

SIST 情報システム

The SIST Information System

菅沼 義昇*, 幸谷 智紀**

Yoshinori SUGANUMA and Tomonori KOUYA

Abstract: The SIST Information System is a kind of system to evaluate education system. Its can display graphs and tables of a distribution of scores on a test, a questionnaire, etc., based on the SIST database system. In this paper, its contents and ability are precisely described.

1. はじめに

教育の効果は、数値的に表現するのが困難な場合、卒業後数十年経てから初めて明らかになる場合、など、その完全な評価は難しいが、教育システムを改善するためには、何らかの方法でシステムの評価を行う必要がある。SIST 情報システム (学内専用) は、教育システム評価の一手段として作成したものである。

学生の成績や各種アンケートの結果は、教育システムに対する評価の元となるデータとして大きな意味を持っている。しかしながら、これらのデータを全教職員に公開することは、個人情報保護の立場から難しい。さらに、公開できたとしても、それらのデータを解析するためには、解析するためのソフトウェアを使いこなせる必要があること、同じような処理をデータが変わる毎に繰り返す必要があること、解析のために時間がかかること、など、新たな問題が発生する。

SIST 情報システムでは、これらの問題を解決するために、個々のデータではなく、解析結果だけを Web 上で公開する。ユーザは、ブラウザの操作を行うことができさえすれば、解析結果をグラフや表として見ることが可能である。また、大学基幹システム (データベース) 上のデータに基づいているため、大学基幹システムにおけるデータの変更が解析結果に自動的に反映されていき、新たな作業を必要としない (現時点では、部分的な実現に止まっている)。

以下、SIST 情報システムと大学基幹システムとの関係、及び、SIST 情報システム自身について詳細に述べていく。

2. データベース

2.1 大学基幹システムとの接続

今回我々は、学内ネットワークからいつでも参照可能な学内情報の統計データ閲覧用の Web ページを Linux マシ

ンである cs-www (133.88.121.100) 上に作成するために、ODBC-ODBC bridge を活用したデータベースネットワークを構築した。本学の学内情報は事務局内ネットワークにある Oracle データベースに保存してある。それを直接参照できれば問題なかったが、事務局内のセキュリティポリシーがそれを許さなかったため、Oracle データベースと cs-www の間に Gateway マシンを設け、そこと cs-www 間に ODBC 接続によるブリッジ (ODBC-ODBC bridge) を架けることにより、Oracle データベース内の情報を間接的に読み取ることができるようにした。この節では、その概要と ODBC-ODBC bridge の技術的な詳細を述べる。

2.1.1 データベースネットワークの概要

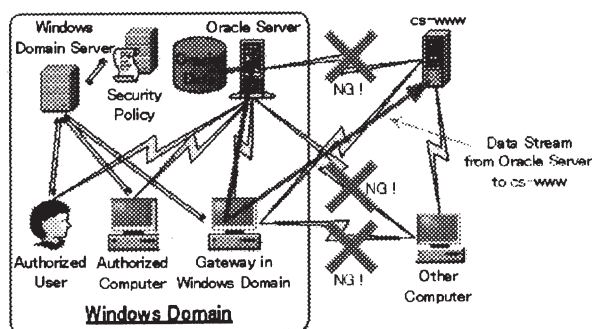


図 1 今回構築したデータベースネットワーク

図 1 に今回構築したデータベースネットワークを示す。以下これに従って解説を行う。本学が業務に用いているデータはすべて Oracle データベースに格納されている。そして、このデータベースを直接参照できるユーザやコンピュータは、ある Windows ドメインに属している必要がある。具体的には、Windows ドメインを管理するサーバ (Windows Domain Server, WDS) によって認証されたユーザやコンピュータでなければ、Oracle データベースを参照することができないようになっている。WDS はあらかじめ

2009年3月3日受理

* 総合情報学部 人間情報デザイン学科

** 総合情報学部 コンピュータシステム学科

め定められたセキュリティポリシーに従ってユーザやコンピュータの認証を行っており、これに反するドメイン外からのアクセスはすべて遮断するように設定されているのが普通である。従って、ドメイン外のコンピュータ (Other Computer) や cs-www から Oracle データベースを直接参照することは不可能である。

