

Excel VBA を用いた履修管理支援システムの構築とその運用

The Construction and Operation of the Learning Management Support System using Excel VBA

池 村 努*

要旨

学生の履修を支援するシステムとして履修管理システムの活用が進んでいる。WBTと共にWEB上で動作する履修管理システムが一般的になりつつあるが、一方で導入にあたり設備の新規導入や更新が必要となっている。今回はパーソナルコンピュータ上で動作するExcelを用いた履修管理システムの構築を行い、小規模システムでの履修管理支援システムの導入と、運用実験を行った。

キーワード：履修支援／履修管理システム／学生支援／Excel VBA／LMS

1. はじめに

著者の所属するコミュニティ文化学科では、学生は多くの科目の中から自分だけの学びを選択し、同時に資格取得を目指している。本学科で取得可能な資格数は11あり、個々の資格取得のためには個々に規定された課目を履修し単位を取得する必要がある。資格はそれぞれの発行団体が定める科目の単位を修得し、所定の試験などに合格することによって取得できる。ところが、複数の資格取得を目指している場合には取得要件を満たしているか、あるいは何が不足しているかについての判断を下しにくくなっていく。そこで本学科では履修条件や修得単位数の集計表をまとめた「履修ノート」を学生に与え、単位取得状況を記録させることにより、一人ひとりの学生が自分自身で管理する体制を構築した。これにより、学生は希望する資格を取得するための条件を自ら把握し、条件を満たすための履修計画を立てることができるようになった。

一方指導に当たる教員は学科に所属する学生全員についての履修状況を把握し、それぞれに適した指導を行わなければならない、学期初めの履修登

録の時期にはいくつもの単位計算を実施して学生からの相談に対応する必要があった。そのため、学生が希望している資格についての取得要件の確認や履修指導はできても、残り数科目の履修で取得要件を満たす可能性のある資格についてのシミュレーションまでは手が回らないことが多かった。また、学生側の一方的な思い込みにより履修の取り消しや変更が行われることがあり、資格取得が困難になるという事態も生じていた。

そこで、Excelの表計算機能を用いることにより、卒業要件と資格取得要件を満たしているかの確認ができる履修支援ブックの作成と、学生支援を円滑に行うことができるよう履修管理支援システムを作成することとした。専用サーバーを用意することなく、パーソナルコンピュータ上でどうさせることのできる本システムの活用により履修支援の効率化と、学生指導の確実性を向上させることを目的とした。

2. 研究概要

本研究では履修管理支援システムとしての①「履修管理支援ブック」の作成、②「履修管理支援ブック」の効率的運用を行うためのExcel VBAアプリケーションの開発、③「履修管理支援ブック」の試験運用と改良を行っている。

* Tsutomu IKEMURA
北陸学院大学短期大学部 コミュニティ文化学科
情報科学

「履修管理支援ブック」を作成する Excel のバージョンを、抽出条件やセルの強調表示などが強化されている Excel 2007 とした。「履修管理支援ブック」を作成するにあたり、先に成績データから単位取得状況を抽出するための関数と、取得条件を満たすかどうかの判断を行う関数の妥当性を確認するため、特定の資格に限定した「資格取得管理ブック」を作成した。「資格取得管理ブック」はカリキュラム年度別に作成し、ブック内に当該カリキュラムで開講された科目の成績データシートと、資格取得要件の取得状況確認シートが格納されている (図 1)。

32	33580062	27	新谷 久美		0	0	0
33	33580356	28	神野 絵里奈		0	0	0
35	33580011	30	田井 亜希		0	0	0
36	33580526	31	高橋 世奈		0	0	0
37	33580551	32	武井 麻友美		0	0	0
38	33580704	33	竹本 晴香		0	0	0
39	33580577	34	竹村 真季		0	0	0
40	33580263	35	棚橋 千春		0	0	0
41	33580747	36	田辺 成美		0	0	0

図 1 資格取得管理ブック

「資格取得管理ブック」の成績データには、教務課の学務システムから提供される CSV 形式の「履修登録データ」から転記する。「履修登録データ」には全学生の成績なども含まれているため、コミュニティ文化学科以外の学生など抽出に不要なデータについてはあらかじめ取り除いた状態で提供してもらっている。表 1 に 2008 年度の「履修登録データ」の一部を記す。

