

デジタル講義資料の再利用支援についての考察

Reusing Digital Course Material

ネットワーク情報学部 松永 賢次
School of Network and Information Kenji MATSUNAGA

Keywords : Faculty Development, e-Learning

1. はじめに

大学の講義は、かつては、10年間も同じ講義ノートを利用した授業が存在したと言われているが、現在ではFD (Faculty Development) が提唱され、受講生にあわせた適切な内容の講義をすることが求められている。そのような状況においても教員が、講義で扱う内容の範囲を、短い期間で大きく変えることは考えにくい。多くの場合は、項目順序の入れ替え、扱う内容の量の増減(詳細度の変化)、例や事例の入れ替え、といった部分的な変更をしていきながら、講義の内容をよりの確にしていくことが考えられる。例えば筆者の場合、毎年継続的に講義をしている授業科目でも、部分的変更のための見直し作業に、講義時間の2～3倍の時間を費やすのが普通である。

また、ある教員にとって、全般的な内容範囲が同じものでも、その講義をA学部にするときと、B学部にするときでは、2つの学部の学生の背景知識が異なる場合がある。専門科目として講義をするときと、教養科目として講義をするときもある。そのような場合、片や背景知識の説明を増やしたり、片やより深い内容を追加したり、といった講義内容の調節が行われる。あるいは市民講座のような一般市民を対象とする場合、高校生相手に入門的な講義をする場合など、同じネタを使って様々な場面で講義・講演をしたとしても、聴衆の背景知識や講義時間によって、説明内容をかなり調節しなければならないということである。

現在、Microsoft社のプレゼンテーションソフトウェアPowerPointを、デジタル講義資料作成ソフトウェアとして使用する教員が増えている。PowerPointのようなアプリケーションを用いることの利点は、資料の編集が容易なことにある。スライドの追加・削除・順序変更、あるスライド内の内容の追加・削除といった編集操作により、適切な講義内容に調節していくことができる。このような編集操作により、既存のデジタル講義資料をもとに新たな講義資料を作成することは、編集操作の量が少なければ「修正」と呼べるものである。「修正」は、一部の字面を直す程度のもので、紙の上で作業をするのであれば赤入れに相当する。修正よりさらに進んだ段階では、内容の部分を入れ替えたりする。PowerPointにおける、スライド単位での入れ替えが多くなると、既存の講義資料をもとに、新しい講義資料

を作成するという「再利用」に近い状況になる。

PowerPointを用いて、「再利用」という状況に近い編集を行うことは、かなり時間がかかる作業である。PowerPointのファイルは、スライドの集まりとなっている。スライドが構造化されているわけではないので、自分が必要する複数のスライドを、PowerPointのファイルの中から抜き出すためには、目で内容を追って確かめていく必要がある。それらを集めて1つのファイルを形成したとしても、スライドをセットして見たときに、対象とする講義に対して適切になっているかどうか判断しなければならない。部分的に不適切であると考えられる場合は、一部を修正・追加することになる。

このような一連の作業を、コンピュータプログラムの開発における再利用の場面と比較してみると、効率が悪いことは自明である。コンピュータプログラムのすべてのコードを読んで再利用をしようとしているようなものである。プログラム開発では、関数、ファイル、モジュール、クラスといったような、コードをまとめた単位で再利用をすることができるし、それを支援するためのアプリケーションソフトウェアも用意されている。プログラム開発と同じように、コンピュータを利用してデジタル講義資料の再利用を支援することは、有意義な研究と考えられる。

本稿は、この研究の中間的な報告である。3つの部分的検討について述べていく。2節では、コンピュータプログラムの再利用からアナログ的に考えたときに、どのように講義資料を再利用する方法があるのかについて述べる。3節では、筆者の研究室で試作した再利用ツールについて紹介し、その利点と問題点について述べる。4節では、筆者の実際の講義に対し講義資料の再利用の実体について調査した結果について述べ、研究の前提として想定していた状況に不足はないのか述べる。最後に5節で、これらの部分的な検討結果から考えられる、これからの研究の方向について述べる。

2. コンピュータプログラムの再利用とデジタル講義資料の再利用のアナロジー

本節では、コンピュータプログラムの再利用の仕組みとそれがどのように発展してきたのか概観し[Orfali 1995, Fayad 1997]、その上で、デジタル講義資料(授業用コンテ

