

CASE を用いた情報システム設計・開発のための授業設計

河村 一樹

宮城大学 事業構想学部

kawamurk@mail.sp.my.ac.jp

デザイン情報学科の情報システムコースでは、情報システム学教育を取り入れ、システムエンジニアの育成を目指している。その中で、筆者は、情報システムの構築のために必要となる方法論や技法を教授する「情報システム設計 I・II」を担当している。これらの科目の演習において、構造化技法をベースにした CASE ツールを導入している。本稿では、以上の内容に準拠した「情報システム設計 I・II」の授業設計について述べる。

Lesson Design for the Information System Design/Development that used CASE(Computer Assisted Software Engineering)

Kazuki KAWAMURA

Faculty of Project Design, Miyagi University

Adopting information system learning education in, the information system course of the design information subject the cultivating of a systems engineer is being aimed at. With the middle, the author is taking charge of "information system design I/II" that does the methodology and technique that become needed for the construction of an information system a professor. I am introducing the CASE tool that makes the structure technique the base in the exercises of these subjects. I express about the lesson design of "information system design I/II" that conformed to the aforementioned contents in this manuscript.

1. はじめに

情報システム学教育に関するカリキュラムは、米国の DPMA(Data Processing Management Association)と ACM(Association for Computer Machinery)、欧州の IFIP(International Federation of Information Processing)/BCS、などを中心に検討が進められてきた。一方、我が国では、IFIP/BCS のカリキュラムを参考にして策定された浦らのカリキュラム^{1) 2)}や、コンピュータサイエンスを包含した情報処理学会(IS 部会)のカリキュラム³⁾が提案されている。

いずれのカリキュラムにおいても、情報システム構築のための方法論や開発技術に関連する領域が取り込まれている。たとえば、浦らのカリキュラムでは、専門領域「C240 情報システムの開発」の中の「C243 情報システムの設計方法」「C244 情報システムの開発技術」が相当する。情報処理学会のカリキュラムでは、「対象世界の分析・モデル化論/実習」や「IS 設計・製造・検査論/実習」が相当する。このように、情報システム設計・開発に関する領域は、情報システム学のカリキュラムにおける基幹科目の一つに位置づけられるといえる。

筆者が所属するデザイン情報学科情報システムコースでは、情報システム学のカリキュラム体系を取り入れた教育を実践している。その中で、筆者は「情報システム設計 I・II」という科目を担当している。

「情報システム設計 I・II」では、講義だけでなく演習を取り込み、より実践的な教育を行っている。演習では、パソコンで稼動する CASE ツールを使用している。これによって、課題として提示する小規模な情報システムの設計と開発が短期間に実施できるとともに、実装した情報システムの動作確認やシステム評価まで行うことができる。これによって、より実践的な演習が可能になり、受講した学生にとっても有益な教育効果が期待できる。

ただし、現時点では、「情報システム設計 I」(前期)が終了した時点であり、「情報システム設計 II」(後期)はまだ開講していない。このため、本稿では、「情報システム設計 I・II」の授業設計という視点から論じることとする。なお、後期が終了した時点で、最終的な科目の評価を実施する予定である。

2. 情報システム設計開発に関するカリキュラム

情報システム設計開発に関するカリキュラムとしては、米国の IS97⁴⁾や通産省情報処理技術者試験のカリキュラム⁵⁾があげられる。

IS97 では、「IS97.7-情報システムの分析と論理設計」「IS97.9-物理設計とプログラミング環境を用いた実装」というコースが用意されている。

IS97.7 では、システム開発モデル、要求決定と仕様化、設計、論理、物理、CASE といった項目があげられている。その中の CASE については、「CASE 及び I-CASE、更にその他の方法論を使う(LO-0112)」「市販の CASE ツールを使用し、上流のドキュメントを作成することができる(LO-0113)」というラーニングユニットの学習目標が設定されている。

