

表計算ソフトマクロによる自動グラフ作成と XSLT スタイルシートの利用

末 木 俊 之

Automatic graph production by spreadsheet soft macro
and use of XSLT style sheet

Toshiyuki SUEKI

1. 表計算ソフトマクロの利用

学生の演習では表計算ソフトとして Excel を教えている。最近は一般的なビジネス用途としての表計算関連の検定向け内容の演習が主となり、マクロを教える機会は少なくなった。表計算マクロは、処理を自動化するための手段であり、実際は、Visual Basic Application という名称の Basic 言語のプログラムとして記述するものとなっている。表計算のシート上のセルをプログラムでの処理対象(オブジェクト)として容易に扱うことができるのでプログラミングを学ぶためには、比較的取り付き易く、敷居が低いものと思われる。例えば、Cells(1,1)=32などと記述すれば、簡単にシートの1行1列目のセルに32という数字を書き込むプログラムとなる。セルというオブジェクトを扱う記述がちょうどプログラム言語の2次元配列の記述と似ているため配列の代わりにセルを使うこともできるので便利である。また変数、配列内に格納されたデータを表計算シート上の任意の場所に書き出して容易にデバッグすることができる。

しかし、もう少し実用的な、特にビジネスツールとしてのマクロ利用演習を考えた場合には急に難しいものとなるようである。ビジネス上で日々発生するデータを管理するようなマクロ

を作製しようとするすると急に敷居が高くなる。表計算ソフトはデータを管理するという面から見ると使いにくい面がある。データを管理するという面では、やはりデータベースソフトが向いているようである。データを登録・削除・変更するようなフォーム(画面)と自動処理用プログラム(マクロ)を作成する場合にはどちらかと言えばデータベースソフト上にシステムを構築するほうが容易である。やはり表計算ソフトは、データベースソフトから抽出したデータを基に分析作業を実施したり、図・表を作成する段階で使用するソフトと言えるだろう。

繰り返し処理、条件分岐などのプログラミング技法の基本を学んでも、その応用でちょっと実用的なマクロを作製しようとする適切な応用例があまり見当たらないように思われる。プログラミングによる条件分岐を使わなくても、大抵のことは、if関数などの関数を使った計算式でできてしまうということもある。

唯一条件分けに基づくセルの色分けを行うのが、マクロのちょっとした応用としては有用であるように思われる。例えば全国の高等学校データ一覧シートがある場合、高校コードから高校種別(国立、公立、私立学校など)が判別できるので、マクロにて高校種別に色分けしてセ

ルの塗りつぶしなどを行うことができる。

複雑なマクロを作製するのではなく、割合簡単にマクロの応用が可能な事例はあまり見当たらないようであったが、グラフを自動的に作製するマクロはそれほど難しいものでもなく、かつ実用性があるのではないだろうかと思いついた。

もちろんグラフを作成するのは、表計算ソフトでは手作業で簡単にできることである。しかし系列が多数存在するグラフ（折れ線グラフなら、線が何本もグラフ内に引かれるようなグラフ）は、とても1枚のグラフ図では表現できない。1系列データごとに1枚のグラフを作成する必要があるだろう。系列ごとに多数のグラフを作製するような場合、またはある系列のみ抽出してグラフを作製するような場合には、手作業では効率が悪すぎる。自動的に、連続的にグラフを作製し、必要であれば自動的に印刷するようなマクロが有用ではないだろうか。

大学で使われる事例としては、高校別入学者数の年度別推移グラフなどが考えられた。（図1.）が高校別入学者数の年度別推移表である。この表はシート上の全学生データからクロス集

計機能であるピボットテーブル機能を利用すれば簡単に作製できる。縦方向には北海道から沖縄までの全国の高校コード・高校名が並び（高校コードの昇順の並び）、横方向には、年度別の入学者数データが並ぶ。表作成までは容易だが、何百という高等学校から入学者があるので、手作業にて高校別に入学者年度別推移グラフを作製するのは容易なことではない。1つの表から多数のグラフを作製しなければならないケースの1例となっている。

2. グラフを自動作製するマクロ

2-1. グラフ自動作製・自動印刷マクロの作製

自動的にグラフを作製し、印刷する作業を連続して実行するマクロとマクロを実行するためのフォーム（画面）を作製するのは、それほど困難なものではない。

グラフを自動作製し、印刷を実行するマクロは、ExcelのVBA(Visual Basic Application)独特の命令体系で記述する必要があるが、これ自体は、マクロの自動記録を利用すれば良い。マクロ自動記録を開始して、手作業にて目的の作業を実施した後、自動記録を終了すればその

図1.

1	A	B	C	D	E	F	G	H	I	J	K
2	高校コード		高校別入学者数								
3	13001A	東京第1高校	13001A 東京第1高校	18	17	11	12	21	11	18	108
4	13002A	東京第2高校	13002A 東京第2高校	0	4	22	10	0	5	16	57
5	13003A	東京第3高校	13003A 東京第3高校	3	5	4	0	0	0	1	13
6	13004A	東京第4高校	13004A 東京第4高校	1	0	2	3	4	1	1	12
7	13005A	東京第5高校	13005A 東京第5高校	1	2	2	0	3	4	0	12
8	13006A	東京第6高校	13006A 東京第6高校	1	7	2	2	0	0	0	12
9	13007A	東京第7高校	13007A 東京第7高校	1	0	0	0	2	3	5	11
10	13008A	東京第8高校	13008A 東京第8高校	0	0	0	2	4	3	2	11
11	13009A	東京第9高校	13009A 東京第9高校	2	2	1	4	0	1	0	10
12	13010A	東京第10高校	13010A 東京第10高校	2	2	1	1	2	1	1	10
13	13011A	東京第11高校	13011A 東京第11高校	2	1	0	0	1	3	2	9
14	13012A	東京第12高校	13012A 東京第12高校	2	0	1	0	3	0	3	9
15	13013A	東京第13高校	13013A 東京第13高校	5	1	1	0	0	1	0	8
16	13014A	東京第14高校	13014A 東京第14高校	3	3	0	1	0	0	0	7
17	13015A	東京第15高校	13015A 東京第15高校	0	0	1	1	1	1	3	7
18	13016A	東京第16高校	13016A 東京第16高校	0	0	1	0	3	1	2	7
19	13017A	東京第17高校	13017A 東京第17高校	1	0	1	2	1	2	0	7
20	13018A	東京第18高校	13018A 東京第18高校	1	1	1	1	1	1	0	6
21	13019A	東京第19高校	13019A 東京第19高校	0	1	1	0	4	0	0	6
22	13020A	東京第20高校	13020A 東京第20高校	1	0	1	1	0	2	1	6
23	13021A	東京第21高校	13021A 東京第21高校	2	0	1	1	2	0	0	6
24	13022A	東京第22高校	13022A 東京第22高校	0	0	0	5	0	1	0	6

間の作業と全く同じ動作を再現するマクロが自動的に作製されている。(図1.)の場合では、マクロ自動記録を開始して、グラフ作製の対象とするセル全体を一旦、表と無関係な未使用N2~U3セルにコピーし、N2~U3のセル範囲をドラッグして選択し、折れ線グラフを作製し(図2. 参照)、印刷し(プレビュー印刷)、印刷終了後はコピーに使用したN列~U列と作製されたグラフを削除して終了するまでの一連の動作を手作業で実行する。そしてマクロ自動記録を終了すれば、(図3.)のようなマクロが自動的に作製される。

自動作製された(図3.)のマクロプログラム

を観察すると、およそ(表1.)の処理が行われていることが想像される。このマクロを部品として使用すれば、表中の任意の開始行~終了行までのデータを基に連続的にグラフを作製して印刷するような動作をするマクロを作製すること

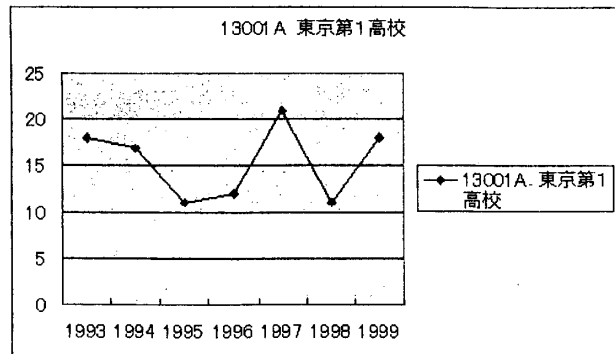


図3.

```

1 Sub Macro10()
2
3 Macro10 Macro
4 マクロ記録日 : 2003/8/20 ユーザー名 : 末木
5
6
7 Range("C2:J2").Select
8 Selection.Copy
9 Range("N2").Select
10 ActiveSheet.Paste
11 Range("C3:J3").Select
12 Application.CutCopyMode = False
13 Selection.Copy
14 Range("N3").Select
15 Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
16 :=False, Transpose:=False
17 Range("N2:U3").Select
18 Application.CutCopyMode = False
19 Charts.Add
20 ActiveChart.ChartType = xlLineMarkers
21 ActiveChart.SetSourceData Source:=Sheets("入学者表").Range("N2:U3"), PlotBy:= _
22 xlRows
23 ActiveChart.Location Where:=xlLocationAsObject, Name:="入学者表"
24 With ActiveChart
25 .HasTitle = True
26 .ChartTitle.Characters.Text = "13001A 東京第1高校"
27 .Axes(xlCategory, xlPrimary).HasTitle = False
28 .Axes(xlValue, xlPrimary).HasTitle = False
29 End With
30 ActiveWindow.SelectedSheets.PrintOut Copies:=1, Preview:=True, Collate:= _
31 True
32 ActiveWindow.Visible = False
33 Selection.Delete
34 Columns("N:U").Select
35 Selection.Delete Shift:=xlToLeft
36 End Sub
37

```

表 1.

