

RDB と XML とを使った 研究会参加登録システムの実装と評価

安部 智貴[†] 吉田 孝廣[†] 牧之内顕文^{††}

[†]九州大学大学院システム生命科学府 〒 812-8581 福岡県福岡市東区箱崎 6-10-1

^{††}九州大学大学院システム情報科学研究院 〒 812-8581 福岡県福岡市東区箱崎 6-10-1

E-mail: [†]{abetomo,t.yosida}@db.is.kyushu-u.ac.jp, ^{††}akifumi@is.kyushu-u.ac.jp

あらまし オブジェクトリレーショナルデータベース管理システムを用いて、研究会の参加登録システムを開発した。データベースにデータの格納、更新、削除を行う時に XML を用いて、XML 形式のデータをリレーショナルデータベースにマッピングする。このシステムでは、ユーザが XML 対応ブラウザにより、参加登録、登録内容照会、登録内容変更、登録内容削除、パスワード変更を行う。管理者は SQL を用いてデータの管理を行う。本発表では、実装経験と評価について述べる。

キーワード Web 利用技術, XML, e-commerce, リレーショナルデータベース

Implementation and Evaluation of a Workshop Registration System Using XML and RDB - a Case Study -

Tomotaka ABE[†], Takahiro YOSHIDA[†], and Akifumi MAKINOUCHI^{††}

[†] Graduate School of Systems Life Sciences, Kyushu University Hakozaiki 6-10-1, Higashi-ku, Fukuoka-shi, 812-8581 Japan

^{††} Graduate School of Information Science and Electrical Engineering, Kyushu University Hakozaiki 6-10-1, Higashi-ku, Fukuoka-shi, 812-8581 Japan

E-mail: [†]{abetomo,t.yosida}@db.is.kyushu-u.ac.jp, ^{††}akifumi@is.kyushu-u.ac.jp

Abstract We developed a workshop registration system for a technical workshop where over three hundred engineers or researchers are expected to gather. The system has a RDB management system as its kernel. The Interface is Web browser which allows to handle XML. The data inputted via the browser is transmitted to the server and forms a XML file. Then, the system stores the XML data into the RDB. In this paper we report the requirements, design, system architecture, and evaluation.

Key words Web technology, XML, e-commerce, Relational Database

1. はじめに

近年インターネットを使用したサービスが増えてきている。情報検索サービス、Web ページ検索サービスなどがある。それらは全て以前は電話やファックスを用いて行われていた。今回研究会のローカルアレンジメント委員会の仕事の一環として研究会参加登録システムを開発した。本論文では、2 節で研究会参加登録システムの設計要求、3 節で研究会参加登録システムのテーブル設計、4 節で研究会参加登録システムの流れ、5 節で研究会参加登録システムの実装、6 節で研究会参加登録システムの評価、7 節でまとめを述べる。

2. 研究会参加登録システムの設計要求

この章では要求に従い設計したことについて述べる。

研究会参加登録システムでは、複数の一覧表を作成する必要がある。それらの一覧表（あるいは、ファイル）は必要な人が違う。参加者一覧表は、研究会のプログラム委員会関係者が、すでに論文を投稿し採用が決定した発表者が参加しているかを確認するために必要となる。宿泊者一覧表は、ローカルアレンジメント委員会関係者が宿泊人数を把握するためやホテルが部屋割りをを行うために必要となる。支払い金額と送金などの一覧表は、財務担当者が参加登録した人で支払いが終わっていない人に送金を督促するために必要となる。

通常の Web ページファイルでの管理では登録時に HTML あるいは XML ファイルを作成し、そのファイルを読み取り、その中から各種一覧表を作成するプログラムを新たに作る必要がある。しかし、リレーショナルデータベースを使えばデータを格納し SQL を用いて一覧表を作成することが非常に容易である。そのため、このシステムでは中核にリレーショナルデータベースを用いる設計とした。

リレーショナルデータベースを用いる為には、ユーザが参加登録する時どのようなインタフェースを使うかが問題になる。一般的なブラウザを使用せずにユーザインタフェースを作成したとする。すると参加登録したい人にそのインタフェースを各自ダウンロードしてもらい登録作業をしてもらうことになる。これは不特定多数の参加登録者を想定した今回のような場合には不可能ではないが実際的ではない。そこで Web ブラウザを使用した登録方法を用いることにした。ユーザインタフェースは Web ブラウザで表示できる。参加登録したい人は、Web ブラウザと Web に接続できる環境があれば参加登録の為の環境が整う。Web ブラウザでは、HTML と XML [1,2,3] という 2 つの形式を使用できる。

リレーショナルデータベースのデータの処理では、SQL を使用する。しかし、SQL を使用するためには何らかのインタフェースを介する必要がある。インタフェースとして、対話型インタフェース、XML インタフェースなどが挙げられる。このシステムでは、XML インタフェースを利用した。これはどちらも XML を使用することで Web ブラウザからのデータを XML に変換し XML インタフェースを介してリレーショナルデータベースに格納することができる。そのため、リレーショナルデータベースは XML インタフェースを備えたものを選ぶ必要がある。