そこで我々は事務局内ドメインに Gateway マシンを一台設置し、Oracle データベースの内容を ODBC インターフェースを介して Access ファイルとして参照できるようにした。そしてそこにフリーの ODBC-ODBC bridge である brodbc [brodbc]のサーバを組み込み、同じ brodbc のクライアントを cs-www に設置して、cs-www から間接的に Oracle データベースの情報を読み取れるようにした (図1内の太矢印線が情報の流れ)。この結果、Oracle データベース内の情報は Gateway を介して cs-www で参照でき、学内のユーザはこの情報をもとに cs-www 内の PHP スクリプトによって加工された統計データを入手できるようになる。

このようなデータベースネットワークには長所と短所がある。長所は以下の通りである。

- ・事務局内 Windows ドメインのセキュリティポリシーの書き換えを必要としないため、書き換え時の過失による重大なセキュリティホールを発生させない
- ・極めて限定されたネットワーク接続でしか Oracle データベース内の情報を参照できないため、セキュリティの確保が容易である (cs-www と Gateway マシン間の接続だけを監視すればよい)
- ・Oracle データベースを直接参照しないため、仮に cs-www マシンが乗っ取られたとしても元の Oracle データベースが破壊されることはない (但し情報自体は盗まれるため、cs-www の監視は常に必要)

反面、次のような短所もある。

- ・Oracle データベースと管轄する Oracle サーバと Gateway マシン双方の I/O 性能によって cs-www へのデータ転送速度の著しい遅延を引き起こす可能性がある (現に、遅延は発生している)
- ・Oracle データベースの直接参照ができないため、cs-www 側でのデータベース参照は大変難しくなる (結局、テーブル構造を別経路で入手して解決せざるを得なかった)

一言でまとめると、セキュリティの確保には有用だが、データ転送速度とシステム構築に必要な時間が多大になりがち、というのが本データベースネットワークの特徴であると言える。

2.1.2 ODBC-ODBC bridge の役割

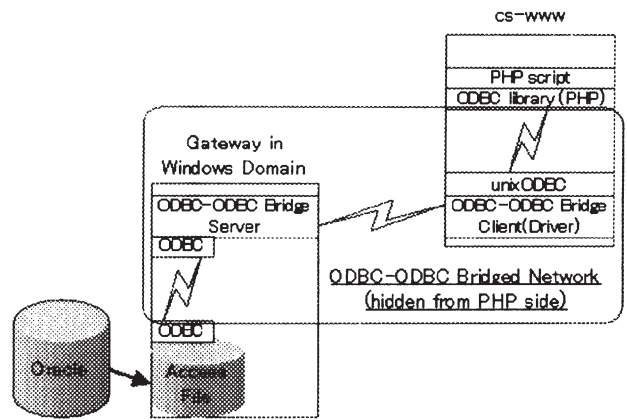


図2 ODBC-ODBC bridgeのレイヤー構造

本節では cs-www と Gateway マシンとの間に架けられた ODBC-ODBC bridge について解説する。まず図2のソフトウェアレイヤー図に沿って説明を行う。

通常、RDBMS (Relational DataBase Management System) は SQL 文を介して他の言語 (C/C++, Java, Perl, PHP, Python, Ruby など) とのやり取りを行うが、SQL 文をどのようなプロトコルによって受け渡すかは RDBMS や言語によって異なる。主要な Web 開発言語の一つである PHP では、各 RDBMS ごとに異なるライブラリとドライバが用意されており、ユーザはこれを使い分けて RDBMS とのアクセスを制御することになる。主として AccessTM データベースとのアクセスに用いられる ODBC (Open DataBase Connectivity) インターフェースも主要な Web 開発言語では使用可能であり、当然 PHP にも ODBC ライブラリが用意されている。Linux 環境でこの ODBC ライブラリと、unixODBC (ODBC ドライバ) [unixodbc]をセットにして使用するのが標準のようである。