ンツ)でそれが概念的にどのような行為にアナログ的に該当するのか述べる。

2. 1 コンピュータプログラムの再利用

コンピュータプログラム同士では、同じような処理内容が頻繁に利用されるため、一度記述したプログラムを再利用することで、コード記述のための時間を節約してきた。すでに実用されているコンピュータプログラムは、実際の動作環境で正しく動作していることが経験的に確認されているという長所もある。

プログラムの再利用には、ソースコードレベルで提供されるものと、オブジェクトコード(機械語)レベルで提供されるものがある。オブジェクトコードレベルの再利用は、再利用者がコードを改変しないということを意味し、それをブラックボックスと見なして再利用することになる。一方、ソースコードレベルの再利用は、再利用者がコードを改変する可能性がある。中身が見えるという意味で、ホワイトボックスという呼び方がされる。再利用されるプログラムの作成者が、オープンに再利用を認める場合、プログラムの著作権等の権利保護のために、改変できないオブジェクトコードレベルの再利用方法を選ぶことが多い。一方で、一つの企業の中で、あるプロジェクトが作成したプログラムコードを再利用する場合には、プログラムのより詳細な内容について理解するために、ソースコードレベルの再利用をする可能性がある。

コンピュータプログラムでは、「関数」(あるいは「手続き」)といった単位で部分をとりまとめることができる。このことは、再利用という観点から大きな利点をもっている。

1. 関数は、名前付けされているので、検索が容易である。
2. 関数単位で修正がなされるので、バージョン管理(その関数の様々な実現を管理すると言ってもよい)をすることで、入れ替えが容易になる。
3. 複数の関数の連続を、1つの関数とする、といった関数の親子関係を作ることで、様々な抽象度の関数を構築できる。

2. 2 コンピュータプログラムの再利用方式の発展

初期の頃からの代表的な再利用手法は「ライブラリ」と呼ばれるものである(図1(a))。ライブラリは、「関数」や「手続き」として記述されたプログラムコードを、プログラマが記述するプログラムから呼び出すことができるようにするものである。ライブラリとして代表的なものは、外部装置とのデータのやりとりをする入出力に関するものである。例えば、C言語であればprintfという文字出力のためのライブラリ関数が用意され、プログラマは出力装置が何かを知らずにプログラムを記述することができる。「ライブラリ」を再利用する場合には、「幹」のプログラムのロジックは、プログラマが用意する。ライブラリは、必要に応じて利用される「部分」になる。

一方、1990年代のオブジェクト指向技術の発展により出

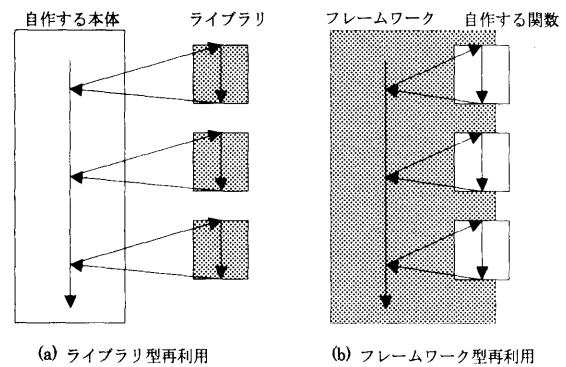


図1 コンピュータプログラムの再利用の形態

現してきた再利用の概念には、「コンポーネント」、「フレームワーク」がある(図1(b))。こちらは、「幹」を再利用しようという考えである。様々な処理業務の主たる流れは共通化しており、それらは再利用し、開発業務プログラム固有の部分のみを変更することでコンピュータプログラムを開発できるということである。フレームワーク(コンポーネント)は、変更可能な部分が関数となっており、その関数をプログラマが書き直すことでカスタマイズできる。フレームワーク(コンポーネント)を構築する上で、頻繁に変更する可能性がある部分(ホットスポット)を、変更可能な関数としての確に抽出できるかどうかは鍵になる。

フレームワーク(コンポーネント)において、プログラマが変更する関数内では、入出力などのライブラリ関数を利用することは当然行われる。したがって「フレームワーク型再利用」の中で「ライブラリ型再利用」は共存することができる。

またオブジェクト指向技術における、別の再利用技術として「デザインパターン」がある。デザインパターンは、プログラムのある一部の構造を再利用するもので、ある抽象的な機能を共通に持ち、具体的な機能の変更点を記述する。