IS97.9 では、各種方法論、設計開発技法、プログラミング言語、検査、実装といった項目があげられている。その中の設計開発技法については、「アプリケーション設計に関する構造化、イベント駆動、及びオブジェクト指向の方法の違いを特定し、設計及び開発のプロセスに対して、これらの方法がどのような意味を持っているかを説明する」というラーニングユニットの教育目標が設定されている。

情報処理技術者試験のカリキュラムでは、第二種共通カリキュラム第 5 部「システム開発の基礎」、第一種共通カリキュラム第 15 部「応用システム開発能力」、高度情報処理技術者「アプリケーションエンジニア育成カリキュラム」が該当する。

これらのカリキュラムでは、情報システム設計開発について構造化技法をベースに、一部オブジェクト指向技法を取り込むとともに、CASE に関する説明(定義、種類、機能)も取り上げている。しかし、実際の演習に CASE をどのように取り込むかといった具体的な指針は明記されていない。

なお、筆者は、以前中央情報教育研究所において、企業内リーダー養成研修コース「ソフトウェア開発技術」の企画と講師としての参加という機会があった⁶⁾。開発技術としては、構造化技法とオブジェクト指向技法を取り上げた。ただし、構造化技法については机上によるグループ演習(OHP シートによる発表)を、オブジェクト指向技法については演習をせずに講義だけという構成であった。この結果、いずれの技法についても、知識としては習得できても、実際に適用できるまで

に至らないという受講生の感想が聞かれた。

3. 科目「情報システム設計Ⅰ・Ⅱ」の授業設計

2のカリキュラムを参考にした上で、「情報システム設計Ⅰ・Ⅱ」の授業設計を行った。その過程について、いくつかの視点からまとめる。

3.1 受講対象

本科目は、デザイン情報学科情報システムコース3年生(40名)を対象に、いずれも必修で2単位を配当している。また、本学は Semester 制度を導入していることから、「情報システム設計Ⅰ」は前期に、「情報システム設計Ⅱ」は後期に、それぞれ開講している。

本コースでは、1・2年次に、コンピュータサイエンスの基礎科目群(学内情報システム、専門紙購読、線形代数学、コンピュータ数学、コンピュータ科学、コンピュータ基礎Ⅰ・Ⅱ、ハードウェアⅠ・Ⅱ、ソフトウェアⅠ・Ⅱ)、および、事例科目群(各分野における適用業務事例)を開講している⁷⁾。プログラミングに関してはC言語が必修であり、本科目を履修する学生全員は、2年次までに一通りのデータ構造とアルゴリズムについて(レベルの差はあるが)習得済みである。

3.2 授業目標

本コースの教育目標は、「情報システムのデザインとマネジメントを、事業プロジェクトとして推進できるシステムエンジニアの養成」となっている⁷⁾。これは、本コースが事業構想学部に属していることから、単なる情報システムの設計や運用に従事するだけのシステムエンジニアではなく、事業プロジェクトという枠組みの中で、能動的に活躍できるシステムエンジニアの育成を目指していることによる。とはいつても、情報システムの構築に関する知識と技術力は、最低習得すべきものといえる。