マクロ行範囲	対 応 す る 処 理
7行目～16行目	折れ線グラフを作製するための表データを一旦、表の外部のシート上の空いているセルにコピーする。 具体的には、グラフX軸の項目表示（年度）に対応する部分、D2～J2セル範囲を N2～U2セルにコピーする。そして折れ線グラフのデータとなる部分 D3～J3セル範囲を N3～U3セルにコピーする。
17行目～29行目	グラフ化対象 N2～U3セル範囲をドラッグして選択し、折れ線グラフを作製する
30行目～31行目	グラフを印刷する。（プレビューモードで一旦停止）
32行目～35行目	印刷の終わったグラフの削除。グラフ化対象データの存在する N 列～U 列を削除。

1	Private Sub CommandButton1_Click()	図 4 .
2		
3	Dim gyo As Integer	
4	Dim x As Integer	
5		
6	Sheets("入学者表").Select	
7		
8	For gyo = 開始レコード.Value To 終了レコード.Value	
9	Range("C2:J2").Select	
10	Selection.Copy	
11	Range("N2").Select	
12	ActiveSheet.Paste	
13	Range(Cells(gyo, 3), Cells(gyo, 10)).Select	
14	Application.CutCopyMode = False	
15	Selection.Copy	
16		
17	Selection.Copy	
18	ActiveWindow.SmallScroll ToRight:=2	
19	Range("N2").Select	
20	Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _	
21	False, Transpose:=False	
22	Application.CutCopyMode = False	
23		
24	Range("N2:U3").Select	
25	Application.CutCopyMode = False	
26	Charts.Add	
27	ActiveChart.ChartType = xlLineMarkers	
28	ActiveChart.SetSourceData Source:=Sheets("入学者表").Range("N2:U3"), PlotBy:= _	
29	xlRows	
30	ActiveChart.Location Where:=xlLocationAsObject, Name:="入学者表"	
31	With ActiveChart	
32	.HasTitle = True	
33	.ChartTitle.Characters.Text = "13001A 東京第1高校"	
34	.Axes(xlCategory, xlPrimary).HasTitle = False	
35	.Axes(xlValue, xlPrimary).HasTitle = False	
36	End With	
37		
38		

```

39     x = MsgBox("実行しますか?", vbYesNo)
40     If x <> vbYes Then
41         ActiveWindow.Visible = False
42         Selection.Delete
43         ActiveWindow.SmallScroll ToRight:=7
44         Columns("M:U").Select
45         Selection.Delete Shift:=xlToLeft
46
47         Exit Sub
48     End If
49
50     If 印刷チェック = True Then
51         If gyo = 開始レコード.Value Then
52             ActiveWindow.SelectedSheets.PrintOut Copies:=1, Preview:=True, Collate:=True
53         Else
54             ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
55         End If
56     End If
57
58     ActiveWindow.Visible = False
59     Selection.Delete
60     ActiveWindow.SmallScroll ToRight:=7
61     Columns("M:U").Select
62     Selection.Delete Shift:=xlToLeft
63
64     Next gyo
65
66 End Sub

```

とができる。これは、プログラミングの基本的な繰り返し処理、条件分岐処理構造を応用するだけのことなのでそれほど難しくはない。繰り返し処理構造を構築し、変数化する部分を注意深く変更することにより(図4.)のようなマクロに作り変えることができる。(図4.)のマクロは、さらにフォーム上のボタンをクリックすることにより動作するタイプのマクロとしての体裁を整えてある。

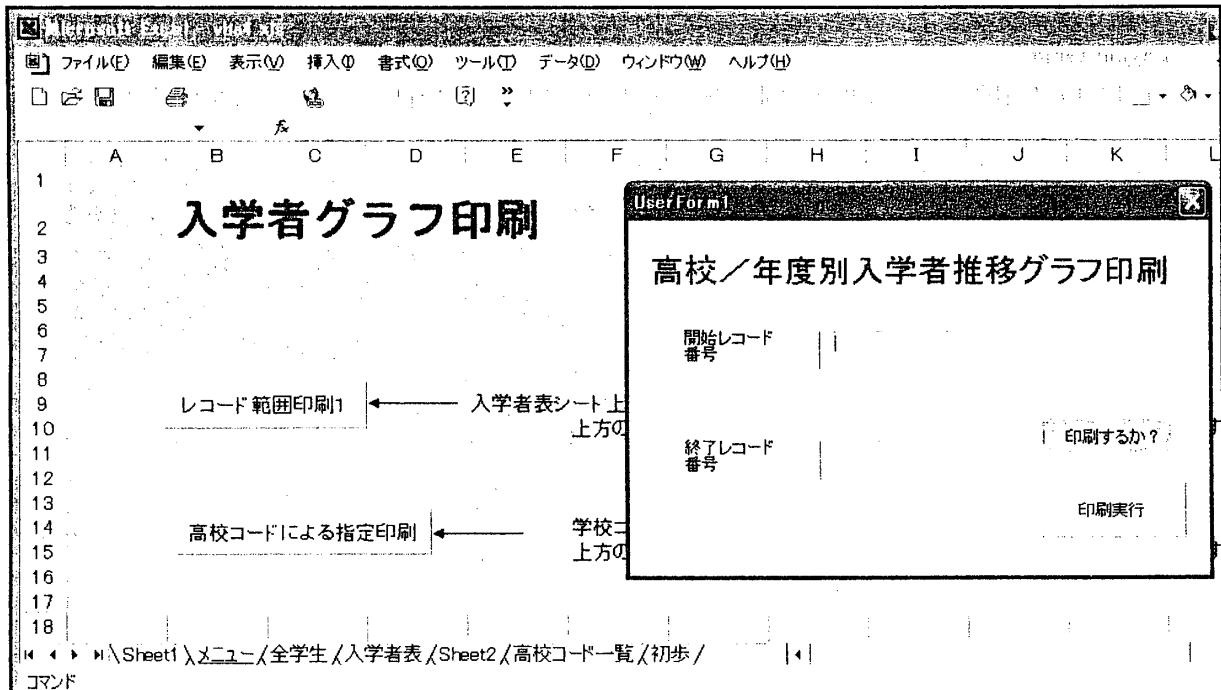
(図5.)のように、1つのシートをメニューにして、**レコード範囲印刷1** というボタンをクリックすることにより、フォームが起動するようにしている。フォームでは、印刷対象開始レコードと、終了レコード番号を入力するためのテキストボックスを用意し、任意の開始レコード～終了レコード範囲の全ての行を対象として自動的にグラフ作製し印刷実行させることができるようにした。フォーム上の **印刷実行** ボ

タンをクリックされると、(図4.) マクロが動作する。

グラフは、表内のデータ並び順通りに高校コードの昇順に個別に印刷される。印刷されるグラフは前後のグラフ同士は特に関連性が無いケースと言えるだろう。このようなケースでは、単にグラフが個別に印刷されるだけで一応の目的が果たせるものと思われる。高校コードの昇順に印刷されたグラフを保管しておけば必要に応じて、目的の高校を探し出してグラフを確認することができるだろう。

2-2. グラフ画像入りホームページを自動作製するマクロ

次に、作製されるグラフの前後のグラフに関連性があり、前後のグラフ同士を比較対照するようなケースを考えてみた。例題として、『カオスとフラクタル ◎Excelで体験』 1999年



株式会社オーム社発行、P.104に紹介されている「グモウスキーとミラの写像」のグラフをExcelにて描く例を取り上げてみた。同書によると、「グモウスキーとミラの写像」は(図6.)の式で与えられる。

(図7.)のExcelシートは、この(図6.)の数式に従って初期値としてセルA3にパラメータ μ 値、セルA4に定数a値(0.008)、セルA5に定数b値(0.05)、セルB3にx(0)の値(0.1)、

セルC3にy(0)の値(0)を書き込み、あとは(図6.)の計算式に従って、x座標の数値をB列に、y座標の数値をC列に、G値をD列

図 6.

$$x(t+1) = y(t) + a[1 - by^2(t)]y(t) + G[x(t)]$$

$$y(t+1) = -x(t) + G[x(t+1)]$$

$G[x(t)]$ は、 μ をパラメータとして、

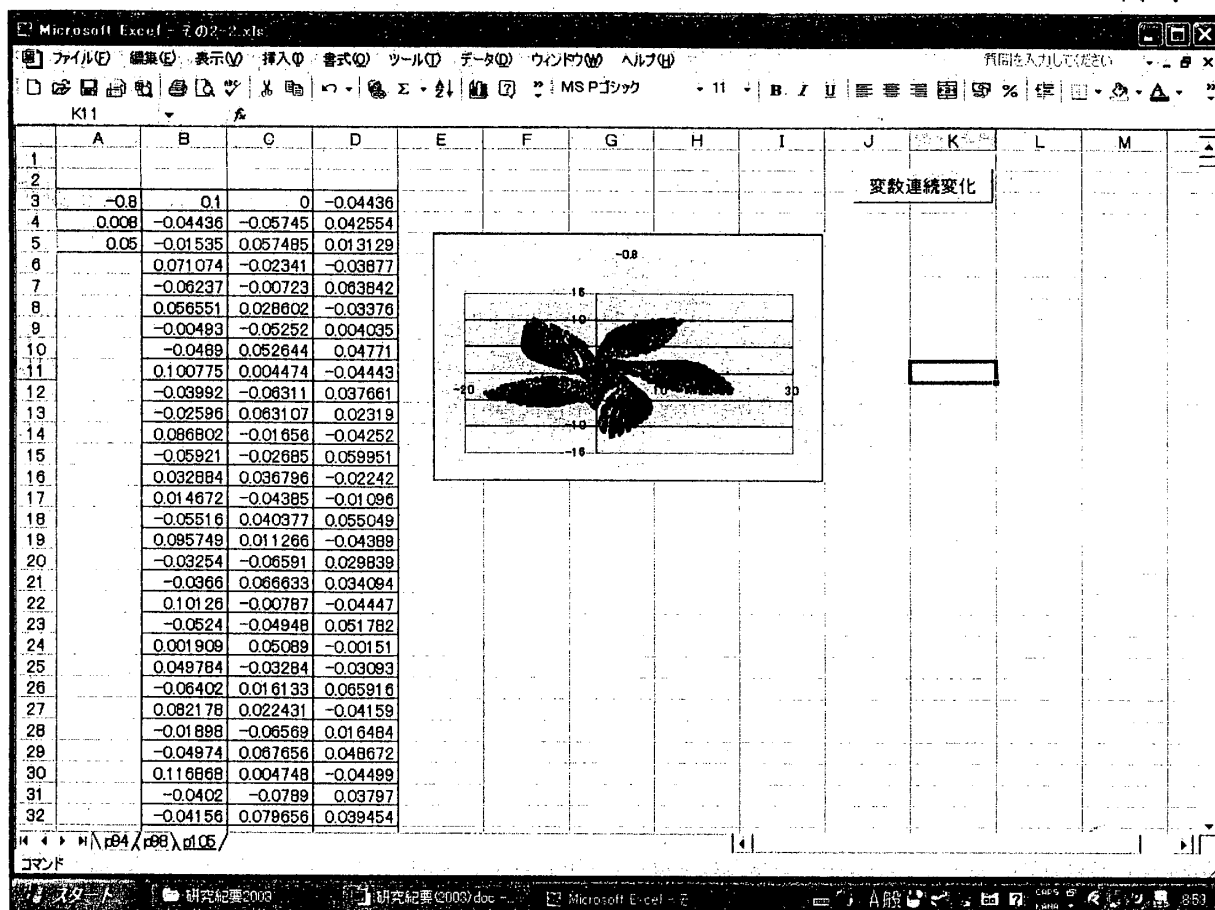
$$G[x(t)] = \mu x(t) + \frac{2(1-\mu)x^2(t)}{1+x^2(t)}$$

に計算させる。そしてB列とC列のデータ全体を選択してx座標、y座標の散布図グラフを製作している。

(図7.)に描かれているグラフは、セルA3のパラメータ μ 値が、 $\mu = -0.8$ の場合のグラフの様子であるが、セルA3のパラメータ μ 値をほんのわずか変えただけでも、グラフの様子は大幅に変化する。

このような場合、パラメータ μ 値を微妙に連続的に変化させた時のグラフの変化を見たくなる。いちいちセルA3のパラメータ μ 値を書き換えてグラフを確認するのは面倒であるし、また数値変更前後のグラフの変化を比較して観察することも難しい。この場合、2-1.節のケースのように、マクロによってパラメータを連続変化させて、グラフを自動印刷させて、印刷されたグラフ図紙面を並べて観察するやり方もあるだろうが、今回のようなケースの場合には、印刷されたグラフ図面を手作業でならべるより、パソコン画面上に連続的にグラフが表示され、画面をスクロールアップ・ダウンすることによ

図 7.



