

# 記述言語系 OOJ 上で実現する個人向け V&V 環境の提案

大木 幹生<sup>†1</sup> 池田 陽祐<sup>†2</sup> 三塚 恵嗣<sup>†3</sup> 畠山 正行<sup>†4</sup>

**概要:** 本研究では分析記述からプログラムまでを一貫して開発するための記述言語系 OOJ と、その個人向けの記述環境を構築した。一方、近年プログラムの信頼性とその評価方法に対する重要性が着目されている。本論文の最終目標もプログラムの信頼性を評価する方法を提案することである。OOJ と記述環境はその基礎となる研究であり、その上にプログラムの信頼性を評価する環境を構築する。その基礎となる概念は V&V と呼ばれる。本発表では V&V の概念を客観的に評価する環境を提案する。この環境は記述環境上の特殊なトレーサの一種であり、OOJ の 4 段階の記述内部にある離散単位の追跡機構である。本研究ではその基本パッケージを試作した。

**キーワード:** トレーサ, V & V, 要求追跡, OOJ, 離散構造化モデル, 科学技術計算

## A proposal of the personal use V&V environment realized on the basis of descriptive language system OOJ

MIKIO OHKI<sup>†1</sup> YOUSUKE IKEDA<sup>†2</sup> KEISHI MITSUKA<sup>†3</sup> MASAYUKI HATAKEYAMA<sup>†4</sup>

**Abstract:** In the present paper, we have designed a descriptive language system OOJ and their descriptive environments for personal use that can be applied from the analysis stage up to the program stage throughout the design and the implementation stage. In recent years, the importance for the reliability and its estimation methods has been focused. The final target of the present paper is to propose the way to estimate the reliability of the program. OOJ and its descriptive environments are the basic research for their purposes. On these environments, we will develop the environments to estimate the reliability of the program. This is called the V&V environments. This environment is a peculiar kind of the tracer on the descriptive environments, and is also the tracing mechanism of the discrete units inside of the OOJ descriptions. In the present paper, we will present a basic package of the V&V environments.

**Keywords:** tracer, V & V, requirement trace, OOJ, discreted and structured model, scientific and technical calculation

### 1. はじめに

現代のプログラム開発は、例えば現象解析のための再現

シミュレーションや工学製品の安全性の確保などのプログラムの発展などに伴って、より高度化し複雑化し続けている。さらにはプログラムの信頼性などへの要求も増えつつある。我々のグループはプログラム開発<sup>\*1</sup>のそのようなニーズに対応して、分析から設計・実装・プログラムに至るまでを一貫して作業できる記述言語系 OOJ を開発した。そしてさらに、この記述言語系 OOJ を実際にプログラム

<sup>†1</sup> 現在、群馬工業高等専門学校教育研究支援センター  
Presently with Technical Support Center for Education and Research, Gunma National College of Technology

<sup>†2</sup> 現在、茨城大学大学院理工学研究科情報・システム科学専攻  
Presently with Graduate School of Information and System Science, Ibaraki University

<sup>†3</sup> 現在、株式会社日立システムズ  
Presently with Hitachi Systems, Ltd.

<sup>†4</sup> 現在、茨城大学工学部情報工学科  
Presently with Department of Computer and Information Sciences, Ibaraki University

<sup>\*1</sup> 本論文で多く用いる「プログラム開発」という用語は個人で開発する小規模なものを指す用語として用いる。よく使われるソフトウェア開発という用語は、特に、ソフトウェア工学に基づく大規模な開発を意味する用語として用いる。この点についての詳細な比較は参考文献 [1], [2] のいずれも表 2 にある。

開発で運用するための OOJ 記述環境も開発した。

この OOJ 記述環境は開発の初期から科学技術計算用のプログラムの簡潔な開発環境とすると共に、その信頼性を確保・向上させるために開発されてきた。本研究においては OOJ の記述環境の上にプログラムの信頼性を向上させるツールあるいは環境の構築を試みる。本発表はその基本アイデアの提案である。

