

# V&V 評価の仕組みを組み込んだ OOJ 記述環境の開発と評価

池田 陽祐<sup>†1</sup> 大木 幹生<sup>†2</sup> 三塚 恵嗣<sup>†3</sup> 上田 賀一<sup>†4</sup> 島山 正行<sup>†1</sup>

**概要:** 本論文では IEEE Std 1012 に準拠した V&V 評価の仕組みを組み込んだ OOJ 記述環境の構築について報告する。OOJ 記述環境は記述言語系 OOJ を支援するための開発環境である。OOJ は離散・構造化モデルに基づいた設計をしており、個人規模や教育に対してのプログラム開発を対象としている。OOJ の開発過程は分析・設計・実装・プログラムの 4 つの段階に分けており、各段階では離散・構造化モデルに関わる作業を行う。ソフトウェア開発の品質評価の概念として V&V があり、それに基づいた品質評価の規格として IEEE Std 1012 がある。OOJ で行う品質評価作業は、OOJ の特性を利用したレビュー環境と各段階の作業で出力されるドキュメントを IEEE Std 1012 の V&V activity に関わるドキュメントに準拠することで V&V 評価を行える仕組みである。この仕組みを組み込んだ OOJ 記述環境により、一貫したプログラム開発と V&V 評価のための成果物を収集できる有効な環境を構築した。想定ユーザ評価では、大学院生を対象に OOJ 記述環境を用いた記述実験を行った。その結果、彼等は OOJ 記述環境を十分に使いこなしており、プログラム開発を行うことで V&V 評価のための成果物を得ることができた。以上から、V&V 評価の仕組みを組み込んだ OOJ 記述環境が V&V 評価を行うための有効な基盤になることが結論できた。

**キーワード:** V&V, OOJ, IEEE Std 1012-2004, IEEE Std 1012-2012

## A development and evaluation of OOJ descriptive environment that build in mechanism of V&V evaluation

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

**Abstract:** In the present paper, we will report the development of OOJ description environment that build in the V&V evaluation based on the IEEE Std 1012. The OOJ description environment is a development environment for supporting a descriptive language system OOJ. OOJ is designed based on the discreted and structured model. OOJ is applied to the program development of a personal scale. OOJ contains four inner stages; the analysis, design, implementation, and program stages. In each stage, the operations have been made using the discreted and structured model. As for the quality evaluation of OOJ, we have developed the review environment using the features of OOJ and the mechanism that V&V evaluation has been carried out by document that corresponds to a document of V&V activity in IEEE Std 1012. IEEE Std 1012 is a standard of the quality evaluation based on V&V. The OOJ description environment build in the mechanism of V&V evaluation, integral program development environment, the function to collect document for V&V evaluate. The user estimations have been performed for describe the OOJ descriptions by the graduate school students. As the results, the description environment have been confirmed to be sufficiently useful and collected document for V&V evaluation is partly valid. We have got the conclusion that the OOJ description environment is useful for the platform of V&V evaluation.

**Keywords:** V&V, OOJ, IEEE Std 1012-2004, IEEE Std 1012-2012

<sup>†1</sup> 現在, 茨城大学工学部  
Presently with Faculty of Engineering, Ibaraki University  
<sup>†2</sup> 現在, 群馬工業高等専門学校  
Presently with Technical Support Center for Education and

Research, Gunma National College of Technology  
<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

## 1. はじめに

現代においては社会のインフラとして至る所に電子機器や計算機が使用され、必要欠くべからざる製品として重要性を増している。特にその内部に組み込まれているソフトウェアに関しても量的な側面だけではなく、安全性や信頼性の面からもより高度な特性が求められるようになってきており、そのため多くの分野でプログラム開発<sup>\*1</sup>技術の重要性もニーズも飛躍的に増している。その結果プログラムは高度化、大規模化する一方、その品質保証の妥当性や品質評価の重要性もより深まっている。