り複数のグラフ図の微妙な変化を観察できるほうが良いのではないかとと思われる。画面をスクロールアップ・ダウンすることにより複数の図面を見せるとなるとホームページを利用することが思い浮かぶ。複数のグラフが連続的に表示されるようなホームページを簡単に・自動的に作製することができたら便利であろう。

エクセルには、シート上の表・グラフからホームページを自動的に作製する機能があるので、マクロ自動記録機能を使えば、グラフを作製し、そのグラフ画像が入ったホームページを自動作製するマクロは簡単に作製することが可能なのではないかと想像された。しかし、複数のグラフ画像が1つのホームページにきれいに収まるような動作をさせるマクロができるものか否かは不明であった。1つのシート上に多数のグラフ画像を自動的に作製させて、その後そのシー

トを強引にホームページ化するやり方もあるのだろうが、いかにも荒っぽい方法である。

試みに、Excel98にて1つのグラフからホームページを作製するマクロを自動作成させてみたのだが上手く利用できないようであった。このマクロを繰り返し処理構造に変更して連続的に画像を生成しホームページ化するマクロを作成してみたところ、結局1つのグラフ画像(GIF形式画像)とその1つの画像を収納するホームページしか作製されなかった。過去に作製されたグラフ画像は削除されて新しいグラフ画像に上書きされて、最新のグラフ画像が1つ作製され残るだけであった。グラフ画像を連続的に生成し、同時にホームページに次々と生み出されるグラフ画像を追加挿入するような機能は無いようであった。

しかし、Excelのより新しいバージョン

1	Sub Macro20	図 8.
2	'	
3	Macro2 Macro	
4	マクロ記録日 : 2001/11/14 ユーザー名 : 末木俊之	
5	'	
6		
7	'	
8		
9	ActiveSheet.ChartObjects("グラフ 1").Activate	
10	With ActiveChart	
11	.HasTitle = True	
12	.ChartTitle.Characters.Text = "-0.8"	
13	End With	
14		
15	With ActiveWorkbook.PublishObjects.Add(xlSourceChart, _	
16	"D:\Chaos & Fractal\その 2.htm", "p105", "グラフ 1", xlHtmlStatic, "L" & CStr(counter), "")	
17	.Publish (True)	
18	.AutoRepublish = False	
19	End With	
20	ChDir "D:\Chaos & Fractal"	
21	End Sub	

(Excel 2000以降)では、画像のホームページへの追加挿入機能が設けられたようである。

Excel 2000にて1つのグラフからホームページを作製するマクロを自動作成させてみると、(図 8.) のようなマクロが生成された。

(図 8.) の行15~19行目の ActiveWorkbook.PublishObjects.Add 命令がグラフ画像入りホームページを作製している箇所であると想像されるが、いくつかのパラメータがあることが注目される。おそらくそれらのパラメータのうちのいずれかを変更することにより、このマクロを実行する度に新たなグラフ画像がホームページに追加される形式でホームページが更新されるようになるのではないかと想像された。具体的には17行目の.Publish (True) のパラメータを(False)に変更することにより、グラフ画像が追加モードでホームページに格納されるように動作を変えることができた。

一部パラメータに手を加えた(図 8.)のマクロを基に、最終的に(図 9.)のマクロを作製した。さらに(図10.)のように、シート上に「変数連続変化」というキャプションの付いた

ボタンを用意し、これをクリックすることによりフォームがシート上に出現するようにした。フォーム上には、初期値、終了値、増分の3つの数値を入力するテキストボックスを配置し、「グモウスキーとミラの写像」のパラメータ μ 値(セル A3)を任意の初期値から、終了値まで、任意の増分で変化させたグラフ画像の入ったホームページを作製できるようにした。(図 10.)の例だと、パラメータ μ 値が、 $-0.8 \sim -0.7$ の間で0.01刻みに変化させた約100枚ほどのグラフ画像が入ったホームページが自動作製されることになる。「実行」ボタンがクリックされると、(図11.)のマクロが動作する。このマクロは、繰り返し処理によりパラメータ μ 値(セル A3)を連続的に変更し、さらに(図 9.)マクロをサブルーチンとして呼び出してホームページを作製させている。またフォーム上には作製するホームページ名を指定できるように、HTMLファイル名テキストボックスも用意した。


```

1 Sub saveChaosGraph(counter As Integer, title As String, fileName As String)
2
3 Macro2 Macro
4 マクロ記録日 : 2001/11/14 ユーザー名 : 末木俊之
5
6
7
8
9 ActiveSheet.ChartObjects("グラフ 1").Activate
10 With ActiveChart
11 .HasTitle = True
12 .ChartTitle.Characters.Text = "-0.8"
13 End With
14
15 With ActiveWorkbook.PublishObjects.Add(xlSourceChart, _
16 "D:¥Chaos & Fractal¥" & fileName & ".htm", "p105", "グラフ 1", xlHtmlStatic, "L" &
17 CStr(counter), "")
18 .Publish (False)
19 .AutoRepublish = False
20 End With
21 ChDir "D:¥Chaos & Fractal"
End Sub

```

図10.

The screenshot displays a Microsoft Excel spreadsheet with a table of numerical data and a fractal plot. A 'User Form' dialog box is overlaid on the spreadsheet, allowing for the configuration of the fractal plot's parameters. The data table contains 32 rows of values, and the fractal plot shows a complex, self-similar pattern. The 'User Form' dialog box includes the following fields:

- 初期値 (Initial Value): -0.8
- 終了値 (Final Value): -0.7
- 増分 (Increment): 0.01
- HTMLファイル名 (HTML File Name): F1

<pre> 1 Private Sub CommandButton1_Click() 2 Dim startValue As Double 3 Dim lastValue As Double 4 Dim plusValue As Double 5 Dim ok As Integer 6 Dim Msg As Integer 7 Dim counter As Integer 8 9 startValue = 初期値 10 lastValue = 終了値 11 plusValue = 増分 12 ok = 0 13 counter = 1 14 15 While ok = 0 16 If startValue > lastValue Then 17 ok = 1 18 Else 19 Worksheets("p105").Cells(3, 1) = startValue 20 Call saveChaosGraph(counter, CStr(startValue), HTMLファイル名) 21 Msg = MsgBox("次実行しますか?", vbYesNo + vbQuestion) 22 If Msg = vbYes Then 23 counter = counter + 1 24 startValue = startValue + plusValue 25 Else 26 ok = 1 27 End If 28 End If 29 Wend 30 End Sub </pre>	<p>図11.</p>
--	-------------

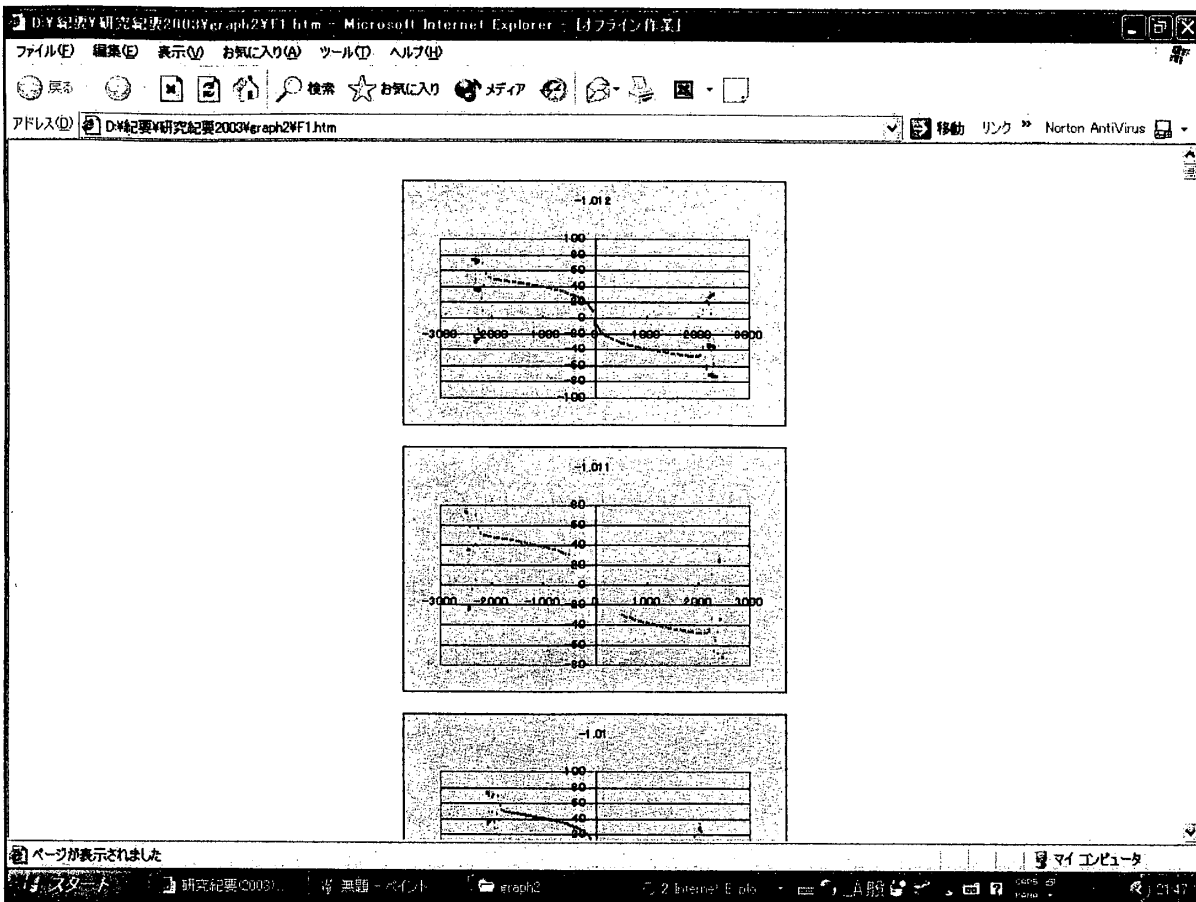
2-3. 自動作製されたグラフ画像入りホームページ

2-2. 節に記述したマクロにより自動作製されたホームページの例を(図12.)に載せた。このホームページはパラメータ μ 値を、-1.012~-0.900の間で0.01刻みに変化させて作製したものである。1つのページに100枚以上のグラフ画像が表示されている。ホームページは、“F1.htm”という名称で作製させた。自動的に“F1.files”という名称のフォルダが作製され、グラフ画像はその中に格納されている。また“F1.files”フォルダ内には、作製された画像一覧データを保持するXML形式のテキストファイル“filelist.xml”が作製されている。(図13.)が“filelist.xml”ファイルの最初の10行程度の様子である。グラフ画像は、最初に作製されたグラフから、“L1_image001.gif”、“L2_image001.

