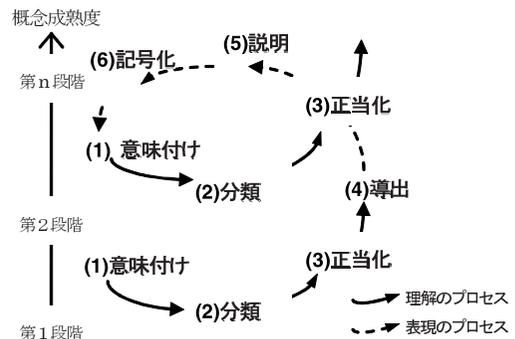


図 4 概念の成熟プロセス

Fig. 4 Maturation process of concepts.

### 3.3 概念の成熟プロセス

概念化プロセスに従い概念モデルを記述しても, 最初に設計者によって記述されたデータ, 情報, 知識, 知恵は主題を正確に表していないことがある. たとえば, データ層に記述された仕様は, すべてが問題を表しているとは限らず, 問題と関係していない部分を持つ可能性がある. そのため, 意味付けから記号化までの 1 回のサイクルだけで概念化を行うのではなく, 段階的に概念化を進めることで概念成熟度を向上させる. 表 2 に示した概念化プロセスの目標を達成させるため, 概念モデルに新たな修正を加える必要がなくなるまで概念化プロセスを繰り返す (図 4). 概念化プロ

セスを繰り返し、主題を正確に記述するまでの手順を概念の成熟プロセスと呼ぶ。

以上の方法で概念化することにより、入出力概念モデルで提示すべき問題と対策を正確に表現する。

#### 4. 設計判断の想起法

複雑な思考過程における設計判断の想起法を定式化するため、CAPIS モデル方式では次の 2 つの方法を用いる。

- (1) 問題と対策との因果関係に基づいた設計判断を想起する対応付けと展開の方法
  - (2) PIS テンプレート (Problem-Issue-Solution) を用いる設計判断の想起法
- 以下、各々について説明する。

##### 4.1 対応付けと展開による設計判断の想起法の考え方

設計判断を見つけるため、思考過程を問題・課題・対策ととらえて課題を明らかにする。さらに、設計判断を想起するには、課題の中の着眼点を明らかにする必要がある。設計者は、着眼点を以下の思考過程で用いる。

- (1) 問題の「原因」に注目して課題を設定するために問題の解決策の選択肢をあげる。この場合、選択肢をあげる理由、すなわち、着眼点は問題の原因を解消する方針を表す。問題の解決が可能な選択肢の中から設計判断を決定する。
- (2) 解消すべき原因が多いことにより選択肢の決定が難しくなる場合、対策の「アイデア」を発想し、それに基づいた方法を試行する。結果を評価することによって「アイデア」に有効性がないと判断した場合、問題の「原因」に対する「着眼点」を変更する。このように、着眼点は設計者に設計判断を想起させる重要な項目であるが、着眼点を文章で表現することは難しい。理由は、設計者の着眼点はいまいなイメージにとどまっている場合が多く、スケッチやメモが文章化の材料として記述されることが少ないことによる。

このため、CAPIS モデルにおける思考概念モデルの記述では、データ層を最初に記述するのではなく、問題と対策との因果関係を分析し、課題を表出する過程で着眼点を見つけることによって設計判断を想起する。これを実現するために以下の対応付けと展開を行う。

- (1) 問題の原因と対策のアイデアの因果関係から着眼点を見つけ、設計判断を想起する操作を「対応付け」と呼ぶ。
- (2) 「対応付け」を行った後に、設計判断を決定す

るまでの選択肢も想起し、それらを比較することにより、設計判断に選択肢が採用された理由、破棄された理由を示すために一時的な選択肢を表示する操作を「展開」と呼ぶ。

##### 4.2 設計判断の想起法の定式化

対応付けと展開における PIS テンプレートの必要性について述べ、その構成法を以下に示す。

##### 4.2.1 PIS テンプレートを用いる対応付けと展開の定式化

設計判断の想起法を定式化するために、問題・課題・対策のナレッジ階層の知恵層の因果関係に注目して対応付けと展開を決定する。しかし、これらの知恵層は問題・課題・対策のナレッジ階層の知恵層の判断・理由・価値を用いて表現するため、以下の点を解決する必要がある。

- (1) 設計者の思考によってはその判断・理由・価値の表現が抽象的になりやすい。
- (2) 問題と課題の違いを混同して表現することがある。

これらを解決するために PIS テンプレートを用いる。すなわち、PIS テンプレートを用いて問題のナレッジ階層の知恵層の「原因」、課題のナレッジ階層の知恵層の「着眼点」、対策のナレッジ階層の知恵層の「アイデア」に注目して対応付けと展開の方法を定式化する。

ナレッジ階層の知恵層を構成する項目を用いて問題・課題・対策を以下のとおり表現する (表 3)。

- (a) 問題・課題・対策の知恵層を判断・理由・価値という項目の組合せで表現する。
- (b) 問題の原因、課題の着眼点、対策のアイデアを因果関係で結ぶ。
- (c) 着眼点から選択肢や意義を想起する。

すなわち、PIS テンプレート (図 5) を用いた対応付けでは、問題の「原因」を選び、さらに、それを改善できる対策の「アイデア」を選ぶ。「原因」と「アイデア」の組によって「着眼点」を設定する。次に、実際の設計作業で思考した選択肢をあげる。このときの

表 3 知恵層を構成する項目  
Table 3 Items of wisdom layer.

