

オープンソースソフトウェアを活用した 教育業務支援システムの開発

佐野 洋

東京外国語大学 外国語学部

E-mail: sano@fs.tufs.ac.jp

概要

不特定多数の学生利用者が、Webブラウザを利用し、インターネット経由で学生情報を閲覧したり管理することのできるグループウェアシステムを開発した。グループウェアの基本設計と開発は、オープンソースであるSkyboard(R)(トップマネジメントサービス(株))を利用した。基本機能を再利用することで、自組織の利用者(学生)が使い易いインタフェースを持つシステム開発を低コストで且つ、短い開発期間で実現することができた。オープンソースウェアの利用促進の一事例として、ソースコードの提供のあり方、活用、修正や変更に必要な条件を検証した。提供側のソフトウェアの汎化の方法論と利用側のコードの再利用の仕方について、ソフトウェア設計上の視点から問題点を挙げる。

Designing and Developing a Groupware by using Open-source Software

SANO, Hiroshi

Tokyo University of Foreign Studies. Faculty of Foreign Studies

Abstract

In this paper a groupware system is discussed. The system we have developed in our university provides all students with useful information about school affairs via the Internet. The main challenge is the reuse of open-source software named Skyboard(R) that has been released by TOP MANAGEMENT SERVICE (TMS) Corp. The re-useability of the open-source software depends on the function and the semantic model of the software design. The function and the semantic model we mentioned here are defined as frameworks of software development process.

Using the TMS-developed software, the development process of the groupware has significant impact on programming performance and cost. Reuse and modification of the software allowed us to easily develop the required software under absolutely strict cost and skill constraints with little paid efforts. Just as open source codes provide us with the re-useable codes of program, also the codes can provide us the clues to writing new codes for functions being required. This paper will focus on the benefits of using open software in conjunction with frameworks of domain analysis.

1 はじめに

本学では、インターネット上で学生が誰でも自由に学務スケジュールや講義情報などの情報閲覧ができるグループウェアTUFs E-SPACE BOARDを開発した。開発と導入のコストを抑えることを目的に、中小企業向けのグループウェア用のオープン・ソフトウェアSkyboard(R)¹(トップマネジメントサービス(株))を利用した。

Skyboard(R)はビジネスユース向けのグループウェア・ソフトウェアである。その機能は、企業組織における一般業務支援である。その機能を教育業務支援のために応用した。オープン・ソフトウェアを利用することによって、低コストによるアプリケーション開発が可能になった。自組織の利用者(学生)が使い易いインタフェースを持つシステム開発を、低コストで且つ短期間のうちに実現することができた。オープンソースウェアの利用促進の一事例として、ソースコードの提供のあり方、活用、修正や変更に必要な条件を検証した。

1.1 教育支援基盤への投資

日本では、20歳～25歳の世代人口が最大数²に達して以来、就学人口は減少の一途を辿っており、小児化が急速に進行している。総人口は、2005年もしくは2004年に最大数に到達すると予想され、その後全人口は、年々減少に転じ、同時に高齢化が進行する。こうした中で教育組織は、生き残りをかけた大競争の中にある。

国立研究機関を含め国立大学の独立行政法人化も議論されている。こうした状況下、教育機関では、教育対象者層の人口構成の変化や動向によって、教育業務形態や運用形態の変更、追加は、今後ますます頻繁に起こり得るものと予想される。業務形態の変更や追加は、教育基盤を支える情報システムにも反映させる必要がある。教育サービスの向上には、魅力的なサービス事業への新規投資だけでなく、絶え間のないシステム機能の見直しや変更、追加と品質維持・管理と継続的な投資が