## 2. 記述言語系 OOJ とその記述環境

### 2.1 記述言語系 OOJ

OOJ は個人が作る小規模の科学技術計算用のプログラムを開発することを主な目的にしており、対象世界の分析モデリングに始まり、その分析記述を基にして設計や実装の段階の記述に変換し、最後に Java, C++, Fortran90 等の言語のプログラムに自動変換するための記述言語系であり、3つの類似性のある記述言語から成り立っており、記述言語間はトランスレータによって結ばれている。OOJ エディタを持つので開発環境でもある。そのような開発環境も含めた記述言語系 OOJ の概要を図 1 に、3つの記述言語の比較を表 1 に示す。その他の特徴としては、OOJ の基本モデルとして離散構造化モデルを採用していることである。このモデルは数値計算に常用されている離散化モデルにちかく、同時に特化したオブジェクト指向モデルとしての特徴も持っている。

### 2.2 OOJ 記述環境

OOJ 記述環境の構成は記述作業を行う OOJ エディタと変換を行うトランスレータからなる。OOJ エディタは分析段階から実装段階までを一貫して記述できるよう 1つのアプリケーションとして実装しており、離散単位を中心とした操作性を実現している。トランスレータは4つの段階を繋ぐように構成されており、分析段階から実装段階のトランスレータでは記述形式のみの変換を行い、意味内容の変換は想定ユーザ自身の作業としている。実装段階からプログラムへのトランスレータは、プログラム変換 [3] を用いて変換対象となるプログラミング言語の種類毎に実装している。また、一貫した相似性の検証作業をするための基盤として、各段階間での追跡ができるよう離散単位間の対応関係データも自動的に生成する。

### 2.3 プログラムの信頼性と V&V の概念

ソフトウェア工学の分野における信頼性は以前から多くの研究が為されている [4]。しかし科学技術計算の分野におけるプログラムの信頼性についてはあまり重点には置かれていない。そこで本論文では、OOJ を対象とした科学技術計算におけるプログラムの信頼性を以下のように定義づける。

(1) プログラムがユーザの分析記述 (OOJ で言えば OONJ

記述) を正しく反映したプロセス (開発手順) を経ているか\*2という**開発過程の信頼性**。

(2) 計算結果の信用性つまり**分析記述の再現性**

という2点にある。この2点を合わせた概念をここで仮に**プログラムの信頼性**と呼ぶことにする。現状では経験的に判断したり、参考文献のシミュレーション計算結果との合致が良かったり、同じ条件の実験との比較で一致すればプログラムの信頼性を得たとされる。しかし、それらは上記(1), (2)についての直接的な解決ではなく別手段を採っているだけである。従来型の検証手段では、ブラックボックスのまま残る開発過程と、対応関係が不明な分析記述の再現性の問題が残る。

一方、上記の(1)と(2)は現在盛んに検討されているV&Vの概念 [5], [6], [7]の一部とほぼ同じ概念である。V&Vとは当初の要求仕様や設計が正確な手順と過程(プロセス)を踏んでプログラムに正しく反映されていることの検証(Verification)と、開発の狙い・要求と意図が達成されたプロダクトとしてのプログラムであることの妥当性の確認(Validation)を指す概念である。

### 2.4 プログラムの信頼性を向上させる特殊なトレーサの考案

図1に見るようにまず対象世界を離散単位でモデリングし、分析記述が作成され、変換や追加を経過しながら実装記述に至り、図1のOOPLトランスレータにより変換されてOOPLプログラムに至る。そこで着目すべき点は分析段階の離散単位は設計・実装と進んで表現形式は変わってもその実体内容は変化しない、という特徴がある。それはOOJが一貫相似性という特性を持っているからである。

**一貫相似性**とは分析記述から設計・実装の段階を経てプログラムに至る過程の4つの段階間の全ての離散単位および離散構造化モデルの関係が上記の相似性を持つことを指す。すなわち4つの段階の対応する離散単位(群)において『表現形式は異なっても、同等/同値の記述や計算結果になる特性』を指す、と定義する。

そうであれば、複数の段階を経過する過程で変換される分析段階の離散単位を(理想的に言えば)プログラムに至るまで追跡すれば、離散単位の表現形式は例えプログラム表現(=プログラム)に変わっても、表現内容が変わらなかったか否かを検証することができることは明らかである。その作業を全ての離散単位(群)に対して行えば各々の離散単位の一貫相似性は検証可能である。

その状況を模式図にしたのが図2である。分析・設計・実装・プログラムの4つの各段階内部には丸印の離散単位とそれらを繋ぐ線で表される相互関係が張られている。そして4つの段階間に張られているリンク線( $l(n1, d1)$ や