XML インタフェースを備えたリレーショナルデータベースは複数存在する。XRel [7], XMLPGSQL [8], HITACHI HiRDB version 7 [6] (以下 HiRDB とする) などが挙げられる。XRel, XMLPGSQL は、XML を格納する為のテーブルは始めから決まっており、タプルとして XML のノードの要素、並び、Path などを格納する。HiRDB では、XML を格納する為のテーブルは、どのようなものでもよく、タプルとして、ノードの要素を格納する。つまり、HiRDB ではテーブルの設計と格納などのトランザクションのプログラムの開発を独立して行うことができ、それぞれが終了した後、プログラムからどのようにデータを処理するか決めることができる。そこでリレーショナルデータベースとして HiRDB を選択した。

システムを稼働させる上でポイントとなるのは、ユーザからの問い合わせにローカルアレンジメント責任者が迅速に対応できることとシステムの障害である。たとえば、ユーザが数度変更を行っており、変更をどのように行ったかわからなくなった場合、履歴をとっておくことで対応できると考える。また、企業の人に参加する場合、入金時の振込名義人が会社名のことがある。その場合、管理者が参加者の所属を調べることで対応できるようにした。システムの障害に対しては、リレーショナルデータベースを使用することで対処する。データのバックアッ

プはリレーショナルデータベース付属ツールを使用し 10 分ごとに行う。

3. テーブルの設計

我々は、前節の要求を満たす為の 6 つのテーブルを作成した。

- 登録 (図 1)
- 登録キャンセル (図 2)
- 宿泊関連 (図 3)
- 宿泊関連履歴 (図 4)
- 返金 (図 5)
- 送金履歴 (図 6)

ユーザが登録を行うと登録内容のうち、金銭に関係ないものはテーブル「登録」に、金銭に係るものはテーブル「宿泊関連」に格納する。ユーザが登録した内容の変更を行うと、テーブル「宿泊関連」に格納しているそのユーザのタプルと同じ内容のものをテーブル「宿泊関連履歴」に格納する。格納した後、テーブル「登録」、「宿泊関連」に変更した内容を格納する。ユーザが登録した内容の削除を行うと、テーブル「登録」、「宿泊関連」に格納しているそのユーザのタプルと同じ内容のものをそれぞれテーブル「登録キャンセル」、「宿泊関連履歴」に格納する。ユーザが登録を削除した時点ですでにこちらに送金していた場合、送金されたお金をそのユーザに返金する必要がある。そのため、削除のときに返金に必要な内容を入力してもらい、その内容をテーブル「返金」に格納する。ユーザから指定口座に送金されてきた場合、管理者は銀行からの通知を基に通帳に書かれた内容をテーブル「入金履歴」に格納する。テーブル「登録」、「宿泊関連」、「登録キャンセル」、「宿泊関連履歴」の属性「登録日」、「登録時刻」は登録を行った日時を格納する。テーブル「登録」、「宿泊関連」、「登録キャンセル」の属性「変更日」、「変更時刻」に最後に変更を行った日時を格納する。テーブル「宿泊関連履歴」の属性「変更日」、「変更時刻」には、内容変更、内容削除を行った日時を格納する。

登録内容変更のときにテーブル「宿泊関連履歴」に、登録内容削除のときにテーブル「登録キャンセル」、「宿泊関連履歴」に金銭に係る変更の履歴を残すことで金銭に係る問い合わせに対応できるようにした。これらにより、同一人物の変更履歴を追跡することが可能になる。