¹URL <http://www.tms-px.co.jp/BOSS2000>

²2000年1月現在。URL <http://www.stat.go.jp/> を参照。

必要となる。

他教育組織との差異化を迅速に進め、生き残り競争に勝つには、迅速且つ投資コストを抑えながらも情報技術を効果的に活用した教育支援のためのシステム開発が不可欠である。

1.2 情報化投資とTCO

技術誌によれば、近年、加速度的な勢いで情報システムが高度化し、分散化し、複雑化する中、一般企業では、TCO (Total Cost of Ownership. 所有者総コスト)の上昇が強く指摘されている。

情報機器や情報システム導入後の運用・管理に対しコスト支払いの比重が増しているという。組織内の情報機器の操作教育や学習コスト、アプリケーション開発費、情報システムで機能するサービスに対する日常的に発生するカスタマイズやトラブルに対する業務支援等の運用管理コスト、および対エンドユーザー・サービスコスト等が挙げられている。

今後も社会全体の情報化が進む中で、情報システムが高度化・複雑化すること、それに伴って一層多様な情報サービスが提供されることが予想される。情報投資に対するTCOの最適化は、企業にとって重要な課題となっているという。

1.3 コストバランス

教育機関における教育支援のための情報システムのアプリケーション開発、維持と運用等も同様の指摘ができる。

一般に、教育事業や教育業務は労働集約性が高いことが特徴であるが、それは、教育活動の直接的な内容に関わる部分のコスト負担として集中すべきことである。教育活動を支援する基盤設備の運用や維持、基盤設備面のサービスに対する利用者支援サービスは、教育活動の負担コストに含めるべきものであってはならない。資源投入は集中と選択が必要であって無駄な投資は避けなければならない。

重要なことは、情報システムに対する投資コストと、そのシステムが機能した時の教育価値の総

合的なバランスである。教育組織における情報システムのTCOを最適化することが求められている。

本稿では、学生のアカウントを管理したり、教官と学生あるいは学生どうしの情報交換の場を提供するグループウェアシステムの開発について述べる。教育業務支援のためのアプリケーション開発の1事例として、教育サービスの向上の目的で導入したオープンソースの利用経験を取り上げる。製品購入に比べると、初期導入コストがほぼ0に近いこと、コードを直接改変できるので自組織用のカスタマイズに対する投資額が低いこと、システム機能の見直し、変更や追加などの継続投資額が低いことなど、システムのライフサイクルにおけるTCOの軽減が可能である。

第2章では、システム開発工程における投下コストと利用効率について概説する。第3章は、アプリケーション開発工程におけるソフトウェアの再利用性について説明する。オープンソフトウェアを利用したアプリケーションの開発について経緯を含めて説明する。第4章ではオープンソースを利用したシステム開発の問題点を説明する。第5章で、導入に掛る学習コスト負担を検討した結果を基に、オープンソースの提供と利用の有り方について提言する。

2 システム開発/利用のコスト

2.1 開発の生産性/利用の効率性

一般に、生産性は、投入に対する産出の比として表わされる。

$$\text{生産性} = \frac{\text{産出}}{\text{投入}} \quad (1)$$

「産出」を産出量で計測する場合は物的生産性で、産出価値で計測する場合は価値的生産性である。システムの開発に焦点をあて、簡単のため産出をソフトウェア量で計測するものとする。

産出物をソフトウェアとし、投入をソフトウェア開発に充てる労力とすると、ソフトウェアの生産性は時間と労働の配分として表現され次式で現される。

$$(\text{ソフトウェア開発の}) \text{生産性} = \frac{(\text{開発}) \text{コード量}}{\text{労力}} \quad (2)$$

商用ソフトウェアであるか無償ソフトウェアであるかに拘らず、一般に教育組織では、ソフトウェアは諸々の学務サービスの自動化と効率化のための手段として応用することが、主なソフトウェアの利用方法であるので、機能したソフトウェアのコード量とそれに対する投入の比を効率性とし、やはり時間と労働の配分として表現する。

$$(\text{ソフトウェア利用の}) \text{効率性} = \frac{(\text{機能}) \text{コード量}}{\text{労力}} \quad (3)$$

