

# 「こみゅぽ」

-青山学院大学生のためのコミュニティポータルサイト-

報告書

2000年12月1日

青山学院大学国際政治経済学部

井田ゼミナール4期生

指導 井田昌之

<http://www.sipeb.aoyama.ac.jp/~mi-home/>

Copyright © 2000 Ida-Lab, SIPEB, Aoyama Gakuin University. All rights reserved.

はじめに...

2000年6月。

井田ゼミナール4期生の2000年度プロジェクトテーマとして、  
井田昌之教授より提示されたものが、

「コミュニティーポータル」。

そこで、私たちは「こみゅぽ」を考えた。

そこへ行けば...

見たい情報がいつでも見られて、  
会いたい人にいつでも会える。

そんな“夢”を実現するために...!!

**Java, Linux, Servlet ...?**

知らないことはいろいろあるけれど...  
とりあえず、やってみよう。

やってみたいから。

そしてもっともっと、みんながコミュニケーションを取れるように...

## 【参加者一覧】

青山学院大学国際政治経済学部  
井田ゼミナール4期生(2000年度3年生)

Arakane Keiko 荒金 啓子	国際経営学科	<a href="mailto:keiko@noa.sipeb.aoyama.ac.jp">keiko@noa.sipeb.aoyama.ac.jp</a>
Onishi Jun 大西 淳	国際経営学科	<a href="mailto:jon@noa.sipeb.aoyama.ac.jp">jon@noa.sipeb.aoyama.ac.jp</a>
Okada Makio 岡田 牧男	国際経営学科	<a href="mailto:macci75@noa.sipeb.aoyama.ac.jp">macci75@noa.sipeb.aoyama.ac.jp</a>
Kang Suijin 姜 秀珍	国際経営学科	<a href="mailto:sussy@noa.sipeb.aoyama.ac.jp">sussy@noa.sipeb.aoyama.ac.jp</a>
Kuroda Hisoshi 黒田 洋史	国際経営学科	<a href="mailto:cafe_kuro@noa.sipeb.aoyama.ac.jp">cafe_kuro@noa.sipeb.aoyama.ac.jp</a>
Shimizu Jun 清水 淳	国際経済学科	<a href="mailto:aqua@noa.sipeb.aoyama.ac.jp">aqua@noa.sipeb.aoyama.ac.jp</a>
Dobashi Ritsuko 土橋 りつこ	国際政治学科	<a href="mailto:ritz@noa.sipeb.aoyama.ac.jp">ritz@noa.sipeb.aoyama.ac.jp</a>
Mori Kenzo 森 健蔵	国際経営学科	<a href="mailto:moriken@noa.sipeb.aoyama.ac.jp">moriken@noa.sipeb.aoyama.ac.jp</a>
Yajima Tatsunari 矢島 達也	国際経済学科	<a href="mailto:arrow@noa.sipeb.aoyama.ac.jp">arrow@noa.sipeb.aoyama.ac.jp</a>
Yamazaki Junko 山崎 淳子	国際政治学科	<a href="mailto:junko@noa.sipeb.aoyama.ac.jp">junko@noa.sipeb.aoyama.ac.jp</a>

《協力を仰いだ方々》

Igarashi Hirotsuka  
五十嵐 浩隆

青山学院大学大学院国際コミュニケーション専攻  
[hiro@sipeb.aoyama.ac.jp](mailto:hiro@sipeb.aoyama.ac.jp)

Ida Yukino  
井田 雪乃

青山学院大学大学院国際コミュニケーション専攻  
[ida-y@sipeb.aoyama.ac.jp](mailto:ida-y@sipeb.aoyama.ac.jp)

Oshima Masatsugu  
大島 正嗣

青山学院大学国際政治経済学部助手  
[osh@sipeb.aoyama.ac.jp](mailto:osh@sipeb.aoyama.ac.jp)

—担当教授—

Ida Masayuki  
井田 昌之 教授

青山学院大学国際政治経済学部教授・工学博士  
[ida@sipeb.aoyama.ac.jp](mailto:ida@sipeb.aoyama.ac.jp)

---

「青山学院生大学生のためのコミュニティーポータルサイト～「こみゅぽ」～」  
報告書作成スタッフ

報告書編集・総責任：大西淳

印刷・製本責任者：矢島達也

表紙・デザイン：土橋りつこ、清水淳



## 【謝辞】

今回のプロジェクト「こみゅぽ」を進めるにあたり下記の企業、団体の方々から多大なるご協力を頂きました。青山学院大学事務システム室ならびに、アンケート調査を実施いただいた青山学院大学生の方々に協力をしていただきました。

株式会社東芝

株式会社オレンジソフト

アルカテル・インターネットワーキング株式会社

日本電気株式会社

株式会社日立製作所

日立電線株式会社

(敬称省略・順不同)

青山学院大学国際政治経済学部井田研究室 4 期生一同、誠に感謝いたします。  
ご協力ありがとうございました。

---

全ての内容およびデータは私たち井田ゼミ 4 期生が誠意をもって調査、研究した結果ではありますが、主観に基づくものであって、その結論は製品・サービス固有の価値を規定しようとするものではないこと、またいかなる製品・サービスについても悪意がないことをご了承ください。

---

# 目次

第1章	プロローグ	p.1~13
第2章	コミュニティポータルそしてこみゅぽ	p.4~15
第3章	サーバ	p.16~30
第4章	ツール	p.31~119
第5章	Web デザイン	p.120~136
第6章	携帯電話端末	p.137~156
第7章	青山祭	p.157~181
第8章	アンケート	p.182~188
第9章	将来のこみゅぽ	p.189~191
第10章	借用機器	p.192~197
第11章	プロジェクト後記	p.198~224
第12章	付録	p.225~235
第13章	参考文献・URL	p.235~240



# 第1章 プロローグ

## 1-1【プロローグ】

＜＜文責：山崎淳子

### 1. プロジェクト？

2000年6月。井田教授から私たちに提示されたのは、2000年度プロジェクトのテーマ、「コミュニティーポータル」だった。まず、私たちの反応は、「え？」。以前どこかで聞いたことのあるような、ないような、どうにも微妙な響きだ。私たちが現時点でゼミ内のコミュニケーションツールとして使っている「intranets.co.jp」。これとは違うのか？

私たちは、まず「コミュニティーポータル」という言葉の意味を探り、私たちにりの概念をつけようと考えた。「コミュニティーポータル」とは何ぞや？

私たちは、誰もが決してたった独りで孤独に生きているわけではない。どんなに自分が孤独だと思っている人だって、何らかの親類、友達、知り合い、属している環境などがあるはずだ。（ないと思っている人だって、日本に住んでいる地球人としての属性を持っているじゃないか！？）そして、その属性（ここではコミュニティーと呼ぼう）が単一であることは少ない。もし会社に行っているお父さんだったら、勤めている会社の従業員であると共に、家族というコミュニティーの一員であって、そして夫婦というカップルのなかの一人なのである。それまたゴルフクラブの会員だったり、ある市町村の市民町民村民の一人だったりする。そう考えていくと、お父さんは大小さまざまな複数のコミュニティーに属していて、かつプライベートな家族を構成しており、また自分という一人の人間として、自分だけのシステム手帳なんかを持っていたりするのである。

そんな一人の人間が関わる、複数の重層化されたコミュニティー。その種類は一人一人によって全く異なる。

それを、自分だけのページに、一気に表示できて、かつどのコミュニティーとも瞬時にコンタクトを取れる、つながれる。それが「コミュニティーポータル」ではないか…？

そこで私たちの考えた「コミュニティーポータル」とは、いつでもどこでも、ブラウザさえあればそこへアクセスできて、また、そこへ行けば、自分の関わるコミュニティーの知りたい情報が何でもGETできて、かつ自分だけの個人としてのスケジュールも管理できるという概念だった。そんな「コミュニティーポータル」なんて、あったらいいよね。

### 2. 「こみゅぽ」を作ろう

そのサイトへ行けば、見たい情報がいつでも見られて、コミュニティー内の会いたい人といつでもつながれて、かつ自分ひとりの、個人的な事柄も管理できる…そんな夢のようなサイトを作ろう、作りたい。

そこで私たちは、これから実現へ向けて試みる、青学生向けコミュニティーポータルサイトを、「こみゅぽ」と名付けた。

ここで、青学生向けと対象を限定したのは、理由がある。万人が、どんなコミュニティーポータルサイトを求めているのかなんて推測するには、私たちはあまりに小さすぎる。私たちが普段生きている環境、つまり“学生”としての立場で、自分たちにあつたらいいもの、便利なものは何なのか？これを見つける方が非常に具体的で手っ取り早かった。そして何よりも、私たちが今、学生なのだから、学生の求めるものや、学生の日常生活に一番精通していると考えたからだ。

「こみゅぽ」に盛り込みたいものは何だろう？

青山学院大学の一学生として考えたら、休講情報や授業情報、取得単位計算システムなんてあつたらいいだろうな。ついでに、青学周辺のおいしいお店や飲み屋、喫茶店なんかも知りたい…。

次に、井田ゼミの一員として考えたら何が欲しいだろう？今、あの人はどこにいたかが、TELする前に分かれば便利だなあ。いつ誰が空き時間でヒマしているのかも知りたい。来週のゼミ情報も分かればいい。

個人的に予定を組むときはどうしたらいいだろう？テストの予定はゼミのメンバーに知られてもいいけれど、デートの約束は知られたくないな。自分の好きなサイトもリンクに張っておきたい。自分の趣味で行きたいサイトなんかもこのページから検索して行ければ便利なこと！

…そんな風に、盛り込みたいものは山ほどあつた。しかし、果たして、自分達にそんな高度なモノは作れるのだろうか？

### 3. Java、Linux、Servlet…！？！？

膨らむ夢の実現には、Javaの技術が欠かせない。そして、できた「こみゅぽ」を各々のPCからアクセスして、誰か管理者が管理するには、ネットワーキングやサーバの話も他人事ではなくなってくる。一体どうすればいいのだろうか？

前期の約3ヶ月間。井田ゼミでやってきた学習はどれも、この「こみゅぽ」作りに役立ちそうだけれど、サラッと齧<sup>かじ</sup>って、ちょこっと頭に入っているだけ。本当に使いこなしてやっていけるのだろうか？

私たちは、それぞれが思い思いの不安を抱えて、より知識を深め、スキルアップを図るために、勉強に励み、情報を集めることから始めた。

#### 4. アンケート

情報を集めるために、ツールや技術的な学習の情報なら雑誌やテキストで勉強できる。でも、私たち以外の学生は、PC のこと、インターネットのこと、コミュニティのこと、そしてポータルサイトのことをどう思っているのだろうか？また、実際に出来あがる「こみゅぽ」は果たして、利用者にどんな感想を持ってもらえるのだろうか？

当事者の私たちだけでは分からないこと、<sup>ちうかん</sup>鳥瞰しなければ見えないもの、比較されなければ気付かないことがいっぱいある。それらはどれも、「こみゅぽ」をもっともっと良くするために、役立つものであるはずだ。そして何より、私たちは、何も、この「こみゅぽ」を自己満足だけで完成させるのではない。現代を生きる学生、そして人々が、もっと便利にもっと楽しく、もっと良く日常を送って、周囲の人とコミュニケーションをうまくとれるように作るものである。

そう考えた私たちは、「こみゅぽ」作成前にアンケート第1弾として、そして実際に青祭で「こみゅぽ」を使ってもらった後にアンケート第2弾として、合計2回のアンケートを実施した。

この報告書は、上記のような“夢”を描いた私たちが、“夢”を実現させるため、悪戦苦闘し、泣き、笑いした日々の記録と、結果すべてである。

## 第2章

### コミュニティポータル&こみゅぽ

2-1 【コミュニティポータル】	p. 5～
2-2 【こみゅぽ】	p. 9～
2-3 【道のり 2000】	p. 13～

## 第2章 コミュニティポータル&こみゅぽ

### 2-1【コミュニティポータル】

＜＜文責：荒金啓子

#### ◎コミュニティポータルの定義

##### ◆言葉の見直しから◆

「コミュニティポータル」という今まで見たことも聞いたこともない言葉を目の前にして、まずそれを、ひとつひとつの言葉に分解して見なおすことからはじめました。

- コミュニティー： 共通の興味、意識、利益、場所を持つ人の集まり
- ポータル： 入り口、表玄関
- ポータルサイト：
  - World Wide Web への入り口という役目
  - ブラウザを立ち上げた時、初めに表示されるサイト
  - Web 上の必要な情報へつながっていく役目を果たす

##### ◆コミュニティポータルの定義◆

それぞれの言葉、そして「ポータルサイト」がどのようなものであるかをふまえて、私たちの考える「コミュニティポータル」を以下のように定義しています。

- あるコミュニティに関わる情報へつながっていく役目を果たす
- あるコミュニティ内での情報共有、意見交換、意思決定を円滑にする役目を果たす
- あるコミュニティの特定情報に対するアクセス・追加・変更の権利の制御ができる
- 複数コミュニティへのアクセスの入り口としての役目を果たす

#### ◎コミュニティポータルの構造 — COMMUNITY IN COMMUNITY

##### ◆対象とするコミュニティの特徴◆

私たちはコミュニティという言葉の定義である「人の集まり」の特徴のひとつとして、「大きな人の集まりには、その中で自然と複数の小さな人の集まりが存在することが多い」という点に着目しました。そして、個人は大きなコミュニティに属しつつ、その内部にある小さなコミュニティも活動単位としているという複数の立場にあることが考えられます。私たちが提案するコミュニティポータルはまず、このような特徴を持ったコミュニティを対象とすることにしました。



### ◆コミュニティをリオーガナイズ◆

前記のような特徴を持つコミュニティでは、コミュニティポータルを提供を小さなコミュニティごとではなく、それら小さなコミュニティを包括する大きなコミュニティごとに行うことを考えました。個人は大きなコミュニティにアクセスし、そこからその下位にある小さなコミュニティへ同時にアクセスするという仕組みです。この方法を採用することによって、散在する小さなコミュニティにそれぞれアクセスする煩わしさを排除し、大きなコミュニティに属する自分として必要な情報と、下位にある小さなコミュニティに属する自分として必要な情報を効率的に得ることが実現できると考えたのです。

### ◆Community in Community というコミュニティ・モデル◆

私たちの提案したコミュニティ・モデルは以下の図 2-1 のようなコミュニティの中にコミュニティが存在するという二重構造になっています。

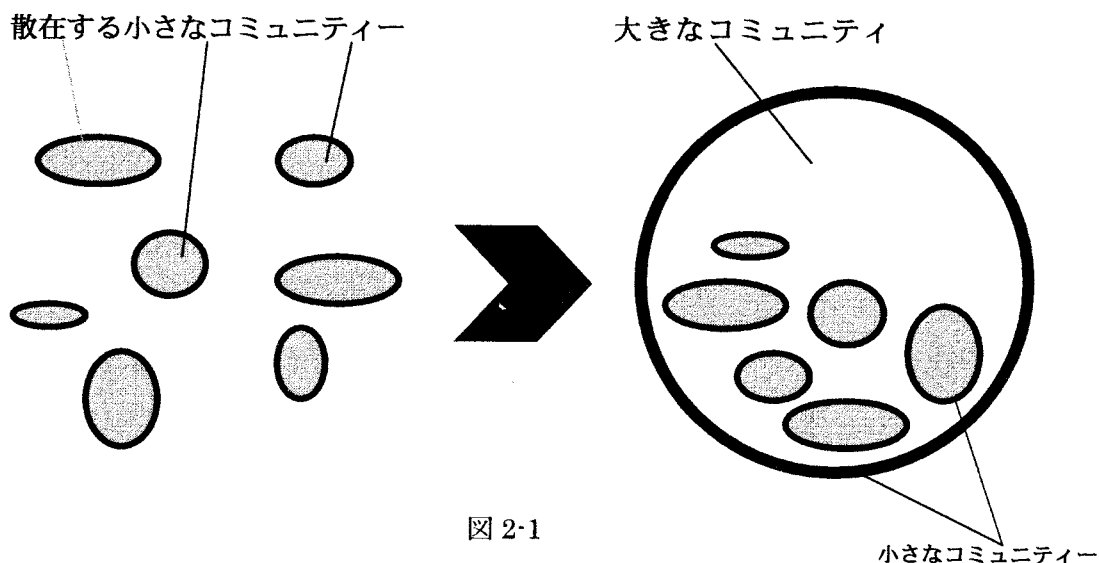


図 2-1

コミュニティポータルではまず大きなコミュニティのサイトにアクセスし、そこからリンクの張られている小さなコミュニティのサイトへと移動できるようにします。

### ◎ユーザの識別 - 必要性和メリット

#### ◆認証の必要性◆

コミュニティポータルでは、ログインの手続き（認証）によって、ユーザを識別する仕組みを持たせます。サイトにアクセスする人は下記の 3 つに分けられます。

- 大きなコミュニティにも属さない人
- 大きなコミュニティには属すが小さなコミュニティにはどこにも属さない人
- 大きなコミュニティと小さなコミュニティに属す人

これらの人々に対し各コミュニティ（大きなコミュニティも含め）ではパブリックにしたい情報とそうでない情報についてアクセスを制御することがコミュニティポータルにおいて必須となることは当然だと考えます。

#### ◆認証は一度だけ◆

ユーザはログイン後、まず大きなコミュニティのサイトにアクセスすることになります。その段階でユーザは確定されているため、小さなコミュニティのサイトにアクセスする時点では再度の認証を必要としません。

#### ◆認証のメリット◆

##### ➤ 情報の追加・変更の許可

ユーザにより特定の情報に対し閲覧だけを許可して、追加・変更を許可しません。

具体例：掲示板

大きなコミュニティに属さない人でも、大きなコミュニティで共有する掲示板の内容を閲覧することは可能ですが、投稿や削除は登録されてログインを済ませたユーザのみ可能となるようにします。

##### ➤ 個人情報の利用

コミュニティポータルに個人情報がある場合、本人のみにアクセスを許可します。

具体例：GPA の確認

GPA の計算には単位数だけではなく成績も必要です。このような個人情報にユーザ本人のみがアクセスできるようにします。

##### ➤ 同じ情報の異なる見え方

閲覧するユーザにより同じ個所の内容を異なる表示にします。

具体例：スケジュール

複数のコミュニティに属す A さんの土曜日の 7:00PM のスケジュールがあるコミュニティでは「デート」であることがわかる一方、異なるコミュニティでは「用事」という表示のみになるようにします。

### ◎Anytime, Anywhere, with Any Device

コミュニティ内でやりとりされる情報の特徴上、コミュニティポータルが様々な端末でアクセスできることを重要と考えます。

### ◆コミュニティポータルにある情報の特徴◆

コミュニティ内でやりとりされる情報には、例えばどのようなものがあるでしょう。

私たちがコミュニティで欲しい情報は

「今、彼はどこで何をしているの？」

「今、彼女の金曜日のスケジュールが知りたい！」

「今、コミュニティ全体のスケジュールを確認したい！」

私たちがコミュニティに伝えたい情報は

「今、直帰で家に向かっています。」

「今、学食に集まってください。」

コミュニティポータルにある情報の特徴として、以下の2つが挙げられます。

- リアルタイムで変化—常に情報が更新されていきます
- 受・発信の時と場の多様性—コミュニケーションはPCの前だけではありません

### ◆様々な端末◆

ネットサービスを受けられる端末には、どのようなものが考えられるでしょう。

- 携帯電話・PHS
- PDA (パームトップ)
- ラップトップパソコン
- テレビ
- ゲーム機
- 固定電話
- ウェアラブルコンピュータ

コミュニティポータルの定義のひとつにある「あるコミュニティ内での情報共有、意見交換、意思決定を円滑にする役目を果たす」を実現するには、必要な時に、自分のいる場所で情報の受・発信が可能であることが大切です。

異なる端末において本当に Anytime, Anywhere, with Any Device を実現するには、コミュニティポータルをプロデュースすること以外にも様々なプレーヤーの協力が必要になります。なぜなら、端末の OS を選ばない、表示方法を選ばない、通信手段を選ばない、通信速度を早くする、といった要素が絡んでくるからです。

## 2-2【こみゅぽ】

〈文責：荒金啓子

「こみゅぽ」は青山学院大学の学生を対象として、私たちの提案したコミュニティポータルの構想を実験的に適用したサイトです。

### ◆なぜ青学生なのか◆

ゼミ生として、私たちも青山学院大学という大きなコミュニティに内在するゼミナールという小さなコミュニティの1つに所属しています。私たちは、まず私たち自身が一番わかっているコミュニティについてコミュニティポータルを作ることがベストであると考えました。

### ◆その名は・・・？◆

サイト名は文字通りコミュニティポータルの略称ですが、その名の由来はプロジェクトについて話し合う中で、毎回「コミュニティポータル」と呼ぶことが面倒になり「こみゅぽ」という略称で呼びはじめたことに端を発します。「こみゅぽ」とひらがな・カタカナを使用する目的は、表記する際に、その名が二つの言葉から成ることをわかり易くすることにあります。

### ◆3つのわたし◆

青学生は大学というコミュニティの中でも様々な側面を持つと考えられます。こみゅぽでは、それらの側面を大きく3つに分類しました。3つのわたしとして捉えることはこみゅぽに盛り込むツールと、サイトのデザイン・構造を検討する上で役に立ちました。そして、それがユーザに使い易いこみゅぽをつくることにつながると考えました。

### ◆こみゅぽのツール◆

こみゅぽに盛り込まれるツールは以下のステップを踏んで検討されています。

- ①「3つのわたし」に分類する
- ②「こまった」を挙げる
- ③「あったらいいな」を挙げる

一人の青学生を「3つのわたし」に分けることで、それぞれの立場において、現在どんなことを不便・不満に感じているかを挙げていきました。次に挙げるのはその抜粋です。そして、それらの立場ごとにこみゅぽに求めるツールのリスト「Wish List」を作成しました。

(ツールの詳細については第4章の「ツール」を参照してください)

## 青学生なわたし

### 青学生のこまった

- 休講情報は9号館に行かないとわからない？
- あの授業ってみんなどう思ってるんだろう？
- あの本がまだ借りられるの?? - 図書館いかないとわかんないジャン -
- 3限みんな休講! - どこでLunch 食べよっか??
- このテキストはもう読まないだろうなあー
- ぱんきょう (般教)、何とったらいいんだろう？

## ゼミ生・サークルメンバーe.t.c.なわたし

### ゼミ生・サークル生のこまった

- 明日のミーティングどの教室借りてやろう？
- 飲み会決定!サークルのみんなにお知らせ!
- メンバーに年賀状書こうっと。
- 緊急会議!学校にいるやつ集まれ!
- 次のミーティングの日時って誰が決めるの?
- ○○クンは水曜日授業あるんだっけ?
- 新入生の自己紹介があると楽しい~
- 情報が更新されたときだけチェックしたい

## わたしなわたし

### わたしのこまった

- かさばるスケジュール帳は、もう持ち歩きたくない
- 新学期前、確実な取得単位をすぐ確かめたい!
- あっ!フロッピー忘れた。。家のPCにあるワードをプリントアウトしたいのに!
- イチイチ違う検索サイトにいくのって。。もっといい方法ないのかなあ
- 新学期!教室番号全部覚えるのって面倒。。
- どこでもメールをすぐチェックできたらなあ
- 使わなくなったものをリサイクルしたい

### ◆こみゅポの構造◆

こみゅポでは、コミュニティポータルの特徴である Community in Community という図 2-2 のような二重構造のコミュニティ・モデルの適用を試みています。そして、前述の「3つのわたし」という捉え方と二重構造のコミュニティ・モデルを組み合わせることによってこみゅポの構造ができています。

「青学という大きなコミュニティが、内在する小さなコミュニティを包含する」



図 2-2

### ◆こみゅポツアー◆

こみゅポでユーザがサイトを利用する際の大まかな流れを、図 2-3 を利用し紹介します。

(ツールの詳細は第 4 章の「ツール」を参照してください)

- ①こみゅポのサイトにアクセス
- ②ログイン
- ③「青学生なわたし」のためのページ
- ④リモコンを表示する  
(「わたしなわたし」のためのツール+小さなコミュニティへのリンク)
- ⑤小さなコミュニティのためのページ

ユーザはログインの手続きをとると、認証前と体裁のほとんど同じページを表示します。そのページでは、どのユーザにも同じように用意された「青学生なわたし」として利用するツールがあります。そのページの中に「リモコン」のボタンがあり、それをクリックすることによってリモコンが表示されます。リモコンには、「わたしなわたし」として使うツール、そして既にそのユーザが登録されている小さなコミュニティー、例えば「ゼミ生なわたし」としてのページへのリンクが用意されています。リモコンではまた、現存する他のコミュニティーの検索、小さなコミュニティーのページの追加、現存するコミュニティーへ登録のためのツールがあります。

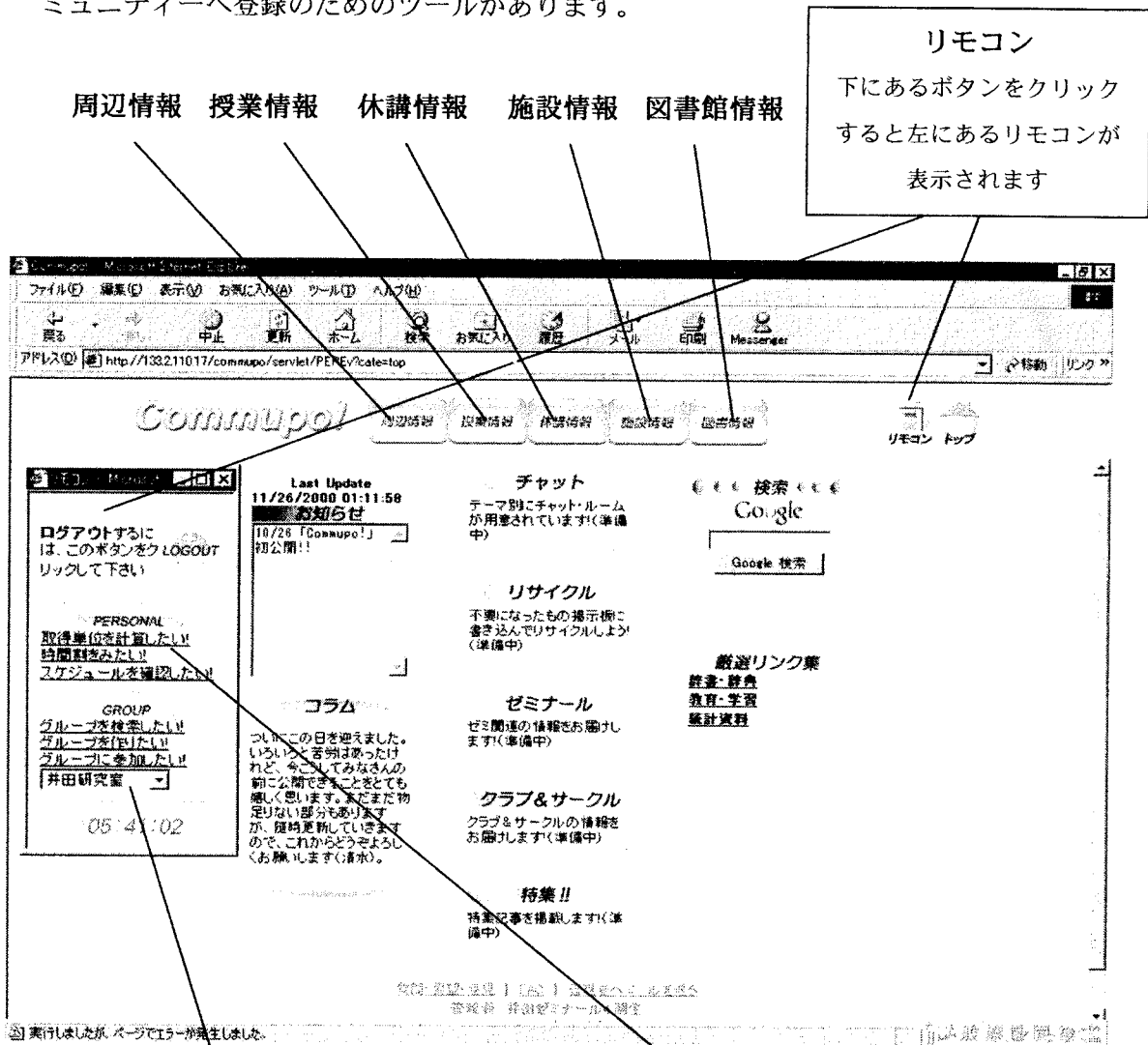


図 2-3

リモコンのプルダウンメニューからコミュニティーを選択すると、小さなコミュニティーのページが新しいウィンドウで開きます。以下の図 2-4 はリモコンから小さなコミュニティーのページにアクセスしたときの例です。

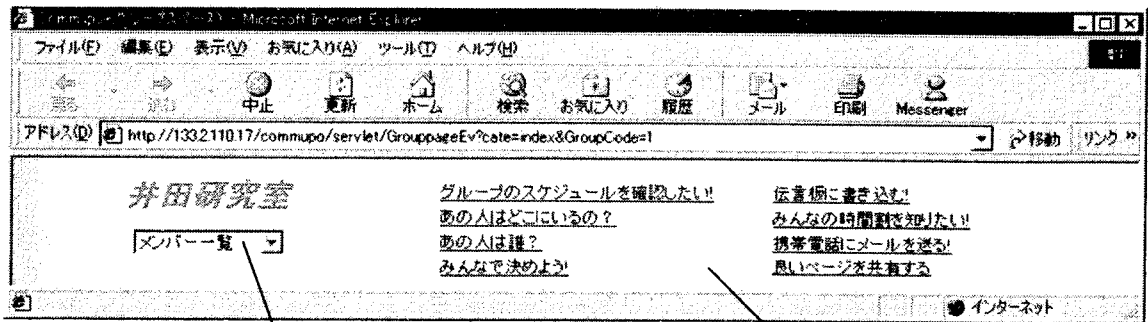


図 2-4

**メンバー一覧**  
 小さなコミュニティーに登録されているメンバーが  
 プルダウンメニューに表示されます

**ゼミ生なわたしが使うツール**

- ①グループスケジュール
- ②あの人ファインダー
- ③メンバープロフィール
- ④投票システム
- ⑤グループ掲示板
- ⑥メンバーの時間割
- ⑦携帯端末へのメール送信
- ⑧リンク集

## 2-3 【道のり 2000】

〈〈文責：岡田牧男

### [はじめに]

私たちは、前節までに紹介してきたような「こみゅぽ」を作りました。しかしここまでの道のりは決して平坦ではありませんでした。この「こみゅぽ」は 10 人と言う比較的少ないメンバーで作上げたものですが、その 10 人が 10 人とも全く違う個性と意見をもっていたので、この今ある「こみゅぽ」の姿になるまでにはたくさんの紆余曲折がありました。ここではその「こみゅぽ」ができるまでの経緯を紹介します。

### [夏の間発表まで]

私たちは前述のような夢を実現するために、まずメンバーを二組に分けて、活動をはじめました。ひとつは「こみゅぽ」に組み込むツールをひたすら作る「アプリ班」と、サーバも含めたシステム全般や、トップページ全体のレイアウトや見た目を作り、WebTop という概念を



実装する「土台班」に分けました。

WebTop というのは、「ブラウザさえあればなんでもできるようにする」という考え方です。つまり、ブラウザ上で文章作成や、表計算、電子メールのやり取りや、データベースの検索ができるようにすることです。私たちはコンピュータの「デスクトップ」と同じように、「こみゅぽ」にはいったら、アイコンが幾つか並んでいて、それをダブルクリックすることによってそれぞれのアプリケーションが動くようにできたらいいと、考えました。

「アプリ班」ははじめからツールを一から作るのではなく、インターネット上から探し出して、利用できるのであれば利用し、修正が必要であれば修正を加えるという形をとることにしました。

しかしそれは大変な作業でした。ツールで使えそうなものを探し出してくるということでしたが、自分たちがコントロールできそうなツールがほとんど見つからなくて、そのままの形でしか利用できないからでした。いくつかツールは公開されてはいましたが、ソースコードまで公開しているところは少なく、ほとんど見つかりませんでした。

「土台班」では、サーバの勉強から始まり、WebTop の設計をはじめました。その要素技術として、私たちはサーブレットを採用しました。それは私たちが比較的Java という言語になれていて、比較的とっつきやすい技術であったので採用しました。「とっつきやすい」とはいつでも、それでもどうすれば動くのか、何が必要なのか調べるのが大変でした。それはまだ資料が少ないからだとも考えられます。「土台班」は中間発表まで、ほとんどサーブレットの勉強に時間を使いました。

8月の下旬に中間発表を行いました。そこでは、これまで私たちがやってきたことを発表しました。そこでは「なぜWebTopである必要があるのか」、なぜJavaである必要があるのかななどの批評をいただきました。私たちはツールを作るということに振り回されて、その先にあるものを少し見失っていたところがあったのではないかと反省しました。

WebTopの実装はここで断念しました。それにはいくつかの理由がありましたが、大きな理由の一つとして、何もかもブラウザ上でする必要が現時点ではなかったことです。今ではすでに高機能なソフトウェアがそれぞれのコンピュータにインストールされています。それらのソフトウェアを使うほうが、ブラウザ上でつかえるソフトウェアを今から作って利用するよりもはるかに手軽で、わざわざブラウザ上で作業する必要がないからです。その代わりに、私たちは「こみゅぽ」に「本当に使うツールのみ」を組み込もうという方向を打ち出しました。「こみゅぽ」でしか実現できないもの限定することによって、逆に「よく使うコミュニティポータルサイト」になるのではないかと考えたからです。

### 【「こみゅぽ」のこの形へ】

私たちはこの時点で、この「こみゅぽ」の対象者を「井田ゼミ生」から「青学生」に変更しました。それまでは、お互い顔の見えると範囲の小さな「コミュニティ」のための、「コミュニティポータルサイト」を作るのであり、「自分たちをモデルにして、自分たちが欲しいと思った機能をどんどん組み込んでいく」という形をとってきました。しかしそれでは井田ゼミ

4期生という小規模のコミュニティーに対象を絞ってしまっただけでは、その中の10人の人たちが欲しいツールを考えても、他人の幸せにはならないとかがえました。10人全員好みはあまりないのです。ただでさえこの10人という小さなコミュニティーでもこれだけ違うのですから、「こみゅぽ」を公開しても、あまり有効ではないかという考えに至りました。そこで対象を「青学生」に変更しました。

しかし、私たちはあることに気が付きました。私たちは「青学生」であると同時に、「井田ゼミ4期生」でもあるのです。「青学生」というコミュニティーの中に、ゼミやサークルなど、いくつもの「小さな」コミュニティーが存在しているのです。私たちは、いろいろな「わたし」を持って生活しているのです。私たちは対象を「青学生」と修正すると同時に、右の図2-5のような二重構造に対応するような「こみゅぽ」を設計しようと決めました。



図 2-5

このような修正の元に、私たちは組み込むツールの内容と種類を再び検討し、それぞれどの階層に入れるべきか、分類を試みました。「青学生」として共有したい情報、「グループのメンバー」として共有したい情報、そして「個人」として利用したい情報に分けたのです。そして、「こみゅぽ」の現在のサービス構成ができました。それから組み込むための作業が始まったのです。

最初に挙げた「夢」の中に、「携帯端末でも見られるページ」というのがありました。私たちの中には「いつでもどこでもどんな端末でも(Anytime, Anywhere, and Any devices=トリプルA)」というイメージは強くありました。コンピュータがある場所という限定が加えられると、どうしても使いにくくなるからです。当初私たちは携帯端末として携帯電話版かPDA版の「こみゅぽ」も作ろうと考えました。結局携帯電話版の「こみゅぽ」を作ることになりましたが、PDA版を作ることは断念しました。それは日本において携帯電話が欧米におけるPDAの役割を果たしていることと、周辺機器などに未知数な部分が多すぎるということで断念せざるを得ませんでした。

### [終わりに]

以上のような曲折を経て、「こみゅぽ」は作り上げられました。私はこの「こみゅぽ」に至るプロセスが、その出来上がったものよりも大事ではないかと感じました。それは時として完成されたときに忘れがちになるものであり、これから先を考える上で参考になる重要な「資料」になると思ったからです。次の「こみゅぽ」を考えるときには、必ず私たちがたどってきたこれまで幾度となく交わされた議論が有効になると私は思います。

# 第3章

## サーバ

- 3-1 【「こみゆぽ」サーバ構築記】 p. 17～
- 3-2 【インストール手順】 p. 20～
- 3-3 【PostgreSQL インストール体験記】 p. 24～
- 3-4 【サーバを構築しよう!!】 p. 25～

## 第3章 サーバ

### 3-1 【「こみゅぽ」サーバ構築記】

＜＜文責：姜秀珍

#### ＜サーバを建てるか建てまいか＞

プロジェクトを始めるにあたって、コミュニティーポータルの実体であるサーバに関して井田先生は私たちに二つの選択肢を提示しました。現在運営中の学校にあるサーバに私たちが作るプログラムを一時的に置かせてもらうかというのが一つの選択肢。実際に自分たちの力でサーバをたて、運営を実践してみるのが二つ目の選択肢です。どちらかを選ばなければなりませんでしたが、私たちの考えるコミュニティーポータルはその要素技術としてサーバサイド Java であるサーブレットを使うことにしていたため、サーバには必ず Servlet Container が動いていることが必須条件です。しかし、一時的にプログラムを置かせてもらう予定だった学校のサーバには、Servlet Container は動作していないため、自然と片方の選択肢はつぶれてしまい、自分たちの独自のサーバを建てることに私たちの決意は固まります。

#### ＜サーバマシーン＞

ようやく、サーバをセットアップすることが決定し、私たちは井田先生からサーバマシーン用としてノート型 PC とデスクトップ型 PC という二つのマシーンが与えられます。当初は、自宅作業が可能ということでノート型を選択、先生からノート PC1 台、HUB、LAN ケーブルなどが入っている黒い Java バック（相当重い）を渡されました。サーバキットは渡されたものの、そのノート PC 純正の CD-ROM ドライブがなかったため、サーバ班の一員である macchi75 は自宅のデスクトップの CD-ROM ドライブを利用し、FTP でインストールする方法を取り、第 1 回目である O/S RedHat6.2 J のインストールの成功にいたしました。

#### ＜サーバソフトウェアを実験マシーンに＞

O/S のインストールに成功、これからは、この普通の Linux マシンをサーバ化してあげなければなりません。ということでまず、Apache のインストールを試みようと思いますが、apache1\_3\_12.rpm を展開したら、「このパッケージは既にインストールされています」というメッセージが出て、展開不可能です。Apache はつい先ほどにダウンロードしたばかりなのに…

なんで？

コマンド“cd”でO/Sの中を懸命に探し回っても、“findapache”で検索しても、Apache という名前がついているフォルダーは存在しません。ゼミの先輩や大学院生の方に問い合

わせたところ、どうやら RedHat6.2J のインストールの際に Apache サーバをインストールすると、自動的に RPM 形式で展開されるため、あちこちのディレクトリーに勝手に置かれてしまうということです。どうしても、Apache パッケージのディレクトリー構造をまとめて配置したい。という場合には、

```
$ rpm -e apache
```

このコマンドで apache のパッケージに含まれる全ファイルを削除しなければなりません。ということでした。削除が終了したら、最初からやり直しです。その次は、以下のコマンドを打てば、インストール完了だということです。

```
$ ./configure
```

```
$ make
```

```
$ make install
```

このようなソースコードインストールの方法もありましたが、Apache のパッケージで最小限知っていなければならないファイルが置かれているディレクトリーは、下記の二つだけであったので、時間短縮のため、簡単でスピーディーなパッケージインストールの方を選びました。

```
httpd.conf が置かれている場所 /etc/httpd/conf
```

```
Apache の起動/停止のファイルが置かれている場所
```

```
/etc/rc.d/init.d/httpd
```

Netscape でさっそく `http://localhost` を確認。Jserv と PostgreSQL をダウンロードしてから、その日の作業は終わりです。

翌日、Jserv を Apache と連動させるために実験マシーンをいじっていたら、急に Xwindows が開かなくなっていました。

なんで？

通常は、LILO Boot でブーティングが終わってから、Xwindows が開いた後、enlightment が実施されます。それが、その途中で何回も再起動しても止まってしまうのです。その後、実験マシーンの O/S は、全く X が開けなくなってしまう、真っ暗な kterm の画面でしか、操作ができなくなってしまうのです。RedHat6.2 の日本語版はまだ不安定なのでしょう。Kterm でしか作業ができないということは、サーバソフトの設定が終了したとしても Netscape で確かめることができなくなってしまう。サーバにとって O/S の不安定は致命的なため、恐怖の O/S インストール第 2 回目が始まりました。

### <2.5 インチ HDD 用 変換アダプター>

この時、なにも知らなかった私は、ブート用 .img ディスクを CD-ROM から作り、純正でない CD-ROM ドライブでのインストールを数回も試してみました。あとでわかったのですが、もともと純正でない CD-ROM ドライブを認識させるのは難しいらしいです。そうしているところ、五十嵐さんからノートの HDD は、デスクトップの HDD と “2.5 インチ HDD 変換アダプター” で繋げてデスクトップの方の CD-ROM ドライブを利用してインストールしたら手早いという情報を入手、早速、秋葉原で購入、家のデスクトップ PC を犠牲にし、ふた

をあけて、アダプターをつなげようとした。それが、差しこめません！

なんで？

ピンの数が合わない！！

なんと、デスクトップの IDE ケーブルの方のピンを差す穴が一つ埋まっていて、アダプターのピンの数と一致しません。翌日、調べてみたら勝手に折って使えば良いという事だったので、赤い電源用の線をアダプターの方の赤い線と向きを合わせて、一つピンを抜いてみました。今度こそは！と思い電源を入れます。

ピピピピピ...