4. 処理の流れ

新規登録、登録内容照会、登録内容変更、登録内容削除、パスワード変更の流れは図 7 に示す。四角は画面を表し、矢印はボタン、リンクをクリックした時の流れを表す。

新規登録の流れは次のようになる。ユーザは研究会参加登録システムトップページで「参加登録ページ」をクリックし新規登録画面に進む。新規登録画面で入力フォームに内容を記入し、「登録内容確認画面へ進む」をクリックし、登録内容確認画面に進む。新規登録画面でシステムが意図しない入力があった場合、登録内容確認画面でメッセージを表示する。登録内容確認画面でユーザが入力した内容で変更したいと思った部分を変更する。また、メッセージが表示された部分をメッセージにあわ

```

create table "登録"(
"カナ姓" nvarchar(20) not null,
"カナ名" nvarchar(20) not null,
"漢字姓" nvarchar(20),
"漢字名" nvarchar(20),
"アルファベット姓" varchar(40),
"アルファベット名" varchar(40),
"性別" nchar(1) not null, /* "男", "女" */
"所属" nvarchar(50) not null, /* ネームプレート用 */
"パスワード" varchar(90) not null,
"連絡先" nchar(1) not null,
/* "自" = "自宅", "勤" = "勤務先/所属研究室" */
"連絡先詳細" nvarchar(50),
/* 連絡先で勤務先/所属研究室を選んだ場合のみ記入 */
"郵便番号" char(8) not null, /* 012-3456 */
"住所" nvarchar(75) not null,
"電話番号" varchar(11),
/* 固定電話, または携帯電話 0123456789, 09012345678 */
"FAX 番号" char(10), /* 0123456789 */
"電子メールアドレス" varchar(50) primary key,
"振込元銀行名" nvarchar(25), /* ??銀行??支店 */
"振込予定日" date, /* 2004-09-12 */
"振込名義人" varchar(40),
"領収書請求" integer not null, /* 1 = "要", 0 = "不要" */
"領収書宛名" nvarchar(40),
/* 領収書要の場合のみ 領収書に用いる宛名 */
"領収書項目" integer,
/* 領収書要の場合のみ
1 = "参加費・登録費込", 0 = "参加費・登録費別" */
"請求書請求" integer not null, /* 1 = "要", 0 = "不要" */
"請求書宛名" nvarchar(40),
/* 請求書要の場合のみ 請求書に用いる宛名 */
"請求書項目" integer,
/* 請求書要の場合のみ
1 = "参加費・登録費込", 0 = "参加費・登録費別" */
"入金" nchar(1),
/* "未" = "未済", "差" = "差異", "済" = "済" */
"登録日" date default current date,
/* 2004-09-12 登録を行った日付 */
"登録時刻" time default current time,
/* 20:00:00 登録を行った日付 */
"変更日" date, /* 2004-09-12 一番最新の変更を行った日付 */
"変更時刻" time /* 20:00:00 一番最新の変更を行った日付 */
);

```

図 1 テーブル「登録」

Fig. 1 Table "登録"

せて変更する。変更が終了したら「入力内容を送信する」をクリックする。登録内容確認画面でシステムが意図しない入力があった場合、登録内容確認画面に戻り、メッセージを表示する。そうでない場合、データベースに内容を格納する。格納するとき、テーブル「登録」の主キーである属性「電子メールアドレス」に登録しようとする内容と同じ内容のものがすでに格納されている場合、格納が失敗し、登録内容確認画面に戻り、メッ

```

/* 登録を取り消したタプルだけが入る */
create table "登録キャンセル"(
"カナ姓" nvarchar(20),
"カナ名" nvarchar(20),
"漢字姓" nvarchar(20),
"漢字名" nvarchar(20),
"アルファベット姓" varchar(40),
"アルファベット名" varchar(40),
"性別" nchar(1), /* "男", "女" */
"所属" nvarchar(100), /* ネームプレート用 */
"パスワード" varchar(90),
"連絡先" nchar(1),
/* "自" = "自宅", "勤" = "勤務先/所属研究室" */
"連絡先詳細" nvarchar(50),
"郵便番号" char(8), /* 012-3456 */
"住所" nvarchar(75),
"電話番号" varchar(11), /* 0123456789, 09012345678 */
"FAX 番号" char(10), /* 0123456789 */
"電子メールアドレス" varchar(50) not null,
"振込元銀行名" nvarchar(50), /* ??銀行??支店 */
"振込予定日" date, /* 2004-09-12 */
"振込名義人" varchar(40),
"領収書請求" integer not null, /* 1 = "要", 0 = "不要" */
"領収書宛名" nvarchar(40),
/* 領収書要の場合のみ 領収書に用いる宛名 */
"領収書項目" integer,
/* 領収書要の場合のみ
1 = "参加費・登録費込", 0 = "参加費・登録費別" */
"請求書請求" integer not null,
/* 1 = "要", 0 = "不要" */
"請求書宛名" nvarchar(40),
/* 請求書要の場合のみ 請求書に用いる宛名 */
"請求書項目" integer,
/* 領収書要の場合のみ
1 = "参加費・登録費込", 0 = "参加費・登録費別" */
"入金" nchar(1),
/* "未" = "未済", "差" = "差異", "済" = "済" */
"キャンセル日"
date default current date,
/* 2004-09-12 キャンセルを行った日付 */
"キャンセル時刻" time default current time,
/* 20:00:00 キャンセルを行った時刻 */
"登録日" date, /* 2004-09-12 登録を行った日付 */
"登録時刻" time, /* 20:00:00 登録を行った日付 */
"変更日" date, /* 2004-09-12 一番最新の変更を行った日付 */
"変更時刻" time /* 20:00:00 一番最新の変更を行った日付 */
);

```

図 2 テーブル「登録キャンセル」

Fig. 2 Table "登録キャンセル"