\*2 通常の業務システムであれば顧客の要求仕様等が使われるが、科学技術計算においては分析記述が出发点となる。

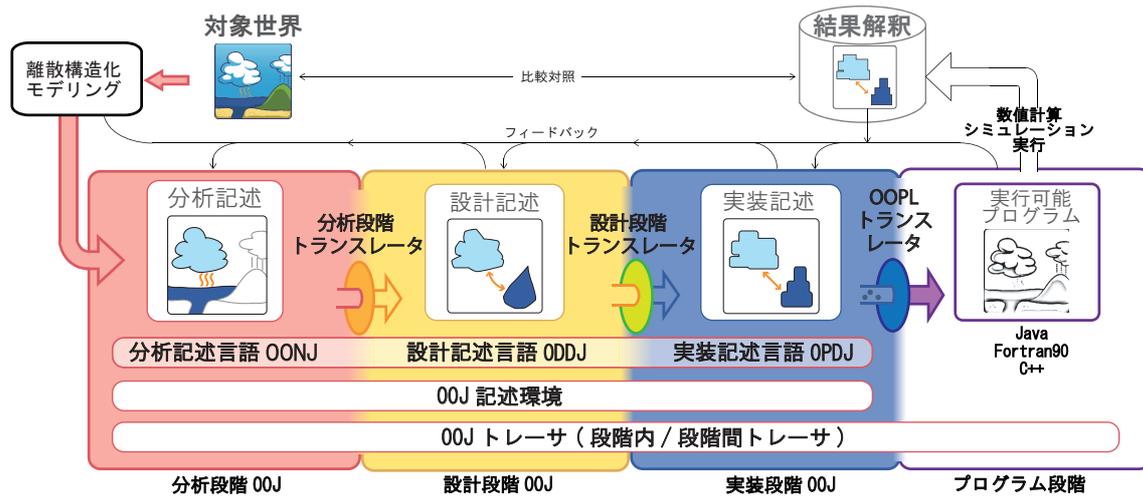


図 1 OOJ を基盤としたプログラム開発過程の全体概念図

Fig. 1 A whole concept diagram of program development processes based on OOJ

表 1 OOJ の各サブ段階言語の特徴と役割分担

Table 1 Concept design and role sharing of each sub language of OOJ

(A) 分析段階記述言語 OONJ	対象世界に関する情報で、プログラム変換に至るまでに必要となる情報(離散単位や構造についての情報)を全て記述すること。最小離散単位の日本語文は単文であることが推奨される。そしてその離散単位の構造化を要求される。数式は各記述者の考えや通常使う記号と方式で記述して良い。この段階では計算機世界に関わる情報は記述しない。
(B) 設計段階記述言語 ODDJ	計算機世界特有な要素の追加や表現変換を行う。例えば、データ型やアクセス属性を加え、属性を変数と呼び変え、配列やリスト構造を導入したりする。この段階では特定の OO プログラミング言語に関わる事項は記述しない。
(C) 実装段階記述言語 OPDJ	各特定プログラム言語((OO)PL) 向けのプログラムに自動変換するための記述を作る。現時点では、Java、C++、Fortran90 に変換可能である。
(D) 統合記述言語 OOJ	上記 3 つの記述言語を概念上で統合した言語であるが、3 つの記述言語の設計主題と解決目標の枠組を示す言語でも。ただし本来は一貫した記述言語 OOJ がまず設計されたが、必要があつて個別のサブ言語が先に発表された。

L(D1,P1)) を辿って行けば分析段階の離散単位がどのように変換されたかを知ることができる。その変換前後の離散要素の比較評価を 3 つのリンクを辿って行えば分析段階の離散単位がプログラムでどうなったかを知ることができる。その情報を合計すれば、分析記述がプログラムに変換されたかが分かることになり、したがって、プログラムの信頼性を評価できる。つまり言い換えればこの一貫相似性の概念には 4 つの段階の離散単位間の「一貫した追跡 (integral trace)」という概念も重要な部分として含まれている。結論としてはリンク線 ( $l(n1, d1)$  や  $L(D1,P1)$ ) を辿るためのトレーサ [8], [9] を構築することでプログラムの信頼性の一部を評価できるのである。

### 3. OOJ 上の特殊なトレーサの概念設計

#### 3.1 OOJ における検証作業

OOJ では分析記述の再現性を実現するための相似性を設定している [10]。

(1) 離散構造化モデルの相似性は、全ての記述が離散構造化モデルで記述されていることを示すものであり、離散構造化モデルを記述するのに特化した OOJ 記述環境 [11] を用いることで相似性は保証される。