<本番サーバマシーン>

結局、実験マシンのインストールでもめている時間が長かったため、もうこれ以上時間の無駄はできず、直接本番マシーン(図 3-1)のインストールに突入することにしました。先生から提供してもらった本番マシンは、実験マシーンに比べると、Dual CPU だけあって、とても快適だという噂を聞いて、インストールのために五十嵐さんの指導の下、macci75と一緒に以下のような手順でサーバセットアップを行いました。



図 3-1 本番サーバ

```
Hostname : semi4
CPU : PentiumPro200MHz ×2
Memory : 96MB
HDD : 20GB
http://133.2.110.17/commupo/
```

## 3-2 【インストール手順】

使用ソフトウェア

<b>O/S</b>	RedHat Linux 6.2J
<b>JDK</b>	JAVA™ 2 SDK, Standard Edition <a href="http://java.sun.com/products/jdk/1.2/ja/download-linux.html">http://java.sun.com/products/jdk/1.2/ja/download-linux.html</a>
<b>Webサーバ</b>	Apache1.3.12 <a href="http://www.redhat.com/swr/i386/apache-1.3.12-25.i386.dl.html">http://www.redhat.com/swr/i386/apache-1.3.12-25.i386.dl.html</a>
<b>Servlet Container</b>	Tomcat3.2 beta4 <a href="http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-4/bin/">http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-4/bin/</a>
<b>コネクタ</b> (ApacheとTomcat間)	mod_jk.so <a href="http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-4/bin/linux/i386/">http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-4/bin/linux/i386/</a>

### Apache1.3.12 インストール

#### 1. Apache パッケージのインストール

- ① `$ rpm -ivh パッケージ名`  
(例) `$ rpm -ivh apachel_3_12.rpm`

#### 2. httpd.conf ファイルを修正

- ① `$ cd /etc/httpd/conf`
- ② httpd.conf を emacs で開きます。  
`$ emacs httpd.conf`
- ③ “ServerName: localhost” の行のコメントアウト (#) を除きます。  
`# ServerName : localhost → ServerName : localhost`

#### 3. Apache の起動/停止

- ① `$ cd /etc/rc.d/init.d/httpd` に移動
- ② 起動 `$ ./httpd start`
- ③ 停止 `$ ./httpd shutdown`

#### 4. ブラウザー (Netscape) で動作を確認

<http://localhost>

“It's working” と書かれた Apache のページが表示されれば完了です。

## JAVA™ 2 SDK, Standard Edition インストール

(JSDK と Tomcat は、わかりやすいように /usr/local のディレクトリーの下に “JAVA” というフォルダーを作り、その下にインストールします。)

```
$ mkdir JAVA /usr/local
```

1. 入手したファイルを展開します。

```
$ tar xvzf ファイル名 置くディレクトリー
```

```
(例) $tar xvzf jdk-1_2_2_0006-linux-i386.tar.gz /usr/local/JAVA
```

2. 環境変数の設定

```
① $ export PATH=/usr/local/JAVA/jdk1.2.2/bin:$PATH
```

```
② $ export JAVA_HOME=/usr/local/JAVA/jdk1.2.2
```

## Tomcat3.2 beta4

1. ファイルを展開

```
$ tar xvzf jakarta-tomcat.tar.gz /usr/local/JAVA
```

2. 環境変数の設定

```
$ export TOMCAT_HOME=/usr/local/JAVA/jakarta-tomcat
```

3. Tomcat の起動/停止

```
① $ cd /usr/local/JAVA/jakarta-tomcat/bin
```

```
② 起動 $ ./startup.sh
```

```
③ 停止 $ ./shutdown.sh
```

## コネクタ (mod\_jk.so) の配置

mod\_jk は、Jserv での mod\_jserv.so と同じ役割を果たす Tomcat と Apache の間のコミュニケーションを取り扱う完全に新しい形の Tomcat-Apache プラグインです。つまり、このファイルの配置ができれば、Tomcat と Apache の連動が可能になります。

1. mod\_jk.so を Apache の libexec に配置します。

```
$ cp mod_jk.so /usr/local/JAVA/apache_1.3.12/libexec
```

2. httpd.conf ファイルに記述を追加

```
① $ cd /etc/httpd/conf
```

```
② $ emacs httpd.conf
```



最終行に

"Include usr/local/JAVA/jakarta-tomcat/conf/mod\_jk.conf-auto" の記述を追加、保存します。

(参考コマンド)

最初は、libexecが見つからなかったので、etc/httpd/modulesに mod\_jk.so を配置していました。これを解決するためには、" ln -sf " でソフトウェアリンクを張ります。

- ① \$ cd /etc/httpd/
- ② \$ ln -sf modules libexec

### 3. Tomcat-Apache 連動を確認

- ① まず、Tomcat と Apache を再起動します。

(Apache は、再起動をしないと、コンフィギュレーションの変化が反映されないからです。)

- Apache の再起動

```
$ cd /etc/rc.d/init.d/httpd
```

```
$ ./httpd restart
```

- Tomcat の再起動

```
$ cd /usr/local/JAVA/jakarta-tomcat/bin
```

```
$ ./startup.sh
```

- ② ブラウザー (Netscape) で連動を確認します。

```
http://localhost/examples
```

- ③ Servlet がちゃんとコンパイルされるかを確認します。

```
http://localhost/examples/servlets/
```

## インストール完了

サーバインストールは無事終了し、http://133.2.110.17/commupo/ という 「こみゆぽ」の URL もいただいて、インターネット上に公開するまで至りました。ということで、ツール班のみんなも自宅から FTP で各自が作ったプログラムを簡単に載せることが可能になりました。しかし、サーブレットのファイルは一体どのディレクトリーに置けば良いのでしょうか。ここからが本当に Tomcat を知る段階に入ります。

Tomcat3.2b4 パッケージをインストールすると、jakarta-tomcat というフォルダーが形成されます。その中にはさらに以下のようなフォルダーで構成されています。

```
LICENSE bin/conf/doc/lib/logs/src/webapps/work/
```

ここで重要なディレクトリーは、conf、webapps の二つです。また、conf の中には、Tomcat の挙動をコントロールする server.xml と Webapplication の挙動をコントロールする web.xml が置かれています。Servlet ファイルが自動 Mapping されるディレクトリーを利用する場合には、この二つのファイルは書きかえる必要はありません。

```
webapps/examples/WEB-INF/classes/
```

には、Tomcat にデフォルトで入っている Servlet と JSP の例が入っています。URL は、<http://133.2.110.17/examples/servlets> や <http://133.2.110.17/examples/jsp/> になりますが、これと同じような構成のディレクトリーを webapps の下に作るだけで Ok なのです。こみゅぽの場合は、

```
http://133.2.110.17/commupo/
```

```
$ pwd
$ /usr/local/JAVA/jakarta-tomcat/webappas/
$ mkdir commupo
$ cd commupo
$ mkdir WEB-INF
$ cd WEB-INF
$ mkdir classes
$ pwd
$ usr/local/JAVA/jakarta-tomcat/webappas/commupo/WEB-INF/classes
```

これで新しく作ったフォルダー構成は、上記のようになりました。で、ここにサーブレットのクラスファイルを置くということです。webapps/examples と全く同じ構成にしてあげました。新しいクラスファイルを置いてからは、必ず、Apache を再起動しなければならないことに注意しましょう。この classes というフォルダーに置かれているサーブレットは、URL では、どのように表記されるのかを見てみましょう。

```
http://133.2.110.17/commupo/servlet/〇〇
```

〇〇には、クラスファイル名になるでしょう。

これで、サーバインストール及び、設定完了です。この一連の作業については、

```
http://jakarta.apache.org/tomcat/jakarta-tomcat/src/doc/uguide/tomcat ug.html
```

に詳しく載っていますので参照しましょう。また、英語が苦手な方は、

```
http://www.slayers.nu/~kumasaka/Gyoumu/tomcat-ugj.html
```

```
http://servlet.good-day.net/tomcat ug j.html のような日本語翻訳のページがお勧めです。
```

### 3-3 【PostgreSQL インストール体験記】

◀文責：黒田洋史

まず、WEB を参考にインストールをする。手元にある PostgreSQL についての特集のある JavaPress を見る。そこには、参考のホームページのアドレスが書いてありそこをみる。それが PostgreSQL mailing list in Japan

(<http://www.sra.co.jp/people/t-ishii/PostgreSQL/>) である。そこには幾つかのバージョンのインストール方法が掲載されていたが、私は最新版の PostgreSQL 7.0.2 を選び、そのインストール解説ページを見ました。そこには、とても親切にインストール方法が記載されています。また、インストールするパソコンに新たに postgres ユーザを作っておかなくてはならないと知り、root になりユーザ db を作りました。(どうも、PostgreSQL は root ではインストールできないらしい。)

手順ですが、まずオリジナルソースを入手します。上記のページには、オリジナルソースの入手できるサイトへのリンクがあり、そこをクリックすることで簡単に入手できます。このサイトには修正済みのバッチ済みソースがあるのがわかったのでそちらを入手します。そのあと、コンパイルを行いました。なので、tar でコンパイルするができませんでした。よくページを見るとコンパイルを行う先の /usr/local/src/ にユーザ db に書き込みの許可がないからだそうです。そこで root になり、/usr/local/src/ に db での書き込み許可を与えました。そしてコンパイルを実行しました。そうしたらいつてくれたようです。コンパイルに時間がかかるようなので一休みしに、席をはずしました。

席に戻ると、コンパイルはうまくいったようでした。少しほっとしました。そのあと、続けてインストールを行いました。インストールはただ、make install と打つだけで OK でした。自動的にファイルが超高速で行われます。本当に目にも止まらぬ速さってこういうのを言うんだってなんか感じました。コンパイルも、無事終わりました。そのあと、実際に PostgreSQL を使うため、内部の環境変数の設定を行いました。Linux の環境変数は .bashrc に書き込むようにしている (bash なので)。よく、私はこの設定の方法を忘れてしまいます。つい最近に JSWDK のインストールでやったばかりなのに、もう忘れてしまい、いろいろ古いノートを見返してやっとやり方を思い出しました。

環境変数の設定が終わったので、initdb とうちデータベースを初期化しました。そのあと、postmaster とうち、postmaster を起動します。しかし、そこで動いてくれませんでした。実は、僕の Linux には古いバージョンの PostgreSQL が入っていてどっちを起動し

たらいいか困っているようでした。そこで私は、`/usr/local/pgsql/bin/`にある `postmaster` を指定し起動しました。そうすると動いてくれるようです。古いバージョンが入っているときだけに起こることらしいが(2 回目に入れたマシンでは起こらなかった)ので) 注意が必要であると思いました。それで、実際に使えるかどうかのテストの為 `regression` を行いました。そうすると、次々にチェック項目が出て結果が表示されました。すべて、`ok` と出たようでした。よかったほっとしました。そのあと、`psql` を使ってちょっと `SQL` で遊んだりしました。これも動いてくれました。本当にほっとしました。

このあと、何度もインストールするチャンスを得てもらいやらせてもらい今まで 3・4 階やってきたところで、やっとやり方に慣れました。実際、Windows の GUI 中心のインストールに慣れている自分にとって違和感が多くありましたが、何度かやると UNIX のほうが自分のマシンが何を今作業しているか分かりやすいので、インストール自体結構早い上、失敗もないので便利かなって今は少し感じています。

## 3-4 【サーバを構築しよう!!】

◀◀ 文責：清水淳

[必要なもの]

～Linux ディストリビューション～

**RedHat6.2J (FTP 版)**

[入手先] 雑誌の付録などを利用します

～JAVA 開発環境～

**JAVA2SDK1.2 (Linux版)**

[入手先] <http://java.sun.com/products/jdk/1.2/ja/download-linux.html>

～Web サーバー～

**Apache (Ver.1.3.14/ソースコード)**

[入手先] [http://www.apache.org/dist/apache\\_1.3.14.tar.gz](http://www.apache.org/dist/apache_1.3.14.tar.gz)

～Servlet Engine～

**Tomcat (バイナリ)**

[入手先]

<http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-3/bin/jakarta-tomcat.tar>

～コネクタ (Apache と Tomcat を繋ぐもの)～

**mod\_jk.so**

[入手先]

[http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-3/bin/linux/i386/mod\\_jk.so](http://jakarta.apache.org/builds/tomcat/release/v3.2-beta-3/bin/linux/i386/mod_jk.so)

- ① Linux をインストールする
- ② JAVA2SDK をインストールする
- ③ Apache をインストールする
- ④ Tomcat をインストールする
- ⑤ コネクタをインストール

## ① Linux をインストールしよう!

まずは Linux ディストリビューションを入手しましょう。製品版を購入しても構いませんが、雑誌の付録としてついてくるもので十分でしょう。ここでは、RedHat6.2J をインストールします。

RedHat6.2J のインストールの仕方については、雑誌の説明書きを読むか、または「redhat JAPAN」(<http://www.redhat.com/jp/>) の「Red Hat Linux 6.2 インストールガイド」(<http://www.redhat.com/jp/support/manuals/RHL62/install-guide/>) を参考にしてください。

\*以下の作業はスーパーユーザーで行ってください。

## ② JAVA2SDK (=JDK1.2.2) をインストールしよう!

1) JAVA2SDK1.2  
(<http://java.sun.com/products/jdk/1.2/ja/download-linux.html>) を Sun Microsystems の Web サイトからダウンロードするか、雑誌の付録などから入手してください。

2) 入手したファイルを「/usr/local」にコピーします。  
`#cp jdk-1_2_2_0006-linux-i386.tar.gz /usr/local`

3) 「/usr/local」に移動して、コピーしたファイルを展開 (=解凍) します。  
`#cd /usr/local`  
`#tar zxvf jdk-1_2_2_0006-linux-i386.tar.gz`

4) 「/usr/local」に「jdk1.2.2」が作られます。

5) JDK1.2.2 のパス変数の設定をします。

\*「パスって何?」という人は何も考えずに以下の作業を必ず行って下さい。

JDK1.2.2 を動作させるのに必要な作業です。

```
#export CLASSPASS=/usr/local/jdk1.2.2/lib/classes.zip
#export PASS=/usr/local/jdk1.2.2/bin:$PATH
```

6) 次回起動時に 5) の作業を行わなくても済むように「.bashrc」というファイルの最終行に次の内容を書き込んでおきます。

```
#emacs .bashrc
```

-----  
(いろいろなことが書かれています)

```
export PATH=/usr/local/jdk1.2.2/bin:$PATH
export JAVA_HOME=/usr/local/jdk1.2.2
```

-----  
(次ページへ続く⇒)

7) パスが正しく設定されているかを確認します。

```
#which java
```

と打って「/usr/local/java1.2.2/bin」と表示されればOKです。

### ③ Apache をインストールしよう!

続いて Web サーバの Apache をインストールしましょう。実際の作業に入る前に、RedHat6.2J をインストールしたときにパッケージに含まれていた Apache を削除します。

```
#rpm -e apache
```

さて、実際にインストールの作業に入りましょう。

1) ダウンロードした Apache のファイルを「/usr/local」にコピーします。

```
#cp apache_1.3.14.tar.gz /usr/local
```

2) 「/usr/local」に移動して、コピーしたファイルを展開 (≒解凍) します。

```
#tar zxvf apache_1.3.14.tar.gz
```

3) 「/usr/local」に「apache\_1.3.14」が作られます。

4)Apache をインストールします。

```
#!/configure  
#make  
#make install
```

5)Apache を起動させてみましょう。

```
# /usr/local/apache/bin/apachectl start
```

と打って、「/usr/local/apache/bin/apachectl start:httpd started」と表示されれば、Apache は起動しています。ためしにブラウザを起動させて、アドレス入力欄に「http://localhost/」と入力してみましょう。「It worked!」と表示されるはずです。

6)Apache を停止させるには、

```
# /usr/local/apache/bin/apachectl stop
```

と入力してあげます。

#### ④ コネクタ (mod\_jk.so) をインストールしよう!

1) ダウンロードしたファイルを「/usr/local/ apache\_1.3.14/libexec」にコピーします。

```
#cp mod_jk.so /usr/local/ apache_1.3.14/libexec
```

たったこれだけで終わりです。

#### ⑤ Tomcat をインストールしよう!

Tomcat のインストールはソースコードからではなく、バイナリから行います。なぜ? ソースコードからのインストールはとても面倒だからです(^\_^)

1) ダウンロードした Tomcat のファイルを「/usr/local」にコピーします。

```
#cp jakarta-tomcat.tar.gz /usr/local
```

2) 「/usr/local」に移動して、コピーしたファイルを展開（≒解凍）します。

```
#tar zxvf jakarta-tomcat.tar.gz
```

3) 「/usr/local」に「jakarta-tomcat」が作られます。

4) Tomcat を起動させてみましょう。

```
#cd /usr/local/Jakarta-tomcat/bin
```

```
#./startup.sh
```

ズラーっとメッセージが流れてくれば、Tomcat は起動しています。ためにブラウザを起動させて、アドレス入力欄に「http://localhost:8080/」と入力してみましょう。猫?のイメージが表示されればTomcat は起動しています。

\* もし起動しない場合は

```
#export PATH=/usr/local/jdk1.2.2/bin/:$PATH
```

```
#export JAVA_HOME=/usr/local/jdk1.2.2
```

```
#export TOMCAT_HOME=/usr/local/Jakarta-tomcat
```

と入力してみてください。

5) Tomcat を停止させるには、

```
#cd /usr/local/Jakarta-tomcat/bin
```

```
#./shutdown.sh
```

と入力してあげます。

6) Tomcat を Apache にアドオンするには、

```
#emacs /usr/local/apache_1.3.14/conf/httpd.conf
```

として「httpd.conf」というファイルの最終行に、「Include /path/to/build/tomcat/conf/mod\_jk.conf-auto」を書き加えます。



7) サンプルが動作するのを確認します。

ブラウザを起動して、アドレス入力欄に

「`http://localhost:8080/examples/servlets`」と入力して、サンプルが表示されれば OK です。

# 第 4 章

## ツール

4-1 【はじめに】	p. 33～
4-2 【講義評価】	p. 34～
4-3 【青山周辺情報】	p. 40～
4-4 【個人/グループスケジュール管理ツール】	p. 49～
4-5 【LIDA (Library of IDA)】	p. 58～
4-6 【グループリンク共有】	p. 64～
4-7 【グループ投票管理】	p. 71～
4-8 【ユーザ管理】	p. 86～
4-9 【コミュニティー管理 (新規作成・検索・参加表明)】	p. 94～
4-10 【K3 君 1 号 (GPA 計算機)】	p. 105～
4-11 【K3 君 2 号 (GPA 計算機)】	p. 107～

4-12 【あの人 Finder】	p. 109～
4-13 【私のツール作成体験記】	p. 111～
4-14 【ツール説明書】	p. 117～

# 第4章 ツール

## 4-1【はじめに】

〈文責：黒田洋史

「こみゅぽ」のシステムやツールの設計及び実装は、裏を支えるものであり、このプロジェクトの実現にとって必要不可欠なものである。私たちは「こみゅぽ」を作る上で以下のような目標を定めた。

- ① 入力した内容が自動的に逐次更新される
- ② 「こみゅぽ」は、人が交わり交流する場である
- ③ 「こみゅぽ」にアクセスしている各個人が一人の人間として尊重されなければならない。そのためには、その人が誰であるのかを特定する必要がある。

「こみゅぽ」を作成する過程で多くのことを体験し学ぶことができた。それと同時に必要不可欠なものを作ることに対する責任感や緊張感を味わうことができた。そこでこの章では、各ツールやシステムごと担当者による仕様書と製作体験記をまとめた。

## 4-2【講義評価】

＜文責：森健蔵、黒田洋史

### ①作成理由

このツールを作成した理由は大きく分けて二つある。1つは、年度始めに行う時間割作成において、既存のシラバスだけでは不十分であるということ。もう1つは教授から生徒への評価があるなら、生徒から教授への評価があっても良いのではないかと思ったからだ。

まず1つ目の理由について。私たち青山学院の学生は年度始めに時間割を決定する基準として、学校側からもらうシラバス、知人、先輩の情報、及び第一週目に実際にその授業に参加してみることである。これらにはそれぞれ欠点がある。シラバスは学校や教授の意図がかなり色濃く反映されており良いことしか書いていないばかりか、講義内容が白紙の授業もある。また、サークルなどに入っていない生徒、入学したばかりの1年生は知人や先輩の知り合いが少ないことから、授業に関する情報を得るのが難しい。さらに、1週目の授業は同じ時限に複数の希望教科が重なって参加出来ないことがある。

2つめの理由について。私たちは常に教授から評価を受ける。それは生徒である以上は当たり前である。しかし、生徒から不満が上がっている授業があることも事実だ。それを私たち学生側が講義を評価する事で改善する事が可能ではないだろうかと考えたからである。生徒からの評価を教授が参考することによって、お互いが恩恵を受ける事が出来るのではないだろうか。

以上に述べた二つの理由から、この青山学院講義評価を作成した。このツールは教授を非難、中傷するものではなく、あくまでも学生がより良い授業を受けられるようになることを目的として作成された。

### ②機能

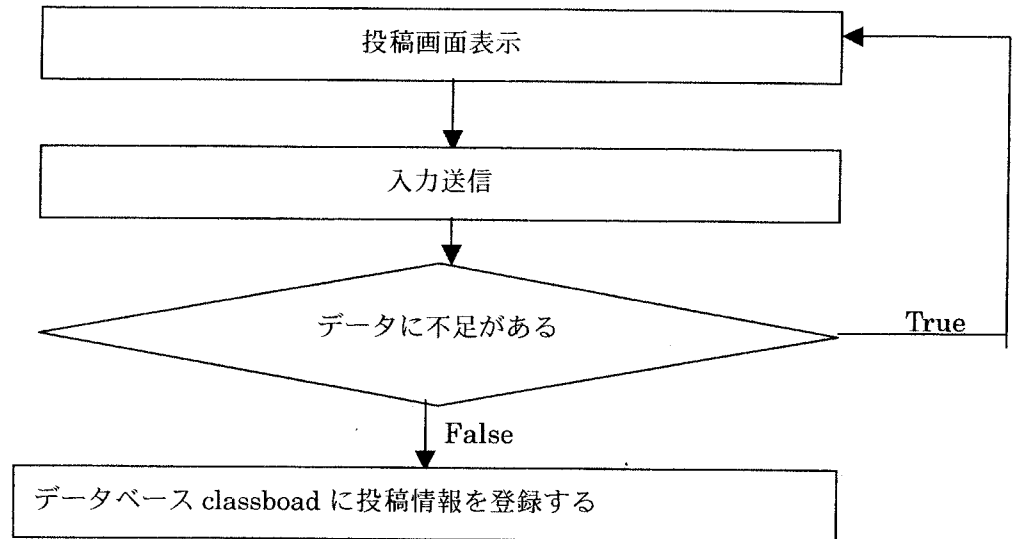
ユーザには始めに評価か閲覧のどちらかを選択してもらう。

評価を選択した場合、評価しようとしている講義がどの学部にも属するものかを選択し、授業名、講義名、講義内容、出席の採り方、課題などを記入してもらう。最後に総合点を1～5でつけてもらう。

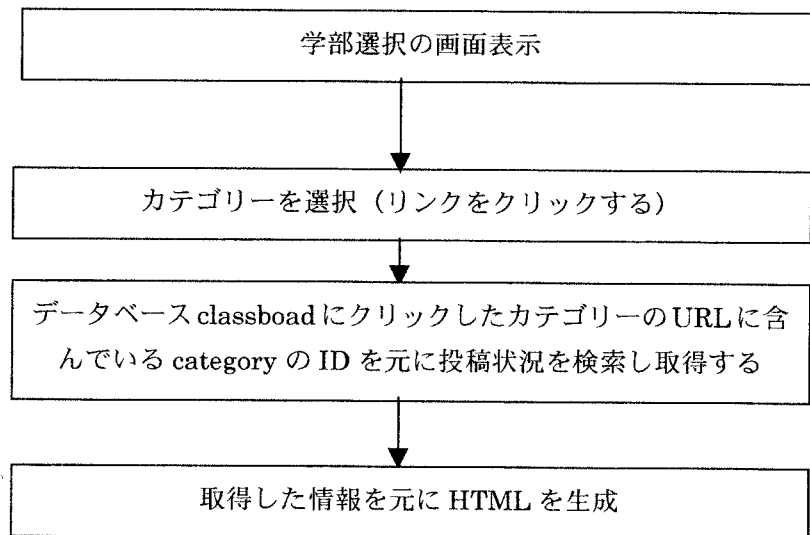
閲覧を選択した場合、知りたい授業が属する学部を選択して「閲覧」をクリックすると他の生徒が評価した内容を閲覧する事が出来る。

### ③アルゴリズム

#### i) 投稿



#### ii) 投稿を見る



### ④使用した技術

JavaServlet とリレーショナルデータベース

### ⑤使用したデータベース

(テーブル名 : classboard)

カラム名	cbid	codename	gakubu	classname	professor
データ型	Int 型	Text 型	Int 型	Text 型	Text 型
詳細	ナンバリングする	投稿者のcodename	学部	授業名	担当教官名

grade	comment	point	nowdate
Text 型	Text 型	Int 型	Text 型
学年	コメント	点数	投稿時の時間を記録

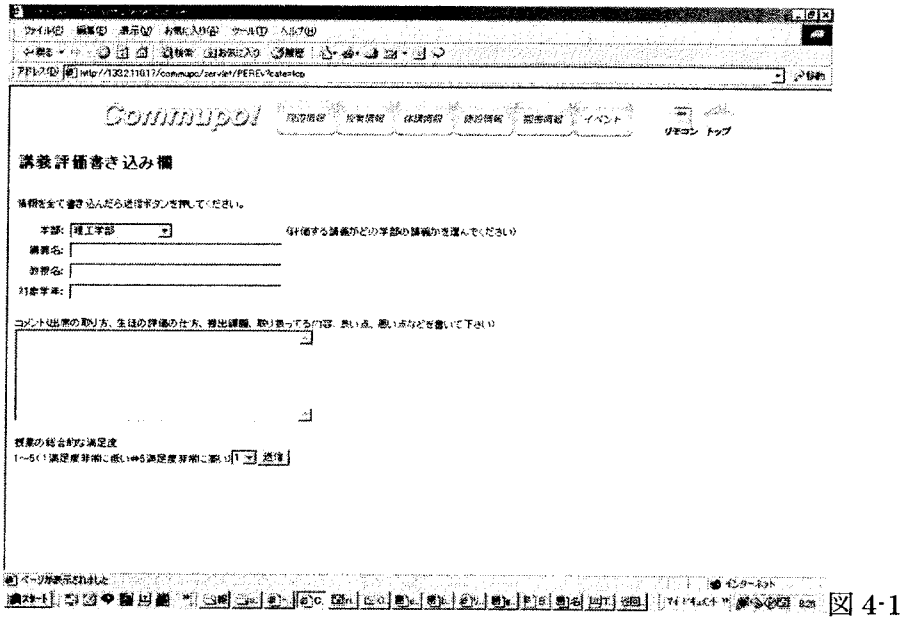
### ⑥製作体験記

今後の改良点は、投稿をしてもらう時にはある程度決まったテンプレートを用意したほうが良いことである。そうすることで、閲覧した時により見やすくなると考えられる。そして、閲覧時の検索をより多くの方法で出来るようにしたほうが良いと思った。今回は学部別だけだったが、教授名、分野別、総合点が満点のものなど色々な要素から検索できればより有効なものになると感じた。(森)

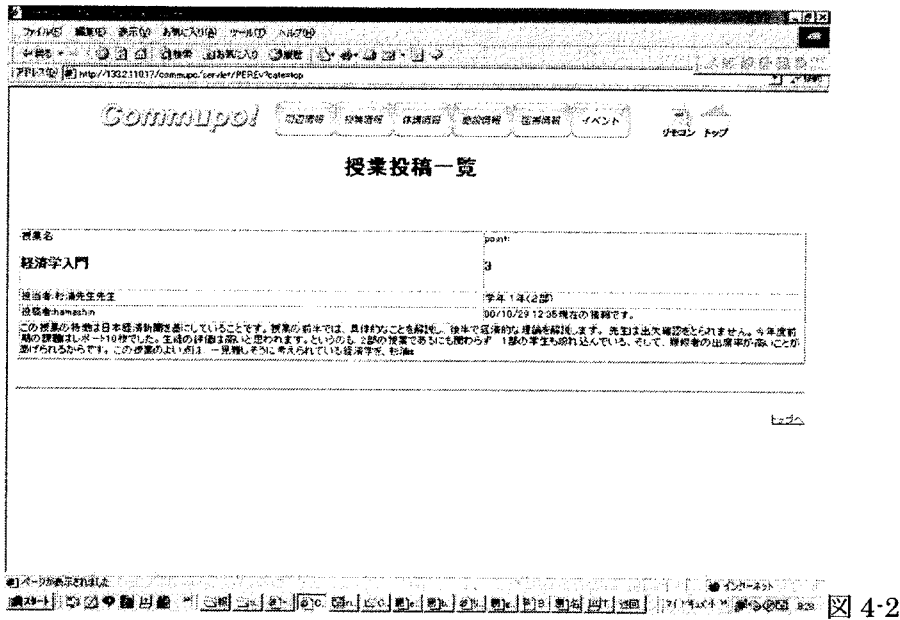
これは、moriken とともに作った 2 つ目のツールである。授業の感想をお互いで交換してもらい、というのがコンセプトだった。実際の製作時には、授業に関する中傷や過激な発言の書き込みを避けるためのガイドライン作りに奔走した。そして、簡単なものであるがどうにか作ることができた。青山祭の時には多くの学生が書き込んでくれたが、心配した中傷といったたぐいの発言は無く安心した。そして、学生は授業の感想を交換する場を求めているのだなと実感した。まだまだ改良に励もうと思う。(黒田)

(参考 : 表示画面)

投稿画面

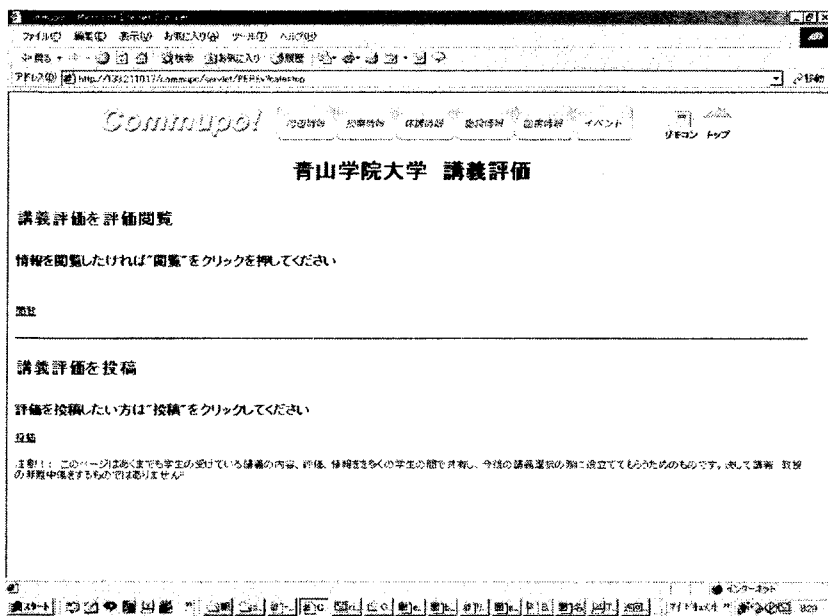


## 投稿閲覧画面





## ガイドライン



4-3

(参考：Java ソース)

```

/*
 * @(#) CLASSControl.java
 *
 * Information BOAD of CLASS
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 */

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CLASSControl{
    CLASSU CLASSU=new CLASSU();
    CLASSConnectDB CLASSDB=new CLASSConnectDB();

    public void write(HttpServletRequest request,HttpServletResponse response,Hashtable parameter)throws
    IOException,ServletException {
        Cookie codenameCookie=getCookieValue(request,"Codename");
        String codename=codenameCookie.getValue();
        int gakubu=Integer.parseInt((String)parameter.get("gakubu"));
        String classname=(String)parameter.get("classname");
        String professor=(String)parameter.get("professor");
        String grade=(String)parameter.get("grade");
        String comment=(String)parameter.get("comment");
        int point=Integer.parseInt((String)parameter.get("point"));
        String nowdate=getNowDate();
        CLASSDB.writeDB(codename,gakubu,classname,professor,grade,comment,point,nowdate);
        CLASSU.finishwrite(response);
    }

    /**
     *

```

```

*   makeList
*   投稿リスト作成
*
*/

public void look(HttpServletRequest request, HttpServletResponse response, Hashtable parameter) throws
IOException, ServletException{

    int gakubu=Integer.parseInt((String)parameter.get("gakubu"));

    Vector classcommentlist=CLASSDB.makeList(gakubu);
    CLASSU.makeList(response,classcommentlist);
}

/**
 *   getNowDate
 *   現在時刻を年/月/日 時:分にする
 */
    private String getNowDate()
    {Calendar rightNow=Calendar.getInstance();
    int yearInt=rightNow.get(Calendar.YEAR)-2000;
    String year=checkInt(yearInt);
    int monthInt=rightNow.get(Calendar.MONTH)+1;
    String month=checkInt(monthInt);
    int dateInt=rightNow.get(Calendar.DATE);
    String date=checkInt(dateInt);
    int hourInt=rightNow.get(Calendar.HOUR_OF_DAY);
    String hour=checkInt(hourInt);
    int minuteInt=rightNow.get(Calendar.MINUTE);
    String minute=checkInt(minuteInt);
    String dateNow="" +year+"/"+month+"/"+date+" "+hour+":"+minute+"";
    System.out.println(dateNow);
    return dateNow;
    }

/**
    checkInt      数値が1桁の時0を頭につける
*/

private String checkInt(int numbers){
    String stringNumber=null;
    if(numbers<10)
        {String stringNumbera=String.valueOf(numbers);
        stringNumber="0"+stringNumbera;}
    else{stringNumber=String.valueOf(numbers);}
    return stringNumber;
}

/**
 *   getCookieValue
 *   クッキーゲット
 *
 */
    private Cookie getCookieValue(HttpServletRequest request,String cookieName){
    Cookie cookies[]=null,cookie=null;

    if((cookies=request.getCookies())!=null){
        for(int i=0;i<cookies.length;i++){
            if(cookies[i].getName().equals(cookieName)){
                synchronized(this){
                    cookie=(Cookie)cookies[i].clone();
                }
                return cookie;
            }
        }
    }
    return null;
}
}
}

```

## 4-3【青山周辺情報】

〈〈文責：森健蔵、黒田洋史

### ①作成理由

この青山周辺情報を作った最大の理由は自分が欲しかったからだ。青学生として半年以上も経つのに、渋谷キャンパスの周りにどのようなお店などがあるかがほとんど知らない。それを解決するために、このツールを提案した。

大きな特徴としては周辺情報を閲覧するだけではなく、自分の知っている情報を投稿できるようにした点があげられる。なぜそうしたかと言うと、ユーザが親しみを感じ興味を示してくれると思ったからだ。加えて、管理者だけの情報ではユーザを十分に満足させることはできないと考えたからだ。ユーザが情報を載せる事で、多くの人が情報を共有できるだろう。

「周辺情報なら Yahoo!などで検索すれば調べられる」と思う方もいるかと思うが Yahoo!に載っている情報は老人から子供まで全ての世代をカバーするもので、僕たち大学生が欲しい情報とは隔たりがある。その点、学生によって書き込まれていく情報は、同世代で似た境遇の人々によるものなのでユーザにとってはとても有効的なものになるだろうと思う。

### ②機能

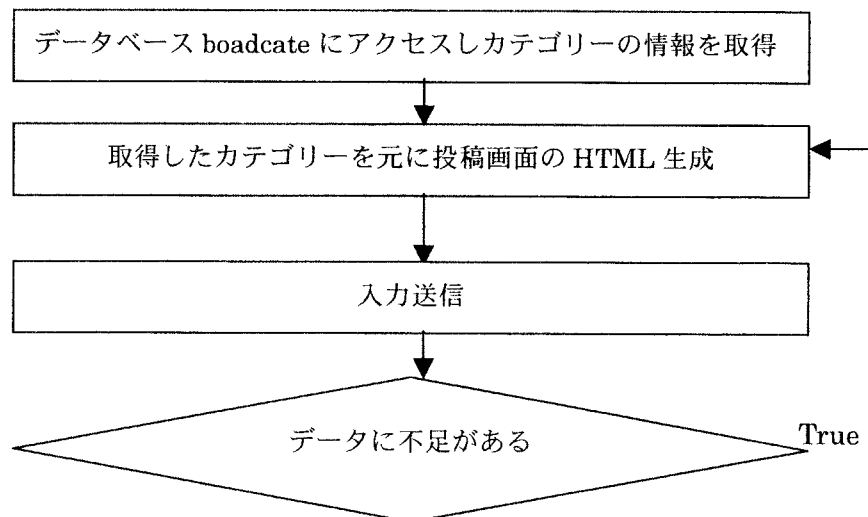
始めにユーザは投稿か閲覧を選択する。

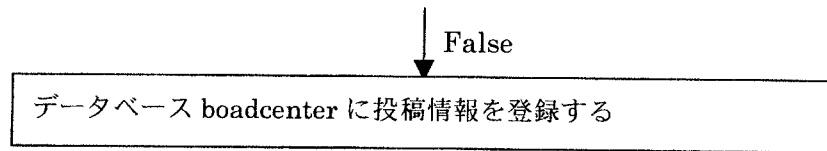
投稿を選択した場合、いくつか用意したカテゴリ（例えば、居酒屋）の中から1つを選び投稿してもらおう。もちろん、新しくカテゴリを作ることも出来る。新しく作ったカテゴリはすぐに閲覧の画面に追加される。

閲覧を選択した場合、ユーザは知りたいの情報のカテゴリを選択する。次に「閲覧」をクリックすると内容が表示される。

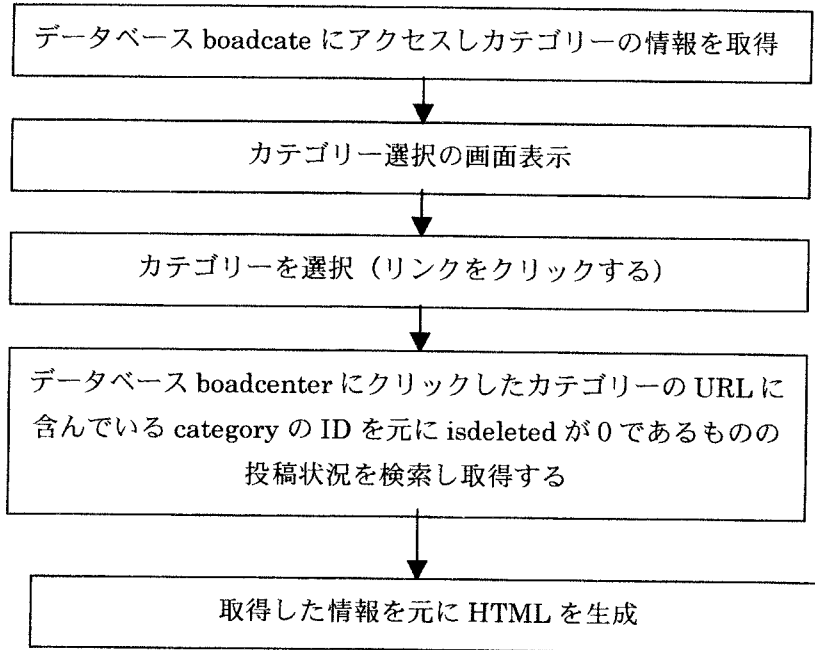
### ③アルゴリズム

#### i) 投稿

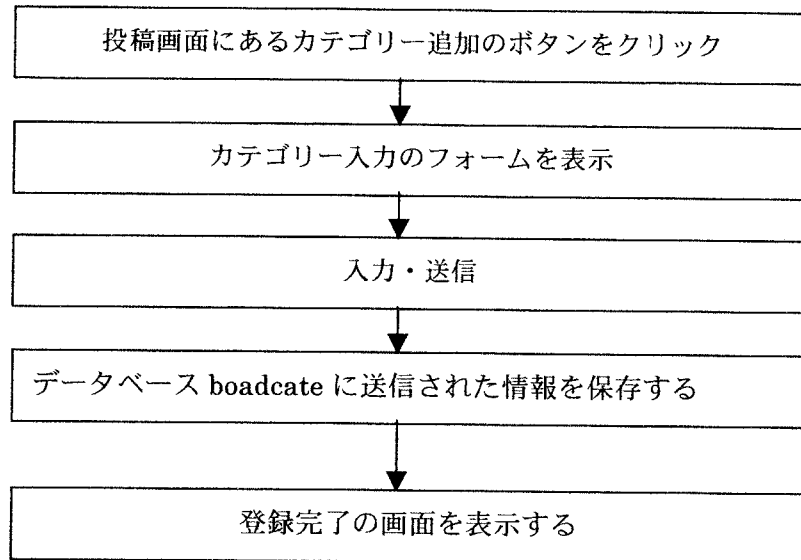




ii) 投稿を見る



iii) カテゴリ追加



④使用した技術

JavaServlet 及びリレーショナルデータベース

## ⑤使用したデータベース

(テーブル名 : boadcenter)

カラム名	commentid	cateid	title	place	comment
データ型	Int 型	Int 型	Text 型	Text 型	Text 型
詳細	ナンバリング する	カテゴリーの ID	タイトル	所在地	紹介文
address	codename	isdeleted	password	nowdate	category
Text 型	Int 型	Text 型	Text 型	Text 型	Int 型
お店への連絡 先	投稿者の codename	0→未削除 1→削除済み	削除時の確認 用パスワード	書き込み時間 を記録	1→ゼミ 2→サークル 3→その他

(テーブル名 : boadcate)

カラム名	cateid	name
データ型	Int 型	Text 型
詳細	ナンバリング する	カテゴリーの 名前

## ⑥製作体験記

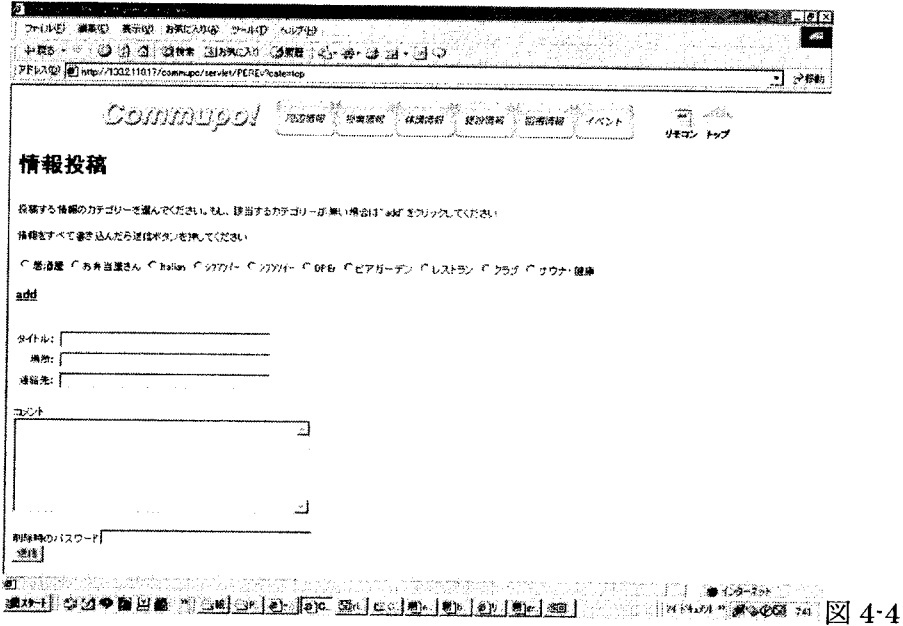
実際使っていただいた方々からはかなり良い評価をいただいた。やはり自分の知っている情報が Web 上に載り、他の人にも役に立つかもしれないという期待感からなのではないのだろうか。その一方で二つの点で指摘を受けた。1つは使い方が難しいということ。もう1つはインターフェイスが寂しいということ。どちらも技術的な問題ではないので時間をかければ解決できる問題であった。時間が無い事を理由にユーザ側の視点が欠如していたことに後悔している。(森)

このツールは、青山学院大学というコミュニティにはぴったりである。大学内コミュニティ用のツールでは中核をなすものになるだろう。

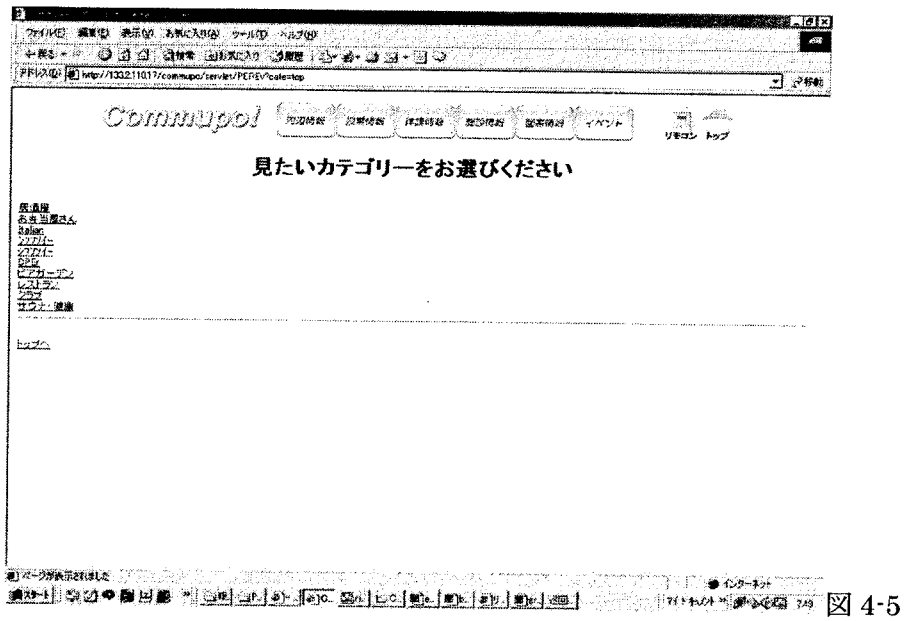
製作には設計を担当した moriken と時間をかけ話し合った。設計者と実装者で理解の違いが起こることだけは避けたかった。そこで、僕も moriken もお互いの夢を十分に理解した上で製作した。彼の大きな夢は「カテゴリーが追加できること」である。今までのものだと管理者の作ったカテゴリーに沿って答えなくてはならなかった。このツールでは自分の欲しいカテゴリが無ければ作ることができるというのが彼の考えだった。僕もその考えに同意した。しかし、すごく不安だった。データベースでは処理できるだろうが、それを HTML にする時に問題が起こるのではと。結果は全く問題なかった。しかし、インターフェイスに問題があるので、その点を改良したい。

(参考：表示画面)