それらの要請の一端に答えるべく、主として個人規模のプログラムの開発を目的とした記述言語系 OOJ [3]<sup>\*2</sup>とその記述環境の構築を行ってきた。その狙いの1つとして、当初の分析記述がプログラムに正しく反映する仕組みを作ることを目的としている。それは要求がプログラムへ正しく反映されていることを検証できることと言い換えられる [4],[5]。しかし、実際にプログラムの品質評価を行うには様々な規格と高度な技術が必要である。

この品質評価を行うための概念として V&V があり、その IEEE 標準規格として IEEE Std 1012-2012 [6] が存在する。この規格は個人規模や教育に対するプログラムの開発<sup>\*3</sup>には複雑であるため、より簡潔な IEEE Std 1012-2004 [7] を使用する。本論文では、IEEE Std 1012-2004 に準拠し、V&V 評価の基盤となる OOJ 記述環境の設計、開発、評価について述べる。

## 2. 離散・構造化モデルと OOJ

### 2.1 離散・構造化モデルの特徴

我々は、この離散化の手法とオブジェクト指向パラダイムを統合して**離散・構造化モデル**を構築した [3]。離散・構造化モデルは実世界に対する分析に特化した OO モデルのサブセットでもあり、オブジェクト指向<sup>\*4</sup>モデルの特長を失っていない。それゆえ、実世界に対する離散化モデルの延長線上にあるので想定ユーザ (2.3 節を参照) にも理解し易く扱い易いモデルであり、OO モデルでもあるため分析からプログラムまで一貫して使える特徴を持つ [3]。

### 2.2 離散・構造化モデルの記述方法:OOJ と OOJ 記述環境

離散・構造化モデルの記述に特化した記述言語として OOJ があり、プログラムまでの開発過程を段階的に分けて体系化している [3]。各段階にはサブ言語があり、これらのサブ言語間には相似性<sup>\*5</sup>を保持するように規定されており、記述上での追跡可能性を実現している。これらのサブ言語からなる記述言語系が OOJ であり、これを支援するための環境が OOJ 記述環境である。

### 2.3 想定ユーザ

OOJ の想定ユーザは自身の専門分野の情報を十分提供すれば計算結果が得られることを望む数値計算ユーザであったが、OOJ の教育適用の効果的な適用 [3] 結果を受けて個人規模や教育に対するプログラムの開発<sup>\*6</sup>を行う理工系の学部生、大学院生を対象とする。

### 2.4 OOJ を用いた V&V 評価機構の設計

品質評価作業 [8] の概念として V&V がある。V&V は当初の要求仕様や設計が正確な手順と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) を指す概念と、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妥当性の確認 (Validation) を指す概念である。V&V に基づいた規格やガイドライン [7],[9],[10],[11] が提案されている。ただし、これらの規格はエンタープライズ向けの開発や大規模な研究開発を対象としており、個人規模におけるプログラムの開発を対象としていない。

そこで V&V の IEEE 標準の規格である IEEE Std 1012-2004 [7] に準拠した **V&V 評価機構** を OOJ 上に組み込むことで個人規模や教育に対するプログラムの開発を対象とするプログラムの開発環境の構築を目指す方針とした。

IEEE Std 1012-2004 では、Life cycle processes が規定されており、それに対応して V&V activity が規定されている。また、V&V activity は4段階に分かれている software integrity level に対応した V&V task が規定されている。

個人規模や教育に対するプログラムの開発を対象としているため Life cycle processes を Development のみに限定し、V&V activity を要求 V&V activity, 設計 V&V activity, 実装 V&V activity, テスト V&V activity に限定した<sup>\*7</sup>。また、social loss が低いプログラムを開発対象としているため software integrity level は最も低いレベル 1 を対象とする。IEEE Std 1012-2004 の Table2 [7] から V&V

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。この点の詳細な比較は参考文献 [1], [2] のいずれも表 2 にある。また本論文では、プログラミング言語 (必要に応じて **PL** と略す。) を用いた記述のみを **プログラム** と呼び、それ以外はすべて単に **記述** と表現する。

