6V−8

大量情報の組織化とオブジェクトデータベースシステム Jasmine を用いたデータベース開発*

久保 正明**, 尾下 真樹**, 金子 邦彦+, 牧之内 顕文+, 森田 亙昭**, 龍 浩志**, 峯 肇史**
+ 九州大学大学院システム情報科学研究院
** 九州大学大学院システム情報科学府†

1 はじめに

大学の研究室で取り扱っている情報は非常に多い. 例えば, 研究で発表した論文や刊行物等の文献, 学生の名簿, 研究内容, 関係組織, 学会などでの発表内容が挙げられる. これらの情報を効率的に組織化し, その情報を外部に対して公開することを目的とした.

このシステムの構築にはオブジェクトデータベースシステム Jasmine を利用した. Jasmine ではアクセス用の言語として ODQL(Object Query Language) が用意されている. しかし Jasmine を利用するには Jasmine 固有の使い方やスキーマ設計に精通している必要がある. そこで一般のユーザにも使いやすいようにユーザインターフェースを用意した. インターフェースには Web ブラウザを用い, データの入力, 検索をブラウザ上で行えるものを作成した.

2 データベース設計

クラス設計の中心となったのは外部の人を含め研究室に関わるすべての人を表す人物クラスと研究業績を表す文献クラスである (図 1). 人物クラスの下位クラスにはメンバーを表すクラス, その下位には職員クラス, 学生クラスを表すクラスを定義した. さらに出張やそのドキュメントのクラスも定義した.

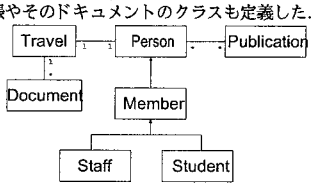

図 1: クラス構造

2.1 文書ファイル

ドキュメント一つに出張報告書などの文書ファイルがある. 出張などで書類を作成しなければならないことは多々ある. この時, 毎回同じ項目を記入するのは煩わしい. そこで, 出張記録をデータベースで管理し, その内容を文書に反映させる. データベースでは出張用ドキュメントを管理する. ドキュメントには文字列置換用の VBA が埋め込まれている. データベースに出張記録を登録するとその内容を記述した CSV ファイルが生成される. ユーザがドキュメントとこの CSV ファイルをダウンロードしワードファイルを開くと CSV ファイルの内容が自動的にワードファイルに入力される.

出張クラス内には出張に必要な項目と CSV ファイル作成, 登録手続きが定義されている. 出張の項目を入力したあとこの手続きが呼び出される. 手続きは以下のようなものである.

1. 置換用の文字列,","(カンマ), データの形式でファイルに出力 (例:name, 九大太郎)
2. ファイルの文字コードを SJIS に変換
3. ファイルをデータベースに登録

UNIX と Windows の改行コードが違うので Windows で作成すると改行コードを一つの区切りとしてくれるが UNIX で作成したものはそうはいかない. そこで CSV ファイルが見にくくはなるがすべての区切りを","(カンマ) にした. (例:name, 九大太郎,year,2001)

3 インターフェース設計

3.1 論文一覧ページ

論文の一覧ページは, 検索エンジンの検索に引っ掛かるように Jasmine で動的に作成するのではなく, あらかじめ作成しておく. 構成は以下のようにしている.

1. 年ごとをページを分割
2. 1 査読の有無, 2 英語か日本語か, 3 日付の順にソート
3. ページを作成する時点でデータベースからファイルを取りだし, そのファイルへのリンクを張る

