

# UMLによる実践的システム設計・開発

## ～ 卒研究生が試みた教材開発 ～

職業能力開発総合大学校 情報システム工学科 深江 裕忠

### 1. はじめに

この数年、IT技術の急速な進歩と過激な開発競争によって、ソフトウェア規模が肥大化しているにもかかわらず、開発期間が極端に短くなっています。また、仕様も市場動向に従って変更されることが多くなり、開発途中で中断し、新しい仕様で作り直す事態も発生しています。

つまり、ソフトウェア開発現場では生産性の向上と仕様変更に対応する開発手法が求められています。この解決策として、UML (Unified Modeling Language)<sup>1)</sup>を用いたシステム設計・開発が注目され、UMLの学習書も多数出版されています。

ところが、これらの学習書はUMLの知識を習得するレベルにとどまり、使いどころやコツなどの知的管理技能レベルまで到達するには不十分な内容となっていることが多いです。UMLを学習した人の多くは、UMLで書かれた図を読めるようになったとしても、UMLを用いてシステム開発をするときには壁を感じます。

そこで、情報工学科の卒業研究<sup>2)～5)</sup>にて、UMLを用いたシステム開発手法を習得するための教材を開発したので、その取り組みについて報告します。本報告の前半では、卒研究生が教材開発を行うことの「ねらい」と「教材開発の取り組み方」について記述します。これは卒業研究指導を行いながら教材開発全般において有用だと感じた部分を抽出したものです。UMLだけではなく、他の教材開発においても応用できる部分です。後半は、開発した「UMLによる

実践的システム設計・開発」について記述します。

### 2. 卒業研究で教材開発をする「ねらい」

最初に、実務実習以外では授業経験のない卒研究生に教材開発を行わせる「ねらい」について説明しておきます。

職業能力開発総合大学校（以下「総合大」という）では、ご存知のとおり、将来の職業訓練指導員（以下「指導員」という）を養成しています。私自身も総合大出身の指導員であり、これまでポリテクセンターと職業能力開発短期大学校にて職業訓練や教材開発を経験してきました。この経験を生かして、指導員の卵として総合大で学んできた卒研究生がさらにレベルアップしてもらうことをねらったのが、教材開発です。この試みには、おもに3つのねらいがあります。

#### 2.1 テキスト作成を通じて技術を深く理解する

教材開発といっても、実習用機器の開発、eラーニングの作成、説明用スライドの作成など、いろいろあります。しかし、卒業研究で作成したのは、授業で配布して講義や実習の資料として使うテキスト教材です。私自身の教材開発の経験から、テキスト教材を開発するということは、

- 学習者が理解しやすく書く。
- 講師が時間の都合上省く説明を含める。

という2点に気をつけて、学習者が疑問に思うような所を網羅し、しかもわかりやすく書く必要があります。このことは、技術を知っているだけではできない作業です。そのため卒研究生は、テキスト教材

の作成過程において自ら疑問点を調べて実際に動作を確認し、自分の言葉で解説を書きます。このことによって、完成したときには幅広く深く理解したという大きな自信がうまれます。

## 2.2 新技術に対応できる指導員を育てる

教材開発のターゲットは、将来的展望が望めるが良質な教材がない技術分野にしています。そして、テキスト教材は文献や資料をまとめるのではなく、必ず実際に試して確認したものを盛り込むようにしています。例えば、演習で作成するプログラムなどは卒研究生が実際に作っています。これによって、新技術を実際に経験してそのメリットを肌で感じるることができます。

さらに、1つ目の「ねらい」とあわせることで、まだ情報の少ない新技術であっても自ら調べて教材を開発できる指導員に育つことをねらっています。

## 2.3 カリキュラム開発能力を鍛える

3つ目の「ねらい」は、カリキュラム開発能力を実践的に鍛えようというものです。総合大生は指導技法を学び、実務実習にて授業実施を体験しますが、授業のカリキュラム自体は与えられたものです。しかし、指導員はカリキュラムをこなすだけでなく、地域や企業のニーズに応じてカリキュラムを開発する能力が必要です。総合大生は、講義ではカリキュラム開発手法について学んできましたが、本格的なカリキュラム開発を経験する機会はありません。そこで、教材開発を通じてカリキュラム開発能力を鍛える第一歩となることをねらっています。

具体的には、教材を作成する前にミーティングを行い、受講生のレベル、訓練時間、実習場の環境、学習目標といった要素を考慮しながら教材に盛り込む順序と分量を決定する作業を行っています。