知恵層 視点 を表す項目	問題	課題	対策
判断	状況	選択肢 ↑ 展開	方法
理由	原因	⇔ 着眼点 ⇐ 対応付け	アイデア
価値	影響	↓ 意義	効果

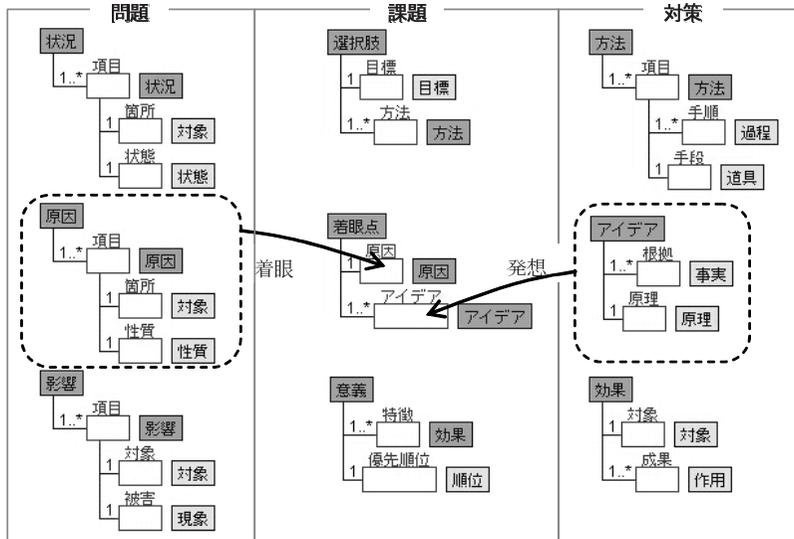


図5 問題・課題・対策を表す PIS テンプレート (オントロジーエディタ法<sup>9)</sup> により作成)

Fig. 5 PIS Template that represents problem-issue-solution (Generated by "Hozo" Ontology Editor<sup>9)</sup>).

選択肢とは、「アイデア」を用いることによって「原因」が改善可能な方法である。

展開では、設計判断を決定するまでの「選択肢」どうしを比較することにより、設計判断に選択肢が採用された理由、破棄された理由を課題の「意義」に示すために一時的な選択肢を表示する。

#### 4.2.2 PIS テンプレートにおける問題・課題・対策の表現法

設計者は、ナレッジ階層における知恵層の項目（判断、理由、価値）を独自の思考結果に基づいて表現することができる。このため、知恵層の項目を用いて問題・課題・対策を記述すると内容が抽象的になりやすいこと、また、問題と課題の違いを混同して表現している可能性があるという不都合がある。

したがって、問題と対策を用いて課題を想起する関係を示した PIS テンプレート（図5）を用いて思考過程の問題・課題・対策で検討された知恵層の内容の記述を容易にする。以下に PIS テンプレートを用いて問題・課題・対策で表現すべき項目を述べる。

##### (1) 問題の表現法

設計者が指摘した問題を明らかにするため、判断・理由・価値を以下のように状況・原因・影響で説明する。

- (a) 問題における判断は「状況」で説明する。問題の状況はいくつかの項目からなっており、さらに、各項目は問題の対象とその状態を表す。
- (b) 問題における理由は「原因」で説明する。原

因は問題の状況に陥る理由を、対象とその性質で説明したものである。

- (c) 問題における価値は「影響」で説明する。影響は問題が起きることによる被害の対象と、それによって予測される現象を表す。

##### (2) 課題の表現法

設計判断を想起することを目的とし、課題の判断・理由・価値を以下のように方針・着眼点・特徴で説明する。

- (a) 課題における判断は「選択肢」で説明する。選択肢は問題を解決できる方法を表した設計判断の候補である。
- (b) 課題における理由は「着眼点」で説明する。着眼点は重要だと考えられた問題の原因と、それを解決するためのアイデアで示される。
- (c) 課題における価値は「意義」で説明する。意義は、各課題の選択肢を比較することで説明される効果とその重要性を表す順位（優先順位）である。

##### (3) 対策の表現法

対策の有効性を明らかにするために、対策を方法・アイデア・効果でとらえる。

- (a) 対策における判断は「方法」で説明する。方法は具体化された問題解決の過程（手順）と、その手段となる道具である。
- (b) 対策における理由は「アイデア」で説明する。アイデアは設計者が着眼して根拠とする事実

と、解決に用いられる原理である。

- (c) 対策における価値は「効果」で説明する。効果は対策によって過去に明らかとなった成果、もしくは、期待される成果である。

### 5. CAPIS モデルにおける思考過程の表現手順

#### 5.1 概要

CAPIS モデル方式では、ナレッジ階層に基づく概念化プロセスを用いて思考過程の表現を、(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進める (図 6)。

入出力概念モデルの記述では、最初に設計ドキュメントの中から問題と対策を表す部分をナレッジ階層のデータ層に挿入する。次に、データ層、情報層、知識層、知恵層の順に概念モデルを記述する。

思考概念モデルの記述では、対応付けと展開により設計判断を表出し、知恵層を記述する。その後、概念化プロセスを進めることによって知識層、情報層、データ層の順に概念モデルを記述する。入出力概念モデルと思考概念モデルの記述手順をまとめて図 7 に示す。