<sup>\*2</sup> Object Oriented Japanese の略

<sup>\*3</sup> 本論文では管理者と開発者と利用者が同一人物であるプログラムの開発として用いる。

<sup>\*4</sup> 以降、必要に応じて OO と略す。

<sup>\*5</sup> 本論文の「相似性」という用語の概念は、ある段階の離散単位と次の段階の離散単位が『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』を指す用語であると定義する [3]。

<sup>\*6</sup> 本論文では自身が使用する目的のプログラムを指す。すなわち開発者とユーザが同一人物である。

<sup>\*7</sup> 開発者と利用者が同一人物であるからコンセプト V&V activity, Installation/checkout V&V activity は対応していない。

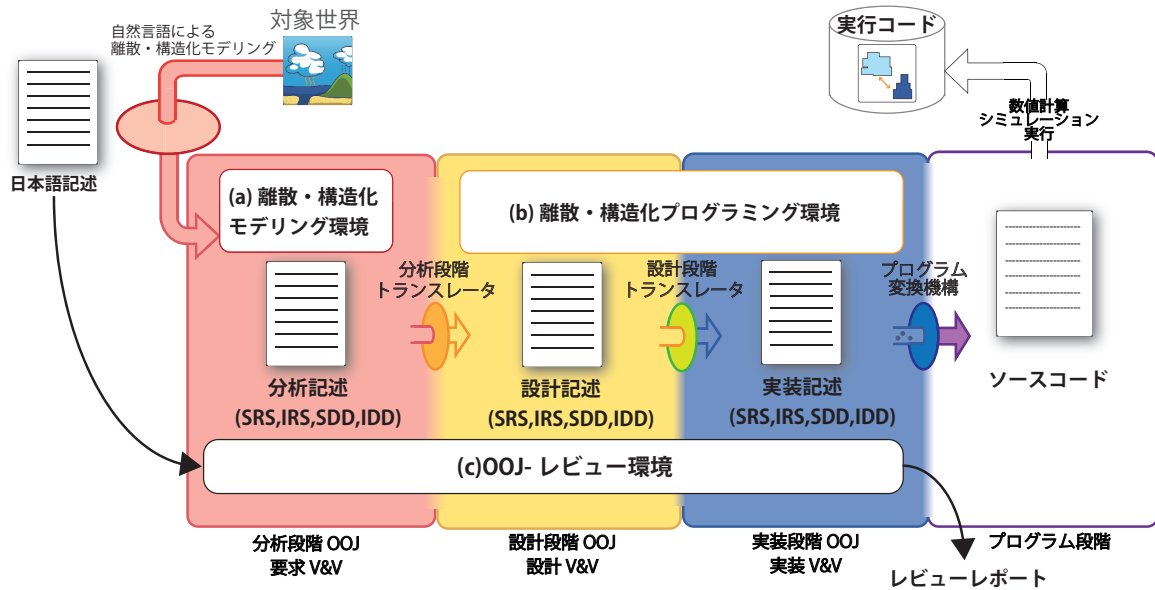


図 1 OOJ 記述環境のシステム構成

Fig. 1 System constitutions of OOJ description environment.

評価機構の V&V task を取捨選択する。その結果、OOJ を V&V 評価機構で対応するのは、各種の Integration V&V test generation と System V&V test generation の 2 つであることが分かる。

ただし、Integration test, System test の 2 種類はテストに関わる手法であるが、テスト作業には様々な種類や方法があり複雑であるため [8]、本研究では V&V 評価の基盤となる仕組みを実現する。

## 2.5 V&V 評価の仕組みの機能

前節で示した V&V task にはそれぞれ入出力のドキュメントがあるが、個人規模や教育に対してのプログラム開発から User documentation, Criticality task report, Task report(s) Criticality analysis は作成する必要が無い。Test plans, Test cases, System test procedure, Integration test procedure はテストの手法やプロセスに深く依存するため本論文では取り扱わない。そのため V&V 評価の仕組みで必要なドキュメントは、SRS \*8, IRS \*9, SDD \*10, IDD \*11, Source code, Executable code の 6 つの入力ドキュメントであることが分かる。