ッセージを表示する。そうでない場合、格納が成功し、登録に使用した電子メールアドレスにメールが届き、参加登録完了画面に進む。参加登録完了画面では、「トップページに戻る」をクリックし研究会参加登録システムトップページに戻るか、「参加登録画面に戻る」をクリックし新規登録画面に戻るか、「登録内

```

/* 宿泊・昼食・懇親会・送迎バス
などお金がかかわるものが入るテーブル */
create table "宿泊関連"(
"カナ姓" nvarchar(20) not null,
"カナ名" nvarchar(20) not null,
"漢字姓" nvarchar(20),
"漢字名" nvarchar(20),
"電子メールアドレス" varchar(50) primary key,
"区分" integer not null,
"会員番号" VARCHAR(10), /* 会員番号 */
"宿泊 2 月 28 日" integer not null,
/* 1 = "宿泊する", 0 = "宿泊しない" */
"宿泊 3 月 1 日" integer not null,
/* 1 = "宿泊する", 0 = "宿泊しない" */
"宿泊日数" integer not null,
"懇親会" integer not null,
/* 1 = "参加する", 0 = "参加しない" */
"昼食 2 月 28 日" integer not null,
/* 1 = "必要", 0 = "不要" */
"昼食 3 月 1 日" integer not null,
/* 1 = "必要", 0 = "不要" */
"昼食 3 月 2 日" integer not null,
/* 1 = "必要", 0 = "不要" */
"昼食数" integer not null,
"バス往路出発地" nchar(1) not null,
"バス復路到着地" nchar(1) not null,
"合計金額" integer not null,
"登録日" date default current date,
/* 初めて登録した時の日付 */
"登録時刻" time default current time,
/* 初めて登録した時の時刻 */
"変更日" date,
/* 一度も変更されないときは NULL 値が入っている */
"変更時刻" time
/* 一度も変更されないときは NULL 値が入っている */
);

```

図3 テーブル「宿泊関連」

Fig.3 Table "宿泊関連"

容を引き続き使用し登録する」をクリックし登録した登録内容をすでに入力した状態の登録内容確認画面に進むか、ができる。ユーザに送られるメールの内容は、パスワードと、登録内容照会、登録内容変更、登録内容削除の URL と、パスワード変更の URL と料金の詳細と入金口座番号である。

登録内容照会、登録内容変更、登録内容削除の流れは次のように処理される。ユーザは研究会参加登録システムトップページで「内容照会ページ」をクリックするか、登録を行ったときのメールに書かれている登録内容照会、登録内容変更、登録内容削除の URL を使用して登録内容照会フォーム画面へ進む。登録内容照会フォーム画面では、新規登録に利用した電子メールアドレスと新規登録完了時に送られてきたメールの中に書かれているパスワードを入力フォームに記入し「ログイン」をクリックする。入力フォームに記入された内容がテーブル「登

```

/* 宿泊・昼食・懇親会・送迎バス
などお金がかかわるものの履歴が入るテーブル */
create table "宿泊関連履歴"(
"カナ姓" nvarchar(20) not null,
"カナ名" nvarchar(20) not null,
"漢字姓" nvarchar(20),
"漢字名" nvarchar(20),
"電子メールアドレス" varchar(50) not null,
"区分" integer not null,
"会員番号" VARCHAR(10), /* 会員番号 */
"宿泊 2 月 28 日" integer not null,
/* 1 = "宿泊する", 0 = "宿泊しない" */
"宿泊 3 月 1 日" integer not null,
/* 1 = "宿泊する", 0 = "宿泊しない" */
"宿泊日数" integer not null,
"懇親会" integer not null,
/* 1 = "参加する", 0 = "参加しない" */
"昼食 2 月 28 日" integer not null,
/* 1 = "必要", 0 = "不要" */
"昼食 3 月 1 日" integer not null,
/* 1 = "必要", 0 = "不要" */
"昼食 3 月 2 日" integer not null,
/* 1 = "必要", 0 = "不要" */
"昼食数" integer not null,
"バス往路出発地" nchar(1) not null,
"バス復路到着地" nchar(1) not null,
"合計金額" integer not null,
"登録日" date default current date,
/* 初めて登録した時の日付 */
"登録時刻" time default current time,
/* 初めて登録した時の時刻 */
"変更日" date,
/* 一度も変更されないときは NULL 値が入っている */
"変更時刻" time,
/* 一度も変更されないときは NULL 値が入っている */
"キャンセル" integer
);

```

図4 テーブル「宿泊関連履歴」

Fig.4 Table "宿泊関連履歴"

録」の属性「電子メールアドレス」と同じものがない場合と同じものがあるが属性「パスワード」が一致しない場合は、登録内容照会フォーム画面に戻りその旨のメッセージ表示する。入力フォームに記入された内容がテーブル「登録」の属性「電子メールアドレス」と同じものがありそのタプルの属性「パスワード」と一致した場合、登録内容照会画面に進む。登録内容照会が目的なら登録内容照会画面を確認し「ログアウト」をクリックしログアウトする。登録内容変更が目的なら「変更する」をクリックし、登録内容変更フォーム画面に進む。登録内容削除が目的なら「削除する」をクリックし、登録内容削除確認画面に進む。登録内容変更フォーム画面では、変更したい内容を記入し「変更確認画面へ進む」をクリックし、登録内容変更確認画面を表示する。システムが意図しない内容があった場合は、登録内容変更確認画面を表示する際にその旨をメッセー

```

create table "返金"(
"カナ姓" nvarchar(20) not null,
"カナ名" nvarchar(20) not null,
"漢字姓" nvarchar(20),
"漢字名" nvarchar(20),
"電子メールアドレス" varchar(50) not null,
"振込先銀行名" nvarchar(50),
/* 記入時は振込, 未記入時手渡し */
"口座種類" nchar(1),
/* "普" = "普通預金", "当" = "当座預金" */
"口座番号" varchar(50), /* 口座番号 */
"口座名" varchar(50), /* 口座名 */
"返金金額" integer not null, /* マイナス金額は返還 */
"返金完了日" date /* お金を返還, 支払い終わった日付 */
);