直観的には、オープンソースを利用すれば、ソフトウェアのダウンロードとインストールほどの小さな労力によってソフトウェアが機能するので、ソフトウェア利用の効率性は高い。

近年、TCOの伸長が問題であることは1.2節で指摘した。それはソフトウェア利用の効率性が低下していることを示し、単位(機能)コード量あたりに投入する労力が増加していると言える。換言すれば、機能量に対する投入労力(学習量)によって、ソフトウェア利用の効率性が変化する。

学習量について次の2点を指摘することができる。

1. ソフトウェア機能の複雑さ(の理解)
ソフトウェア自身の複雑さとソフトウェアが実現しようとするドメインのモデルの複雑さ
2. ソフトウェアが機能する情報機器の複雑さ(の理解)
機器自身の複雑さと装置を複合させることによる複雑さ

この2点は初学者に対する情報教育(コンピュータサイエンス教育やコンピュータリテラシー教育)の重要なポイントでもある。

一般に、ソフトウェアを利用するための労力には、(1)提供機能を知る(調査する)時間、(2)機能の動作概念を理解する学習時間と(3)利活用の方

法を考える時間が含まれる。TCOの増大は、製造者側がソフトウェアの機能の複雑さを実現する行数の拡張進度以上に、利用者側の応用のために費やす自己学習時間が大きくなっていることを示す。

2.2 アプリケーション開発工程

一般的なアプリケーション(ソフトウェア)開発工程(図1:文献[1, 2, 3]などから引用)を参照し、利用者側の自己学習が必要な項目の分布を観察する。

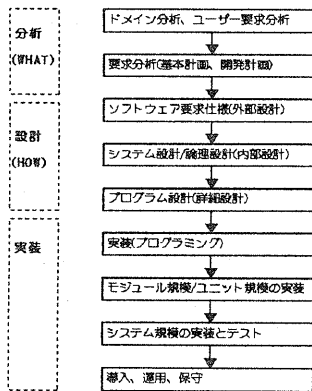


図1: 落水モデルによる一般的な開発工程

利用者が「ソフトウェアを使う」ことは、図1の「実現」囲み部分の成果を利用することである。

「分析」囲み部分の知識の学習には、ドメインを理解する力と分析力が求められ、「設計」囲み部分の知識の学習は、機能を形式化し組織化する能力が必要である。特徴的なことに、商用ソフトウェアであっても、購入可能なのは実現部分の成果であって、「分析」囲み部分と「設計」囲み部分の(知識の)詳細は、一般に公開されない。無償では勿論、対価を支払っても要求仕様書や設計仕様書を手に入れることはできない[1, 2]³。

ソフトウェアを使って、「何を(WHAT)」「どのように(HOW)」機能させているかを知るのには、我々は、通例、商品カタログ、添付の機能説明書

³ 出願された特許、社内技報、学術論文等の参照物があるが、一般の利用者にとっては現実的なドキュメントではない。

や一般流通の関連する技術書籍を参照し学習しなければならない。

問題の第1は、我々が汎用的なドメイン分析の方法論を知らないために、ソフトウェアの要求(機能)仕様を正しく理解することができないことにある。第2には、ソフトウェアでは機能実現の方法は、一般に幾通りも存在するから、実現されたソフトウェアから設計仕様を推論することができないことである。

一般に組織では、何らかの業務フローが存在している。業務分析は、無意識の理解を意識化して業務フローを客観化する作業である。さらにフローを分解し、分割と統合を繰り返し自動化部分を特定して代替技術に転換することで省力化を行う。要求仕様は、技術のことばで語ることが肝要で、設計が滞ったり、機能実現のための可能性が限定されるのは、ニーズは熟知しているが、要求仕様を技術的なことばで仕様に変換するための技術知識に欠ける場合である。