## 3. OOJ 記述環境の全体構成と概念設計

### 3.1 V&V 評価の仕組みを組み込んだ OOJ 記述環境の全体構成

前節で示した OOJ 記述環境の全体構成を図 1 に示す。OOJ 記述環境は離散・構造化モデリング環境 (図 1-(a))

と離散・構造化プログラミング環境 (図 1-(b))、OOJ-レビュー環境 (図 1-(c)) の 3 種類の環境から成る。

離散・構造化モデリング環境 (図 1-(a)) は OOJ の分析段階に対応し、対象世界を日本語で表現した日本語記述に対して離散・構造化モデリングを行う環境である。離散・構造化プログラミング環境 (図 1-(b)) は OOJ の設計段階と実装段階に対応し、分析記述をプログラムへ変換するための環境であり、最終的には Java, Fortran90, C++ のソースコードが出力される。一方のレビュー環境 (図 1-(c)) は、開発環境で蓄積された分析記述、設計記述、実装記述を利用したレビューを行う環境である。

2.5 節で示した SRS や SDD の規定されているテンプレート [12] と表現は異なるが、OOJ の分析記述には SRS, IRS, SDD, IDD の個人規模や教育に対してのプログラム開発として必要な内容は含まれている [1],[2]。設計記述、実装記述は、分析記述と表現は異なるが内容自体は同等であるため分析記述と同様である。ソースコードと実行コードは実装記述を変換・実行することで得られる。

### 3.2 離散・構造化モデリング環境の概念設計

OOJ では想定ユーザの特徴や個人規模や教育に対してのプログラム開発であることから理解性や書き易さが重要であると判断し、開発の始まりには日本語を用いて対象世界を記述した日本語記述を用いる。その際には特殊な表記の規定はせず、想定ユーザ自身で理解でき対象世界の内容と開発が書かれていれば良い。そして、この日本語記述に対して離散・構造化モデリングを適用できるように離散・構造化モデリング環境を設計する。

離散・構造化モデリング環境を用いた作業手順を示す。

\*8 Software Requirements Specification の略記 [12].

\*9 Interface Requirements Specification の略記 [13].

\*10 Software Design Description の略記 [14].

\*11 Interface Design Document の略記 [15].

始めに日本語記述内の単語や文を離散単位として抽出する。そして抽出した離散単位に対して相互関係を付与する。離散・構造化モデリング環境では、日本語記述に対して離散・構造化モデリングを離散化と構造化に分けて行う。離散・構造化モデリングを行うことで、例えば SRS [12] で規定されている Functions, Objects, Feature が明示的に記述される。以上のように日本語を用いた理解性と書き易さを利用し、離散・構造化モデリングを形式化した操作を実現し、個人規模や教育に対してのプログラム開発に必要な SRS の項目が記述される。

### 3.3 離散・構造化プログラミング環境の概念設計

分析記述では、離散・構造化モデルを用いて対象世界が表現されている。また、離散・構造化モデルはオブジェクト指向モデルを特化したモデルでもある。そのため各離散単位をプログラム内のデータや処理へ変換することでプログラムへの変換が可能である。以上の作業を**離散・構造化プログラミング**と本論文では呼ぶ。

離散・構造化プログラミングは2つの過程に別れており、1つ目は分析記述内の各離散単位に「プログラムへの変換情報」であるデータ型やアクセス修飾子、処理の種類を付与する作業である。2つ目は、変換先のデータ構造や処理に適合するように文を整形する作業である。プログラムへの変換情報は、変換先のプログラム言語である Java, C++, Fortran90 から抽出した共通仕様を基盤としており、プログラム変換機構を用いた変換処理と関連している。