2. 3 デジタル講義資料とのアナログ的な比較

デジタル講義資料の再利用を、「コンピュータプログラム」の再利用と比較して考えてみたい。

幹の部分は作り直す、部分は既存のものを活用していくという「ライブラリ型再利用」の考え方をとるとすると、プログラム例、事例、統計資料といった各種資料は再利用し、それに対する説明の流れは講義者が新たに作っていくといったやり方がある。

一方、主たる講義の流れはすでにあるものを活用していき、そこから参照する部分は作り直していくという「フレームワーク型再利用」の考え方をとるとすると、再利用する講義から参照する図、演習課題などを作り込むやり方となる。

デザインパターンに近い考え方としては、講義の進行のベストプラクティスを再利用することが考えられる。ある講師の講義の進行方法が、学生の理解を高めることがわかった場合、そのような進行方法を再利用していくものである。

以上に検討したように、デジタル講義資料（授業用コンテンツ）の再利用は、概念的には、コンピュータプログラムの再利用と同じようにできることがわかる。

3. デジタル講義資料の作成支援ツールの試作と再利用支援への問題点

筆者の研究室では、デジタル講義資料の開発支援するための研究を行い、いくつかのツールを試作してきた[畠山2000, 畠山2001, 井上2003]。

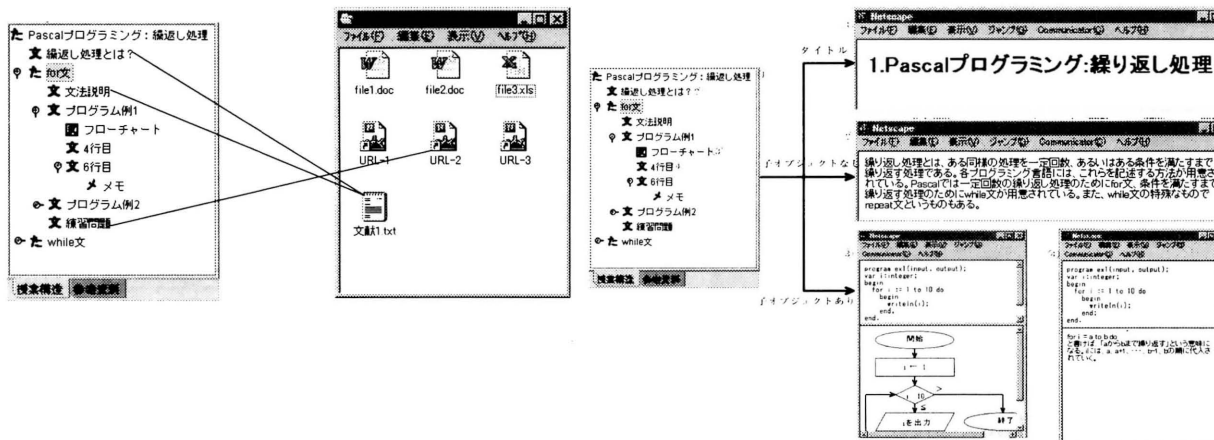


図2 階層型管理ツールの概念 [畠山2000, 畠山2001]

3.1 階層構造をなすコンテンツの開発支援

[畠山2000, 畠山2001]では、デジタル講義資料を構成する要素(タイトル, 文章, 図, メモなど)を階層管理し、各要素を編集できるツールを試作した(図2)。編集した授業コンテンツを1つのプレゼンテーションとしてとりまとめて、Web上で配信することができる。このツールは、講義者が通常の対面講義のために利用することもできるが、オンライン学習用コンテンツの作成ツールとして使用することも考慮している。

デジタル講義資料を階層構造化することは、2節で述べたようなプログラムの構造化と同じ特徴である、再利用の単位の明確化につながる。編集者は様々な粒度の再利用や修正といった編集をすることができる。この研究では、階層構造化することで編集を容易にしたが、再利用の観点からは、各構成要素が多くなってきたときに、適切なものを検索する仕組みが必要となってくる。

3.2 LOMを利用したコンテンツの検索・合成支援

そこで[井上2003]では、デジタル講義資料にLOM (Learning Object Metadata) 形式[LOM2002]のメタデータを付加し、メタデータと共にサーバ上に登録し、検索できるようにし、それらを組み合わせて再利用できるツールを開発した。多くのデータがデジタル化され、インターネットを介してそれらを配布できるようになると、検索するということは、次にそれを再利用するということにつながる。検索できた情報は、自分のコンピュータに取り込み、それ