表 1 履修登録データ (一部)

	A	B	C	D	E
1	レコード番号	履修時期	曜日	時限	番号
2	20070122	20072	5	4	61
3	20070123	20072	5	4	61
4	20070124	20072	5	4	61
5	20070125	20072	5	4	61
6	20070126	20072	5	4	61
7	20070127	20072	5	4	61

「履修登録データ」にはレコード番号や曜日データなど成績の抽出には不要な箇所が入っている (表 2)。これら不要データを取り除き、データ抽出と「資格取得管理ブック」にデータの転記を行うためのマクロ機能の開発を行った。

マクロの動作が確認できた後、卒業要件について確認を行う「履修管理支援ブック」を作成し、卒業要件の達成状況と、学科で取得可能な 11 資格の取得状況について確認を行うための関数を構

築した。

「履修管理支援ブック」が完成した段階で、実際に履修指導に用い、問題点の洗い出しと、改善ポイントの明確化を行い、改良を行った。

3. 履修管理支援ブック開発

全体の方針として、フォームを用いガイダンスに従って実行する方式とする。抽出条件は後々の編集が容易になるよう、別のファイルを設定。処理手順は以下の 3 ステップに分けて実行する。

Step1 「履修登録データ」データの事前処理

Step2 成績データからの抽出処理

Step3 集計表へのデータ転記

またファイル構成を、上記方針に基づいて以下の 4 種類りとした。

- ・ 07to08 科目コード .xlsx
- ・ 成績管理支援ブック .xlsm
- ・ 履修登録データ
- ・ テンプレート類

表 2 履修登録データ (2008 年度書式)

列	タイトル	書式	要/不要
A	レコード番号	文字列	不要
B	履修時期	文字列	要
C	曜日	文字列	不要
D	時限	文字列	不要
E	番号	文字列	不要
F	学籍番号	文字列	要
G	カリキュラム	数字	要
H	科目	文字列	要
I	学科	文字列	不要
J	GSK7033	文字列	不要
K	担当者	文字列	不要
L	成績	文字列	不要
M	素点	数字	要
N	GSK7007	文字列	不要
O	履修クラス	文字列	不要
P	履修名列	数字	不要
Q	GSK7010	数字	不要
R	GSK7011	文字列	不要
S	GSK7021	文字列	不要
T	GSK7022	文字列	不要
U	GSK7031	文字列	不要
V	GSK7032	文字列	不要
W	GSK70A1	文字列	不要
X	GSK70A2	文字列	不要
Y	GSK70A3	文字列	不要
Z	科目名	文字列	不要
AA	氏名	文字列	不要
AB	名列	文字列	不要

3.1 テンプレート「情報処理士取得確認ブック」の作成

初めに「資格取得管理ブック」の一つとして、情報処理士の取得要件を確認する「情報処理士取得確認ブック」を作成した。同資格は資格取得のための要件が複雑に設定されているため、ブックを他に流用する際に条件を簡略化することで対応できることが期待できるためである。ブック中に抽出された成績を格納するシートと、成績を集計するシート、データ確認用の名簿及び取得要件を記述したシートで構成されている。情報処理士取得に関わる科目数は52科目あるため、成績を集計するシートは科目数分の52配置することになる。シート名には科目コードを用いている。

成績を格納するシート内には、表2のデータ中、「履修時期」「学籍番号」「科目」「素点」を転記する。転記する際の列番号及び並び替えについて表3の通りとしている。

表3 転記する際の並び順

列	タイトル	書式	並び替え	優先順位
A	履修時期	数値	降順	1
B	学籍番号	数値	昇順	3
C	科目	数値	昇順	2
D	素点	数値	降順	4

表3に記した以外のデータは現段階の本システムでは扱っていない。履修時期の優先順位を高く、降順に並べ替えることにより、再履修した場合のデータを優先的に抽出する事が出来る。

成績を集計するシートでは、学生を行方向に、科目を列方向に配置する。学生データは名列番号と学籍番号を持ち、科目は科目コードと単位数、資格に関わる条件（科目群）つ。集計の際には検索キーとして学籍番号と科目コードを用いるようにしている。両者が交叉するセルには、表引き関数を用いて、該当するデータが存在するかの確認を行う。もしデータが存在しない場合は空欄を表示し、データが存在する場合には素点の値が60以上であれば「1」を、60未満であれば「0」を、履修中であれば「S」を表示するようになっている。