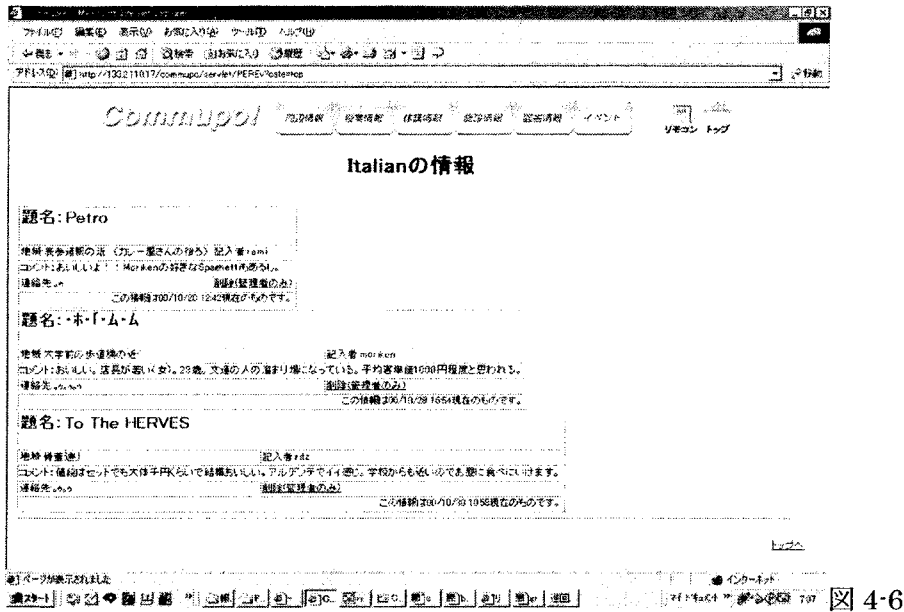
### 投稿画面



### 34 閲覧時のカテゴリ選択



### 投稿閲覧画面



(参考 : Java ソース)

```

/*
 * @(#) BOADCenter.java
 *
 * Information BOAD of AOYAMA
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 */

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class BOADCenter extends HttpServlet {
    BOADConnectDB BOADCDB=new BOADConnectDB();

    public void write( HttpServletRequest request, HttpServletResponse response,Hashtable parameters) throws
        ServletException, IOException{
        response.setContentType("text/html; charset=EUC-JP");
        PrintWriter pw = response.getWriter();

        try {
            Vector catelist=BOADCDB.getCateList();
            createForm(pw, request,catelist);
        } catch (Exception e) { e.printStackTrace();
            throw new ServletException(e.toString());
        }
        pw.flush();
        pw.close();
    }

    public void write2(HttpServletRequest request, HttpServletResponse response,Hashtable parameters) throws
        ServletException, IOException {
        Cookie Codamecookie=getCookieValue(request,"Codename");
        String Codename=Codamecookie.getValue();
        response.setContentType("text/html; charset=EUC-JP");
    }
}

```

```

PrintWriter pw =response.getWriter();
String nowdate=getNowDate();
String cateid=(String)parameters.get("cateid");
String title=(String)parameters.get("title");
String place=(String)parameters.get("place");
String address=(String)parameters.get("address");
String comment=(String)parameters.get("comment");
String password=(String)parameters.get("password");
if(cateid==null | title.equals("") | place.equals("") | address.equals("") | comment.equals("") |
    password.equals("")){
makeErrorpage(pw);
}
else{
try {
BOADCDB.setComment(parameters,Codename,nowdate);
createStatus1(pw,parameters);
} catch (Exception e) {e.printStackTrace();
throw new ServletException (e.toString());
}
}
}
}

```

```

public void look(HttpServletRequest request,HttpServletResponse response,Hashtable parameters)throws
    IOException,ServletException{
response.setContentType("text/html; charset=EUC-JP");
PrintWriter pw = response.getWriter();
Vector catelist=BOADCDB.getCateList();

pw.println("<html><head><title>見たいカテゴリーをお選びください</title></head><body
    bgcolor=white><center><h1>見たいカテゴリーをお選びください</h1></center><br>");
Enumeration catedata=catelist.elements();
while(catedata.hasMoreElements()){
String cateid=(String)catedata.nextElement();
String name=(String)catedata.nextElement();
pw.println("<a href=BOADEV?cate=look2&cateid="+cateid+">"+name+"<br>");
}
pw.println("<hr><br><p align=right><a href=BOADHTML?page=index>トップへ</a></body></html>");
}

```

```

public void look2(HttpServletResponse response,Hashtable parameters)throws IOException,ServletException{
int cateid=Integer.parseInt((String)parameters.get("cateid"));
Vector commentlist=BOADCDB.getComment(cateid);
String catename=BOADCDB.getCatename(cateid);
Enumeration commentdata=commentlist.elements();
response.setContentType("text/html; charset=EUC-JP");
PrintWriter pw = response.getWriter();
pw.println("<html><head><title>"+catename+"の情報</title></head><body
    bgcolor=white><center><h1>"+catename+"の情報</h1></center><br>");
while(commentdata.hasMoreElements()){
String commentid=(String)commentdata.nextElement();
String mcateid=(String)commentdata.nextElement();
String title=(String)commentdata.nextElement();
String place=(String)commentdata.nextElement();
String comment=(String)commentdata.nextElement();
String address=(String)commentdata.nextElement();
String codename=(String)commentdata.nextElement();
String isdeleted=(String)commentdata.nextElement();
String password=(String)commentdata.nextElement();
String nowdate=(String)commentdata.nextElement();
if(isdeleted.equals("0")){
pw.println("<table border=1><tr><td colspan=2><h2>題名 : "+title+"</h2></td></tr>");
pw.println("<tr><td>地域:"+place+"</td><td>記入者:"+codename+"</td></tr>");
pw.println("<tr><td colspan=2>コメント : "+comment+"</td></tr>");
pw.println("<tr><td>連絡先:"+address+"</td><td><a href=BOADEV?cate=delete&commentid="+commentid+">削
    除(管理者のみ)</a></td></tr>");
}
}
}

```



```

pw.println("<tr><td colspan=2><p align=right><small>この情報は"+nowdate+"現在のものです。
        </small></td></tr></table>");
}
}
pw.println("<hr><br><p align=right><a href=BOADHTML?page=index>トップへ</a></p>");
pw.println("</body></html>");
}

public void addcate(HttpServletResponse response,Hashtable parameters)throws IOException,ServletException{
response.setContentType("text/html; charset=EUC-JP");
PrintWriter pw = response.getWriter();
pw.println("<html><head><title>カテゴリ追加</title></head><body bgcolor=white><h1><center>カテゴリ追加
        </center></h1><br><form method=post action=BOADEv><input type=hidden name=cate
        value=addcate2>カテゴリ一名<input type=text name=name><br><input type=submit value=ok>");
pw.println("</form><br><hr><br><p align=right><a href=BOADHTML?page=index>トップへ
        </a></body></html>");
}
public void addcate2(HttpServletResponse response,Hashtable parameters)throws
        IOException,ServletException{
String catename=(String)parameters.get("name");
response.setContentType("text/html; charset=EUC-JP");
PrintWriter pw = response.getWriter();
if(catename.equals("")){
makeErrorpage(pw);
}
else{
BOADCDB.setCate(catename);

pw.println("<html><head><title>カテゴリ追加しました</title></head><body bgcolor=white><center><h1>カテ
        ゴリ追加しました</h1></center><br><br><br><hr><br><p align=right><a
        href=BOADEv?cate=write>書き込み</a><br><p align=right><a href=BOADHTML?page=index>トップへ
        </a></body></html>");
}
}

public void createForm(PrintWriter pw, HttpServletRequest request,Vector catelist) throws IOException{

pw.println("<html><head>");
pw.println("<title>keijiban2</title>");
pw.println("<body bgcolor=white>");
pw.println("<h1 align=¥left¥>情報投稿</h1>");
pw.println("<br>");
pw.println("投稿する情報のカテゴリを選んでください。もし、該当するカテゴリが無い場合は¥add¥をクリック
        してください<p>");
pw.println("情報をすべて書き込んだら送信ボタンを押してください);
pw.println("<form method=¥POST¥" action=BOADEv>");
pw.println("<input type=hidden name=cate value=write2>");
Enumeration catedata=catelist.elements();
while(catedata.hasMoreElements()){
String cateid=(String)catedata.nextElement();
String name=(String)catedata.nextElement();
pw.println("<input type=radio name=cateid value="+cateid+">"+name+");
}

pw.println("<br>");
pw.println("<a href=BOADEv?cate=addcate><h3>add</h3></a>");
pw.println("<br>");
pw.println("<table>");
pw.println("<tr><td align=¥right¥>タイトル : </td>");
pw.println("<td><input type=¥text¥" name=¥title¥" size=¥50¥></td></tr>");
pw.println("<tr><td align=¥right¥>場所 : </td><td>");
pw.println("<input type=¥text¥" name=¥place¥" size=¥50¥></td></tr>");
pw.println("<tr><td align=¥right¥>連絡先 : </td><td>");

```

```

pw.println("<input type='text' name='address' size='50'></TD></TR>");
pw.println("</TABLE><p>");
pw.println("コメント<br>");
pw.println("<textarea rows=8 cols=50 name='comment'>");
pw.println("</textarea><br>");
pw.println("<p>削除時のパスワード<input type=password name=password size=50><br>");
pw.println("<input type='submit' value='送信'>");
pw.println("<br>");
pw.println("</form></body></html>");

}

public void createStatus1(PrintWriter pw,Hashtable params)throws IOException,ServletException {
pw.println("<html><head>");
pw.println("<title>tnaiyou</title>");
pw.println("<body bgcolor='###fff'>");
pw.println("<h1>投稿ありがとうございました</h1>");
pw.println("<br><p>");
pw.println("投稿確かにお預り致しました。<br>");
pw.println("情報提供ありがとうございます。");
pw.println("<hr><br><p align=right><a href=BOADHTML?page=index></body></html>");
}

public void delete(HttpServletResponse response,Hashtable parameter)throws IOException,ServletException{
int commentid=Integer.parseInt((String)parameter.get("commentid"));
response.setContentType("text/html; charset=EUC-JP");
PrintWriter pw = response.getWriter();
pw.println("<html><head><title>削除用のパスワードを入力ください</title></head><body
    bgcolor=white><center><h1>削除用のパスワードを入力ください。</h1></center><br><br><form
    method=post action=BOADEv><input type=hidden name=cate value=delete2><input type=hidden
    name=commentid value="+commentid+"><center><input type=password
    name=password></center><br><center><input type=submit value=ok></form><br><hr><br><p
    align=right><a href=BOADHTML?page=index>トップへ</a></p></body></html>");
}

public void delete2(HttpServletResponse response,Hashtable parameter)throws IOException,ServletException{
int commentid=Integer.parseInt((String)parameter.get("commentid"));
String password=(String)parameter.get("password");
String passdata=BOADCDB.getPassword(commentid);
response.setContentType("text/html; charset=EUC-JP");
PrintWriter pw = response.getWriter();
BOADCDB.delete(commentid);
if(passdata.equals(password)){
pw.println("<html><head><title>削除しました</title></head><body bgcolor=white><center><h1>削除完了しまし
    ました。</h1></center><br><hr><br><p align=right><a href=BOADHTML?page=index>トップへ
    </a></P></body></html>");
}
else{
pw.println("<html><head><title>パスワードが違います。</title></head><body bgcolor=white><center><h1>
    パスワードが違います。</h1></center><br><center><h2>トップへお戻りください。削除にはパスワ
    ードが必要です。</h2></center><br><hr><br><p align=right><a href=BOADHTML?page=index>トッ
    プへ</a></p></body></html>");
}
}

private void makeErrorpage(PrintWriter pw)throws IOException{
pw.println("<html><head><title>情報が不足しています。</title></head><body bgcolor=white><center><h1>情報
    が不足しています。</h1></center><br>もう一度ご記入ください。<br><hr><br><p align=right><a
    href='JavaScript:history.back()'>戻る</a></p></body></html>");
}

/**
 *      getCookieValue
 *      クッキーゲット
 *
 */

```

```

*/

private Cookie getCookieValue(HttpServletRequest request,String cookieName){
Cookie cookies[]=null,cookie=null;

if((cookies=request.getCookies())!=null){
for(int i=0;i<cookies.length;i++){
if(cookies[i].getName().equals(cookieName)){
synchronized(this){
cookie=(Cookie)cookies[i].clone();
}
return cookie;
}
}
}
return null;
}

/**
getNowDate
現在時刻を年/月/日 時:分にする
*/
private String getNowDate()
{Calendar rightNow=Calendar.getInstance();
int yearInt=rightNow.get(Calendar.YEAR)-2000;
String year=checkInt(yearInt);
int monthInt=rightNow.get(Calendar.MONTH)+1;
String month=checkInt(monthInt);
int dateInt=rightNow.get(Calendar.DATE);
String date=checkInt(dateInt);
int hourInt=rightNow.get(Calendar.HOUR_OF_DAY);
String hour=checkInt(hourInt);
int minuteInt=rightNow.get(Calendar.MINUTE);
String minute=checkInt(minuteInt);
String dateNow="" +year+"/"+month+"/"+date+" "+hour+":"+minute+"";
System.out.println(dateNow);
return dateNow;
}

/**
checkInt 数値が1桁の時0を頭につける
*/

private String checkInt(int numbers){
String stringNumber=null;
if(numbers<10)
{String stringNumbera=String.valueOf(numbers);
stringNumber="0"+stringNumbera;}
else{stringNumber=String.valueOf(numbers);}
return stringNumber;

}
}

```

## 4-4【個人/グループスケジュール管理ツール】

〈〈文責：矢島達也、黒田洋史

### ①作成理由

ミーティングや飲み会などの予定を立てるために、グループのメンバー全員にメールを出したり、電話をかけたたりしなければならない。さらに悪いことに、返事がなかったりして、スムーズに予定を立てることができないことがよくある(少なくとも私たちの間ではそのようなことがある。) 这样的问题を解決するために、このツールを作成した。

### ②ツールの機能

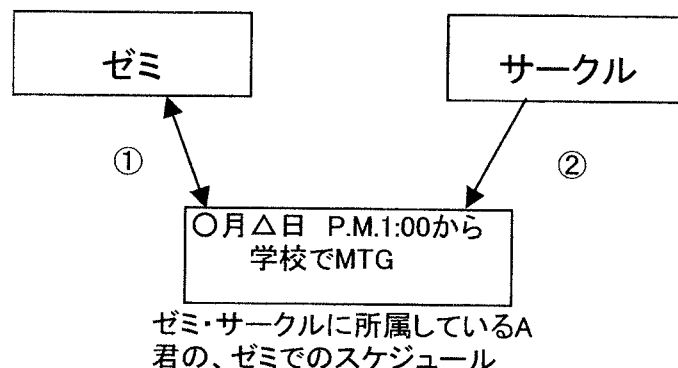
#### i)個人用スケジュール管理ツール

個人のスケジュールを管理する。自分のスケジュールを入力すると、入力された情報は一覧表となって反映される。

#### ii)グループ用スケジュール管理ツール

グループのスケジュールを管理する。基本的な機能は個人用スケジュール管理ツールと同様だが、以下の2つの機能が付け加えられている。

- ・個人用スケジュール管理ツールで入力された情報はグループ用スケジュール管理ツールにも反映される(プライバシー保護のため「予定がある」とだけ表示される)ので、グループ各メンバーの都合の悪い日を確認でき、予定を立てることが簡単になる。
- ・複数のグループに入っている場合、あるグループのスケジュールは、他のグループにとっては関係のない情報だ。



上の図を例にして説明する。

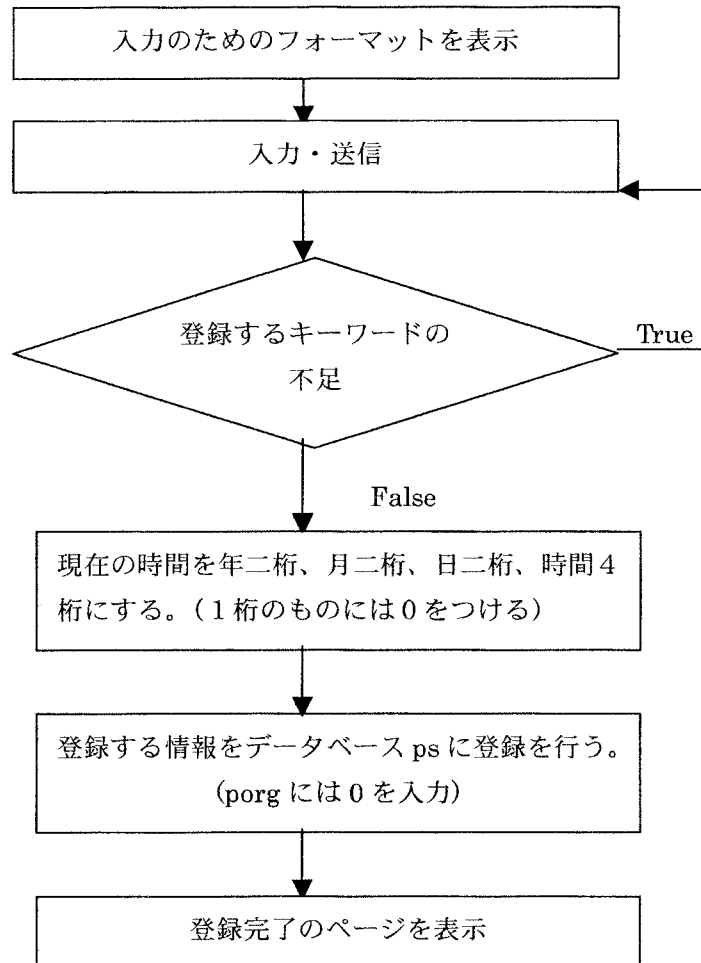
- ①ゼミのスケジュールなのでそのまま表示される。
- ②サークルのメンバーから見ればゼミでのスケジュールはA君自身のスケジュール

なので、個人の予定として、サークルのスケジュールには「予定があります」と表示される。

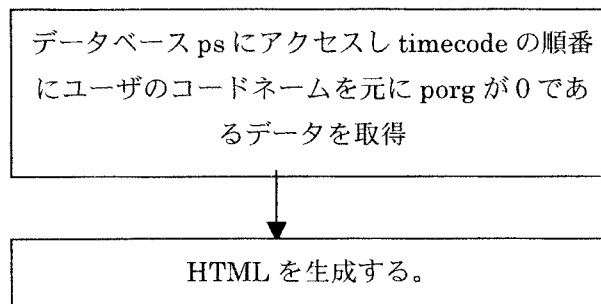
### ③アルゴリズム

(個人スケジュール)

i) 入力画面

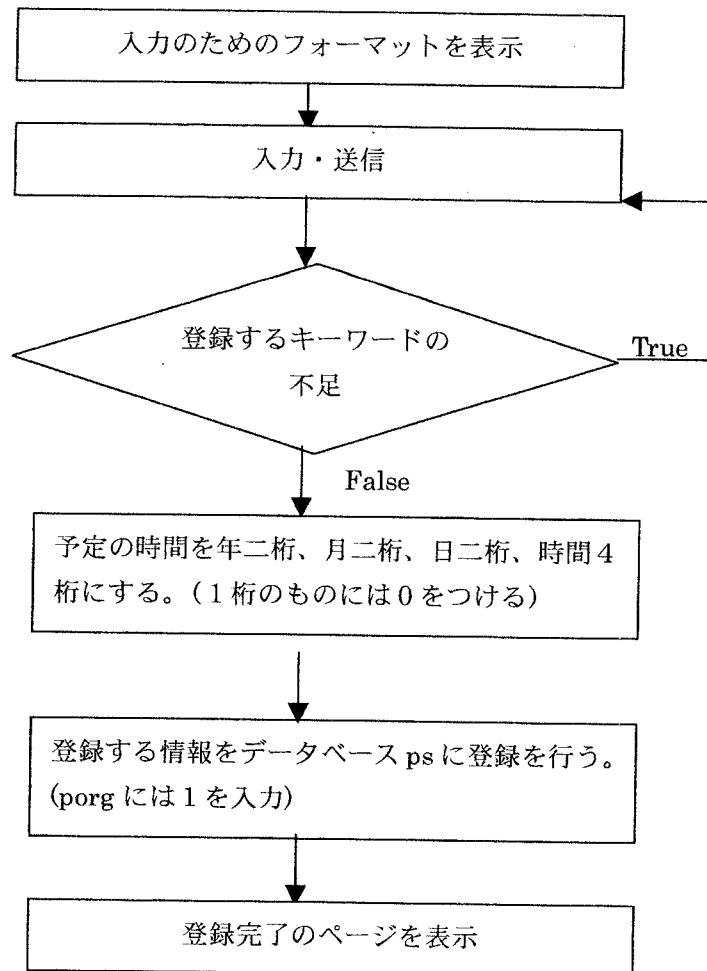


ii) 個人予定表示

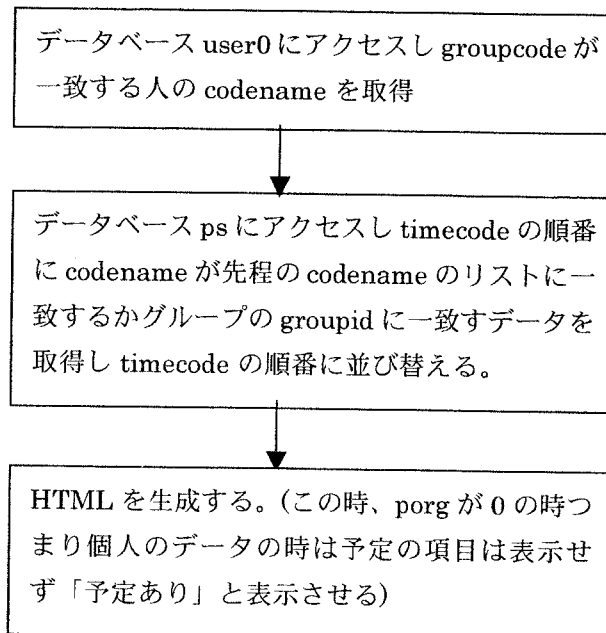


(グループスケジュール)

i) 入力画面



ii) グループ予定表示



#### ④作成に使った技術

最初は Swing を使って個人用のスケジュール管理ツールを作成したのだが、今回は双方向的、動的なツールの作成が主な目標だったので、Servlet を使って作成し直した。

#### ⑤使用したデータベース

(テーブル名:ps)

カラム名	pscode	codename	year	month	day
データ型	Int 型	Text 型	Text 型	Text 型	Text 型
詳細	ナンバリング する	個人 → codename グループ → groupid	年	月	日にち

hour	minute	Title	content	timecode
Text 型	Text 型	Text 型	Text 型	Text 型
時間	分	題名	詳細	時間 (年 2 桁 月 2 桁、時間 四桁)

nowdate	porg	Groupcode
Text 型	Text 型	Text 型
登録した時間	0 → 個人 1 → グループ	所属している グループの groupid

#### ⑥今後の改良ポイント

- ・現在は反映す式が一覧表となっているが、これをカレンダーの日にちをクリックするとスケジュールを確認できる形にしたい。
- ・個人用スケジュール管理ツールの自由度を高くしたい。日記やメモ帳としての使ったりと、スケジュール入力だけでなく色々な方向へ使っていければよいと思う。その際、ツールに切替作用をつけて区別を付けられればよいと思う。(グループのスケジュールに反映させるか否か)
- ・あのひと Finder との連携。

#### ⑦製作体験記

Servlet はまだ新しい技術で難しい面も多々あって苦しんだ。しかし、「産みの苦しみ」を経験できたということは自分にとって大きかった。その一方で、「企画・設計の楽しみ」も経験した。「楽しみ」と「苦しみ」の両方を経験したことは、今後の自分にとって大いにプラスになると思う。

Java 学習歴 2 ヶ月弱の自分にとって、経験不足がこのツールを作るのに大きく影響し

た。どこから手をつけていいのかわからず、結局書籍に頼らざるを得ない状況だった。最初に体験した Swing は自分と相性が合っていたので、理解もしやすく、いいものを完成させることができたが、Servlet(今回の最終形)の方は本を読んでも全く理解できず(もともと本自体の内容が難しすぎたこともあるが)、できることと言えば、サンプルを真似て、少し応用することくらいだった。ツールの作成を通じて、理想と現実の溝を埋めるための技術の足りなさを感じ、また、自分の頭の中の理想をそのまま形にすることの難しさも同時に感じた。(矢島)

入力したデータは普通ならば入力した順番に表示されるが、それを予定している時間の順番に並び替え表示するにはどのようなアルゴリズムで表示するかを決めるのが大変だった。予定している時間を8桁の数字にしてデータベースに格納し保存した。実際に表示する時はその数字を順番に並び替えることで順番に表示させるようにした。

このスケジュール管理がグループと個人の両方で共用できる唯一のツールだった。データの整合性をどのようにとるべきかに悩んだが、ひとつのデータベースのテーブルを用い個人のデータかグループのデータかを区別するための番号を保存することで対応させた。この方法は便利だったので、他のツールでも生かせるのではないだろうか。(黒田)

(参考：表示画面)

#### グループスケジュールの一覧

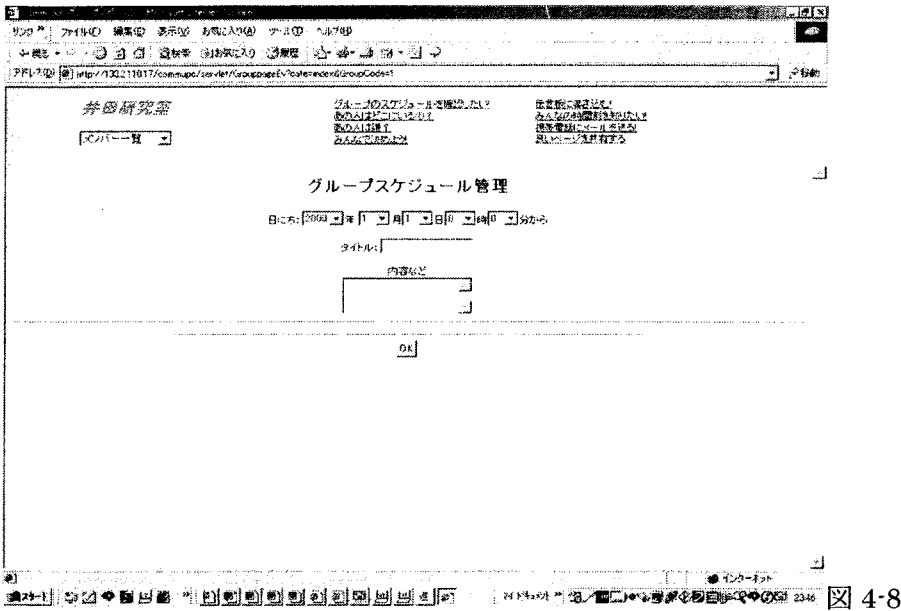
The screenshot shows a web browser window with the URL `http://130211017/comapp/servlet/GroupappE?orderCode=0`. The page content includes a header for '井田研究室' (Iida Laboratory) and a main title 'グループ予定一覧'. The table below lists the following data:

日時	担当	内容
10月1日16時00分	金典	11月22日
10月1日14時45分	金典	総会議
10月11日12時30分	金典	2007年度
10月26日14時30分	tsukuro	予定がありません。
10月26日06時00分	arrow	予定がありません。
10月26日14時00分	金典	岡山県立美術館にて展示
10月29日11時00分	arrow	予定がありません。
10月29日16時30分	tsukuro	予定がありません。
10月29日16時40分	tsukuro	予定がありません。
10月30日01時28分	金典	お話し
10月30日12時45分	tsukuro	予定がありません。
10月30日13時00分	金典	2007年度
10月30日19時00分	金典	打ち上げ
11月08日16時00分	arrow	予定がありません。
11月11日12時30分	arrow	予定がありません。
11月11日12時20分	金典	打ち上げ
11月14日06時30分	tsukuro	予定がありません。

グループの予定を登録込む



## グループスケジュール登録



(参考 : Java ソース)

```

/*
 * @(#) PSCenter.Java
 *
 * PlanSheet
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 */
import Java.io.*;
import Java.util.*;
import Javax.servlet.*;
import Javax.servlet.http.*;

public class PSCenter extends HttpServlet {
    PSConnectDB PSCDB=new PSConnectDB();
    public void write(HttpServletRequest request,HttpServletResponse response,Hashtable parameters) throws
    ServletException, IOException {
        response.setContentType("text/html; charset=euc-jp");
        PrintWriter pw = response.getWriter();

        try {
            createForm(pw, request);
        } catch (Exception e) {
            e.printStackTrace();
            throw new ServletException(e.toString());
        }
        pw.flush();
        pw.close();
    }

    public void write2(HttpServletRequest request,HttpServletResponse response,Hashtable parameters)
    throws ServletException, IOException,NumberFormatException {

        response.setContentType("text/html; charset=euc-jp");
        PrintWriter pw = response.getWriter();
        Cookie Codenamecookie=getCookieValue(request,"Codename");
    }
}

```

```

String Codename=Codenamecookie.getValue();
String year=(String)parameters.get("year");
String month=(String)parameters.get("month");
String day=(String)parameters.get("day");
String hour=(String)parameters.get("hour");
String minute=(String)parameters.get("minute");
String title=(String)parameters.get("title");
String content=(String)parameters.get("content");
long timecode=Long.parseLong(year+month+day+hour+minute);
String nowdate=getNowDate();
int porg=Integer.parseInt((String)parameters.get("porg"));
if(title.equals("") | content.equals("")){
    pw.println("<html><head><title> 情報 が 足 り ま せ ん </title></head><body
bgcolor=white><center><h1> 情報 が 足 り ま せ ん </h1></center><br><center><a
href=¥\"JavaScript:history.back()¥\">ここを</a>クリックして記入画面に戻ってください</body></html>");}
else{
    try {
        PSCDB.write2(Codename,year,month,day,hour,minute,title,content,timecode,nowdate,porg);
        pw.println("<html><head>");
        pw.println("<title>あなたの予定を登録しました。 </title></head>");
        pw.println("<body bgcolor = ¥\"#ffff¥\">");
        pw.println("<center><h2>あなたの予定を登録しました。 </h2></center>");
        pw.println("<br><hr width><br><p align=right><a href=PSEv?cate=look> ト ッ プ へ
</a></body></html>");

    } catch (Exception e) {
        e.printStackTrace();
        throw new ServletException(e.toString());
    }
    pw.flush();
    pw.close();
}

}

public void look(HttpServletRequest request,HttpServletResponse response,Hashtable parameters)throws
IOException,ServletException{
    response.setContentType("text/html; charset=euc-jp");
    PrintWriter pw = response.getWriter();
    Cookie Codenamecookie=getCookieValue(request,"Codename");
    String Codename=Codenamecookie.getValue();
    Vector planlist=PSCDB.getPlanlist(Codename);
    setPlanlist(planlist,pw);
}

private Cookie getCookieValue(HttpServletRequest request,String cookieName){
    Cookie cookies[]=null,cookie=null;

    if((cookies=request.getCookies())!=null){
        for(int i=0;i<cookies.length;i++){
            if(cookies[i].getName().equals(cookieName)){
                synchronized(this){
                    cookie=(Cookie)cookies[i].clone();
                }
                return cookie;
            }
        }
    }

    return null;
}

private void createForm(PrintWriter pw, HttpServletRequest request)
throws IOException {

    pw.println("<html><head>");
    pw.println("<title>個人スケジュール管理</title>");
    pw.println("<body>");
    pw.println("<center><h2>個人スケジュール管理</h2>");
    pw.println("<p>");
    pw.println("<form method = ¥\"post¥\" action=¥\"PSEv?cate=write2¥\">");

```

```

        pw.print("日にち : ");
        pw.println("<select name=year><option value=2000>2000<option value=2001>2001</select>年");
        pw.println("<select name=month><option value=01>1<option value=02>2<option value=03>3<option
value=04>4<option value=05>5<option value=06>6<option value=07>7<option value=08>8<option
value=09>9<option value=10>10<option value=11>11<option value value=12>12</select> 月 <select
name=day>");
        for(int i=1;i<10;i++){
            pw.println("<option value=0"+i+">"+i+"");
        }
        for(int i=10;i<=31;i++){
            pw.println("<option value="+i+">"+i+"");
        }

        pw.println("</select>日<select name=hour>");
        for(int j=0;j<10;j++){
            pw.println("<option value=0"+j+">"+j+"");
        }
        for(int l=10;l<=24;l++){
            pw.println("<option value="+l+">"+l+"");
        }
        pw.println("</select>時<select name=minute>");
        for(int k=0;k<10;k=k+5){
            pw.println("<option value=0"+k+">"+k+"");
        }
        for(int m=10;m<60;m=m+5){
            pw.println("<option value="+m+">"+m+"");
        }

        pw.println("</select>分から<br><br>");
        pw.println("タイトル : <input type=text name=title><br><br>");
        pw.print("内容など");
        pw.println("<br>");
        pw.println("<textarea rows = 3 cols = 20 name = ¥\"content¥\">");
        pw.println("</textarea>");
        pw.println("<hr>");
        pw.println("<hr width =60%>");
        pw.println("<input type=hidden name=porg value=0>");
        pw.println("<input type = ¥\"submit¥\" value = ¥\"OK¥\">");
        pw.println("</form></center>");
        pw.println("</body></html>");
    }

```

```

public void setPlanlist(Vector planlist,PrintWriter pw)throws IOException{
    pw.println("<html><head><title>予定一覧</title></head><body>");
    pw.println("<body bgcolor=white><center><h1>予定一覧</h1></center><br>");

    pw.println("<table border=1><tr><td width=15%>");
    pw.println("日時</td>");
    pw.println("<td width=50%>内容");
    pw.println("</td></tr>");

    Enumeration enum=planlist.elements();
    while(enum.hasMoreElements()) {
        String pcode=(String)enum.nextElement();
        String month = (String)enum.nextElement();
        String day = (String)enum.nextElement();
        String hour = (String)enum.nextElement();
        String minute =(String)enum.nextElement();
        String title=(String)enum.nextElement();
        pw.println("<tr><td>");
        pw.println(""+month+" 月 "+day+" 日 "+hour+" 時 "+minute+" 分 </td><td><a
href=PSEv?cate=detail&pcode="+pcode+">"+title+"</a></td></tr>");
    }
    pw.println("</table>");
    pw.println("<hr width = 60%>");

```

```

        pw.println(" タイトルをクリックすると詳細が見れますよ。 <br><p align=right><h3><a
href=PSEv?cate=write>書き込む</a></h3></p></body></html>");
    }

```

```

    public void detail(HttpServletRequest request, HttpServletResponse response, Hashtable
parameters)throws IOException,ServletException{
        response.setContentType("text/html; charset=euc-jp");
        PrintWriter pw = response.getWriter();
        Cookie Codenamecookie=getCookieValue(request,"Codename");
        String Codename=Codenamecookie.getValue();
        int pscod=Integer.parseInt((String)parameters.get("pscode"));
        Vector detail=PSCDB.getDetail(pscod);
        Enumeration detaildata=detail.elements();

        String month=(String)detaildata.nextElement();
        String day=(String)detaildata.nextElement();
        String hour=(String)detaildata.nextElement();
        String minute=(String)detaildata.nextElement();
        String title=(String)detaildata.nextElement();
        String content=(String)detaildata.nextElement();

        pw.println("<html><head><title>予定詳細</title><head><body bgcolor=white><center><h1>予定詳細
</h1><br><table border=1><tr width=60%><td>"+month+" 月 "+day+" 日 "+hour+" 時 "+minute+" 分の予定
</td></tr><tr><td><center><h2>"+title+"</h2></center></td></tr><tr><td>"+content+"</td></tr></table><br
><br><hr><br><p align=right><a href=PSEv?cate=look>戻る</a></body></html>");
    }

```

```

/**
 * getNowDate
 * 現在時刻を年/月/日 時:分にする
 */
private String getNowDate()
{Calendar rightNow=Calendar.getInstance();
int yearInt=rightNow.get(Calendar.YEAR)-2000;
String year=checkInt(yearInt);
int monthInt=rightNow.get(Calendar.MONTH)+1;
String month=checkInt(monthInt);
int dateInt=rightNow.get(Calendar.DATE);
String date=checkInt(dateInt);
int hourInt=rightNow.get(Calendar.HOUR_OF_DAY);
String hour=checkInt(hourInt);
int minuteInt=rightNow.get(Calendar.MINUTE);
String minute=checkInt(minuteInt);
String dateNow=""+"year+"/"+"month+"/"+"date+" "+hour+":"+minute+"";
System.out.println(dateNow);
return dateNow;
}

```

```

/**
checkInt      数値が1桁の時0を頭につける
*/

```

```

private String checkInt(int numbers){
String stringNumber=null;
if(numbers<10)
    {String stringNumbera=String.valueOf(numbers);
    stringNumber="0"+stringNumbera;}
else{stringNumber=String.valueOf(numbers);}
return stringNumber;
}
}

```

## 4-5 【LIDA (Library of IDA)】

◀文責：黒田洋史

### ①作成理由

図書館に本を捜しに行く前に、目的の本を借りることができるのか、保管されている場所はどこかなどを検索できれば、時間を有効に使えるのではないかと思い作成した。

### ② 機能

今回は井田先生の蔵書を管理することにする。

#### i) 蔵書一覧

保管している本を一覧することでキーワードがあいまいな場合でも検索できるようにする。

#### ii) 蔵書検索

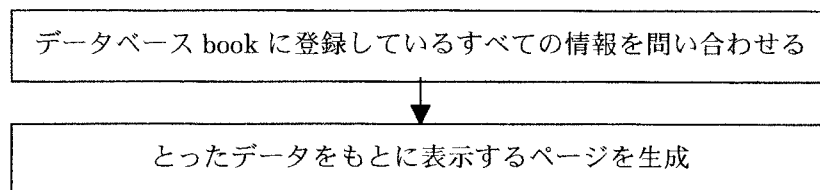
保管している本をユーザが入力したキーワードを元に検索する機能である。

#### iii) 新図書登録

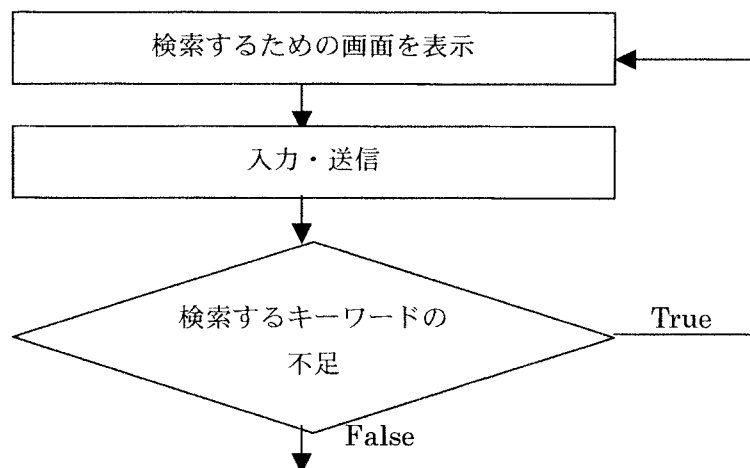
新しく購入した本の情報を登録する機能で管理者用のパスワードが必要になる。

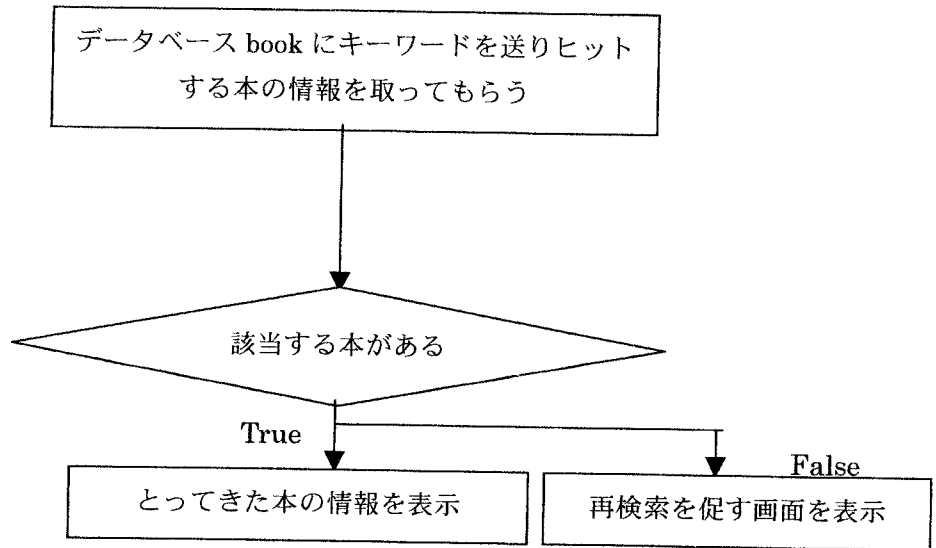
### ③アルゴリズム

#### i) 蔵書一覧

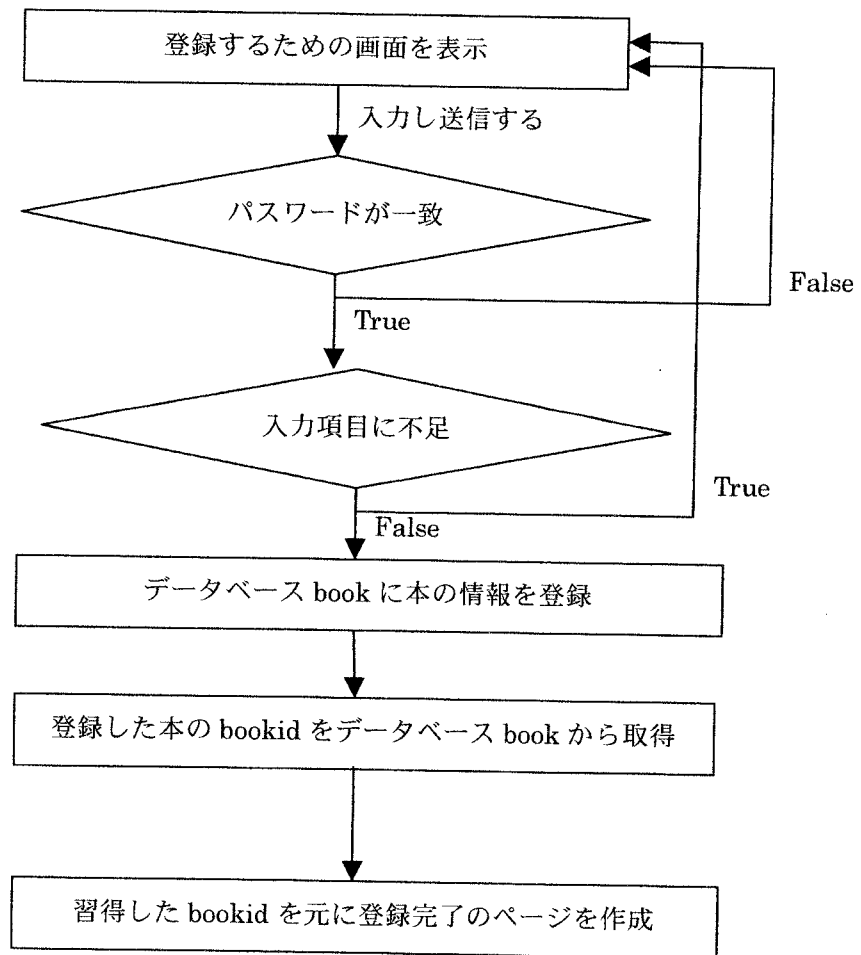


#### ii) 蔵書検索

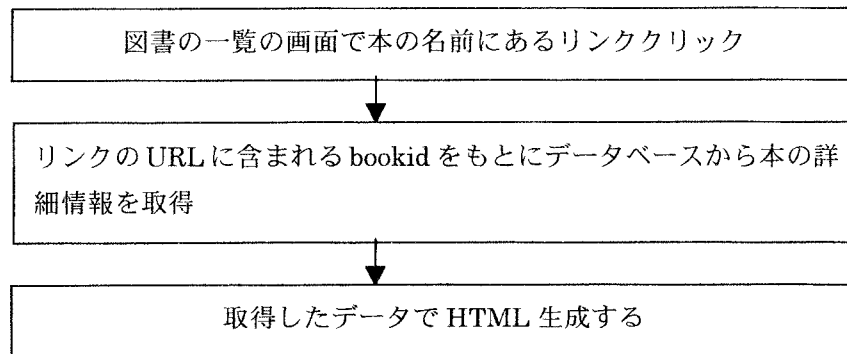




iii) 新図書登録



#### iv) 図書詳細表示



#### ④作成に使った技術

JavaServlet 及びリレーショナルデータベース

#### ⑤データベース詳細

(テーブル名 : book)

カラム名	bookid	japanese	kindofbook	category	bookname
データ型	Int 型	Int 型	Int 型	Text 型	Text 型
詳細	ナンバリ ング する	和書→0 洋書→1	図書→1 雑誌→2 他→3	本のキー ワードを 入れてお く	本の題名
	publisher	writer	publishedyear	info	nowdate
	Text 型	Text 型	Text 型	Text 型	Text 型
	出版社名	著者	出版年	備考	登録した時間 を記録

#### ⑥改良すべき点

- ・本の貸し出し状況の管理と利用状況の管理ができる機能を設ける。
- ・本を返し忘れていた人へ電子メールでコンタクトを取れる機能を設ける。
- ・本の感想などを交換できる掲示板のような機能を設ける。
- ・登録する時に本のジャンルをリストから選んでもらい、それを検索に生かすようにする。
- ・新図書登録の際、パスワードを管理者に入力してもらったが、それを変更できるように作っていないので変更できるようにする。