以上の教育目標の実現に向けて、各科目が設定されているわけだが、本科目では、次のような授業目標を設定している。

『システムエンジニアとして必要になる情報システム構築のための知識と技能を習得すること、および、最先端の情報システムとその構築技術に対する関心を持ち続けること』

3.3 検討事項

授業を設計する上で、二つの事項について検討した。一つはどのような情報システム設計開発の

方法論や技法を採用するかであり、もう一つは演習をどのように組み込むかということである。

(1) 方法論と技法について

70年代に提唱された構造化技法-構造化分析(SA)/設計(SD)/プログラミング(SP)-は、大規模なメインフレームコンピュータによるオンラインシステムの構築が全盛の頃、大いに普及した。ただし、専用回線に接続された端末は、ノンインテリジェントであるとともに、CUIが主流であった。また、この技法に応じた方法論には、段階的詳細化、情報隠蔽化、抽象化、トップダウンアプローチなどがあげられる。一方、90年代以降に登場してきたオブジェクト指向技法-オブジェクト指向分析(OOA)/設計(OOD)/プログラミング(OOP)-は、ワークステーションやパソコンを中心としたクライアントサーバシステムの構築に採用されるとともに、徐々に普及しつつある。その際のクライアントパソコンは、分散ネットワーク(LAN、インターネット、イントラネット)に接続されるとともに、GUIに変わっている。それだけでなく、最近では、分散オブジェクト指向技術による情報システムの構築も試みられ始めている。

このような状況の中で、構造化技法はソフトウェア開発現場でも開発実績のある技法として定着している。一方、オブジェクト指向技法や分散オブジェクト指向技術については、最近になって標準化が進められ始めており、まだ開発実績も少ないといえる。

それだけでなく、本コースの開講科目は、C言語が中心であり、C++(各ゼミナールでは対応予定)まで対応できていない。このため、実装技術が(構造化プログラミングを適用できる)手続き型言語に依存していることから、それに合わせる必要がある。

以上のことから、本科目では構造化技法を中心に取り上げることにした。ただし、構造化技法については知識だけでなく技能として習得させること、オブジェクト指向技法と分散オブジェクト指向技術に関しては知識として習得させること、をそれぞれ前提条件にした。

(2) 演習について

(1)で述べたように、オブジェクト指向関連技術については知識としての習得を目指すことから講義中心でもよいが、構造化技法については知識だけでなく技能としての習得を目指すことから、講義だけでなく演習まで用意する必要がある。その

際に、机上での演習では、動作確認ができないなど、フィードバックがかけにくいことから、教育経験をうまく現場の仕事に適用できないといったことに留意する必要がある。

これより、本科目の演習では、CASE(Computer Assisted Software Engineering)を用いた情報システム設計開発を行うことにした。構造化技法を採用している CASE には、米国の製品を中心に多くのもの—たとえば、Design/1, リアルタイム SA/SD, SAVER, SEWB/SDF, StP, Teamwork, IEW, TMS/IPSE など—があるが、本学のコンピュータ演習室の環境(サーバー: Windows/NT マシン, クライアント: Windows95 パソコン, ネットワーク: Ethernet LAN)で稼働できる製品でなければならない。また、それほど高価ではなく、アカデミックディスカウント(50 セット分)が適用できる必要もある。

その結果、SCC 社の「CASE トレーナー(Ver2.0)」という製品を採用するに至った。これは、もともと構造化技法習得のための教育用 CASE であったこと、他の教育機関でも採用実績があったこと、スタンドアロンでも稼働すること、などの理由からである。ただし、導入した製品は、Windows3.1 対応であったため、プログラムの一部分を Windows95 用にカスタマイズする必要が生じる。

3. 4 授業のシラバス

科目毎に設計したシラバスを、以下に示す。

(1) 「情報システム設計 I」

講義が中心になる科目である。このため、情報システム構築に必要な知識の習得を目標にする。なお、授業で使用する教科書は、筆者の執筆したもの⁹⁾を用いる。

【第1回：科目ガイダンス】

科目の学習目標、情報システムの歴史的変遷、情報システム学のカリキュラムについて取り上げる。

【第2回：ソフトウェア序論】

ソフトウェアのプロセスモデルとコストモデル、および計算モデル(ソフトウェアパラダイム)について取り上げる。それとともに、ウォーターフォールモデルと手続き型プログラミングパラダイム(実装のためのプログラミング言語は C 言語)を前提にすることを明示する。