```

図5 テーブル「返金」

Fig. 5 Table "返金"

```

create table "送金履歴"(
"カナ姓" nvarchar(20) not null,
"カナ名" nvarchar(20) not null,
"漢字姓" nvarchar(20),
"漢字名" nvarchar(20),
"電子メールアドレス" varchar(50) not null,
"振込名義人" varchar(40) not null,
/* 通帳に書かれた振込名義人 */
"振込元銀行名" nvarchar(50),
/* 振込名義人と金額だけで個人を特定できない場合, 銀行に
問い合わせをして振込元銀行名を入れる.
そうでない場合は, NULL 値が入る */
"受領金額" integer not null, /* */
"受領日" date not null, /* 通帳に書いてある日付 */
"受領確認日" date, /* 2004-09-12 管理者が確認した日付 */
"受領確認時刻" time, /* 20:00:00 管理者が確認した時刻 */
"受領確認番号" float
/* 受領日は整数を入れる.
記入ミスが生じた時のみ少数を使用する */
);

```

図6 テーブル「送金履歴」

Fig. 6 Table "送金履歴"

ジ表示する。登録内容変更確認画面で内容を確認，変更し「変更内容を変更する」をクリックする。システムが意図しない内容があった場合は，登録内容変更確認画面に戻りメッセージを表示する。そうでない場合は，登録内容変更終了画面に進む。登録内容変更終了画面では，これ以上作業をする必要がなければ「ログアウトする」をクリックしログアウト画面へ進み変更作業を終了する。登録内容を照会したい場合は「内容照会画面に戻る」をクリックして登録内容照会画面に戻る。さらに登録内容を変更したい場合は，「さらに変更する」をクリックして登録内容変更フォーム画面へ進む。登録内容削除確認画面では，すでに入金した場合は返金用の入力フォームに内容を入力して「削除する」をクリックして登録内容削除確認（振込先確認）画

面へ進む。入金はしていない場合は「削除する」をクリックして登録内容削除終了画面へ進む。登録内容削除確認（振込先確認）画面を表示する時，返金のための内容に不備がある場合は，メッセージを表示する。登録内容削除確認（振込先確認）画面では，返金の為の内容を確認，変更し「削除する」をクリックする。システムが意図しない内容がある場合は登録内容削除確認（振込先確認）画面に戻りメッセージを表示し，そうでない場合は，登録内容削除終了画面に行く。登録内容削除終了画面では，「トップページへ戻る」をクリックしてトップページに戻る。登録内容変更終了画面を表示すると同時にメールが届く。そのメールの内容は，パスワードと，登録内容照会，登録内容変更，登録内容削除の URL と，パスワード変更の URL と料金の詳細である。登録内容削除終了画面では，参加登録が削除された旨が書かれている。

パスワード変更について説明する。ユーザは参加登録完了画面，登録内容変更終了画面が表示される時に送られるメールに書かれているパスワード変更の URL を使用してパスワード変更画面へ進む。パスワード変更画面では，登録に使用した電子メールアドレス，メールに書かれているパスワード，新しいパスワードの3つの内容を書き込む。内容を書き込んだら「パスワード変更」をクリックする。テーブル「登録」の属性「電子メールアドレス」に入力した電子メールアドレスが格納されていない場合，テーブル「登録」の属性「電子メールアドレス」に入力した電子メールアドレスが格納されているが，そのテーブルの属性「パスワード」と入力したパスワードが一致しない場合，あるいはパスワードと新しいパスワードが一致する場合のいずれかの時，新しいパスワードと新しいパスワード（確認用）が一致しない場合はパスワード変更画面に戻りメッセージを表示する。それ以外の場合は，パスワード変更完了画面へ進みパスワード変更が完了する。そのとき同時に新たなメールが送られる。

5. 研究会参加登録システムの実装

我々は，Web サーバ，Java servlet コンテナ，リレーショナルデータベースを用いて研究会参加登録システムを実装した。Java servlet コンテナは Web サーバからデータを受け取りリレーショナルデータベースにデータを格納する為に必要である。環境は以下である。

Web サーバ：Apache 2.0.52 [4]

Java servlet コンテナ：Apache Jakarta Tomcat 5.0.28 [5]