情報処理士取得には、必修科目とI～IVの科目群をそれぞれ定められた単位数取得する必要がある。学生データには、科目群ごとに集計するセル

を設けてあり、単位数に「1」あるいは「0」を掛け合わせた数値を合計して表示するようにしてある。その集計結果を基に、資格取得の判定を行うことができるようになった。

さらに履修時期の情報を用いて、履修中の科目も「1」と表示するようにして、資格取得の見込みがあるかどうか判断することができるシートを設けた。

これらにより、適切な成績データを与えることで学生の資格取得状況が一覧として把握できるようになった。「情報処理士取得確認ブック」を原型として、「履修管理支援ブック」を作成し、データ処理の確認を行った。

3.2 履修登録データ加工マクロの作成

3.1に述べたとおりのデータを「履修登録データ」から抽出し、「履修管理支援ブック」の成績を格納するシートに転記するために、まず履修登録データの内容の確認を行った。データの内容は表1、表2に示したとおりである。これを表3に示したデータに集約するための操作を可能な限り自動化して行えるようにExcel VBAを用いたマクロを構築する。

「07to08科目コード.xlsx」は、4つのシートから構成される。シート「科目コード」「抽出条件」「資格リスト」「名簿」。年度に併せて訂正が必要なファイルは一括してこのファイルにまとめる。「抽出条件」には資格ごとに成績の検索を行う際の検索値を記述する。また、図3で用いるリストの値も保存し、必要に応じて修正を行うことができるように構成している。

マクロを含む「成績管理支援ブック.xlsx」は、

図2 フォーム「成績管理支援ツール」

ファイルを開くとフォーム「成績管理支援ツール」(図2)が自動的に表示される。フォーム上には3つのボタンを配置し、それぞれにマクロを登録する。上からStep.1「成績データ前処理」、Step.2「処理画面表示」、Step.3「データコピー実行」となっている。また、ワークシート上には手順を記述し、実行時の手助けとなるようにしている。

3.2.1 Step1 成績データ前処理

Step1「成績データ前処理」にはマクロ「成績前処理マクロ」と「素点処理マクロ」が連続して実行されるように設定してある。

i. 「成績前処理マクロ」

「履修登録データ」を選択し、次に実行されるマクロが扱うことのできる形式にデータ変換をする。前述の通り、「履修登録データ」には不要な部分が多く存在するため、不要部分の削除と、データの並び替え、そして抽出処理を行う年度に合わせたデータの加工までを行う。実行の手順は以下の通りである。

- ① 「履修登録データ」を開くためのダイアログボックスを表示させる。その中から、「履修登録データ」のファイルを選択し開く。
- ② 開かれたファイルが「履修登録データ」を含んでいるかを判定する。判定方法はセルB1の内容を確認する事によって行う。「履修登録データ」であれば、表2の通りセルB1に「履修時期」が入っている。もし異なる場合は処理を中断する。
- ③ 表2で「削除対象」としてある列、A・C・D・E・I・J・K・L・N・O・P・Q・R・S・T・U・V・W・X・Y・Z・AA・ABの削除を行う。
- ④ 表引き関数を使用するため、文字列で保存されている「履修登録データ」を数値化する。この処理を「履修時期」「学籍番号」「科目」に対して行う。
- ⑤ データの並び替えとデータ列の入れ替えを行う。優先順位と並び替えの基準及び入れ替える順番は表4に示す通りである。
- ⑥ 処理が完了したデータを「一時ファイル.xlsx」として名前を付け、「履修登録データ」と同じフォルダに保存する。

- ⑦ 次にフォーム「素点処理」(図3)が表示される。

表4 「一時ファイル」形式

列	タイトル	書式	並び替え	優先順位
A	履修時期	数値	降順	1
B	学籍番号	数値	昇順	4
C	科目	数値	昇順	3
D	素点	数値	降順	5
E	カリキュラム	数値	昇順	2

ii. 「素点処理マクロ」

プルダウンメニューの2007年度前期から2010年度後期まで、4年間8セメスターの中から処理を行いたいセメスターを選択して「実行」ボタンを押す。ここでは、終了したセメスターまでを選択することを想定している。選んだセメスターに併せて変数「sotenhosei」に0～7の数値を設定し素点処理を行う。