ここで、授業経験の少ない卒研究生が考えたカリキュラムでは実用的な教材にはならないと感じるでしょうが、事実そのとおりです。しかし、「ねらい」はカリキュラム開発能力を鍛えることです。すなわち、教育工学やeラーニングなどで用いられている教材設計手法「インストラクショナルデザイン (Instructional Design)」<sup>6)</sup>を学習し、その手法に従って作成できるようになることが目的です。また、完成した後に

必ず評価を行い、PDCAサイクルを何度もまわさないとカリキュラムは完成しないということも経験させています。

もし、実用的な教材が欲しいのなら、卒業後や、研究を引き継いだ人がPDCAサイクルを何度もまわして完成させるという方針です。

## 3. 教材開発の取り組み方

### 3.1 インストラクショナルデザイン

インストラクショナルデザインとは、良質な教材を作成するための開発手法です。PDCAサイクルをまわしながら、体系的なアプローチで教材を開発します。授業と教材開発の経験が少ない卒研究生でも、良質な教材を作成できるようにするために採用しました。

インストラクショナルデザインの大きな教材開発手順を、図1に示します。

最初に教材を使用することで学習者が達成する学習目標を設定します。さらに、学習目標が「○○を理解する」「○○を習得する」といった曖昧な言葉で設定していると到達度を測定することができないので、基準と条件を加えて明確化します。学習目標が完成すると、次の3種類のテストを作成します。

- 学習者が前提条件を満たしているか調べる「前提テスト」。
- 学習者が教材の内容について未修得であることを調べる「事前テスト」。
- 学習を終えたときに学習目標を到達したか調べる「事後テスト」。

そして、学習内容の分析を行って、適切な順序とボリュームで学習できるようにします。また、教材メディアや指導方法なども決定します。その後、教材を作成し、教材の品質について評価を行い、改善を繰り返していきます。

卒研究生を指導するに当たって、力を入れているのは、学習目標の明確化と課題分析です。この2点が不足していると良質な教材にはなりません。また、経験の少ない卒研究生でも取り組みやすい内容です。それに対して、学習者のレベルを分析するのは卒研究生にとっては難しいです。そのため、学習者は卒研

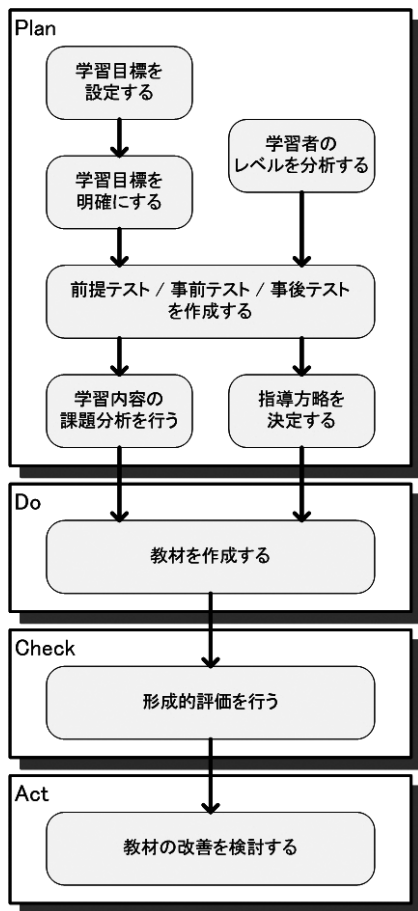


図1 インストラクショナルデザインの大きな教材開発手順

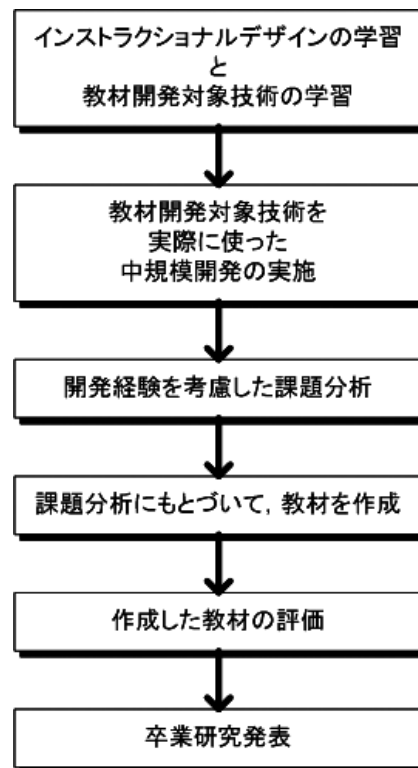


図2 教材開発スケジュール

生にとってなじみのある「総合大情報システム工学科〇年生」といった形にしています。

指導方略と教材の作成については、インストラクショナルデザインよりも、私の指導員経験をもとに解説や実例を示したほうが卒研生は理解しやすいようです。この部分については、より体系化した形にしてみたいところです。