リレーショナルデータベース：HiRDB

Web サーバ用マシン OS：Solaris 9

Java servlet コンテナ，リレーショナルデータベース用マシン OS：Windows 2000 server

実際の画面を図 8, 図 9, 図 10, に示す。

HiRDB における XML 形式のデータの処理法を説明する。XML 形式のデータの処理をするには，次の手順で行う。

- (1) 格納する XML ファイルを作成
- (2) XML データマッピングに必要な次のファイルを作成
 - xms ファイル: マッピングセット定義

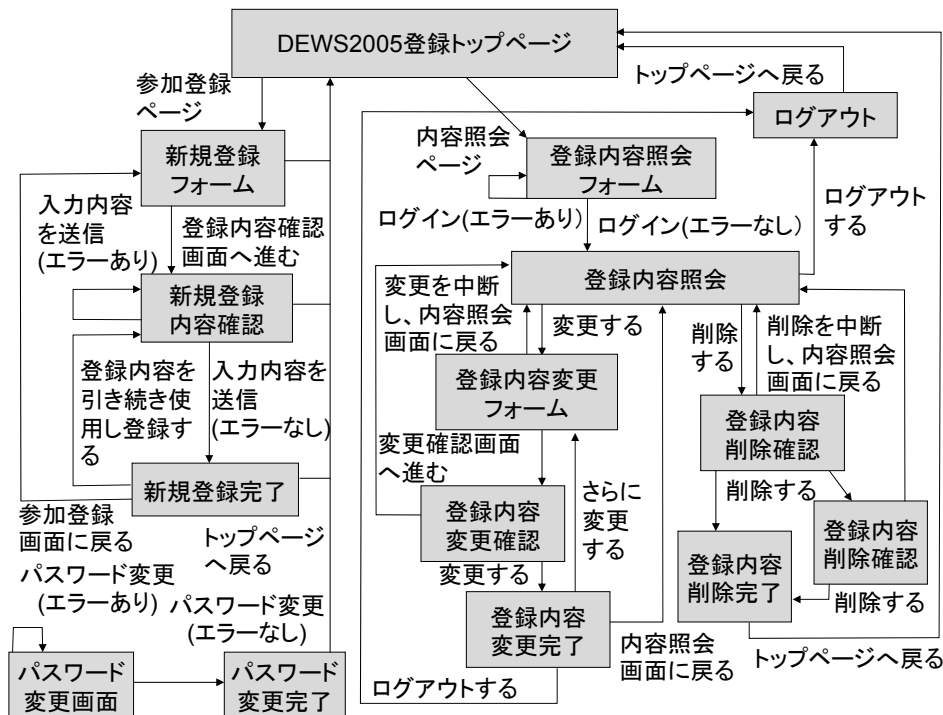


図7 新規登録，登録内容照会，登録内容変更，登録内容削除，パスワード変更の流れ図

Fig. 7 Flow of web pages

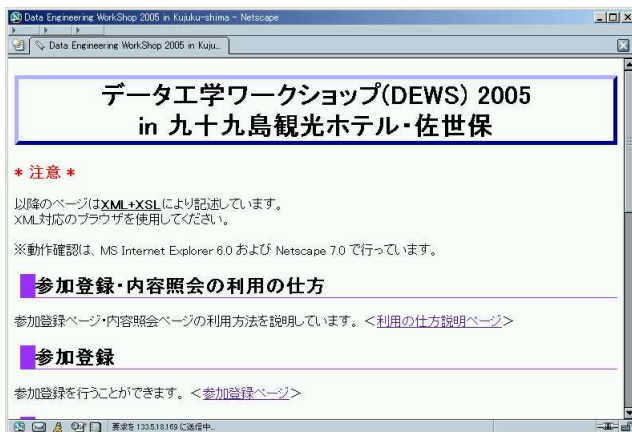


図8 研究会参加登録トップページ

Fig. 8 Toppage of registration

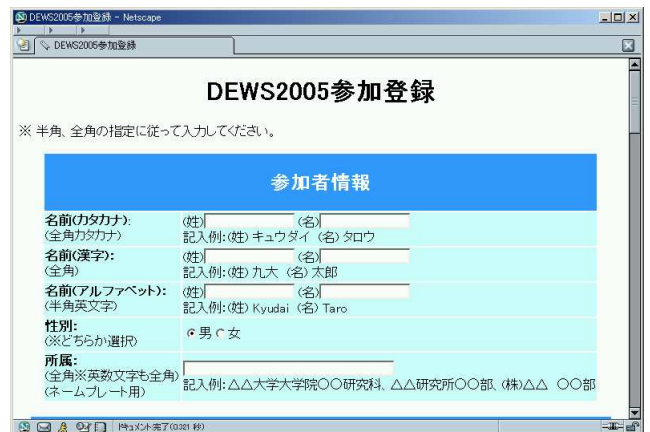


図9 新規登録フォーム画面

Fig. 9 New registration form

- xmp ファイル: マッピング定義
- dtd ファイル: マッピング元 XML タグ定義
- tdf ファイル: マッピング先テーブル定義
- vdf ファイル: ビュー定義

(3) XML を処理する為の Java プログラムを作成

XML データマッピングでは，XML 中のデータをデータベースの各属性に対応付ける．HiRDB では，任意のテーブルの任意の属性に XML データマッピングを行うことが出来る．xms ファイルでは，xmp ファイルのパスを記述する．xmp ファイルでは，dtd ファイルのパス，vdf ファイルのパスを記述する．xmp ファイルで xml のタグとデータベースの属性を結びつけ