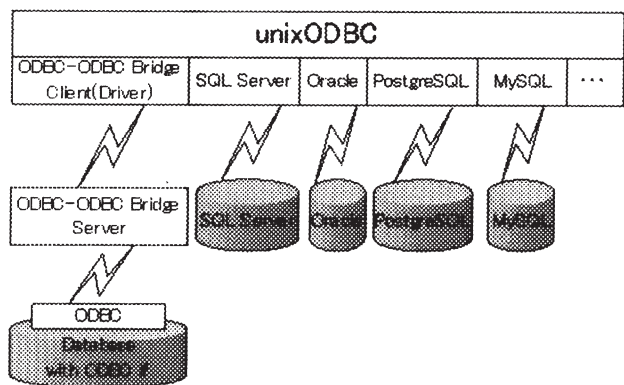


図3 unixODBCと各種RDBMSとの連携

この unixODBC は、Vine Linux 標準のものでは /etc/odbc.ini に ODBC アクセス用の設定を書くようになっている。ODBC インターフェースは Microsoft 製品では古

くから使用されてきたせいもあって、unixODBC も Windows 上で動作する主要な RDBMS と連携して使用できるよう、各 RDBMS 用のドライバを組み込めるようになっている (図 3)。

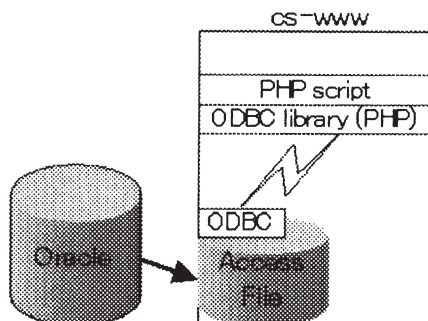


図4 PHP側からのアクセス概念図

このように unixODBC を組み込むことによって、PHP 側からはどの RDBMS を用いているかを全く意識することなく、ODBC ライブラリ関数だけを使用して全ての ODBC インターフェース付き RDBMS を操作することができるようになる (図 4)。

今回はネットワーク上に存在する二つの ODBC インターフェース間を接続するために、ODBC-ODBC bridge を組み込んだ。2008 年現在、この unixODBC 用のドライバが提供されている ODBC-ODBC bridge は Easysoft 社の開発したもの[easysoft]と、個人で開発されたオープンソースの brodbc [brodbc]が入手可能であったので、今回は後者の最新版 (brodbc03.tar.gz) を使用した。Gateway マシンに必要な ODBC-ODBC bridge サーバは別途インストールし、Windows 上のサービスとして常時起動しておく必要がある。

2.1.3 Brodbc を用いた ODBC-ODBC bridge の構築手順

本節では Vine Linux 3.1 で unixODBC + brodbc03.tar.gz (ODBC-ODBC bridge) を使う手順を示す。

(1) brodbc [brodbc]の最新版 brodbc03.tar.gz を cs-www 上にダウンロードし解凍する。その中の ODBC-ODBC bridge サーバ (orbrd.exe) を Windows 側にインストールし、下記項目をチェックしておく。

- ・サーバを組み込んだ Gateway マシンの IP アドレス (以下、gatewayIP とする)
- ・ポート番号 (22002 がデフォルト。以下、gatewayPort とする)
- ・ODBC 接続のための項目
- ・DSN (sampleDSN)
- ・Database 名 (sampleDBName)

- ・ユーザ ID (sampleUID)
- ・パスワード (samplePWD)

(2) UNIX ODBC ドライバを cs-www にインストールする
 (3) brodbc03.tgz のうち、client/makefile.ux を次のように書き換える (libodbc には SQL API がなくなっていたので)

```
ODBCLIB = /usr/lib/libodbcinst.so.1.0.0
```

(4) make して brodbc.so を /usr/lib にコピー

(5) /etc/odbcinst.ini にドライバ名 (brodbc03) を追記

```
[brodbc03]
Description = ODBC-ODBC bridge
Driver = /usr/local/lib/brodbc.so
UsageCount = 1
```

