

飛行船制御を題材とした プロジェクト型ソフトウェア開発実習

沢田 篤史^{†1} 小林 隆志^{†2} 金子 伸幸^{†2,*1}
中道 上^{†1} 大久保 弘崇^{†3} 山本 晋一郎^{†3}

本論文では、我々が先導的 IT スペシャリスト育成推進プログラムの一環として実施中のプロジェクト型ソフトウェア開発実習の教育プログラムと教材開発について説明する。本実習では、軟式飛行船制御の組み込みソフトウェア開発を題材に、大学院修士課程 1 年生がチーム開発を行う。本実習の主目標は、管理されたプロジェクト開発を経験することとソフトウェア品質向上における検証技術の重要性を理解することである。本論文では、この狙いを持って設計した実習プログラムと、開発した教材について紹介し、半年間の実習を実施した結果得られた学習成果と課題について説明する。

A Project Based Learning Using Airship Control Software Development

ATSUSHI SAWADA,^{†1} TAKASHI KOBAYASHI,^{†2}
NOBUYUKI KANEKO,^{†2,*1} NOBORU NAKAMICHI,^{†1}
HIROTAKA OHKUBO^{†3} and SHIN-ICHIRO YAMAMOTO^{†3}

This article presents how we have designed a project based learning (PBL) class for graduated students in the Leading IT Specialist Education Program of MEXT. Master course students form several development teams to struggle with the development of embedded software for airship control in our class. The students of this class are expected to touch with the real experience of team development in a disciplined project and to have deep understanding on the importance of verification and validation technology throughout the development of quality software. This article overviews the design of and teaching materials for our PBL class. Discussions on the results and lessons learned from the half-year operation of this class are also presented.

1. はじめに

我々は、文部科学省が進める先導的 IT スペシャリスト育成推進プログラム^{*1}のうちの一点のメンバとして 2006 年度から活動してきた。この取り組みの中で、我々はソフトウェア工学に関する基礎的な技術力、それら技術を実践する力、変化へ適応する力を養成するための教育カリキュラムを設計し、実際に運用している。

本論文では、この教育カリキュラムの中で実践力を養うために設計したプロジェクト型開発実習 (PBL: Project Based Learning) について説明する。本 PBL では、軟式飛行船制御の組み込みソフトウェア開発を題材に、大学院の修士課程 1 年生がチーム開発を行う。本 PBL の主目標は、管理されたプロジェクト開発を経験することとソフトウェア品質向上における検証技術の重要性を理解することである。本論文では、この狙いを持って設計した実習教育の内容と、開発した教材について紹介し、2007 年度後期に行った実習の結果得られた学習成果と課題について考察する。

以下、2 章では、我々が実施している先導的 IT 技術者教育カリキュラムの概要と、その中で本 PBL の位置付けについて説明する。次いで 3 章では、本 PBL の学習目標とそれを達成するために設計・実装した教育プログラムと教材について説明する。4 章では、2007 年度後期に行った実習の結果得られた教育効果についてまとめる。さらに、5 章で関連する取り組みとの比較に基づき、本 PBL の問題点や課題について考察した後、6 章で本論文のまとめをする。

2. IT 教育カリキュラムと PBL の位置付け

開発型の実習は、専門知識や技術を実践する能力を養うのに効果的であるとされ、先導的

†1 南山大学

Nanzan University

†2 名古屋大学

Nagoya University

†3 愛知県立大学

Aichi Prefectural University

*1 現在、株式会社ネットレックス

Presently with NETREQS Corporation

本論文は、情報処理学会組込みシステムシンポジウム 2008 にて発表した同題目の論文¹⁾を、シンポジウムでの議論を参考に改訂したものである。

*1 http://www.mext.go.jp/a_menu/koutou/it/index.htm

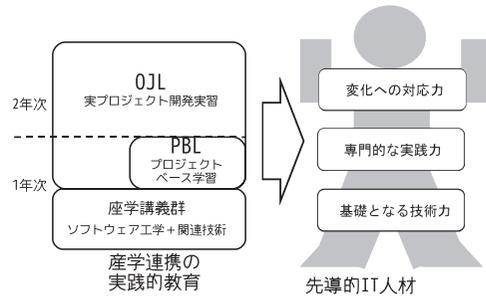


図 1 OCEAN の概要
Fig. 1 Overview of the OCEAN program.

IT スペシャリスト育成推進プログラムでも各拠点が PBL 型の教育プログラムを取り入れている²⁾。

2.1 OCEAN のカリキュラム

本論文で我々が提案する PBL は、先導的 IT スペシャリスト育成推進プログラムの中で、名古屋大学を中心とし、南山大学、愛知県立大学、静岡大学、ほか連携企業 6 社が参画する「OJL による最先端技術適応能力を持つ IT 人材育成拠点の形成」(OCEAN: On the job learning Centered Education for Advanced engiNeers)^{*1}と銘打つ教育プロジェクトにおけるカリキュラムの一要素である。

OCEAN の概要を図 1 に示す。OCEAN では高度な専門性を持つ IT 人材に求められる素養を、ソフトウェア工学に関する基礎的な技術力とその実践力、変化への適応力と定義し、座学での講義群に、PBL および OJL (On the Job Learning) という開発実習型の科目を組み入れたカリキュラムによって、大学院修士課程の学生へこれらの素養を教授することを狙っている。

座学での講義は修士課程の 1 年目に配置される。表 1 に示すとおり、ソフトウェア工学を中心とした技術要素について、学部レベルのソフトウェア工学講義に比べ、深く掘り下げた内容を教授する。各科目半期 1 講時ずつの授業時間 (1.5 時間 × 15 回の 22.5 時間) を割り当て、ソフトウェア工学のカリキュラムガイドライン SE2004³⁾ の知識体系のうちコンピュータサイエンス基礎と数学基礎を除く大項目を全科目でほぼカバーするよう設計した。

表 1 座学講義一覧

Table 1 Lectures in the OCEAN program at a glance.

科目名 (各 22.5 時間)	SE2004 知識体系大項目*									
	PRF	MAA	DES	VAV	EVO	PRO	QUA	MGT	SAS	
IT 技術倫理と社会										
ソフトウェア要求工学										
ソフトウェアアーキテクチャ										
ソフトウェア設計技術										
正当性検証と妥当性確認										
ソフトウェアモジュール化技術										
ソフトウェアプロジェクト管理										
コードリーディング										
ソフトウェアレビュー										
IT ネットワーク										net **
組み込みシステム開発技術 I										emb **
組み込みシステム開発技術 II										emb **
分散システム開発技術										net, inf **

— 講義の中心的内容として深く講述, — 講義に関連する話題として講述

* PRF: 専門職業倫理, MAA: モデルと分析, DES: 設計, VAV: V&V, EVO: 保守・発展, PRO: プロセス, QUA: 品質, MGT: プロジェクト管理, SAS: 専門領域知識