### 3.4 OOJ-レビュー環境の概念設計

レビュー技法 [8] には様々な技法があるが想定ユーザが実施可能であるためには実施コストが低いレビュー技法でなければならないが、2.5 節で示したように客観的に評価できるレビュー結果の出力は必要である。つまり実施コストが低くレビュー結果を残せるレビュー技法が必要である。

また、OOJ はその特性により分析記述、設計記述、実装記述は追跡性を持つため分析記述から実装記述までの対応関係を静的に評価することができる [3]\*<sup>12</sup>。対応関係を静的に評価する際には自然と離散単位に注目する。離散単位内には、記述内容が書かれているため離散単位の記述内容を静的に評価するレビューができる。全ての離散単位に対してレビューを行うことで、各記述間のレビューができる。このように OOJ の特性である追跡性を主体としたレビューを行うための環境が OOJ-レビュー環境である。

## 4. OOJ 記述環境の実装

### 4.1 離散・構造化モデリング環境の実装

離散化の作業では、離散単位とする単語や文を選択し、

離散単位の種類である「モノ・概念」、「属性」、「振る舞い」、「相互関係」のいずれかとして抽出する。抽出された単語や文は離散単位の種類と対応した色で装飾されることで、離散単位として何を抽出したかを文中で確認できる。

構造化の作業は、抽出された離散単位に OOJ の集約階層と相互関係を付与していく。まず、離散単位を OOJ の集約階層に位置づける。集約階層には最大・中間・最小の3階層が存在し、何れかの階層へ位置づけなければならない。次に相互関係として抽出された離散単位を用いた構造化を行う。相互関係は「種類」と「向き」、「相手先」の3項目から構成される。

### 4.2 離散・構造化プログラミング環境とプログラム変換機構の実装

3.2 節でも述べたように離散・構造化プログラミングは離散単位にプログラムへの変換情報を追加する作業と文の整形作業である。

まず、プログラムへの変換情報としてデータ型、アクセス修飾子、命令の種類を一覧の中から選択することで追加していく。変換情報を追加することでそれに対応した書き方が表示され、表示された書き方に従って文を整形する。これらの処理を全ての離散単位に対して行う。

プログラム変換機構は、Java, C++, Fortran90 のプログラムへ変換するために unfold/fold 方式 [16] を用いて実装している。各離散単位には複数の対応関係が存在しているが、実際には想定ユーザが選択することで一意的な対応関係となる。

### 4.3 OOJ-レビュー環境の詳細設計

3.4 節では分析記述、設計記述、実装記述の利用を前提とし、OOJ の特性である追跡性を利用した OOJ-レビュー環境を構築した。

OOJ-レビュー環境では分析段階、設計段階、実装段階の3段階を1画面上に表示しており、個々の離散単位と全体の双方が分かり易いように集約関係をエッジとしたツリーとして表現している。対応関係の評価や発見事項については編集画面を用いて記載する。また、対応関係を評価する過程で内容の矛盾や不適切な表現等を修正できるように離散単位の単語や文の編集を可能にしている。このように OOJ の追跡性から離散・構造化モデルの手法を利用することで OOJ レビュー環境を実現している。

## 5. OOJ 記述環境による記述実験とその効果

### 5.1 記述実験の前提

OOJ 記述環境の利用効果を確認するために情報工学専攻(修士)の大学院生2人に対して OOJ 記述環境を用いた Java プログラムの開発を行なってもらった。

評価方法は実装段階まで行ったあとに聞き取り調査を行

\*<sup>12</sup> これは「要求が反映されていることを確認できる」ことであり、レビューで期待される効果の1つでもある [8]。

い、その後レビューに同席し議論や聞き取り調査を行った。

## 5.2 離散・構造化モデリング環境の効果

被験者の院生は前提である日本語記述を知識や資料を参照しながら容易に作成した。そして作成された日本語記述に対して離散・構造化モデリング環境を利用してもらい回答を得た。