図3 フォーム「素点処理」

「履修登録データ」のフィールド「素点」にそれぞれの学生が取得した点数が記入されている。履修放棄や、期末試験受験資格喪失の場合、0点が素点として記入されるが、履修中の科目も0点として素点が記入されており、データ上は見分けがつかない。そのため素点処理とは、「履修登録データ」の素点について、終了したセメスターの素点を1～59点の範囲で仮の点数を素点の代わりに入力する操作である。本成績処理では1点を素点として記録するようにした。

- ① 素点処理マクロを実行すると、ファイル「一時ファイル.xlsx」をアクティブにする。
- ② これから作成するシート「抽出条件」と同名のシートがファイル「一時ファイル.xlsx」に存在するかどうか確認を行う。存

在しなければシートを追加し、シート名を「抽出条件」とする。

- ③ シート「抽出条件」が存在し、何らかのデータが入力されている場合、この後の処理の妨げになる可能性があるため、不要となるデータを消去する。
- ④ シート「抽出条件」のセル A1 に「履修時期」、B1 に「素点」と入力する。
- ⑤ フォーム「素点処理」で選んだセメスターにより得られた変数「sotenhosei」の値を元に、抽出条件をセル A1 ～ B9 に入力し抽出を実行する。抽出条件は素点が“0”で、かつカリキュラムの値が与えられたセメスターを満たす値となる。
- ⑥ D 列の素点を選択し、値を“1”とする。
- ⑦ 次の処理に備え、シート「抽出条件」に入力した条件を全て削除する。
- ⑧ フォーム「素点処理」を閉じ、フォーム「成績管理支援ツール」(図 2) を表示、成績前処理マクロを終了する。

3.2.2 Step2 処理画面表示

図 4 フォーム「資格/分野選択フォーム」

Step.2「処理画面表示」を押すと、フォーム「資格/分野選択フォーム」(図 4) を表示する。フォームは大きく二つの部分に分かれ、上には年度を選択する部分。下には処理を実行する部分となっている。処理は成績抽出用と、テンプレート作成用に画面を切り替えられるようになっている。テン

プレート作成マクロは、テンプレートを構成するシートを自動的に作成するようになっている。

i. 「抽出処理マクロ」

フォーム「成績管理支援ツール」Step.2 のボタン「処理画面表示」をクリックすると、フォーム「資格/分野選択フォーム」が表示される。同時にファイル「07to08 科目コード.xlsx」を表示し、ファイル「一時ファイル.xlsx」が表示されていない場合は「一時ファイル.xlsx」も表示する。最新の成績データに対し、成績データの前処理を行ったデータが一時ファイルとして存在する場合は、この「処理画面表示」から操作を行うことができる。

手順として抽出を行う学生の入学年度を指定する。年度の指定により、ファイル「07to08 科目コード.xlsx」に対し抽出処理が行われ、ユーザーフォーム 1 下のプルダウンメニューの内容が変化する。プルダウンメニューから抽出したい資格を選び、「資格を決定し成績を抽出」ボタンを押すと、資格に合わせた抽出をファイル「一時ファイル.xlsx」に対し実行する。卒業要件について抽出する場合は「年度を決定し成績を抽出」ボタンを押すと、年度に合わせた抽出をファイル「一時ファイル.xlsx」に対し実行する。

ii. 「シート追加マクロ」

マクロ「シート追加」では、ファイル「一時ファイル.xlsx」にシートを追加し、科目とカリキュラムを基準に抽出した成績データのコピーを行う。このマクロは資格と卒業要件のどちらにも対応するように処理の統一化を図っている。処理の始めに、ファイル「一時ファイル.xlsx」のシート「抽出条件」に抽出条件として「科目」と「カリキュラム」を入力するセルの準備を行う。次にファイル「07to08 科目コード.xlsx」のシート「科目コード」から「科目」のコードをコピーし、ファイル「一時ファイル.xlsx」のシート「抽出条件」に科目のリストとして貼り付ける。貼り付けた際に、科目数を変数「cnt」として保存する。これらの値を用いてシートの追加と成績データの抽出を行う。シート追加と成績データの抽出、及びデータのコピーは科目の数だけ繰り返すため、変数