【第3回：情報システム分析工程 I】

データ分析として、E-R モデルと正規化技法に

ついて取り上げる。

【第4回：情報システム分析工程 II】

プロセス分析として、構造化分析について取り上げる。

【第5回：情報システム設計工程 I】

入出力設計として、画面設計と帳票設計について取り上げる。画面設計では、CUI だけでなく、GUI についても取り上げる。

【第6回：情報システム設計工程 II】

ファイル設計として、コード設計、レイアウト設計、データベース設計について取り上げる。ただし、C 言語で実装できるファイル(シーケンシャル/ランダムファイル)を前提にするとともに、データベースに関しては関係データベースを取り上げる。関係データベースであれば、SQL インタフェース機能をもつプリプロセッサ(たとえば、Informix-ESQL/C など)を用いることが可能になるからである。

【第7回：情報システム設計工程 III】

プロセス設計として、オンライン系プログラムを対象に構造化設計を、バッチ系プログラムを対象にワーニエ法とジャクソン法を、それぞれ取り上げる。

【第8回：情報システム開発工程 I】

「プログラム=データ構造+アルゴリズム」という視点から、データ構造とアルゴリズムについて、C 言語をベースにして復習する。その際に、入出力処理(画面関連)とファイル処理について再度取り上げる。

【第9回：情報システム開発工程 II】

プログラム開発として、構造化プログラミングについて取り上げる。これも C 言語を前提にしたプログラミングの仕方とトップダウンによる開発手順について説明する。なお、後期の演習では、CASE によりプログラムが自動生成されることを明示する。

【第10回：ソフトウェア開発環境】

ソフトウェアツールと CASE について取り上げる。また、後期に使用する CASE トレーナーについても簡単に触れる。

【第11回：これからの情報技術 I】

オブジェクト指向技法について取り上げる。ここでは、構造化技法の問題点を明らかにした上で、その解決策一つとしてオブジェクト指向技法があること、その変遷と標準化の動向(UML)について取り上げる。

【第12回：これからの情報技術II】

分散オブジェクト指向技術(COM/DCOM, CORBA, JavaRMI)について取り上げる。

【第13回：これからの情報技術III】

インターネット技術による情報システム開発(CGI, Java)と、イントラネット構築について取り上げる。

(2) 「情報システム設計II」

【第1回：科目ガイダンス】

科目の学習目標、演習課題、演習手順について取り上げる。

【第2~4回：CASEトレーナーの操作演習】

「CASEを学ぶ」⁸⁾にしたがい「アルバイト給与計算システム」を例にして、一通りのCASEトレーナーについての操作を演習する。なお、ソースコード生成は、C言語とする。

【第5~12回：課題演習】

小規模な事務処理系の適用業務に関する課題を二つ取り上げ⁹⁾、そのどちらかを選ばせ、個別に演習を実施する。演習の最終目標は、C言語のソースプログラムを生成した上で、Cコンパイラ(Visual C++ Ver1.51)で翻訳し実行プログラムを稼動し動作確認ができるまでとする。

【第13回：課題のグループ発表】

最終的に出来上がった演習課題について、全員でレビューを行わせる。レビューの際に、各人の開発にあたっての工夫点と問題点についても発表させる。

以上の作業を終えた時点で、CASEトレーナーで作成した各ツールのドキュメント一式と作業実績報告書(作業内容、作業時間)をまとめさせ提出させる。

3. 5 CASEトレーナーとその演習

ここでは、CASEトレーナーそのものと、CASEトレーナーを用いた演習内容について取り上げる。

(1) CASEトレーナーの概要

CASEトレーナーの特徴としては、

- ・E-Rダイアグラムをサポート
- ・構造化分析/設計をサポート
- ・プログラミング言語(COBOL: COBOL/2 あるいは WATCOM COBOL, C: WindowsベースのC)を自動生成、このため、プログラミングの必要なし
- ・COBOLの場合はCUIベースの画面レイアウトを、Cの場合はGUIベースの画面レイアウトを、それぞれサポート