#### 5.2 入出力概念モデルの記述法

入出力概念モデルの記述は、設計ドキュメント上の主題 (問題、および、対策) を表すことを目的とする。そのための準備として、指摘した問題や実施した対策

の結果となった機能、クラス、モジュール等のシステム構成要素に関する記述をナレッジ階層のデータ層に挿入する必要がある。

入出力概念モデルの記述では表 2 に示した概念化プロセスに従って、データ層の内容を基にして情報層、知識層、知恵層の順に問題や対策の内容を記述する。特に、知恵層においては主題をデータ層に記述したシステム構成要素と一致させるように、知恵層を基にして知識層、情報層、データ層の順に概念化プロセスをさらに進め、さらに、各階層の主題が一致するように修正する。

#### 5.3 思考概念モデルの記述法

思考概念モデルの記述は、設計判断の選択肢で構成された課題を表すことを目的とする。特に、思考概念モデルの記述手順は、以下の項目が重要である。

- (1) PIS テンプレートをを用いた対応付けと展開によって想起された設計判断の記述
- (2) 設計判断の内容を理解するための設計作業の手順や設計技術の使い方の記述

対応付けと展開によって定義された知恵層を基に、知識層、情報層、データ層の順に概念化プロセスに従って課題を記述する。その後、データ層に記述した設計技術の内容が知恵層の設計判断と一致するように情報層、知識層、知恵層の順に概念化プロセスをさらに進める。

##### 5.3.1 対応付けによる設計判断の記述手順

対応付けによる設計判断の記述手順においては、思考概念モデル上の課題の設計判断を識別するために、問題 (入力概念モデル) と対策 (出力概念モデル) の関係を分析することが重要である。具体的には、問題の「原因」と対策の「アイデア」をもとに思考した着眼点と、課題の選択肢を設定することである。

PIS テンプレート (図 5) をを用いた対応付けによる

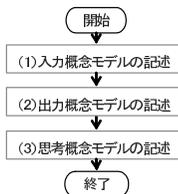


図 6 思考過程の表現手順

Fig. 6 Expression procedure of design thought processes.

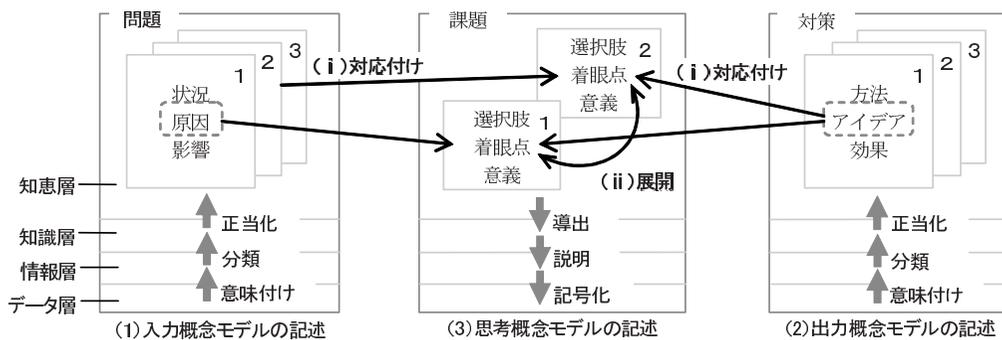


図 7 入出力概念モデルと思考概念モデルの記述手順

Fig. 7 Description procedure of conceptual input-output model and conceptual thinking model.

設計判断の記述手順を以下に示す。

- (1) 問題の「原因」を課題の「着眼点」に記述する（着眼する）。同様に、対策の「アイデア」を課題の「着眼点」に記述する（発想する）。
- (2) 設計判断として対策に用いられた「選択肢」を記述する。「選択肢」には(1)で選択した「原因」を解消し、そのための「アイデア」を用いている判断の内容を記述する。

### 5.3.2 展開による設計判断の記述手順

設計判断の展開による記述手順は、課題における選択肢を評価し、その中から設計判断を決定するまでの過程を記述する中で、対応付けにおいて列挙された着眼点ごとに課題を記述することと、各選択肢を対比して特徴を示すことにより設計判断を決定した根拠を表現することが重要である。

PIS テンプレート（図5）を用いた展開による設計判断の記述手順を以下に示す。

- (1) 対応付けによる結果だけでなく、設計判断を決定するために採用された別の「原因」と「アイデア」を課題の「着眼点」に記述する。
- (2) (1)で設定した「着眼点」に基づいて「選択肢」を記述する。
- (3) 採用された設計判断を表す課題では、「なぜこの選択肢を選んだか」が分かるように、別の選択肢と比較して「意義」を記述する。

## 6. 適用実験

### 6.1 実験目的と手順

#### (1) 実験目的

CAPIS モデル方式を用いた思考過程表現の有用性を調べるために、限られた時間内での表現のしやすさと、他の設計者に思考過程の設計判断が伝達されていることを確認する実験を行う。本実験ではETロボットコンテスト<sup>10)</sup>のためのライントレースロボットのオブジェクト指向モデリングにおける思考過程を対象とする。

ETロボットコンテストは、ライントレースロボットがコース上のラインを光センサで読み取って周回する速さと、UMLで記述した分析・設計モデルの良さを競うことを目的としている。毎年コンテストでは参加者が異なるため、ライントレースロボット開発の経験者のノウハウやアイデアを利用・発展できるように設計判断を伝える必要がある。このことから、経験者による思考過程を表現し、ETロボットコンテストに初めて参加する設計者へ経験者の設計判断が伝達可能であることを確認するとい

う目的で、本実験では以下の手順に従い適用実験を進めた。

#### (2) 実験手順

##### 手順1 経験者による思考過程の表現

ETロボットコンテストに参加して分析・設計を行ったことのある4人の経験者(A, B, C, D)が、オブジェクト指向モデリングで行った問題解決の手順をメモ(約300字の自由形式)に記述する。次に、筆者らがメモを参照しながら問題解決の手順についてインタビューを行う。その後、経験者はCAPISモデル方式を用いて入出力概念モデルの記述を行い、思考概念モデルを記述することにより、思考過程を表現する。