る．dtd ファイルでは，XML の形式を記述するもの．tdf ファイルでは，マッピングを行うデータベースのテーブルの定義を書く．vdf ファイルでは，tdf ファイルのパスを記述し，その tdf ファイルで宣言したデータベースのテーブルの属性の中でマッピングしたい属性だけを指定する．複数のテーブルの外部キーを設定する時も vdf ファイルで定義する．

テーブル「返金」に格納する XML ファイル，そのための XML データマッピングに必要な dtd ファイル，vdf ファイル，tdf ファイル，XML ファイルのデータをテーブル「返金」に格納した時のタプルの例を図 11, 12, 13, 14, 表 1 に示す．

表1 XMLファイル格納後のタプル
Table 1 Tuple to be inserted data in database

カナ姓	カナ名	漢字姓	漢字名	電子メールアドレス	振込先銀行名	口座種類	口座番号	口座名	返金金額	返金完了日
アベ	トモタカ	安部	智孝	abetomo@db.is.kyushu-u.ac.jp	福岡銀行 六本松店	普	010101	安部 トモタカ	15150	NULL

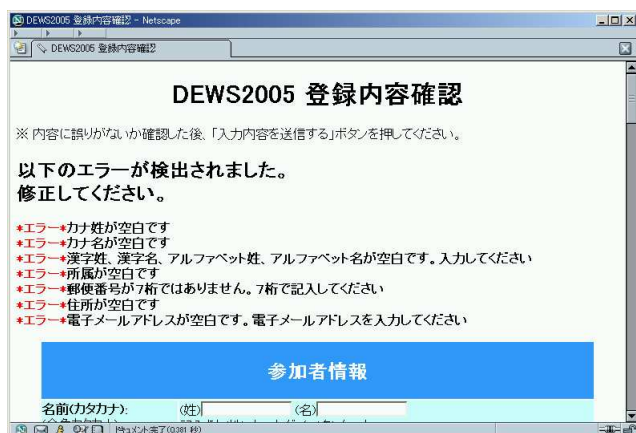


図10 登録内容確認画面 (エラーメッセージ表示)

Fig. 10 Error message form

6. 評価

システム開発は、2人で行った。開発期間は、9月から12月の4ヶ月間である。開発時間は、約256時間である。

負荷テストを2度行った。どちらのテストも30分間で行った。テスト1度目は、研究室内からのみ行った。人数は15人である。テスト2度目は、研究室内、東京、大阪などから行った。人数は、30人である。この2度のテストで本システムは問題なく動作することを確認した。

本システムは2005年1月17日(月)から2月15日(火)17時まで実際に稼働した。これは論文投稿終了後からホテルへの宿泊者数通知までの間である。登録回数、キャンセル回数、変更回数のグラフを図15に示す。採否通知は2月9日(水)のため、参加登録のピークは2月10日(木)前後と考えていた。2月10日(木)は、それまでの稼働日に比べ登録回数が最大であった。しかし、実際のピークは、×切前日と当日であった。2月14日は56、2月15日は60であった。特に、15日午前は30、15日午後は30であった。

7. まとめ

我々は、研究会参加登録システムの設計、実装を行った。要求に従い、種々トラブル回避を考慮しシステムを開発した。実際にテストを行ったところ、問題なく動作した。1月17日から約一ヶ月間稼働させ評価する予定である。評価結果は研究会で報告する予定である。

謝辞 本論文作成にあたり、「日立 HiRDB アカデミック支援プログラム」に参加させていただいた株式会社日立製作所様に感謝致します。また、実装したシステムのテストにご協力くださいました皆様に感謝致します。本研究の一部は、日本学術

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE 参加登録
SYSTEM "../registration_returnmoney.dtd">
<参加登録>
<参加者情報>
<個人>
<名前>
<カタカナ>
<カナ姓>アベ</カナ姓>
<カナ名>トモタカ</カナ名>
</カタカナ>
<漢字>
<漢字姓>安部</漢字姓>
<漢字名>智孝</漢字名>
</漢字>
</名前>
</個人>
<電子メールアドレス>abetomo@db.is.kyushu-u.ac.jp
</電子メールアドレス>
</参加者情報>
<返金>
<振込先銀行名>福岡銀行六本松店</振込先銀行名>
<口座種類>普</口座種類>
<口座番号>010101</口座番号>
<口座名>安部トモタカ</口座名>
<返金金額>15150</返金金額>
</返金>
</参加登録>
```

図11 格納するXMLファイル

Fig. 11 XML file data for the tuple to be inserted

振興会科学研究費補助金課題番号 15650017,(A)(2)16200005 による。

文献

- [1] World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (Third Edition), <http://www.w3.org/TR/2004/REC-xml-20040204/>, (2004)
- [2] World Wide Web Consortium: Extensible Stylesheet Language (XSL) Version 1.1, <http://www.w3.org/TR/2004/WD-xsl11-20041216/>, (2004)
- [3] World Wide Web Consortium: XSL Transformations (XSLT) Version 1.0, <http://www.w3.org/TR/1999/REC-xslt-19991116/>, (1999)
- [4] The Apache Software Foundation, <http://www.apache.org/>
- [5] The Apache Jakarta Project, <http://jakarta.apache.org/>
- [6] 株式会社日立製作所: HITACHI HiRDB version 7, <http://www.hitachi.co.jp/Prod/comp/soft1/hirdb/>
- [7] Masatoshi Yoshikawa, Toshiyuki Amagasa, Takeyuki Shimura and Shunsuke Uemura: "XRel: A Path-Based Approach to Storage and Retrieval of XML Documents using Relational Databases", ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 110-141, August 2001