分析のドメインが多岐にわたると、問題解決能力及び、分析技術力に加えて、広範な技術知識が要求される。従って、資金投入が可能な場合、要求分析と設計のための技術知識を購入することが行われる。ソフトウェアの入手後にも可能で、一般にカスタマイズと言われる。

2.3 要求分析と設計方法の改善

機能上の汎用性が高い(ワードプロセッサのように要求仕様が一般化している)ソフトウェアは、分析と設計フェーズの仕様を利用者が知る必要がない。しかし、一般には業態は多様なので、従来から、ソフトウェア開発プロセスの中で、要求分析と設計方法を改善する試みが行われている[1, 2, 4]。

ソフトウェア工学の問題には、正当性/検証性、頑強性、拡張性、再利用性、保全性、互換性、効率性、移植性、使いやすさ、保守がある[1]。一方、ソフトウェア利用の視点からの問題は、開発されたソフトウェアの安全性、品質、信頼性である[1]。

いずれも定量的な測度(メトリクス)の規定が難しく、そのため(現在のところ)ソフトウェア製品の安全性や品質を立証する定格仕様記述は、ほとんど見ることができない。

3 システム開発工程

3.1 開発アプリケーション(目標)

開発システム WWW-DB 連携の学生情報管理ソフトウェア

開発システムは、Web ブラウザーをインタフェースとする情報提供のための学生用グループウェアである。履修案内、開講状況などの学務情報の連絡を始め、学生相互の情報交換サービスの提供、サークル活動の支援、携帯電話からの WEB アクセスによる情報提供等の教育業務を支援することがシステムの目標機能である。基本構成を図2に示す。

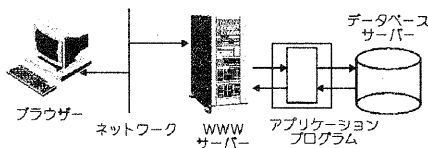


図 2: WWW-DB 基本構成図

3.2 ドメイン特徴

グループウェアは、組織を構成するメンバーの行動管理や知識共有を目的とするソフトウェアである。従って、その(要求)機能は、個人を特定して情報を管理するという基本はあるものの、組織の行動形態(企業であればビジネスモデルに強く依存)と組織員個々人の行動形態(会社、部署の運用形態に強く依存)に左右される。従って、企業向けグループウェアはカスタマイズする必要がある。

教育機関では、グループウェアの持つ個々人のスケジュール管理機能は必要性がなく、情報共有機能に重点が置かれる。機能変更については学生の要求と行動を調査することで仕様を決める。特徴的なことは、入学と卒業によって年度単位でおよそ構成員の1/4が入れ替わることである。データベースのメンバー所属管理の仕組みを変更しなければならない。

3.3 開発か購入か

3.3.1 製品調査

グループウェア製品としては、Lotus Notes(R)、Microsoft Exchange(R)が代表的である。大規模システムでの運用も可能である。一方で、導入コストが高く、維持や運用サービスには担当の技術管理者が必要である。本学では、設備機器制約とコスト制約のため、こうしたグループウェア製品の導入はできない。

機能を限定し、サービス規模を小さくしたグループウェアの1種であるコラボレーションツールも多く販売されている[5]。価格も、5000円(2ユーザー)~100万円(ユーザー数無制限)と導入コストは比較的低い。しかし、テキスト形式でデータを保存する構成になっているなど、本格的なデータベースを採用していないことが多く、サービス規模の拡張性に乏しい。さらに企業向けの製品であるため、自組織(教育組織)の学務フローに適合させるカスタマイズが難しい⁴。

3.3.2 技術調査(学習)

本章では、アプリケーションソフトウェアの利活用を効果的に行うための自己学習量の点から、オープンソフトウェアの効率的な利用法を研究事例として、ソフトウェアが提供する機能の概念理解や技術習得に在る問題を考察する。