##### 手順2 思考過程を用いた設計者によるモデリング

ETロボットコンテストに初めて参加する設計者(a, b, c)は、経験者が表現した思考過程を用いてライントレースロボットのモデリングを行う。その後、筆者らは思考過程が伝達されたかを確認するため、設計判断を採用・採用しなかった理由、思考過程表現のメリット、困難だったことの項目に関するインタビューを行う。

##### 手順3 インタビュー結果に基づく考察

手順1, 手順2を行った後、経験者による思考過程表現の結果と設計者によるモデリングの結果とインタビューの結果を用いて思考過程の複雑性と多様性の解決について考察を行う。

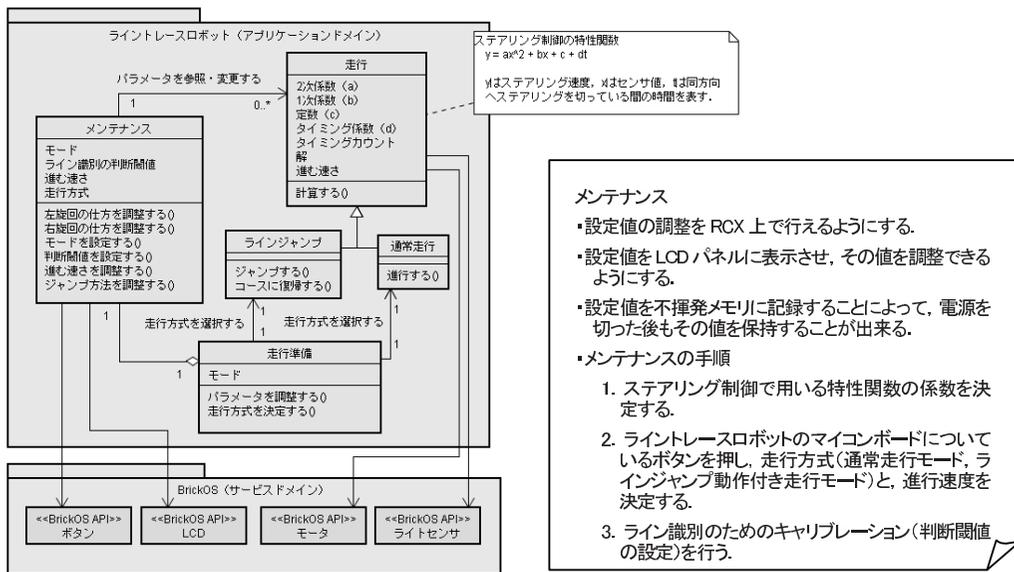
### 6.2 経験者Aによる思考過程の表現結果

実験結果の中で経験者Aによる「メンテナンスクラスの導入」に関する思考過程は詳細な項目を持ち、かつ、設計判断の有効性が分かりやすい表現である。以上の点から、経験者Aによる思考過程の表現結果を用いて検討する。

経験者Aは、クラス図(図8)においてライントレースロボットの走行を制御するときに用いられるパラメータを調整・把握できるように、「メンテナンス」のクラスを抽出した。クラスを抽出するまでの思考過程を表現する手順を以下に示す。

#### (1) 入力概念モデルの記述

クラス図における「走行」クラスをデータ層とし、ナレッジ階層に基づく概念化を用いて入力概念モデルを記述した。「走行」クラスにはステアリング制御のためのパラメータが多数ある。そのことから「走行」クラスにある「複雑なパラメータ」を問題として指摘した。すなわち、入力概念モデルのデータ層にはまだメンテナンスを導入していないクラス図を記載し、知恵層では問題の状況として「走行テスト



メンテナンスクラスに関連する部分

実験で作成した問題解決のメモ

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.

<p><b>知恵層</b></p> <p>■ 状況(判断) 箇所: ライトレースロボット 状態: 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。</p> <p>■ 原因(理由) 箇所: プログラムメモリ 性質: 組み込み機器である為、コンパイル後マシンにプログラムを転送する必要がある。</p> <p>■ 影響(価値) 対象: 制御に用いるパラメータ 被害: プログラムにアルゴリズムを追加することでパラメータが複雑化し、テスト効率が低下する。</p>	<p><b>知恵層</b></p> <p>■ 選択肢(判断) 方法: メンテナンスのクラスを作る。</p> <p>■ 着眼点(理由) 原因: 走行用のパラメータを設定するたびにソースコードを変更し、コンパイルした後、プログラムを転送するため、実行テストの作業が非効率的になる。 アイデア: LCDパネルからパラメータを設定する方法をとる。</p> <p>■ 意義(価値) 特徴: テスト作業の効率が向上する。</p>	<p><b>知恵層</b></p> <p>■ 方法(判断) 手段: パラメータの設定をするためにメンテナンス操作を行うクラスを導入する。</p> <p>■ アイデア(理由) 根拠: LCDパネルからライトレースロボットを操作することができる。 原理: パラメータをEEPROMに保存する。</p> <p>■ 効果(価値) 対象: 走行テスト 成果: ソースコードの変更にしにパラメータの調整が可能になった。設定内容が把握できるようになった。</p>
---	--	--

(a) 問題: 複雑なパラメータ

(b) 課題: パラメータの設定方法

(c) 対策: メンテナンスクラスの導入

図 9 経験者 A が記述した問題・課題・対策

Fig. 9 The problem-issue-solution that expert A described.