「cnt」の回数繰り返すようにループ処理を行っている。シート追加は「科目」のコードを上から順に選択して科目コードをシート名として作成。抽出を行う際は、変数「optbtn」の値を元に「カリキュラム」の値を2007あるいは2008として入力。科目はファイル「07to08科目コード.xlsx」のシート「科目コード」から「科目」のコードを上から順に設定して抽出を行い、抽出された結果を該当のシートにコピーする。このとき、抽出結果が存在しない科目については、データのコピーを行わない。この繰り返し処理を終了後、フォーム「成績管理支援ツール」を表示してマクロ「シート追加」を終了する。

マクロの実行の様子を次に示す。

- ① シート追加マクロを実行すると、まずファイル「一時ファイル.xlsx」をアクティブにする。
- ② これから作成するシート「抽出条件」と同名のシートがファイルに存在するかの確認を行う。存在しなければシートを追加し、シート名を「抽出条件」とする。
- ③ もしもシート「抽出条件」にデータが存在する場合、この後の処理の妨げになる場合があるので、不要となるデータを消去する。
- ④ シート「抽出条件」のセルA1に「科目」、B1に「カリキュラム」と入力する。
- ⑤ 変数「optbtn」の値が“1”であれば、セルB2に“2007”と入力し、“2”であれば、セルB2に“2008”と入力する。
- ⑥ フォーム「資格／分野選択フォーム」を不可視状態にする。

	A	B	C	D	E
1	科目	カリキュラム		科目コード	
2		2007		98	
3				99	
4				7300	
5				7601	
6				7602	
7				7603	
8					
9					
10					

図5 科目コードを貼り付けた状態

- ⑦ 条件によって抽出された状態にあるファイル「07to08科目コード.xlsx」のシート「科

目コード」から、A列の「科目」をコピーし、ファイル「一時ファイル.xlsx」のシート「抽出条件」のD列に貼り付ける。図7では、医療関連資格の科目コードを貼り付けた状態を表している。

- ⑧ 変数「kamoku」を配列変数として要素数を指定しないで定義、「cnt」と「i」を整数型として定義する。
- ⑨ D列の行数を確認し、行数から1を引いた値を変数「cnt」に設定する。図7の状態ではD列は7行目までであるので、変数「cnt」は“6”が設定された状態になる。
- ⑩ 配列変数「kamoku」の上限を変数「cnt」から1を引いた値に設定する。図5では“5”が設定された状態になる。

	A	B	C	D	E
1	科目	カリキュラム		科目コード	i=0
2	98	2007		98	i=1
3				99	i=2
4				7300	i=3
5		kamoku(i)		7601	i=4
6				7602	i=5
7				7603	
8					
9					
10					

図6 セルD2の値を格納

- ⑪ 変数「i」が0から上限値 (cnt - 1)、になるまで、6回以下⑫から⑱の処理を繰り返す。
- ⑫ 配列変数「kamoku(i)」に「i + 2行目, D列」のセルの値を格納する。
- ⑬ 抽出条件のセルA2に配列変数「kamoku(i)」の値を入力する。図6では1回目の処理なので、変数「i」の値は“0”のため、「kamoku(i)」にセルD2の値を格納している。これでセルA1:B2の範囲に抽出条件が作成された。
- ⑭ 得られた「カリキュラム」と「科目」を元にシート「成績」の抽出を行う。
- ⑮ 新しいシートを作成し、配列変数「kamoku(i)」の値をシート名とする。
- ⑯ シート「成績」のセルA1から下方に向けて選択を行い、アクティブセルとなったセルにデータが存在しない場合は抽出結果が

28	20071	33580270	98	85
29	20071	33580288	98	79
30	20071	33580296	98	83
31	20071	33580300	98	60
32	20071	33580318	98	85
33	20071	33580326	98	79
34	20071	33580334	98	85
35	20071	33580342	98	80

図7 シート追加と成績データコピー

無いものと判断して、抽出結果のコピーを行わない。

- ⑰ 抽出結果が存在する場合、シート「成績」の抽出結果を作成したシートにコピーする。(図7)
- ⑱ 次の変数「i」を実行する。全ての処理が完了したら図1のフォーム「成績管理支援ツール」を表示してマクロを終了する。