図2のシステム開発のための調査項目を挙げる。

1. データベース/データベースサーバー
2. Webサーバー呼び出しのプログラミング方法
3. データベースアプリケーション開発

上記の各項目について、自組織でのアプリケーション開発を前提とした、著者の技術調査項目とその内容を、図3を参照して説明する。

⁴試用版提供する製品が多い。製品価格を抑え、導入後のカスタマイズサービスを商品化することも多い。対象の市場は中小企業もしくは、企業の部門単位である。

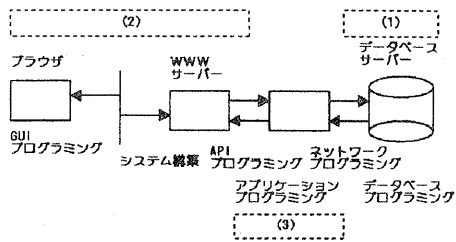


図 3: WWW-DB 機能構成図

(1) データベース/データベースサーバー 中小規模クラスのRDBS(Relational DataBase System)についての調査を実施し、一部 RDBS の導入実験も実施した(DB2, Interbase, PostgreSQL⁵)。各ソフトウェアのオンラインマニュアルページを参照した。

データベースへのアクセス技術については、ODBC/JDBC⁶、Perl モジュールなどを利用してデータベースサーバーへのネットワーク接続試験を実施した。関連技術書を参照した[6, 7]。

(2) Web サーバ呼び出しのプログラミング方法 アプリケーションの開発のフレームワークは、CGI, HTML 埋め込み Script, Java[8], Open-ZOLAR⁷について調査を行った。

(3) データベースアプリケーション開発 データベースを用いるアプリケーションの試験開発を実施した。C++言語、Java 言語を使ったデータベースアプリケーションの小規模な試験開発を実施した。統合開発環境 (IDE⁸)として C++Builder, JBuilder, VisualBasic⁹を導入し、小規模アプリケーションの試作を行っている。言語学習は、C++, Java, Visual Basic, SQL の其々について技術教科書を通じて自学した。

⁵DB2: 日本IBM(株)製, Interbase: インプライズ(株)製, PostgreSQL: URL <http://www.postgresql.org/>

⁶ODBC: Open DataBase Conectivity, JDBC: Java DataBase Conectivity

⁷URL <http://www.isr.co.jp/~muc/openzolar/>

⁸Integrated Development Environment

⁹C++Builder, JBuilder: インプライズ(株)製IDE, VisualBasic: マイクロソフト(株)製IDE

3.4 労力とオープンソース利用

技術調査では、評価のために各種ソフトウェアのインストール、試験のため関連技術書の購読、実装のためプログラム言語の学習をした。取得技術確認のため小規模アプリケーション開発作業が必要である。いずれも、本務以外の空いた時間(終業後や休日)を利用するので、(1)開発に集中できない、(2)継続開発ができない、(3)技術レベルが未熟、あるいは断片的でシステム開発に結びつかない、(4)開発ツールに適宜コスト投入ができない等の問題がある¹⁰。

延べ時間数は特定できない。しかし、約2年間の調査を続けたものの、個々の要素技術の学習に時間を費やして終わった。要素技術を組み合わせアプリケーションを実現するシステム開発のために必要となる学習時間の見積りが困難で、且つ実装の作業時間が取れる見込みがないなどの理由から、これ以上の開発を停止した。

コスト制約のため製品購入は出来ない。技術(開発力と作業時間)制約のためプロジェクト開発が出来ず、このような状況で、オープンソースの利用は有望な選択肢である。

4 オープンソースの活用

文献[4]は、開発システムの品質を改善したり、利益率の向上を図る上で重要なことは、開発のプロセスを測ることであると指摘する。本章はオープンソースを利用したシステムの開発工程を示し、開発過程の経験的な投下コストを説明し、オープンソース利用の対費用効果について議論する。