を再利用する。コンピュータに取り込むことが著作権上、問題がある場合には、ハイパーリンクをはって関連教材として示すこともある。これも広い意味で再利用と言ってよい。

メタデータは、データにつけられたデータであり、主としてデータを検索するために作られる。図書館における図書の書誌情報が、メタデータの伝統的な代表例である。近年では、インターネット上に様々な情報・コンテンツが流

通しているため、それらの検索のためにメタデータが使われる。メタデータの表現形式は、XMLを用いることが多く、新聞記事やBLOG等で利用されているRSS (Rich Site Summary)は一般ユーザが恩恵を受けていることであろう。LOMはXMLに準拠したメタデータ形式であり、既存に用意されているXML検索ツールやプログラミング言語ライブラリを用いて、容易に検索することができる。また、LOMは国際標準であるため、グローバルな教育コンテンツの検索への可能性を持つことができる。

この研究では、LOMの様々な要素の値を検索することができるようになったが、より高度な再利用をするためには、オブジェクト間の関係についてどのような記述をしていくべきか検討していく余地がある。3.1節で紹介した畠山の研究では、構成要素の説明文, 図, 表, 証明, 例, 事例, 演習問題で再利用できるようになっており、文書処理システムLaTeXの構造ともほぼ一致している。これらの構成要素は、単独で利用されるものではないので、構成要素と説明したい概念とをリンクで結んでネットワーク化し、リンクに情報を付加することが適当であると考えられる。

4. デジタル講義資料の再利用はあり得るのか

講義資料を再利用するという事は、直感的にはありうることである。ここでは、筆者がネットワーク情報学部で担当している2つの講義（インターネット情報システム、プログラミング入門）をもとに、再利用の可能性について

調査をした結果を示す。

4. 1 事例1：インターネット情報システム

この講義は、インターネットの仕組み、World Wide Web (WWW)の仕組み、WWWを主としたインターネット情報システムについて概説をする。ネットワーク情報学部2年次以上に配当されている科目である。2002年後期に1度目の講義を行い、2003年～2005年(前期)に至るまで4シリーズの講義を行った。講義資料のほとんどは、PowerPointを用いて作成している。講義の基礎となる教科書[小泉2001]は一貫しており、基本的な講義の流れは4年間変わっていない。同じネットワーク情報学部生を対象として毎年開講する科目では、講義資料は小さな「修正」の積み重ねで済むのかどうか検討したい。

2002年度後期のスライド(153枚)と2005年度前期のスライド(197枚)を比較したものを表1にまとめた。

表1 「インターネット情報システム」における2002年と2005年のスライド変更

章		2002	2005	追加	削除	修正	同一	(2-A)以	(b)の修
		スライド 総数	スライド 総数	スライド 総数	スライド 総数	スライド 総数	スライド 総数	外的追加	正
	ガイダンス(教科書外)	13	20	7	0	6	7	6	0
1	インターネットのメカニズムとWebの機能	15	0	0	15	0	0	0	0
2	インターネットプロトコルとHTTP	31	52	25	4	20	7	21	8
3	サーバサイドクライアントサイドのWeb技術	31	41	10	0	7	24	8	2
4	Web構築のためのさまざまな最新技術	26	33	6	1	13	14	4	5
5	WebサーバとWebアプリケーションサーバの構築	22	29	7	0	1	21	3	1
6	Webサーバのセキュリティ対策	15	22	3	0	3	13	0	0
	合計	153	197	58	20	50	86	42	16

スライドの追加には、

- (1) 2002年度のスライドの説明項目に含まれていない内容のものが新たに登場した場合。
- (2) 2002年度のスライドの中にあった項目であるがより詳しい説明のためにスライドを追加した場合。

(2-A) 文章の増強

(2-B) 補足説明のためのデータを掲載

(2-C) 補足説明のための事例を掲載

がある。これらのうち、(2-A)は「修正」と見ることができるが、それ以外は、既存の説明を再利用しながら新しい説明を追加したという「再利用」と見なすことができ、全体で42個のスライドが相当した。