gif”、“L3_image001.gif”という具合に、連続的に・規則的に変化する名称で作製されていることが分かる。

(図14.)がマクロにて自動作製されるホームページ、グラフ画像、XMLファイル関連図である。グラフ画像を格納している“F1.files”フォルダのサイズは、865KBにもなっていた。グラフ画像(GIF形式画像)1つのサイズは3~4KB程度であるが、それが100個以上作製されてフォルダ内に格納されている。“F1.htm”ホームページには、それら全てのグラフ画像が表示され、画面をスクロールアップ・ダウンすることによりパラメータ μ 値の変化に対応したグラフ画像の変化を観察することができる。昨今のパソコンは処理速度も速くメモリ容量も多くなっているので、パーソナルパソコン内に格納されたこのホームページを表示するには特に

図12.



```

<?xml xmlns:o="urn:schemas-microsoft-com:office:office">
  <o:File HRef="L1_image001.gif" PublicationID="L1" />
  <o:File HRef="L2_image001.gif" PublicationID="L2" />
  <o:File HRef="L3_image001.gif" PublicationID="L3" />
  <o:File HRef="L4_image001.gif" PublicationID="L4" />
  <o:File HRef="L5_image001.gif" PublicationID="L5" />
  <o:File HRef="L6_image001.gif" PublicationID="L6" />
  <o:File HRef="L7_image001.gif" PublicationID="L7" />
  <o:File HRef="L8_image001.gif" PublicationID="L8" />
  <o:File HRef="L9_image001.gif" PublicationID="L9" />
  <o:File HRef="L10_image001.gif" PublicationID="L10" />

```

図13.

ストレスは感じなかった。(インテル Celeron プロセッサ・1.80Gz、メモリ 1 GB の WindowsXP パソコンを使用している。)

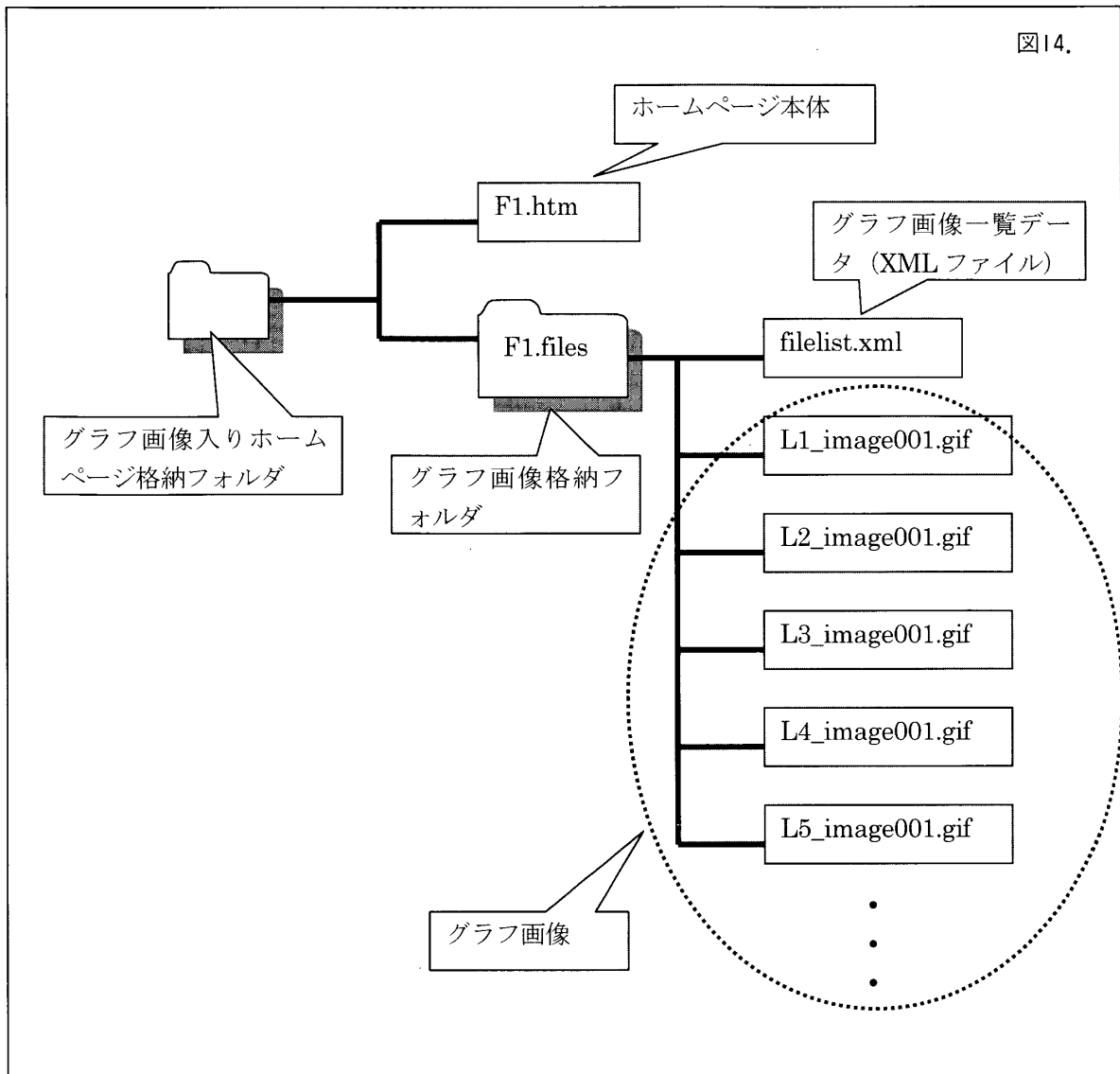
3. グラフ画像自動作製マクロの利用

2-2. 節にて、複数のグラフ画像を格納したホームページを簡単に作製できるようになった。

しかし2-2. 節で取り上げた「グモウスキーとミラの写像」グラフは、定数 a 値=0.008、定数 b 値=0.05の初期条件で、パラメータ μ 値を、-1.012~-0.900の間で0.01刻みに変化させて作製させたものに過ぎない。

定数 a、b 値を変えればまた違ったグラフになる。またパラメータ μ 値もまだかなりの変動

図14.



余地がある。2-2. 節の初期条件では、およそパラメータ値が、 $-1.012 \sim 0.999$ の間で変化させて意味あるグラフを作製できるようである。(図10.) のフォームにてパラメータ μ 値の初期値 -1.012 、終了値 0.999 、増分 0.01 と入力して、グラフ画像約2000個を表示する1つのホームページを強引に作製することも可能である。昨今の処理速度が速く、メモリ容量の大きなパソコンではさほどストレスは感じないようであるが、目的とするパラメータ値前後のグラフの様子を手早く観察するには、細かいパラメータ範囲ごとに分割したホームページを作製し、目次フレームから個々のページを呼び出せるようなフレ

ーム分割ページを用意する方がより便利であると思われる。

3-1. 自動作製ホームページの単純な利用

(図10.) のフォームを1回実行することにより、(図14.) のような1つのホームページとグラフ画像入りのフォルダが作製される。パラメータ μ 値の初期値・終了値範囲を変えてマクロを実行することにより、簡単に複数個のホームページを作製することができる。(表2.) に(図10.) のフォームから20回のマクロ実行により作製した、20個のホームページの一覧を載せた。

この20個のホームページへのリンクを目次フ

表 2.

ホームページ名	パラメータ μ 値初期値	パラメータ μ 値終了値
F1.htm	-1.012	-0.900
F2.htm	-0.900	-0.800
F3.htm	-0.800	-0.700
F4.htm	-0.700	-0.600
F5.htm	-0.600	-0.500
F6.htm	-0.500	-0.400
F7.htm	-0.400	-0.300
F8.htm	-0.300	-0.200
F9.htm	-0.200	-0.100
F10.htm	-0.100	0.000
F11.htm	0.000	0.100
F12.htm	0.100	0.200
F13.htm	0.200	0.300
F14.htm	0.300	0.400
F15.htm	0.400	0.500
F16.htm	0.500	0.600
F17.htm	0.600	0.700
F18.htm	0.700	0.800
F19.htm	0.800	0.900
F20.htm	0.900	0.999

フレームに格納した、二分割フレームホームページを作製したものが(図15.)である。目次フレームは単純にテキストエディタで手作業にて作製した。左フレームのアンダーラインが引かれたパラメータ範囲値をクリックすることにより、(表2.)に記述した対応するホームページが右フレームに表示される。このホームページは、自動作製された複数のページを単純に使用して目次フレームによって全体として1つの本の形式にまとめ上げたものと言える。