(2) 形式化された作業の相似性は OOJ の設計段階以降で行う形式化されたプログラミングを行った箇所の相似性である。これらの作業は想定ユーザが行うが、OOJ の記述規則では相似性が既に保証されており、それを組み込んだ OOJ 記述環境で全て支援されている。そのため想定ユーザが OOJ 記述環境に従った作業を行う限り、相似性が保証される。

(3) 記述の意味内容における相似性は記述の意味内容についての相似性である。OOJ において記述の内容は全て最小離散単位で記述され、分析記述の最小離散単位では文法を規定しておらず自由な記述ができることを前章で示した。そのため意味内容を OOJ 記述環境で取得し、相似性を保証するのは不可能である。OOJ では個人で一貫した作業を行うことが前提にあるため、これは想定ユーザが検証すべき相似性だとした。

以上のことから (1), (2) は OOJ の記述規則と OOJ 記述環境により相似性が保証されている。(3) は意味内容の相似性であり、想定ユーザが相似性を検証することが可能であり、OOJ ではこの作業を「相似性の検証作業」と呼ぶ。また (1) は分析段階から実装段階までの各記述が離散構造化モデルで記述されており、OOJ 記述環境では離散単

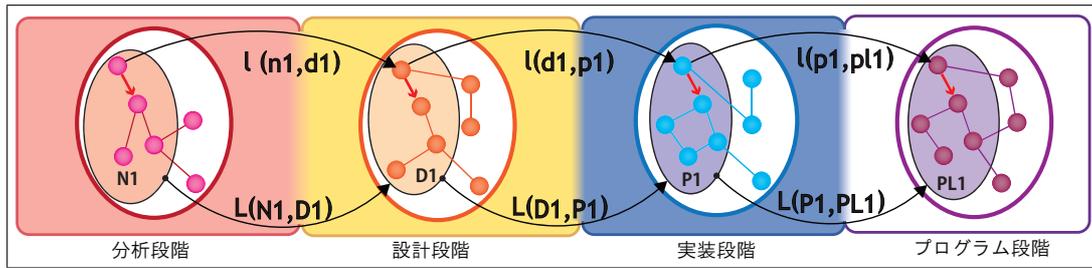


図 2 離散単位間の対応関係リンクの模式図

Fig. 2 Correspondence links among discrete units

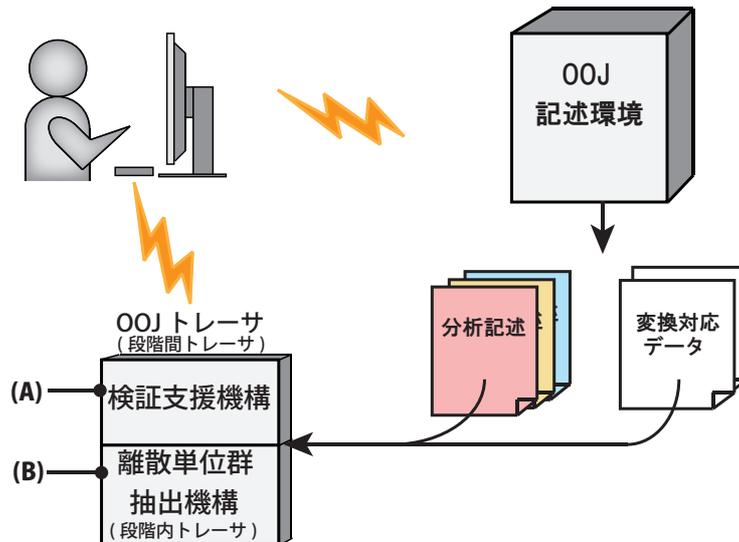


図 3 OOJ トレーサのシステム構成

Fig. 3 System organization of OOJ tracer

位が追跡性を持つよう設計されている。そこで各段階の離散単位を比較することで相似性の検証作業とし、開発過程の信頼性の検証を行う。この支援機構として OOJ トレーサを開発している。

### 3.2 OOJ トレーサの概要

OOJ トレーサと OOJ 記述環境との概要を図 3 に示す。OOJ 記述環境から分析記述、設計記述、実装記述と離散単位間の追跡データが記述されている変換対応データを取得する。これらのファイルを用いて、図 3-(A) に示す相似性の検証作業を支援するための機構を介し、想定ユーザは OOJ トレーサ (段階間トレーサ) を操作する。