** net: ネットワーク中心システム, emb: 組み込み・リアルタイムシステム, inf: 情報システム

カリキュラムの中核をなすのは OJL で、我々 OCEAN が新たに提案する教育手法である。連携企業から実製品レベルの開発課題の提供を受け、学生と大学教員、企業からのプロジェクトマネージャおよび開発担当者が協力し、1 年半 (修士 1 年後期からの 3 半期) をかけて開発にあたる。OJL では、それぞれの開発課題での経験を通じ、ソフトウェア工学に関する知識を深化させるとともに、変化への適応力を養う。1 年半の OJL 実施期間のうち、修士 1 年後期は座学での講義や PBL と並行して行うことから、開発問題の概要や背景知識の理解などにあて、本格的な課題への取り組みは修士 2 年次の 1 年間に行う。

一方、PBL の役割は、座学での講義群と OJL との橋渡しで、ソフトウェア工学に関する知識を開発に適用する実践力の養成である。

2.2 PBL の位置付けと目的

OCEAN カリキュラムにおける PBL の役割は、OJL で実製品レベルの開発課題に取り組む前段階として、座学で学んだソフトウェア工学の知識を開発に適用する実践力を、チームによるソフトウェア開発の経験を通じて養うことである。

我々は全体カリキュラムの検討に先立ち、連携企業の協力のもと、OJL を試行的に実施した。その結果、チームでの開発経験や CASE ツールの活用経験、ソフトウェア品質に対す

*1 <http://www.ocean.is.nagoya-u.ac.jp/>

る考え方については、企業の求めるレベルと学生のレベルとの間でギャップが大きく、OJL への本格的な着手前に補う必要が認められた。

これら不足する知識や経験については、実際の開発経験に基づいて身につけるのが効果的であるという判断から、我々は次の学習目標を掲げた実習科目を PBL としてカリキュラムに取り込むこととした。

- 管理されたプロジェクトでの開発を経験し、チームによる問題解決作業と個人の役割を理解する。
- ソフトウェアプロセスにおける開発作業を実際に経験し、また CASE ツール活用の効果について理解する。
- ソフトウェアの品質向上における検証技術の重要性を理解する。

本 PBL は OJL の先行段階の実習であるという位置付けから、短い期間（修士 1 年次の後期）で完結できることが求められる。つまり、上記の学習目標を短時間で効果的に達成することを主眼とした、比較的小規模な題材を選定する必要がある。

3. 飛行船制御ソフトウェアを題材とした PBL

我々は、前章に述べたような目的と制約条件のもとで PBL の設計を行った。図 2 には、2.2 節で述べた学習目標および制約と、本 PBL の設計判断との関係を示す。

開発課題の検討にあたっては、学部学生の教育でロボットなどの組み込みシステムを課題とした取り組みにおいて、特に学習意欲維持の観点で良好な学習効果が報告されている^{(4),(5)}ことから、組み込みソフトウェアを対象に調査を行った。いくつかの候補について検討した結果、情報処理学会組込みシステムシンポジウムの特別企画 MDD ロボットチャレンジ^{(6),(7)}でも取り上げられている軟式飛行船（図 3）の制御ソフトウェアを題材として選定した。MDD

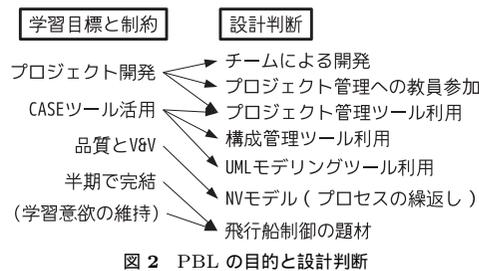


図 2 PBL の目的と設計判断
Fig. 2 Objectives and design rationales of our PBL.

ロボットチャレンジでは飛行船の自律制御という課題に取り組むが、半期という短い期間で完結する必要があることから、人手による遠隔操縦ソフトウェアの開発と機能改善に焦点をあて、教材と実習プログラムの設計を行った。

本 PBL における開発プロセスは、ソフトウェア品質保持の重要性を理解させることを目的に、NV モデルと呼ぶ 3 つのフェーズからなる構成として設計されている。NV モデルの詳細については 3.2 節で説明する。また、開発作業では、UML モデリングや構成管理のための CASE ツール、プロジェクト管理ツール、問題追跡ツールなどをそれぞれ必要に応じて利用できるような環境整備をした。

本章では以下、開発対象として準備した飛行船システムの概要について述べた後、PBL での実習手順について説明する。

3.1 開発対象システムの概要

図 4 に飛行船操縦システムの概要を示す。本システムは、飛行船の駆動を司る Air_MPU、飛行船と地上局との無線通信を司る Rf_MPU、ユーザの操作入力を飛行船制御命令に変換する UI_MPU という 3 つのサブシステムからなる。

飛行船は水平方向（前進・後退・左右回転）を制御するために搭載された左右 1 個ずつのプロペラと、上下方向（上昇・下降）を制御する 1 個のプロペラによって駆動される。飛行船に搭載される Air_MPU には、与えられた命令に従って 3 個のプロペラをそれぞれ回転させる機能のほか、超音波探信により床からの高度を計測する機能、ジャイロセンサにより水平方向の回転角速度を計測する機能が備えられる。

Rf_MPU の役割は、UI_MPU からシリアル通信を介して与えられる飛行船制御命令ピッ



図 3 PBL の題材（軟式飛行船）
Fig. 3 Blimp airship hardware.

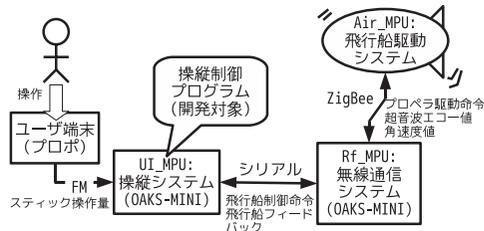


図 4 飛行船操縦システムの概要
Fig. 4 Blimp airship control system.

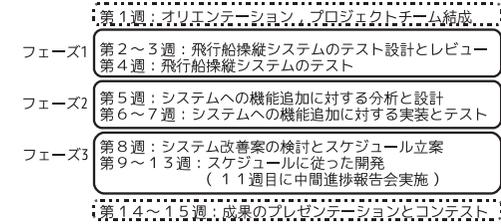


図 5 PBL の進行スケジュール
Fig. 5 PBL course schedule.