(6) /etc/odbc.ini に下記を追記して、ODBC から "sampleBrodbc" でアクセスできるように設定

```
[sampleBrodbc]
Driver = brodbc03
Server = gatewayIP
Port = gatewayPort
DSN = sampleDSN
Database = sampleDBName
Username = sampleUID
Password = samplePWD
Servercharset = Shift_jis
Charset = eucjp
Debug = 0
```

(7) cs-www から isql コマンドを下記のように用いて ODBC 経由の接続が可能であることを確認

```
$ isql sampleBrodbc
+-----+
| Connected! |
|          |
| sql-statement |
| help [tablename] |
| quit      |
+-----+
```

SQL> // このプロンプトが出れば OK

(8) php-odbc ライブラリを cs-www にインストールする。もし Apache2 をソースからコンパイルしている場合は、PHP インタプリタも作り直しておく

(9) 下記のような PHP スクリプトを作成し、接続テストを行う

```
<html>
<body>
<?php
```

```

//データベースへの接続を開く
$conn_id = odbc_connect("sampleBrodbc",
"sampleUID", "samplePWD");
if ( !$conn_id ) {
    print "データベースにつながりません\n";
    exit;
}

// SQL Query 発行
$result = odbc_do($conn_id, "SELECT * from テーブル名");
if( !$result ) {
    print "警告!: 結果が正しく戻ってきません.\n";
}
else {
    while ( odbc_fetch_row($result) ) {
        print odbc_result($result, 1) . ", " .
odbc_result($result, 2) . "\n";
    }
}

//データベースへの接続を閉じる
$ret = odbc_close($conn_id);
?>
</body>
</html>

```

2.2 SIST 情報システム用データベースの作成

大学基幹システムにおけるデータの変更を解析結果ヘリアルタイムに反映させるためには、大学基幹システムと直接接続し、そこから得られたデータに基づき解析を行うのが理想であるが、

- ・接続に時間がかかりすぎる (20 秒程度)
- ・大量のデータ (テーブルによって異なるが、数千件以上) を読み込めない
- ・ODBC が SQL の一部しかサポートしていないため、使用に不都合が生じる場合がある。
- ・アンケート結果、プレースメントテストの結果など、大学基幹システムに存在しないデータがある

などの理由により、SIST 情報システムが存在する Web サーバ上に、MySQL によって専用のデータベース (以下、SIST_info と呼ぶ) を構築することとした。SIST_info は、大きく分けて二つの部分からなっている。一つは、大学基幹システムのデータに基づく部分であり、学生や企業に関する情報を保存する 23 個のテーブルから構成される。

基本的に、これら 23 個のテーブルは、一部のテーブル

を除き、半期または 1 年に一度更新されるデータによって構成されており、適切な時期に更新を行えば、これらのテーブルを使用してほぼリアルタイムの情報を得ることができる。最初に、その時点で存在するすべてのデータを取得しておき、その後、変更のあった部分だけを更新していく方法が最も効率的ではあるが、プログラムのバグ、更新情報の誤りなどによって、場合によっては、大学基幹システムとの不整合が生じかねない。そこで、更新するプログラムを簡単にし、プログラム上のバグを少なくすることを優先し、すべてのデータを一括して更新する方法を採用した。ただし、成績を保存したテーブルだけは、データ量が多いため、半期ごとにデータを追加していくこととした。

ここで問題となるのが、先に述べた「一度に大量のデータを取得できない」といった問題である。現在のシステムにおいてこの問題を回避するために、

- 1) 大学基幹システムに接続する
- 2) 適当な量 (たとえば、1991 年度から 1995 年度までのデータ) のデータを読み込んで処理する。
- 3) 大学基幹システムとの接続を切る

を繰り返してすべてのデータを取得するといった方法をとらざるを得なかった。もちろん、これらの操作を手動で行うわけではなく、実際には、Java アプレットを介して行っている。