スライドの修正には、

- (a) 文字の修正・追加
- (b) 図の追加
- (c) 例の追加
- (d) 参考URLの修正・追加
- (e) データ値の修正・追加

が考えられる。この中で(b)図の追加のみ「再利用」と見なすことにし、全体で16個のスライドが該当した。

スライドの追加(1)は、この3年間の中で、かつては必ずしも知る必要はなかった項目が、現在ではネットワーク情報学部生として知っておくべきものになってきたことが原因である。例えば、ADSLや無線LANの普及にともない、「家庭に無線LAN対応のブロードバンドルータを設置して家庭内LANを構築する」ために必要な知識は、ネットワーク情報学部生にとって知らなければならない項目になってきている。「大学に設置された学習支援のためのサーバ型システムについて、その動作を概念的に説明できる」、「ワームの攻撃方法の概要を理解し、それを防ぐ方法を知る」といったことも、将来、組織のネットワーク情報化の指導的な立場に立つネットワーク情報学部生にとって重要となってきている項目である。これらは、講義の外からの要因によって引き起こされた変更と考えられる。

スライドの追加(2)、およびスライドの修正は、講義を試みた結果、学生の理解が不十分であると感じた場合に内容を加える、あるいは黒板等で補足説明をしていたものを取り入れるという結果によっておこっている。これらは、講義の中からの要因によって引き起こされた変更と考えられる。

153個のスライド(2002年度)に対して、3年後に42個のスライド追加、16個の修正という量は、小さな修正という量よりは多く、また新しいものになっているというほど量は多くはなく、再利用しながら発展していると考えられる。このように、ネットワーク情報学部生という同じ対象だったとしても、講義実施年が異なれば、構成要素の部分の追加・変更がなされるという「再利用」が起こっていることがわかる。

4. 2 事例2：プログラミングの入門科目

2004年後期に、ネットワーク情報学部の「プログラミング入門」を担当した。本来、1年生配当科目であるが、筆

者が担当したクラスは、3年次以上の再履修クラスである。この講義を担当するにあたって、1997年～2000年にかけて作成していた、経営学部情報管理学科の「コンピュータ実習」(1年次配当)をもとに講義資料を作成した。

プログラミング入門とコンピュータ実習は、ともに、プログラミング未経験者を対象としたプログラミングの入門科目であり、講義の基本的な達成目標は一致しているがいくつかの点で違いがある。

1つはコンピュータ実習ではPascal言語を使用しているのに対して、プログラミング入門はC言語を使用していることである。この2つのプログラミング言語は、ともにAlgol系統に属し、また考案された時期も1970年前後とほぼ近いことから、概念的にはほとんど同じと言ってもよい。また、学生たちが将来、様々なプログラミング言語を用いてプログラムを作成する可能性が高いことから、入門科目においては共通する概念を強調し、C言語やPascal言語のみに固有の話題に深く立ち入らないように、講義内容を心がけている。最も変化が少ない場合には、コードの記述のみがPascalからCに変わるだけの可能性があり、説明は同じものを再利用できれば望ましい。

PowerPointで作成した「プログラミング入門」(C言語)の教材は、「コンピュータ実習」(Pascal言語)用にWordで作成した教材をもとに作った。これらは形態が違うので比較が難しい。そこで、プログラミング入門の教材を、もし逆にPascal言語用にするとしたら、

- (1) そのまま使える
 - (2) 文章の説明部分は使えるが、コード例は言語を変えなければならない
 - (3) 同様の概念を扱うためには説明が変わる
- の3分類のどれに相当するか数え上げた(表2)。(3)に該当するのは、ヘッダーファイル、switch文の制御、for文の考え方の違い、文字列型の違いが主なところである。約半分がコード例を変更しなければならないが、コード例を切り分けることができるのであれば、80%以上のスライドを再利用できることがわかった。

表2 プログラミング入門の言語変更(CからPascal)によるスライドの影響

	合計	そのまま使用できる	コード例を変更すれば使用できる	全体の変更が必要
1 プログラムの基本	19	4	13	2
2 分岐	15	5	9	1
3 繰り返し	22	6	9	7
4 配列	14	8	5	1
5 多重ループと多次元配列	14	8	6	0
6 関数	13	3	10	0
7 文字列	12	1	3	8
合計	109	35	55	19

コンピュータの操作を伴うような科目では、アプリケーションのバージョンアップ、使用するアプリケーションや言語の変更、といったことが起こる可能性があるが、ある程度変更を想定して講義資料を作成していれば、再利用が