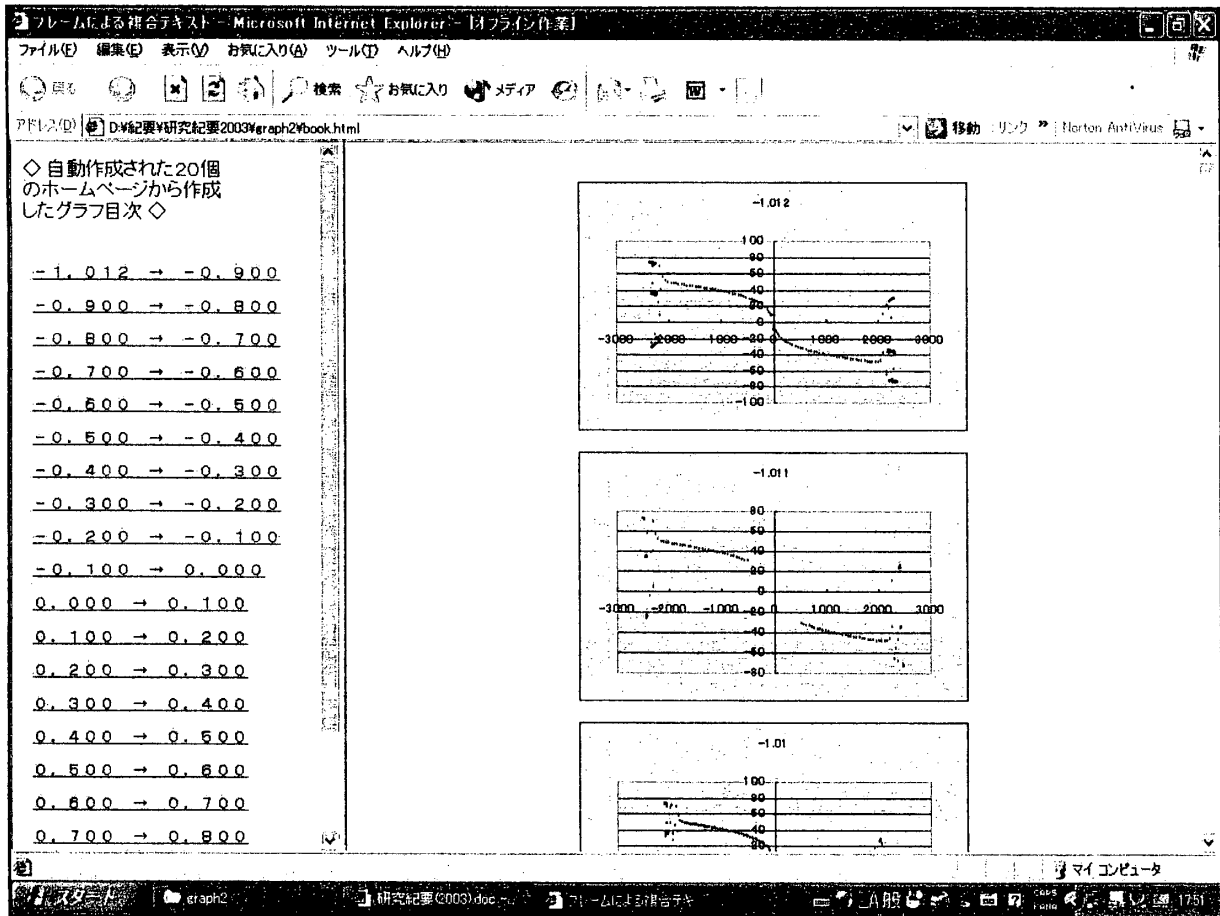
3-2. 画像データベースXMLファイルからのホームページ作製

3-2-1. 画像データベース作製

3-1. 節では、20回のマクロ実行により自動的に作製される20個のホームページを(図15.)の“book.html”という名前の目次フレーム付ページにまとめてみた。

マクロ実行により自動的に作製されるホームページのソーステキストはあまり美しくない。また後にホームページの表示スタイルを好みのスタイルに変更するのも困難である。自動作製されたホームページは、それ以上の応用は難し

図15.



いようであった。

次に、XML 技術を応用して、(図15.)と同様なホームページを作製する別のやり方を試みてみた。

まずグラフ画像を作製するのを1回のマクロ実行にて済ますことにする。これにより全てのグラフ画像 (GIF 形式画像) に統一的な名前を付けて作製することができるようになる。(図10.) のフォームにて初期値-1.012、終了値0.999、増分0.01と入力してマクロ実行することにより、1つのホームページとグラフ画像約2000枚、そしてすべての画像を収納する1つのフォルダが作成される。自動作製されたホームページは捨てる。フォルダ内に作製された画像のみ使用することにした。自動作製されたグラフ画像の名称は Excel シートで管理することができる。マクロにて、“F0.htm” という名称でホ

ームページを作製すると、“F0.files” という名称のフォルダが自動作製され画像が格納される。この場合には Excel シート上に (図16.) のような画像情報データベースを作製すれば良い。これは手作業で行うが、自動作製されるグラフ画像 (GIF 形式画像) 名称は規則的に変化しているのでさほど難しいことではない。Excel シート上の C 列が、画像名称の規則的に変化する部分である。D 列は、画像名称の共通な固定部分である。E 列は、画像が格納されているフォルダ名称である。これも全ての画像に共通な部分である。固定部分は全ての行に渡って同一セル内容を単純にコピーすれば済む。規則的に変化する C 列の部分も Excel のオートフィル機能を使えば手作業で簡単に作成できる。A 列はグラフのパラメータ μ 値であるが、0.01ずつ規則的に変化しているのでこれも同様にオートフィル

図16.

	A	B	C	D	E	F	G	H	I	J
1	パラメタ	グラフ名	グラフ名変化部	グラフ名共通部	パス名					
2	-1.012	F0.files/L1_image001.gif	L1	image001.gif	F0.files/					
3	-1.011	F0.files/L2_image001.gif	L2	image001.gif	F0.files/					
4	-1.01	F0.files/L3_image001.gif	L3	image001.gif	F0.files/					
5	-1.008	F0.files/L4_image001.gif	L4	image001.gif	F0.files/					
6	-1.008	F0.files/L5_image001.gif	L5	image001.gif	F0.files/					
7	-1.007	F0.files/L6_image001.gif	L6	image001.gif	F0.files/					
8	-1.006	F0.files/L7_image001.gif	L7	image001.gif	F0.files/					
9	-1.005	F0.files/L8_image001.gif	L8	image001.gif	F0.files/					
10	-1.004	F0.files/L9_image001.gif	L9	image001.gif	F0.files/					
11	-1.003	F0.files/L10_image001.gif	L10	image001.gif	F0.files/					
12	-1.002	F0.files/L11_image001.gif	L11	image001.gif	F0.files/					
13	-1.001	F0.files/L12_image001.gif	L12	image001.gif	F0.files/					
14	-1	F0.files/L13_image001.gif	L13	image001.gif	F0.files/					
15	-0.999	F0.files/L14_image001.gif	L14	image001.gif	F0.files/					
16	-0.998	F0.files/L15_image001.gif	L15	image001.gif	F0.files/					
17	-0.997	F0.files/L16_image001.gif	L16	image001.gif	F0.files/					
18	-0.996	F0.files/L17_image001.gif	L17	image001.gif	F0.files/					
19	-0.995	F0.files/L18_image001.gif	L18	image001.gif	F0.files/					
20	-0.994	F0.files/L19_image001.gif	L19	image001.gif	F0.files/					
21	-0.993	F0.files/L20_image001.gif	L20	image001.gif	F0.files/					
22	-0.992	F0.files/L21_image001.gif	L21	image001.gif	F0.files/					
23	-0.991	F0.files/L22_image001.gif	L22	image001.gif	F0.files/					
24	-0.99	F0.files/L23_image001.gif	L23	image001.gif	F0.files/					
25	-0.988	F0.files/L24_image001.gif	L24	image001.gif	F0.files/					
26	-0.988	F0.files/L25_image001.gif	L25	image001.gif	F0.files/					
27	-0.987	F0.files/L26_image001.gif	L26	image001.gif	F0.files/					
28	-0.986	F0.files/L27_image001.gif	L27	image001.gif	F0.files/					
29	-0.985	F0.files/L28_image001.gif	L28	image001.gif	F0.files/					
30	-0.984	F0.files/L29_image001.gif	L29	image001.gif	F0.files/					
31	-0.983	F0.files/L30_image001.gif	L30	image001.gif	F0.files/					
32	-0.982	F0.files/L31_image001.gif	L31	image001.gif	F0.files/					

で簡単に作成できる。そして B 列が最終目的のグラフの画像名称が格納される列であるが、これは E 列、C 列、D 列データを単純に文字列連結することによって得られる。これは、=CONCATENATE 関数を使えば良い。

このように、コピー、オートフィル、関数の機能を使用して、(図16.) の約2000個のグラフデータを表計算シート上に手作業で作製する。

“グラフ F0.xls”という名称の Excel ファイルとして作製した。

3-2-2. 画像データベースの XML 文書化作業

次には、Excel シート上の画像データから XML テキストファイルを作製した。今回は単純にワープロソフト (Word) を使って、Excel シート上のデータを差し込み印刷するやり方で作製してみた。

具体的には、“グラフ F0.xls”シート上のデータを、一旦“グラフ F0.txt”というカンマ区切りの CSV 形式のテキストファイルとして保存し (図17. 参照)、“グラフ F0.doc”という名称のワープロ文書 (図18. 参照) を作製し、画像データを差し込んだ。(図17.) は、“グラフ F0.txt”ファイルの先頭から11行目までの様子である。(図18.) における《パラメータ》、《グラフ名》の箇所に Excel シート上の同名のデータが差し込まれることになる。エクセルシート上のデータを直接ワープロに差し込む方法でも問題ないはずであるが、今回はカンマ区切りの CSV 形式のテキストファイルとして一旦エクセルシートデータを保存し、それをワープロに差し込んだ。差し込み印刷操作を実行するとグラフ画像データ数分のページを持つワープロ文章が作製される。このワープロ文章を、テキストファ

パラメタ, グラフ名, グラフ名変化部, グラフ名共通部, パス名 図17.

```
-1.012, F0. files/L1_image001. gif, L1, _image001. gif, F0. files/  
-1.011, F0. files/L2_image001. gif, L2, _image001. gif, F0. files/  
-1.01, F0. files/L3_image001. gif, L3, _image001. gif, F0. files/  
-1.009, F0. files/L4_image001. gif, L4, _image001. gif, F0. files/  
-1.008, F0. files/L5_image001. gif, L5, _image001. gif, F0. files/  
-1.007, F0. files/L6_image001. gif, L6, _image001. gif, F0. files/  
-1.006, F0. files/L7_image001. gif, L7, _image001. gif, F0. files/  
-1.005, F0. files/L8_image001. gif, L8, _image001. gif, F0. files/  
-1.004, F0. files/L9_image001. gif, L9, _image001. gif, F0. files/  
-1.003, F0. files/L10_image001. gif, L10, _image001. gif, F0. files/
```

<record> 図18.

```
<パラメタ><パラメタ></パラメタ>  
<グラフ名><グラフ名></グラフ名>  
</record>
```

<record> 図19.

```
<パラメタ>-1.012</パラメタ>  
<グラフ名>F0. files/L1_image001. gif</グラフ名>  
</record>  
<record>  
<パラメタ>-1.011</パラメタ>  
<グラフ名>F0. files/L2_image001. gif</グラフ名>  
</record>  
<record>  
<パラメタ>-1.01</パラメタ>  
<グラフ名>F0. files/L3_image001. gif</グラフ名>  
</record>
```