本学では、カリキュラムを、Ⅰ類 (人間・文化科目)、Ⅱ類 (学部共通専門基礎科目)、Ⅲ類 (学科専門科目) の 3 つの類によって構成しているが、大学基幹システムでは類毎の GPA を計算することができない。そこで、更新されたデータに基づき、GPA と単位取得率を計算し、上で述べたテーブルとは別に新たなテーブル GPA_table を作成している。テーブル GPA_table には、学生毎、かつ、半期毎に、Ⅰ類、Ⅱ類、及び、Ⅲ類に対するその期における GPA の分母、GPA の分子、履修単位数、取得単位数を保存してある。同時に、その期までの累積値も保存してある。また、同様に、年度初めと年度末の学生数からなるテーブル number_table を更新時期に合わせて作成している。

以上述べた処理は、具体的には、図 5 に示す管理者用のページから実行可能である。すべてのデータを初期状態から更新するには 5~6 時間必要とする (GPA に関しては、計算時間そのものが 3 時間程度かかる)。しかし、学生の成績と GPA_table の更新は、追加データだけに対して行われるため、通常は、数十分で更新手続きを実行可能である。

図 5 は、図の左側のメニューから、「SIST データベースの更新」を選択した場合を示しているが、このページでは、ユーザの管理、テーブル情報の出力、ファイルを利用したテーブルの作成・データの追加、テーブルの削除なども実

行可能である。

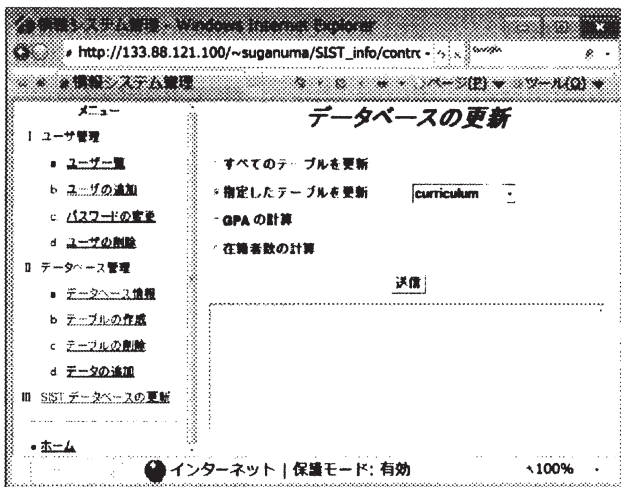


図6 データベース管理画面

SIST_info を構成するテーブルの他の一つの部分は、大学基幹システムに存在しないデータである。このようなデータの内、現時点で、SIST_info に存在するデータは以下通りである。

- 1) 過去に行ったアンケート結果
- 2) 数学、及び、英語プレースメントテストの結果
- 3) 教育開発センターへの訪問者データ

これらのデータの内、アンケートやプレースメントテストの結果は、MySQL によって読み込み可能なファイル形式に変換し、そのファイルに基づき、図5に示したページを利用して、その都度テーブルをオフラインで作成している。教育開発センターへの訪問者に関しては、訪問者があるたびに、SIST_info 内のテーブルに直接データを追加しているため、ほぼリアルタイムの更新が行われている(図6参照)。