・COBOLの場合、いくつかのファイル編成(順次だけでなく、索引順、相対順)をサポートなどがあげられる。これより、ソフトウェア開発工程の上流から下流までを網羅した統合化CASE(Integrated CASE)に位置づけられる。

プログラミングが基本的には不要(ただし、一部変更が必要)なので、開発工程が短縮化できる。この結果、実行プログラムによるシステムの動作確認も可能になる。これが、机上での演習と最も異なる点であり、情報システム設計開発における実体験が、学習効果をより高めることが期待できる。それだけでなく、GUIベースの画面を扱うことができ、C言語によるWindowsシステムのAPIなどを考慮する必要がない。つまり、履修している学生のように、基本的なC言語のプログラミング能力があれば、APIの知識がほとんどなくても、GUIベースの情報システム構築を体験することができる。

また、CASEトレーナーが稼動する環境は、次のようになる¹⁰⁾。

- ・Windows3.1 日本語版が動作するバージョンのMS-DOSVer3.3 日本語版以上、MS-DOS/V, IBM DOS/V
- ・80386_{SX}以上のCPUを搭載し、メインメモリ640KB以上でプロテクトメモリ4MB以上
- ・ハードディスクは、最少構成で6MB以上の空き容量、フルセットで14MB以上の空き容量
- ・3.5インチ(2HD)フロッピーディスク
- ・Windows3.1に対応したディスプレイ、マウス、プリンタ、日本語FEP

これより、スタンドアロンのパソコンでも稼動でき、環境設定に手間がかからないといえる。

CASEトレーナーは、ウォータフォールモデルを前提に、ソフトウェア開発工程を分析フェーズ→設計フェーズ→開発フェーズという手順で進める(図1)。

分析フェーズのデータ分析に関しては、「エンティティ関連ダイアグラム」「エンティティ詳細記述」といったツールが用意されている。「エンティティ関連ダイアグラム」は、データの要件とそれらの関連について記述する図を管理するとともに、エンティティ間の結合関係を明らかにすることによって管理すべき情報要求を表すツールである。「エンティティ詳細記述」は、エンティティを

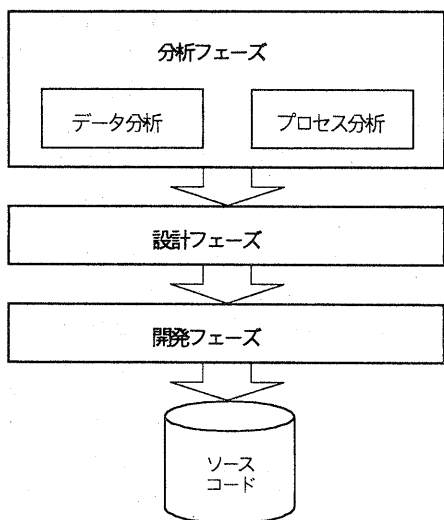


図1. CASE トレーナーの工程

構成する属性，属性の階層，データタイプを記述するツールである。

分析フェーズのプロセス分析に関しては、「データフローダイアグラム」「ミニスペック」といったツールが用意されている。「データフロー図」は、データ処理の流れを記述するツールである。なお、Ver2.0 からは、制御プロセスとイベントフローが新たに追加されており、他のプロセス制御とトリガを表現することができる。「ミニスペック」は、詳細化すべきプロセスを、構造化言語によって記述した仕様を管理するツールである。

設計フェーズに関しては、「データ構造定義」「画面・帳票レイアウト」「構造化チャート」「アクションダイアグラム」「イベントダイアグラム」といったツールが用意されている。「データ構造定義」は、データの物理的な属性を定義するツールである。「画面・帳票レイアウト」は、画面あるいは帳票に出力するためのレイアウトを設計するツールである。画面に関しては、生成するソースコードが COBOL の場合は CUI ベースに、C 言語の場合は GUI ベースに、それぞれレイアウトを作成できるようになっている。「構造化チャート」は、プログラムのモジュール構造を表現するためのツールである。「アクションダイアグラム」は、各モジュールのロジックやモジュール同士の呼出しの定義を記述するツールである。「イベントダイアグラム」は、GUI ベースの画面を作成する際に必要になるイベントを定義するためのツールである。