イル形式で“F0-0.xml”という名称で保存した。(図19.)のテキストファイルがそれである。

(図19.)は、作製されたテキストファイルの先頭12行の様子である。

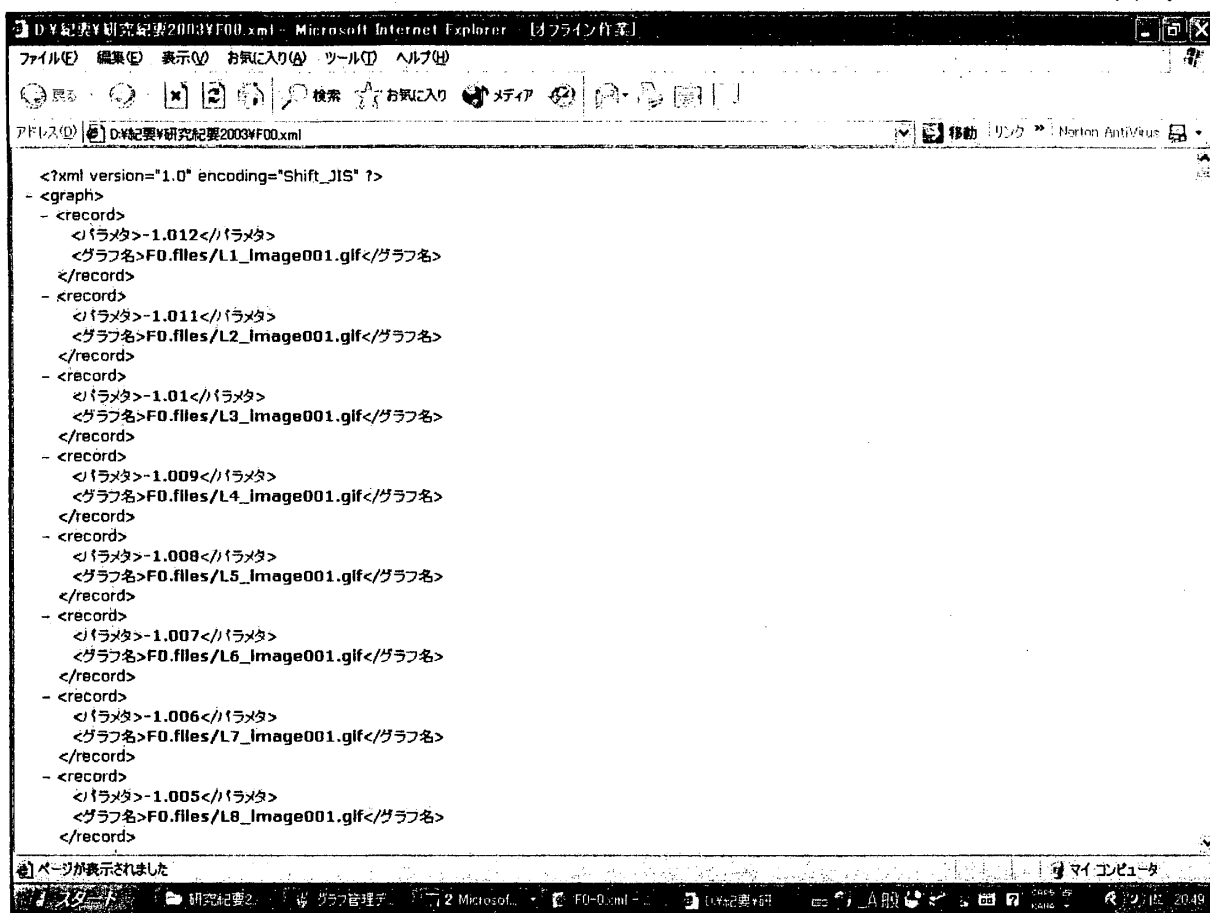
最後に、“F0-0.xml”テキストファイルを基にテキストエディタにてXMLファイルとして

の体裁を整えるための2, 3行を追加して、“F00.xml”テキストファイルを作成した。こ

れをインターネットエクスプローラで開くと

(図20.)の様に表示される。XMLファイルとして認識されて問題なく表示されていることが

わかる。



3-2-3. XSLT スタイルシートによる XML 文書のホームページ化

約2000個のグラフ画像データを格納する XML 文書ができあがった。これを何らかの方法で HTML 文書に変換するか、または CSS スタイルシートなどによって表示スタイルを付与すればホームページが出来上がることになる。

しかし単純にホームページ化するだけであると、2000個もの画像全てを表示するサイズの大きなホームページが作製されてしまう。3-1.節の(図15.)のページのように、表示する画像を絞り込んで表示する複数のホームページを XML ファイルから作製し、目次付き二分割フレームホームページにまとめ上げることを試みてみた。

XSLT (Extensible Stylesheet Language Transformations) スタイルシートを使えば、

XML ファイル内のデータを絞り込み、ホームページとして表示させることができる。

まずパラメータ μ 値が $-1.012 \sim -0.900$ の範囲に該当するグラフ画像のみ抽出してホームページとして表示させるためのスタイルシート “F0-1.xsl” (図21.) を作製した。

(図21.) の11行目~15行目の部分が、`` という HTML 言語における画像をページに表示するためのタグ部分を生成している箇所である。10行目は、表示すべきグラフ画像を絞り込んでいる部分である。XML 文書内のパラメータデータが、 -1.012 より大きく、かつ、 -0.900 より小さい画像のみ抽出する条件判定文となっている。

次に、“F00.xml” XML ファイルをコピーし、“F0-1.xml” を作製した。

```

1 <?xml version="1.0" encoding="Shift_JIS" ?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
3
4   <xsl:template match="/">
5
6     <HTML>
7     <BODY>
8       <xsl:for-each select="graph/record">
9         <xsl:choose>
10          <xsl:when test="パラメタ &gt;=-1.012 and パラメタ &lt;=-0.900">
11            <xsl:element name="IMG">
12              <xsl:attribute name="SRC">
13                <xsl:value-of select="グラフ名" />
14              </xsl:attribute>
15            </xsl:element>
16            <BR />
17          </xsl:when>
18        </xsl:choose>
19      </xsl:for-each>
20    </BODY>
21  </HTML>
22 </xsl:template>
23 </xsl:stylesheet>

```

図21.

```

1 <?xml version="1.0" encoding="Shift_JIS"?>
2 <?xml-stylesheet type="text/xsl" href="F0-1.xsl"?>
3
4 <graph>
5 <record>
6   <パラメタ>-1.012</パラメタ>
7   <グラフ名>F0.files/L1_image001.gif</グラフ名>
8 </record>
9 <record>
10  <パラメタ>-1.011</パラメタ>
11  <グラフ名>F0.files/L2_image001.gif</グラフ名>
12 </record>
13 <record>
14  <パラメタ>-1.01</パラメタ>
15  <グラフ名>F0.files/L3_image001.gif</グラフ名>
16 </record>

```

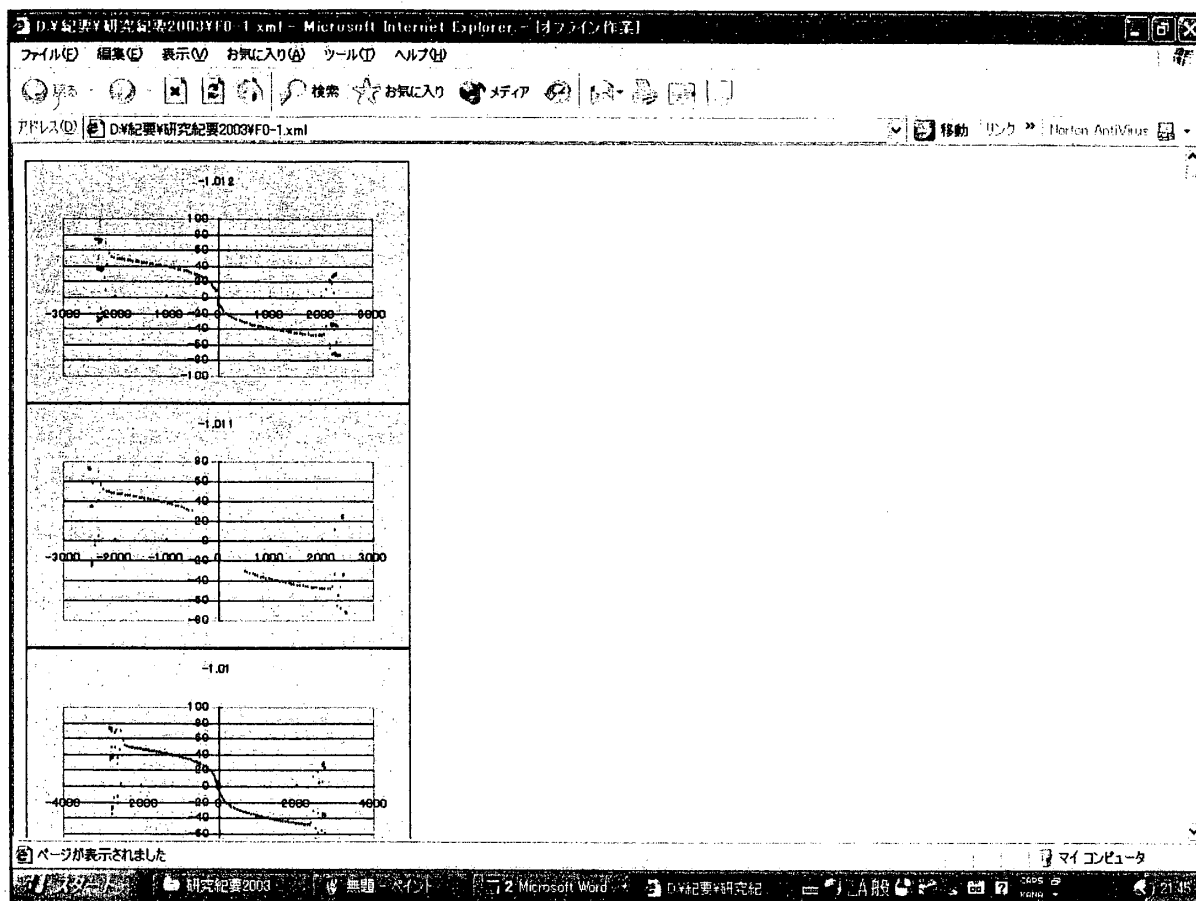
図22.

(図22.) は、“F0-1.xml” ファイルの先頭から16行目の様子である。2行目にスタイルシート“F0-1.xsl”を参照する指定がある。これにより、Internet Explorer ブラウザソフトで“F0-1.xml”ファイルを表示させると、スタイルシート“F0-1.xsl”ファイルの記述に従ってXMLデータがホームページの形式に変換されて表示される。(図23.) は、“F0-1.xml”ファイルが表示された時の様子である。XMLファイルが

スタイルシートの作用により、パラメータ μ 値が-1.012~-0.900の範囲に該当するグラフ画像のみに絞り込まれてホームページ形式に変換されて表示されていることが分かる。

後は、同様にして全部で20個のスタイルシートとXMLファイルを作製すれば良い。“F0-1.xml”ファイルと“F0-1.xsl”ファイルをコピーし一部分変更を加えれば完成する。具体的には、スタイルシートは、(図21.)における10行目の

図23.