被験者からの回答から離散・構造化モデリング環境の操作方法は離散・構造化モデルと同様のイメージであることが分かった。

以上のことから大学院生においては離散・構造化モデリング環境を利用することで、日本語記述に対して想定通りの離散・構造化モデリングができ、SRS, IRS, SDD, IDD に一部に相当する分析記述を作成することができた。

## 5.3 離散・構造化プログラミング環境の効果

4.2 節で示した離散・構造化プログラミングの作業内容である「変換情報追加作業」と「文の整形作業」毎に聞き取り調査を行い回答を得た。

変換情報追加作業については、オブジェクト指向プログラム言語を利用した経験があれば操作が容易であることが分かった。そのため少なくとも1つのプログラム言語を利用した経験のある想定ユーザならば同様の効果が期待できるが、離散単位の数が増える原因となる作業負担や操作性に関して今後の課題も分かった。

変換されたプログラムのコメントを基に実装段階の記述を確認することで実装記述内の離散単位とプログラムの処理との対応関係を確認できた。対応関係から設計記述、実装記述ともに分析記述と同程度のSRS, IRS, SDD, IDDの内容が書かれていることが分かった。

これらのことから離散・構造化プログラミングを利用することで、多少の作業負担はあるが、離散・構造化モデルをプログラムへ変換することが可能であることが分かった。また、実装記述とプログラム間も容易に対応関係を確認でき、設計記述と実装記述が分析記述と対応しているため分析記述と同じ程度のSRS, IRS, SDD, IRSを含むことが分かった。

## 5.4 OOJ-レビュー環境の効果

OOJ-レビュー環境では少人数でのレビューを想定しているため評価実験は開発者である被験者の大学院生1人を含め3人で行った。

その結果、OOJ-レビュー環境を用いることで欠陥の発見やリファクタリングと同等の効果を行え、レビュー環境として最低限の効果を持っていることが分かった。また、分析記述の内容が設計段階以降の各段階の記述へ適切に反映されていることを確認することもできた。以上の発見事項はレビューレポート内に含まれている。

以上のことから個人規模や教育に対してのプログラム開発においてOOJ-レビュー環境を利用することで、レビュー未経験者でも従来のピア・デスク・チェックと同等のレビュー効果を実現することができた。

## 6. 関連研究との比較と考察

### 6.1 MATLAB との比較と考察

MATLAB [17] は、豊富なライブラリとユーザに適したインターフェイスを用いたソフトウェアであり、モデル検査を用いたモデルベースデザイン [18] を用いて確認、検証、テストの作業をV&Vであるとしている。

一方、OOJは個人規模や教育に対してのプログラム開発を対象としており、V&Vに関してはドキュメントの収集・管理を主体としており異なる。以上のことよりOOJとMATLABは用途や目的に応じて使い分けるべきである。また、さらに想定ユーザの多くが将来MATLABを利用すると推測されるためMATLABへのプログラム変換を実現することでOOJの有用性が高まると期待できる。

### 6.2 MDA との比較と考察

MDAは、ソフトウェア工学におけるモデル駆動開発 (Model Driven Developments 以降、MDD と略記) の手法の1つであり、OMGで正式に規格として認められている [19]。多くの企業で行われる製品開発や組込みシステムの開発、いわゆる業務処理システムの開発で使用されている。MDAではモデル検査技術を用いた品質評価技術の開発が行われており、ソフトウェアへの適用に関しても様々な技術が提案されている [20]。

ただし、MDAを利用して開発する場合やモデル検査技術を利用するには高度な経験と技術が必要であり、想定ユーザの利用には適していないといえる。

### 6.3 OOJ 記述環境の総合評価と考察

OOJ記述環境は、離散・構造化モデルとOOJの追跡性を利用することで分析記述の作成からレビューまでを一貫して行えるプログラム開発環境を実現した。また、記述実験からプログラム開発過程によりV&V評価に必要なSRS, SDD, IRS, IDDに相当する分析記述、設計記述、実装記述とソースコード、実行コード、を得られ、静的な品質評価作業のレビュー結果としてレビューレポートも得られた。以上のことからOOJ記述環境は個人規模と教育に対してのV&V評価の基盤となるといえる。