ト列を 3 個のプロペラに対する回転命令に分解し、ZigBee 通信を介して Air_MPU に送信することである。また、Air_MPU から受信する超音波探信のエコーバック値とジャイロセンサの角速度値を、飛行船フィードバック情報のビット列にとりまとめて UI_MPU に返送する機能も備える。

UI_MPU はユーザからの操作入力を受け、それに応じた飛行船制御命令を生成して Rf_MPU に送信する。ユーザの操作は各種模型のラジオコントロールに用いるプロボ端末によって行われる。端末のスティック操作量が FM 通信を介して UI_MPU に入力される。

これら 3 つのサブシステムのうち、UI_MPU 上で動作し飛行船制御命令を生成する操縦制御プログラムが本 PBL での開発対象である。UI_MPU はオクス電子社製の組み込みボード OAKS16-MINI^{*1} 上に実現されており、C 言語によるクロス開発で作業が進められる。操縦制御プログラムがプロボ端末から操作量を読み取るドライバとそのインターフェースは与えられているので、

- プロボ端末操作の飛行船制御命令への割当て機能
- 飛行船フィードバック（高度，角速度）を利用した操縦支援機能を実現するプログラムを開発することになる。

3.2 実習のプロセス

本 PBL は、5~6 名程度の学生と、1 名の教員または教員相当のアドバイザによってチームが構成されることを想定して設計した。アドバイザの役割は主にプロジェクトの進捗管理である。図 5 に本 PBL の実施スケジュールを示す。実習は、1 週間につき連続する 2 講時（3 時間）を、15 週実施することを前提に設計した。実際の開発作業は 12 回分で、前後に

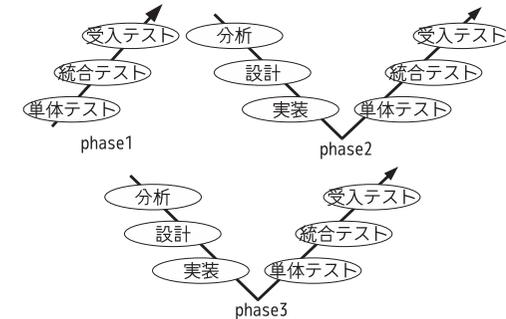


図 6 PBL プロセスの NV モデル
Fig. 6 The “NV” model of our PBL process.

導入（1 回）と成果発表（2 回）のための時間を設けた。開発は 3 つのフェーズに分割されており、それぞれ、

- フェーズ 1：テストケースの設計とテスト実行
- フェーズ 2：与えられた機能拡張仕様に基づくシステムの再開発
- フェーズ 3：機能拡張仕様の立案と、管理されたプロジェクトに基づくシステムの再開発

を目的とする。

ソフトウェア開発プロセスの V 字モデルとの対応関係に基づいて全体のプロセスを見ると、図 6 に示すように、V 字の右半分にあたる検証のプロセスを 1 回と V 字 2 回の繰返しから構成される。その形状から我々はこれを NV モデルと呼んでいる。

フェーズ 1 プロセスの詳細を図 7 に示す。フェーズ 1 では、UI_MPU 上で動作する操縦

*1 http://www.oaks-ele.com/oaks_board/oaks16_mini/

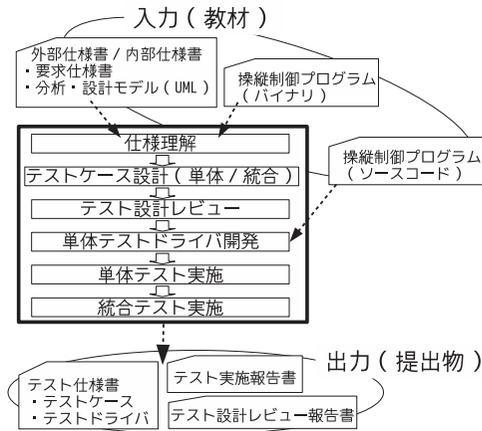


図7 フェーズ1のプロセス
Fig. 7 Phase 1 process.

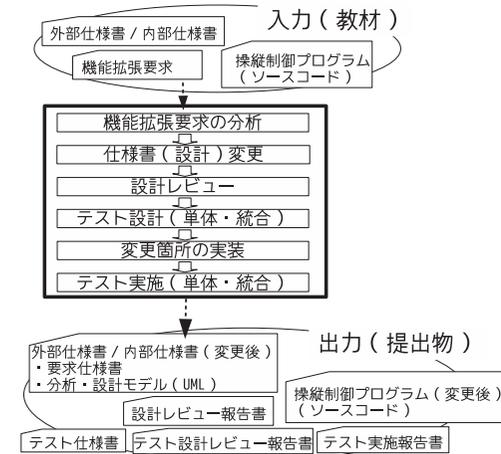


図8 フェーズ2のプロセス
Fig. 8 Phase 2 process.

制御プログラムの外部仕様書と内部仕様書に基づき、単体テストと統合（システム）テスト^{*1}のテストケース設計とテストの実施を行う。学生にはテスト対象となる操縦制御プログラムが教材として与えられる。また、操縦制御プログラムの仕様書として、IEEE830-1998⁸⁾の形式に従って記述された要求仕様書のほか、オブジェクト指向による分析設計結果のUMLモデル（ユースケース図、クラス図、シーケンス図、アクティビティ図）が与えられる。

フェーズ1でのテストはブラックボックステストにより行い、テストケース設計の段階では学生に操縦制御プログラムのソースコードを開示しない。したがって、単体テストではまず内部仕様書から個々のクラスにおけるメソッドが正しく動作するか否かを確認するテストケースを設計した後に、テストドライバを開発してテストを実施する。統合テストでは、ユースケース図に示されたユースケースに沿ってシステムの正常動作（機能特性）や操作感（非機能特性）を確認する。これらのプロセスの結果として、学生には、テスト仕様書（単体テスト、統合テスト）、テスト設計レビュー報告書、テスト実施報告書の提出を求める。

学生は、フェーズ1のプロセスを通じ、要求仕様書とUMLモデルとの間の追跡性や一貫性を保持することが、テストを実施するうえで、またプロダクトの品質を向上させるうえで

重要であることを実体験に基づき理解する。

フェーズ2では、フェーズ1のプログラムに対する機能拡張を行う。フェーズ2プロセスの詳細を図8に示す。フェーズ1で教材として与えられるプログラムは、スティックの操作量を飛行船制御命令に変換するだけの単純なものであり、飛行船からはフィードバックとして床からの高度や水平方向の回転角速度が得られるが、それらの情報は操縦制御にいったい利用していない。このプログラムで実際に飛行船を思いのままに操縦するのはかなり難しいので、フェーズ2の開発課題として、フィードバック情報を用いて飛行船の高度を一定に維持する機能を加えることを課題とする。すなわち、フェーズ2で学生はフェーズ1の要求仕様書とUMLモデルに対して高度維持機能を付け加えるための変更を行う。その後プログラムを実装し、単体テスト、統合テストを行って、システムの正常動作を確認する。これらの結果として、学生には、変更後の設計仕様としてのUMLモデル、設計レビュー報告書、テスト仕様書、テスト設計レビュー報告書、テスト実施報告書、実装したソフトウェアのソースコードの提出を求める。このように、フェーズ2では、V字モデルに従う開発をひととおり経験し、各工程で行うべき作業内容を理解することができる。

フェーズ3では、フェーズ2までのシステム使用経験に基づき、各チームが独自の改善点について提案する。フェーズ3プロセスの詳細を図9に示す。チームでは、プロジェク

*1 開発対象の操縦制御プログラムは小規模なシステムであるので、本PBLでは統合テストとシステムテストを区別せずに行った。

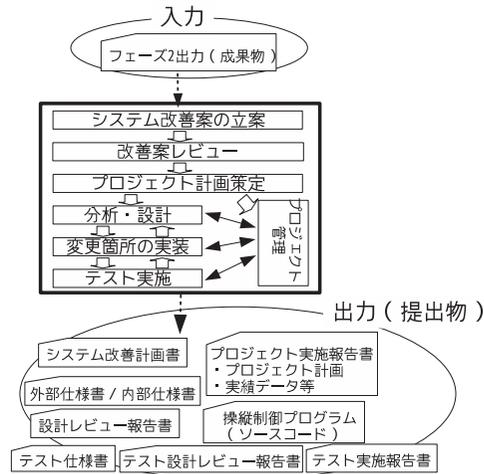


図9 フェーズ3のプロセス
Fig.9 Phase 3 process.