表示すべきグラフ画像を絞り込む条件判定部分を変更するだけで済む。また XML ファイルは、(図22.)における2行目の参照するスタイルシート名を変更するだけで済む。

作製した XML ファイルとスタイルシートの一覧を(表3.)に示す。

3-2-4. 目次フレーム付き二分分割フレームページ作製

最後に(表3.)に一覧表示した20個のXMLファイルへのリンクを目次フレームに入れた、二分分割フレームホームページの形式でまとめたものが(図24.)である。左フレームのアンダーラインが引かれたパラメータ範囲値をクリックすることにより、(表3.)に記述した対応するXMLファイルが右フレームに表示される。(図24.)はパラメータ μ 値0.300~0.400の範囲に

対応する“F0-13.xml”ファイルが右フレームに表示されている様子である。3-1. 節にて作製した(図15.)ホームページと見た目はほぼ同一のフレーム分割ホームページが作製された。

4. 結び

表計算ソフトは、一般的にデータ分析結果を表としてまとめ、さらにそこからグラフ作製するという流れで使用されるのだろうが、グラフを扱う段階で手作業では効率が悪いケースがあるようだ。このケースに対応する作業を自動化するマクロについて検討してみたのだが、マクロ作製自体はそんなに難しいものではなさそうである。プログラミング技法の条件分岐、繰り返し処理のちょっとした実用的な応用例の1つとして有用なものができそうである。多量のグラフを自動的に作製し、印刷する。または多量

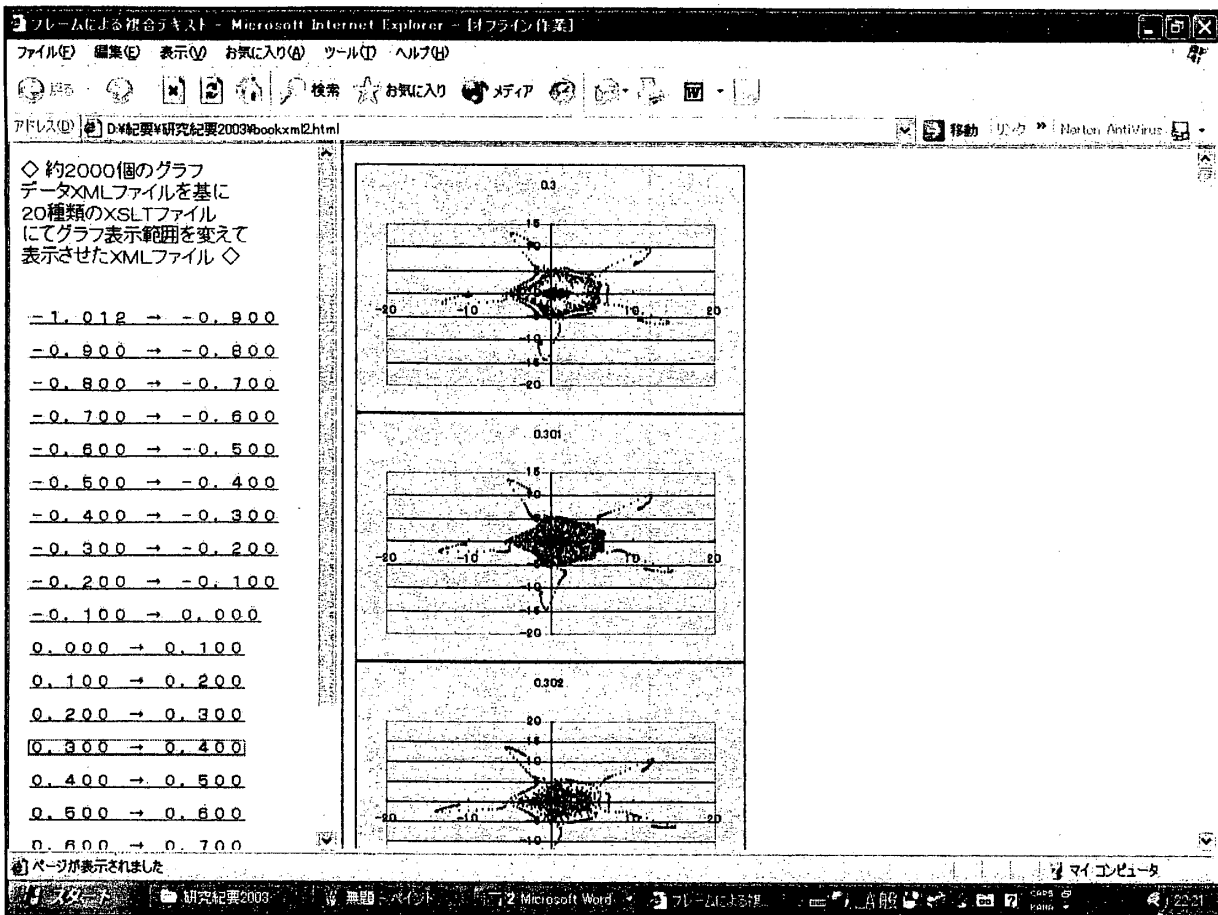
表 3.

XML ファイル名	参照するスタイルシート名	パラメータ μ 値初期値	パラメータ μ 値終了値
F0-1.xml	F0-1.xsl	-1.012	-0.900
F0-2.xml	F0-2.xsl	-0.900	-0.800
F0-3.xml	F0-3.xsl	-0.800	-0.700
F0-4.xml	F0-4.xsl	-0.700	-0.600
F0-5.xml	F0-5.xsl	-0.600	-0.500
F0-6.xml	F0-6.xsl	-0.500	-0.400
F0-7.xml	F0-7.xsl	-0.400	-0.300
F0-8.xml	F0-8.xsl	-0.300	-0.200
F0-9.xml	F0-9.xsl	-0.200	-0.100
F0-10.xml	F0-10.xsl	-0.100	0.000
F0-11.xml	F0-11.xsl	0.000	0.100
F0-12.xml	F0-12.xsl	0.100	0.200
F0-13.xml	F0-13.xsl	0.200	0.300
F0-14.xml	F0-14.xsl	0.300	0.400
F0-15.xml	F0-15.xsl	0.400	0.500
F0-16.xml	F0-16.xsl	0.500	0.600
F0-17.xml	F0-17.xsl	0.600	0.700
F0-18.xml	F0-18.xsl	0.700	0.800
F0-19.xml	F0-19.xsl	0.800	0.900
F0-20.xml	F0-20.xsl	0.900	0.999

のグラフ画像を作製しつつ、かつそれらの画像を表示する単純なホームページを自動的に作製するようなマクロは割合簡単に作製することができた。

グラフを連続的に作製し、紙に印刷して終了するマクロの場合は、後は手作業で印刷物を管理すれば良いだけであるが、グラフ画像が実体として表計算シート上または画像ファイルとして残るケースは、その後の取り扱いが意外と大変なようである。表計算シート上にグラフ画像を並べて表示させるのは、使い勝手が悪そうで

あるし、画像をきちんとシート上に並べて配置するようなマクロを作製するのも面倒であろう。やはりグラフを画像ファイルとして残し、ホームページ上に表示させて使うのが昨今のパソコン環境では便利なのであろう。ホームページは、画面をスクロールアップ・ダウンすることにより画像を連続的に観察することができ、しかも印刷も可能である。グラフ画像を連続的に表示するホームページは単純なものならマクロにて自動的に作製できた。昨今の高速な CPU、多量なメモリを搭載するパソコンで個人的に見るだ



けの用途であればこのホームページが作製できた段階で満足してもいいのかもしれない。しかしホームページには多量の画像が含まれ、巨大なサイズになる。また画面のスクロールアップ・ダウンの操作でしか表示する画像を切り替えることができないのでは使い勝手が悪すぎる。

その場合、1つのホームページを複数のページに分け、目次フレームなどを用意して全体を1つの本の形式のホームページに統合するやり方が最も単純なやり方だろう。3-1. 節で作製したページがそれである。作製するグラフ画像範囲を変えてマクロを何回か実行すれば、複数のホームページも簡単に作成できる。

しかしその後でホームページの表示レイアウト等を好みのスタイルに変更したりすることは簡単にはできない。また、今回の例のように画像数が多量なケースは、まだまだページ分割が





不十分である。1つのページにつき10個程度の画像を表示させようとするれば、2000個の画像がある場合には200個のページを作成し、それら200個のページへのリンクがある目次を作製しなければならない。相当な手間がかかる事態となる。また基本的にいくつかに分割して自動的にホームページ作製させた副産物として作製されるグラフ画像の名称は統一が取れていないので後の利用が難しい。連続的に・規則的に変化する名称を持つ画像を生成させようとするのなら、1回のマクロ操作で画像を生成させる必要がある。やはり自動的に作製されるホームページをそのまま利用しようとするのは問題がある。目次から分割作製した固定的なページを呼び出す方式にも限界があるようだ。そう簡単には作製できないだろうが、動的に自動的に変化するようなホームページの方が好ましいよ

うだ。クライアントからの要求に従って、ホームページ上に表示する画像を柔軟・動的に絞り込んで表示できるような動作をするページである。それを目指すなら、マクロにて自動的に作製されるホームページは捨てて、グラフ画像ファイルのみを流用するほうが良さそうである。グラフ画像をデータベース化し、その情報を基に、表示画像を絞り込んで表示するようなホームページを作製することになるだろう。グラフ画像データベースとしては、XML テキストファイルをまず作成して利用を試みた。

まず Excel シート上に画像データベースを構築し、それから画像データを格納する XML ファイルを作製してみた。単純なデータ構造の XML テキストファイルなら Excel シート上にデータベースを作製し、ワープロの差込印刷機能を使って簡単に作製することができた。

次は作製した XML ファイルを使用してのホームページ作製を試みた。1つの XML ファイルが存在する時、ホームページ上にその XML ファイル内のデータの1部分を絞り込んで表示する簡単な方法として、Data Island があった。Data Island は、Internet Explorer だけの機能であり、ホームページ内に、XML データを格納し表示することができる機能である。『VB と ASP でつくる XML』（2001年株式会社ピアソン・エデュケーション発行）を参照して、グラフ画像データ XML ファイルを格納したホームページを作製してみた。

(図25.) の様なホームページが作製できる。

(図25.) は、XML ファイル内の最初の10件のデータが一覧表示されている状態である。画面下の  ボタンをクリックすれば、次の10件のデータ表示に切り替わる。 ボタンをクリックすれば、前10件のデータ表示に切り替わる。 ボタンをクリックすれば、最初の10件のデータ表示に戻る。 ボタンをクリックすれば、最後

の10件のデータ表示に戻る。またホームページ上に表示するデータ個数を変更するためのテキストボックスも用意されている。(図25.) 例では、テキストボックス内の数値が、10になっているため、10個のデータが表示されている。

(図26.) はホームページのソースである。6行目~25行目は XML ファイルが挿入されている箇所である。17行目~19行目は途中省略のある箇所である。実際は2000件以上のグラフ画像データが入る。27行目~43行目は、ホームページ上にテーブル構造を作り、XML ファイルデータを格納し表示する設定の箇所である。44行目~51行目は、表示データの切り替え操作をするためのボタンを作っている箇所である。6行目~25行目の場所に XML ファイルをコピーして挿入し、27行目~43行目を表示データに合わせて多少修正するくらいの作業で簡単に作製することができるので便利である。一応ホームページ上にデータを絞り込んで表示するという機能を有している。しかし残念なことに、この Data Island は、画像名称は表示できるが、`` タグを使えない。画像を表示することができなかった。この方法は今回のような、XML テキストデータを単純表示するだけでは無いケースには使えなかった。