3.1.1 論文一覧ページの作成法

Jasmine では Web ページを odb-get.exe という CGI スクリプトにアクセスすることにより呼び出すことができる。これを Web 上ではなく直接シェルスクリプトから操作することにより論文一覧ページの作成を可能にした。スクリプト実行後返ってきた文字列をファイルに保存すると Web ページの完成となる。このシェルスクリプトを実行すると論文をデータベースから検索して, そのリストを年別にファイルに出力する. 論文の PDF ファイルは論文リストと同じディレクトリに出力され, そのファイルにリンクが張られる. 生成したファイルへのリンクするページは管理者が HTML ファイルを生成する. またこのスクリプトも管理者が定期的に実行させる必要がある. 以下がこのスクリプトである.

```
#! /bin/sh
REQUEST_METHOD=GET
export REQUEST_METHOD
DIR_PATH=directory_path
TEMPLATE=template_name
YEAR=1989
THISYEAR=`date +%Y`
while test $YEAR -le $THISYEAR
do
QUERY_STRING="WIT_template=${TEMPLATE}
    &dirPath=${DIR_PATH}&is_list=TRUE
    &year_condition=only&onlyyear=${YEAR}"
export QUERY_STRING
echo $YEAR
/PATH/odb-get | sed '1,9d' > ${DIR_PATH}list
${YEAR}.html
```

CGI への引数の受け渡しは GET を用いる.

3.2 文献検索

文献検索処理を以下のように作成した. これは一般的な検索エンジンと同様な機能を持ったものである.

1. 検索条件指定入力画面による条件入力

   検索条件入力テンプレートにより, 文献を検索するための条件項目を提供する. 文献の発行年, 種別, 言語, 査読, キーワードの 5 つの条件項目がある. 値を設定しておく. キーワードに対しては, テキスト入力ボックスを与え, ユーザが直接文字列を入力する. 値の受け渡しには INPUT タグを用いている.

*Organizing mass data and the development of database using an object database system Jasmine , Masaaki KUBO , Masaki OSHITA , Kunihiko KANEKO , Akifumi MAKINOUCHI , Nobuaki MORITA , Hiroshi RYU , Tadafumi MINE

†Graduate School of Information Science and Electrical Engineering, Kyushu University,6-10-1 Hakozaki, Higashi-Ku, Fukuoka 812-8581, Japan

2. 検索結果表示テンプレートを呼び出し，入力値を渡す

   検索開始ボタンが押されると，検索結果表示テンプレートが呼び出され，ユーザの選択した選択肢に応じて，設定しておいた値が文字列値としてそのテンプレートに渡される．キーワードの場合は，ユーザが入力した文字列が直接渡される．

3. ODQL 文による条件検索式の実行

   検索結果表示テンプレート内では受け取った値をもとにODQL 文が実行される．発行年は，文字列値を整数値に変換する．また，キーワードに空白で区切られた複数の単語が指定された場合，空白を取り除き複数の文字列の集合に変換する．文献クラスのインスタンスの集合を格納する変数を用意し，ODQL 文の条件検索式を満たすオブジェクトが変数に格納される．発行年，種別，言語，査読の有無，キーワードの順でそれぞれの条件検索式により絞り込みを行ないながら最終的な結果を得る．キーワードの処理については後述する．さらに結果集合を，査読の有無，日英，発行年の順でソートを行なう．

4. 検索結果の表示

   先ほどの結果，詳細情報へのリンク，論文ファイルへのリンクを表示させる．

3.2.1 文字列検索

キーワードに対しては，文字列検索を行なう．文字列検索には，引数に指定された文字列にマッチするかどうかを調べる関数 like() を使用した．部分マッチングをサポートするため，キーワードとして入力された文字列の前後にアスタリスク"*" をつけたもの (入力された語が"keyword" の場合，"*keyword*") を Jasmine で用意されている like 関数の引数として与える．アスタリスクを付加するのには，これも Jasmine で用意されている文字列接続関数 stringCat() を使用した．文字列 abstract に"keyword" という語が含まれるかどうかを調べる場合，ODQL 文は

```
abstract.like("*".stringCat("keyword").stringCat("*"))
```