これにより、リスト、コンボボックス、グループボックス、チェックボックス、プッシュボタン、ラジオボタンなどのウィンドウシステムにおけるコントロールオブジェクトの指定ができるようになっている。

開発フェーズに関しては、ソースコードの生成ツールが用意されている。ソースコードの種類としては、COBOL/2、WATCOM/COBOL、Windows_C のどれかを生成することができる。

(2) 演習手順

演習については、3.4の(2)での述べたようにこちらから演習課題を指示する予定である。

演習の進め方は、次のようになる。

フェーズ1：課題の要求仕様分析

フェーズ2：エンティティ関連ダイアグラムの作成

フェーズ3：エンティティ関連記述の作成

フェーズ4：データフローダイアグラムの作成

フェーズ5：ミニスペックの作成

フェーズ6：データ構造定義の作成

フェーズ7：構造化チャートの作成

フェーズ8：画面レイアウトの作成

フェーズ9：イベントダイアグラムの作成

フェーズ10：帳票レイアウトの作成

フェーズ11：アクションダイアグラムの作成

フェーズ12：ソースコードの生成

フェーズ13：動作確認テスト

フェーズ14：各ドキュメントの印刷

以上の中で、現状の CASE トレーナー(Ver2.0)では、分析フェーズのツールで作成したシステムに関するメタ情報の中のプロセスに関するもの(フェーズ4とフェーズ5)は、設計フェーズのツールに連動しない。このため、DFDやミニスペックの情報から、構造化チャートなどの詳細なプロセスは生成されない点を明らかにしておく必要がある。

また、実際の演習は、コンピュータラボ1で実施することになる。ここに設置されている80台のパソコンの中の50台に、スタンドアロンでの使用を前提に CASE トレーナーをインストールしている。CASE トレーナーは、最初にプロジェクトを作成する必要がある。プロジェクトとは、開発するシステム毎の名前であり、これによってすべてのメタ情報が一元的に管理される。そのプロジェクトの作成には、名前のほかにディレクトリと生成言語を指定する。

そこで、学生全員に同じプロジェクト名をつけさせるとともに、ディレクトリはフロッピーディスク(ドライブ:A)として指定させる。これによって、「アルバイト給与計算システム」の操作演習までは、どのパソコンでもできるように考慮する。ただし、フロッピーディスクでは、アクセス時間が遅く、作業に支障をきたすので、同一演習後の個々の課題作成に関しては、プロジェクト名はそれぞれ別につけさせるとともに、ハードディスク上のディレクトリとする。

3.6 実施状況

本科目は、今年度の前期から開講している。このため、現時点では「情報システム設計I」が終了した段階である。43名の受講生のうち、5名が不合格になっている。その内訳は、レポートの未提出者(2名)、および、筆記試験において60点未満の者(3名)である。

最終授業終了時のアンケート(自由記述)を累計すると、表1のようになった。

表1. 授業評価のアンケート

項目	件数
システム開発の全体像が把握できた	28
CASEを用いた演習をやってみたい	20
構造化技法について興味が沸いた	10
オブジェクト指向技法に関心が高まった	8
CではなくJavaによるシステム開発をしてみたい	3
話が抽象的でよくわからなかった	3
プログラミングにしか興味が無い	1

(複数回答)

これより、前期の「情報システム設計I」については、ほぼ授業目標は達成されたといえる。しかし、実務経験のない学生にとっては情報システムを開発する動機づけがあまり明確になり得ないことや、プログラミングにしか興味を示さない学生もいることが明らかになった。これらについては、次年度の講義内容において対処しなければならない課題といえる。