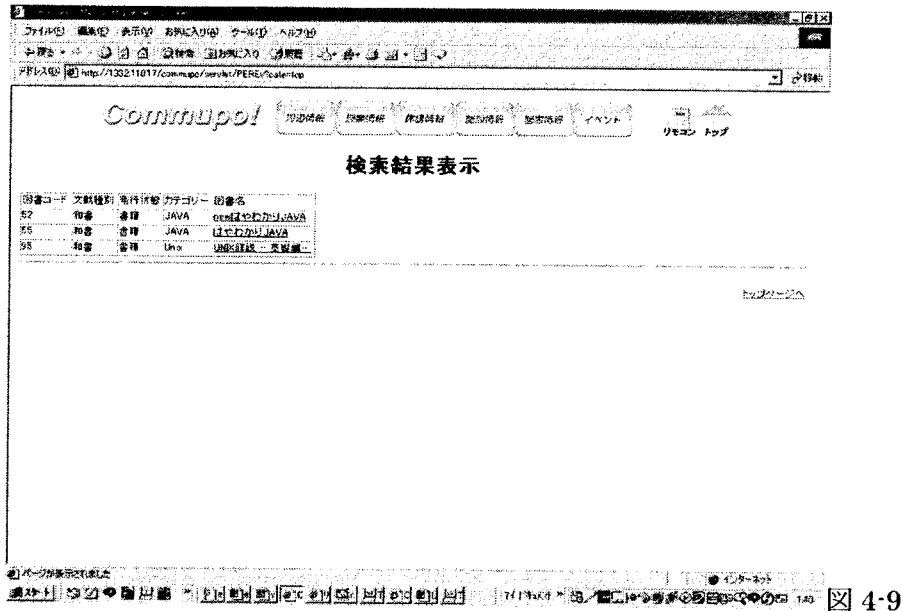
#### ⑦製作体験記

はじめは井田文庫という名で呼ばれていたが、これでは恰好がつかないので

Lida(Library of IDA)という名前に変えた。名前を変えたのは、出来上がったあとでのことだが・・・。これは井田先生の蔵書を整理管理するためのツールである。蔵書の調査はritzとsussyが引き受けてくれていたので、僕はそれを格納するプログラムを作成するだけであり、作業は比較的楽だった。青山キャンパス、厚木キャンパスに渡る先生の蔵書を1冊1冊記録した彼女らは苦勞したはずだ。このツールを通して初めて取り組んだのは検索だ。まったく同じのキーワードを探すこと(完全検索)は想像できたが、一部分がヒットすれば表示する検索(部分一致検索)はなかなか想像できなかった。先輩に聞いて調べると、「select<column> from <table> where <column> like % word %」とすればいいことが分かった。困ったときはサイトをチェックすればよく、そういう意味で自分の辞書、つまり困った時の神頼みページを持つ必要があるのに気付いた。Lidaに関してはデバックに苦勞することなく出来上がった記憶がある。改良の余地はあったのだが、そこで改良を後回しにしてしまった。結局、改良はできないまま青山祭を迎えてしまい、とても残念だった。

(参考・画面表示)

検索結果一覧の画面







```

        response.sendRedirect("../login.html");
    }
    else{
        int BookID=Integer.parseInt((String)parameter.get("BookID"));
        LIDABookData BookDetailData=LIDADB.getData(BookID);
        LIDAU.makeDetail(response,BookDetailData);
    }
}

public void datasearch(HttpServletRequest request,HttpServletResponse response,Hashtable parameter)
throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        String key=((String)parameter.get("key"));
        String word=((String)parameter.get("word"));
        if(key.equals("nondata")|word.equals(""))
        {LIDAU.searchError(response);
        }

        else{Vector resultdata=LIDADB.search(key,word);
        if(resultdata.size()==0){
            LIDAU.searchNondata(response);
        }
        else{LIDAU.search(response,resultdata);}
        }
    }
}

public void inputBookData(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter) throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        String Japanese=(String)parameter.get("Japanese");
        String Kindofbook=(String)parameter.get("Kindofbook");
        String Category=(String)parameter.get("Category");
        String Bookname=(String)parameter.get("Bookname");
        String Publisher=(String)parameter.get("Publisher");
        String password=(String)parameter.get("password");
        if(!password.equals("masayuki"))
        {LIDAU.passwordFailed(response);}
        else{

            if(Japanese.equals("") | Kindofbook.equals("")|Category.equals("") | Bookname.equals("")
|Publisher.equals(""))
            {LIDAU.searchError(response);}
            else{
                LIDABookData LIBD =new LIDABookData();
                LIBD.setJapanese(Integer.parseInt((String)parameter.get("Japanese")));
                LIBD.setKindofbook(Integer.parseInt((String)parameter.get("Kindofbook")));
                LIBD.setCategory((String)parameter.get("Category"));
                LIBD.setBookname((String)parameter.get("Bookname"));
                LIBD.setPublisher((String)parameter.get("Publisher"));
                LIBD.setWriter((String)parameter.get("Writer"));
                LIBD.setPublishedyear((String)parameter.get("Publishedyear"));
                LIBD.setInfo((String)parameter.get("Info"));
                String nowdate=getNowDate();
                LIBD.setNowdate(nowdate);
                int newbookID=LIDADB.inputBookData(LIBD);
                LIDAU.inputedBookData(newbookID,response);
            }
        }
    }
}

}
/**

```

```

*getNowDate
*現在時刻を年/月/日 時:分にする
*/

private String getNowDate(){
    Calendar rightNow=Calendar.getInstance();
    int yearInt=rightNow.get(Calendar.YEAR)-2000;
    String year=checkInt(yearInt);
    int monthInt=rightNow.get(Calendar.MONTH)+1;
    String month=checkInt(monthInt);
    int dateInt=rightNow.get(Calendar.DATE);
    String date=checkInt(dateInt);
    int hourInt=rightNow.get(Calendar.HOUR_OF_DAY);
    String hour=checkInt(hourInt);
    int minuteInt=rightNow.get(Calendar.MINUTE);
    String minute=checkInt(minuteInt);
    String dateNow=""+"year+"/"+month+"/"+date+" "+hour+":"+minute+"";
    System.out.println(dateNow);
    return dateNow;
}

/**
 *
 *checkInt      数値が1桁の時0を頭につける
 *
 */

private String checkInt(int numbers){
    String stringNumber=null;
    if(numbers<10)
    {String stringNumbera=String.valueOf(numbers);
    stringNumber="0"+stringNumbera;}
    else{stringNumber=String.valueOf(numbers);}
    return stringNumber;
}
}
}

```

## 4-6 【グループリンク共有】

＜＜文責：黒田洋史

### ①作成理由

グループで、有用なホームページなどを共有・紹介したいケースが多くある。この時、メールを使って教えるのが一般的だが、メールを見ることができない場合もある。そこで、ブラウザさえあればどこでも見られるような方法が必要であると考え、「こみゅぽ」のグループのページ内でチェックできればと思い作成した。

### ②機能

#### i) リンク登録

リンクを登録する機能。

#### ii) リンク一覧

リンク登録をしたページを一覧にしたもので、そのリンクをたどった回数をカウ

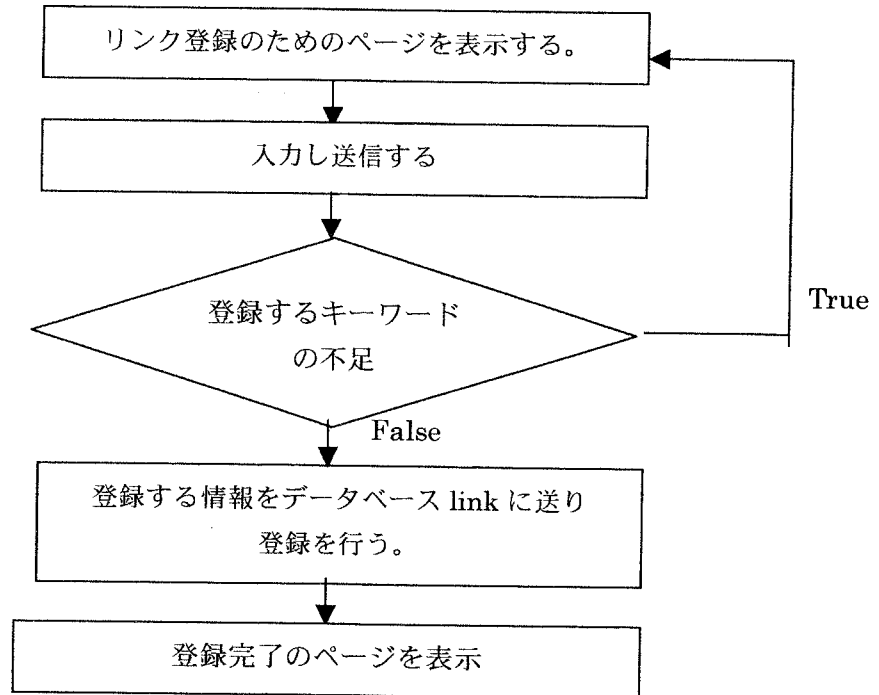
ントし、アクセス数の多いものから順に並べたものである。

iii) アクセスカウント

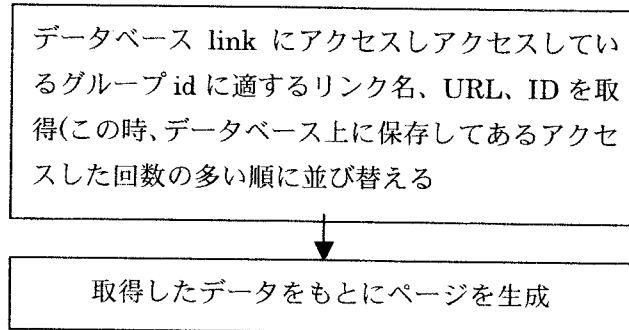
リンク一覧からアクセスした(リンクをクリックした)回数をカウントしデータベースに保存する機能である。

③アルゴリズム

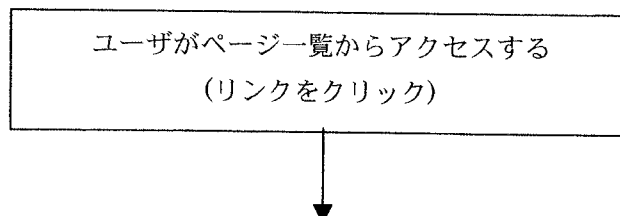
i) リンク登録

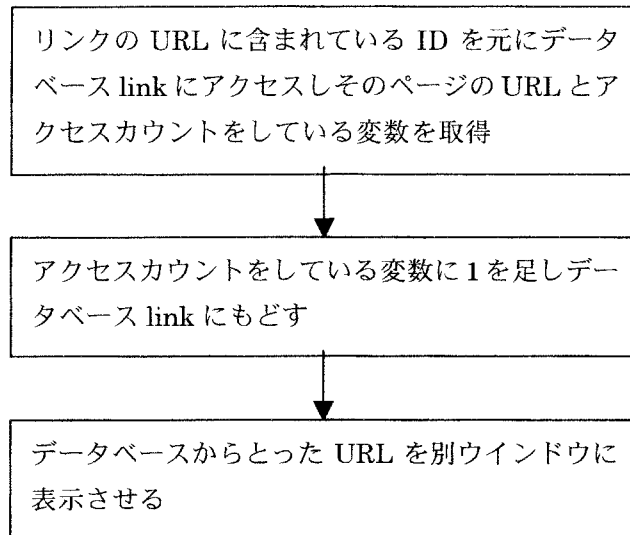


ii) リンク表示



iii) アクセスカウント





#### ④作成に使った技術

JavaServlet 及びリレーショナルデータベース

#### ⑤データベース詳細

(テーブル名 : link)

カラム名	linkid	groupid	codename	linkname	url
データ型	Int 型	Int 型	Text 型	Text 型	Text 型
詳細	ナンバリングする	グループの ID を保存する	登録した人の codename を保存する	リンク先の名前を保存する	リンク先の URL を保存

point	deleted	nowdate
Int 型	Int 型	Text 型
リンク一覧からのアクセス回数を保存	0→未削除 1→削除済み (表示しない)	登録時の時間を記録する

#### ⑥改良すべき点

- ・ リンクの削除ができる機能を設ける。
- ・ カテゴリ別に登録・表示できる機能を設ける。
- ・ リンク先のページの感想等を交換できる機能を設ける。

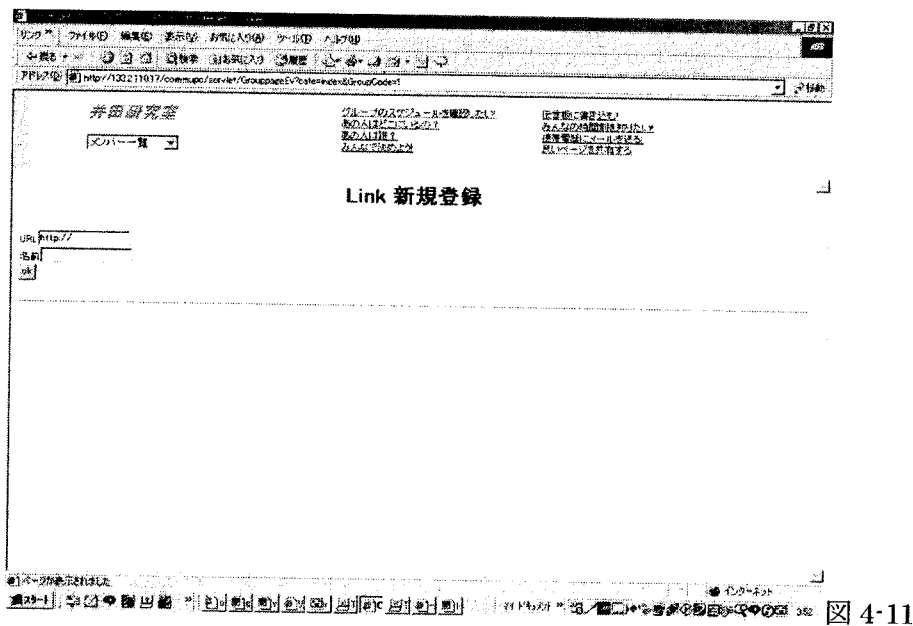
## ⑦製作体験記

お気に入りのリンクをコミュニティで共有しようという考え方からできたものである。ここまでだったらよくあるものと変わらない。そこで、自分でないとできない機能を加えようと思った。それがランキング機能である。この機能は、アクセスが多い順番にリンクを並び替える機能である。本当によいものを共有するには絶対に必要であり、まずはアルゴリズムを考えた。そして、リンクがクリックされた回数を得点化してデータベースに保存し、クリック数に応じて得点が増えるというアクセス回数を利用する方法をとることにした。しかし、問題になったのがそのアクセス回数の記録であった。これを、データベースの得点を読み込んで、その得点に1を足しデータベースに再び戻すようにすることで解決した。

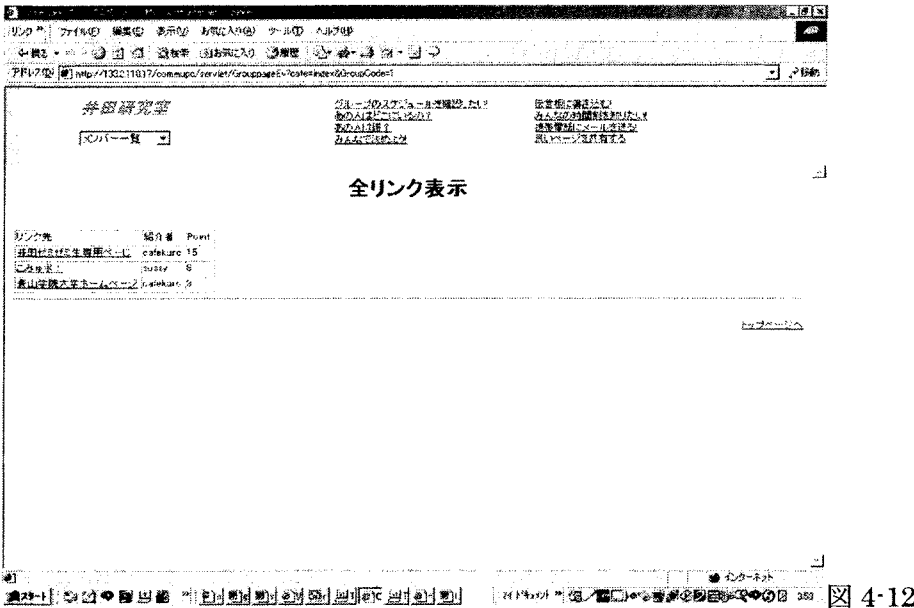
この作業を通して、色々試しながら1つの結果にいたるというプロセスを、そして、目標としていた機能を作り上げる喜びも学ぶことができた。

## (参考・画面表示)

### リンク登録画面



## リンク表示画面



(参考 : Java ソース)

```
/*
 * @(#) LinkControl.java
 *
 * Link share tool
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 *
 */

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class LinkControl{

    LinkConnectDB LinkDB=new LinkConnectDB();
    LinkUi LinkU=new LinkUi();
    Center center=new Center();
    public void regist(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter) throws IOException,ServletException{
        boolean checkLogs=center.checkLogs(request,response);
        if(!checkLogs){
            response.sendRedirect("../User/login.html");
        }
        else{
            /**Cookie GCookie=getCookieValue("GroupID",request);
            int GroupID=Integer.parseInt(GCookie.getValue());*/
            int GroupID=Integer.parseInt((String)parameter.get("GroupCode"));
        }
    }
}
```

```

        Cookie CCookie=getCookieValue("Codename",request);
        String Codename=CCookie.getValue();
        String groupcode=String.valueOf(GroupID);
        String Linkname=(String)parameter.get("Linkname");
        String URL=(String)parameter.get("URL");
        String Nowdate=getNowDate();
        if(Linkname.equals("") | URL.equals("")){
            LinkU.nonData(response,groupcode);
        }
        LinkData LD=new LinkData();
        LD.setGroupID(GroupID);
        LD.setCodename(Codename);
        LD.setLinkname(Linkname);
        LD.setURL(URL);
        LD.setNowdate(Nowdate);
        LinkDB.regist(LD);
        LinkU.finishRegist(response,groupcode);
    }
}

/**
 *   getCookieValue
 *       必要な Cookie をもってくる。
 */

private Cookie getCookieValue(String name,HttpServletRequest request){
    Cookie cookies[]=null,cookie=null;

    if(cookies=request.getCookies()!=null){
        for(int i=0;i<cookies.length;i++){
            if(cookies[i].getName().equals(name)){
                synchronized(this){
                    cookie=(Cookie)cookies[i].clone();
                }
                return cookie;
            }
        }
    }
    return null;
}

/**
 *   getNowDate
 *       現在時刻を年/月/日 時:分にする
 */
private String getNowDate()
{Calendar rightNow=Calendar.getInstance();
int yearInt=rightNow.get(Calendar.YEAR)-2000;
String year=checkInt(yearInt);
int monthInt=rightNow.get(Calendar.MONTH)+1;
String month=checkInt(monthInt);
int dateInt=rightNow.get(Calendar.DATE);
String date=checkInt(dateInt);
int hourInt=rightNow.get(Calendar.HOUR_OF_DAY);
String hour=checkInt(hourInt);
int minuteInt=rightNow.get(Calendar.MINUTE);
String minute=checkInt(minuteInt);
String dateNow="" +year+"/"+month+"/"+date+" "+hour+": "+minute+"";
System.out.println(dateNow);
}

```



```

return dateNow;
}

/**
 * checkInt 数値が1桁の時0を頭につける
 */

private String checkInt(int numbers){
    String stringNumber=null;
    if(numbers<10)
        {String stringNumbera=String.valueOf(numbers);
        stringNumber="0"+stringNumbera;}
    else{stringNumber=String.valueOf(numbers);}
    return stringNumber;
}

/**
 * list リンクリスト作成
 */

public void list(HttpServletRequest request,HttpServletRequest response,Hashtable
parameter) throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../User/login.html");
    }
    else{
        /** Cookie GCookie=getCookieValue("GroupID",request);
        int GroupID=Integer.parseInt(GCookie.getValue());*/
        int GroupID=Integer.parseInt((String)parameter.get("GroupCode"));
        String groupcode=String.valueOf(GroupID);
        Vector LinkData=LinkDB.list(GroupID);
        LinkU.list(LinkData,response,groupcode);
    }
}

/**
 * connect
 * リンク接続
 */

public void connect(HttpServletRequest response,Hashtable parameter)throws
IOException,ServletException{
    String SlinkID=(String)parameter.get("LinkID");
    int LinkID=Integer.parseInt(SlinkID);
    LinkData LD=LinkDB.getLinkData(LinkID);
    int NowPoint=LD.getPoint();
    String URL=LD.getURL();
    LD.setPoint(NowPoint+1);
    LinkDB.setLinkData(LD,LinkID);
    response.sendRedirect(URL);
}
}

```

## 4-7【グループ投票管理】

≪文責：黒田洋史

### ①作成理由

グループで予定を立てる場合、全員がそろって決めることはなかなか難しい。メールを利用するという方法もあるが、メールを書くのを面倒がる人もいて、いい方法とはいえない。そこで、手軽に、かつ簡単に利用できる WEB 上で決めることができればと思い作成した。

### ②機能

#### i) 投票項目作成

誰でも簡単に質問事項を作れるように、ページごとにその手順を示すようにした。質問内容も、選択式と記述式の2つを用意した。

#### ii) 質問一覧機能

受け付け中の質問や既に締め切った質問を一覧することができる機能である。これによって、答えなくてはならない質問を見分けることができるようにした。

#### iii) 回答機能

回答者がスムーズに答えることができるように心がけた。回答時に、投票項目を作ったときにパスワードを登録した場合はそのパスワードのチェックをするようにした。

#### iv) 回答状況確認機能

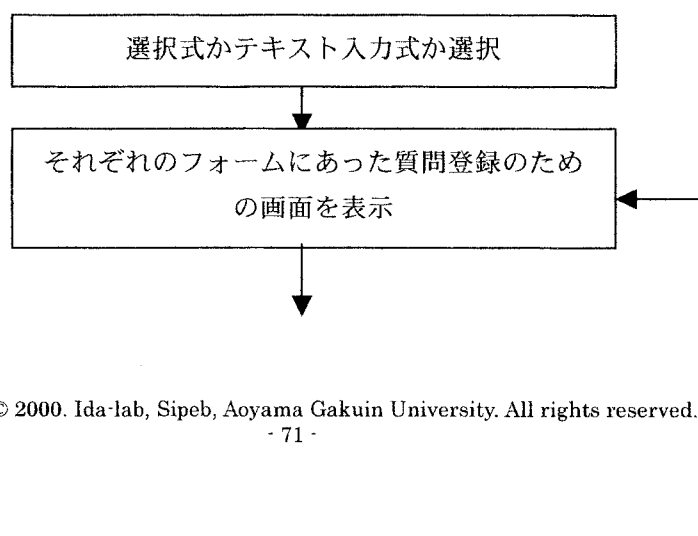
回答確認には投票項目作成時に登録したパスワードが必要になる。それを入力することで誰が何に投票したかがわかる。

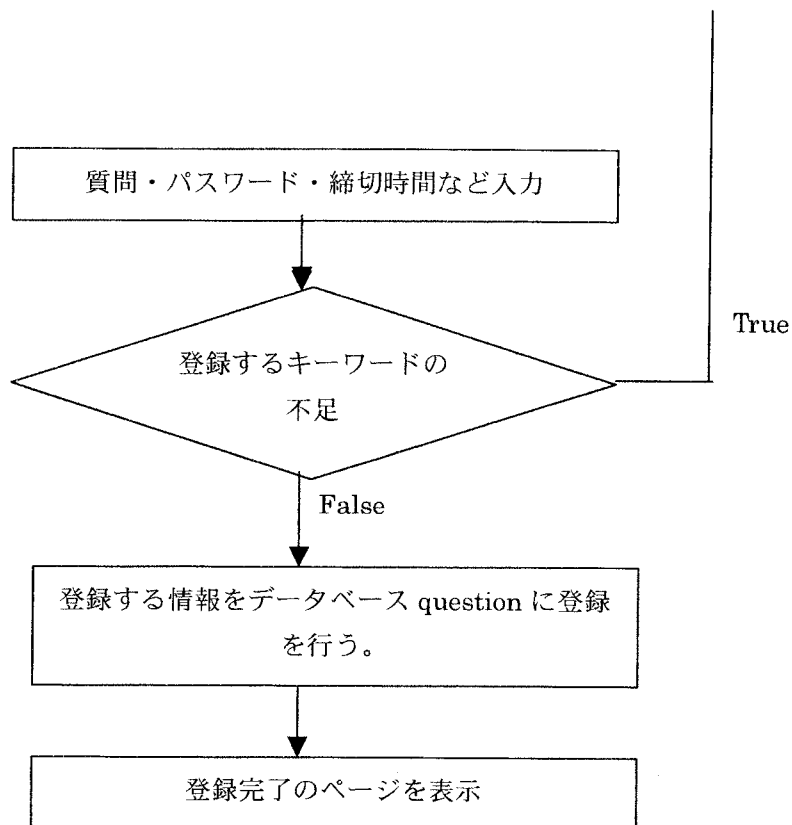
#### v) 終了処理

回答の受付を終了するには、回答状況確認機能の画面で終了処理を選ぶと終了する。

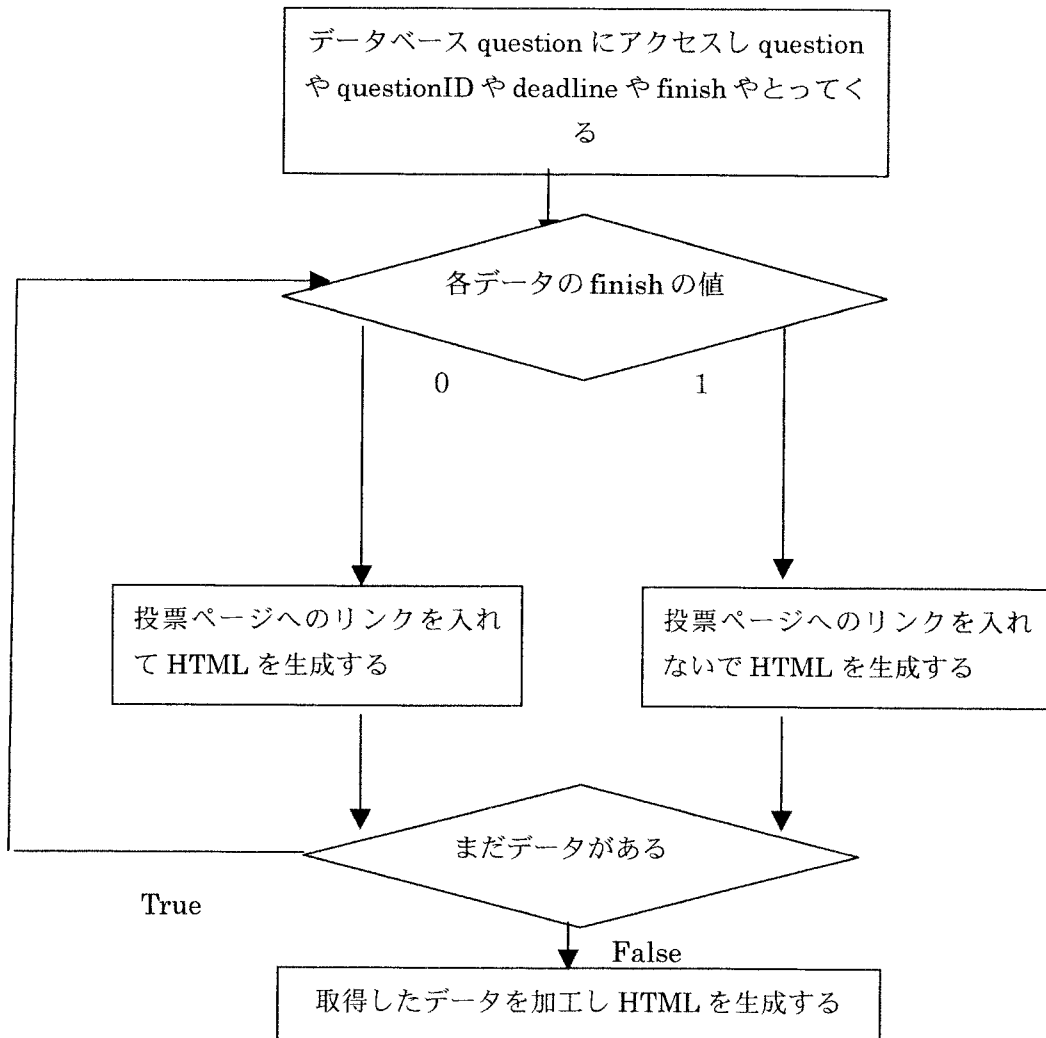
### ③アルゴリズム

#### i) 投票項目作成

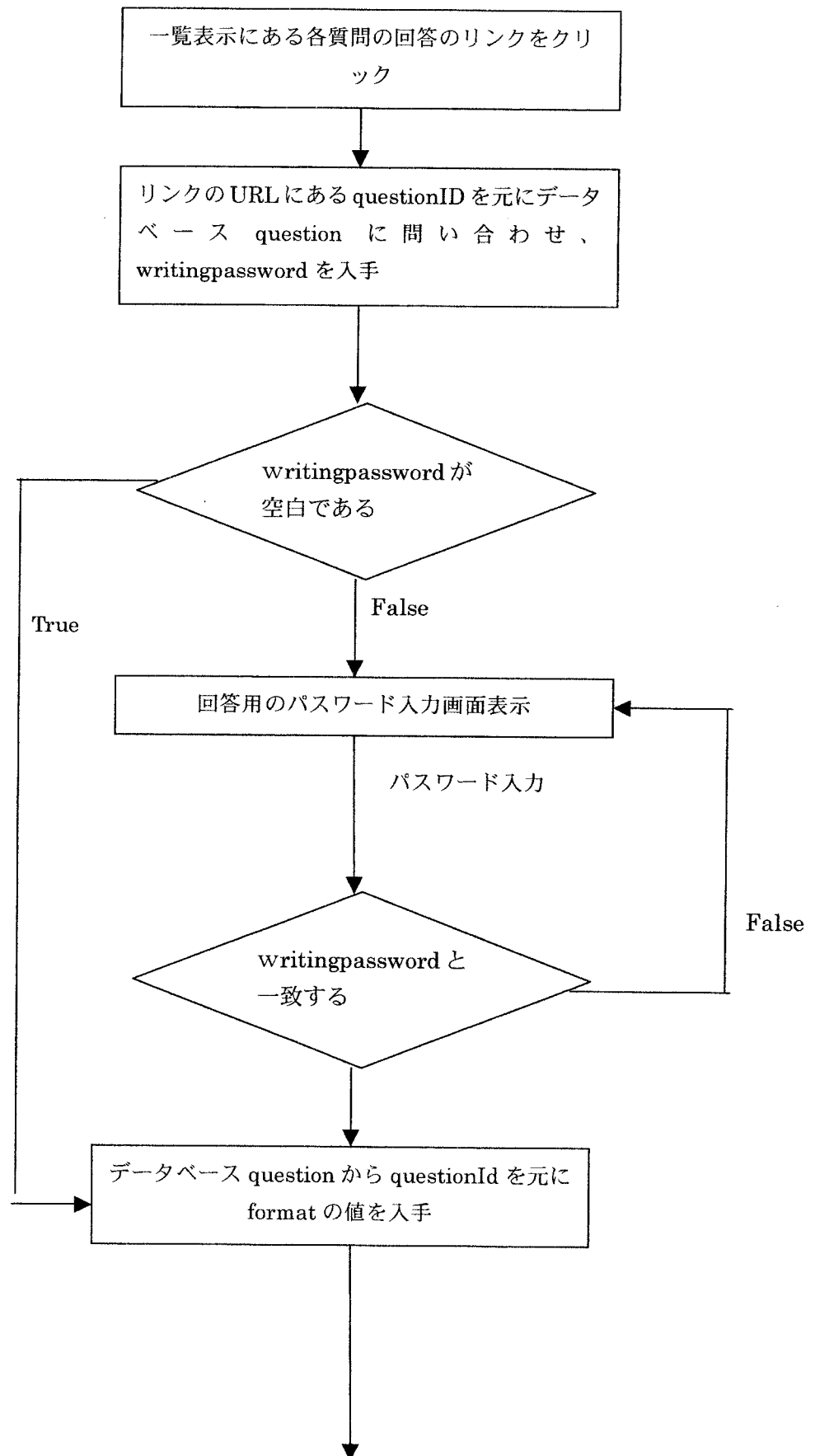


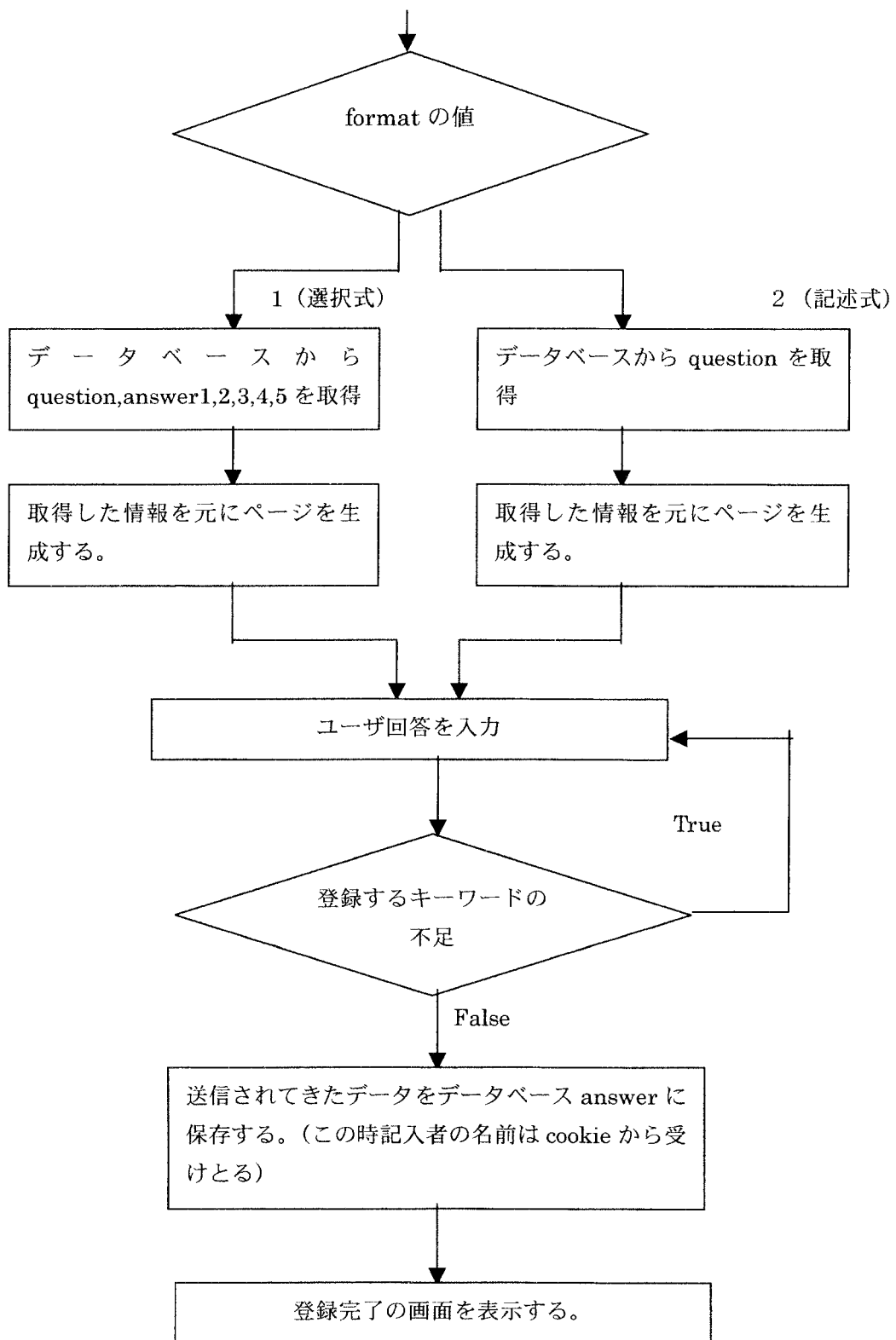


ii) 質問一覧表示

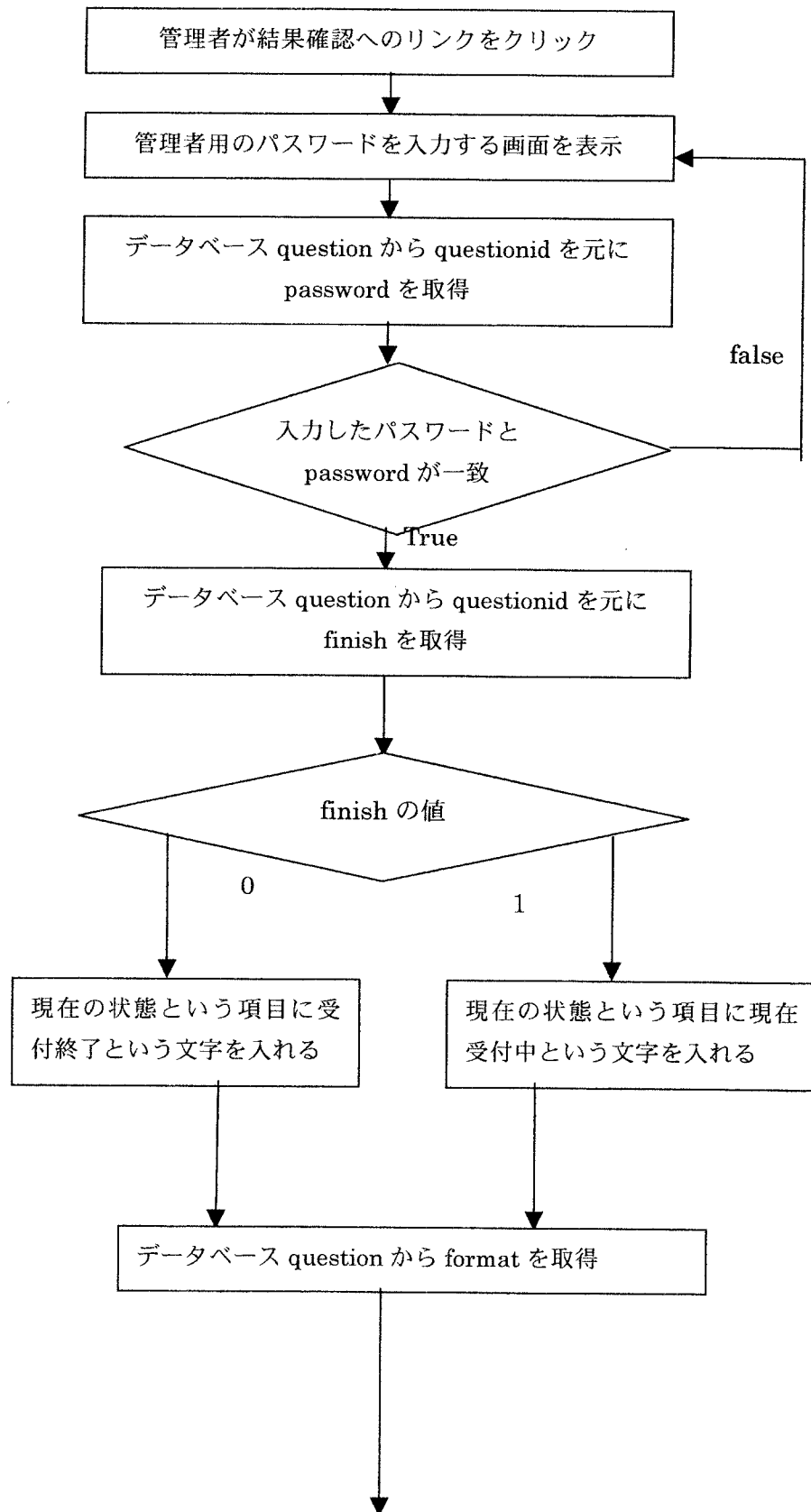


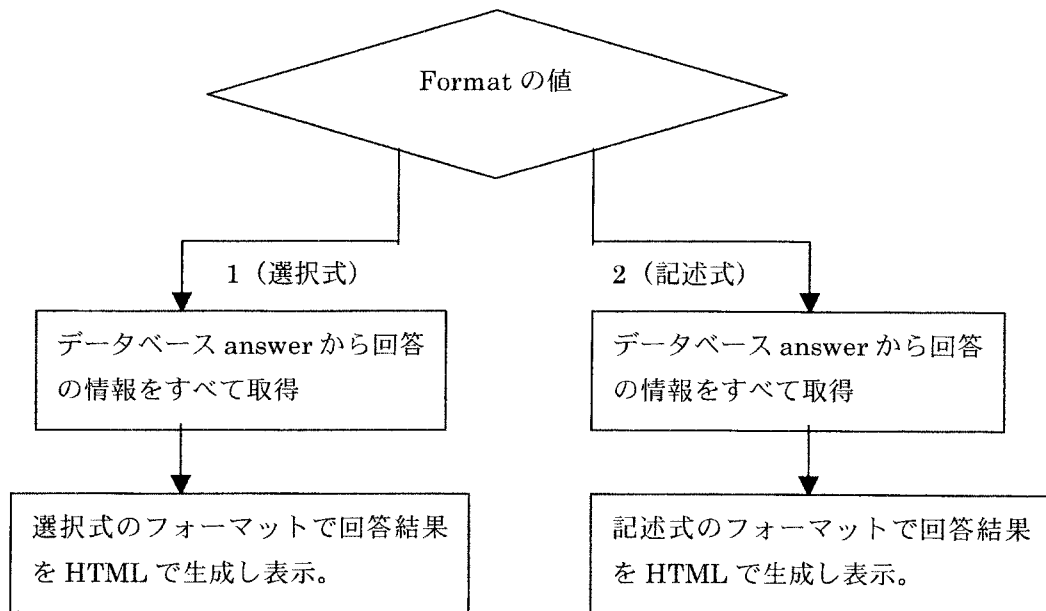
iii) 回答機能



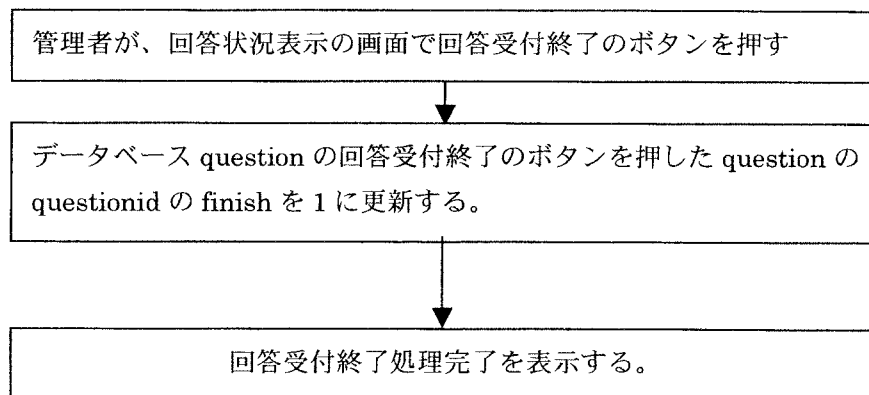


iv) 投票状況一覧





v) 回答受付終了



④作成に使った技術

JavaServlet 及びリレーショナルデータベース

### ⑤関係するデータベース詳細

(テーブル名：question)

カラム名	questionid	managercodename	question	nowdate	deadline
データ型	Int 型	Text 型	Text 型	Text 型	Text
詳細	ナンバリングする	作成者(管理者)のcodename	質問文	作成時間	回答締切時間

password	writingpassword	format	anwer1	answer2
Text 型	Text 型	Text 型	Text 型	Text 型
回答結果確認時のパスワード	回答時のパスワード(設定は任意)	回答形式 1→選択式 2→記述式	選択式の時 選択肢 1	選択式の時選 択肢 2

answer3	answer4	answer5	finish	groupcode
Text 型	Text 型	Text 型	Int 型	Text 型
選択式の時選 択肢 3	選択式の時選 択肢 4	選択式の時選 択肢 1	回答受付中? 0→受付中 1→受付終了	使用するコミ ュニティーの groupid

### ⑥改良すべきポイント

- ・青山祭の来場者から「使いづらい」という意見が出たので、全体のシステムを見直し、使いやすいものを作る。
- ・WEB 上で投票や回答を集めるという新しい方法なので、ヘルプ画面や GUI をうまく使って、より利用しやすいものを作る。

### ⑦製作体験記

困ったのはグループで物事を決める方法にもいろいろな形式があるのではないかとことだった。しかしよく考えてみると、選択式か記述式の 2 つしかないということに気づき、回答には選択肢か記述式かを選択できるようにした。また、質問を登録する際のアルゴリズムを考えるのにも苦労した。ユーザが満足するものを作りたい、そんな気持ちで設計をした。1 つの目標として、普通のアンケートのように、1 ページで複数の質問を載せたかった。しかし、「それは難しい、いやできないのでは」と思ってしまった(今、考えるとできるような気がするのだが)。結局、1 ページ 1 項目ということにすることにした。また、投票できる人を制限したいような質問の場合、その処理をどうしたらよいのかを考えた。質問者に投票を許可する人を指定してもらうという方法も考えたが、それは質問を作った人の手間になる。そこで投票ページに行くためのパスワードを質問作成時に登録してもらい(登録は任意で登録してない時は、投票時パスワードは要





## 投票状況一覧

井田研究室

たけなすのブログを閲覧しよう  
あなたに投票しよう  
あなたに投票しよう  
あなたに投票しよう

投票終了しました。  
あなたに投票しよう  
投票管理にログインしよう  
あなたに投票しよう

### ただいまの投票状況

No	投票管理者	質問	質問が設置された日	締め切り	
1	cafe-kuro	人との付き合いのためのMTG	00/10/29 16:39	2000/10/31 00:00	投票 投票権なし(管理者のみ)
3	cafe-kuro	明日晴れるか	30/10/30 12:47	2000/10/31 00:00	投票 投票権なし(管理者のみ)
4	macc176	明日のあはれはどのくらい?	30/10/30 12:57	年/月/日 時刻のフォーマットで入力してください	投票終了しました。 投票権なし(管理者のみ)

結果作成中...