トの進捗監視役である教員のアドバイスを得ながら、提案された改善点について作業量の見積りやリスクの分析を行い、納期までの開発スケジュールやチーム内での作業分担を立案する。これら開発作業は、プロジェクト管理ツールを用いて記録するようにし、当初計画に対する進捗が担当の教員へ定期的に報告される。また、プロジェクト管理の一環として第11週目に担当教員および他チームに対して進捗状況を中間報告する機会を設けている。これらの結果として、学生には、フェーズ2で提出したプロダクトに加え、システムの改善計画書とプロジェクト実施報告書の提出を求める。このように、フェーズ3では、自ら設定した目標の実現に向けた問題解決のプロセスを実行することと、管理されたプロジェクトにおける個々の役割やコミュニケーションについて理解することが大きな目的となっている。

3つのフェーズからなる実習プロセスにおいて、フェーズ1で検証プロセスから開始する構成により、ソフトウェア品質を保持するための検証技術の重要性に関する理解を深めることを狙っている。また、フェーズ2と3とでV字プロセスを2回繰り返す構成は、筆者らが過去に関わった実習型の教育から得られた知見^(9),10)に基づいて考案した。問題解決型の実習を行うためには問題領域に関する背景知識が重要であることから、1回目の開発では正解を用意してそれに導くことで問題領域の理解を促し、2回目に同様の課題に対して問題解

表2 CASE ツール一覧
Table 2 CASE tools utilized in our PBL.

ツール名	用途
Enterprise Architect	UML モデルの閲覧と編集
Subversion	版管理・構成管理
TimeTrackerFX	工数管理・プロジェクト管理
Trac	進捗管理・バグ管理

決型の開発課題を与えるという戦略を採用した結果である。

フェーズ3終了後には、チームごとの開発成果のプレゼンテーションと飛行船操縦競技を含む報告会を配置している。プレゼンテーションを行い、質疑応答を通じた評価を受けることで、学生は自身の行った作業内容やその意義を振り返ることができる。また、競技の導入は、明確な目標を意識させることで開発・学習に対する意欲の維持を狙った結果である。

3.3 CASE ツールの利用

本PBLでは、CASEツールを開発に利用し、その効果や適用にあたっての制約について理解することも目的の1つとなっている。表2に利用した主なツールを示す。

UMLモデルの閲覧と編集には SparxSystem 社の Enterprise Architect^{*1}を用いた。OCEAN では受講生全員にこのライセンスを配布し、ソフトウェア設計技術などの座学講義でもこのツールを前提として演習を行っている。

Subversion^{*2}には、GUIツールとして Tortoise SVN^{*3}が存在し、Microsoft Windowsを用いた開発と親和性が高いことから採用した。各チームのためのリポジトリを専用のサーバ上に用意し、教員へのプロジェクト資料提出や進捗報告もこの上で行うことができるようにした。

デンソークリエイイト社の TimeTrackerFX^{*4}は、主にプロジェクト計画と作業工数の記録に用いた。

Trac^{*5}は、個々の作業項目に対する進捗の管理や、バグ対処結果の追跡に用いることができるよう用意した。

*1 <http://www.sparxsystems.jp/ea.htm>

*2 <http://subversion.tigris.org/>

*3 <http://tortoisesvn.tigris.org/>

*4 <http://www.denso-create.jp/service/products/timetrackerFX/>

*5 <http://trac.edgewall.org/>

4. 実習結果の評価

前章に示した設計に基づき、2007年度後期(2007年10月~2008年1月)に、名古屋大学10名、南山大学9名、愛知県立大学5名の合計24名の学生を対象としたPBL実習を行った^{*1}。開発チームは大学混成の6名ずつからなる4チームとし、それぞれのチームに教員がアドバイザーとして参加した。

本章では、期間中に行った学生へのアンケートの結果と、2008年4月に行った知識レベル調査の結果から、本PBLの効果について評価する。

4.1 アンケート結果

本PBLを受講する学生に対して、次のとおり期間中3回にわたりアンケートを実施した。

- (1) 第7週目(フェーズ2終了後)
- (2) 第11週目(フェーズ3中間進捗報告会后)
- (3) 第13週目(フェーズ3終了後)

アンケートは記名式で行ったが、誰がどのような意見を書いたかについては授業担当の教員には知らせない旨を学生に通知したうえで回答を集めた。いずれの回も質問内容は共通で、PBLにおいて学んだことや難しいと感じたことについて自由記述を求めている。表3に、各回において主だった意見を取り上げる。

これらの内容から、回を重ねるごとにプロジェクト管理、チーム開発について難しさを感じるとともに理解が進んでいることが見てとれる。また、2回目のアンケートからグループ内での情報共有についての要望が出ている点から、開発の進展にともなうチーム内でのコミュニケーションが活発になっていることも見てとれる。

各回を通じ、チーム開発の難しさ、コミュニケーションの大切さを経験できたという感想が出ていることから、本PBLの目的の1つである、チームによるプロジェクト開発の経験と難しさの理解は達成できているのではないかと考える。

一方、ハードウェア故障などに起因するトラブル、スケジュールが厳しい、教え方が不親切といった授業運営に対する不満は各回とも多く寄せられている。

4.2 知識レベル調査

OCEANでは教育カリキュラム全体を通じた教育効果を評価することを目的に、表4に示す方法で、自己評価式の知識レベル調査を行った。調査では、2008年に入学した修士1

表3 アンケート結果(抜粋)

Table 3 Students' answers to the questionnaires.

1回目	複数人で作業をやることの難しさを感じた。(チーム開発面, 他6件) 日程にきつい, 授業計画が理想的すぎる。(授業運営面, 他5件) 機器のトラブルに対応することが大変だと感じた。(機器トラブル面, 他3件) 実際にシステムの設計・実装・テストをひととおり行うのはとても大変だと実感した。(開発プロセス面, 他3件) 開発環境を整えることが難しかった。(開発環境面, 他1件)
2回目	全体的に時間が足りなかったです。(授業運営面, 他5件) グループで活動するうえで, プロジェクトなど設備の面でもっと改善の余地があると思う。(開発設備面, 他4件) プロジェクトを成功させるためにはプログラムが書けるだけではダメでチームワークなどが大事。(チーム開発面, 他2件) 便利なツールがあるが使いこなせていない。(開発ツール面, 他1件)
3回目	工数管理, ドキュメント管理, チームで行う作業の難しさ・大変さを味わえたと思う。(チーム開発面, 他6件) もう少しハードウェアに依存しない教材の方がソフトウェア開発用としてはふさわしいのではないかと 思う。(機器トラブル・開発課題面, 他4件) 各種ツールの使い方に関して, 説明が欲しかった。(授業運営面, 他3件)

表4 知識レベル調査方法

Table 4 Specifications of the knowledge level survey.