### 3.2 スケジュール

卒業研究のスケジュールは、図2のようになっています。

このスケジュールの特徴は、最初にインストラクショナルデザインの学習を行うことです。卒研生が教材開発の設計手順を学習することで、以後の作業について、教材開発を意識するようになります。

もし、教材開発対象技術を実際に使った中規模開発を先にはじめると、卒研生は新技術を使うことだけに熱中してしまう傾向があります。そのため、教

材開発に必要な「初心者がつまづくポイント」「理解するまでの過程」といったことを意識せずに進めるため、教材を作成するのが困難になります。

インストラクショナルデザインの入門書<sup>7)</sup>を使って学習しましたが、その効果は時間が経つにつれてじわりじわりと出てきました。

もう1つの特徴は、教材開発に入る前に、教材開発対象技術を使った中規模開発をすることです。場合によっては、この開発だけで卒業研究期間が終了してしまう危険性があります。しかし、卒研生自身の経験を生かした課題分析と教材作成を行うためには必須の作業といえます。そうでなければ、資料をまとめただけの工夫のない教材が出来上がってしまうからです。

理想としてはPDCAサイクルを2回程度まわしたいのですが、現実には教材を作成したところで卒業研究期間が終了してしまいます。この点に関しては、卒業研究指導上の改善が必要だと考えています。

## 4. UMLによる実践的システム設計・開発教材

今までの記述は、平成17年度から2年間にわたっ

てUMLの教材開発を通じて得たものです。そして、「UMLによる実践的システム設計・開発教材」というテキスト教材と実習課題システムのプログラムが完成しました。

ここからは、「UMLによる実践的システム設計・開発教材」について紹介します。

#### 4.1 UMLとは

UMLは、OMG (Object Management Group) が定めたソフトウェアの設計・開発用モデリング言語です。開発するシステムをさまざまな側面から分析して、表1にあるような図を作成します。図の描き方と意味がOMGによって厳密に定められているため、ソフトウェアの仕様書として利用できます。また、従来の文書で書かれた仕様書と比べて、顧客や開発者同士での解釈の齟齬が少なく、全体的な見通しが良いためシステムを把握しやすいという利点があります。

また、ソフトウェア開発では必須となっているオブジェクト指向の考え方が反映されているので、オブジェクト指向に従った設計を図で書くことができます。

ところで、図は手書きで作成することもできますが、図同士が密接に連携しているため、1箇所の変更が他の図にも影響を与えます。そのため、自動的に他の図も連動して変更してくれるUMLツールを用いて作成することが望ましいです。また、UMLツールは自動的に図からプログラムに変換する機能を持っているため、生産性が向上するメリットがあります。すなわち、顧客の要望変更があったときに、従来の開発手法では仕様書とプログラムの修正に手間がかかったのが、UMLを用いることで、図を変更するだけで自動的に関連するプログラムなどが修正されます。したがって、UMLを用いると仕様の変更を簡単に行うことができます。

ただし、UMLを用いるメリットを得るには、いくつかの注意が必要です。まず、中規模以上の設計・開発を対象としているので、小規模な設計・開発に適用しても煩雑になるだけです。また、13種類の図をすべて使うのではなく、開発対象と工程に合わせて必要なものだけを使います。例えば、業務システ

表1 UML 2.0で定められている図

名称	内容
ユースケース図	システムの機能と利用者
クラス図	システム中のクラス
シーケンス図	オブジェクト間の処理の流れ
パッケージ図	クラスの配置
コンポーネント図	コンポーネントの配置と依存関係
配置図	ハードウェアの配置
アクティビティ図	機能実行の流れ
オブジェクト図	オブジェクトの状態
コミュニケーション図	オブジェクト間のメッセージ交換
コンポジット構造図	システムの構成
相互作用概要図	全体的な処理の流れ
ステートマシン図	システムの状態遷移
タイミング図	状態遷移のタイミング

ムだとユースケース図、クラス図、シーケンス図が重要になります。組み込み系の開発だとユースケース図、クラス図、ステートマシン図、タイミング図が重要です。そして、顧客ヒアリングやチームを組んだ開発体制がないと顧客や開発者同士での解釈の齟齬がないというメリットを得ることは当然できません。