図 4-14

## 回答画面

井田研究室

たけなすのブログを閲覧しよう  
あなたに投票しよう  
あなたに投票しよう  
あなたに投票しよう

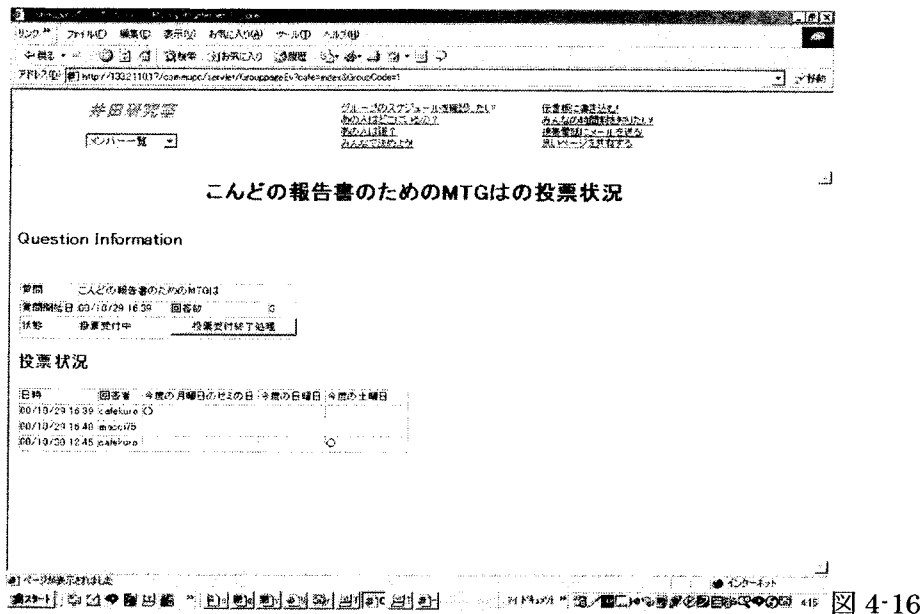
投票終了しました。  
あなたに投票しよう  
投票管理にログインしよう  
あなたに投票しよう

### 問題明日晴れるか

回答欄

図 4-15

## 集計画面



(参考: Java ソース)

```

/*
 * @(#) VoteControl.Java
 *
 * Voting System
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 */

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class VoteControl extends HttpServlet{
    VoteUi vu=new VoteUi();
    ConnectVoteDB CVD=new ConnectVoteDB();
    Center center=new Center();

    /**
     * getCookieValue
     *      必要な Cookie をもってくる。
     */

    private Cookie getCookieValue(String name,HttpServletRequest request){
        Cookie cookies[]=null,cookie=null;

        if((cookies=request.getCookies())!=null){
            for(int i=0;i<cookies.length;i++){
                if(cookies[i].getName().equals(name)){
                    synchronized(this){
                        cookie=(Cookie)cookies[i].clone();
                    }
                    return cookie;
                }
            }
        }
    }
}

```

```

        }
    }
    return null;
}

/**
 * controlForm TEXT 型か Radio 型か判別する
 */

public void controlForm(HttpServletRequest request,HttpServletResponse response,Hashtable parameter)
throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        int Formatcode=Integer.parseInt((String)parameter.get("Format"));
        String groupcode=(String)parameter.get("GroupCode");
        if(Formatcode==1)
            vu.formRadio(request,response,Formatcode,groupcode);

        if(Formatcode==2)
            vu.formText(request,response,Formatcode,groupcode);
    }
}

/**
 * controlMakeRadio
 * 無記入の検索およびデータのクラス化
 */

public void controlMakeRadio(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter)throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        String groupcode=(String)parameter.get("GroupCode");

        if(parameter.get("Question").equals("") | parameter.get("Deadline").equals("") | parameter.get("Answer1").equals("")
        ) | parameter.get("Answer2").equals("") | parameter.get("Password").equals(""))
            {vu.nodata(request,response,groupcode);
            }
        else{
            QuestionData qd=new QuestionData();
            setData(request,qd,parameter,groupcode);
            qd.setAnswer1((String)parameter.get("Answer1"));
            qd.setAnswer2((String)parameter.get("Answer2"));
            qd.setAnswer3((String)parameter.get("Answer3"));
            qd.setAnswer4((String)parameter.get("Answer4"));
            qd.setAnswer5((String)parameter.get("Answer5"));
            CVD.setQuestion(qd);
            vu.endMake(request,response,groupcode);
        }
    }
}

/**
 * controlMakeText
 * 無記入の検索およびデータのクラス化
 */

public void controlMakeText(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter) throws IOException,ServletException{

```

```

        boolean checkLogs=center.checkLogs(request,response);
        if(!checkLogs){
            response.sendRedirect("../login.html");
        }
        else{
            String groupcode=(String)parameter.get("GroupCode");

if(parameter.get("Question").equals("") | parameter.get("Deadline").equals("") | parameter.get("Password").equals("
"))
            {vu.nondata(request,response,groupcode);
            }
        else{
            QuestionData qd=new QuestionData();
            setData(request,qd,parameter,groupcode);
            qd.setAnswer1((String)parameter.get("Answer1"));
            qd.setAnswer2((String)parameter.get("Answer2"));
            qd.setAnswer3((String)parameter.get("Answer3"));
            qd.setAnswer4((String)parameter.get("Answer4"));
            qd.setAnswer5((String)parameter.get("Answer5"));
            CVD.setQuestion(qd);
            vu.endMake(request,response,groupcode);
        }
    }
}

/**
 *setData      共通するパラメータをクラスに送る
 */

private void setData(HttpServletRequest request,QuestionData qd,Hashtable parameter,String groupcode){
    Cookie codenameCookie=getCookieValue("Codename",request);
    String Managercodename=codenameCookie.getValue();
    qd.setManagercodename(Managercodename);
    qd.setQuestion((String)parameter.get("Question"));
    String date=getNowDate();
    qd.setNowDate(date);
    qd.setDeadline((String)parameter.get("Deadline"));
    qd.setPassword((String)parameter.get("Password"));
    qd.setWritingpassword((String)parameter.get("Writingpassword"));
    qd.setFormat(Integer.parseInt((String)parameter.get("Format")));
    qd.setGroupcode(groupcode);
}

/**
 * controlVote
 *投票リスト管理部
 */

public void controlVote(HttpServletRequest request,HttpServletResponse response,Hashtable parameter)
throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        String groupcode=(String)parameter.get("GroupCode");
        Vector alllist=CVD.getAllList(groupcode);
        Enumeration data=alllist.elements();
        vu.makeAllList(request,response,data,groupcode);
    }
}
}

```

```

/**
 * inputVote
 * 投票フォーム記入用紙管理部 1
 *
 */

public void inputVote(HttpServletRequest request, HttpServletResponse response, Hashtable parameter)
throws IOException, ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        int QuestionID=Integer.parseInt((String)parameter.get("QuestionID"));
        QuestionData mQuestionData=CVD.getAllData(QuestionID);
        String groupcode=(String)parameter.get("GroupCode");
        if(!mQuestionData.getWritingpassword().equals(""))
            vu.makeFormPassword(response,QuestionID,"votepass",groupcode);

        else{inputVote2(response,mQuestionData,QuestionID,groupcode);}
    }
}

/**
 * inputVote2
 * 投票フォーム管理部 2
 *
 */

public void inputVote2(HttpServletResponse response,QuestionData mQuestionData,int QuestionID,String
groupcode)throws IOException,ServletException{

    if(mQuestionData.getFormat()==1)
        vu.makeFormRadio(response,mQuestionData,QuestionID,groupcode);
    if(mQuestionData.getFormat()==2)
        vu.makeFormText(response,mQuestionData,QuestionID,groupcode);
}

/**
 *
 *checkWPassword
 *パスワードチェック部
 */

public void checkWPassword(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter)throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
        String groupcode=(String)parameter.get("GroupCode");
        int QuestionID=Integer.parseInt((String)parameter.get("QuestionID"));
        QuestionData mQuestionData=CVD.getAllData(QuestionID);
        if(parameter.get("password").equals(mQuestionData.getWritingpassword())){
            inputVote2(response,mQuestionData,QuestionID,groupcode);
        }
        else{
            vu.passwordError(response,groupcode);
        }
    }
}

/**
 * putVoteControl

```

```

*
*   投票処理部
*/
public void putVoteControl(HttpServletRequest request, HttpServletResponse response, Hashtable
parameter)throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
String groupcode=((String)parameter.get("GroupCode"));
int QuestionID=Integer.parseInt((String)parameter.get("QuestionID"));
Cookie codenameCookie=getCookieValue("Codename",request);
String Codename=codenameCookie.getValue();
AnswerData mAnswerData=new AnswerData();
mAnswerData.setQuestionID(QuestionID);
mAnswerData.setAnswer((String)parameter.get("Answer"));
mAnswerData.setCodename(Codename);
String NowDate=getNowDate();
mAnswerData.setNowDate(NowDate);
CVD.putVoteDB(mAnswerData);
vu.thankyouVote(response,groupcode);
    }
}
/**
*   controlCheck
*   パスワード表示画面
*/

public void controlCheck(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter)throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{ int QuestionID=Integer.parseInt((String)parameter.get("QuestionID"));
String groupcode=(String)parameter.get("GroupCode");
vu.makeFormPassword(response,QuestionID,"checkpass",groupcode);
    }
}

/**
*   checkPassCheck
*   パスワード照会
*/

public void checkPassCheck(HttpServletRequest request,HttpServletResponse response,Hashtable
parameter)throws IOException,ServletException{
    boolean checkLogs=center.checkLogs(request,response);
    if(!checkLogs){
        response.sendRedirect("../login.html");
    }
    else{
String groupcode=(String)parameter.get("GroupCode");
int QuestionID=Integer.parseInt((String)parameter.get("QuestionID"));
QuestionData mQuestionData=CVD.getAllData(QuestionID);
String password=(String)parameter.get("password");
if(!password.equals(mQuestionData.getPassword())){
    vu.passwordError(response,groupcode);
}
else{
    Vector allanswer=CVD.getAllAnswer(QuestionID);
    if(mQuestionData.getFormat()==1){
        vu.makeRadioAnswerList(response,allanswer,mQuestionData,QuestionID,groupcode);
    }
    if(mQuestionData.getFormat()==2){
        vu.makeTextAnswerList(response,allanswer,mQuestionData,QuestionID,groupcode);
    }
}
}
}

```





## 4-8 【ユーザ管理】

〈文責：黒田洋史

### ①作成理由

「こみゅぽ」の特徴は、ログインしている個人が特定され、そのログインしている人に合った情報を提供するということである。それを実現するためには、ユーザ管理は必要不可欠である。

### ②ツールの機能

#### i) ユーザ登録

初めて「こみゅぽ」を使ってもらう時に自分の個人情報を登録してもらう。それと同時に、次回以降ログインするためとコードネーム（=ニックネーム）とパスワードも登録する。

#### ii) 認証

ユーザ登録時に登録したコードネームとパスワードを入力した後、ログインしている人を特定し、それを同時に個人情報に対するセキュリティーを整える。

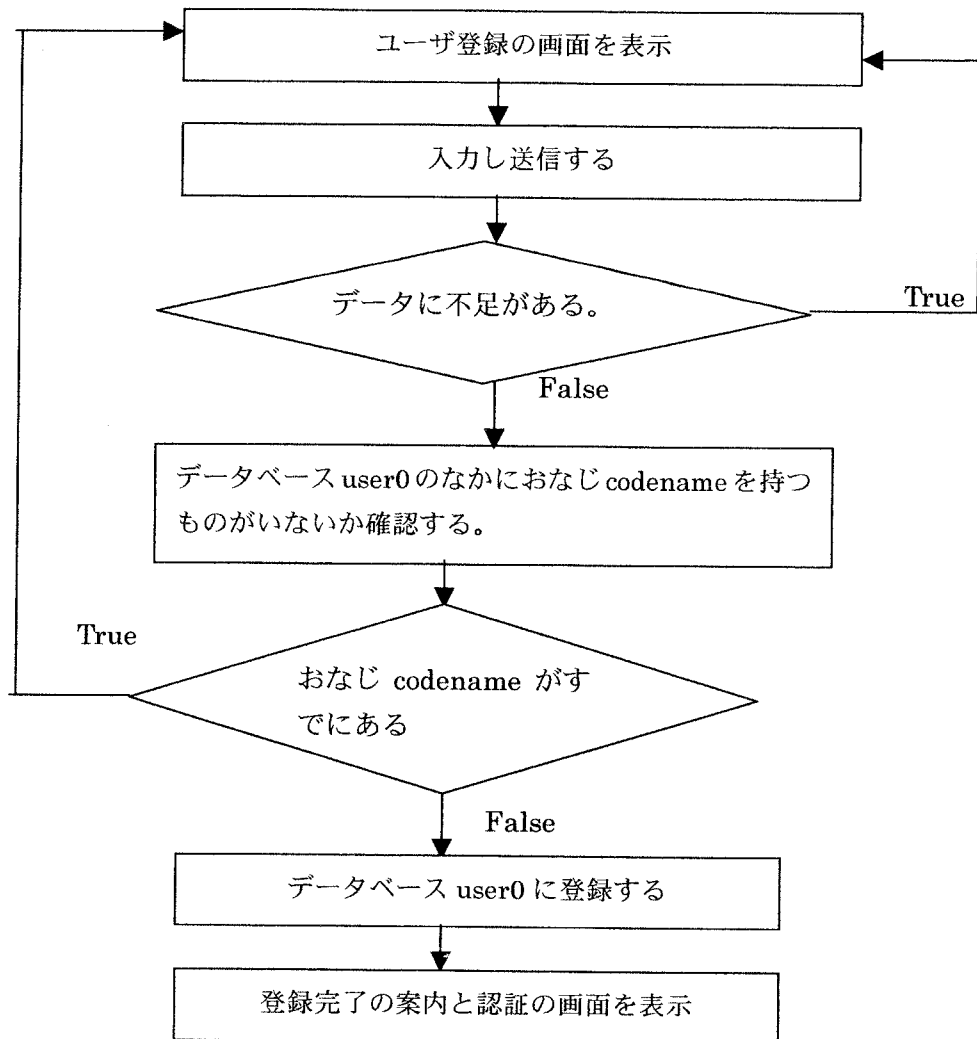
認証後、各ユーザ専用のページに移動する、つまり HTTP 要求を出す時確認される HTTP-Cookie（ログインするたびに変わる onetimepassword とユーザ固有の codename の 2 つ）をブラウザに持たせる。同時に、この 2 つの値をデータベースに保存しページ移動時のユーザ確認の時に使う。

#### iii) ユーザ確認

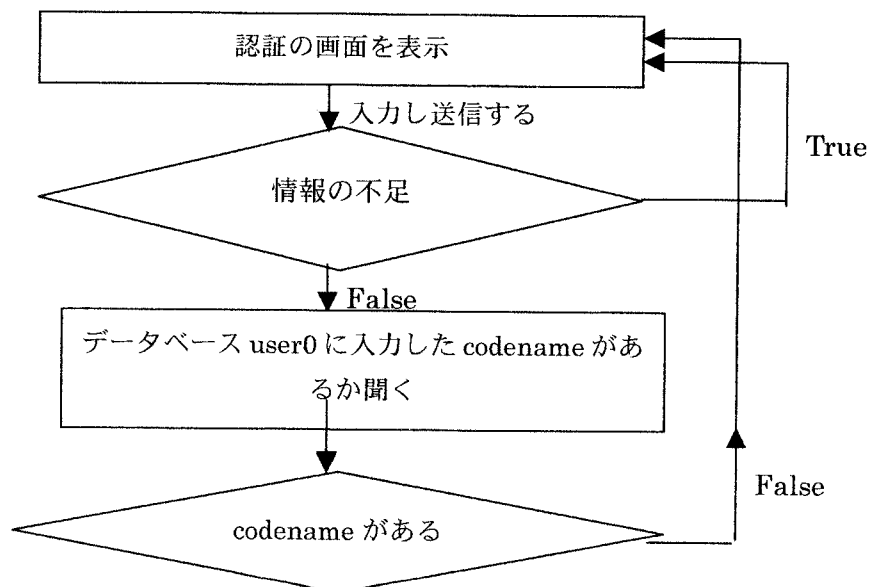
ページに移動するたびに、認証を受けた人が使っているかを確認する。ここで何らかの問題があれば、認証の画面に自動的に移動させ、それ以降のページを見せないようにする。

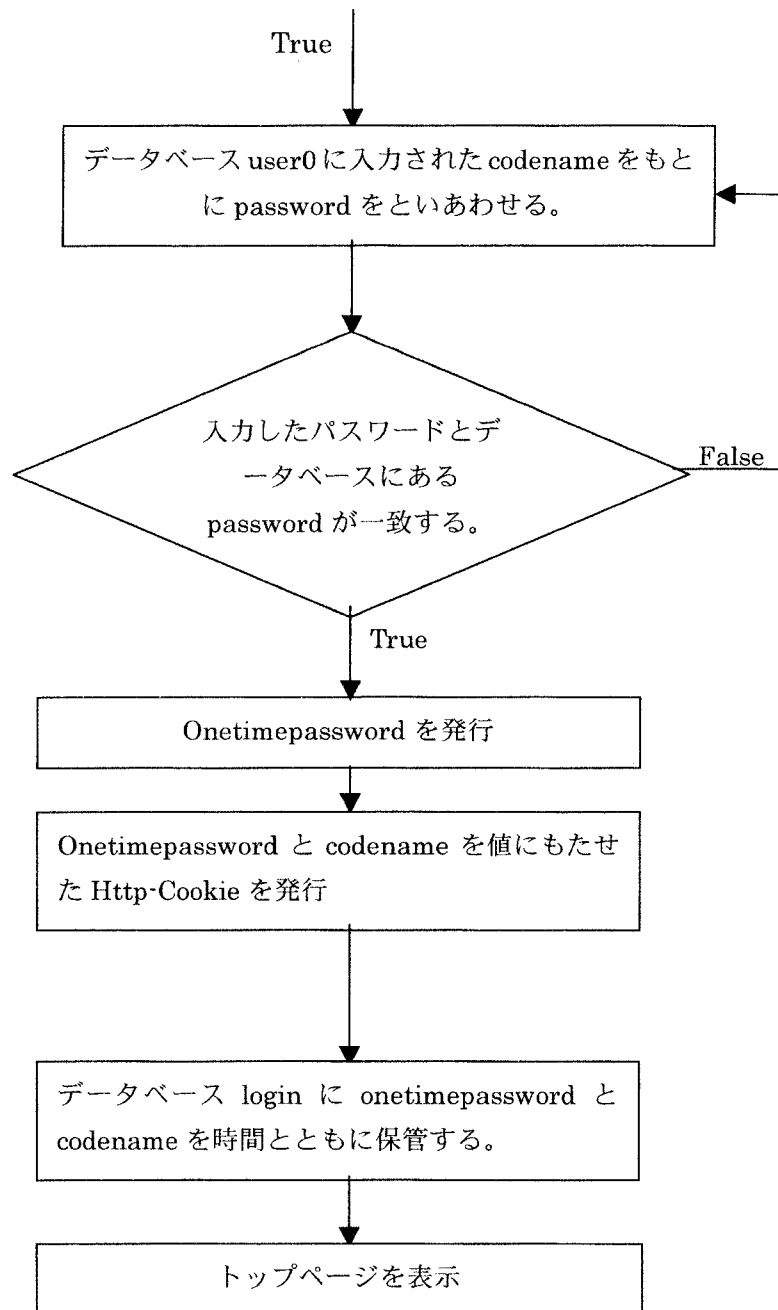
### ③アルゴリズム

#### i) ユーザ登録

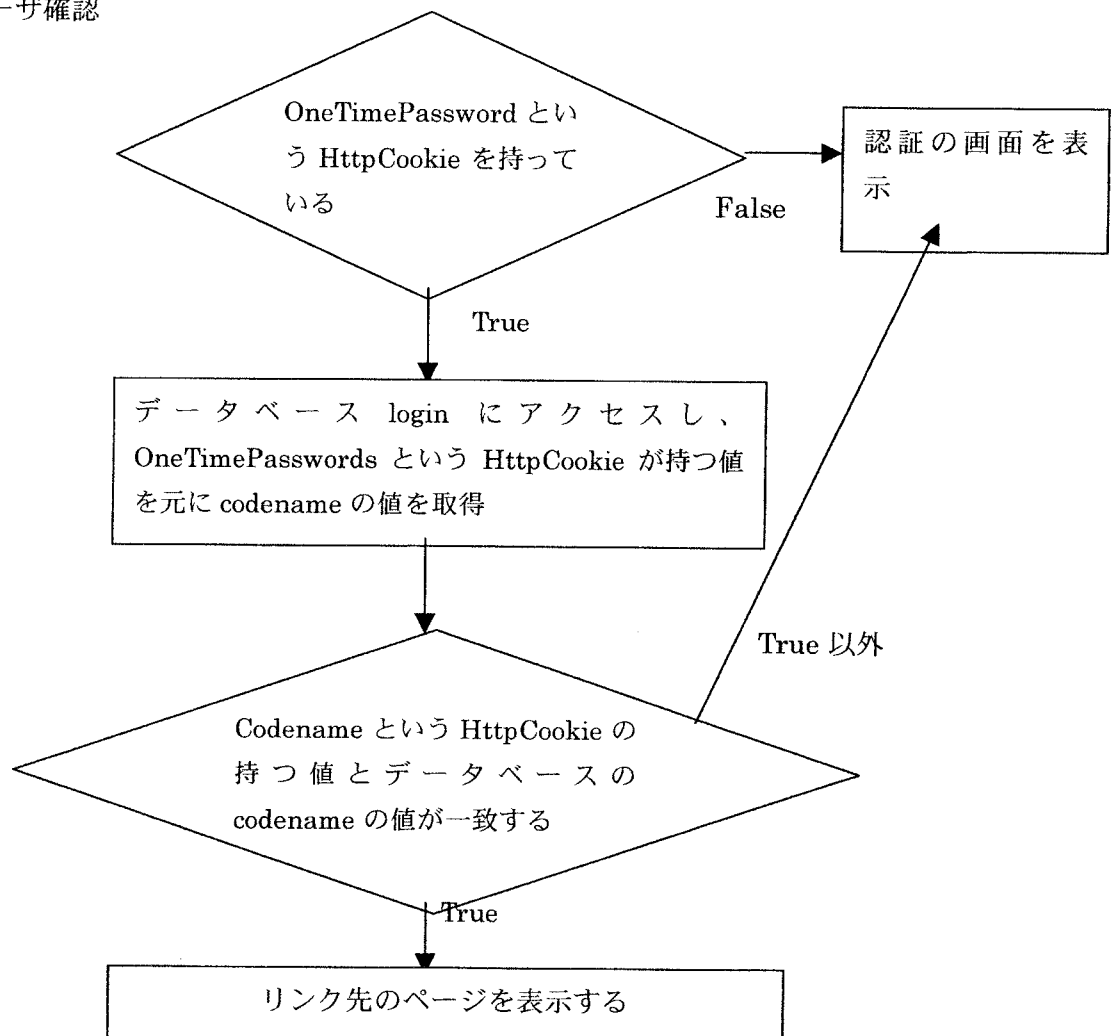


#### ii) 認証





iii) ユーザ確認



④作成に使った技術

JavaServlet 及びリレーショナルデータベース

## ⑤データベース詳細

(テーブル名：user0)

カラム名	userid	codename	name	studentid	password
データ型	Int 型	Text 型	Text 型	Text 型	Text 型
詳細	ナンバリングする	ニックネーム	本名	学生番号	ログイン時のパスワード

gakubu	gakka	grade	email	comment
Text 型	Text 型	Text 型	Text 型	Text 型
学部	学科	学年	メールアドレス	コメント

address	tel	groupcode
Text 型	Text 型	Text 型
住所	電話番号	所属しているグループの groupid を順に”, ”中にいれ並べていく

### login

カラム名	loginid	nowdate	codename	onetimepassword
データ型	Int 型	Text 型	Int 型	Text 型
詳細	ナンバリングする	ログインした時間を記録	ログインした codename	ログインするたびに変わるパスワードを記録

## ⑥改良すべき点

- ・登録時の項目を再検討する。
- ・登録時に、各情報（本名など）を公開しても良いか悪いかを尋ねるようにする。
- ・学校のデータベースにつなぎ登録時の入力情報を少なくする。
- ・ユーザ確認の際のアルゴリズムの不備が多くバグが起こるかもしれないので検討しなおす。

## ⑦製作体験記

こみゅポの製作時のコンセプトにある、個人を特定しその個人にとって必要な情報を提供するという概念を実現するための重要なツールである。まず、個人を特定するためにユーザ登録と認証を導入することを決めた。ユーザ登録と認証の作成は9月の早い段階でできていた。問題となるのは、ログインした人に合った情報を提供するために、HTTP 要求が出るたびに個人を確認し個人にあった情報を表示させることであった。そ

のために、いろいろと調べ Http-cookie を利用し、ログインしたユーザの codename を持たせることにした。しかし、問題が1つあった。それは、ログインしていない人が何らかの方法で入ってきた時に接続をはじく機能をいかにして持たせるかということである。これは、ログイン時にパスワードを発行しそのパスワードをデータベースに保存し、同時に Http-cookie に保存し、それを Http 要求のたびに確認し、確認が取れない時は自動的に認証画面に移動させることにした。そして、プログラミングした。

プログラムも実際大変だった。ユーザ確認のプログラムは、すべての Java ファイルに組み込まないといけない(全てが不備の無いものにしないとイケない)ので、これを作成する以前のプログラムを直していった。それぞれでコンパイルしテストしデバッグするのは大変だったが実際に出来上がった後は嬉しいし、インターネットの構造がわかったようで得した気分になった。

### (参考：表示画面)

#### ユーザ登録

登録

コードネーム

名前

学年番号  パスワード

学部  学科

メールアドレス

コメント

性別

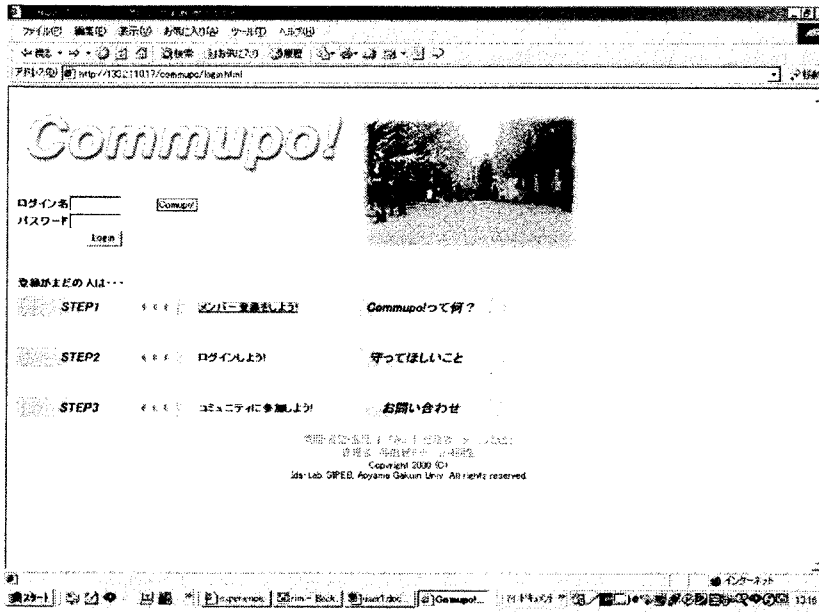
電話番号

その他

Copyright © 2000. Ida-lab, Sipeb, Aoyama Gakuin University. All rights reserved.

4-17

## 認証(こみゆポのトップページ)



4-18

(参考: Java ソース)

```

/*
 * @(#) UserControl.Java
 *
 * User
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 */
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class UserControl extends HttpServlet{
    UserConnectDB UCDB=new UserConnectDB();
    UserUi UU=new UserUi();
    public void regist(HttpServletResponse response,Hashtable parameter) throws
    IOException,ServletException{
        String Codename=(String)parameter.get("Codename");
        String Name=(String)parameter.get("Name");
        String StudentID=(String)parameter.get("StudentID");
        String Password=(String)parameter.get("Password");
        String Gakubu=(String)parameter.get("Gakubu");
        String Gakka=(String)parameter.get("Gakka");
        String Grade=(String)parameter.get("Grade");
        String Email=(String)parameter.get("Email");
        String Comment=(String)parameter.get("Comment");
        String Address=(String)parameter.get("Address");
        String Tel=(String)parameter.get("Tel");

        if(Codename.equals("")|Name.equals("")|StudentID.equals("")|Password.equals("")|Gakubu.equals("
")|Gakka.equals("")|Grade.equals("")|Email.equals(""))
        {
            UU.noData(response);
        }
        else{
            boolean codenameDouble=UCDB.checkCodenameDouble(Codename);

```





```

/**
    getNowDate
    現在時刻を年/月/日 時:分にする
*/

private String getNowDate()
{Calendar rightNow=Calendar.getInstance();
  int yearInt=rightNow.get(Calendar.YEAR)-2000;
  String year=checkInt(yearInt);
  int monthInt=rightNow.get(Calendar.MONTH)+1;
  String month=checkInt(monthInt);
  int dateInt=rightNow.get(Calendar.DATE);
  String date=checkInt(dateInt);
  int hourInt=rightNow.get(Calendar.HOUR_OF_DAY);
  String hour=checkInt(hourInt);
  int minuteInt=rightNow.get(Calendar.MINUTE);
  String minute=checkInt(minuteInt);
  String dateNow="" +year+"/"+month+"/"+date+" "+hour+":"+minute+"";
  System.out.println(dateNow);
  return dateNow;
}

/**
    checkInt  数値が1桁の時0を頭につける
*/

private String checkInt(int numbers){
  String stringNumber=null;
  if(numbers<10)
  {String stringNumbera=String.valueOf(numbers);
  stringNumber="0"+stringNumbera;}
  else{stringNumber=String.valueOf(numbers);}
  return stringNumber;
}

}

```

## 4-9 【コミュニティー管理（新規作成・検索・参加表明）】

〈〈文責：黒田洋史

### ①作成理由

「こみゅぽ」にはたくさんコミュニティーが存在するようになる。その際、コミュニティー自体を管理するということが必要になってくる。また、管理と同様に誰でも自由にコミュニティーを作成できる環境が必要になってくる。それらを簡単にできるようにするため作成した。

### ②機能

#### i) コミュニティー作成

誰でもいつでもすぐに自分のコミュニティーを作れるようにする機能である。コミュニティーの作成と同時にコミュニティー専用のページができ、利用できるようにする。

ii) コミュニティー検索

ユーザがコミュニティに参加したい時に自分が所属したいコミュニティを検索できるような機能である。検索は、コミュニティ名、コミュニティの種類（ゼミ、サークル、委員会、その他など）と登録時の紹介文の内容から検索できる。

iii) コミュニティー参加意思表示及び参加許可

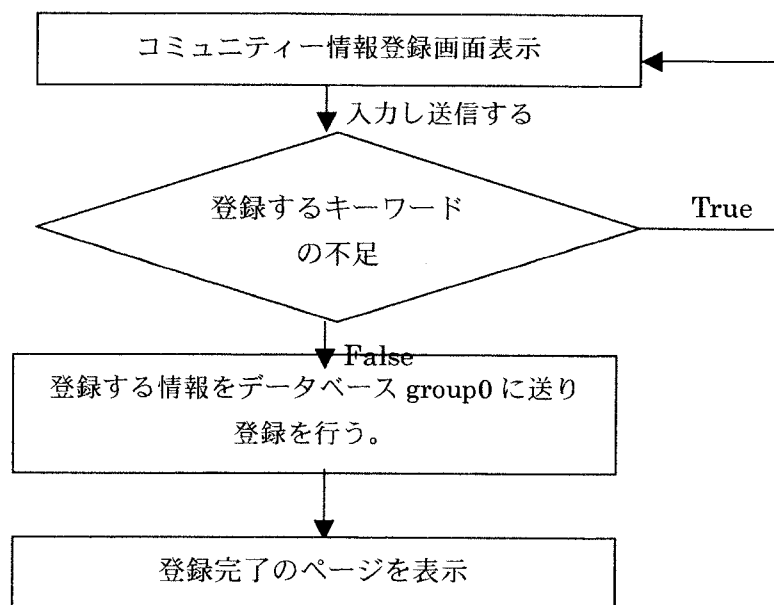
コミュニティに参加したい人はコミュニティの管理者（作成した人）に参加したいということを伝え、許可をとる必要がある。

iv) コミュニティーのページへの移動

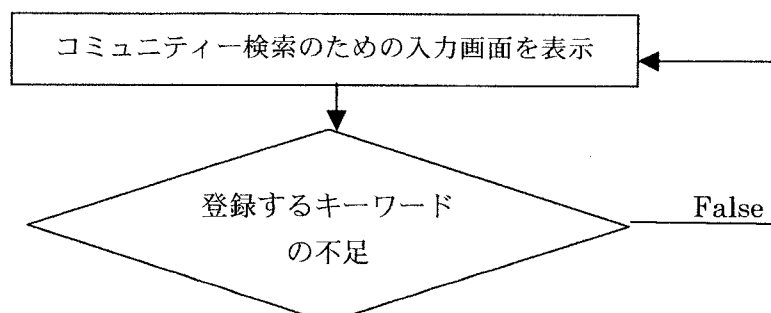
他のコミュニティへ移動するためには、ユーザは自分の所属しているコミュニティのリストから参加したいコミュニティを選択する。すると、そのコミュニティのページが新しいウィンドウで開く。

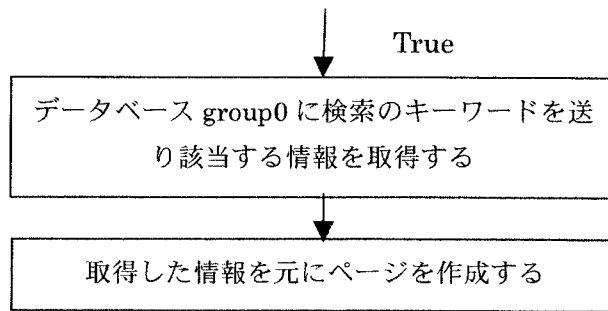
③アルゴリズム

i) コミュニティー作成

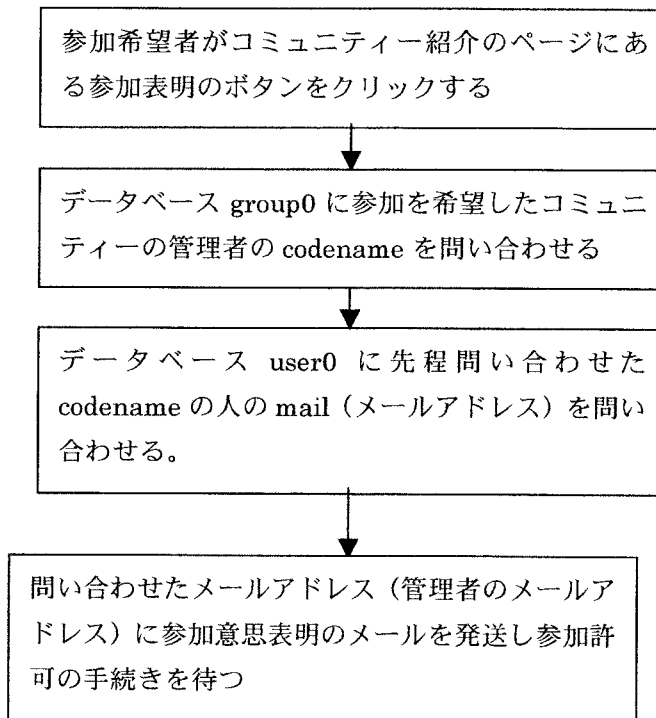


ii) コミュニティー検索

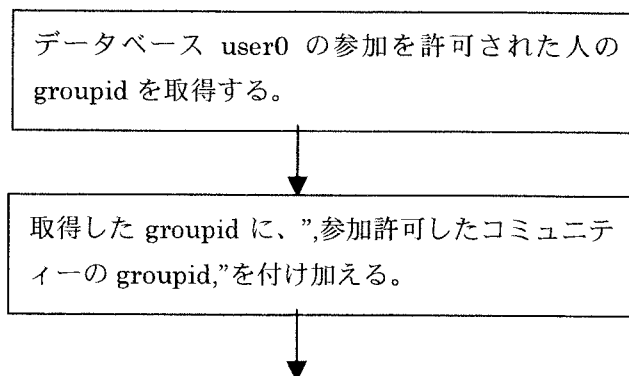




iii) コミュニティー参加意思表示及び参加許可  
 (注：青山祭で展示する都合上、参加許可なしでコミュニティに参加できるようにして展示は行った)



管理者が許可を出した場合



付け加えた groupid をデータベース user0 の参加を許可された人のセルに戻す

④コミュニティのページへ移動

リモコンのボタンを押す。

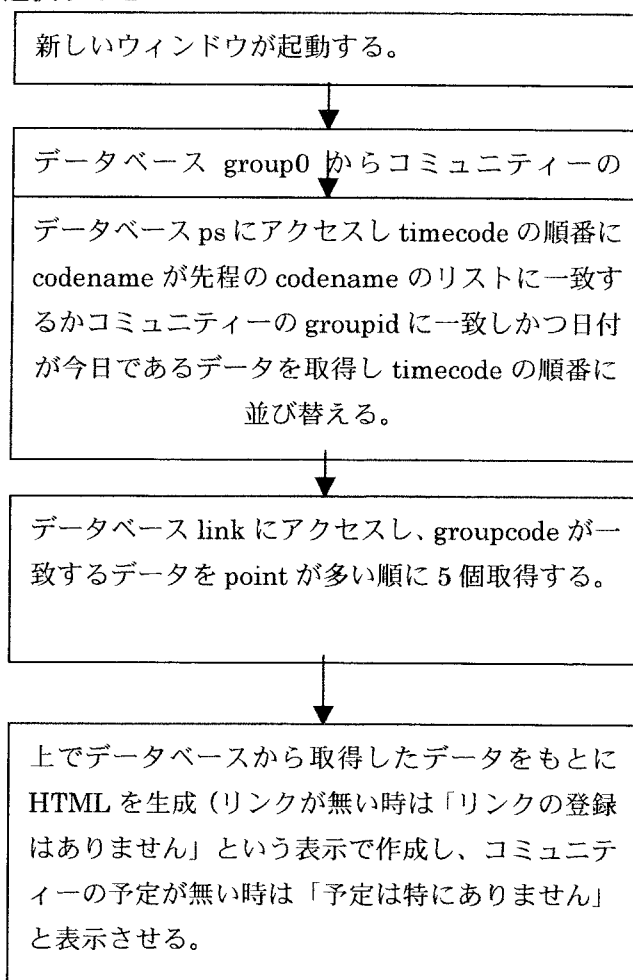
データベース user0 に使っているユーザの codename を元に groupcode の値を取得

groupcode の値を, の間の数字ごとに分ける。

groupcode を元にデータベース group からそれぞれの group の groupid と name を取得

HTML を生成 (コミュニティのリストは選択メニューにする)

コミュニティーを選択すると・・・



#### ④作成に使った技術

JavaServlet 及びリレーショナルデータベース

#### ⑤関係するデータベース詳細

(テーブル名 : group0)

カラム名	groupid	maneger	groupname	comment	category
データ型	Int 型	Text 型	Text 型	Text 型	Int 型
詳細	ナンバリングする	作成者 (管理者) の codename	コミュニティーの名前	説明・PR 文	1→ゼミ 2→サークル 3→その他

(テーブル名:ps)

カラム名	pscode	codename	year	month	day
データ型	Int 型	Text 型	Text 型	Text 型	Text 型
詳細	ナンバリング する	個人 → codename コミュニティー → groupid	年	月	日にち

hour	minute	title	content	timecode
Text 型	Text 型	Text 型	Text 型	Text 型
時間	分	題名	詳細	時間 (年 2 桁 月 2 桁、時間 4 桁)

nowdate	porg	groupid
Text 型	Text 型	Text 型
登録した時間	0→個人 1→コミュニ ティー	所属している コミュニティーの groupid

(テーブル名 : link)

カラム名	linkid	groupid	codename	linkname	url
データ型	Int 型	Int 型	Text 型	Text 型	Text 型
詳細	ナンバリング する	コミュニティー の ID を保 存する	登録した人の codename を 保存する	リンク先の名 前を保存する	リンク先の URL を保存

point	deleted	nowdate
Int 型	Int 型	Text 型
リンク一覧か らのアクセス 回数を保存	0→未削除 1→削除済み (表示しない)	登録時の時間 を記録する

### ⑥改良すべき点

- ・コミュニティー作成時の記入内容の項目を充実させ、参加したユーザがそのコミュニティーに関してよくわかるようにする。

- ・コミュニティの参加意思表示から参加許可までのプロセスをセキュリティーに問題が出ない範囲で簡略化して参加許可までの手続きを短くする。
- ・WEB上で外部からの参加者を募集したいコミュニティ（サークルなど）とWEB上で参加者を募集しないもしくはしたくないコミュニティがあるのを考慮してコミュニティ作成時に選択することができるようにし、希望しないコミュニティの情報は検索ができないようにする。

## ⑦製作体験記

個人ページからコミュニティページへのアプローチをなすこの機能がとても重要であり、かつ、製作も大変だった。まず、みんなでどういう流れで移動していくのかを考えて構想を練った。もともと企画書の作成の段階で、個人のスペースとコミュニティのスペースを設け、それを行き来することによって成立させるということは決まっていたので、実際にはどうやって実現するかを話し合った。その時にみんなの夢や目標、そして現実を持っている技術力などを比較しながら話しながら決めた。全員がJavaプログラミングができるわけではなかった。サーバの設定などを中心に勉強して作業する人、アンケートを取ったり、発表展示の舞台である青山祭の内容を決めたりする人など、1人1人が専門の分野を持って作業を進めているので、Javaの知識に関しても分かっている人とそうでない人の差が大きかった。そのため、話し合いをするにもとても苦労した。実際にやってみないとわからないことも多かった。技術面の問題点をわかるように説明もしなければならない。そこで意見を戦わせ、少しずつ決めていく。意見を戦わせるのは実のところ面倒だったこともある。しかし、今となってはいい思い出である。でも、もう少し戦わせてもよかったかなとも思う。Javaの知識量が多いほうが強いという空気を作ってしまったような気もする。もっと、Javaが不得意な人にも気を使うべきだったかなと、今になって思う。しかしあの時はあの時でお互い必死だったのかなとも思う。お互いの意見をぶつけあいながら話し合いを進めた。以前決めた個人のスペースとコミュニティのスペースをウィンドウで分けてその間を行き来できるようにする。そして、メインウィンドウとは別のウィンドウにリモコンを作り、そこに所属するグループのリストを表示させその中から行きたいグループを選択し、選択すると新しいウィンドウが立ち上がりそのグループのページが開くというやり方をとることに決まった。やっと全体が見えてきたかなという空気がみんなの中で流れた。