調査時期	2008年4月
調査対象1	2007年度入学生(修士2年生)20名
調査対象2	2008年度入学生(修士1年生・入学時)35名
設問の設計	SE知識項目から60個のキーワードを抜粋
回答方式	自己評価による5段階(0~4)回答

年生とコース1年終了後の修士2年生に対して、SE2004³⁾およびJ07-SE¹¹⁾の知識項目から抜粋した60個のキーワードに対して自分の知識レベルを評価させ、結果を比較することで教育効果をはかることを目的とした。知識レベルの評価は次に示すとおり5段階とした。

- レベル0: まったく知らない
- レベル1: 教科書や講義などで見聞きしたことがあり, 関連する資料を調べることができる
- レベル2: 言葉の意味を理解し, 説明することができる
- レベル3: 言葉に関する技術・技法をソフトウェア開発に利用することができる

*1 静岡大学はリモート拠点として本PBLとは別に設計したPBLを実施する。

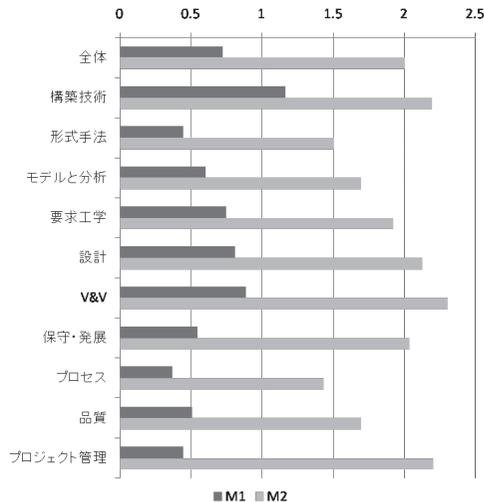


図 10 知識調査結果 (大分類)

Fig. 10 Result of the knowledge level survey.

- レベル 4: 技術の背景にある制約や特性について理解し, 特定の問題に対する向き・不向きを比較評価できる

レベル 1 から 4 は, それぞれ Bloom らによる知識レベル分類¹²⁾ の “knowledge”, “comprehension”, “application”, “analysis” に相当する。

図 10 に知識レベル評価結果を示す。横軸は回答結果のレベルの数字を単純に平均したもので, 濃い灰色は修士 1 年生, 薄い灰色は修士 2 年生の平均を示す。縦軸には 60 項目すべての平均と, 設問を 10 カテゴリに分類して平均を示した。全体で修士 2 年生の知識レベルが平均 2.00 (1 年生は 0.72) である中で, 構築技術, 設計, V&V (確認と検証), 保守・発展, プロジェクト管理の 5 項目が 2 以上の比較的高い値を示していることが分かる。

この結果は, 同一の学生らを対象とした調査を反映するものでないこと, 表 1 に示した座学での講義や, 修士 1 年後期に行われた OJL による学習成果を加味しなければならないことから, 本 PBL のみの学習成果を表すものではないので注意が必要であるが, 本 PBL の目的が, プロジェクト開発経験, CASE ツールの利用技術, 品質と検証技術に関する知識の修得であったことを考慮すると, 上記 5 項目の高い学習成果に PBL も少なからず貢献しているものと考えている。特に, プロジェクト管理の項目に対しては, 修士 1 年生と比べ

た 2 年生の知識レベルの差が大きい。これは, 先に示したアンケート結果からも読み取れるとおり, プロジェクト管理を実際に経験することで知識が深化した結果が現れたものと考えている。

また, 各受講生が作成した最終報告会でのプレゼンテーションや最終レポートからは, プロジェクト管理や検証技術に対する知識の深化を示す結果として, 次の内容を読み取ることができる。

- 悲観的なスケジュールリングの方針を採用することで納期までに開発を完了することができた。
- 報告書の作成というタスクの見落とし (軽視) に気付き, 改善することができた。
- 仕様書など, 文書を正確に記述することでチーム内での情報共有・意思疎通を円滑にできた。
- テスト設計からテスト設計レビュー, テストコード作成, テストまでをひととおり行うことで, テストにいかにか工数がかかるか実感できた。今後プロジェクト管理を行うときにこの経験を活かすことができると思う。

1 点目の内容からは, 作業見積りやスケジュールリングの手法を机上の知識としてではなく, 実際にソフトウェア開発に適用するとどのような効果があるかを理解し, チーム状況に合わせた開発戦略を選択できていたことが分かる。また, 2 点目の内容からは, フェーズ 2 で開発を行った結果, ソフトウェア開発プロジェクトで行うべき作業内容に対する理解が深まっていることが分かる。3 点目からはチーム開発における文書化の重要性を体感できていることが分かる。4 点目からは, テスト作業の実経験を通じて, その工数に関する感覚を得ていることが分かる。これらはいずれも座学講義のみでは理解しにくい経験に基づく知識である。このように, 実経験に基づいて本 PBL の内容を振り返り, 開発プロセスの改善提案をするなど, 最終報告書の内容からプロジェクト管理や検証に関する知識の深化をうかがい知ることのできる学生は全 24 名中 17 名であった。これらは本 PBL の学習効果の現れであると考えている。

5. 考 察

本章では, 本 PBL が当初想定した目標に照らして妥当に設計されているかどうかについて考察するとともに, ソフトウェア技術者教育カリキュラムの要素としての本 PBL について, 関連する取り組みと比較しながら議論する。

表 5 PBL 設計に対する課題
Table 5 Problems on our PBL design.

設計判断	課題と要改善事項
チーム開発	分担できる作業が少ない。チームの人数と作業分担や開発範囲の関係について検討・改善の必要あり。
CASE ツール	CASE ツールの習熟に時間がかかる。座学講義との連携、使用法を自習できるような解説資料を充実する必要あり。
NV モデル	要求技術に関する学習効果が小さい。座学講義や OJL との連携により補う必要あり。
飛行船制御	チームメンバが分担できる作業が少なく、ソフトウェア開発の工夫がもたらす効果が小さい。課題の与え方や機能拡張の仕様について再検討の必要あり。