4.1 ソースコードの入手と利用環境の構築

4.1.1 システム構成の知識

グループウェアの解説誌[9]¹¹や技術誌[3]には、詳細なシステム解説があるので参照する。但し、解説本の説明内容を理解する前提として、基本的な

¹⁰例えば、プログラム言語学習の進度が遅いため、ある IDE ツールは、購入して梱包を解く前に次の版が出荷された。

¹¹オープンソースを提供する会社が解説誌の出版も行っている。

ネットワークの知識、WEBアプリケーションの知識、UNIX オペレーティングシステムに関する背景知識が求められる。

Skyboard は、WWW-DB 連携のアプリケーションを提供する。図4([9]:151 頁から引用)に示す。

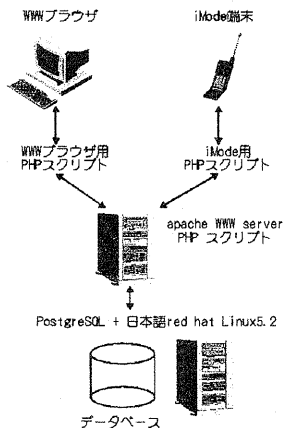


図 4: Skyboard アプリケーション

4.1.2 ドメイン知識

アプリケーションが実現する機能を理解するには、グループウェアに関するモデル知識と、当該ソフトウェアが機能前提としている会社の業務フローに関するドメイン知識が必要である。ドメイン分析結果を機能実現した結果については、解説誌[9]を参照する。

4.2 ソースコードの解読

ソースコードの解読には、PHP3 プログラミング言語の知識が必要である。一部、JavaScript 言語が利用されている。WEB ベースのインタフェースを持つので、基本的な HTML 言語(特に入出力関係のタグの機能)知識が必要である。

データベースへのアクセスには、SQL 言語が使われる。但し、データのFETCHやSTOREなど基本操作に係るものだけで良い。SQL 言語についての入門書レベルである。オープンソースを提供会社が出版する解説本[10]を参照する。

4.3 変更によるシステム開発

4.3.1 機能分析と設計

オープンソースで提供されている機能仕様と業務フローモデルの対応を確認する。制御ブロック、関数名、変数名と機能の対応付けをする。次に、自組織の学務フローについては仕様化し、その要求表現のことはを、オープンソースで提供されている機能仕様のことはに合わせる作業が必要である。約2ヶ月を要した。

4.3.2 プログラミング

基本的にはモデル化されたロジックは変えない。制御部分と視覚化部分の変更が主なプログラミング過程である。複雑な変更は、例えば、データベース項目の変更に伴う SQL 文や出力フォーマットの変更である。簡単な変更は、表示部分の文字列修正等である。

実装では、既習得の Perl 言語の知識を援用し、PHP3 言語のマニュアルを参照し実装を行った。空いた時間を利用するという、過去2年間の技術調査実施過程と同じ時間の使い方、約2ヶ月を要した。1999年11月上旬に、SkyBoard をダウンロードして以来、約4ヶ月でシステムが部分稼働している(2000年1月下旬現在)。

4.4 オープンソースのメトリクス

単純なメトリクスでは、コード行数とコメントの比率、変数名と機能の対応関係、変数名の名前付けの一貫性などがある。また、オンラインドキュメント記載の内容評価などがある。ソースコードの解読と変更部分の特定、新規開発と不要部分の特定などは、コードの可読性と密接に関係する。システムの完動後に稿を改めて報告する。

5 おわりに

オープンソースは、現在、資産コストの低減のために導入されている事例が多い。例えば、無償配布されるオペレーティングシステム(OS)である Linux は、商用 OS に比べて導入コストは低く、幾

つかの教育機関でもクライアントマシンのOSとして利用されている。こうした事例は、導入前提となる技術力が背景に在る場合が多い。

本稿では、オープンソースとして提供されているグループウェアをカスタマイズする労力について、関連技術の学習項目なども含めて事例紹介をした。2年余り断続的に要素技術の調査と技術学習に費やした。自組織内の独自システム開発を断念後、オープンソースを利用し、約4ヵ月の実装作業によって部分稼働している。完全稼働に向けて作業を継続中である。