4. おわりに

以上、情報システム設計開発の方法論・技法に関する科目「情報システム設計I・II」における授業設計について取り上げた。その中での特徴的な点としては、演習方法があげられる。

通常であれば、机上での演習と作成した仕様書によるウォークスルーが中心になる。しかし、これでは、情報システム構築技術を知識だけでなく技能として習得することは難しいといえる。つまり、分析・設計した内容を、いかに実装に結びつ

けるかを体験すること、さらには、実装した情報システムの動作を確認すること、などによって始めてフィードバックがはかられ、情報システム構築のための技能が蓄積できるといえる。

そこで、実装のためプログラミングを行う演習を取り込むことが考えられる。演習時間が十分設定されていれば可能であるが、本学のようなカリキュラム編成上での時間的な制約、および、学生のプログラミングに関する力量の限界があり難しい。このため、CASEを利用してプログラムを生成することによって、実装時間を短縮する演習を組み立てた。その際に、生成されたCプログラムを、自分でトレースして一部変更しなければならず、他人のプログラムを読む演習にもなる。

それとともに、構造化技法(E-R, SA, SD, SP)を支援するCASEを用いることによって、一貫した方法論と技法を習得することができる。自己流の分析や設計に頼ることなく、体系化された構造化の指針にもとづく技法を学ぶことは、将来システムエンジニアを指向する学生にとっても意味あることといえる。

現時点では、「情報システム設計II」を開講していないので、演習まで含めた総合的な授業評価ができないが、個々人がどこまで演習課題を達成できるかを測定したい。それとともに、CASEを用いることによって、どのような情報システム構築に関する学習効果が生じるのかについても把握したい。

今後の課題としては、構造化技法だけではなく、オブジェクト指向技法について、講義だけでなく演習として実施するための授業設計についても検討したい。具体的には、UMLを用いてオブジェクト指向分析設計を行い、それをオブジェクト指向プログラミング言語(C++, Javaなど)を用いて実装するような演習形態である。あるいは、分散オブジェクト指向技術による実装による演習形態である。

参考文献

- 1) 浦昭二, 他: 情報システムの教育体系の確立に関する総合的研究, 平成3-4年度科学研究費補助金(総合研究A)研究成果報告書, 1992年
- 2) 細野公男, 浦昭二: 情報システム人材の教育体系の確立について, 情報処理, Vol.34, No.6, pp.778-788, 1993年
- 3) 情報処理学会編: 大学等における情報システ

- ム学の教育の実態に関する調査研究, 情報処理学会, 1992年
- 4) ACM,AIS,AITP: IS'97 Model Curriculum and Guideline for Undergraduates Degree Programs in Information Systems, Data Base, Vol.28, NO.1, pp.1-94,1997.
 - 5) 通産省編: 高度情報化人材育成標準カリキュラム, 中央情報教育研究所, 1997年
 - 6) 河村一樹: 構造化およびオブジェクト指向によるソフトウェア設計教育の検討, 情報処理学会研究報告, Vol.92, No.57, pp.1-9, 1992年
 - 7) 河村一樹, 藤井章博: 事例研究にもとづくシステムエンジニア育成のカリキュラム, 情報処理学会研究報告, Vol.97, No.125, pp.7-14, 1997年
 - 8) SCC編: CASEを学ぶ, SCC, 1995年
 - 9) 河村一樹: 情報システム設計・開発技術, 近代科学社, 1999年
 - 10) SCC編: CASEトレーナーユーザーガイド, SCC, 1994年
 - 11) 沖山豊, 河村一樹, 北村公彦, 菅野篤: システムの設計と開発, ムイスリ出版, 1990年
 - 12) 河村一樹: 第一種共通テキスト6ソフトウェア工学, 中央情報教育研究所, 1994年
 - 13) 河村一樹: ソフトウェア工学入門, 近代科学社, 1995年
 - 14) 河村一樹: 入門情報科学シリーズ6ソフトウェア工学, ソフトバンク, 1995年