5.1 PBL 設計方針の妥当性

本 PBL の目的と設計判断との関係については、3 章の図 2 に示したとおりであるが、本節では以下、次の点から設計判断の妥当性について議論する。

- チーム開発を想定した設計について
- CASE ツールの活用方針について
- NV モデルの採用について
- 題材としての飛行船制御について

結論として、これらの設計判断はそれぞれの目的に対しておおむね妥当であったと評価しているが、今後の改善に向けた課題も存在する。表 5 に示すのは主な課題である。

5.1.1 チーム開発を想定した設計について

複数名の学生によるチーム開発を想定した PBL 設計については、前章のアンケート結果が示しているとおり、プロジェクト開発におけるチーム内コミュニケーションを通じたプロジェクト運営の難しさを体験させるのに効果があったと考えている。OCEAN カリキュラムは大学間連携で実施されていることから、異なる大学に所属する学生間の交流を促進する目的で、混成チームを編成して実習を行った。学生は、普段地理的に分散した大学に通学しながら、本 PBL や座学授業が開講される日のみに一カ所に集合する形態で授業を受けている^{*1}が、この状況がはからずもチームコミュニケーションの難しさを増大させ、結果として良い教育効果をもたらしたと考えている。

一方、1 チームに想定する学生の人数については再考の余地がある。本 PBL に関わった

*1 名古屋大学は名古屋市千種区、南山大学は愛知県瀬戸市、愛知県立大学は愛知県長久手町に位置しており、OCEAN の授業は名古屋市東区の南山大学サテライトキャンパスで実施している。各大学から OCEAN 教室への移動時間は 30 分から 1 時間程度。

教員からは、5～6 名からなるチーム構成の割にメンバが役割分担して実施できる作業が少なく、個人ごとの作業負担に大きな差異が生じていたという感想を受けている。詳細についてはプロジェクトの記録を分析し、定量的な評価に基づいて議論しなければならないが、想定するメンバ数に見合った課題の与え方については、予測される作業とその中で並行実施可能な作業を実習資料の中に例示することで改善できると考える。

教員のプロジェクト参加については、2007 年度の実施においては、特にフェーズ 3 での開発プロジェクトを成功に導くのに効果があったと評価している。ただ、教員参加の効果については、個々のチームメンバや教員自身の資質に依存するところが大きいと考えており、この点については、今後本 PBL の教授法を文書化することで対処する予定である。

5.1.2 CASE ツールの活用方針について

本 PBL では、開発の各作業において各種 CASE ツール(表 2)を活用することを求めた。これに対して、学生アンケート(表 3)では CASE ツールの使い方に習熟するのが難しいという意見が多く寄せられた。PBL を実施した結果としてこれらの CASE ツールに対する知識は向上し、当初の目的は達成できたものと考えているが、CASE ツールの習熟に要する時間がプロジェクトにおける開発作業を圧迫していたことは否めず、それがアンケート結果での、時間が足りないという意見にもつながっている。今後は、実習資料中のツール使用法に関する解説を充実させる、座学講義において当該の CASE ツールを先行的に利用する機会を設ける、ベンダの協力のもとで利用講習会を開催するなどの対応により改善を行う必要がある。

5.1.3 NV モデルの採用について

本 PBL では、システムを一から開発することはせず、あらかじめ教員側で教材として用意したプログラムの検証から始め、機能拡張や改変を繰り返し行う NV モデルに従った構成をとっていることも特徴である。検証技術を重視するという方針、限られた時間に効果的な開発を行う必要性からこのような構成を選択し、前章の知識レベル調査(図 10)でもその効果が推測できる結果が得られている。この設計判断の背景には、実システムレベルの開発では、一から要求を分析する能力より、再利用資産を理解して利用する能力が求められるようになってきた現状認識が存在する。ただし、この設計判断を反映するように、同調査では、設計や構築の技術に関しては比較的高い値であるのに比べ、要求分析(モデルと分析)については全体平均より低い値となっている。再利用型の開発においても要求工学は重要な技術であるので、PBL だけでなく講義や OJL を含むカリキュラム全体の中で効果的に教授する方法について今後検討する必要がある。

一方、組み込みシステムの開発においては、対象システムに関するドメイン知識を早期に修得することが開発効率化の鍵である。本 PBL では、フェーズ 1 で対象システムを実際に動作させて検証しながら飛行船システムのドメイン知識を深めることができる。飛行船に限らず、ドメインに依存する知識が重要となるシステムを題材とする開発実習型の教育においては、本 PBL の NV モデルを採用することで一定の効果が得られると考えている。ただし、前述したとおり、フェーズ 1 で検証作業を行うコース設計は、検証技術を重視する方針のもと、限られた時間に効率良く実習を進める目的で設計した結果である。要求工学やドメイン工学の知識を深めることがより重要な場合や対象システムの物理的イメージが把握しにくいアプリケーションにおいては、ドメイン分析や業務分析の作業にフェーズ 1 の時間を割り当てるといった戦略をとることで学習効果を高めることができると考える。

5.1.4 題材としての飛行船制御について

本 PBL で飛行船制御問題を題材として選択した狙いは、主に学習意欲を維持させるところにあった。2007 年度の実施を通じ、学生は楽しみながら開発に参画していた。また、フェーズ 2 および 3 で飛行船に搭載されたセンサからのフィードバック情報を姿勢制御に利用する課題では、制御工学の前提知識をほとんど持たない状況から、PID 制御などの制御理論について自主的に学習しプログラムに反映する積極的な態度をすべてのチームが見せており、高い学習意欲を維持できたと評価している。

一方で、プロジェクト開発を経験させること、検証技術の重要性を理解させること、CASE ツールを活用させることという本 PBL の目標にとって、この題材が妥当であるかどうかについては議論が必要である。

まず、プロジェクト開発経験に対する飛行船システムの妥当性については、5.1.1 項すでに述べたとおり、5~6 名のチームに与える課題内容としては、分担実施できる作業が少ないという問題がある。これは飛行船システムそのものの問題というよりは、開発範囲と課題の設定に関する問題であり、たとえば、図 4 において ULMPU 上の操縦システムのみを開発対象とするのではなく、Rf.MPU 上の無線通信システムや Air.MPU 上の飛行船駆動システムのソフトウェアを開発の対象に加え、それぞれにメンバを割り当てて並行開発を行うなどの方法を検討する必要がある。

また、飛行船システムの特徴として、ハードウェア特性や物理的要因に左右される部分が大きく、ソフトウェアの開発実習にふさわしくないのではないかという意見が学生アンケートへの回答として数件寄せられている。すでに述べたように、組み込みソフトウェア開発では外界とのやりとりや外界からの影響を十分に考慮することが重要であり、ドメインをいか

に効率良く分析できるかがプロジェクトの成否の鍵を握る場合が多い。特定ドメインでの技術のみに長けた技術者を育てることは我々の目的ではないが、ドメイン知識の重要性を理解する能力は実践的な技術者に必要であり、それを教育する目的に本 PBL の飛行船制御ソフトウェア開発は適していると考えられる。ただし、フェーズ 1 で与えるプログラムの仕様やフェーズ 2 での機能追加の仕様については、よりソフトウェアの品質やソフトウェア設計上の工夫が反映されるよう検討を行う必要がある。

5.2 ソフトウェア技術者教育としての PBL

本節では、我々が設計した PBL での教育内容が、ソフトウェア工学教育に用いられる知識体系をどの程度網羅しているのかを考察するとともに、関連する既存の実習型教育との比較を行う。