可能であることがわかる。

5. 今後の課題

この節では、本研究をさらに発展させていくための課題について述べていく。

(1) 小さな部分の変更に対する再利用

デジタル講義資料の再利用が、個人(あるいは小さな人数)を支援することを目的とするのであれば、変更可能な部分の切り出しをどうすべきかを検討していかなければならない。これは、フレームワークのホットスポットの考え方と類似性がある。

PascalからCに言語を変更することで例をあげれば、線形探索のプログラム例は全体を交換しなければならないのだろうか? 10行程度とはいっても、このくらいの量があれば、切り分けられる単位と見なすことができる。デジタル講義資料作成管理支援ツールを構築したときに、別のファイルやオブジェクトとして管理対象と見なすことができる。線形探索のプログラムの解説部分が、比較演算子の違い(Pascalの=, Cの==)以外がすべて同じような場合はどうだろうか? この程度で、まったく別の管理単位を用意しなければならないようでは、再利用可能とはいえない。再利用可能な教材とするためには、言語固有の記述(ここでは演算子そのものを示してしまう)はしないようにする、という規律も考えられるが、受講生に親切とは言えない。

講義資料のある小さな部分の変更というのは、データ値やURLを最新のものに変更するといったような文脈でもありうることである。コンピュータプログラムの世界でも、外部環境の変化によって変化しやすい値(例えば税率)は、そこだけすぐに変更できるように切り出されていることが多い。デジタル講義資料の世界でも、外部環境に影響されやすいものを管理する仕組みを用意することが考えられる。作成者は、影響される可能性がある部分に何らかのマーキングをしておき、それを後で再利用するときには、支援ツールがそれを一覧化して、再利用者が短時間で変更を判断できるようにするといった方法が考えられる。

(2) 大規模な再利用に向けて

将来は、大学や学会等で、大規模に教材を再利用する仕組みが導入されることが考えられる。そのような場合、妥当な資料を検索する仕掛けを用意していかなければならない。

コンピュータプログラミングの再利用は、「関数」という単位で行われる。フレームワークの立場では、再利用しない変更可能な部分が「関数」となる。関数が結合するかどうかは、名前と型の一致(インタフェースの一致)によって判断される。

講義資料の場合には、コンピュータプログラムの関数のような明確なインタフェースを用意できないのが現状であ

る。資料を講義に付けても、それが妥当なのかどうかわかるすべがない。関数ではインタフェースが一致したものが候補になるので、検索する際に絞り込みが可能であるが、講義資料ではできない。

もちろんコンピュータプログラムの場合でも、インタフェースがいくらか一致したからといって、その計算が妥当かどうかは最終的にはセマンティクスの問題になる。その関数が、どのような計算をするものなのかわかって、正しい使い方かどうかわかる。関数がどのような計算をするのか示すためには、コメントとしてプログラムに挿入されることが多い。コメントは、コンパイル時には無視されるものなので、プログラミング言語の機構の外で解決していると考えられる。Javadocのように、一定の基準でプログラム内にコメントを書くことによって、関数の仕様ドキュメントを作るというのがその一例である。Javaのような膨大なライブラリ群は、パッケージにまとめられ、それが階層的に管理されるという仕組みに、検索の容易性の基礎になっていると思われる。教材においてもそのような仕組みを基礎においた管理ツールを作成していく必要がある。

参考文献

- [Fayad 1997] Fayad, M.E. and Schmidt, D.C.: Object-Oriented Application Frameworks, Communication of the ACM, Vol. 40, No. 10 (Oct. 1997), pp. 32-38.
- [LOM2002] IEEE Standard for Learning Object Metadata, IEEE 1484.12.1-2002.
- [Orfali 1995] Orfali, R., Harkey, D. and Edwards, J.: The Essential Distributed Objects Survival Guide, John Wiley & Sons Inc, 1995.
- [井上2003] 井上貴司: 教育における教材の検索と合成支援, 専修大学経営学部情報管理学科卒業論文, 2003年1月.
- [小泉2001] 小泉修: 図解でわかる Web 技術のすべて——HTTP からサーバサイド構成まで, 日本実業出版社, 2001年.
- [畠山2000] 畠山英明, 松永賢次: 遠隔教育における教材開発支援環境, 情報処理学会第61回全国大会, 2000年10月.
- [畠山2001] 畠山英明: 遠隔教育のための教材開発支援環境, 専修大学経営学部情報管理学科卒業論文, 2001年1月.

本稿は、2002年度専修大学個別研究助成「デジタルメディア教材を再利用可能にするためのシステムの設計と構築」を受けて行った研究成果である。