#### 4.2 UMLを学習するときの問題点

UMLに関する市販本は多数存在します。私も何冊か読んだことがあります。しかし、UMLの入門書はUMLの読み方とUMLツールの使い方だけを紹介したものが多くあります。逆にUMLについて詳細な解説が載っている書籍は、読者の前提条件を高いレベルに位置づけている本が多くあります。すなわち、UMLの初心者から使いこなすレベルに到達するための実践的なテキスト教材がないのです。厳密には、いくつか見かけましたが、それらはUML 1.4という古いバージョンで書かれています。

さらに、学生を対象にUMLを学習する際には、システム開発経験がないために、次の点で学習につまずいてしまいます。

- チームでの共同開発の方法がわからない。
- 複数の利用者を考慮した分析・設計方法ができない。
- 開発工程とUMLの図の対応がわからない。
- UMLの図から図、図からプログラムへの展開方

法がわからない。

それに加えて、オブジェクト指向の考え方が前提なので、学生にとっては敷居が高くなっています。

#### 4.3 開発した教材の内容

開発した教材の特徴をまとめると、表2のようになります。ところで、UMLを使いこなす職種を大別してみると、2つに分かれます。1つは、顧客の要望をヒアリングして業務分析を行いUMLの仕様書にまとめるシステム設計者。もう1つは、UMLの仕様書に従ってクラスの分析を行い、プログラミング開発をするシステム開発者です。この教材では、システム開発者としてのUMLの使い方を学習します。

学習者は、2～3人でチームを組んで、課題システムを実習で作成します。作成したテキストでは、次のような弁当屋システムの作成をするように指示しています

今回はお弁当屋のシステムを作成してください。店は店舗販売形式で営業します。

この店ではまずお客さんが店に来たら、店舗内にある端末で「注文したいメニュー」「個数」を入力します。すると、整理番号が発行され「注文明細」が印刷されます。これをレジまで持っていくと、注文完了となります。

さらに、端末から入力された客の注文は、店舗内にあるデータベースを介して調理場に送られ、調理場ではこれを見ながらお弁当を作ります。

また、店内のお弁当のメニューはすべてデータベースで管理されていて、経営管理者用の端

末からこれを実行することができます。例えば、「新メニューの追加」、「廃棄メニューの削除」、「メニューの部分的な変更」などが可能です。

この弁当屋システムは、卒研究生が実際に制作した中規模システムです。図3（左）のユースケース図から開始して、画面設計、クラス図、シーケンス図と設計していき、最後に図3（右）のようなシステムを完成しました。テキスト教材の実習では、卒研究生が経験したのと同じ手順で行います。ただし、各図の作成については実習課題となっているので、同じ図やプログラムが出来上がるとは限りません。しかし、システムの機能は同じになるはずで

この課題の作成を行いながら、図とプログラムの関係や設計方法を学習していくのが、目標の1つになっています。

もう1つの目標が、チームで開発できるようにすることです。例えば、オブジェクトの使用者に配慮したクラス設計方法を学習します。

ところで、チーム開発の場合、複数の人が同時にプログラムを作成するので、手渡しやファイル共有などでファイルを交換していると、どのバージョンが最新なのかわからなくなったり、ちょっとしたミスで他人のプログラムを壊したりすることになります。実際の開発現場では、このような事故をなくすためにバージョン管理システムを使ってプログラムの共有を行っています。

そこで、この教材でもSubversion<sup>8)</sup>というバージョン管理ツールを使うようにしています。Subversionとは、実際の開発現場で広く使われていたCVSとい

表2 開発した教材の特徴とねらい

特徴	ねらい
Java,オブジェクト指向,データベースとSQLを学習済みの人が対象	情報システム工学科3年生を想定
実習でシステム開発をしながらUMLの使い方を習得する	知識だけでなく,UMLの使い方の実践的学習を実施
課題として,弁当屋の受発注システムを開発する	学生がイメージやすく,他のシステム開発にも転用可能
詳しく学習するのは,ユースケース図,クラス図,シーケンス図のみ	知識よりも使いこなすことを重視
ユースケース図,クラス図,シーケンス図の順に作成する	開発工程と各図の対応関係や図から図へのつながりを習得
ユースケース図とクラス図の間に,画面設計の工程を入れる	学生がシステムをイメージできないことへの対策
チームで課題を作成し,チーム開発用ツールの使い方を習得する	現場で使われているチーム開発ツールを経験する
チーム開発を考慮したオブジェクト指向開発手法を習得する	オブジェクト指向を効果的に使う手法を学習する