時の効率が悪い」ことをあげた (図 9(a)).

(2) 出力概念モデルの記述

クラス図における「メンテナンス」クラスをデータ層とし、出力概念モデルを記述した。設計結果として導入されたメンテナンスクラスには、「走行」クラスが持つパラメータを参照・変更するという関連を持たせている (図 8)。クラス図の補足説明として、メンテナンスを実行する手順は、ユーザからのボタン操作によってステアリング制御の特性関数に用いる変数値、走行方式、ライトレースロボットが進む速さ、ライン識別のための判断閾値の各項目を設定することによって進められた。以上のデータ

をもとに、知恵層における対策の方法として「メンテナンス操作を行うクラスを導入する」ことをあげた (図 9(c))。

(3) 思考概念モデルの記述

メンテナンスクラスを導入するまでの設計判断を表現するため、経験者 A は対応付けと展開を以下の手順で行った。

- (a) 経験者 A は「LCD パネルからライトレースロボットが操作できる」というアイデアを利用して「ソースコードを変更する必要がある」という原因の解消を着眼点とした。そのために、課題では「メンテナンスのクラス

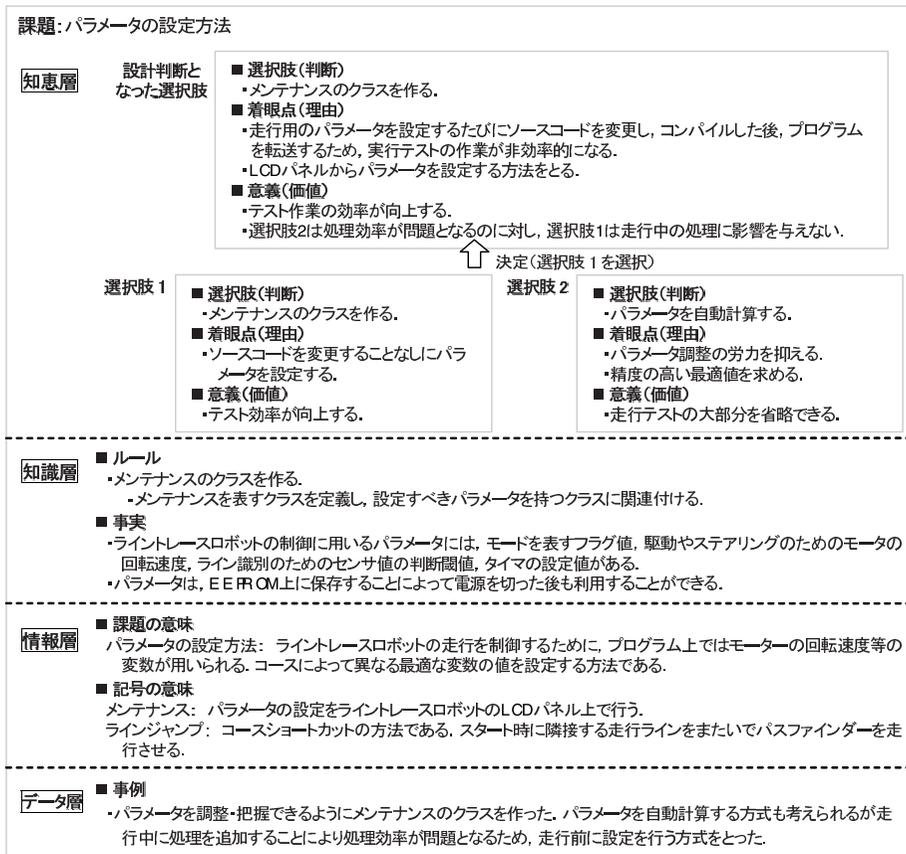


図 10 熟練者 A が記述した思考概念モデル

Fig. 10 The conceptual thinking model that expert A described.

を作る」という設計判断を想起し、課題として記述している。

- (b) さらに、問題の原因から「パラメータ調整の労力を抑える」に着眼点をおいて、パラメータの自動計算という選択肢を想起した。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」という特徴がある。そのため、設計判断となつた選択肢の意義に、この選択肢との比較結果を記述している。

対応付けと展開の後、経験者 A はメンテナンスを作るためのルールや事実を知識層に記述し、さらに知恵層を基に情報層、データ層の内容を記述している(図 10)。

### 6.3 経験者 4 人の思考過程の比較結果

経験者 4 人とも CAPIS モデル方式で思考過程を効率的に表現できていることを確認するため、各経験者が設定した問題・課題・対策と思考過程の表現に要した作業時間を表 4 に示す。

以下、各経験者の思考過程を比較したときの概念モデルの特徴と作業時間の差異について述べる。

#### (1) 概念モデルの特徴

各経験者の設計意図や常識・専門知識に関する内容は手順 1 で作成したメモには記述されていないが、CAPIS モデルの思考概念モデルの各層に記述されている。

たとえば、経験者 A のクラス図やメモ(図 8)からでは各クラスが持つ役割の内容は不明確である。対照的に、経験者 A の思考概念モデル(図 10)では、情報層で「メンテナンス」がパラメータの設定をライトレースロボットの LCD パネル上で行うことや、メンテナンスの結果を用いる「ラインジャンプ」がコースショートカットの方法を表していることを説明している。さらに、知識層ではメンテナンスに用いられるパラメータの種類を説明しており、知恵層で、「メンテナンスクラスを作る」という設計判断とこれを正当化する理由と価値を説明しており、各クラスが持つ役割が明確に表現されてい

表 4 各熟練者が記述した思考過程と作業時間  
Table 4 The thought processes of experts and process time.