3.2.3 Step2データコピー実行

一時ファイルに抽出した成績データを、あらかじめ用意したテンプレートにコピーする。

実行時のマクロの動作を以下に示す。

i. 「データコピーマクロ」

- ① 変数「fname」「mypath」「atai」「row」「cel」を文字列形として定義する。
- ② 変数「kamoku」を配列変数として要素数を指定しないで定義する。
- ③ 「cnt」と「i」を整数型として定義する。
- ④ ファイル「07to08科目コード.xlsx」のシート「資格リスト」を選択状態にする。
- ⑤ Step.2の状態により3つに分岐する。(図8)

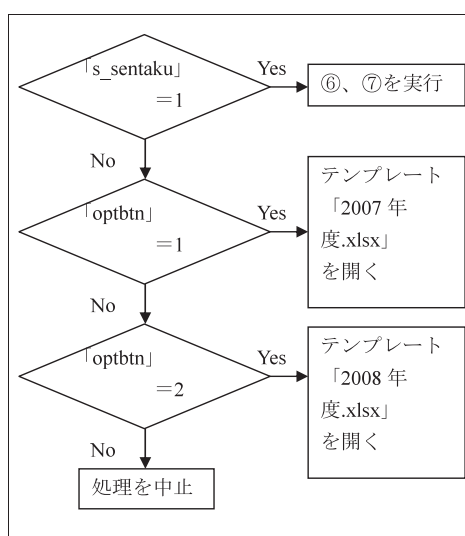


図8 処理の分岐

- ⑥ 変数「s_sentaku」の値が1の時、変数「row」にフォーム「資格/分野選択フォーム」のプルダウンメニューから設定された変数「sikaku」に1を加えた値を設定、変数「cel」に変数「optbtn」の値を設定、変数「atai」はシート「資格リスト」の範囲(K2:L17)の中から「cel」を列方向、「row」を行方向に検索してテンプレートのファイル名(表5)を設定する。図9では、テンプレートを検索する様子を示す。2007年度が選択され、プルダウンメニューから「医療管理秘書士」が選択されたとすると、変数「optbtn」には“1”が、「sikaku」には“9”が設定され、変数「row」には“10”が設定される。この値を元に、表5から列方向に1列目、行方向に「10」行目の値を検索すると、ファイル「¥医療関連 07.xlsx」が得られる。
- ⑦ 今開かれているファイルのパスを変数「mypath」に設定し、そのパスから変数「atai」に設定したファイル名のテンプレートを開く。
- ⑧ 変数「s_sentaku」の値が1ではなく、か

表5 資格用テンプレート抽出表

	2007年度資格用	2008年度資格用
	1	2
0	¥司書資格 07. xlsx	¥司書資格 08. xlsx
1	¥幼免関連 07. xlsx	¥幼免関連 08. xlsx
2	¥幼免関連 07. xlsx	¥幼免関連 08. xlsx
3	¥中免資格. xlsx	¥子ども英語 08. xlsx
4	¥子ども英語 07. xlsx	¥ビジネス 08. xlsx
5	¥情報関連 07. xlsx	¥ビジネス 08. xlsx
6	¥情報関連 07. xlsx	¥情報関連 08. xlsx
7	¥プレゼン 07. xlsx	¥情報関連 08. xlsx
8	¥介護保険. xlsx	¥プレゼン 08. xlsx
9	¥医療関連 07. xlsx	¥医療関連 08. xlsx
10	¥医療関連 07. xlsx	¥医療関連 08. xlsx
11	¥ホームヘルパー. xlsx	¥レクイン 08. xlsx
12	¥社会福祉主事. xlsx	¥E コース 08. xlsx
13	¥レクイン 07. xlsx	¥シスアド 08. xlsx
14	¥E コース 07. xlsx	
15	¥フード. xlsx	