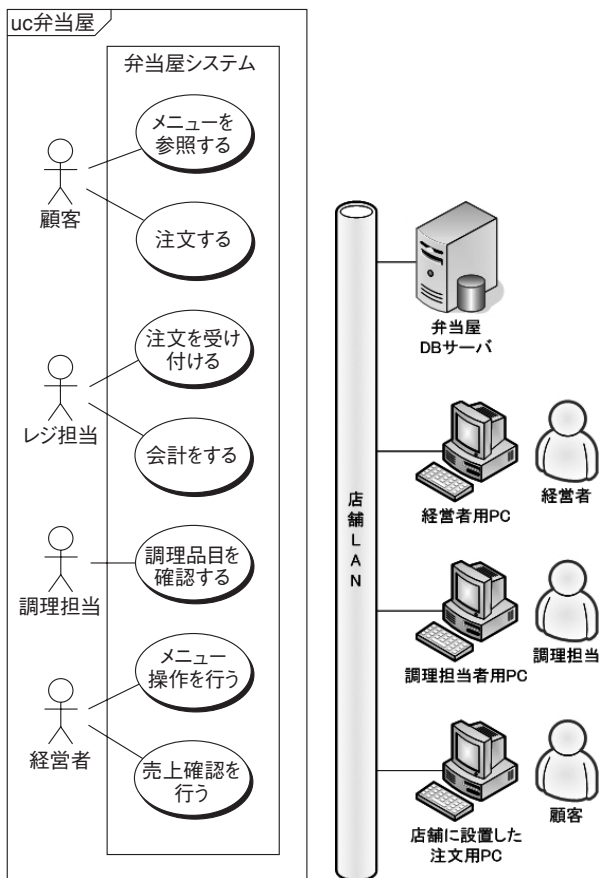


図3 弁当屋システムのユースケース図(左)と 弁当屋システムの概要(右)

うバージョン管理ツールの後継版です。現在、CVS からSubversionへの置き換えが進んでいます。

Subversionを使うと、図4のように、チームの1人が変更したファイルをSubversionサーバに登録することで、他の人にファイルを渡せます。それだけでなく、ファイルの変更箇所が衝突しているかどうかのチェックや、衝突したときの処理方法、間違えたときに過去に簡単に復旧できるなどの優れた機能が提供されます。これらの具体的な使い方や注意する点についても学習します。

## 5. 評価と課題

### 5.1 実習課題の評価と課題

作成した教材の評価を得るために、実習課題のシステムについては、平成18年11月に行われた第14回職業能力開発研究発表講演会でパネル展示を行いました。この会場にて、実習課題システムを稼働し、卒研究生が説明を行うことで見学者の反応を掴むこと

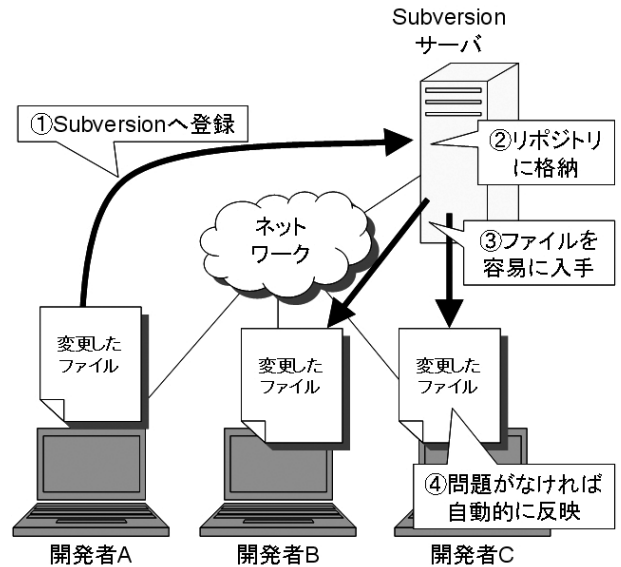


図4 Subversionの動作

にしました。

その結果、見学者の多くは弁当屋のシステムとして動いているのは確認できたが、もっと便利になる機能の追加を提案してきました。これは事前に予想している反応です。なぜなら、実習用課題という前提で作成したので、あえて便利な機能や華やかな装飾は外して、最も基本となる機能だけに絞ってシンプルに設計したからです。なお、提案された追加機能のいくつかは、実習後の追加課題として次の改訂で盛り込む予定です。