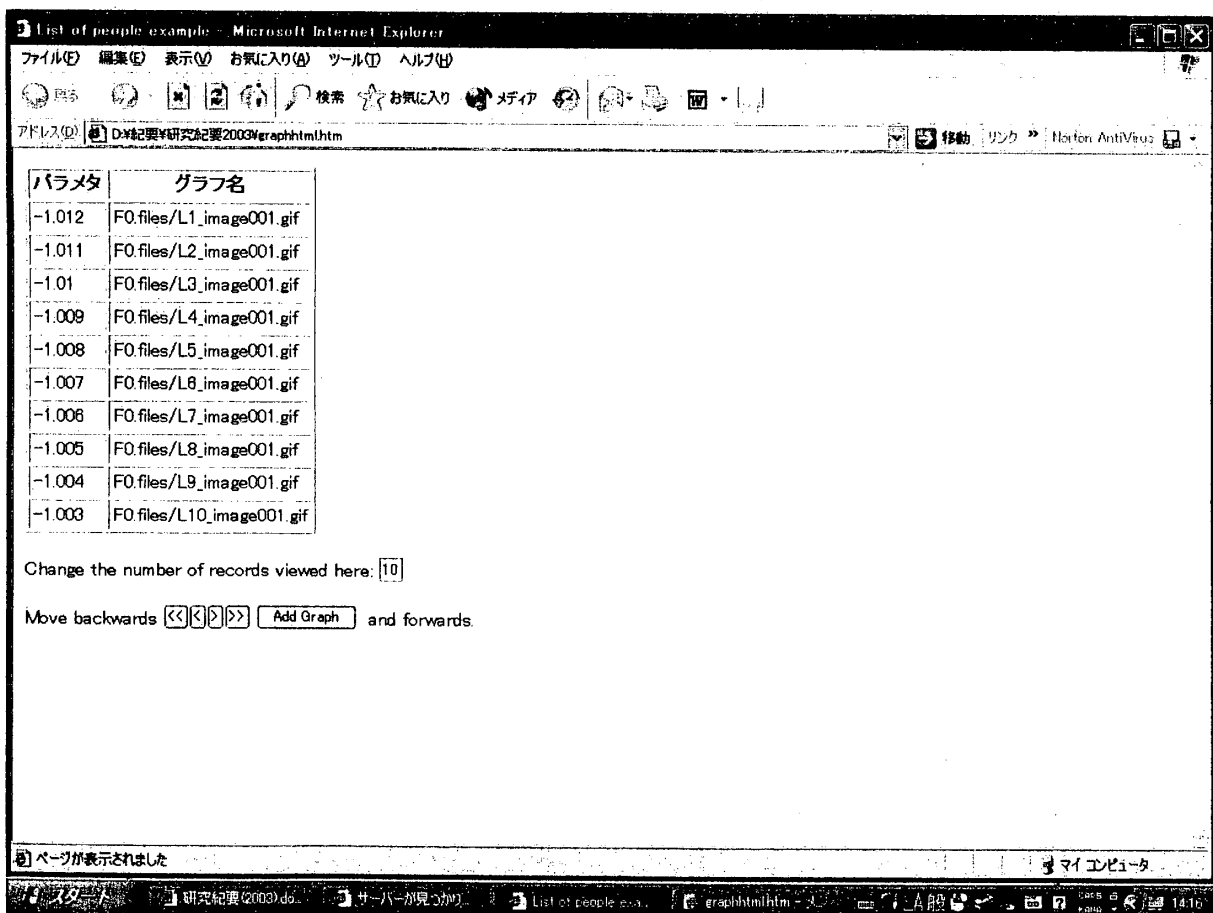
そうすると、データを絞り込んでホームページとして表示させる簡単な手段としては、XML ファイルに XSLT (Extensible Stylesheet Language Transformations) スタイルシートを組み合わせて、ホームページとしてのレイアウトを付与して表示させる方法しかとりあえず無いようであった。XSLT プロセッサの働きにより、XSLT スタイルシートを XML ファイルに適用することにより、XML ファイルをホームページ形式に構造変換させることができる。XSLT スタイルシートの指示によりデータの絞り込みなども可能であった。Microsoft 社の

ブラウザソフト Internet Explorer Version6では、MSXML3という XSLT プロセッサが組み込まれているので、XSLT スタイルシートを XML ファイルに適用してホームページとして表示させることがブラウザ上でできるため、面倒な手順が不要である。XML テキストファイル内に参照する XSLT スタイルシート名を記述しておけば、ブラウザソフトが XML テキストファイルを表示しようとする時に、自動的に XSLT スタイルシートを XML ファイルに適用してホームページ形式に構造変換し、ホームページとして表示してくれる。3-2. 節では XSLT スタイルシートを作製し、そこに記述した指示により表示データが絞り込まれて表示される XML ファイルを作製した。表示データの絞り込み条件を変えた XML ファイルを複数作製し、目次フレームから呼び出して表示するよ

うにした。3-1. 節で作製したホームページと見た目は同じである。XML ファイルと XSLT スタイルシートの組み合わせでデータを絞り込んで表示するページを作製することが割合簡単にできることが確認できた。

今回は、表計算マクロの実用的かつ簡単な応用を検討してみたのだが、結果として多量のデータ（特に画像）を扱う羽目になり、昨今進歩著しい XML 技術を使うことにもなってしまった。単純な構造の XML ファイルを扱っただけであるが、1つの XML テキストファイルから XSLT スタイルシートを適用することによってホームページを生み出すという1例を作ることもなった。カスケーディングスタイルシート (CSS) は、特定のデータ構造に表示スタイル・レイアウトを付与する働きしかないが、XSLT スタイルシートにはデータ構造の構造

図25.



```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2 <HTML><HEAD><TITLE>List of people example</TITLE>
3 <META http-equiv=Content-Type content="text/html; charset=shift_jis">
4 <META content="MSHTML 6.00.2800.1141" name=GENERATOR></HEAD>
5 <BODY>
6 <XML id=xmlGraph>
7 <?xml version="1.0" encoding="Shift_JIS"?>
8 <graph>
9 <record>
10 <パラメタ>-1.012</パラメタ>
11 <グラフ名>F0.files/L1_image001.gif</グラフ名>
12 </record>
13 <record>
14 <パラメタ>-1.011</パラメタ>
15 <グラフ名>F0.files/L2_image001.gif</グラフ名>
16 </record>
17
18
19
20 <record>
21 <パラメタ>0.999</パラメタ>
22 <グラフ名>F0.files/L2012_image001.gif</グラフ名>
23 </record>
24 </graph>
25 </XML>
26
27 <TABLE id=tblGraph dataSrc=#xmlGraph cellPadding=3 dataPageSize=10 border=1>
28 <THEAD>
29 <TR>
30 <TH>パラメタ</TH>
31 <TH>グラフ名</TH>
32 </TR>
33 </THEAD>
34 <TBODY>
35 <TR>
36 <TD><SPAN dataFld=パラメタ></SPAN></TD>
37 <TD><SPAN dataFld=グラフ名></SPAN></TD>
38 </TR>
39 </TBODY>
40 </TABLE>
41 <P>Change the number of records viewed here:
42 <INPUT onBlur=tblGraph.dataPageSize=this.value; style="WIDTH: 20px" value=10>
43 </P>
44 <P>Move backwards&nbsp;
45 <BUTTON id=cmdfirstPage onclick=tblGraph.firstPage()>&it;&it;</BUTTON>
46 <BUTTON id=cmdpreviousPage onclick=tblGraph.previousPage()>&it;</BUTTON>
47 <BUTTON id=cmdnextPage onclick=tblGraph.nextPage()>&gt;</BUTTON>
48 <BUTTON id=cmdlastPage onclick=tblGraph.lastPage()>&gt;&gt;</BUTTON>
49 <INPUT id=add onclick=xmlPeople.recordset.addNew() type=button value="Add Graph">
50 &nbsp;&nbsp;and forwards.
51 </P>
52 </BODY>
53 </HTML>

```

図26.

変換をする機能がある。XML ファイル内の必要なデータのみ抽出し、並べ替え、必要であれば集計などの作業を実行し、ホームページ形式で表示させることができる。XSLT スタイルシートを使えば、XML テキストファイル内を解

析するパーサの関数を使って特別なプログラムを作成するほどの手間を必要とせずに XML テキストファイルからホームページを生成させて表示させることができる。単純な XML 技術の利用なら表計算ソフト Excel とワープロソフト

ト Word でも対応できそうであり、今回試みたレベルなら XSLT スタイルシートの学生演習用の 1 つの例題として使えるかもしれない。

ただし次のレベルの段階として、XML ファイルと XSLT スタイルシートの組み合わせで動的に変化するホームページの作製について検討する必要がある。表示するデータを固定的に絞り込んで表示するのではなく、ホームページ上での操作指示に従って柔軟に・動的に表示データを切り替えるようなページが望ましい。しかしこれはさすがに Javascript などのスクリプト言語で、それなりのスクリプトを作製しホームページに入れなければならないだろうと想像される。このレベルは私が学生の演習で扱っているレベルを超えている。しかしホームページに表示する画像データなどが多量あるようなケースでは、手作業で沢山の静的なホームページを作製し、目次から呼び出せるように作るのはかなり面倒である。この方法には基本的に限界があるようだ。限界が予想される以上、それに代わる手段を考えてみるというは有意義であろう。

サーバ側にて ASP (Active Server Pages) のようなプロセスが動作していて、クライアントパソコンからの要求・指示に従ってホームページ上に表示する画像を絞り込んで表示するような特別なスクリプトをホームページに入れることができる環境があれば、もちろんいろいろなやり方で実現可能だろう。XML ファイル、XSLT スタイルシートを使わなくても可能である。Access などのデータベースに格納した画像情報を使うやり方も可能である。そのようなサーバ側での特別な環境を必要とし、サーバ側で動作する特別なスクリプト (プログラム) を作製することなく、なるべく単純なやり方で実現したい。今後は動的なページ作製に XML ファイルと XSLT スタイルシートの組み合わせ

でどこまでシンプル方法で対応ができるものであるか試みてみたい。シンプルに実現可能であるならば、学生へ提示する 1 つの例題として提示することができると思われる。

[参考文献]

- 1) 白田昭司、東野勝治、井上祥史、伊藤敏、葭谷安生共著『カオスとフラクタル ① Excel で体験』 1999年 株式会社オーム社
- 2) Mark Wilson、Tracy Wilson 著、浜田真理訳、浜田光之監訳『VB と ASP でつくる XML』 2001年株式会社ピアソン・エデュケーション
- 3) Michael Kay 著、IDEA・C 訳、佐藤直生監修『XSLT バイブル』 2002年 株式会社インプレス
- 4) 井上孝司著『Web コンテンツ作成のための XSLT 入門』 2002年 株式会社毎日コミュニケーションズ
- 5) 大田一郎・柴田史久著『XML ツールキット』 2001年 株式会社秀和システム