設計者	記述した思考過程 (問題・課題・対策)	入力概念モデルの記述時間	出力概念モデルの記述時間	思考概念モデルの記述時間	全体作業時間
経験者 A	問題: 複雑なパラメータ 課題: パラメータの設定方法 対策: メンテナンスクラスの導入	60 分	40 分	50 分	150 分
経験者 B	問題: 無駄 (ジグザグ) な走行の軌道 課題: 速度の向上とマシンの安定化 対策: ステアリング角度の制限	25 分	35 分	20 分	80 分
経験者 C	問題: 責務が類似しているクラス群 課題: クラス図のシンプル化 対策: クラスの再構成	60 分	40 分	90 分	190 分
経験者 D	問題: コースから脱線する確率 課題: キャリブレーションの方法 対策: 高精度な中央値の計算	55 分	70 分	80 分	205 分

る。設計判断を展開した結果においては、「パラメータを自動計算する」という破棄された選択肢も思考されていることが分かった。この選択肢は、実験手順 1 のインタビューの時点においても経験者 A に意識されていない項目であり、設計判断を展開するときに初めて記述された。他の 3 人についても同様の結果が得られたことから常識・専門知識、設計意図の記述も行われており、設計判断の想起法の定式化が行われていることを確認した。

## (2) 作業時間の差異

思考過程を表現するのに要した時間は最長で約 205 分、最短で約 80 分であった (表 4)。

作業時間が短かった経験者 B は概念モデルの記述を 1 度行ったことがあり、慣れによるものと考えられる。それに加えて、問題や課題として記述した内容も比較的簡単なものであることも理由として考えられる。

他の 3 人については、入出力概念モデルの記述においては作業時間の差異はあまり見られないが、思考概念モデルを作成する段階で作業時間の差異が生じている。差異の要因は主に対応付けの作業であった。経験者 C の場合、対応付けの作業で設計判断を想起して、思考概念モデルを記述するまでに約 90 分を要している。入力概念モデルと出力概念モデルに選ぶべき問題と対策の内容の記述を誤ったため再度入出力概念モデルを記述し直しており、その後の修正作業に時間を要している。経験者 D の場合、思考過程を最も詳細に記述しており、特に展開に時間を割いて選択肢を 3 つ記述していたことから他の経験者より作業時間を要した。

## 6.4 思考過程を用いた設計者によるモデリングの結果

経験者 A と経験者 B の思考過程を用いて別の設計者 (a, b, c) がモデリングを行った結果、表 5 のように設計判断の採用状況が分かれた。設計判断を採用

表 5 思考過程を用いた設計判断の採用状況

Table 5 The cases of applying design decisions with thought processes.

設計判断	設計者による採用・不採用の理由
経験者 A の設計判断	<ul style="list-style-type: none"> <li>• 対策の内容に既存のモデルと競合する部分があるため不採用 (設計者 a)</li> <li>• メンテナンスクラスは導入が簡単であることと、テスト効率を上げることができるため採用 (設計者 b)</li> <li>• 既存のモデルではデバッグをすると多くの時間を費やすため採用 (設計者 c)</li> </ul>
経験者 B の設計判断	<ul style="list-style-type: none"> <li>• ステアリングの角度を制限して走行を安定させることの記述を見て採用 (設計者 a)</li> <li>• マシンの挙動が安定する見込みがあるため採用 (設計者 b)</li> <li>• 既存のモデルで現在直面している問題を含んでいるが、そのアイデアを用いることの効果が検証されていないため不採用 (設計者 c)</li> </ul>

(または不採用) した理由は設計者自身の思考過程と経験者の思考過程を比較した結果の類似性によるものであった。その比較方法として、以下のように各設計者は問題、課題、対策の各知恵層を用いて検討していることが分かった。

### (1) 問題に関する思考過程の比較

クラス図を作成する設計者自身が直面している問題と経験者の思考過程における問題との比較結果を示している。

### (2) 課題に関する思考過程の比較

経験者の思考過程における課題の「意義」を用いて設計判断を採用した理由を説明している。表 5 では、経験者 A が思考概念モデル (図 10) の課題に示した「テスト効率をあげること」を設計者 b が説明に用いている。

### (3) 対策に関する思考過程の比較

「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」を設計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。

## 6.5 考 察

### (1) 思考過程の複雑性の解決について

経験者 A による思考過程の表現結果と経験者 4 人の思考過程の比較結果により、CAPIS モデル方式を用いて経験者は思考過程を限られた時間で問題・課題・対策に分けて表現できることを確認した。また、PIS テンプレートをを用いた対応付けと展開によって、図 9 のように経験者が設計判断の選択肢を想起していることを確認した。

さらに、思考過程を用いた設計者によるモデリング結果では、経験者の設計判断を採用した理由を説明するために、問題・課題・対策の各知恵層を用いていることが分かった。このことにより、各設計者は経験者の思考過程を用いて問題をとらえ、利用すべき設計判断を選択してモデルに反映させることができたと考えられる。

以上のことから、PIS テンプレートをを用いた対応付けと展開で思考過程を問題・課題・対策に分けることにより複雑性を解決して思考過程を表現できることを確認した。

### (2) 思考過程の多様性の解決について

経験者 A による思考過程の表現結果により、クラス図やメモの記述内容では言及されなかった用語の意味や常識・専門知識、設計意図を明示して課題の設計判断を記述していることを確認した。これは、ナレッジ階層に基づく概念化プロセスを用いることにより、データ層、情報層、知識層、知恵層の階層構造で経験者の設計判断が表現できたためと考えられる。