しかし、僕はそれを実現させないといけない。それにはどうしたらいいのか考えた。グループの作成、検索は今まで自分の作ってきたプログラムのテリトリーで何とかできた。そのあとグループのページに移動させる仕組みを考えるのが一番のネックだった。

どうしたらできるか考えた。困ったので、アルゴリズムをもっと細かくして考えようと思ってみた。そうすると自然と答えが見えてきた。この場合まず、リストを選ぶ。そして新しいウィンドウを開きそこにグループのページを表示させる。このことがわかり、自分でも作れるような気がしてきた。次にそれぞれの作業のアルゴリズム等を考える。リストを選んで次のページへ行くというのはJavaScriptにあったのを覚えていたのでそれを

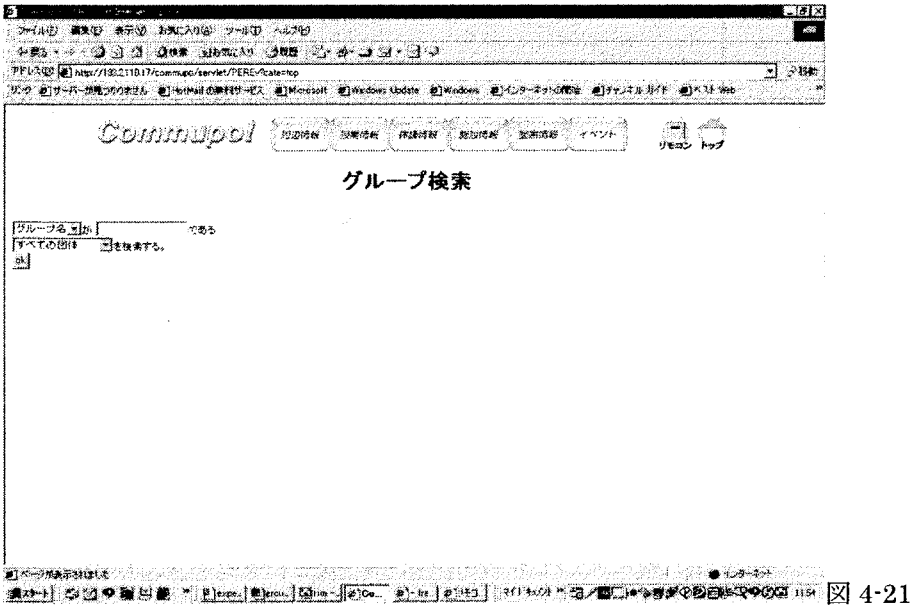




## グループのトップページ



## グループ検索



(参考 : Java ソース)

```

/*
 * @(#) GroupControl.Java
 *
 * Groupage
 * Copyright (C) 2000 Hiroshi KURODA
 *
 * This program is licensed under the GPL, whose whole statements
 * are included in the file 'COPYRIGHT' which must be distributed
 * with this source program.
 *
 */
import java.io.*;
import java.util.*;
import javax.servlet.*;

```

```

import javax.servlet.http.*;

public class GrouppageControl {
    GrouppageUi GPU=new GrouppageUi();
    GrouppageConnectDB GCDB=new GrouppageConnectDB();
    Center center=new Center();

    public void enter(HttpServletRequest request,HttpServletResponse response,Hashtable parameters)throws
    IOException,ServletException{

        String groupcode=(String)parameters.get("groupcode");
        String onepass=(String)parameters.get("onepass");
        String codename=(String)parameters.get("codename");
        response.sendRedirect("http://133.2.110.17/commupo/servlet/GrouppageEv?cate=index&GroupCode="+
+groupcode+");
    }

    public void makeindex(HttpServletRequest request,HttpServletResponse response,Hashtable
parameters)throws IOException,ServletException{
        boolean checkok=center.checkLogs(request,response);
        if(!checkok){
            response.sendRedirect("../login.html");
        }
        else{
            String groupcode=(String)parameters.get("GroupCode");
            Cookie Codenamecookie=getCookieValue(request,"Codename");
            String Codename=Codenamecookie.getValue();
            String Groupname=GCDB.getGroupname(Integer.parseInt(groupcode));
            GPU.makeindex(Groupname,Codename,groupcode,response);
        }
    }

    public void groupname(HttpServletRequest request,HttpServletResponse response,Hashtable
parameters)throws IOException,ServletException{
        boolean checkok=center.checkLogs(request,response);
        if(!checkok){
            response.sendRedirect("../login.html");
        }
        else{
            String groupcode=(String)parameters.get("GroupCode");
            Cookie Codenamecookie=getCookieValue(request,"Codename");
            String Codename=Codenamecookie.getValue();
            String Groupname=GCDB.getGroupname(Integer.parseInt(groupcode));
            GPU.groupname(Groupname,response);
        }
    }

    public void menu(HttpServletRequest request,HttpServletResponse response,Hashtable parameters)throws
    IOException,ServletException{
        boolean checkok=center.checkLogs(request,response);
        if(!checkok){
            response.sendRedirect("../login.html");
        }
        else{
            String groupcode=(String)parameters.get("GroupCode");
            Cookie Codenamecookie=getCookieValue(request,"Codename");
            String Codename=Codenamecookie.getValue();
            Vector memberlist=GCDB.getMembername(groupcode);
            GPU.menu(memberlist,response);
        }
    }

    public void subframe(HttpServletRequest request,HttpServletResponse response,Hashtable
parameters)throws IOException,ServletException{
        boolean checkok=center.checkLogs(request,response);
        if(!checkok){
            response.sendRedirect("../login.html");
        }
        else{
            String groupcode=(String)parameters.get("GroupCode");

```

```

        Cookie Codenamecookie=getCookieValue(request,"Codename");
        String Codename=Codenamecookie.getValue();
        GPU.subframe(groupcode,response);
    }
}

public void workspace(HttpServletRequest request,HttpServletResponse response,Hashtable
parameters)throws IOException,ServletException{
    boolean checkok=center.checkLogs(request,response);
    if(!checkok){
        response.sendRedirect("../login.html");
    }
    else{
        String groupcode=(String)parameters.get("GroupCode");
        Vector linklist=GCDB.getLink(groupcode);
        Vector memberlist=GCDB.getMemberlist(groupcode);
        long nowdatecode=getNowDateCode();
        Vector planlist=GCDB.getPlan(groupcode,memberlist,nowdatecode);
        int votenumber=GCDB.getvoteNumber(groupcode);
        GPU.workspace(response,linklist,planlist,votenumber);
    }
}

public long getNowDateCode()throws NumberFormatException{
    Calendar now=Calendar.getInstance();
    String myear=String.valueOf(now.get(Calendar.YEAR));
    int month=now.get(Calendar.MONTH)+1;
    String mmonth=String.valueOf(month);
    if(month<10){
        mmonth="0"+mmonth;
    }
    int date=now.get(Calendar.DATE);
    String mdate=String.valueOf(date);
    if(date<10){
        mdate="0"+mdate;
    }
    String nowdate=myear+mmonth+mdate+"0000";
    System.out.println(nowdate);
    long nowdatecode=Long.parseLong(myear+mmonth+mdate+"0000");
    return nowdatecode;
}
/**
 *
 *   getCookiValue
 *       クッキーゲット
 *
 */
private Cookie getCookieValue(HttpServletRequest request,String cookieName){
    Cookie cookies[]=null,cookie=null;

    if((cookies=request.getCookies())!=null){
        for(int i=0;i<cookies.length;i++){
            if(cookies[i].getName().equals(cookieName)){
                synchronized(this){
                    cookie=(Cookie)cookies[i].clone();
                }
                return cookie;
            }
        }
    }
    return null;
}
}
}

```

## 4-10 【K3 君 1 号 (GPA 計算機)】

〈〈文責：岡田牧男

### 1. きっかけ

大学院に進学するにはたいてい GPA という平均評定のようなものが判断材料のひとつになる。大学院に興味を持つ私はまず最初に GPA が何たるかを調べた。なにやらややこしい。何でも、

「大学の成績の中で AA を 4, A を 3, B を 2, C を 1 とし、これにそれぞれに単位数を乗じ、それらの数の総計を取得総単位数で除する。」ということらしい、と書いてある。さっぱりわからない。理解に苦しむということは当然算出も難しい、ということである。

加えて、GPA の算出がカリキュラムの決定にも絡んでくる。特に固執する必要もないのではあるが、自分が今どれくらいの成績を取っているのかという情報はかなり次の年のカリキュラムを作る上でも重要な要素となる。進学を考えていて、それなりの成績が必要な場合、授業のとり方も少しずつ変わってくるのである。

そこで私はこれらを簡単に計算できる計算機が必要であると考えた。私の所属する国際政治経済学部の学生は知っている限り向学心があり、GPA という値に興味を持っている人も少なくはない。そこで、このような計算機をひとつ作って、みんなで使えるようにすればいいじゃないか、と思ったのが GPA 計算機、K3 君 1 号の発端である。

### 2. 設計概念

中身は実際たいした事はない。結局「計算機」であることにはかわりがないからである。だから最初はアプレット版電卓をイメージして、それから GPA の算出に特化できるように修正を加えていった。ネットワーク上で行う作業であるので、できるだけ機能は限定して、操作が難しくなるようなことはできる限り避けた。

### 3. 設計

計算を実現するためのプロセスは以下のとおりである。

1. 数値、またはデータを入力する
2. 計算する
3. 計算結果を表示する

もう少し洗練すれば、細くなることはなるが、大まかな部分では、以上のようなになる。

GPA を計算するときに必要な情報は、以下のとおりである。

- ・ (個々の) 取得単位数
- ・ それぞれの成績

このデータを用いて、値を導き出すのであるが、それを表す計算式が、以下のとおりである。  
(イメージ添付?)

加えて、すべての科目において同じ成績であるはずがないので、個別の計算結果を「どんど

ん」追加していく仕組みが必要である。

1科目のみの計算のときは、成績評価がそのまま GPA になるので必要がないが、全部の取得単位の GPA を算出するときにはそうはいかない。

GPA の計算にはそれぞれの科目とそれぞれの成績を積算し、その総和を計算しなくてはならない。したがって、前回計算された GPA の値だけでは次の計算ができないのである。

そこで、プールされた単位数という変数値をひとつ用意して、毎回以前に計算された GPA の結果とプールされた単位数をかけて、それぞれの科目とそれぞれの成績の積の総和を算出する仕組みを考えた。数式にすると以下ようになる。

$$((gp * pu) + (gr * un))$$

gp=前回算出した GPA の値

pu=前回までの入力した単位の総和

gr=今回入力した科目の成績(換算済み)

un=今回入力した科目の単位数

この式から導かれた結果を前回までに入力した単位の総和と今回入力した科目の単位数の和で割ることによって、GPA が算出できるようにした。

要するにこういうことである。

$$gp = (((gp * pu) + (gr * un)) / tu);$$

gp=前回算出した GPA の値

pu=前回までの入力した単位の総和

gr=今回入力した科目の成績(換算済み)

un=今回入力した科目の単位数

tu=単位の総和

#### 4. 実装

実装にはたいした不思議なことに時間はかからなかった。実装作業のほとんどが計算ルーチンをどう作るかに費やした。ほとんど自分の苦手な数学の世界であった。GUI 関係の部品はいかなり早く作り上げることができた。しかし本当に計算がなかなかうまくいかなかった。最初に出来上がったツールを試したとき、その計算結果には「infinity」という言葉が並んでいた。「僕の成績は無限大!？」とかかなりびっくりしていたが、それは浮動小数点の扱いについて細かいところがあって、そこに少してこずった。int と float と double と、いろいろ変数の形が違うのがいまいちつかみにくかった。

#### 5. 反応・今後の課題

GPA の存在自体を忘れてる人が多い中、いいねえと返してくれる人が多かった。今のところ間違った値を出すということを報告されていないので、おそらく計算に関してはこれでいいということにしたい。今後の課題として、入力したもののリストをどこかに表示できるようにすることができれば、よりいいモノができるのではないだろうか。

## 4-11 【k3 君 2 号 (GPA 計算機)】

〈〈文責：岡田牧男

### 1. きっかけ

私に限らず、多くの学生にとって毎年4月にやってくる大変なことがひとつある。それはカリキュラム作成である。学生用の食堂では4月になると授業要覧と白紙の時間割表をにらんでいる光景を目にするだろう。カリキュラムを組むということは短期間に相当の苦勞を要するのである。

まず自分の好きな科目だけをとっていくことがまず大体不可能である。それは必修科目とかぶっていたり、卒業要件単位を満たすために必要な科目とかぶっていたり、個人的なスケジュールとかち合わなかったりと、事情は複雑に絡み合う。そこで多くの学生はある種の妥協を強いられる。しかし苦勞はそれだけでは終わらない。申請できる単位には上限が設けられているのだ。制限単位と、卒業要件単位である。

一般の学生は年間大体40単位程度しか申請することができない。これ以上申請することは不可能である。さらにどの学部においても、卒業要件単位が設定されている。卒業するには大体124から130強の単位が必要になる。しかも単純に130単位取得すればいいという話ではない。何種類かカテゴリー別に分けられていて、その中でも32単位取得すること、などという制限が設けられているのだ。単純な足し算の世界ではない。その上で、どの授業が自分の興味と合うとか、相性の問題などが組み合わされる。とてもじゃないが、1週間で理想のカリキュラムは作成するという話は、かなり無理がある。

そこで私はカリキュラムを作成するストレスを少しでも軽減できないのかと考えた。自分の取得単位などのデータと連動して、少しでも組み立てやすくなるようにならないか考えた。そこで考え出されたのが、k3 君 2 号である。

このk3 君 2 号はカリキュラム作成支援ツールとして、普通に時間割に書き込む感覚で自分の制限単位や卒業要件単位との関係をはっきりさせたりすることができる。

### 2. 設計概念

基本的な構造として、科目を入力(申請)すると、そこから制限単位と卒業要件単位の残りを計算するという形である。計算のための個人用のデータと連動できるようにも考えた。ユーザーは自分のIDを入力するだけで、自分の必要な情報が自動的に入力されるようできたらいいと考えた。

### 3. 設計

設計にはとても手を焼いた。今この原稿を書いている時点でも、依然としてバグは多数存在している。データベースと連動する前に、コアの部分を作成しようと考えた。まずはじめにGUI部品を作った。一日6コマ×6日×前期後期分(×2)で72のパネルを作成し、それぞれに科目名を書き込む場所と、申請単位数を選択するチョイスコントロールを作った。選

扱える科目名をあらかじめ書き込もうとするとそれは学部の違いなどもあり、途方もない作業になるので、今回は敬遠した。しかしこれはデータベースを用いると造作もないことなのかもしれない。それから結果を表示するためのラベルやテキストボックス、それに暗証番号入力用の部品も用意した。

それから計算のシステムを設計した。計算自体はそれほど難しいものではなかった。ただ苦勞したのは「やり直し」の部分だった。カリキュラムの作成は修正の繰り返しなので、それに対応していないとなると、使いにくくなると思ったので、それができるようにした。修正がかかると当然計算結果も変化しなくてはならず、その対応させるのに苦勞した。

あとから追加したものであるが、下の部分に暗証番号を入力できるようなパネルを作り、ある特定のパスワードを入力すると該当する人の制限単位や取得済み単位などの情報が、いきなり入力されるという機能を追加した。暗証番号が丸見えになるのはちょっとよくないので、アスタリスクに置き換えて、見えないようにした。それからユーザーのデータを用意して、きちんとパスワードを入力した時点で、卒業用件単位などを表示できるようにした。そのようなデータは、簡略化、柔軟性を持たせるのために本来データベースに入れるべきものであるが、今回はそこまで時間が足りなかったので、プログラムのソースコードに組み込んだ。

#### 4.実装

実装には本当に時間がかかった。プログラム本体の量が半端じゃなく大きくなったからだ。大まかな実装ステップとして

GUI 部品の配置

科目申請時のアクション(計算)

簡易版認証のアクション

の順番で実装した。

実装していく段階で、最初は単純に考えていたものが、どんどん複雑な事情を含んでいることを感じた。もともとのサイズの大きさもあったが、それに加えて、修正分が大きかった。

一番大変だったのは、修正するときの対処法を考えることと、(履修申請の)システム上不可能な組み合わせに対してどう対応するか、ということと、暗証番号の入力が大変だった。直せば直すほど、どんどんファイルサイズは雪だるま式に大きくなるし、どんどんバグが発見されていくので、大変だった。暗証番号はどうやってエコーをかけるのか、最初まったくわからず、char 型の変数の値を持って来ればよいということまではわかったが、問題の文字コードを探すのに一苦勞した。

#### 5.反応・今後の課題

結果として操作系統が複雑になってしまったので、反応はいまいちだった。そして、重かった。携帯電話に搭載される予定の Java のファイルサイズがひとつ 10K だと言うことを考えると、300K は重すぎた。そしてダウンロードに時間がかかるから、ストレスはたまる一方だった。まずデータベースを考えないツールを作るのはちょっと無謀だったのかもしれない。

れない。今後の課題としては、これ以上デバッグを続けるよりも、データベースに接続する形を作る必要があるように思う。そうでもしないと、根本的な問題は解決しないと思った。

## 4-12【あの人 Finder】

＜文責：岡田牧男

### 1. きっかけ

きっかけは夏の終わりごろだった。夏休みが中間地点を過ぎたころに、私はあの人 Finder (図 4-22) を作ろうと思い立った。それはある一瞬のひらめきであった。

私はゼミのミーティングをしようとメンバーのスケジュールを調整しようとした。しかし、それはとても困難を要した。夏休みに入ってしまうとたとえ同じゼミであろうが、会う機会がぐっと少なくなるうえ、何週間もはずすようなスケジュールを入れてしまうので、なかなか全員が集まるような時間がないのである。

特定の人間を捜すのも一苦勞である。「あの人 Finder の担当の岡田と連絡を取りたいんだけど、携帯に全然でないし、メール送っても何にも返ってこない。あいつはどこにいるんだ？」というような事態はそう少なくは無いはずである。ひょっとしたら栄養失調で入院しているかもしれないし、意図的にコンタクトを絶っているのかもしれない。あるいは長い旅に出ているのかもしれない。いずれにせよ、憶測の領域を越えることはなく、現実の問題は依然として残る。そして途方にくれる経験を私は何度も遭遇している。

問題は他にもある。「誰がいつ何をしているのか」というようなメンバー一人一人を網羅したスケジュール表を誰も持っていないうえに、どこにもないのだ。ミーティングひとつやるにも、いちいち全員に連絡を取って逐一調整をしていかななくてはならない。これは非常に面倒な作業である。

ある程度しっかりとした活動場所を持ち、組織的にしっかりとしているグループに関しては「部室に行く」などの一箇所のいわゆる「集合場所」があるから問題はない。しかし、ゼミのような流動的なグループ、しかも集合場所を持たないグループにおいては至極困難な話である。

マネージャーまたはグループのリーダーがメンバーのスケジュールを把握していれば、それでこのような問題は解決される。しかし柔軟性にかける。突然のスケジュール変更やキャンセルに対して完全に対応することが難しいからだ。また他のメンバーがそのスケジュールを確認することもある程度厄介なことである。何せスケジュール表を持っている人を捜さなくてはならず、その当の持っている人(リーダー)がつかまらなかったらおしまいである。

このような問題を解決するものは、「Web 上にメンバー全員のスケジュール表を置く」ことであった。インターネットに接続する環境さえあれば、誰でも参照することができる。スケジュール表を誰が持っているかなんて考えなくてもいい。とにかくパソコンがある場所



を捜せば、それで済んでしまうのである。これで簡単にスケジュールを組むことができるようになった。

しかし私は満足しなかった。スケジュール表はやっぱり「重い」のである。メンバー全員のスケジュールが一覧「できて」しまう。もちろんそれぞれがきちんとスケジュールを書き込んでくれていれば、それなりに正確なスケジュール表が手に入る。しかし、私が欲しい情報とは「彼(または彼女)が今つかまるかどうか」「捉まえるにはどうするのか」ということである。スケジュール表では十分すぎるほどの情報がありすぎる。

私はこのような困難に対して、なんとかならないものかと考えた。必要な情報を必要な量だけに限定しているスケジュール表が必要。そこで考え出されたのが「あの人 Finder」である。

## 2. 設計概念

一言で表すならば、「簡易版グループスケジュール表」である。それは「探す」ことに重点をおき、「いつ」「誰が」「どこにいるのか」を検索するシステムである。それゆえできる限り機能はそれだけに限定した。メンバーはあらかじめ自分のスケジュールを登録し、「あの人 Finder」を利用するときは簡単な

操作だけにしようと考えた。はじめは別の人が作っているスケジュール表と連動しようと考えた。しかしその試みは時間の都合で実現させることはできなかった。

## 3. 設計

「あの人 Finder」はあくまでも人を「探す」機能に限定している。したがって、スケジュールを登録するような機能をこのツールに実装することは設計の中には含まなかった。機能を大きく分けると以下のように分類される

1. メンバーのスケジュールを読み込む
2. ユーザーが入力した条件を読み込んで、データベースから読み込む
3. 検索して出てきた結果を表示する。

さらに自分が気にしていったのはできるだけ「軽く」作ることに重点をおいた。どのページに組み込んでおいても動けるようにするためである。それはまた「すぐに使える」ようにするための工夫でもある。

## 4. 実装

実装するための技術として、Java Applet を採用した。これには特別な理由はないが、自分の中で一番慣れ親しんでいる技術のひとつだったので、まずアプレットで作ることにした。

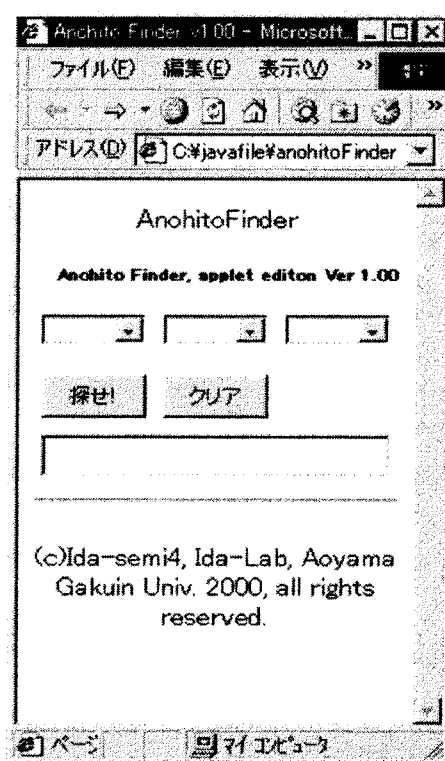



図 4-22

とにした。

プログラムを書いた順番は、以下のとおりである。

1. GUI 部品を配置
2. チョイスコントロールにパラメータを振る
3. それぞれのボタンにアクションを振る
4. データを組み込む

残念ながら、データを外部から取り込むというモジュールを  4-22 ができなかった。それは勉強が足りないことが主要因である。どうスケジュール表と連携を取ればいいのか分からなかった。今回は仕方がなかったので、ソースコードの中にスケジュールデータを入れることにして、そこから引っ張ってくるということにした。

一番技術的な部分で困ったところは、チョイスコントロールにイベントを付加する部分である。ここのところは自分では対処することができなかったため、大学院生にアドバイスをお願いして、できるようにしていただいた。

#### 5. 反応と今後の改良点

青山祭で実際にお客さんに使ってみてもらった。しかし実際に自分のスケジュールを入れてみるなど、使ってみないとわからない部分も多く、「これはいい！」といわれる事は残念ながらなかった。やはり時間をある程度とってもらって使ってもらわないと、本当の評価を得ることができないような気がした。

悔やまれる点が一番根本的な部分である、スケジュールツールのデータベースとの連携までたどり着くことができなかったことである。これができなかったために実際に運用するという段階までいたらなかった。これはスケジュールツールの仕様がほとんど最後まで決まらなかったことと、そのうえでどう接続すればいいのか、どの技術を使えばいいのか判断に迷っている時間が長すぎた。今後の改良点として、スケジュールツールで作成されるデータベースに加えて、グループのデータベースにも接続する方法を探る必要がある。

## 4-13 【私のツール製作体験記】

〈〈文責：黒田洋史

### サーブレットとの出会いとサーブレットエンジンのインストール

先生からコミュニティーポータルという題目をもらった時、正直言って困った。技術的にも経験的にも何にも無い自分の力でそんなものできるのかよと。プロジェクトを楽しもうって思っていたけど、いったい何をしたらいいのかなんて正直言ってわからなかった。そこでみんなで夢を語った。いろいろな夢が出た。そんな中でみんなで漠然とあがっていたものがある。それは、自動的に更新されるということである。管理者がいなくても、自

分が書き込んだことがグループのページにすぐに反映される。そんなものができたらいいねと話していた。僕の中でも自動更新がひとつの目標のようになった。でも、そのために何が必要かはわからなかった。そんな中、定期試験もありあつという間に、7月も半ばに入っていた。どうしたらいいのかお互いいろいろなサイトや本を調べたものが出てこなかった。そんな中僕にとってひとつの方向性を示してくれた本があった。それは、アメリカに旅行に行く前 **macci** がたまたま電話で教えてくれていた **Java press** である。その本の特集に DB との接続の仕方が載っているとされた。そして本屋に行って実際に手にとって読むとこれだと直感して、すぐ購入した。そして、初めてデータベースとサーブレットのことを知る。なんか、正直言って本を読んでもわからなかった。だから実際に **postgreSQL** と **TOMCAT** をインストールして試してみようと思った。本に書いてあったサイトを参考にまず、**TOMCAT** をインストールしてみる。インストールはうまくいったようである。しかし、サーバを立ち上げようと **start** を打っても動かない。本にもサイトにも書いてない。起動の方法なんて本当に基本的なこと書いてないのだろう。でも僕はわからない・・・(でも、後に本当に基本的なことであるというのに気づくのだが・・・それは2ヶ月以上たったあとである)。気をとりなおして、**postgreSQL** をインストール。これはうまくいったようである。でも、**postmaster**(**postgreSQL** を起動するコマンド)を打っても動かない。**postgreSQL** の前のバージョンがありどっちかわからないというエラーが出ている。なので、**postmaster** があるディレクトリ(**/usr/local/pgsql**)を直接示し打ち込む。そうすると動く。どうも、僕が使っている **Linux(Vine)** にはもともと **postgreSQL** が入っていたようである。あー、そんなこと知らないよ。でも、どう消したらいいかわからない。まあ、**postgreSQL** は動くしいかなって思い、放って置いた。そして、**JavaPress** やサイトを参考に **psql** 上で **SQL** をうち、データベースを作ったりして遊んでみた。結構面白かった。でも、データベースで遊んでいただけで、**postgreSQL** は僕にとっての高級な玩具のような状態がほぼ2ヶ月続いた。

## Java のお勉強

8月から **EC-ONE** でインターンさせてもらい、**keiko** と大学院生の井田雪乃さん(雪乃さん)と、一緒に働かしてもらうことになる(僕にとっては働くというより勉強させてもらうといったほうがいいかもしれない)。そこで、サーバ周りのシステムについて勉強させてもらうことになる。その前に、**Java** の基礎を勉強するというので、「独習 **Java**」という本を使って勉強する。幸運なことに、**Java** の勉強に1週間前後時間をもらえた。ひたすら、**PC** に向かい作ってみた。この雪乃さんに勉強しやすいということで紹介してもらった「独習 **JAVA**」という本は、説明と例題がセットになっていて受験時代の参考書を思い出す。でも、この方式が逆に僕のやる気を刺激してくれたような気がする。勉強したことを実際に作ってみて実感しながら進むという流れが良かった。だから、勉強しているのが大変ではなかった。ばんばんこなせていけた。量をこなしながら、**Java** のデータ型についてやクラスやオブジェクトの関係など実際に **Java** をやっていくうちに大事になっている知識を勉強できた。そういう意味で最初の1週間は自分にとって土台を作ることができたな

あとと思った。また、そういう中でアプレットで作っていた認証（パスワードと ID をソースに入れてあるというものだが）を勉強したばかりの多元配列という多対多の組み合わせの配列を使って作り直したりしてみた。そして、まるで英語を学んだあと無理やり文を作って遊んでいる中学生みたいに遊んでいた。でも、面白かった。

また、7月中にインストールした TOMCAT のことを雪乃さんに聞いてみた。そうすると、実験環境で使うには TOMCAT より JSWDK のほうが良いという話を聞いた。また、本運用には apache+J-Serve のほうが良いということも聞いた。また、サーブレットエンジンのことも少し聞いた。そこで、試しに WINDOWS に JSWDK をインストールしてみた。環境変数など、聞き慣れない言葉に遭遇しながらやった。一番詰まったのは、環境変数の書き換えだ。しかしながら、WEB のページが詳しく書いていてその通りやったり、JAVA カンファレンスのサーブレット部会の ML のアーカイブを参考にしたら何とかできた。こういう時に役に立つのが ML のアーカイブだった。というのは、僕みたいに詰まった人が ML に投稿していてその質問とそれに対する先輩(上級者)の答えがそこにあり参考になったからだ。また、同時に悩んでいるのは自分だけではないということが実感でき、やっていく勇気ができた。一人だけの問題じゃなかったのだという実感が何度も自分を救ってくれたような気がする。そして、やっとサーバが動いてくれた。そしてそこにある、サーブレットでできたサンプルデモを触ったりして遊んでいた。デモ、デモがどうして動いているのかはわかっていなく魔法で動いているような感覚だった。

### サーブレットのお勉強

Java の勉強がある程度進んだあと、servlet の勉強をさせてもらった。はじめは、雪乃さんが書いたプログラムや WEB 上にある勉強用サイトのプログラムの真似をしたりした。実際自分でうってみるといろいろなことが分かった。一つが、データの型がとても重要であるということである。Java にはデータの型が多く存在する。文字列の String 型、数値の int 型など多くのデータの型が決まっている。データ型を無理やり変更したりすることはできない。それを変更するメソッドが決まっています。このデータ型という話は servlet だけでなく、Java でも問題になることであるし基礎中の基礎である。しかしながら、初心者の僕にとってはとても難しくはじめはエラーが出たり困った時は重たい本を広げ見ながら文を入力した。この時作ったプログラムははっきり言って雪乃さんが作ったプログラムを直したりしただけなのだが、出来上がりがしっかりとコンパイルできた時とても嬉しかった。こうやって人が作ったプログラムを読んで真似して作ってそれを実際に動かしてその動いたことに対する喜びを糧に学んでいくという「まねぶ」ということはとても重要なことであるなあと実感した。そんなこんなで勉強を進めていき、だんだんと頭の中でプログラムが作ることができるようになり、本に頼らないでできるようになった。そういう勉強と同時に、自分でサイト構築時に使われるシステムの中身の部品を雪乃さんに頼まれ作るようになる。実際にサイトに組み込まれると思うと少し緊張したけど、実際に作っていく過程からも様々なことを学べると同時に、出来上がった時の喜びはすごかった。出来上がるたびに万歳をできてしまっていたのを記憶している。でもこの時の小さな部

品作りがすごく自分のなかでの Servlet の面白さを教えてくれたものであったと思う。

### ツール作成とそこで感じたこと

そのあと、だんだんとツールを作り始める。そうするといろいろなことがわかってきた。その一つ目が 2 年次に授業で学んだ Applet と違う様子であるということである。Applet の場合、エラーはコンパイル時にほとんど出てくれ実行時にはあまりエラーは出ない。一方 servlet の場合はコンパイル完了までは多くの場合は行き着くのであるが、実際にテストしてみるとエラーの画面がブラウザに表示されることが多かった。エラーの画面さえも出ないということもあり、そういう時のデバッグの作業は大変だった。ソースの所々に持っているパラメータをサーバのプロンプト画面に表示させたりプログラムを断片的に細かくして動かしてみたりする作業が多々必要だった。しかしながら、このデバッグの作業でパラメータを吐き出させてみることで実際に自分のプログラムが動くのを確かめられるのが面白かった。

2 つ目が、設計と実装の二つの難しさを体験したことである。設計と実装という二つの言葉を聞いたのは合宿の時だった。その前までもプログラミングはしてはいるが実際には二つとも体験はしていたのだが実感してみると自分がかっこいいことをやっているような気がして面白かった。みんなの夢やほしい機能を聞き盛り込む。それらをもとに必要な機能を絞り込み紙に書いてみる。そしてそれをアルゴリズム化してみる。そしてそれをもとにデータベースなどの設計もする。ここまでが設計であるがとてもここまでの設計のほうが実際にプログラムするより時間がかかったしすごく頭を使った。プロの人なら別かもしれないが超初心者の僕にとって実装しながら全体の設計を考えたり、プログラムの流れを考えることはできなかつた。なので、設計する時点でしっかりしたものを考え紙に書いておかないと実装中に混乱してしまうのでそういう中で緊張して設計した。そして、僕が設計を書いたノートは 2 冊にのぼり自分にとって大切な宝物になっている。一方、実装も設計で考えたアルゴリズムの工程一つ一つをもっと分解し考えていく作業が多く、実装で間違えると設計も無になるというプレッシャーもあり実装は実装で大変であった。自分が作成したツールの多くは設計から実装まで担当したが、幾つかは設計と実装を別の者が担当したものもあり、設計者と実装者の間では何度も話し合い設計内容を設計者と実装者のお互いで吟味する。そして実装者は実装しそれを、また再びテストし設計者の夢が実現されたか確認してもらおう。このような議論とチェックをしてもらうこの作業は特に面白い。僕は実装を担当することが多かった。設計者と話したものを実現して作れることに喜びを感じたし、出来上がりを嬉しそうに設計者にチェックしてもらおうのが嬉しかった。実装者と設計者の間の関係はコミュニケーションをするには大変な面もあったがお互い良いものを作ろうという風にながらばれたのは良い刺激になった。

三つ目はコミュニケーションが普通以上に重要であるということである。みんながみんな、Java ができるわけではない。みんなの中にはサーバの設定などを中心に勉強して作業する者、アンケートを取ったり、発表展示の舞台である青山祭の内容を決めたりする者と一人一人が専門の分野を持って進めているので Java の知識に関しても差が大きい。Java

## 4-13【ツール説明書】

〈文責：大西淳

青山祭の当日、皆様にそれぞれのツールがどのようなものか、理解していただくためにパソコンの横にツールの説明を簡潔にまとめた、くるくる回転する、メニューのようなものを針金で作りました。その針金を骨組みにして、円柱状に組み立て、以下の文書を貼り付けました。

### 周辺情報：

青山学院大学の周辺情報(居酒屋、レストラン etc)を見ることができます。また、見るだけでなくあなたが知っている情報を書き込んで、他の人たちと情報を共有することもできます。

### 講義評価：

教授が生徒を評価するのではなく、ここでは生徒が教授の講義を評価するものです。評価を投稿したり、他の人が評価したものを閲覧することが可能です。新年度に授業を選択する際の大きな手助けにきっとなるでしょう。

### LIDA:

本の蔵書情報を見られる機能で、検索画面によって簡単に検索できます。今後は貸し出し情報を入れることで貸借の状況が自分で行かなくても見られます。つまり図書館に行かずに確認することができるのです。

### グループメンバーのプロフィール表示：

年末の年賀状書きなどの使いたい時に見つからない住所録。これをWEB上で作ってみました。データは常時Web上で更新され、紛失の心配もありません。

### 投票システム：

グループで活動するとき決を採らなければならない時が多々あります。全員が顔を合わすことができないと大変です。それをWEB上で行うことで、決を採る人は締め切り日に結果を見るだけでよくなりました。同時にその結果を全員で共有することができます。

伝言板：

グループ間でみんなに連絡したいことがあるので、WEB 上で見かけるいわゆる掲示板を置きました。グループに所属している人のみ見ることができるので、荒らされたり管理を強化する必要が無く安全に利用できます。

LINK 集：

各個人が見つけた良いサイトをグループで共有する機能です。ただのリンク集ではなくそのリンクを使った人の数を集計しその順番に並ばれるので有用なページが比較的簡単に見つかります。

個人用スケジュール管理：

あなたのスケジュールを管理するツールです。あなたのスケジュールをそのまま書き込むもよし、日記を書いていくもよし(自由度を高くしました)。色々な用途に使えます。

グループのスケジュール管理：

所属するグループのメンバーのスケジュールを管理するツールです。メンバーの予定を確認する事ができます。複数のグループに所属している場合、各グループのメンバーに適した表示をします。

あの人 Finder：

グループメンバーが「いつ」「どこに」いるのか検索できるツールです。日時を指定し、メンバーの名前を選択すると、その人がどこにいるのか瞬時に出てきます。

k3 君 1 号(GPA 計算機)：

GPA という平均評定のようなものを計算する電卓です。左側のボックスに科目の単位数を、右側のチョイスボックスからその科目の成績を選択すると、その GPA を算出します。同じことを繰り返すことによって、取得科目全体の GPA を計算することができます。これで面倒、分かりにくいことで有名だった GPA を簡単に計算できます。

k3 君 2 号(単位計算機)：

カリキュラムを編成するときに、時間割に書き込む要領であとどれだけ単位を申請できるか、また卒業にどれだけ単位を申請しないといけないのかを一発で計算できます。た

だし、全学共通科目と、各学部専門科目、また外国語専門科目など、個別に要件単位が異なるので完全な卒業要件単位の算出は現在できません。これによって来年春のカリキュラム編成がぐっと楽になります。



# 第5章

## Web デザイン

5-1【Web サイト・デザイン】	p.122
5-2【Web サイト作成開始】	p.122
5-3【Web サイト・デザイン＝機能】	p.122
5-4【Web サイト・デザイン＝見栄え？】	p.122～
5-5【人の振り見て我が振り直せ】	p.123
5-6【そのサイトはどんなサイト？】	p.123
5-7【ナビゲーションの大切さ】	p.123～
5-8【フレームの問題点】	p.124～
5-9【ナビゲーションに JavaScript?】	p.126
5-10【レイアウト、配色】	p.127
5-11【ぱっと見てわかる】	p.127～
5-12【リンクフリー？】	p.128～

5-13【クロス・ブラウザ】	p.129
5-14【誰もが利用できる】	p.129
5-15【他の人に見てもらおう】	p.130
5-16【ついに完成!! しかし、・・・】	p.130～
5-17【WEB サイト・デザインのまとめ】	p.132～

# 第5章 Web デザイン

＜＜文責：清水淳

## 5-1【Web サイト・デザイン】

(注)このページ以降では、「Web サイト」「ホームページ」「Web ページ」を次のように区別しています。

Web サイト…Web ページの集まり。

ホームページ…Web サイトで、初めに開かれることを想定して作成されたページ。

Web ページ…ホームページを含めた全てのページ。

## 5-2【Web サイト作成開始】

9月中旬、Web サイトがないと何も始まらないこと気付き、それまでの Java によるツール作成を諦めて Web サイトを作成し始めました。とは言うものの、何から始めていいか分かりませんでした。これまでもいくつか Web サイトを作成してきたものの、それはあくまで個人の趣味の範囲を出るものではありません。

そこで、Web デザインに関する本を1冊購入し、Web サイト・デザインとはどういうものかを勉強することから始め、それと同時に実際にさまざまな Web サイトを見て研究しました。その過程の中で、素晴らしい本、Web サイトに出会い、非常に多くのことを学びました。これは、その経験にもとづく記録です。

## 5-3【Web サイト・デザイン＝機能】

まず始めに言いたいことは、Web サイト・デザインはそのサイトの「機能」の1つだということです。例えどんなに素晴らしいシステムを作ったとしても、それを利用する人がいなければただの無用の長物に過ぎません。システムの本当の力を発揮させるのは Web サイト・デザインです。多くの人を呼び込み、システムを活かす機能、それが Web サイト・デザインなのです。

## 5-4【Web サイト・デザイン＝見栄え？】

一般的に見て、Web サイト・デザインというと、見栄えの問題と受け止められがちです。僕もそれが一番重要だと思っていました。しかし、そうではありませんでした。もちろん見栄えも重要な要素ですが、それ以上に重要なことがあるのです。そのことを教えてくれた1つの言葉があります。「ユーザビリティ」という言葉です。ユーザビリティとは簡単に言えば使いやすさということです。そのサ

イトがどれだけユーザーの立場に立って作成されているかということです。

## 5-5【人の振り見て我が振り直せ】

今回のサイト作りでもっともよい手本となったのは、素晴らしいと言われているサイトではなく、見るのに好ましくないとされるアダルトサイトや違法サイトでした。Web サイト・デザインに関しての知識がない状態で素晴らしいと言われているサイトを見ても、何が素晴らしいのか分からないし、見たものをどう反映させればいいのか自分でもよく分かりませんでした。その分野に関する良さを見分ける能力、説明する能力が身につけていなかったからです。逆に、人間という生き物は不思議なもので、あれが悪いこれが悪いというのはなぜかとてもよく分かるのです。そこで、そのようなサイトの悪い点を自分なりに整理してみることにしました。まずは自分の分かるところから始めたわけです。まさに、「人の振り見て我が振り直せ」の言葉通りです。そうすることで、今までは分からなかった素晴らしいサイトの良さがはっきりと見えてくるようになりました。そうなれば、次にやるべきことは、実際にサイトを作ることです。

## 5-6【そのサイトはどんなサイト？】

実際にサイトを作り始める前に知りたかった情報は何よりもどのようなコンテンツを提供するのかというでした。具体的にはどんなコンテンツを作るのかが分からなければ、そのサイトはどんなサイトで何をするためのサイトなのかも分かりません。あやふやな気持ちのままですべてのサイト作成をせざるを得なくなってしまう。それでは出来上がったものも中途半端に終わってしまうでしょう。やはり、Web サイトに最も重要なのはコンテンツなのです。

## 5-7【ナビゲーションの大切さ】

実際にサイト作成でもっとも悩んだのがナビゲーションをどうするかということです。ナビゲーションとは、カーナビゲーションに代表されるように、「進む方向を示すこと」です。Web では「コンテンツへたどり着くための手助けをすること」と考えればいいでしょう。

デパートへ買い物に行ったときのことを想像してみてください。そのデパートへ行くのは初めて、どこのフロアに何が売っているのか全く分かりません。そんなときは、案内係に聞いたり、パンフレットや案内板を見たりするでしょう。Web でもそれは同じです。違うのは、現実世界ではいろいろな手段が考えられますが、Web には画像とテキストがあるだけだということです（音声も考えられますが、実際に機能するかは分かりません）。その限られた手段を有効に使って、サイトの中でいかにユーザーが迷わないようにナビゲートするかが重要なのです。

このナビゲーションをどうするかは最後の最後まで悩みました。展示中に公開したサイトの中で

は、フレーム機能と、JavaScript を利用して作った小さなウィンドウをリモコンに見立てたものを併用しました。しかし、これはどちらも工夫はしたもののとても使いやすいとは言えないものでした。完全に失敗です。

## 5-8【フレームの問題点】

ナビゲーションとしてよく使われるのは、フレーム機能(以下、フレーム)です。フレームはとても便利です。フレームを使わなければならないようなこともあります。初めてフレームを使ったときは純粹にすごい機能だと感動し、その後自分でサイトを作るときもよく利用しました。しかし、次第にフレームが持つ欠点が見えてきました。その問題点とは、

- ① ユーザーの画面を制限してしまう
- ② 使い方によっては、著作権に違反する可能性がある
- ③ 思うように機能してくれない⇒サイトを管理するのが難しい

などです。

- ① パソコンにはさまざまな種類があります。ノート型パソコンを使っている人もいれば、デスクトップ型パソコンを使っている人もいます。そして、それぞれで画面(モニタ)の大きさはバラバラです。特に、ノート型パソコンを使っている多くの人は基本的には小さな画面を使っています。そのような人たちは、ただでさえ小さな画面なのだからその画面を目一杯使いたいと思うでしょう。しかし、フレームを使うことでこれを妨げてしまうことになります。大きな画面では使いやすくても、小さな画面では使い勝手を著しく低下させてしまうことになってしまいます。実際、フレームを使って最初に作成したサイトを 800×600 の画面をもつ Libretto100 で見てみると、自分のパソコンの画面(1024×768)では問題なく見ることができていたのにもかかわらず、今まで表示されていなかったスクロールバーが表示されるようになってしまいました。

HTML の策定を行っている事実上の標準団体 WWW Consortium(以下、W3C)は上記のような理由から、次世代の HTML である XHTML ではフレームはサポートしないことを決めています。以下にその部分を引用します。

### 1.3.6. Frames

Frames are not supported. Frames depend on a screen interface and may not be applicable to some small appliances like phones, pagers, and watches.