		Col	
		1	2
Row	0	¥司書資格07.xlsx	¥司書資格08.xlsx
	1	¥幼免関連07.xlsx	¥幼免関連08.xlsx
	2	¥幼免関連07.xlsx	¥幼免関連08.xlsx
	3	¥中免資格.xlsx	¥子ども英語08.xlsx
	4	¥子ども英語07.xlsx	¥ビジネス08.xlsx
	5	¥情報関連07.xlsx	¥ビジネス08.xlsx
	6	¥情報関連07.xlsx	¥情報関連08.xlsx
	7	¥プレゼン07.xlsx	¥情報関連08.xlsx
	8	¥介護保険.xlsx	¥プレゼン08.xlsx
	9	¥医療関連07.xlsx	¥医療関連08.xlsx
	10	¥医療関連07.xlsx	¥医療関連08.xlsx
	11	¥ホームヘルパー.xlsx	¥レクイン08.xlsx
	12	¥社会福祉主事.xlsx	¥Eコース08.xlsx
	13	¥レクイン07.xlsx	¥シスアド08.xlsx
	14	¥Eコース07.xlsx	
	15	¥フード.xlsx	

図9 テンプレートの検索

つ変数「optbtn」が1の場合、2007年度の成績データが抽出されていると判断してファイル「2007年度生.xlsx」を開く。

- ⑨ 変数「s_sentaku」の値が1ではなく、かつ変数「optbtn」が2の場合、2008年度の成績データが抽出されていると判断してファイル「2008年度生.xlsx」を開く。
- ⑩ 変数「fname」に今開かれたファイルのファイル名を設定する。
- ⑪ ファイル「一時ファイル.xlsx」のシート「抽出条件」を選択し、D列の行数を確認。行数から1を減じた値を変数「cnt」に設定する。
- ⑫ 配列変数「kamoku」の上限を変数「cnt」から1を引いた値に設定する。
- ⑬ 変数「i」が0から上限値（cnt - 1）になるまで以下⑮から⑰の処理を繰り返す。
- ⑭ 配列変数「kamoku(i)」に「i + 2行目, D列」のセルのデータを格納する。
- ⑮ シート「kamoku(i)」を選択して、セルA1から下方に向けて選択を行い、アクティブセルとなったセルにデータが存在しない場合は抽出結果が無いものと判断する。
- ⑯ 抽出結果が存在する場合、シート「kamoku(i)」の範囲（A2:D2）から下方方向に範囲選択してコピーし、テンプレートのシート「kamoku(i)」のセルA2以下に貼り付ける。
- ⑰ ファイル「一時ファイル.xlsx」のシート「抽

出条件」を選択し、次の変数「i」を実行する。全ての処理が完了したら、不可視状態にしていたフォーム「資格／分野選択フォーム」を削除、テンプレートを表示する。

- ⑱ メッセージボックスに「完了しました。」というメッセージを表示させ、メッセージボックスが閉じられたら、ファイル「一時ファイル.xlsx」と「07to08科目コード.xlsx」を削除する。
- 19 最後に開かれているテンプレートファイルのファイル名「fname」に、作業を行った日付を加えて別名保存して完了する。

4. 履修管理支援ブックの試験運用と改良

2007年度生と2008年度生の卒業要件に関わるテンプレート、それぞれの年度毎の資格に関わるテンプレートを作成し、データ抽出及び履修指導への試験運用を行った。ここでは卒業要件の確認を行うための「履修管理支援ブック」について述べる。

従来手作業で単位取得状況と卒業要件の確認を行っていた場合、学生1名ずつについて行っていた確認作業を、本システムを用いることで、同一カリキュラムの学生について全学生分をまとめて行うことが出来るようになった。3つのステップを実行する所用時間はおよそ3分程度で終了している。1名の確認を手作業で行うのとほぼ同じ時間であった。正確さに関しては、テンプレートと、抽出条件をまとめたファイルの精度に影響されるため、カリキュラムごとの確認が重要である。2008年度に初期段階のマクロが完成した際、「履修登録データ」を用いて手作業で作成した成績管理データとの比較を行った。結果、テンプレートに記述したデータ集計用関数の設定ミスで異なった結果が導き出された。設定の見直しにより手作業で得られる比較用の結果と同様の結果が導き出され、改善されたことが確認された。

本システムの完成により、2008年度入学生の履修指導及び、資格取得支援に向けた指導の作業性が向上した。また、個々の学生の資格取得に向けた指導ポイントが明確になった。学生が取得する意志を持っていなかった資格について、何らかの科目を追加履修すれば資格取得に繋がるという