このことから、ナレッジ階層に基づく概念化プロセスを用いることにより、多様性を解決して思考過程を表現できることを確認した。

## 7. 関連研究

Garcia ら<sup>11)</sup> が提案する ADD は、特定分野において表現力の高い議論モデルを記述することができるが、設計ドキュメントに設計意図の情報を組み込むための作業負荷の大きさが問題となる。CAPIS モデル方式は思考過程が持つ複雑性を考慮し、簡潔な問題・課題・対策で構成される概念モデルで設計判断を表現することができる。また、インタビューを行う第 3 者が各概念モデルを最初に記述し、それを基に思考過程を表現する経験者が詳細を補完する手順をとることができるので、経験者自身に多くの負担はかからない。

野間口ら<sup>12)</sup> は思考過程のモデルを用いた設計知識管理システム DDMS を開発している。このシステム

は、設計者への問合せによって設計過程（操作履歴や背景情報等）を半自動で文書に記録する方式をとっており、設計者の負担を軽減できるメリットがあると考えられる。しかし、設計者の暗黙的な知識（ここでは設計判断の理由を指す）を得るために質問文を用いて設計者に回答させるので、設計作業の遂行に支障をきたす可能性があるだけでなく、回答結果は設計判断に対する理由を表現するにとどまっており、問題から解決までの因果関係を示していない。CAPIS モデル方式は、思考過程を問題・課題・対策に分割して表現するため、問題から解決までの因果関係で設計判断を表現することができる。特に、対応付けと展開によって設計判断の選択肢や着眼点を想起し、最終的な設計判断を選択するまでの思考過程を表現するところに特徴がある。

Tsumaya<sup>13)</sup> は、設計作業の観察結果から 7 つの知識操作（知識・情報の獲得、知識・情報の再構成、情報の確認、矛盾の解消、知識・情報の修正、解のシンセシス、解の分析）とそれらを実際の設計作業に対応付けるための基本語彙を分類している。しかし、それらの基本語彙を用いて試行錯誤が繰り返されるソフトウェア設計の思考過程を表現するには多大な労力を必要とする。すなわち、思考過程を文章として記述した結果から、各分脈に対して基本語彙をあてはめて分類しなければならない。そのような思考過程の文章には試行錯誤の内容があると同時に、複数の基本語彙が同時にあてはまる場合や、文脈の境目がはっきりしていない場合がある。CAPIS モデル方式では、PIS テンプレートをを用いた対応付けと展開で思考過程を問題・課題・対策に分ける方法でこの複雑性に対処している。

Takeda ら<sup>14)</sup> は、実世界における設計過程を機能空間から属性空間への写像ではなく試行錯誤を行う段階的な作業として、CAD 支援のための設計過程モデルを構築した。Takeda らの計算可能な設計過程モデルでは、解くべき問題を指摘する過程や結論を評価する過程が詳しくモデル化されていない。CAPIS モデル方式は、入力概念モデルにおいて問題を指摘する過程から、出力概念モデルにおいて対策を実行し評価する過程までの主観的な価値判断を知恵層で表現する手段となる。

協同設計のプロセスを記述する Valkenburg<sup>15)</sup> の方法は、設計者間で行われる会話の内容をエピソード単位に分割し、4 種類の活動（問題の設定、活動プランの決定、問題解決、評価）に分類するものである。各エピソードは設計者間の会話であるため、設計者に利用された常識や専門知識、エピソードの背景となる文

脈等は明らかでない。ナレッジ階層に基づく概念化プロセスは、設計者個人の詳細な思考過程をエピソードごとに表現することを可能にする。

Guarino ら<sup>8)</sup> は概念化について「一部の現実の構造に制約された暗黙的なルールを記号化する内包的な意味構造を示すこと」と定義している。また、Maedche<sup>16)</sup> は記号・事物・概念の3つで表現されたミーニングトライアングル<sup>17)</sup> を用いて、「記号と事物の関係で定義される意味」が概念の表現であることを説明している。しかし、両者とも意味構造の詳細や概念化の方法を示していない。CAPIS モデル方式ではそれらの項目を明示すべきであると考え、記号や意味の定義を基本プロセスとし、それを基にして命題やその命題の正しさ(判断の理由と価値)をナレッジ階層で表現することができる概念化プロセスを定義した。

データ、情報、知識、知恵等について Ackoff<sup>5)</sup> や Bellinger ら<sup>6)</sup> は各分類を定義しているが、各分類の区別は知識利用の目的によって異なるためにいまだ議論が分かれている。溝口<sup>18)</sup> はデータ、情報、知識を表現の役割(ロール概念)として整理したが、データや情報間の階層関係についてまでは述べていない。それに対して筆者らのナレッジ階層に基づく概念化プロセスでは、階層関係を持ったデータ、情報、知識、知恵を用いて問題や課題等の対象(主題)を表現できることを主張する。すなわち、ナレッジ階層を記述することにより、設計ドキュメント(データ層上の要素)を理解して判断の結果(知恵層上の要素)を表現するまでの設計者の思考を示すことができると考える。

優れた思考過程をだれもが使えるようにテンプレート形式で表現したソフトウェアパターン(アーキテクチャパターン、アナリシスパターン、デザインパターン、フレームワーク等<sup>3)</sup>) が構築されてきた。背景となる問題や用いるべき理由を理解せずに使用した場合、ソフトウェアパターンは逆にソフトウェアの再利用性を低くする可能性がある。CAPIS モデル方式で表現する思考過程は、分野に特化した経験者によるソフトウェアパターンを命題形式で知識層に示しているが、それだけではなく知恵層でソフトウェアパターンを効果的に用いることができるように設計判断の理由や価値等の表現を提供すると考えられる。