一方で、各記述は離散構造化モデルで記述しており、例えばアルゴリズムなどは複数の構造化された最小離散単位として表現される。このようなアルゴリズム記述間に着目した相似性の検証作業が必要なのは明らかである。そこで各記述内から構造化された離散単位群を抽出するための機能として図 3-(B) の離散単位群抽出機構 (段階内トレーサ) を実装した。

以上のように離散単位と構造化された離散単位群の 2 つの点から相似性の検証作業を行えるよう設計した。

### 3.3 検証支援機構の概要

OOJ の目的は「分析記述がどれほど忠実に再現できたか？」を示すことでもある。これは変換されたプログラムに対して、分析記述の再現性の定量化と言い換えることができる。分析記述の再現性を定量的に示すには、離散単位毎の相似性を定量的に示す必要があることから「相似性を定量的に表現する。」ことが検証作業の目的である。定量的に相似性を表現することで客観的な評価を行えるようになるため「数値的な判断基準」を記述するべきである。

また、設計段階以降は段階的に詳細化する作業であり離散単位の数が増えることが想定される。この場合には「追跡先を編集できる」ことが必要である。

このようにプログラムの評価を行う手法として、ソフトウェア品質評価の技術がある [12]。ここで挙げられている品質評価項目として機能性や保守性、移植性などがある。ただし、OOJ において評価すべきなのは分析記述の再現性であり目的が異なり、想定ユーザの特徴から簡易で分かりやすい技術が必要であるためこれらの品質項目を現在は取り入れていないが、ユーザ評価によって検討を考える。

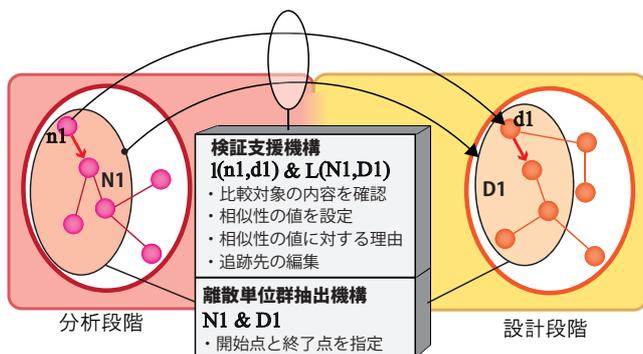


図 4 OOJ トレーサにおける検証作業の仕組み

Fig. 4 Mechanism of verification with OOJ tracer

### 3.4 離散単位群抽出機構の概要

4.2 節で示したように離散単位抽出機構は、相互作用情報伝達 (mp) に着目して記述内の構造化された離散単位群の抽出を行う。

抽出方法は離散構造化モデルで記述されていることを利用し、開始点となる離散単位と終了点となる離散単位を想定ユーザが指定することで、2 点の間の構造化された離散単位群を切り出す。これは、想定ユーザは意図した構造化が行われているかの確認も行うことが可能である。

### 3.5 OOJ トレーサを用いた検証作業の概要

図 1 を基に OOJ トレーサを用いた検証作業の概要を図 4 に示す。図 4 には離散グラフ的に表現した各段階の記述を示し、その内部には比較対象となる離散単位 (図 4 の n1, d1) と離散単位群 (図 4 の N1, D1) を示している。

まず、OOJ 記述環境では離散単位毎に追跡性を持つよう設計されているため図 4 の n1 と d1 を比較対象として得ることができる。そして前節で示した検証支援機構を用いて、分析者であり開発者でもある想定ユーザ自身が評価を行う。追跡先の編集も同時に行い、自身が保証できる相似性を設定する。他の離散単位にも同様の作業を行っていく。

構造における相似性の検証作業も必要であり、これを実現するために離散単位群抽出機構を用いて比較対象を得る。これら比較対象にも同様に検証支援機構を用いて相似性の検証作業を行い、構造の間における相似性の評価・保証をすることができる。

想定ユーザはこれらの作業を任意の離散単位と離散単位群に対して行い、分析記述の再現性の評価・保証を行う。

## 4. OOJ トレーサの実装

### 4.1 検証支援機構の実装

検証支援機構の記述画面を図 5 に示す。図 5 は分析記述と設計記述間の最小離散単位間 (図 5 の (A)) における比較を行っており、この比較対象は離散単位が持つ追跡性によって取得している。想定ユーザの特徴から簡潔な表現による定量化が適切だとし、相似性を 5 段階評価で行える