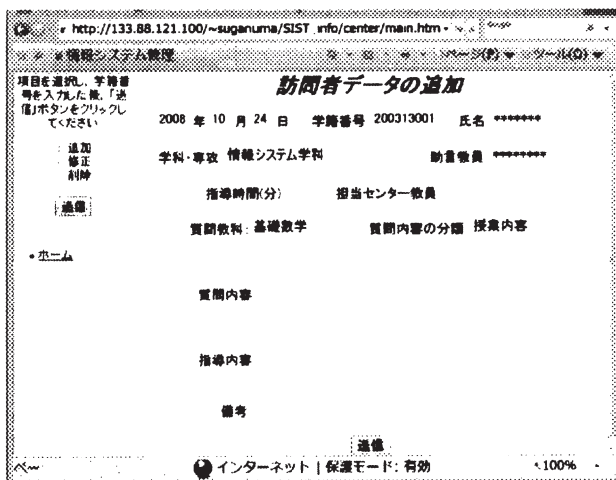


図6 教育開発センター訪問者データの入力画面

3. SIST 情報システム

3.1 動作と構成

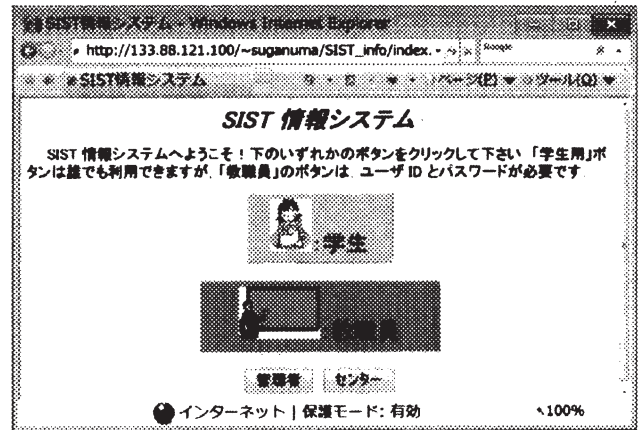


図7 トップ画面

SIST 情報システムに入ると、図7に示すようなページが表示される。「学生」と書かれたボタンは誰でも利用でき、このボタンをクリックすることによって、現時点では、入学年度毎の GPA、単位取得率、及び、取得単位数の分布を見ることができる。これ以外のボタンには、それぞれ、ユーザ ID とパスワードが設定されており、左下の「管理者」と書かれたボタンをクリックすると管理者用(図5)、右下の「センター」と書かれたボタンをクリックすると教育開発センター訪問者データの入力用(図6)のページへ移動可能である。

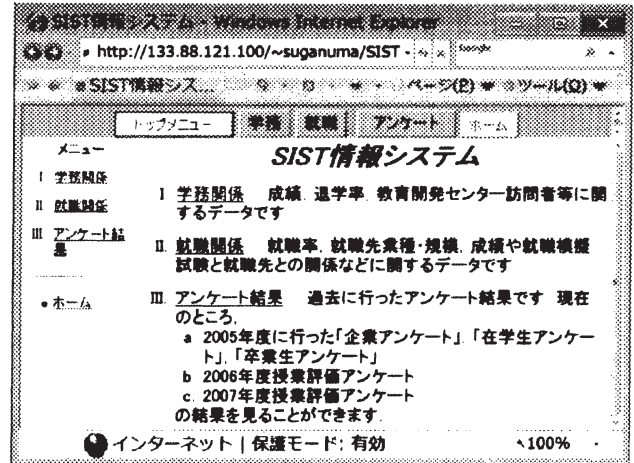


図8 メイン画面

最も一般的に利用するのが、「教職員」ボタンである。このボタンをクリックすると、図8に示すようなページが表示される。例えば、就職先を業種によって分類した結果を見たいような場合は、この画面において「就職関係」を選択すると、図9に示すような画面になる。

図9に示す画面において「就職先企業の分類」、さらに、「就職先企業の業種別分類」を選択すると、図10に示す画面となる。この画面において、年度を入力し、かつ、適

当な項目を選択した後、「積み上げ式棒グラフ」、または、「円グラフ」ボタンをクリックすれば結果が表示される。なお、一般に、グラフボタンは、表示する項目によって異なったグラフボタンが表示される。