## 7. 結論と今後の課題

本論文ではV&V評価の仕組みの設計とそれらを組み込んだプログラム開発環境としてOOJ記述環境を構築した。このOOJ記述環境を利用した記述実験の対象ユーザにおいては、分析記述の作成からレビューまでを一貫して行う

ことでV&Vに必要なドキュメントを収集することができた。これにより個人規模と教育に対してのプログラム開発と品質評価作業を行うことができたといえる。また、IEEE Std 1012-2004で規定されたドキュメントを作成・管理できることが分かった。以上のことから当初の目的であるV&V評価の仕組みを組み込んだOOJ記述環境の開発と評価を実現することができた。

今後の課題として、分析記述とSRS, IRS, SDD, IDDとの比較による詳細化と検証, 実用化の観点から記述例の蓄積, 記述実験, 詳細な記述ノウハウやライブラリ等の蓄積がある。

## 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 5(3), 63-78, 2012-09-28.
- [2] 池田陽祐, 三塚恵嗣, 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 6(1), 156-171, 2013-03-12.
- [3] 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 OJ 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—, 情報処理学会論文誌: 数理モデル化と応用, 発行予定(2013年).
- [4] 電子情報通信学会, 入手先 ([http://www.ieice-hbkb.org/files/01/01gun\\_12hen\\_06.pdf](http://www.ieice-hbkb.org/files/01/01gun_12hen_06.pdf)), (accessed 2013-01-30).
- [5] ISO/IEC 25000: Software Engineering, Software product Quality Requirements and Evaluation (SQuaRE), Guide to SQuaRE, 2005.
- [6] IEEE Std 1012-2012-IEEE Standard for System and Software Verification and Validation, IEEE Standard Association, 2012.
- [7] IEEE Std 1012-2004-IEEE Standard for System and Software Verification and Validation, IEEE Standard Association, 2005.
- [8] 独立行政法人情報処理推進機構 (IPA), 高信頼化ソフトウェアのための開発手法ガイドブック—予防と検証の事例を中心に—, ソフトウェア・エンジニアリング・センター (SEC), 2011年6月1日.
- [9] 日本計算工学会編, 工学シミュレーションの標準手順, JSCESS-HQC002:2011 A Model Procedure for Engineering Simulation, 日本計算工学会, 2011年5月31日.
- [10] ASME, “Guide for verification and Validation in Computational Solid Mechanics”, V&V10-2006.
- [11] ASME, “Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer”, V&V20-2009.
- [12] IEEE Std 830-1998-IEEE Recommended Practice for Software Requirements Specifications, IEEE Standard Association, 1998.
- [13] IEEE Std 100-2000-IEEE 100 The Authoritative Dictionary of IEEE Standards Terms Seventh Edition, IEEE Standard Association, 2000.
- [14] IEEE Std 1016-2009-IEEE Standard for Information Technology—Systems Design—Software Design Descriptions, IEEE Standard Association, 2009.
- [15] IEEE Std 24765-2010-Systems and software engineering – Vocabulary, ISO/IEC, 2010.
- [16] 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).
- [17] MathWorks MATLAB(online), 入手先 (<http://www.mathworks.co.jp/products/matlab/>), (accessed 2013-08-26).
- [18] MathWorks MATLAB(online), 入手先 (<http://www.mathworks.co.jp/jp/help/simulink/gs/model-based-design.html>), (accessed 2013-08-26).
- [19] Object Management Group, MDA Specifications(online), 入手先 (<http://www.omg.org/mda/specs.htm>), (accessed 2013-01-30).
- [20] 片山卓也, 高信頼性組み込みソフトウェア開発のための技術課題, 情報処理, pp.488-490, 2006年47巻5号.