## 8. おわりに

本論文では思考過程を表現する CAPIS モデル方式を提案した。PIS テンプレートをを用いた対応付けと展開を行うことにより思考過程を問題・課題・対策に分けて表現し、各表現をナレッジ階層で記述することに

よって思考過程をデータ層から情報層、知識層、知恵層までの階層関係で容易に表現できることを示した。

CAPIS モデル方式の特徴を以下にまとめる。

- (1) 問題・課題・対策の因果関係を表すように、思考過程を入力概念モデル、思考概念モデル、出力概念モデルの3つで構成する。各概念モデルを記述するために、「ナレッジ階層に基づく概念化プロセス」と「PIS テンプレートをを用いた対応付けと展開」を使用する。
- (2) ナレッジ階層に基づいて概念モデルを記述するために意味付け、分類、正当化、導出、説明、記号化からなる概念化プロセスを用いる。
- (3) PIS テンプレートをを用いて問題と対策の関係を分析し、得られた着眼点を基に設計判断を決定する対応付けの方法、および、最終的な決定となった設計判断と一時的な選択肢を比較することにより、経験者の設計意図を明らかにする展開の方法を用いる。適用実験では、ライントレースロボット開発の経験者4人によって記述された思考過程においてクラス図やメモで経験者に言及されなかった常識・専門知識、設計意図等が表現されている点で、経験者によって思考過程が表現できることを確認した。また、経験者の思考過程を他の設計者が用いてモデリングを行った結果、思考過程の設計判断が理解されていることが明らかになった。これらのことにより、CAPIS モデル方式を用いて複雑性と多様性を解決して思考過程を表現できることを確認した。

今後の課題としては、適用実験における思考過程の表現に要した作業時間に1~3時間のばらつきがあり、対応付けに必要となる項目を漏れなく記述している場合とそうでない場合があった。だれでも効率良く思考過程を表現できるように、ナレッジ階層やPIS テンプレートの記述項目に関する文章表現法とその解釈手順を表すメタモデルが必要である。概念モデル上の項目をオントロジ記述言語を用いて定義し、思考過程の支援方法を開発したい。

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) Polanyi, M.: *The Tacit Dimension*, Routledge & Kegan Paul Ltd. (1966).
- 2) 畑村洋太郎, 小野耕三, 中尾政之: 機械創造学, 丸善 (2001).
- 3) 中谷多哉子, 青山幹夫, 佐藤啓太: ソフトウェアパターン, 共立出版 (1999).
- 4) Brown, W.J. and Malveau, R.C. (著), 岩谷

- 宏 (訳): アンチパターン, ソフトバンク (1999).
- 5) Ackoff, R.L.: From Data to Wisdom, *Journal of Applied Systems Analysis*, Vol.16, pp.3-9 (1989).
  - 6) Bellinger, G., Castro, D. and Mills, A.: Data, Information, Knowledge & Wisdom.  
<http://www.systems-thinking.org/dikw/dikw.htm>
  - 7) 大山勝徳, 武内 惇, 藤本 洋: ナレッジ階層に基づく概念化による思考過程支援の構想, 人工知能学会第8回セマンティックウェブとオントロジー研究会, SIG-SWO-A403-03 (2004).
  - 8) Guarino, N. and Giarretta, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification, *Towards Very Large Knowledge Bases*, Mars, N. (Ed.), pp.25-32, IOS Press (1995).
  - 9) 古崎晃司, 來村徳信, 佐野年伸, 本松慎一郎, 石川誠一, 溝口理一郎: オントロジー構築・利用環境「法造」の開発と利用—実規模プラントのオントロジーを例として, 人工知能学会論文誌, Vol.17, No.4, pp.407-419 (2002).
  - 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) Garcia, A.C.B. and Howard, H.C.: Acquiring design knowledge through design decision justification, *Artificial Intelligence in Engineering Design and Manufacturing*, Vol.6, No.1, pp.59-71 (1990).
  - 12) 野間口大, 下村芳樹, 富山哲男: 設計者の思考過程のモデルを利用した設計知識管理システム, 人工知能学会論文誌, Vol.20, No.1, pp.11-24 (2004).
  - 13) Tsumaya, A., Nomaguchi, Y., Yoshioka, M., Takeda, H., Murakami, T. and Tomiyama, T.: Verification of a Model of Synthesis — The Methods for Verification and Results, *13th International Conference on Engineering Design (ICED 01)*, pp.229-236 (2001).
  - 14) Takeda, H., Veerkamp, P., Tomiyama, T. and Yoshioka, M.: Modeling Design Processes, *AI Magazine*, Vol.11, No.4, pp.37-48 (1990).
  - 15) Valkenburg, R. and Dorst, K.: The reflective practice of design teams, *Design Studies*, Vol.19, No.3, pp.249-271 (1998).
  - 16) Maedche, A.: *Ontology Learning for the Semantic Web*, chapter 2, Kluwer Academic Publishers (2002).

- 17) Ogden, C. and Richards, I.: *The Meaning of Meaning*, Routledge & Kegan Paul Ltd. (1923).
- 18) 溝口理一郎: オントロジー工学, pp.213-215, オーム社 (2005).

(平成 17 年 10 月 27 日受付)

(平成 18 年 5 月 9 日採録)



大山 勝徳 (学生会員)

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。