5.2.1 ソフトウェア工学知識体系との関係

本項では、これまでに述べた本 PBL の設計方針と、2007 年度の実施結果を振り返り、ソフトウェア工学教育カリキュラムにおける本 PBL の位置付けについて整理する。表 6 は、SE2004³⁾ の知識領域と、本 PBL の前提となる知識および本 PBL で新たに得られる知識についてまとめたものである。この内容から、本 PBL の前提知識として、コンピュータ科学とソフトウェア工学に関する基礎的な知識に加え、要求工学や設計技術、検証技術に関しては少し踏み込んだ知識を必要とすることが分かる。本 PBL は、2 章で紹介した OCEAN カリキュラムの一環として設計したが、これらの必要知識をあらかじめ教育できる大学学部などのカリキュラムに組み込むことのできる汎用性を持っているといえる。また、本 PBL の受講後には、それぞれの前提知識を実開発を通じて深化させ、チーム開発（実施・管理）に必要な知識や品質に対する意識を高めることができるので、企業などでソフトウェアやシステム開発に携わることのできる専門家の養成を目的とする場合に有効な実習であるといえる。

5.2.2 既存の実習型教育との比較

技術者教育を目的としたプロジェクト型実習教育においては、実システム開発を題材として用いること¹³⁾ や、産学連携による取り組みが効果的であることが報告されている¹⁴⁾。これらは、必要とする知識や実践力の大部分を PBL のみで教授することを目的する場合に有効な方法であると評価できる。一方、OCEAN のカリキュラムでは、座学での講義、PBL、OJL を総合して技術者に必要とされる能力を養成することを目的としており、PBL が担う教育上の役割がこれらの先行事例とは異なる。

本 PBL の課題は実システムと比べると小規模であり、産学連携などの取り組みも行って

表 6 本 PBL の前提知識と獲得知識
Table 6 Prerequisite and acquired knowledge in our PBL course.

領域 *	前提知識	獲得知識
CMP	プログラミング基礎 (CMP.cf.1), アルゴリズムとデータ構造 (CMP.cf.2), 抽象化 (CMP.cf.4), コンピュータの構成 (CMP.cf.5), プログラミング言語基礎 (CMP.cf.9)	問題解決技法 (CMP.cf.3), API 設計・利用 (CMP.ct.1), コード再利用 (CMP.ct.2), エラー・例外処理 (CMP.ct.6), 組み込みシステム構築 (CMP.ct.13), 開発環境 (CMP.tl.1)
FND	—	ソフトウェア開発における価値の考慮 (FND.ec.1, 2, 4)
PRF	—	チーム開発作業と個人の役割 (PRF.psy.1, 2), コミュニケーションスキル (PRF.com.1~4)
MAA	モデリングの原理 (MAA.md.1), モデル表現と意味 (MAA.md.5), データ・振舞い・構造のモデリング (MAA.tm.1~3), 完全性・一貫性・頑健性 (MAA.af.1, MAA.rfd.4, 5), 追跡性 (MAA.af.5), 要求分析基礎 (MAA.rfd.1, 2), 要求獲得 (MAA.er.1, 2), 要求仕様化 (MAA.rsd.1~3)	ドメインモデリング (MAA.tm.4), 組み込みシステムのモデリング (MAA.tm.7), 要求のトレードオフ・リスク (MAA.af.4, MAA.rfd.6), 要求レビュー (MAA.rv.1), 受入れテスト・品質評価 (MAA.rv.3, 4)
DES	設計の基礎 (DES.con.1, 2, 4, 5), 機能指向設計 (DES.str.1), オブジェクト指向設計 (DES.str.2), 詳細設計技法 (DES.dd.1)	設計と要求との競合・トレードオフ (DES.con.5, 6), 設計支援ツール (DES.ste.1)
VAV	V&V の基礎 (VAV.fnd.1, 2, 5), テスト技法 (VAV.tst.1, 3~6, 8)	レビュー (VAV.rev.1~3), 例外処理のテスト (VAV.tst.2), 品質特性のテスト (VAV.tst.7), 不具合分析 (VAV.par.1~4)
EVO	保守の基礎 (EVO.pro.1, 2)	保守の計画 (EVO.pro.5), プログラム理解と波及解析 (EVO.ac.2, 4)
PRO	プロセス基礎 (PRO.con.1, 2, PRO.imp.1, 2)	プロセス測定・分析 (PRO.con.3), プロセス改善 (PRO.con.4), 品質分析・制御 (PRO.con.5), チームプロセス (PRO.imp.5)
QUA	ソフトウェア品質の基礎 (QUA.cc.1~4)	品質の作り込み (QUA.cc.5), 品質と人間・プロセス・ツール・方法論・技術 (QUA.cc.6)
MGT	マネジメント基礎 (MGT.con.1~5)	プロジェクト計画 (MGT.pp.1~6), 変更制御 (MGT.ctl.1), モニタリング (MGT.ctl.2), 計画修正 (MGT.ctl.4), 構成管理 (MGT.em.1~4)
SAS	—	組み込みハードウェア (SAS.emb.1), 組み込み向け開発環境 (SAS.emb.2)

* CMP: コンピュータ基礎, FND: 数学・工学基礎, PRF: 専門職業倫理, MAA: モデル化と分析, DES: 設計, VAV: V&V, EVO: 保守・発展, PRO: プロセス, QUA: 品質, MGT: プロジェクト管理, SAS: 専門領域知識

いない。より実システム開発に近い題材を選定し、産業界からの意見を採り入れることが望ましいことは確かであるが、本格的な実習 (OJL) の予備段階であるという位置付け、半期という短期間の実施であること、現課題に対しても「負担が大きい」、「スケジュールが厳しい」という意見が学生から多く寄せられていることなどを考慮すると、現在の課題と同等、もしくは少し小規模化するのが適切なレベルであると考えている。

6. おわりに

本論文では、筆者らが参画する IT 技術者教育プロジェクト OCEAN において設計したソフトウェア開発実習の教育プログラムと教材について紹介した。

本 PBL は、軟式飛行船制御の組み込みソフトウェア開発を題材にし、NV モデルと呼ぶ、3つのフェーズからなる繰返しプロセスにしたがってチーム開発を行う。教育の対象は大学院修士課程 1 年生であり、実習の主目標は、管理されたプロジェクト開発を経験することと

ソフトウェア品質向上における検証技術の重要性を理解することである。

この目的を持って設計した PBL プログラムによる実習を、2007 年度後期 (2007 年 10 月から 2008 年 1 月の間) に行った。学生へのアンケートからは、チームによるプロジェクト開発について理解が深化していることが推測された。また、カリキュラム全体に関する知識レベルの調査では、構築技術、設計、確認と検証、保守・発展、プロジェクト管理などの分野で高い学習成果を示していることが分かった。

今後の課題として、実施を通じて得られた経験や評価結果を参考に PBL の設計を改善することが第 1 にあげられる。また、学習効果を定量的に測定する基準や測定方法の検討についても今後の重要な課題である。我々は、今後これらの課題を解決しながら、他大学や大学院などで本 PBL と同様の実習を実施できるよう、利用機材やソフトウェア、講義資料、教員用の指導要領などを整備し、その成果を公開する予定である。