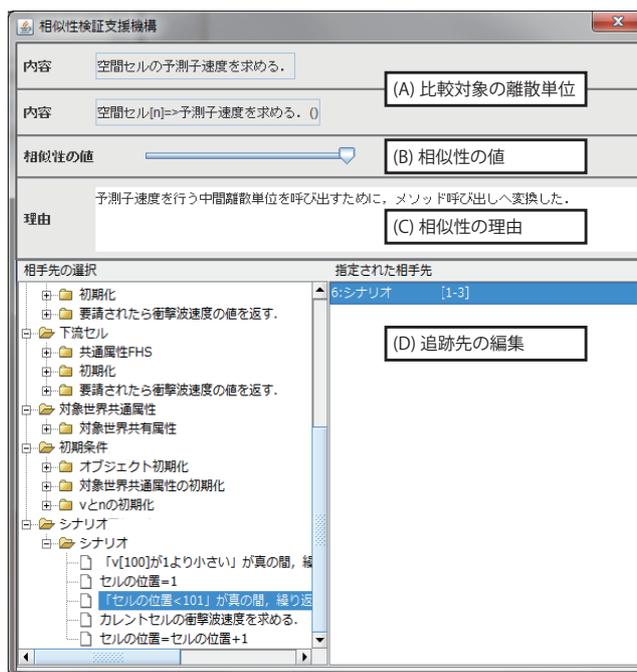


図 5 検証支援機構の記述画面

Fig. 5 A screen example of verification support editor.

ようにした (図 5 の (B)). 前章で示した「数値的な判断基準」の記述は図 5 の (C) で行い、図 5 の (D) では離散単位の指定で追跡先を設定できるよう実装した。

図 5-(A) では分析記述と設計記述の離散単位が一对一で表現されているが、一对多の表現も想定している。OOJ では分析記述から実装記述までの一貫した記述作業を行うため、検証作業も一貫して行うことが必要である。

### 4.2 離散単位群抽出機構の実装

離散単位群抽出機構の記述画面を図 6 に示す。各段階の記述は節点が離散単位である木構造で表現され、開始点と終了点となる離散単位を選択する (図 6 の (A)). 離散単位群抽出機構で対象とするのは相互作用情報伝達 (mp) であり、「中間離散単位と最小離散単位の構造化」 (図 6 の (B)), 「最小離散単位間のシーケンシャルな構造」 (図 6 の (C)), 「mp」 (図 6 の (D)) の 3 種類の相互関係である。これらは同じ離散単位で複数記述される場合もある。そこで 3 種類の相互関係を想定ユーザが記述した箇所を優先順位を付けて評価する。

### 4.3 V&V 環境として必要な機能

3.2 節で示した検証支援機構と離散単位群抽出機構から構成される OOJ トレーサの開発を行い、OOJ 記述環境と組み合わせることで、OOJ の目的である分析記述の再現性の定量化を実現するための基盤となることを示した。ただし、利用したユーザのコメントや関連研究との比較から V&V として利用するためには機能不足であることが分かった。具体例として以下の機能を挙げる。

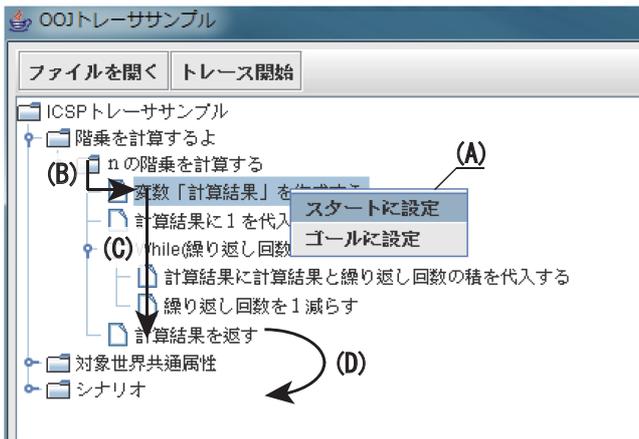


図 6 離散単位群抽出機構の記述画面

Fig. 6 A screen example of extraction mechanism

(1) 定量化した分析記述の再現性を GUI 表示する機能  
 (2) 検証項目を想定ユーザが判断して構築できる機能

上記の (1) の機能は離散構造化モデルで構造化されていることを利用し、離散グラフを用いて再現性を示す方法を検討している。並行して想定ユーザの意見も反映させる。

(2) は V&V の規格やガイドラインでは計算対象や計算方法に応じて検証項目と検証作業が設定されているからである。そのため多分野の V&V のガイドラインに合うように検証項目を選択できることが求められるからである。