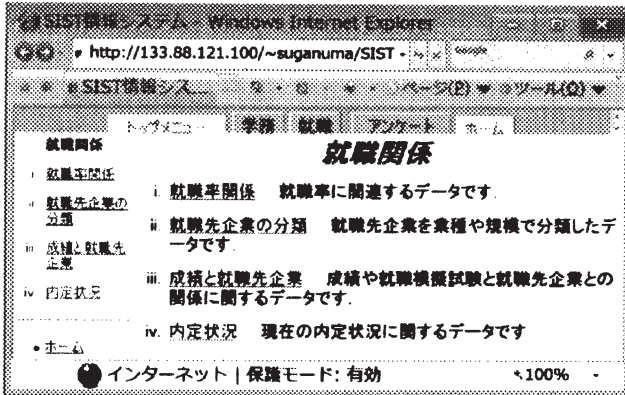


図9 就業関係の画面

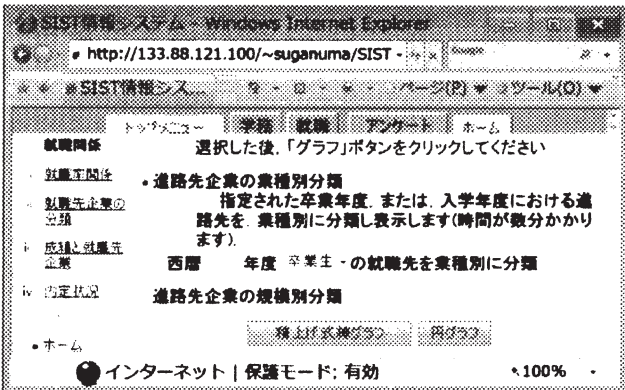


図10 就業関係の画面(業種別分類を選択した場合)

グラフボタンがクリックされると、結果を表示するためのプログラムに制御が移る。このプログラムは、PHP で記述されており、データベースに接続し必要なデータを収集し、必要な計算を行い、その結果を、表形式で出力し(場合によっては、表形式の出力は行わない)、かつ、結果を Java アプレットにパラメータとして渡すことによってグラフを描画する。上で述べた例の場合は、career_kind_graph.php がこの処理を行い、その結果、図 11 に示すようなページと図 12 に示すようなグラフ(円グラフの場合における「情報」に対する結果)が表示される。

プログラム career_kind_graph.php は、データベースから必要なデータを引き出し、必要な計算を行い、その結果を表として出力している。さらに、積み上げ式棒グラフ、または、円グラフを出力するために、APPLET 要素を出力している。

この結果出力されるのが図 11 であり、そのソースは、大きく分けて、TABLE 要素と APPLETT 要素から構成されている。円グラフの場合、全体及び学科毎にグラフが出力されるため、5つのAPPLET要素が記述されている。た

だし、アプレットのサイズが 0 となっているため、図 11 にアプレット自身は表示されない。

Java アプレットは、パラメータとして受け取ったデータに基づき、新しく生成した Window 上に指定されたグラフを描き、そのグラフの例が図 12 になる。一般的に、描くことが可能なグラフは、棒グラフ、折れ線グラフ、積み上げ式棒グラフ、円グラフ、散布図、レーダーチャート、及び、ボード線図(片対数グラフ)であるが、現在の SIST 情報システムで利用しているのは、最初の 5 つのグラフである。描かれたグラフは、マウスを使用することによってサイズを変更したり、表示方向を変更することも可能である。なお、グラフ描画プログラムの詳細に関しては、「C/C++とJava」の20章のプログラム例を参照されたい。

業種	人数	割合	人数	割合	人数	割合	人数	割合	全体	
輸送用機器	20	34.7	31	5	3.3	6	8.2	39	13.3	
電気・電子機器	5	6.2	20	26.0	37	6	82	33	11.6	
機械	11	18.6	13	19.9	5	6	6.6	32	11.2	
その他	3	9.9	5	6.5	5	9.3	6	8.2	24	8.4
情報処理業	0	0	0	0	6	0	16	21.9	7.7	
高社	2	2.5	3	3.9	6	11.9	1	1.3	20	7.0
建設業	6	7.4	7	9.1	0	0	1	1.4	14	4.9
化学	1	1.2	1	1.3	10	18.5	1	1.4	15	4.6

図11 就業関係(表示結果)

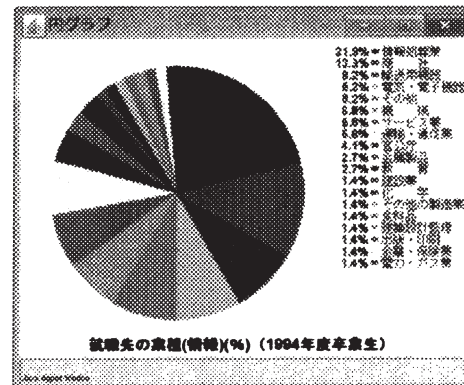


図12 就業関係(結果のグラフ)