```

<?xml version="1.0" encoding="Shift_JIS"?>
<!ELEMENT 参加登録 (参加者情報, 返金) >
<!ELEMENT 参加者情報 (個人, 電子メールアドレス) >
<!ELEMENT 個人 (名前) >
<!ELEMENT 名前 (カタカナ, 漢字) >
<!ELEMENT カタカナ (カナ姓, カナ名) >
<!ELEMENT カナ姓 (#PCDATA) >
<!ELEMENT カナ名 (#PCDATA) >
<!ELEMENT 漢字 (漢字姓, 漢字名) >
<!ELEMENT 漢字姓 (#PCDATA) >
<!ELEMENT 漢字名 (#PCDATA) >
<!ELEMENT 電子メールアドレス (#PCDATA) >
<!ELEMENT 返金 (振込先銀行名, 口座種類,
口座番号, 口座名, 返金金額) >
<!ELEMENT 振込先銀行名 (#PCDATA) >
<!ELEMENT 口座種類 (#PCDATA) >
<!ELEMENT 口座番号 (#PCDATA) >
<!ELEMENT 口座名 (#PCDATA) >
<!ELEMENT 返金金額 (#PCDATA) >

```

図 12 XML データマッピング用の DTD ファイル
Fig. 12 DTD file to mapped data in database

```

<?xml version="1.0" encoding="Shift_JIS"?>
< viewdef tableDef="registration_returnmoney.tdf">
< entityList >
< entity name="返金">
< column name="カナ姓"/><!-- カナ名 -->
< column name="カナ名"/><!-- カナ姓 -->
< column name="漢字姓"/><!-- 漢字名 -->
< column name="漢字名"/><!-- 漢字姓 -->
< column name="電子メールアドレス"
isPrimaryKey="yes"/>
< column name="振込先銀行名"/>
< column name="口座種類"/>
< column name="口座番号"/>
< column name="口座名"/>
< column name="返金金額"/>
</entity >
</entityList >
< relationList >
< table name="返金">
</table >
</relationList >
</viewdef >

```

図 13 XML データマッピング用の VDF ファイル
Fig. 13 VDF file to mapped data in database

```

<?xml version="1.0" encoding="Shift_JIS"?>
< database dbms="HiRDB" version="07-00"
dbUrl="http://www.hitachi.co.jp/db/
delete/registration_returnmoney">
< table name="返金">
< column name="カナ名" type="NVARCHAR(20)"/>
< column name="カナ姓" type="NVARCHAR(20)"/>
< column name="漢字姓" type="NVARCHAR(20)"/>
< column name="漢字名" type="NVARCHAR(20)"/>
< column name="電子メールアドレス"
type="VARCHAR(50)"/>
< column name="振込先銀行名"
type="NVARCHAR(50)"/>
< column name="口座種類" type="NCHAR(50)"/>
< column name="口座番号" type="VARCHAR(50)"/>
< column name="口座名" type="VARCHAR(50)"/>
< column name="返金金額" type="integer"/>
< column name="返金完了日" type="date"/>
</table >
</database >

```

図 14 XML データマッピング用の TDF ファイル
Fig. 14 TDF file to mapped data in database

- [8] 油井 誠, 森嶋 厚行: PostgreSQL を用いた多機能な XML データベース環境の構築, 情報処理学会論文誌: データベース, Vol. 44, No. SIG 12(TOD 19), pp. 11-22, Sep. 2003

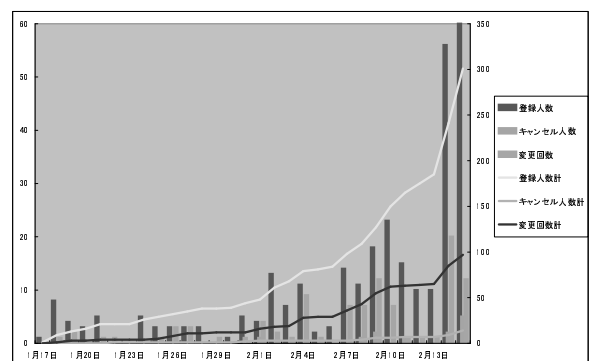


図 15 登録回数, キャンセル回数, 変更回数のグラフ
Fig. 15 Error message form