(WWW Consortium, 2000)

- ② 著作権に違反する可能性があるということの根拠は URL が機能しないということにあります。次の2つの図(上-図1、下-図2)を見比べてみてください。

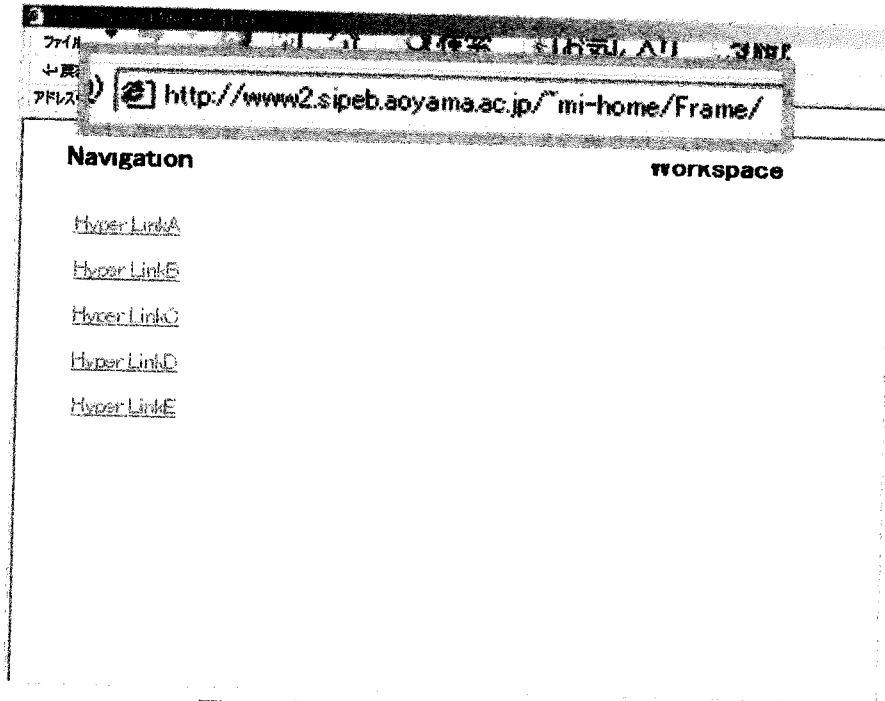


図 5-1 右のフレームにはまだ何も読み込んでいない

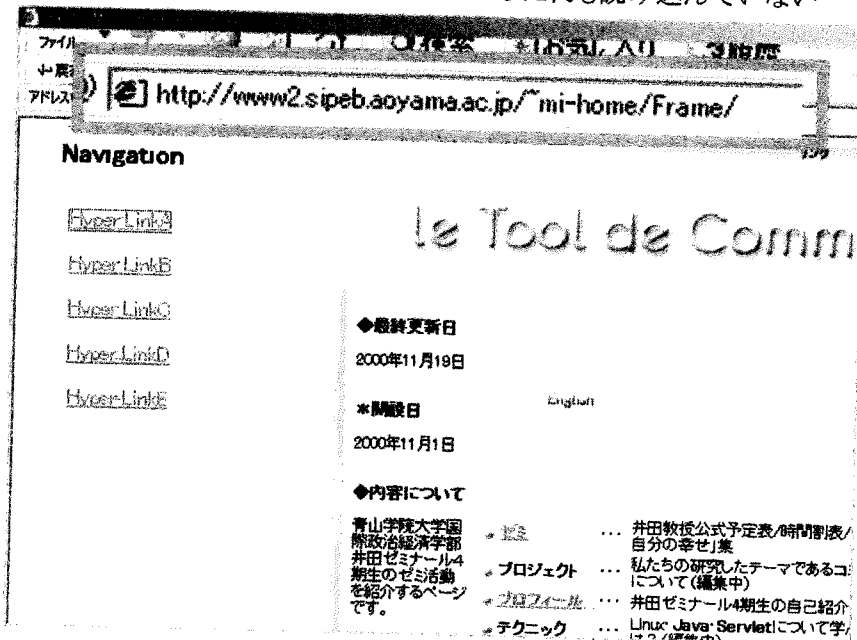


図 5-2 右のフレームに違うサイトを読み込んだ

図 5-1 では、右側のフレーム(「Workspace」)には何も読み込まれていない状態です。図 5-2 では、右側のフレームに違うサイトの内容を読み込んでいます。しかし、アドレス入力欄の URL (Universal Resource Locator) に変化はありません。つまり、URL が機能

していないのです。言ってみれば、自分の家の住所を勝手に書き換えられたのと同じことです。これは、リンク先を新しいウィンドウで開くことで解決できます。しかし、次から次へと開かれるポップアップウィンドウ形式の広告を思い出してみてください。不快な思いをしたことがあるはずです。さらに、複数のウィンドウを切り替えなければならなかったり、新しく開いたウィンドウによってツールバーが占拠されてしまったりします。小さな画面になればなるほど、この問題は深刻になります。また、複数のウィンドウを開きすぎると、低スペックのマシンではフリーズする原因となります。一度でもこのようなことが起これば、ユーザーは二度とそのサイトを訪れてはくれないでしょう。

- ③ これまでの経験からフレーム機能を使ったサイトは、そうでないサイトより管理が難しくなってしまう。1つのページ作る場合、フレームを使わないページなら1つのファイルを作ればいいのですが、フレームを使ったページでは最低でも3つのファイルを作らなければなりません。そして、フレームをサポートしていないブラウザ用に別のページを用意しなければなりません。単純に考えれば、それだけでもその労力は4倍もかかるわけです。また、リンク先の変更などにはとても神経を使います。さらに、そのサイトの構造が複雑になってしまいます。将来にわたって一人の人がそのサイトを管理してゆくのならいいのですが、多くの場合はそうではないでしょう。

このようなことを考えると、フレームを使うことはなるべく避けたかったのですが、その時点ではいい代替案が思いつかず、仕方なくフレームを使うことにしました。その場合でも、横スクロールバーが表示されるのは避けたかったので、上下に分ける方法をとりました。縦スクロールバーは表示されてしまいますが、横スクロールバーが表示されるよりもストレスを感じる数が少ないという統計結果もあるので、フレームによる弊害を最小限に抑えることができたのではないかと思います。

僕が言いたいのは、フレームを使ってはいけないということではなくて、なるべくなら使わないほうがいいということです。そもそもフレームはナビゲーションのことを考えて考案されたものです。ナビゲーションとして正しく使うのなら問題はないと思います。その場合でも、他に代替案がないか考えたほうがいいかもしれません。もちろん、それ以外の場合は乱用すべきではありません。補足ですが、フレームに代わる案として、W3Cはスタイルシートを使うことを薦めています。

## 5-9【ナビゲーションに JavaScript?】

ナビゲーションに JavaScript を利用したのですが、これは大失敗でした。JavaScript はブラウザによって正しく動作しないこともあるからです。もし JavaScript が動作しないブラウザを使用した場合、ナビゲーションが機能なくなってしまうわけですから、その存在意義がなくなってしまいます。まるで、幽霊のような存在です。見える人には見えるけど、見えない人には全く見えないのです。重要な部分に、不安定な技術は使わないほうが安心です。

## 5-10【レイアウト、配色】

テーマがコミュニケーションなので、なんとかして楽しさや開放感を出したいと思いました。そのため、レイアウトには左右上下を囲むようなレイアウトでは閉塞感がでてしまいます。そこで左右上下には何も手をつけませんでした。コンテンツの配置も一区間に小さくまとめるのではなく、画面を広く使ってなるべく分散させるようにしました。そうしたことで、モニタという小さな限られたスペースの中で、広がりをもたせることができ、開放感も出しました。加えて、T字型に「こみゅぽ」の基調色である緑色を置くことで、乱雑さを出さないようにしました。

楽しさを表現するために、配色には気を配りました。メンバーの中からは同系色でまとめてほしいという意見が多くあったのですが、そうしてしまうと落ち着いた雰囲気が出すぎてしまいます。格調高さを表現したいのなら、そうしたほうがいいでしょう。しかし、表現したいのは、楽しさや賑やかさです。これを表現するために、全色相を使い、明度、彩度を高くしました。全色相を使うことで、楽しさや賑やかさ、そして華やかさを出すことができます。

配色はとても難しく、苦労しました。色にもそれぞれイメージがあります。配色の効果は大きく、それだけでそのサイトのイメージを決めてしまいます。しかも、そのイメージはなかなか変えることはできません。それだけに、慎重に決めなければなりません。

余談ですが、配色は人間の行動にも大きく影響を及ぼすことはよく知られています。例えば、黄色と赤色の組み合わせは食欲を高めるそうです。そういえば、マクドナルドやモスバーガーなどのファーストフード店のトレードマークは黄色と赤色の組み合わせが多いですね。

## 5-11【ぱっと見てわかる】

パソコンを使っていて常々思っていることは、「なんて分かりにくいんだろう…」ということです。ボタンの数がやたらと多かったり、しかもそのボタンを押したら何が起こるかいまいち分からなかったり。いろいろいじってみたいけど、パソコンが動かなくなったら困るし(実際に何度も動かなくなったこともしばしばあります)。ヘルプを読んでも、難しい用語が多すぎて何がなんだかさっぱり。ヘルプを読むためのヘルプが必要なんじゃないかと思ってしまうのです。

せっかく多くの人が自分のパソコンを持ち始めたのに、こんな状況ではその恩恵をうけるのは難しいでしょう。こんな感じだから、「コンピュータはマニアがやるものだ」なんて言われてしまうのではないのでしょうか。

なんとか分かりやすくできないだろうか、というのが大きな課題でした。ただ作って、できたから使ってくださいというのは誰にでもできます。作った側からみれば、自分が作ったものなのだから簡単に使えて当たり前です。大事なのは、初めて使う人が簡単に使えるかどうかです。

そこで、どのコンテンツで何ができるのか説明書を加えるようにしました。今まで分かりづらいつ感じていたのは、説明が十分にされていないからなのです。ただ、説明が詳しければ詳しいほど



それを読まないというのが人間の心情のようですから、なるべく2~3行の簡潔で、簡単な言葉を使った説明にとどめました。ちょっとした説明でもあれば、それだけで安心できるものです。

これはリンクに関しても同じことが言えます。どのサイトでも「リンク」というコンテンツがありますが、そのほとんどはただリンク先が書いてあるだけのことが多いようです。リンク先はどんなサイトで、何ができるのかは書いてありません。いちいちそのサイトに行って確かめる必要があります。その数が10くらいならいいですが、100、200となるとそんなことをしている余裕はありません。リンク先のサイトが自分にとって役にたつサイトなのかどうかは、勘や運に左右されます。これでは非常に効率が悪いといわざるを得ません。それがインターネットというものだと言ってしまえばそれまでなのかもしれませんが、リンクを張っているということは、そのサイトを一度でも訪れたことがあるはずですから、そのサイトがどんなサイトなのか分かっているはずで、分かっているなら、それを書けばいいだけです。たったそれだけのことで、ただのリンク集からずいぶんと使い勝手のいいリンク集に生まれ変わります。

## 5-12【リンクフリー？】

学生生活に役立つリンク集をつくらうと思ったのですが、ちょっとした疑問が浮かびました。「リンクって張るのに許可はあるのだろうか」。これまでにたくさんのサイトを見てきましたが、「このページはリンクフリーです」というようなメッセージが書かれていることが非常に多いです。ということは、逆を言えば、他のサイトへリンクするのに許可はあるのでしょうか。そこで、いろいろと調べてみました。結論から言うと、無許可で他のサイトにリンクすることは、法的な観点からみれば、基本的には問題はないようです(上でふれたフレームを使った場合などには注意が必要です)。つまり、リンクに許可はいらぬということになります。また、リンクに許可は必要ないとする人はたちの多くは、World Wide Web の開発者として知られる Tim Berners-Lee 氏の主張を根拠にしているようです(『Design Issues - Architectural and philosophical points』のうち『Links and Law: Myths』<URL: <http://www.w3.org/DesignIssues/LinkMyths.html>>、1997)。

- There is no reason to have to ask before making a link to another site
- You are responsible for what you say about other people, and their sites, etc., on the web as anywhere
- We cannot regard anyone as having the "right not to be referred to" without completely pulling the rug out from under free speech.

ただ、「リンクはホームページにして下さい」というような要望があればそれに答えるべきではないかと僕は思います。というのも、ホームページの URL に比べて、その他の Web ページの URL は変わってしまう可能性が高いからです。多くの Web ページから構成されているサイトの場合は特にそうです。サイトの構成上、ディレクトリ構造を変えざるを得ないこともあります。

単純にリンクするだけなら問題はないということは分かりました。しかし、どのページに、どのようにリンクするかはまた別の問題だと思います。例えば先ほどのフレームに関する問題でいえば、フレーム内に別のサイトを読み込む場合は、その読み込まれた内容は自分書いたものではないことを目に付くような形で明記すべきでしょう。

補足として、現実には「リンクフリー」と明記したほうがユーザーを増やすのには効果的です。

## 5-13【クロス・ブラウザ】

だんだんと完成に近づいてきたところで、思わぬミスをしてしまいました。Netscape Navigator (以下、NN)での動作確認を行っていなかったのです。マズイ…、と思ってすぐに NN で表示してみました。心配した通り、Internet Explorer (以下、IE)と同じようには表示されませんでした。何でこうも違うのかというくらい、NN ではぐちゃぐちゃでした。これまでの経験から考えて、IE よりも NNの方が HTML の文法に厳格のようです。つまり、これは当たり前なことなのですが、NN では正しく書かないと正しく表示してくれないのです。IE では多少の文法の誤りに対しては柔軟で、ちゃんと表示されてしまうので、その間違いには気付きませんでした。しかも、JavaScript やスタイルシート、DHTMLを使った場合には注意が必要だったのです。IEとNNではこれらに対する対応が全く異なるからです。それぞれが独自の拡張を行っているため、一方では表示されても、もう一方では表示されないというケースが多くあるのです。その時は、そのことをすっかり忘れていたのです。

ここからの修正が大変でした。細かい文法チェックを何時間もかけて行いました。正しく書いているはずなのに、思ったように表示されないということもありました。それはそれで仕方がないと諦めなければならない部分もでてきました。結局、このような修正を最後の最後まで続けることになってしまいました。

メンバーからは、とりあえず IE だけで表示できればいいと言われてたりもしましたが、その意見はどうしても受け入れることはできませんでした。最低でも、IE と NN でだけは同じように表示させたい(=クロス・ブラウザ)というのが僕のこだわりだったのです。

## 5-14【誰もが利用できる】

1 つのテーマとして、誰もが使えるようにするということを考えていました。具体的にしたことは、①画像には必ず alt 属性をつける、②色に頼るようなページにはしない、ということです。この2つのことは視覚に異常を持っている人にとっては深刻な問題だからです。

- ① alt 属性というのは、画像が表示されないとき、代わりに表示されるテキストのことです。alt 属性を指定しておかないと、画像が表示されないときには何も見えなくなってしまいます(画像が破損していることを示すファイルは表示されます)。しかし、alt 属性を指定してお

けば、画像の代わりにテキストが表示される(図 5-4)ので、画像が表示された場合(図 5-3)と同じように利用できます。

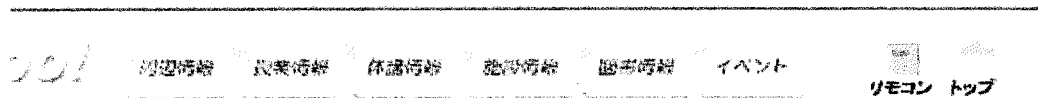


図 5-3 画像を表示した場合

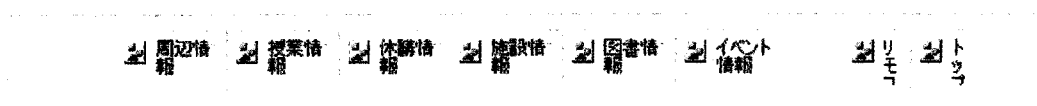


図 5-4 画像を表示しない場合

- ② 「赤いボタンをクリックしてください」としてしまうと、赤色と緑色を見分けるのが難しい人は、どうすることもできなくなってしまいます。このようなことを避けるためにも、色に頼ったナビゲートするのは避けるべきです。

## 5-15【他の人に見てもらおう】

このような修正を繰り返すと同時に、何度もメンバーのみんなに動作チェックを行ってもらいました。自分では大丈夫だと思っているところほど、ちゃんと表示されていなかったり、分かりにくかったりして、その部分の修正にも多くの時間を割かなければなりません。しかし、この第3者からのチェックが最も重要な部分だということはわかっていたので、妥協はしませんでした。もしかしたら、メンバーのみんなはいい加減しつこいと思っていたかもしれませんが、ここで妥協をしてしまうと全ての価値を下げてしまうことにもなりかねないので、何度もチェックを繰り返したのです。

## 5-16【ついに完成!! しかし、・・・】

こうして出来あがったものが、青山祭で展示した「こみゆぽ」でした(図 5-5)。その時点ではそれなりに満足し、訪れた人からも概ねいい評価を受けていたのですが、それは学生として見た場合のこと。さらにいろいろな人の意見を聞いていると、今まで見えなかった問題点が見えてきました。

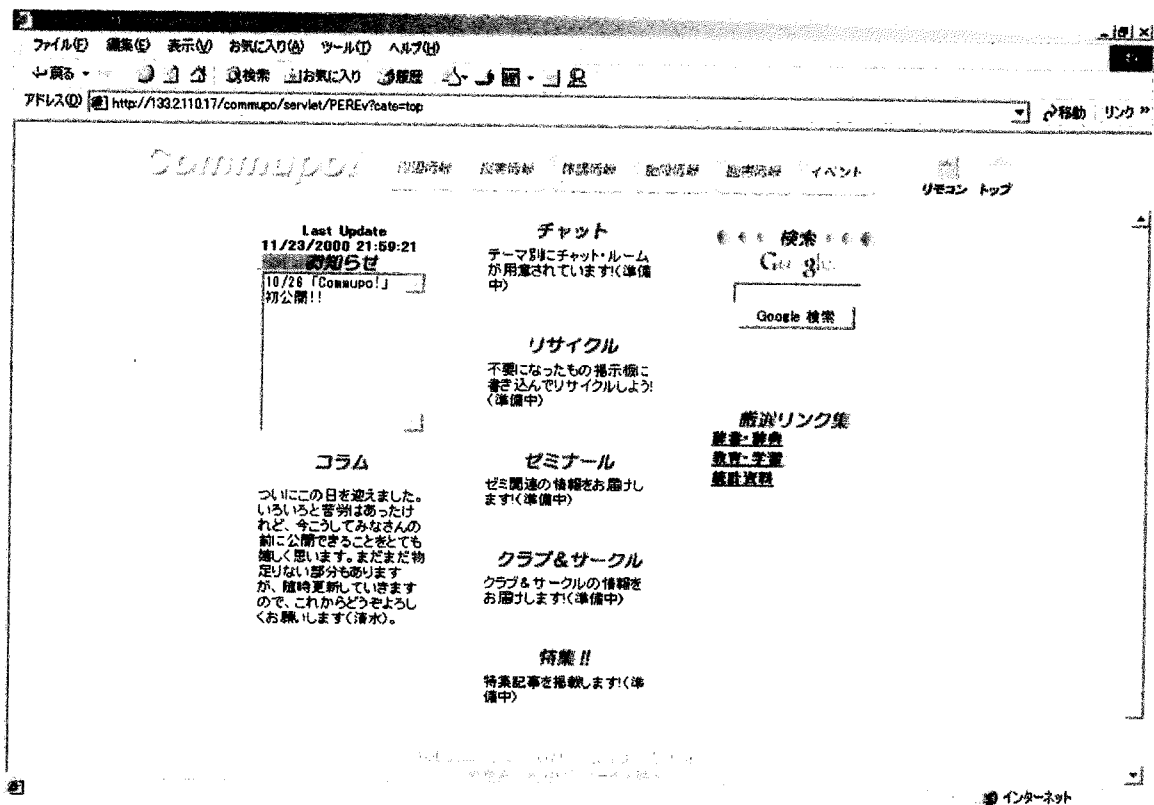


図 5-5

そこで、ここまでの問題点を修正して、井田ゼミナール 4 期生の Web サイト (URL: <http://www2.sipeb.aoyama.ac.jp/~mi-home/semi4/>) を作成しました。ここで得た知識を生かして新しい「こみゆポ」を作成する予定です。是非、ときどき覗いてみてください。今よりもっと進化した「こみゆポ」をお見せできると思います。

## 5-17【WEB サイト・デザインのまとめ】

いい Web サイトを作る要素とは、「使いやすさ」「分かりやすさ」「見やすさ」「アクセシビリティ」を備えているかどうかだと思います。もちろん、その作ろうとしているサイトの性質によって、選択する項目と、切り捨てなければならない項目がでてくるのは当然です。それを見極めるのも Web サイト・デザインの重要な部分だと思います。

### 1. 使いやすさ

#### 1) 適切なナビゲーション

ナビゲーションとは、コンテンツへたどり着くための手助けをすることです。自分が今どこにいて、そこからどう進めば目的のコンテンツまでたどり着けるかを示したものです。Web サイトが使いやすいかどうかは、ナビゲーションが使いやすいかどうかで決まると言っても決して大げさではありません。

ナビゲーションに JavaScript などを使うのは好ましくありません。すべてのブラウザで使用した JavaScript が同じように動作するとは限らないからです。使う必要がある場合には、少なくとも Internet Explorer と Netscape Navigator のいくつかのバージョンでちゃんと動作するかどうかをチェックする必要があります。

ナビゲーションに画像を使う場合は、画像を表示させない、または出来ない場合に備えて、それとは別にテキストでのナビゲーションも用意すべきです。

#### 2) 情報へのアクセスしやすさ

ホームページから目的のコンテンツまでにクリックしなければならない数はなるべく 2 回までにとどめます(メニュー⇒サブメニュー⇒コンテンツ)。何度もクリックしているのにいつまでたっても目的のコンテンツへたどりつけないのでは、ユーザーをいたずらにイライラさせるだけです。

#### 3) インターフェースへの工夫

例えば、むやみに点滅するボタンを使ったり、ステータスバーに任意の文字を流したりするのは避けるべきです。点滅する、またはアニメーションは注意をひきつけるのに有効な手段ですが、使いすぎると逆に注意が散漫になってしまい逆効果です。また、ステータスバーに文字を流すと、リンク情報が表示されなくなってしまいます。余計な機能をつけることに時間を費やすより、どうしたら使いやすくてできるかに時間を費やしたほうがいいでしょう。

#### 4) 逃げ道をつくる

全てのページに「ホームページ」へ戻るボタンを用意しておきます。そうすれば、サイト内で迷ってしまってもまた初めからたどり直すことができます。また、「ホームページ」以

外からたどりついたユーザーも、難なく「ホームページ」に行くことができます。

#### 5) 十分、かつ簡潔な説明

それぞれのコンテンツに対して、そのコンテンツでは何ができるのかを説明してあげます。そうすれば、自分に有用なコンテンツかどうかを調べる時間を節約できるし、ぱっと見て、何が出来るサイトなのかが分かりやすくなります。その説明は、2～3行にとどめておきましょう。説明が長くなればなるほど、ユーザーは読まなくなるからです。

## 2. 分かりやすさ

#### 6) 分かりやすい表現

分かりやすい表現とは、誰が読んでも分かるような表現を心がけるということです。文字で書かれたものは自分が思っている以上に意思が伝わらないものです。E-mailでも「そんなこと書いたつもりじゃないのに」と思った経験は誰にでもあると思います。そういうことを避けるためにも表現には気をつかうべきです。特に、専門用語や略語の使うときは注意が必要です。では、どうすればいいのでしょうか。具体的な表現をすることとを心がけることです。抽象的な表現は複数解釈を生んでしまい、それが分かりづらさにつながってしまいます。具体的なイメージが浮かべば、ずっと分かりやすくなります。

#### 7) リンク先の明示

他の Web サイトなどにリンクする場合には、リンク先のサイトにはどんなコンテンツがあり、何ができるのかを書いておくと親切です。他人の評価は当てにはならないけど、ちょっとした参考にはなります。

#### 8) 予想を裏切らない

人が目的を果たすために行動するときは、必ず予想をとまないとします。例えば、「この道をまっすぐ行ったら大通りに出るはずだ」といったように。しかし、その予想が外れると急に不安になってしまい、どうすればいいのかわからなくなってしまいます。

Web でもそれは同じことです。封筒の画像があれば、多く人はそれをクリックすればメールを送ることができると思うはず。しかし、いきなり違うページに切り替わったりしたら戸惑ってしまうでしょう。これぐらいならまだ許せますが、広告のためのウィンドウが開いたりしたら戸惑いを通り越して、不愉快な気持ちになりさえします。「工事中」のページも、原則は作るべきではないと思います。これこそ期待を裏切るページの典型です。これなら最初からそのページはいけないうようにして、準備中であることを書いておけばいいのです。

人が安心するのは、予想が現実になる時です。安心が信頼を生み、そしてそれがリピーターを増やすことになるのです。

### 9) ターゲットをはっきりさせる

まず覚えておかなければならないのは、万人にうける Web サイトは存在しないということです。そこでターゲットをはっきりさせる必要がでてきます。ターゲットをはっきりさせれば、そのターゲットに合わせたコンテンツ作りができるようになります。そのサイトに初めて訪れる人が 100000 人いるよりも、100 人の人が 100 回訪れるようなら、そのサイト作りは成功だと思います。

### 10) メニューの項目を増やしすぎない

メニューの増やしすぎにも注意が必要です。確かにメニューが多ければそれだけ内容が豊富だと思わせることはできますが、メニューの選択にも困ってしまいます。一度に表示されるメニューの項目は、人間工学的に見て、5 つぐらいが適当なのだそうです。どうしても多くせざるを得ない場合は、グループ分けをして、サブメニューを有効に使えばいいでしょう。メインメニューとサブメニューに分ける場合でも、どのメインメニューにどのサブメニューがあるのかが分かるようにしたほうがいいでしょう。

## 3. 見やすさ

### 11) 解像度に左右されないデザイン

Web サイト作りで犯しやすい過ちの 1 つに、自分のパソコンではちゃんと表示されるのに、他のパソコンで見るとおかしなところで改行されていたり、スクロールバーが出てしまったりということがあります。これは大きな画面を使っている人が陥りやすい罠です。もし不特定多数の人をターゲットにしたサイトを作るのなら、画面の大きさに注意を払いながら作業を行う必要があります。

特にノート型パソコンでは、640×480 の解像度をもつものも決して少なくはありません。できれば 640×480 でも問題なく見ることができればいいのですが、この解像度に合わせてサイト作りを行うと使えるスペースが極端に狭くなってしまいます。これを 1280×1024 の解像度をもつパソコンでみると却って空白が目立って見づらくなることさえあります。少なくとも 800×600 では同じように見えるようにすればいいでしょう。

### 12) 特定のブラウザに左右されないデザイン

ホームページに、「このページは Internet Explorer5.0 以上でご覧下さい」という文句が書かれていることがあります。個人的には、この文句をあまり好きになれません。自ら自分の能力の低さを認めてしまっているような気がしてなりません。

特定のブラウザでしか表示されない原因としては、JavaScript やスタイルシート、DHTMLを使っているということがあげられます。IE ではきちんと表示されても NN では表示されない場合が多くあります。これは IE では最新の技術を積極的にサポートしているけど、NN は必ずしもそうではないからです。

ブラウザにもたくさん種類があります。IE や NN のほかにもテキストだけを表示するブラウザもあります。これらすべてのブラウザでチェックする必要はないと思いますが、で

きるだけ多くのブラウザで、そしてできるだけ多くのバージョンでチェックするのが好ましいでしょう。

多くのブラウザで表示させるためには、なるべく HTML の文法にそって記述することです。文法に間違いがなくてもどのブラウザでも同じように表示されるとは限りませんが、少なくともブラウザがエラーを返すことはなくなります。

### 13) 統一感

書体、ポイント、配色など統一できるところはなるべく統一します。統一感があるだけで、安心感・信頼感が生まれます。これは、スタイルシートを使えば簡単にできます。特に配色には気をつける必要があります。配色による効果は絶大です。ホームページを見て真っ先に飛び込んでくるのは配色によるイメージです。さらにそのイメージはなかなか変えることはできません。例えば、青山学院の校章を思い出すことはできなくても、緑を使っているということは思い出すことができます。配色による効果は、私たちの心の奥底に訴えかける強烈な無言のメッセージなのです。

### 14) レイアウト

そのサイトをどんな性質にしたいかで、レイアウトは全く違ってきます。どのくらいの情報量があるのか、そしてその情報をどのように伝えたいのか、何のために Web サイトをつくるのかをもう一度見直してみましょう。どこに何を配置すれば、自分の思いを伝えることができるのかをよく考えて下さい。そうすれば、レイアウトは自然と決まってくる。

### 15) 全体のバランス

左右上下のバランスの悪いページはとても見づらいものです。不快感を与えることもあるかもしれません。かと言って、センタリングを乱用するべきではありません。センタリングをすれば、バランスを整えることができると考えるのは間違いです。

## 4. アクセシビリティ(「こころ Web」<http://www.kokoroweb.org/>を参考)

W3C は「どのような表示代行手段(例えば、デスクトップ=ブラウザ・音声ブラウザ・携帯電話・自動車用個人コンピュータなど)を使っても、またどのような環境(例えば、騒音の中・暗すぎたり明るすぎる部屋・手動操作が要らない状況など)を強いられていても、これに従えば、すべての利用者がウェブ内容を入力」(W3C,1999)できるような指針を「Web Content Accessibility Guidelines 1.0」で示しています。以下は、そのうちでもより重要だと思われる部分です。

### 16) 画像には alt 属性を必ずつける

alt 属性は画像の代わりに表示されるテキストのことです。alt 属性を指定しておけば、画像が表示されない場合でも、前もって指定しておいたテキストでその効果を補うことができます。また、音声ブラウザでは画像の代わりに alt 属性で指定したテキストが読み上



げられます。

#### 17) むやみに英語・専門用語・略語を乱用しない

例えば、「BBS」などの用語はインターネットに慣れ親しんでいる人ならば掲示板を指すことは常識になっています。しかし、そうでない人には全く分からないかもしれません。ある人には常識なことでも、ある人には非常識なこともあります。誰が見ても分かるように、英語や専門用語、略語の乱用は避けましょう。

#### 18) 各ページに適切なタイトルをつける

音声ブラウザでは、そのページの内容を判断するのにタイトル情報が使われています。そのページに合わせて適切なタイトルをつけましょう。

#### 19) 単語内にスペースを入れない

音声ブラウザではスペースを文の区切りと判断します。「政治」と書けば「せいじ」と読み上げてくれますが、「政 治」と書くと「まつりごと、おさむ」と読み上げてしまいます。単語を整えるためにスペースを使うのは避けたほうが無難です。

#### 20) フレームの使用は必要最低限に

フレームは画面インターフェースに大きく影響されます。携帯端末では正しく表示されないかもしれません。フレームはナビゲーションのための機能だということを忘れて下さい。広告のためだけに使うのは避けるべきです。どうしても必要だという場合以外は使わないようにしたほうがいいでしょう。

## 第 6 章

### 携帯電話端末

- |                               |         |
|-------------------------------|---------|
| 6-1 【携帯電話端末企画書】               | p. 138  |
| 6-2 【携帯電話端末】                  | p. 139  |
| 6-3 【自分達のこみゅぽ】                | p. 139～ |
| 6-4 【将来のこみゅぽ】                 | p. 148～ |
| 6-5 【青山祭で用意した Power Point 資料】 | p. 152～ |

## 第6章 携帯電話端末

＜＜文責：大西淳

### 6-1【携帯電話端末企画書】

携帯電話でこみゅぽを作成する前にしっかりとした指針を自分の中で決定すべく以下のような企画書を9月下旬作成しました。公になることはありませんでしたがその概要は以下の通りです。

使用者： 青山学院井田研究室  
使用期間： 2000/10/28～2000/11/1

我々青山学院大学井田ゼミナール研究会ではグループで情報を共有できるというコミュニティポータル「le tool de community」を想定し計画を進めてきました。その中のひとつの手段に今日その携帯電話としてではなく、モバイル端末としてその勢力を増してきた、NTTDoCoMo社のi-modeを来る10月28, 29, 30日の青山祭でコミュニティポータルのまさにツールの一環として利用しようと考えています。本来ならば、i-modeに限らずツーカー、au社のezWebやJ-Phone社などの他社機種も使用しグループウェアとしての制限を減らさなければいけないところですが、あいにくコンテンツ作成の点からも考えまして、i-modeのみの選択をしなければなりませんでした。

我々は、「誰でも、いつでも、どこでも」を指標にコミュニティポータル「le tool de community」を製作してまいりました。我々の認識では、ブラウザさえあれば、グループの情報を入手できるという構想があります。ブラウザさえあれば、世界中のどこからでも情報を入手できるのです。たとえ自分のパソコンでなくても、ブラウザさえ借りてしまえば、まるで自分の空間があるかのように、グループ内のメンバーの情報が共有できるのです。今後、パソコンは一家に何台もという時代がやってきて、HDDは大容量のものが作れるので、ホームステーションの設置により家庭のパソコンにおいて一台一台に搭載される必要性がなくなるかもしれません。もはやHDDは不要の時代になるかもしれません。そうした時を考えたときに、ブラウザのみの使用は大変に期待できるものであると思います。

ブラウザのみの使用ならば、同じ条件が普段我々の使用している携帯電話端末があるではないかという考えに至り、携帯電話業界でも真っ先にインターネットコンテンツを配信し、また使用人口の多いNTT DoCoMo社のi-modeを使用することになりました。コンテンツといいいましてもwwwに、比較してみますとさまざまな制限もございますので、まずは簡単な休講情報、周辺情報、青山のお店情報をHTMLで製作し、コミュニティのコンテンツに組み入れようかと考えています。将来的には私たちの最終目標である、コミュニティの二重構造、よりユーザからの視点で充実した機能・項目の完成を目指しています。

## 6-2【携帯電話端末】

＜＜文責：大西淳

私たちの当初からの目的であった、いつでも、どこでも、誰でもという概念の中に必ずしもブラウザだけという考えにとどまることはありませんでした。そうした中で夏前からの企画段階でPDAや各種 PDA、携帯電話の有力性というものも頭の中に入れておりました。しかし、プロジェクトを進める中でブラウザに特化した作業内容でも予想以上に作業が困難であるということ、様々な規制や問題点、技術的な未熟さの中で携帯電話端末やその他の機器に割り当てる時間がとても少なくなってしまうのです。それもそのはずで、実は携帯電話というハードに特別にこだわる必要がないということに気が付いたからなのです。私たちが本当に必要としているものはある特定のハードの仕様に基づいて作り上げるものではなく、夢のあとにハードが追随すべきであると思ったのです。

ただ今の日本では携帯電話は高校生を含む若い世代を中心にした必須コミュニケーションツールになっています。昔のようにビジネスで使う高価な代物という意識がなくなる一方で、生活面ではそれなくして生活できないまでにその需要は高まりました。こんな必須コミュニケーションツールこそが、実は次の世代に継承され生き残っていく唯一の情報機器なのかもしれません。そのような考えから、携帯電話端末は特別な視線から注目することにしました。

しかし！2000年12月にJava搭載の携帯電話が発売されるにあたって様々な人がJava開発キット（MIDP, etc）を手に入れ、少ない10k程度の容量の中で工夫が必要だ、コードはスマートに書かなければいけないなどと様々な規制があります。しかし、私たちが今回のプロジェクトを通じて感じたことはこれら技術と企画の段階の順序が逆転しているということ言うまでもありません。ある特定の者によって社会全体の利益が妨げられているということに気が付きました。その反面技術的な問題は、今ある事実を受け止め、その範疇で様々な工夫を凝らさなければいけないということも学びました。このようなことから私は、コミュニティポータルをインターネットの玄関口という広い意味で捉え、今の私の技術でできること（自分達のみみゅぽ）、また自分では実現できなかったもの（将来のみみゅぽ）の提案することに至ったのです。

## 6-3【自分達のみみゅぽ】

### (1) 自分達のみみゅぽを作成するにあたり

自分達のみみゅぽを作成するにあたり、すべての携帯電話会社のコンテンツ仕様に基きそれぞれに作成してこそ、万人を満足させる本来のみみゅぽの姿です。しかし、私たち

が本来目指すべきところは、今できる技術を利用して最大限にコミュニティを共有しようというのではなく、既存の技術の向こうにあるものを見出していこうというのが研究です。様々に考えた結果、仕様を一つに統一し、いつでも、どこでも、誰でもという枠からはみ出てしまうのを承知でNTT DoCoMo社のi-modeの仕様にに基づきコンテンツを作成することになりました。この決定が青山祭の約一ヶ月前、いままでの議論で必要だと思われたもの全てを加味しながら、私たちのできる限りのこみゅぽ携帯電話バージョンは作成されていたのです。

ちょうど作成開始から一週間ほど経った10月上旬頃にNTT DoCoMo社からMobile V-Campusが発表され、早速NTT DoCoMo社にその概要を教えてもらおうと、またあのような企業が何故ゼミというところに目を付けたかなど問い合わせしてみたのですが、結局返事は返って来ませんでした。構想はあっても、技術的な遅れで追い越されてしまう現実を目の当たりにしたのでした。しかし決定的に違ったのは私たちのこみゅぽが青山学院大学の学生として仲間と共有できる情報、その上位レベルに井田セミナーがあり、ゼミ生だけで共有したいもの、青学生として提供したい情報という風に、何重もの階層の中でコミュニティを形成し情報を共有できるのに対し、Mobile V-Campusがゼミと言う一重の枠に過ぎなかった点でした。後にNTT DoCoMo社の技術的な面で何重もの階層を作ることが不可能になってしまいましたが、考えは私たちの方が勝っていたことは言うまでもありません。

## (2) 学生が必要と感じた機能

パソコンでは起動に時間がかかる、重たい等の理由で見るとまでは至らない、しかし必要と感じる情報を多くの学生から意見を聞きました。その中で下位レベル（みんなで共有できる情報）としてあったら便利だと思われる項目機能を学生の視点から実際にコンテンツの中に実装することを前提に取り上げました。

### [株式情報]

できるだけ最新の株式情報を提供できる項目。携帯電話からのインターネット証券も学生の間で最近で人気だけに、株価を注目する人にとっては必要な機能です。

### [ライブ情報]

現在どこで、誰の、いくらでどのくらいの規模でライブコンサートが開かれているのか等、詳細な情報を簡単に入手できます。

### \*[映画館情報]

自分の属すコミュニティに関係する地域の各映画館の上映情報。

#### [教室変更]

学校に設置してある情報を端末から受け取れる教室変更の情報を自分の携帯電話で確認できる。この項目は実際に教室変更される頻度が少ない上、個人情報を引き出せないので、実現しませんでした。

#### [授業評価]

教授から一方的に評価を得るのではなく、学生からも評価を受けると言う発想からできた項目。学生は年頭の履修科目の選定材料として利用できるのと、広くいろいろな人から授業に対する感想や評価を得ることができます。

#### [教室番号]

コミュニティに入る際に様々な入力をする事で情報を共有できる。例えば、一年間の授業時間割を入力しておけば、コミュニティ内の人間がどこの教室でどの授業を受けているのか簡単に把握できます。井田セミナーのゼミ生時間割は実際に Java applet で参照できるようにになっています。

(<http://www2.sipeb.aoyama.ac.jp/~mi-home/semi4/semi/timetable/schedule.html>)

#### \*[終電時刻]

以外に調べるのが面倒な終電時刻。だったら最初から基本的な情報として組み込んでしまおうというのが狙いです。

#### \*[明日の天気]

天気はとても大変重要な要素ですが、これもまた欲しい時に無い情報の一種でいざ情報を提供しているサイトを見に行くと有料だったりします。こんな時はコミュニティの人に頼って、知った人が情報を提供できるようにしておけばいいのです。

#### [学食人気ランキング]

お昼時になって、何を食べよう。そんな時にこの機能が役に立ちます。この学食の人気メニューを食券販売機と連動させることで、生の情報を簡単に入手できます。今日は何が売れているのか、食堂へ行く前に確認できます。

#### [行事]

年間行事というのは以外に忘れがち。コミュニティで共有したい記念日、スケジュール、メンバーの誕生日などを登録しておけば、その日になる前に事前に携帯電話にお知らせしてくれます。

#### \*[災害時の避難場所]

災害という非常な事態の時には何が真実で、なにが嘘のでまかせ情報なのか区別をする

のが非常に困難です。そのような時の事を考えて、事前に正確な情報を登録しておき災害時に慌てることなく、避難場所や緊急時の連絡先を手元の携帯電話にお知らせしてくれる情報項目。

#### [校内トイレ検索システム]

校内にはたくさんのトイレがあります。やはり利用するところはきれいなところでというのが大事な要素なのかもしれません。今自分がいるところからもっとも近いところで清潔なところで、かつ近いところを簡単に検索してくれるのです。男性にはあまり重要な機能ではありませんが、女性にとってはとても大事な機能なのかもしれません。

#### \*[銀杏情報]

青山学院大学の青山キャンパス特有の機能ですが、季節感を唯一感じるものかもしれません。

#### \*[現在発売新製品]

時代に最先端に生きる者、業界のニーズを的確に把握する機能としては必要かもしれません。その日に発売されるCD、食品、飲料、家電、などジャンルは様々。今最先端のものは何か、学生固有の機能にとどまらないかもしれません。

#### [学生運動情報]

青山学院大学では、学生運動はあまり盛んではありませんが、大学によっては学生運動の過激さ故に起こる問題は様々です。バリケードを張って校内に入れないということもあるかもしれません。そんな情報を提供するのがこの機能です。

#### [チャペル説教予定]

礼拝の予定は簡単に把握できないことがあります。いつ行われているのか、どんな説教を予定しているのか。そんな時は携帯電話で簡単に知ることができるのです。

#### [施設情報]

学校にはいろいろな施設を利用することができます。図書館、マルチメディア教室、トレーニングルーム、厚木キャンパスの駐車場。それらの利用状況や講義で使われない時間を取り出すことができます。

#### \*[周辺情報]

コミュニティがメインとする地域を対象に、喫茶・飲食、銀行、本屋などの場所を網羅する機能。ありそうであまりない機能です。

#### [OB, OG 検索]

学校外では基本的に OB, OG 検索は現在できません。認証の問題さえクリアすればこのような情報も入手できるようになります。

#### [レポート一覧]

情報端末では個人のレポート課題は参照できますが、携帯電話ではまだこのサービスを利用することができません。やむを得ず学校にいけない場合でも家などから調べてレポートに着手できます。

このような機能が一般の学生からのニーズの多くでした。以外に身近なところの簡単な情報を携帯電話端末で入手したい意向が結果から知りうることができました。\*は実際に今回のプロジェクトで作成した機能です。

(3) 井田セミナーの場合 (<http://133.2.110.17/commupo/i/>)

#### [コンセプト]

では実際に私たち井田セミナーで共有したい情報はどのようにすべきか、運用すべきかという時にどの機能を盛り込むかという問題にあたります。6月から活動してきた中で本当に必要なものは何なのか、どんなものがあればいいのか、必要とする機能は洗練されなければいけません。しかし、実際に i-mode で実現でき、機能的に満足できる機能はごく僅かだということに気がつきました。本当に欲しい機能は実はまだ現段階では学生にとっては難しいということを確認させられました。登山家が山の地図を見て、この山を登るにはまだ完全な道具がないと足踏みしている現状を打破すべく、現段階でできること、それをまず実現してみようというところから始まりました。

まず私たちの理念である、青山学院大学の学生としての一面、井田ゼミとしての一面。青学生として共有しても良い情報、井田ゼミだけに共有したい情報を管理する場合。この構造を多重化するときにやはり認証の問題が発生します。しかし、web コンテンツと比較した場合に、i-mode で実現できる認証は限られてきます。ワンタイムパスワード、セッション ID を Form で投げる形で利用すれば認証はかけられます。しかし、時間的制約で残念ながら私の力では実現できませんでした。このことから井田ゼミとして上位レベルで情報を共有した場合を想定し、また青山祭で初めて見た人でも分かるようなコンテンツ作成をすることになりました。機能は一般の学生でも必要と思われたものを順次盛り込んでいきました。

#### [仕様]

電子掲示板やチャットを作成する際に .cgi を利用しました。プロジェクト発表の約1週間前に i-mode でも servlet 技術を利用できることを知りました。しかし、この時点での



方向転換は無理と判断し、現行のまま cgi を利用しました。そのほかのページにつきましては HTML 2.0, 3.2, 4.0 のサブセットとして定義された CompactHTML (C-HTML) を、NTT DoCoMo 社が一部独自に拡張させた i-mode 専用のページ記述言語 (i モード対応 HTML) を使用しました。