前述したように、オープンソースは導入のための技術習得に学習コストが発生する。同時に、利用ソフトウェアに対する自己管理責任が発生し、ソフトウェアの導入と運用・品質の維持は、導入者自身が実施しなければならない¹²。

ソフトウェアの開発工程で示したように、ソフトウェアを有効利用するための、ドメインの分析知識や設計上の知識など商用ソフトウェアであっても、利用者が負担する学習量(範囲や程度、時間)に関する議論はほとんどない[11]。情報システム導入後の利益効果を測定する手法の議論が活発化しているように、今後、情報システムやアプリケーションのソフトウェア・ライフサイクルのコストを測定する議論が必要である。

オープンプロジェクト文化へ

作った方が良いか、借用した方が良いか。コスト制約と技術制約の中で決まる。本学のように両者とも制約がある場合には、情報サービスを提供するため、オープンソース利用は有効な手段である。

オープンソースの利活用は、同等機能製品を購入する代替物として「目に見えるコスト」の低減には有効であるが、導入工程でのコスト、および導入後、自組織への適合やサービス機能の維持と運用等、ソフトウェアの利用効率を向上させるには、技術学習負担は大きく、サービス全体のコストバランスを考える上では、まだまだ改善するべ

¹²ネットワーク上の幾つかのサイトでは、技術学習やノウハウ共有等の互助を目的とするサポートサイトもあるほか、一部企業では、オープンソースの技術に対する有償のサポート商品の販売もみられる。

き事柄は多い。

文献[12]で指摘するように情報教育についての支援環境の構築例もある。教育組織は、教材開発や教育方法論の開発など従来からも活発に行われてきたが、今後は、開発した組織内だけで利用するのではなく、オープンプロジェクト文化の枠組の中で共通利用ができる仕組みを開発するべきである。

謝辞

トップマネジメントシステムズ(株)には、システム開発にあたり支援をいただいた。ここに感謝する。

参考文献

- [1] Bヘンダーソン・セラーズ著、大森健児訳：「オブジェクト指向ソフトウェア工学」、海文堂、1994。
- [2] E. ヨードン、松原友夫訳：「オブジェクト指向システム設計」、プレスティンホールトッパン、1995。
- [3] 「企業ユーザーのためのLinux入門～実践編」、no.73～no.76、日経オープンシステムズ、1999年4月～7月、日経BP社、1999。
- [4] M. ロレンツ、J. キード著、宇治邦明監訳、オージス総研訳：「オブジェクト指向ソフトウェアメトリクス」、プレスティンホール、1995。
- [5] 「Web コラボレーションツール」、pp164-pp174、日経インターネットテクノロジー、6月号、1999。
- [6] 玉川竹春著、三田典玄監修：「C++Builder データベースプログラミング」、プレスティンホール、1998。
- [7] ケント・ライズドルフ+ケン・ヘンダーソン著、郡司芳昭訳、三田典玄監修：「C++Builder 入門」、プレスティンホール、1997。
- [8] ジェーソン J. マンジャー著、スリーエーシステムズ訳：「実例で学ぶ JavaScript 入門」、日経BP社、1997。
- [9] トップマネジメントサービス著：「オープンソースでつくるグループウェア」、LOCUS社、1999。
- [10] トップマネジメントサービス著：「SQL データベース構築入門」、LOCUS社、1999。
- [11] 小林正幸他著：「教育機関ソリューションフレームワーク」、pp11-pp15、NEC 技法、Vol52 No.7、NEC、1999。
- [12] 山岸純子、他：「小・中・高等学校における情報教育支援体制確立の試み」、情報処理学会研究報告、99-CE-53、1999。