のように書ける．

3.3 データ登録ページ設計

Web 上からデータベースへのデータ登録を行うテンプレートを作成した．テンプレートの構成内容はデータ入力ページ，入力内容確認ページ，データ再入力ページ，入力データ登録、完了ページの 4 つである．基本的な流れは次のようになる。

1. データ入力ページ
2. 入力確認、良ければ 3 へ。訂正は 4 へ。
3. データ入力完了
4. データの再入力。2 へ。

3.3.1 データ入力ページ

データ入力ページでは登録用のフォームを用意し，ユーザによるデータ入力を受ける．

データ入力後にユーザが登録ボタンを押すことで，次の入力確認ページへと移行する．ページを移行する際，入力された全てのデータは引数として次のページへ引き渡される．

3.3.2 入力確認ページ

入力確認ページでは入力ページで入力されたデータを一覧として出力し，入力内容の確認をユーザに対して行う．このデータベースでは論文のタイトルは必須であるとか、著者の選択で同じ人物を複数個選べないなどの制約がある。これらの制約を確認する必要がある。入力内容に制約を満たしていないものがあれば，その項目にエラー内容を表示する．ここで監視するエラーには以下のようなものがある．

- クラス定義で必須属性に指定されているものは必ずデータを入力してあること
- 日付，ページ数などの項目は必ず数値が入力され，かつ矛盾のない (月は必ず 1 から 12 の値など) データが入力されていること

制約を監視するプログラム例を以下に示す．

```
文献のタイトル"title"は入力必須項目
// 必須属性である title にデータが入力されているかを調べる．
<!IF 'wit_title==""'>
論文名が未入力です
<!DO "ok=1">
//データが入力されていなければエラーメッセージを返し，エラーフラグを立てる
<!/IF>
```

このプログラムは HTML の中に埋め込まれており、この HTML が呼び出されるのと同時に Jasmine 内ではプログラムが実行されている。

3.3.3 再入力ページ

入力確認ページにおいてエラー有りと表示された場合，あるいは入力確認ページで表示された内容に対してユーザが取り消しを選んだ場合表示される。データ再入力後は再び確認ページへ移動する．

3.3.4 入力データ登録ページ

入力データ登録ページは，入力内容にエラーがなくユーザが確認内容に納得したときに表示される．入力データをデータベース内に格納し．データ登録が完了したことをユーザに知らせる．

3.3.5 登録ページでの JavaScript

データ登録をする上でユーザの負担を軽減するため，以下のような工夫を行った．

第 1 に，ある項目の内容に応じて，初期値が決定するような別の項目が存在するような場合，一つめの項目のデータが決定された時点で JavaScript を用いた設定関数を呼び出す．この関数によりもう一つのデータに自動的に初期値を与え，ユーザの入力する必要のある項目数を削減するようにした．

以下は文献の種類を選んだときにそれに応じて初期値を設定するスクリプトである．

```
<select name="kind" onChange="pointCheck()">
//リストボックスの内容が変更したときに関数 pointCheck を呼び出す．
<option value = "0"> 選んでください
<!FOREACH tt t>
<option value="<!REPLACE t.name>" >
<!REPLACE t.name>
<!/FOREACH>
</select>
//javascript を使用した関数"pointCheck"
function pointCheck{
if(document.publication.kind[n].value=="国際論文誌"){
document.publication.is_international.checked =true;
document.publication.is_journal.checked=true;
document.publication.is_refereed.checked=true;
  }
.....
}
```

第 2 に再入力ページは，入力ページ同様にはじめからデータ入力を行うのではなく．前回入力されたデータを引数として受け取り，それを各入力項目の初期値として与えることにより必要な個所を変更するだけで良いようにした．確認ページに INPUT タグですべての値を保持しておき，それを再入力ページに渡す．再入力ページでは INPUT タグの VALUE に値が入るようになる．

4 まとめ

本稿ではオブジェクトデータベースシステム Jasmine を用いた研究室内の情報のデータベース化と設計、内部での処理を開発という観点から述べた．