アドバイスを加えることができるようになった。このことで学生支援システムの第一歩に繋がったと思う。

5. 2009年度に向けての改良

本システムは基本部分を同一システムのままにして、周辺ファイルを変更することにより、カリキュラムの変更に対応できるよう構成している。その特徴を活かして、2009年度カリキュラムに対応させることを試みた。

まず「履修登録データ」の形式について確認をした。表6に示すとおり、2008年度から2009年度にかけてデータの形式が変化している。また無くなったデータがあるなど多くの変更がみられるため、並べ替えの処理を行う必要が生じている。現段階ではシステムが受入れ可能な形式への処理は手作業のままで行っている。

表6 「履修登録データ」書式の変化

列	09		08	
A	—	—	レコード番号	20080122
B	履修時期	20091	履修時期	20082
C	曜日	4	曜日	5
D	時限	5	時限	4
E	番号	61	番号	61
F	学籍番号	33590271	学籍番号	33580016
G	カリキュラム年度	2009	カリキュラム	2008
H	科目コード	0053	科目	0051
I	学科	60	学科	60
J	—	—	GSK7033	0
K	担当者	10001	担当者	10001
L	成績	0	成績評価	6
M	素点	0	素点	60
N	開講期	1	開講期	1
O	履修クラス	1600A	履修クラス	1600A
P	名列	001	名列	046
Q	氏名	石川 かおり	氏名	山口 美恵
R	科目名	生涯スポーツ I	科目名	健康論
S	学年	1	—	—
T	分野	0	—	—
U	クラス	A	—	—
V	クラス名列	—	1	—
W	コマ	—	1	—
シート名称	履修者名簿		成績	

次に検索条件などをまとめた「08to09科目コード」の内容を2008年度及び2009年度カリキュラム対応に変更した。開講科目、どの資格に関連がある科目なのかというデータを、学生要覧に照らして確認、修正を行った。

「08to09科目コード」の変更を受けて「履修管理支援ブック」本体のマクロについて確認を行った。結果、2008年度から2009年度にかけて実施されたカリキュラムの見直しにより、マクロの一部に手を加える必要が生じたため一部変更した。変更箇所は取得可能資格数の変化と、「08to09科目コード」の検索条件記述セルの変更に伴う検索

範囲の記述の書き換えとなっている。

最後に、テンプレートの変更を行った。2008年度に作成したテンプレートは、卒業要件と資格取得条件の確認を別のテンプレートで行う方式としていた。これを一体化し、「履修管理支援ブック」の中に資格の取得要件を集計するシートを追加し、卒業要件単位取得状況を集計するシートの中に資格取得状況及びその見込みを転記するように修正した。この変更により、資格取得状況の確認のために、別々に行っていた作業を一括で実施することが可能となり作業効率が向上した。

以上の変更により、2009年度後期授業開始時には、資格取得のアドバイスと、単位取得状況の確認に役立てることができるなど、履修指導を本システム活用により効率的に行うことができた。

6. まとめと考察

本「履修管理支援システム」構築の目的は、きっかけとしては特定の資格取得状況確認の効率化であった。実際にテンプレートファイルを作成し、「履修登録データ」からテンプレートファイルへの転記機能を作成する内に、履修管理支援システムとしての要素を持たせることに成功したと思う。

また、資格取得状況確認のためにデータの転記を手作業で行っていた際には、3～4時間掛けて作業をしていたこともあり、履修管理作業の効率化に寄与したと思う。

このシステムをさらに強化、改良するために、本システムについて考察したところ、次のような問題点が見つかった。

- ・ データに前年度の学生データがあると、処理がオーバーフローする点
- ・ 教務課より渡される「履修登録データ」のフィールドと、読込データフィールドとの一致
- ・ 地域関連科目としてのシティカレッジ読込
- ・ シミュレーション機能の追加
- ・ 成績データ修正機能の追加
- ・ 履修プラン提示機能の追加
- ・ マクロ更新作業の複雑さ
- ・ 履修の前提条件判定機能

ここに挙げた問題点の改良によってさらに LMS としての機能を向上させ、処理手順の単純化、システムの改良を継続していきたい。将来的

には他学科での運用についても検討し、専用のシステムを持たない環境での汎用 LMS の完成を目指したい。