また、実習課題システムを卒研究生が1ヵ月程度で完成したことに驚いている人もいました。UMLを使うことで生産性が向上していることを実感する反応といえます。

こうして、実習課題システムの展示は、トラブルもなく無事に終了しました。また、致命的に抜け落ちている機能などの指摘もなかったです。

### 5.2 テキスト教材の評価と課題

職業能力開発研究発表講演会が終わった後は、卒研究生はテキスト教材の開発に着手しました。各章ごとに課題分析と担当を決めて作成し、平成19年1月末に完成しました。

この完成したテキストを、総合大の教員と各施設の指導員、計35名にアンケート用紙とともに渡しました。アンケートは、「テキストと授業方式の相性について」「授業の準備について」「授業時間について」

「テキストの内容について」の5分野に関するいくつかの設問を、5段階評価で回答します。

この原稿を執筆しているときは、ちょうどアンケートの返答を待っている段階ですが、現時点で数件の回答がきています。

それによると、授業の準備については、時間がかかること以外は特別なものを用意する必要がないという評価が多かったです。これは、入手しやすく現場でよく使われているフリーソフトを中心に開発環境を用意した点と、情報技術についてある程度のスキルがあれば授業を実施することが可能であることを示していると考えられます。

また、テキストの各項目のボリュームと順序についても、比較的良い評価を得ています。テキストを作成する前に課題分析を行って、適切なボリュームと順序を維持できたことが大きいと考えます。

しかし、テキストにはメリハリがなく、何を実行すればよいのかわかりにくいという評価も得ています。同じような評価で、全体的な構成がわかりにくいというの也有ります。原因としては、卒研2名がテキストを分担して作成したために、全体構成やメリハリがわかりにくくなってしまったと考えられます。課題分析によって適切に作業分担できたことで、逆に連携する意識が弱まってしまったので、今後の作成への注意点としたいと思います。

## 6. おわりに

このようにして、「UMLによる実践的システム設計・開発」のPDCAサイクルが1回まわりました。この間に経験したことは、今後の別の教材開発に生かしたいと思います。

また、「UMLによる実践的システム設計・開発」も2回目のPDCAサイクルに入り、テキストを改訂していきます。次のバージョンでは、アンケート評価で指摘された部分を改善するとともに、学生を対象とするのではなくセミナーでも使えるような教材に仕上げたいと考えています。

### <謝辞>

最後に、この教材開発を実行してくれた卒研生の

表3 完成したテキスト

用途	用途
配布用テキスト	UMLとツールの解説と使い方 (141ページ)
課題の参考解答	課題システムのUMLとプログラム (173ページ)

藤村君、田尻さん、近藤君、和田君、そして快く卒研を任せてくれた寺町先生、松嶋先生、忙しいなか私の専門外であるインストラクショナルデザインのゼミの実施に協力と助言をいただいた新井先生に感謝します。また、無理を承知でお願いして、テキストの評価に協力してもらった数多くの方、ありがとうございます。皆さんの協力によって、テキストを完成し、改善することができます。

### <参考文献>

- 1) Dan Pilone, Neil Pitman: 『UML 2.0 クイックリファレンス』, オライリージャパン, 2006.
- 2) 藤村優太: 『UMLによるシステム設計・開発演習の教材作成 ~ Enterprise Architectを用いたシステム設計 ~』, 平成17年度職業能力開発総合大学校情報工学科卒業論文, 2006.
- 3) 田尻真子: 『UMLによるシステム設計・開発演習の教材作成 ~ Eclipse Javaを用いたシステム開発 ~』, 平成17年度職業能力開発総合大学校情報工学科卒業論文, 2006.
- 4) 和田浩仁: 『UMLによる実践的システム設計・開発教材の作成 ~ UML設計編 ~』, 平成18年度職業能力開発総合大学校情報工学科卒業論文, 2007.
- 5) 近藤雄亮: 『UMLによる実践的システム設計・開発教材の作成 ~ チーム開発編 ~』, 平成18年度職業能力開発総合大学校情報工学科卒業論文, 2007.
- 6) Walter Dick, Lou Carey, James O. Carey: 『はじめてのインストラクショナルデザイン』, ピアソン・エデュケーション, 2004.
- 7) 鈴木克明: 『教材設計マニュアル』, 北大路書房, 2002.
- 8) Subversion開発プロジェクト: 『Subversion解説書』, IDGジャパン, 2004.