3.2 機能

現時点で利用可能な機能をまとめると、以下に示す通りである。

A.学務関係

a.GPA と単位取得率

- ・成績 (GPA) の分布
- ・入学年度毎、または、学期毎の GPA の推移
- ・単位取得率 (取得単位数/履修単位数) 及び取得単位数の分布

- ・入学年度毎, または, 学期毎の単位取得率の推移
- ・個人別成績データ (パスワードが必要)

b.成績間の関係

- ・数学プレースメントテスト, 英語プレースメントテスト, 及び, 指定された年次・学期・類 (I類, II類, III類) の成績間の関係

c.入試種別と成績

- ・入試種別と数学プレースメントテスト, 英語プレースメントテスト, または, 指定された年次・学期の成績との関係

d.退学率関係

- ・年度毎の年間退学率, または, 入学年度毎の累積退学率
- ・入試種別, プレースメントテスト, 1年前期の成績と退学率との関係

e.教育開発センター関係

- ・月ごとの学科別訪問回数, 教科別訪問回数, 入学年度別訪問回数, 及び, 入学年度別実訪問者数
- ・年度ごとの学科別訪問回数, 教科別訪問回数, 入学年度別訪問回数, 及び, 入学年度別実訪問者数
- ・教育開発センターを利用したことがある学生の数と入学者に対する比率
- ・学生をプレースメントテストまたは成績 (GPA) によってグループ分けし, 各グループにおける教育開発センター利用経験者数とその割合
- ・数学プレースメントテストの結果によって学生をグループ分けし, 学グループごとに, 教育開発センターを利用したことがある学生とない学生との成績 (GPA) を比較
- ・指定した学籍番号に対応する学生の訪問記録
- ・全データ

B.就職関係

a.就職率関係

- ・卒業年度, または, 入学年度における進路先を, 就職, 大学院, その他の進学, 及び, その他に分類
- ・卒業年度毎, または, 入学年度毎の就職率の推移を表示 (卒業生の数を分母とし, 就職, 進学など, 進路が決まった学生の数を分子)

b.就職先企業の分類

- ・卒業年度, または, 入学年度における進路先を, 業種別に分類し表示
- ・卒業年度, または, 入学年度における進路先を, 上場/非上場別, 資本金別, または, 従業員数別に分類

c.成績と就職先企業

- ・卒業時までの成績, 一般常識模試の結果, または, 基礎学力模試の結果と一般常識模試の結果, 基礎学力模試の結果, 就職先の資本金 (対数値), または, 就職先の従業員数 (対数値) との関係

d.内定状況

- ・現時点における内定状況 (このデータだけは, 大学基幹システムからリアルタイムにデータを取得している)

C.アンケート

- 2005年度に行った企業アンケート, 在学生アンケート, 及び, 卒業生アンケート
- 2006年度授業評価アンケート
- 2007年度授業評価アンケート

4. おわりに

一応, アンケートや学務関係に関する部分を作成したが, その機能として十分であるとはいえない. 特に, 入試関係部分を拡充していく必要がある. また, 特定の教育上の措置がどのような効果を上げたかといったようなことを見るためには, 個別の対応が必要であろう. いずれにしろ, 定常的なアップデートが必要である.

反応速度の遅さも気になる場所である. ネットワーク, MySQL, PHP, Javaなどが影響を与えていると思われる. Javaに関しては, 最初のグラフだけは表示に時間がかかるが, それ以降の表示に関しては特に問題がない. 他の3つの要素に関しては, 今後, その原因を追究し, 改善していきたい.

大学基幹システムと接続したといっても, その範囲は限定されている. 個別に, オフラインでテーブルをアップデートしていかなければならない部分が多く残っており, これらを今後どのようにするかは大きな課題である.

参考文献

- 1) <http://www.amy.hi-ho.ne.jp/jbaba/brodabc/brodabc.htm>
(ODBC プロトコル・ブリッジ)
- 2) http://www.easysoft.com/products/data_access/odbc_odbc_bridge/ (ODBC-ODBC Bridge)
- 3) <http://www.unixodbc.org/> (unixODBC)
- 4) <http://www.sist.ac.jp/~suganuma/cpp/mokuji.htm>
(C/C++ と Java)