謝辞 本 PBL の開発は、文部科学省研究拠点整備等補助金 (先導的 IT スペシャリスト

育成推進プログラム)による助成のもとで行われた。また、本PBLに用いられている技術の一部は、科学研究費補助金(基盤研究(C)19500028, 21500042)および南山大学パッへ奨励金I-A-2(平成20年度), I-A-1(平成21年度)の助成による研究の成果である。

飛行船システムをPBL題材として選定し、教材の開発を行うにあたり、(株)東陽テクニカの二上貴夫氏に技術的協力を得た。記して謝意を表す。

参 考 文 献

- 1) 沢田篤史, 小林隆志, 金子伸幸, 中道 上, 大久保弘崇, 山本晋一郎: 飛行船制御を題材としたプロジェクト型ソフトウェア開発実習, 組込みシステムシンポジウム2008論文集, 情報処理学会シンポジウムシリーズ, Vol.2008, No.9, pp.5-14 (2008).
- 2) 文部科学省: 先導的ITスペシャリスト育成推進プログラム8拠点のプロジェクト概要 — 世界最高水準の高度IT人材育成を目指して(2008).
- 3) The Joint Task Force on Computing Curricula: Software Engineering 2004 – Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, A Volume of the Computing Curricula Series, IEEE Computer Society and Association for Computing Machinery (2004).
- 4) 酒居敬一, 荻原剛志: 情報系学部教育におけるアセンブリ言語を用いた組込みシステム開発実験の試み, 組込みシステムシンポジウム2007論文集, 情報処理学会シンポジウムシリーズ, Vol.2007, No.8, pp.6-14 (2007).
- 5) 小倉信彦, 渡辺晴美: ロボットコンテストを利用した組込み教育の実践, 組込みシステムシンポジウム2007論文集, 情報処理学会シンポジウムシリーズ, Vol.2007, No.8, pp.15-22 (2007).
- 6) MDD ロボットチャレンジ編集委員会(編): MDD ロボットチャレンジ2004: 産学連携による組込みソフトウェア開発の実践, 情報処理学会 (2005).
- 7) MDD ロボットチャレンジ編集委員会(編): MDD ロボットチャレンジ2005: 産学連携によるモデルベース組込み開発の実践, 情報処理学会 (2006).
- 8) IEEE 830-1998 IEEE Recommended Practice for Software Requirement Specifications, IEEE Computer Society (1998).
- 9) 沢田篤史: 大学におけるモデリングの教育, ソフトウェアシンポジウム2007予稿集, pp.95-96, ソフトウェア技術者協会 (2007).
- 10) 沢田篤史: 組込みシステムを題材としたモデリング教育, ウィンターワークショップ2008・イン・道後論文集, 情報処理学会シンポジウムシリーズ, Vol.2008, No.3, pp.3-4 (2008).
- 11) 阿草清滋, 西 康晴, 沢田篤史, 鷲崎弘宣: 特集 情報専門科目カリキュラム標準 J07: ソフトウェアエンジニアリング領域(J07-SE), 情報処理, Vol.49, No.7, pp.743-749 (2008).
- 12) Bloom, B., Englehart, M., Furst, E., Hill, W. and Krathwohl, D.: *Taxonomy of*

Educational Objectives: The Classification of Educational Goals, Handbook I: Cognitive Domain, Longmans (1956).

- 13) 井上 明, 金田重郎: 実システム開発を通じた社会連携型PBLの提案と評価, 情報処理学会論文誌, Vol.49, No.2, pp.930-943 (2008).

- 14) 松澤芳昭, 大岩 元: 産学協同のProject-based Learningによるソフトウェア技術者教育の試みと成果, 情報処理学会論文誌, Vol.48, No.8, pp.2767-2780 (2007).

(平成21年2月2日受付)

(平成21年9月11日採録)



沢田 篤史(正会員)

1995年京都大学大学院工学研究科情報工学専攻博士後期課程研究指導認定退学, 同年奈良先端科学技術大学院大学情報科学研究科助手, 1997年京都大学大学院工学研究科助手, 同年同大学大型計算機センター助教授, 2002年同大学学術情報メディアセンター助教授, 2007年南山大学数理情報学部教授を経て, 2009年より同大学情報理工学部教授, 現在に至る。博士(工学)。ソフトウェア工学, 組み込みシステム工学等の研究に従事。電子情報通信学会, 日本ソフトウェア科学会, システム制御情報学会各会員。



小林 隆志(正会員)

1997年東京工業大学工学部情報工学科卒業。1999年同大学大学院情報理工学研究科計算工学専攻修士課程修了。2004年同専攻博士課程修了。2002年同大学学術国際情報センター助手。2007年名古屋大学大学院情報科学研究科附属組込みシステム研究センター特任准教授, 2009年より同研究科情報システム学専攻准教授, 現在に至る。工学博士。ソフトウェア設計方法論, ソフトウェア再利用技術, 複合メディアコンテンツの管理・検索, Webサービス連携等の研究に従事。電子情報通信学会, 日本ソフトウェア科学会, 日本データベース学会, ACM各会員。



金子 伸幸

2002年名古屋大学工学部電気電子情報工学科卒業。2004年同大学大学院工学研究科情報工学専攻博士前期課程修了，同後期課程へ進学。2007年同大学院情報科学研究科附属組込みシステム研究センター研究員。2008年より株式会社ネットレックスに勤務。ソフトウェア再利用，要求工学，ソフトウェア開発環境等の分野に興味を持つ。日本ソフトウェア科学会

会員。



中道 上 (正会員)

1999年奈良工業高等専門学校専攻科電子情報工学専攻修了。2004年奈良先端科学技術大学院大学博士前期課程修了，2007年同大学博士後期課程修了，1999年住友金属システム開発株式会社（現，アイエス情報システム株式会社）入社。2004年日本学術振興会特別研究員（DC1），2007年南山大学数理情報学部講師，2009年同大学情報理工学部講師。博士（工学）。ソフトウェアユーザビリティ，ソフトウェア開発プロセス改善の研究に従事。電子情報通信学会，IEEE 各会員。



大久保弘崇 (正会員)

1992年名古屋大学工学部情報工学科卒業。1997年同大学大学院博士課程修了。1997年愛知県立大学助手，2009年より愛知県立大学講師。工学修士。オブジェクト指向計算モデル，代数的仕様記述法，ソフトウェア工学等の研究に従事。日本ソフトウェア科学会会員。



山本晋一郎 (正会員)

1987年名古屋大学工学部卒業後，同大学大学院に進学，1991年同大学助手，1996年講師。1998年愛知県立大学情報科学部助教授，2007年同大学准教授。プログラミング言語処理系，ソフトウェアの形式的開発手法，ソフトウェア開発環境に関する研究に従事。近年は，細粒度のソフトウェア・リポジトリに基づいたCASEツール・プラットフォームに関する研究を進めている。電子情報通信学会，日本ソフトウェア科学会各会員。