## 5. OOJ 記述環境下の V&V 環境の提案と結論

本論文で提案する V&V 環境とは、図 3 または図 4 の検証支援機構と離散単位抽出機構の 2 つを内部構成とする OOJ トレーサである。両機構は既に実装されて OOJ トレーサとして OOJ 記述環境と連携して動いているが、未だ十分ではない。

また OOJ 記述環境下の V&V 環境には設計上の前提条件がいくつかあり、V&V 環境もその影響を受ける。

- (1) 記述環境も個人が使うだけなので、V&V 環境も個人使用を前提とする。
- (2) 個人利用であること。複数人の利用を前提としない。
  - (A) 分析・設計・実装の段階を全て 1 人で行う。
  - (B) V&V 環境も数千行の個人規模のプログラムの検証を目的とする。
- (3) 分析記述がプログラムに忠実に反映しているか否かについての妥当性の確認はユーザ自身が自分で行う。

結論としては、提案した V&V 環境は稼動しており、当初の設計結果は満たしている。しかし、今後の課題として 4.3 節で述べた機能以外にも、一貫相似性という制約条件を外して柔軟なトレーサ環境 [13] を運用できる仕組みの構築、また関連研究の探索を十分に行いたい。

## 参考文献

- [1] 池田陽祐, 三塚恵嗣, 加藤木和夫, 大木幹生, 上田賀一, 島山正行, UML との比較に基づくオブジェクト指向分析設計記述言語 OONJ の評価, 情報処理学会論文誌: 数理モデル化と応用 Vol.5, No.3, 63-78 (Sep. 2012).
- [2] 池田陽祐, 上田賀一, 島山正行, UML との比較評価に基づくオブジェクト指向分析設計記述言語 OONJ の記述技法の特徴, 情報処理学会論文誌: 数理モデル化と応用, 2012 年 9 月 4 日 受理. Vol.34, (Mar. 2013, 予定).
- [3] 吉田紀彦, 自動プログラミングハンドブック第 4 章「プログラム変換手法による自動プログラミング」, オーム社, pp. 109-120(1989).
- [4] 電子情報通信学会, [http://www.ieice-hbkb.org/files/01/01gun\\_12hen\\_06.pdf](http://www.ieice-hbkb.org/files/01/01gun_12hen_06.pdf) (accessed 2013-01-20).
- [5] ASME, “Guide for Verification and Validation in Computational Solid Mechanics”, V&V10-2006 .
- [6] ASME, “Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer ”, V&V20-2009.
- [7] What is Verification and Validation. , 2009-July-24, <http://www.nafems.org/publications/brows.buy/qa/vandv/access2013/01/10>.
- [8] 永松泰成, 島山正行, オブジェクト指向日本語一貫記述環境 OOJDE の開発, 情処研報, 01-SE-136, pp.25-32, (2002).
- [9] 沼崎隼一, 池田陽祐, 三塚恵嗣, 島山正行, オブジェクト指向一貫記述言語系 OOJ におけるトレーサの開発, 第 171 回 SE 研究会報告, 2011-SE-171-6, (Mar. 2011).
- [10] 島山 正行, 池田 陽祐, 三塚 恵嗣, 大木 幹生, 加藤木和夫, 上田 賀一, 離散構造化モデル記述言語系 OOJ の構築と効果的な利用法—分析からプログラムまでの一貫開発と V&V 評価実現の検討—, 第 92 回 MPS 研究会, 2012-MPS-92-O (Feb. 2013).
- [11] 池田 陽祐, 三塚 恵嗣, 上田 賀一, 島山正行, 実世界を直截に記述可能な OOJ 記述環境の開発とその利用効果, 第 92 回 MPS 研究会報告, 2012-MPS-92-E (Feb. 2013).
- [12] ISO/IEC 25000, Software engineering , Software product Quality Requirements and Evaluation (SQuaRE), Guide to SQuaRE, 2005-08-01.
- [13] 我妻智之 神谷慎吾 大平直宏 松下誠 楠本真二 井上克郎 「メタモデルに基づくトレーサビリティ技術の提案」, 信学技報 2005-09 KBSE2005-20 IEICE Technical Report.