#### [ディレクトリ構造]

フォルダー管理ではないので、実際にはディレクトリではなく階層もなくフラットですが見やすくするために以下のような記述をしております。

<<全 68 ファイル、合計サイズ 1.51MB>>

index.html

·overview.htm

·prof.htm

·井田教授公式 HP

·itdc.htm

·area.htm

·movie.htm

·a.htm

·ka.htm

·sa.htm

·ta.htm

·na.htm

·ha.htm

·ma.htm

·ya.htm

·ra.htm

·wa.htm

·diagram.htm

·bank.htm

·asahi·b.htm

·kousanshinyou·b.htm

·kokumin·b.htm

·sakura·b.htm

·sanwa·b.htm

·jyouanashinnyou·b.htm

·sumitomo·b.htm

·daiichikangyou·b.htm

·daiwa·b.htm

·chuoshintaku·b.htm

-toukai-b.htm  
-toukyousouwa-b.htm  
-toukyoutomin-b.htm  
-toukyoumitubishi-b.htm  
-tuyoushintaku-b.htm  
-doueishinyou-b.htm  
-nihonkougyou-b.htm  
-higashinihon-b.htm  
-fuji-b.htm  
-hokuriku-b.htm  
-yachiyo-b.htm  
-yokohama-b.htm  
-café.htm  
-book.htm  
-q-ko.htm  
-monday.htm  
-tuesday.htm  
-wednesday.htm  
-thursday.htm  
-friday.htm  
-saturday.htm  
-ginnan.htm  
-ibbs.cgi  
-i-sho.cgi  
-nph-minichat.cgi  
-iLycos 検索エンジン (i-mode 用)  
-イミダス  
-英和辞典 (英辞朗)  
-newcd.htm  
-weather.htm  
-urgent.htm  
-tel.htm  
-member.htm  
-cafeкуро.htm  
-macci75.htm  
-moriken.htm  
-arrow.htm  
-aqua.htm

·keiko.htm  
·sussy.htm  
·ritz.htm  
·junko.htm  
·jon.htm

#### [実際のインターフェース]

図 6-1 は P502i による index.html 画面の実際のインターフェースです。撮影は学校地下食堂で撮影しました。青山キャンパスの地下食堂は電波が入らないので表示は圏外になっています。また、画面表示を確認する段階で何度も同じ画面を呼び出すので、画面メモという i-mode 独自のキャッシュを利用しています。そのため画面下方に「メモ」と表示されますが、実際はもっと多くの情報が表示されます。i-mode では全機種対応の HTML を書く場合、画面の小ささから横文字全角 8 文字まで、画像はモノクロ 2 階調の GIF 形式に限られます。また、転送速度との関係でファイルの容量は画像を含めて 2KB 未満を推奨しています。左に表示されているこみゅぽの TOP 画面は GIF 画像を含め 683 バイトになっています。

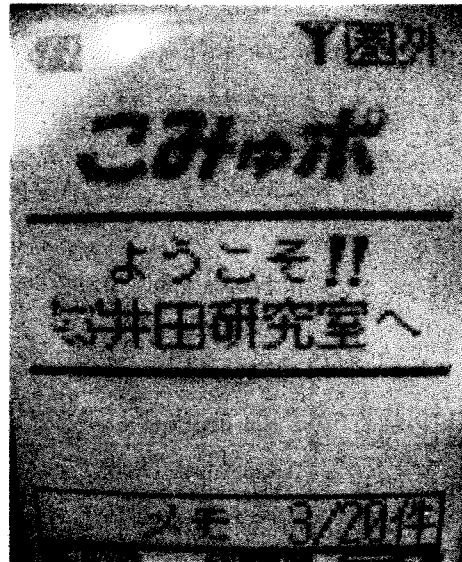


図 6-1

「井田研究室へ」の左横にある猫の絵文字は NTT DoCoMo 社が i-mode 専用を用意した 166 個ある絵文字から”&#”に 10 進コードを加えて表示させたものです。絵文字は i-mode 端末で表示させて初めて見ることができるので確認が非常に大変でした。

#### [苦勞したところ]

約一ヶ月前という時に作業が開始しました。それまで、Java の本を片手にコーディングであったり、ソースを引っ張ってくるために英語サイトをくまなく探したりして、毎日が地獄にいるような日々でした。プロジェクトまでの期間アルバイト先の休み時間等を利用してソースを探すことも多々ありました。しかし、この時になってそれから解放されるような気持ちになって少し気が楽になったというのも少しありました。しかし、実際には地道な作業で、web ページと異なりほとんどの装飾ができない分、ページ量を多くしなければ見栄えしないということを気づきました。そのためにデザインや創造性といった面に気を使うことができず、ただ時間だけが経っていったような気がしました。

しばらくしてから i-mode 用のタグ辞典を購入しました。やはり i-mode 専用のタグを使わなければならないというのもありましたし、そこに認証のヒントが隠されているのではと購入に至りました。渋谷のブックファーストに清水君と下調査に行きましたが、結局認証に関する書籍は見つけることが出来ませんでしたので、とりあえずタグ辞典だけを購入しました。

Compact-HTML といえども今まで知らなかったような機能がたくさんあり、この頃になってようやく作成のコツをつかんで、本当に携帯電話のコンテンツを作成しているという実感が湧いてきたのでした。

i-mode では画面の小ささから 8 文字という横幅の文字制限というものがあります。Web ページとは異なり、ローカルで出来上がり確認ができないことと、画面の実際の表示は実機で見なければわからないので、仕上げたファイルをアップロードし、少しでも間違いや構成上おかしい箇所はその都度修正しなければなりません。これを業界では妥当性チェックと呼ぶそうです。

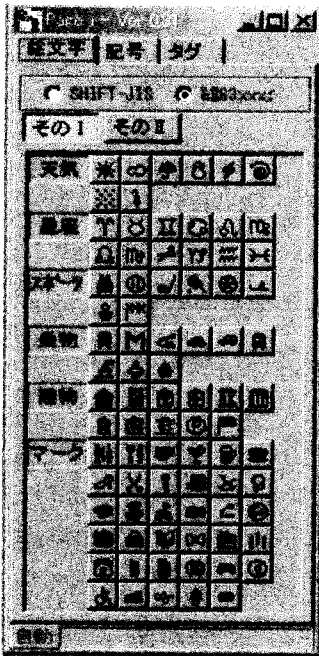


図 6-2

作成はテキストエディタにて、平打ちでも十分に可能な量でしたが、絵文字のコードを把握できませんでしたので、C-HTML 支援という左のソフト(図 6-2 参照)を使用しました。各ボタンを押すことで、簡単にタグや絵文字をエディタに挿入することができます。このソフトの使用によって、いままで困難だった絵文字のコード番号の照合が大変楽になりました。しかし、以前インターフェースの確認は、実機に頼るのみでした。実機で実際に画面の調整をしなくてもパソコンの画面上で確認できるようなシミュレータはありましたが、やはり精度的に難しいものがありましたので目で確認が一番ということを再認識しました。

今回のプロジェクトでは情報が大西淳という一人の人間によって提供されてしまったので、いままでの i-mode コンテンツとなんら変わりなかったのですが、本来情報はコミュニティの中で投げ交わされ、一人の管理者によって制御されるものではありません。ですので、一人で情報を収集するという点でも苦労しました。また、天気予報、映画館情報、休講情報は頻繁に更新されるものですが、一人でできると思っていた更新作業は学業との両立で頻繁にできないことが分かりました。この苦労が実は、ユーザが常に新しい情報を求めているという意味で、解決されればビジネスという形で成り立つことなのです。IT 革命、情報時代に「スピードは命」とはまさにこのことなのでしょう。

上述した機能の中で、「今日発売の新製品」という機能は実は、自分が昔のアルバイト先で働いている時に思いついたものなのです。コンビニエンスストアでは毎週新製品が約 10 品目ずつ増えます。これほどまでに入れ替わりの激しい理由は消費者が、常に最新の物でしか満足しないからなのです。毎週新しい製品が陳列し、頭に残らない製品も多くあります。その中で、勝ち残ったものだけが、スペースを確保され今も売れ続けているのです。当時 50 万円ほどの資金で、コンビニエンスの新製品に特化するのではなく、自動車、アパレル、ありとあらゆる製品を取り上げ、地元の友人と二人でこのようなサイトを実際に立ち上げようとも考えました。この話は消費者に限った話ですが、無料で情報を提供するという差異を除けば、携帯電話一般ユーザも同じなのです。常に新鮮な情報でなければい

けない、古い情報はまったく価値がないということ、こんなことを情報の世界に入る前に小さなことで発見したのです。このような意味で「今日発売の新製品」が項目として加えられています。

#### [青山祭]

青山祭は10月28, 29, 30日の3日間に渡り行われました。連日の天候の悪さにもかかわらず多くの方々に私たちのプロジェクトが見てもらえることができました。携帯電話のブースは外ブースというものを設け、ベンチや植物をレイアウトすることで実際に外の環境を再現してみました。しかし、221教室が不運なことにより携帯電話の電波状況が悪くなかったため、当日来ていただいたみなさまが実際に手に触れてコンテンツを見るときには非常に不便なものになってしまいました。

学生に多く見てもらおうと教室を出て、一般の学生に見てもらった結果大変良い反応を受けました。是非続けて欲しいとの要望もありましたが、青山祭期間中のみしか更新できなかった結果を見ると、存続は厳しいように思います。

教室内でも青学生に限らず、社会人、主婦、他学生、これから青山学院を受験しようと考えている人などいろいろな人に見てもらえる機会がありました。また、他学生からは関東全域の大学で統一して今回のような情報を発信してみても、というような意見もいただきました。また、逆に i-mode と謳ってしまったからでしょうか、まったく興味を示さない人も多くいました。しかし、忘れてはいけません、私たち井田ゼミの研究が i-mode のコンテンツ作りということではないことを。次の項目では、携帯電話端末が（携帯電話端末とは呼ばない方がよいのかもしれませんが）これからのコミュニティにおいてどのような役割を果たしていくのか、どのような機能があれば良いのかということを検証していきます。いわば、いままでの話は過去の話なのです。私たちの夢はここにあります。

### 6-4 【将来のこみゅぽ】

将来のこみゅぽはどうなってしまうのだろう、機能はどんなものが付加できるのだろう。と技術のめざましい進歩により迷ってしまうのではないのでしょうか。

携帯電話端末に限っては言えばここ数年で飛躍的に技術の進歩がありました。それは消費者の需要が今まで以上に高くなったから、若年層に爆発的な支持を受けたからではないのでしょうか。その意味で将来一般ユーザはパーソナルコンピュータよりも携帯電話端末を操作が簡単で、手軽に持ち運びできる端末として、社会が自然と他の機器を淘汰していくのではないのでしょうか。また、急成長を遂げてきた携帯電話はとどまるところをしりません。急成長の背景にメインユーザが若者という常に新鮮なモノを要求する人間層があるからなのです。携帯電話技術はこれから、Java, IMT-2000, W-CDMA というような技術を展開していきます。これらの技術によっていままで考えられなかったことが、現実起こってしまうのです。また通信速度の飛躍的向上もすぐに実現します。これによっていままでやりとりされていたファイルの種類は映像や動画など全く異なったものになるでしょう。

この通信速度の向上が私たちコミュニティを形成する上で一番重要な要因だと思います。現在の携帯電話の画面を見る限りでは通信速度の向上はあまり大きな問題ではありません。しかし、これからの携帯電話として考えるのではなく、携帯端末として考えた時に通信されるデータは大容量にものが多くなります。レポートを転送したり、講義90分の授業の様子を転送したり。私は将来携帯電話という言葉はなくなると思います。新しい機器が別の名称で多目的に使用できる端末が開発されると思うからです。

コミュニティを形成する上で通信の段階で障壁があってはスムーズなやりとりができないというのも高速通信の重要性と考える一因です。例えば日常の会話で、自分の発する声が途切れ途切れであったり、声がなかなか相手に届かないということがあったらどうでしょう。これからは単に「声」がデジタルな形で相手に伝わるという違いでコミュニケーションを考える時になんら違いは無いように思います。

上述したようにパーソナルコンピュータよりも携帯端末がこれからの情報時代を担っていくものだと述べましたが、最先端の時代になって誰しもが携帯端末を持つようになれば、その機器はありとあらゆる情報の発信源となり、情報を発信する場合には個人を特定する合鍵となり、それなくしては情報を発信できないという「コミュニティ時代のキー」になるのではと思います。フィンランドではすでにSIMカードによって個人を特定し様々な決済を可能にしていますが、日本もそうなる日は遠くないと思います。

今のインターネットを考える時、情報は一つの機関や人物によって発信されていますが、今後はインターネットに参加する者（コミュニティ内に人間）が情報の発信者にならなければなりません。今はテープで言えばA面のみ使用にとどまっているのです。決して悪いことではないのですが、B面もあるのならば使ったほうがより豊かになることは言うまでもありません。以下に記すものは6月からのプロジェクトを通じて必要だと思った、現在では実現に至らない機能です。

#### [必要と思った機能]

私たちが、コミュニティの構想の中であつたら便利だと思った機能、近い未来に実現を願う機能の提案です。

#### 1. GPLi(Global Present Location indicator)

この機能は現在で確立しつつあるPHSでの位置情報サービスと似ています。しかし決定的に大きな違いは双方向での位置確認ができることです。携帯電話は常に基地局の電波でつながった状態になっていて常時接続に近いのです。そんな特徴を利用して、電車の駅、高速道路など公共の場所に通信用ではなく、小さな基地局を設置し、より具体的な場所を特定することができれば、目的の人が何時にどこに到着するかも簡単に把握することができます。また、このような基本情報を簡単に入手できるようになれば、公共の場所での場所の確認を通話することなくできるのです。全ての人の場所を把握できては困るので、コミュニティの中の特定の相手に相互の許可を得た上で実現できるようにすれば良いので

す。つまり A 君と B さんが互いに許可すれば、その二人はいつでも相手の居場所が確認できるのです。また、許可する度合いを変更し、居場所を抽象化させることでプライバシーも確保できると思います。

また常に、オンラインの状態の携帯電話なので相手がサービス圏外にいる、会社にいる、など相手の状況をこちらで判断し通話を控えることもできます。

グループで利用する場合はさらに便利になると思います。どこで、集まればいいのかと急に決断しなければならない時に複数人の中心を割り出し、この GPLi が簡単に答えを出してくれることと思います。

次世代携帯電話が確立する頃には、全世界を自由に飛びまわれる乗り物が開発され、友人が簡単に世界の裏側にいるようなこともあるかもしれません。そんな意味で相手の位置確認は非常に重要な機能になると思います。青山祭の時点では Pli でしたが、こんなことで GPLi に進化しました。

## 2. DVM(Digital Voice Mail)

知り合いの間でもあまり、評判の良くなかった機能なのですが。高速通信の実現によりまた、圧縮技術の向上により生の声をメールとして送信することができるのです。生の声を送信するのならば、直接電話すればいいではないかという声がありましたが、急用で今すぐ伝えなければならないのに、相手が電話に出ないという場合はどうでしょう。また、現在ある機能で留守番電話があるではないかというご指摘も受けました。しかし、メールを何回も聞きたい場合、留守番電話センターに通じない所ではこの DVM が強力に効果を発揮するのです。また、発信側での代替案としてメールをテキストで打てばいいのではという案がありますが、もしこの人がメールを携帯電話のボタンで書くのが苦手だったらどうなるでしょう。また、声の抑揚によって文字では表現できないことも生の声では表現できると思います。やはり、このような場合を考えて自分の声を直接携帯電話に吹き込むことで送信できる機能はすばらしいと思います。また、それに伴って音声で吹き込んだデータを正確にテキストに書き換える技術もあればいいのではという意見もいただきました。これがあれば、井田ゼミ内での発言や意見を書く書記も必要なくなります。

## 3. SMAT(Synchronized Multiple Access Technology)

いままでのキャッチホンや 3 者通話と似ている部分があるのですが、これは複数の人間が同時にネットワーク上で場所を選ばずに会議に参加できるものです。固定電話は無料になって、携帯電話の通話料も格段に安くなり、通話できる時間はいままでとは比較にならほど長くできるようになると思います。そんな特性を利用して、複数間での通話ができるのです。おしゃべりと違って、本来会議というものは一人が発言する場合であって同時に何人もの人たちが覆いかぶさって話すものではないと思います。ですから、この機能は十分に可能だと思います。ただ、実現して欲しい機能は現在誰が会議に参加しているのか、論題は何なのか、などその場にいなければ把握できない事柄を表示してくれる機能を具備することです。

#### 4. ASD(Automatic Signal Display)

この機能はコミュニティとはかけ離れているのですが、携帯電話に必要な機能を考える時にこのように関係のないものの方が思い浮かんでしまうので、最後まで実用的な段階まで考えた ASD を最後に紹介しようと思います。もしかするとこれから後になってコミュニティに必要な機能かもしれません。

この機能は現在の GPS を利用して行われているサービスの延長だと思います。でも何故開発されていないのか不思議に思ったので、自ら提案してみたいと思います。携帯電話にカーナビを接続してサービスを受けるのは最近ではごく普通になりました。しかし、カーナビは道路を教えてくれるものです。なぜ標識の表示はないのでしょうか。一つ一つの看板の費用を考えれば小さなチップを道路に埋め込んで、そこを車両が通るたびに窓ガラスの上方に薄く表示させた方が効果的ではないのでしょうか。車が道路に埋め込まれたビーコンによって自動で運転できる技術があります。そのビーコンと一緒に標識も埋め込んでしまえば、簡単に ASD が実現できるのではと思いました。看板がなくなることで、景観も良くなります、一つ一つの制作費、修理費が浮くのでとても経済的です。運転が自動化されれば、標識は関係なくなるからなのではないでしょうか。

携帯電話はパーソナルコンピュータと同じく無限の可能性を秘めています。またそれらを利用する人たちも無限の使い方を知っているのです。だからこそ、未来をより発展したものにするためには情報の自由な発信の場としてコミュニティが必要なのです。そのツールとして、ほんの小さなファクターとして携帯電話を取り上げました。これから、ハードはどうなっていくか想像もつきません。これから先はやはり、ハードの時代に依存する時代ではなくアイデアや独創性を持った知的所有者が重要になってくると思います。そんな意味でコミュニティは知的所有者の宝庫となりうることで私は大いに期待しています。

～コミュニティは宝箱なのです～



## 6-5【青山祭で用意した Power Point 資料】

### 携帯電話はこうでなきゃ！

～ゼミ生からの提案～

2000年度青山学院大学国際政治経済学部井田ゼミナール

2000年10月

資料 6-1

### コミュニティの形成に大きく関わる携帯電話

- ◆ JAVA, IMT-2000, W-CDMAなどの出現でいままで考えられなかった夢のような機能が実現してしまう？！
- ◆ 高速通信の実現により。
  - 迅速で早いデータ転送。
  - やり取りされるデータの量。
  - 音声のみに限らず.....
- ◆ コミュニティに必要なもの、、、それは仲間どうしの無償の情報共有。つまり、情報バリアフリーこそが次世代のコミュニティを築いていく！！
- ◆ 携帯電話もはや通話を可能にするものではない！！
  - コミュニティという層の「キー」であり、
  - コミュニティを実現する上で最も重要な道具となり、
  - コミュニティにあまり失敗を許さずすべてが携帯電話によって操作される。

資料 6-2

## コミュニティってなんですか？



- ◆ 仲間同士が集まり、互いに便利に効率的に情報を共有できる場
  - 販の知れてる仲間、そうでない機会等々、
  - 家族、友達、サークル、ビジネス、その種類はさまざま
  - 共有できる情報はやまほどあるのに、なんでしないの？
- ◆ かつての「アゴラ」を現世で繰らそうではないか！
  - 広場に集まって好き勝手に言って、情報を飛び交わせ！
  - 昔ながらの習慣をなぜここで止めようか！
  - 既成な情報の絶え間ない流れをグループでリアルタイムに共有しよう！
  - 皆に利益になる情報は惜しまず提供しよう。
- ◆ コミュニティに参加しよう！
  - 携帯電話、wwwブラウザ、PDA、一人一つの何か .ac を使用して

資料 6-3

## インターネットを考える

- ◆ 情報提供者による一方的な情報配信ではなく、自らの手で作り上げる、自らの情報交換による、自らに良いサイトという発想
- ◆ 衆多の意見交換による、より洗練された情報、意見交換、共有
- ◆ 自分たちの意見が、内容となり、真の本質となり、サイトとなり、かざらない情報の発信源がそこにある
- ◆ Webの向こうの大海、情報の海原、情報こそが力となり、そこに存在する人間の集まり、つまりコミュニティが結ぶ無数の可能性
- ◆ その一手段として、私たちの目指すところのコミュニティポータルが存在し、「こみゆ水」が私たちの掲げる将来のインターネット、



資料 6-4

## 携帯電話、3つの提案

- 私たちが、コミュニティの構想のなかであつたら便利だなあと  
思つた機能、近い未来に実現可能を願う機能の提案
  - **PLI(Present Location Indicator)**
    - コミュニティ内の人達がお互いの居場所を確認できるよ
  - **IVM(Original Voice Mail)**
    - 笑顔をメールで送ってしまおう!
  - **SMART(Synchronized Multiple Access Technology)**
    - 遠くの場所で携帯電話を使ってミーティングをしまおう



資料 6-5

## PLI (Present Location Indicator)

- 探したい相手の現在位置する携帯電話の最寄の基地局を把握し、逐次基地局位置情報を、許可した相手(グループ)のみに画面上にリアルタイムに通知する
  - 駅や公共施設に基地局経由装置を設置すれば、どの電鉄のどこの駅の近くにいるか把握できて、目的地もあらかじめ知っていれば、到着予定時刻もグループで簡単に把握できる!
  - 公共の場での無駄な携帯電話の使用を極力控えることができる!
  - どこで来ればいいのか、PLIが簡単に答えを出してくれるでしょう!
  - これで誰が遅刻して、誰がオンタイムで、何時ごろに話し合いが始められるかも簡単に解決!グループの疲まりでは、もう確は適用しない?
- かけたい時に電話をかけられない、そんな煩わしさはもうない!
  - PLIによって常に情報をストリーミングしてるので、今現在その人が圏外なのか、またまた地下にいた時だったのか、無駄な電話をかける必要がないのです!
- 同じ基地局エリアに互いに許可した人達が入ると着信で知らせてくれる
- 誰が同じエリアにいるのか、画面上に一覧表示され簡単に居場所を把握できる、
- 迷子になったおじいちゃんも簡単に検索できる!

資料 6-6

## DVM (Digital Voice Mail)

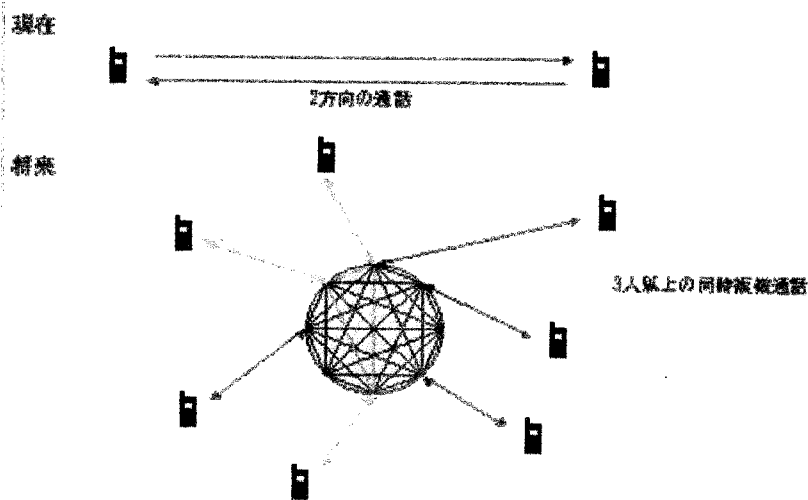
- いままでのテキストベースはボタンで打つのは面倒！！
- せっかく時間かけて書いてもこれだけ？！うそー。
- 電子メールは相手の感情がわからない・・・顔文字で我慢しよう。
- 携帯の文字読みにくい、、、眼鏡かけないと。
- メール打ってる時間ない！！

こんな症状ありませんか？！こんな時は

- いままで困難とされた音声ファイルの転送が高度な圧縮技術と飛躍的な通信速度の向上により、テキストベースとなら変わる方法でメールを送受信できます。
- 電子メール同様に自分の都合のいいときにDVMを確認することができます。
- 携帯電話に直接入力した音声そのまま相手に一発送信！
- 煩わしいボタン操作も一切必要ない！携帯電話からボタンが潰れるのもそう遅くない！！

資料 6-7

## SMAT (Synchronized Multiple Access Technology)



資料 6-8

## **SMT (Synchronized Multiple Access Technology) ?**

- ◆ 現在の携帯電話はもちろん人間での通話を可能にします。キヤッチホン機能もありますが、基本的には2方向の通話に限られます。
- ◆ 30回線は最大5人までの同時通話を可能にする。会議やミーティング場外を選ばずに会議ができたらなんてすばらしいんでしょう？！かけた相手も会議をしていない場合、呼び出し音は「会議に参加しますか？」会議に参加して時点で、通話は始まる。会議での発言者は必ず一人、論理上はできる.....
- ◆ ある人気教授の授業を聞きたい、、、でも風邪で聞けない.....あ！こんなときは30回線！友達も教授の声もぼっち通話！
- ◆ 緊急の会議発生！でもあれはOOさんとOOさんしかわからない！こんな時

資料 6-9

## **携帯電話のなぜ**

- ◆ 我々井田ゼミナールでは6月から当プロジェクトに10人で関わってきました。その中でこれがあつたらなんて便利なんだろう、あればいいのに、10人という人間の少数のグループでもまとまりがない。今回のプロジェクトで切に思うこのようなシステムの実現 コミュニティの楽しさ。
- ◆ きっと技術的には可能なものばかり、でもなんでないの！という意見を皆様にも知ってもらいたい
- ◆ 世の中でコンテンツがあふれているのに、なんで新しいタイプのシステムがないのか？！同じ物の生産の繰り返し、世の中はもう飽きている！
- ◆ そんな状態で、ブラウザ・携帯電話を始め、自分たちでできる部分は実際に作成し、便利でツカエル機能をコミュニティポータルとして作成しようというのが最大の狙い

資料 6-10

# 第7章

## 青山祭

7-1 【青山祭出展】	p. 158～
7-2 【各ブースの説明】	p. 159～
7-3 【221 教室ネットワーク】	p. 162～
7-4 【当日アンケート・実施と結果】	p. 165～
7-5 【パネル】	p. 176～
7-6 【ポスター】	p. 181

# 第7章 青山祭

## 7-1【青山祭出展】

〈文責：矢島達也〉

私たちは、*le tour de France* から “*le tool de community*” と題して展示を行い、プロジェクトの成果を一般の学生や、社会人の方々にプレゼンテーションや実際に触れていただくことで見ていただきました。

### ○展示の目的

私たちは今回のプロジェクトを進めていく中で、一般の学生が、私たちの提供する学生情報システムに対してどのような反応を示すのか調べることの必要性を感じました。この学生情報システムには、私たちゼミ生が「あったらいいな」と思う機能を盛り込んでいます。しかし、ゼミ生の欲しいものがそのまま学生の欲しいものであるかということと必ずしもそうではありません。学生の率直な意見が私たちのプロジェクトにプラスになると考えました。

そこで私たちは、たくさんの学生が集まる青山祭において、私たちのプロジェクトを紹介し、学生情報システム「Commupo! (こみゅぽ)」を実際に使ってもらって、その意見を聞くことにしました。

### ○展示期間

2000年10月28日(土)～30日(月)

(30日は午後1時で終了)

### ○展示場所

青山学院大学

青山キャンパス 2号館 221教室

### ○当日のレイアウト

〈レイアウトのコンセプト〉

“Anytime, Anywhere, with Any device”をコンセプトとしてブースを設定しました。一つの教室に様々な環境を配置するのは大変でしたが、ブースを細かく分けることで Anywhere を実現しました。また Any device においては WebTouch, 携帯電話端末、パソコンを用意し環境を整えました。

(その他概要、目的は各ブースの説明のところを参照して下さい。)

<レイアウト図>

実際のレイアウトは以下の図 7-1 の通りです。

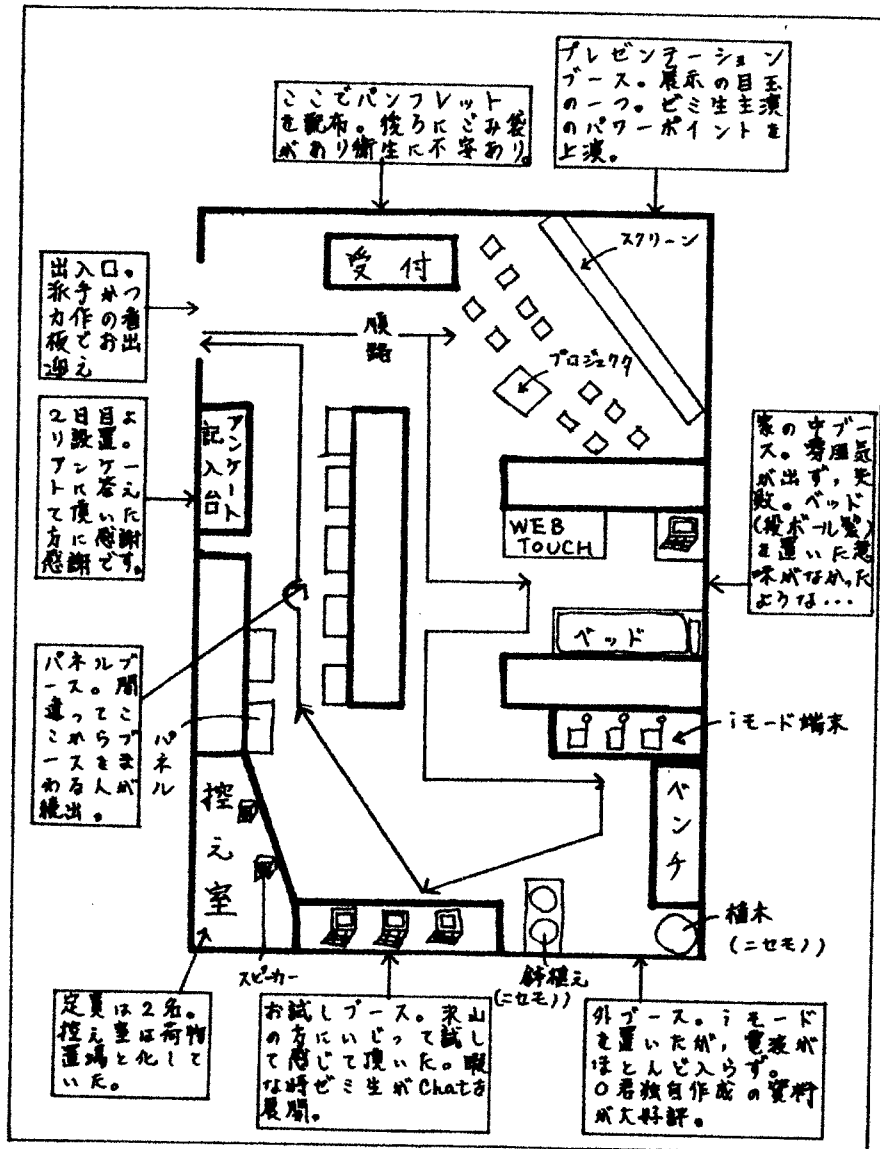


図 7-1

7-2 【各ブースの説明】

1. プレゼンテーションブース

<ブースの概要・目的>

私たちが今プロジェクトで考えたこと、理念等の紹介をするために、プレゼンテーションを1時間に1回、約10分間行ないました。このプレゼンテーションを見た上で各ブースを回って頂いた方が、理解が早いと思い、入り口から一番近い場所にこのブースを設置しました。



#### <期間中の反響>

エンターテインメント性を持たせるために、そして聞いている人にわかりやすくするために、ゼミ生の1人(cafekuro)にモデルとなってもらい、「こみゅぽ」がある学生とない学生を(写真と声にて)おもしろおかしく演じてもらいました。これが大好評で、場内に笑いが起き、その場が和んでいました。観客の皆さんの反応もよく、雰囲気は最高でした。プレゼンテーションのあとの質疑・応答タイムでは、社会人の方々から鋭い質問・意見がたくさん出ました。今後のプロジェクトに大いに参考になるものばかりでした。総じて満足をして頂けたようです。

## 2. 家の中ブース

#### <ブースの概要・目的>

私たちの提供する学生情報システムが家庭においてどのような役割を果たすのかを模索するブースです。ベッドや机をおいて部屋の中を再現し、その環境の中で実際に使って頂くことで、近未来での、自分の使用の様子を想像してもらうことを目的として設置しました。

また、「Web Touch One」という、電話とインターネットが一体化した機器を置きました。これは、インターネットの機能をタッチパネル式にして簡単にしたもので、近未来の家庭でのインターネットが簡単になるであろうとすることを示すために設置しました。まだ日本語版が出ていないので、ここでは英語のホームページを表示しました。

#### <期間中の反響>

家の中という雰囲気が出なかったせいか、来てくださった方々が注目したのは、「Web Touch One」の方でした。その操作の簡便性が注目されたのではないのでしょうか。「これはいつ発売されるのか？」と聞いてくる方もおり、期待の高さが窺えました。

このブースで印象に残っている人の意見として、「作る側にとっては独創性を出していくことができるが、使う方としては、独創性がどんどん失われてしまう」というものがありました。使う側としてこのようなことを考えたことがなかったので、この意見は自分にとってとても新鮮なものでした。

## 3. 外ブース

#### <ブースの概要・目的>

通学途中の時や授業の合間の時に「こみゅぽ」の休講情報や、周辺情報などを確認するという環境を実感してもらうためにこのブースを設置しました。観葉植物やベンチを置いて外の雰囲気を出すようにしました。i-mode用のサイトを作り、それを公開して実際にアクセスして頂いて実際に使って頂きました。

また、ゼミ生独自の、携帯電話の将来像に関する資料もそのブースに置いてアカ

デミックな面も作り出しました。

#### <当日の反響>

携帯電話でのインターネットを使っている人が多いためか、多くの方が関心を示していました。社会人の方々は、jon の作った資料を熱心に見ていました。携帯でのインターネットに対して大きな期待を持っている人が非常に多いことが分かりました。

教室に電波がほとんど入らなかったため、教室内でサイトをきちんと見せることができる機会がそんなに多くなかったのですが、総じて反響はよいものでした。

### 4. お試しブース

#### <ブースの概要・目的>

たくさんの方々にサイトを実際に使っていただくために、このブースではコンピュータを複数台(3台)設置しました。ゼミ生の説明のもと、たくさんの方に使って頂きました。

#### <当日の反響>

大多数の人は感心していました。「これ公開しないの?」と実際に使いたいという人もいました。そして、不平不満を出す方はほとんどいなかったように思います。私たちのサイトが認められたということはいずれにしてもうれしいのですが、もっと不満を出して欲しかったという気持ちもあります。

### 5. パネルブース

#### <ブースの概要・目的>

私たちがこの学生情報システム「こみゅポ」を作るために使った技術、サーバ、ネットワークなど、裏側を説明するためのブースです。このブースには、①当日のネットワーク図、②サーバの説明、③認証がどのように行なわれているか、④Servletの説明、⑤私たちの考える「こみゅポ」の将来像、そして⑥学生に対して行なったアンケートの結果をパネルにして貼り付けました。

#### <当日の反響>

使った技術に興味がある人となない人でこのブースにかかる時間が全然違いました。Java に興味があったり、サーバに興味がある人は長く時間をかけ、ゼミ生にも積極的に質問をしていました。一方、コンピュータに興味のない人は、敬遠して素通り、ということが結構ありました。何も分からない人でも分かるように書くことの難しさをこの時に感じました。

## 7-3 【221 教室ネットワーク】

＜＜文責：大西淳

青山祭でのネットワークを担当することになってしまいました。そもそも、NATという小さな機械を先生から借りたので、誰がその管理をするかという大変安易なことで、担当が私になってしまったのです。ただ、NAT といってもお値段は4~5万円で、小さな「箱」だったので、大丈夫とたかをくくっていました。受け取った時はただの箱としか思えなくて、設定が必要ということは先生の方から聞いていましたが、設定するボタンも無かったので、たいした作業ではないと自分で勝手に思い込んでしまっていたのです。その思い込みがあとあとになって大変なことになってしまったのです。着実に青山祭までの作業も進んで、日にちもどんどん迫って来て、他の授業の課題もたくさんあるのに、いよいよ青山祭まで一週間という時に先生から NAT の設定は終わったのかと聞かれました。設定のほとんどがサーバの方で行われるものだと信じて、設定も実験もなにも行っていませんでした。その時に初めて NAT の設定がいかに重要かということを知らされ、大慌てで作業に取り掛かりました。普段は使わない Libretto ff 1050 を学校に持って行って作業は開始されました。設定はやはり本番となるべく同じ環境に近いように、国際政治経済学部のサーバの置かれている 8 号館 4 階の部屋で進められました。ところ狭しと、サーバ機が何台も保管されていて、ここに国際政経の中核があるのだと実感しました。また同時に、国際政経のサーバの保管されている部屋がこんなにも小さなところだったということもその時同時に感じました。とにかく、私たちのサーバから出ている LAN ケーブルを無数の配線から探して、早速作業に取り掛かりました。もっと早くから始めていればこんなに焦らなくても良かったのにと、本当に後悔しました。しかし、ここまでの作業を振り返って見たときにもっと早くとはどんなことなのだろうと自分に質問します。果たして本当に自分に時間があつたのか、もっと有効に活用できたのか。青山祭までの期間が残り 3 日くらいになったこの時期でしょうか、時間の大切さ、使い方を改めて考えた時は。そんなことを考えながら、説明書とのにらめっこが始まりました。Linux でやるべきか、Windows でやるべきか四苦八苦していると、先生が手伝いに来てくれました。先生にこれまでの経緯を話すと、「このままではだめ。やらなければいけないことを書き出して整理しないと成にもできない。」とおっしゃいました。すべてを書き出して、やらなければいけないことを整理しました。家に帰ってから、説明書をくまなく読み、何がいけなかったのかということも考え直しました。また、いままで曖昧だったネットワークの知識を完全にすべく、ゼミの前期で使用した「TCP/IP ネットワーキング」という本を読み返しました。これらの作業で今まで見えなかった問題点がたくさん出てきました。足りなかった機材もすべて用意し、木曜日の授業（企業論）も投げ捨てて、ネットワークを完成させなければ！という使命感のもとに作業を開始しました。昨日とは違って、知識もついて、段階を踏んで作業を記録していたので、若干の苦労はありましたが、思いの他設定は早く終わってしまいました。いざ終わってみるとあっけなく、昨日はまでの苦労はどこへ行ってしまったのだろうという感じでした。そしてネ

けなく、昨日はまでの苦労はどこへ行ってしまったのだろうという感じでした。そしてネットワークに関しては、当日を待つばかりとなりました。

下準備は終わっていたので、残す作業は 2 号館 221 教室への物理的接続と、教室内の配線になりました。NAT の設定に比べれば何をすればいいかも理解していたし、本当の配線だったのでむしろ楽しかったです。221 教室に最も近く LAN が来ているのは隣の 11 号館のビルでした。そこで特別に許可をいただき 11 号館の 3 階に英語講義準備室があるので、そこから私たちのネットワークを引っ張らせていただくことになりました。しかし！ビルが違うのと階も違うので LAN の長さは大変なことになってしまいます。仕方がないのでビルの渡しは空中で、階の差はスロープの勾配を使うことになりました。青山祭当日に 100m あるオレンジ色の LAN を用意して、丁寧にビルのスロープに沿って配線しました。ちょうど高さが合った良いところで一旦配線を終了し、221 教室に戻りモリケンに投げる旨を言いました。しかし、戸惑ったように私の話を理解していませんでした。投げれば、LAN の束をばらばらにしてしまうのではと思ったのでセロハンをスロープ→221 してからその先を手繰って引く方法を取りました。思いの外簡単でした。教室内の配線もとても簡単でした、途中 WebTouch の位置変更で HUB の移動はありましたが、予想以上に LAN ケーブルが長かったのか、教室が小さかったのか分かりませんが、複数本予想していたところを、一本でカバーできたりであつという間に終わってしまいました。問題はここからだったのです、本当に昨日の NAT の設定が消えずに残っているか、実際に 221 教室で作動するのか。大変不安でした。東芝さんのパソコンを開封して、早速ネットワークに接続します。全てを設定し、いざネットワークに接続してみます。“ネットワークに届きません” の表示。これほどまでに端的に私の心を傷つけた言葉はありません。この言葉に全ての含意があったからです。呆れた私は「そうですか。」と投げやりたい気持ちになりました。しかし、そんなことで諦めてはいけません。とにかく NAT のローカル IP である 192.168.0.1 へ telnet しアクセスしてみます。しかし、それすらもできないのできつと NAT のローカル IP 自体が数日の間に消えてしまったのではと思いました。telnet 画面では確かに接続中、、、になっているのと、NAT までのアクセス時間は長かったので実際に IP が合っているかあっていないかというのは 1 分くらい待たないといけないので、とてもいらいらします。そうこうしている今度は“ホストまでの経路がありません” と表示されたりします。NAT の設定の時に何百回と見た表示だったので、あの時の苦しみを思い出してしまいました。原因が究明されないので、さすがに英語講義準備室を疑いました。あとになって疑って申し訳なかったことに、NAT から出ている一発目の macci75 所有の不良 HUB が、全てをそこでせき止めていたことが分かったのです。こんなこともノードの方で LINK を確認していれば解決できたことでした、、、なんで気づかなかつたのだろう、、、あんなこんなで万事解決され、proxy サーバ設定でインターネットも見られるようになって、インターネットカフェでもできるくらいの環境が整ったのでした。

青山祭当日の朝になりました、ネットワークも完成していて(図 7-2)何もすることがなかったの、とりあえず音響のネットワーク?!を作りました、去年は BOSE のスピーカーを使うなど贅沢な音響環境でしたが、今年はレイアウトに使用してしまった莫大な予算と、ゼミ内にマリー=アントワネットがいたので、予算を大幅に縮小され、私の私物を持つてくることになってしまったのです。とくに嫌だったのが、電車での持ち運びでした。スピーカーと本体を別々の日に 2 回に分けて運びました。スピーカーを前日に本体を当日にというような感じで。合計で 2 回嫌な目に遭いました。しかもよりによって一番家の遠い私になんで!!!? 両日とも朝早い集合だったので、ただでさえギュウギュウの通勤電車で、きっと他の人は私のかばんの固さに迷惑したことでしょう。学校にはアンテナのプラグがなかったので、whip 式のアンテナもすぐに作ってラジオを聞くという苦勞もありましたが、かねてからの音響設置の希望も実って大変満足でした。

[2号館 221 教室]

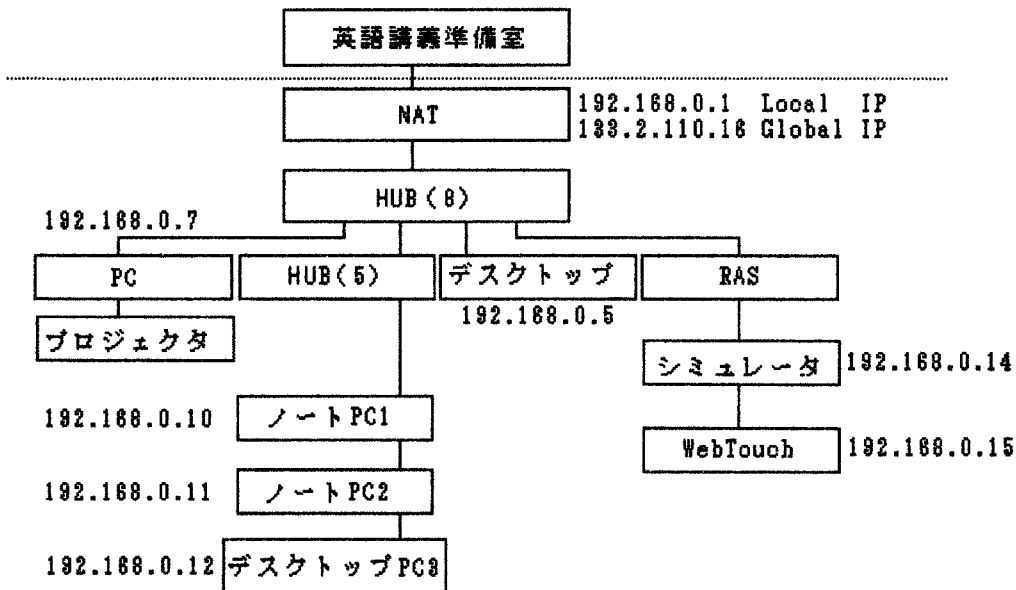


図 7-2

## 7-4 【当日アンケート・実施と結果】

＜文責：山崎淳子

実施日 - 2000年10月28日(土)～30日(月)

実施場所 - 青山学院大学青山キャンパス2号館221教室

回答 - 47

有効回答数 - 47

00/10/28～30

本日は井田ゼミ4期生プロジェクト「こみゆポ」の展示にご来場ありがとうございます。お客様からのご意見、ご感想を反映させて、さらによりよい「こみゆポ」を作りたいので、最後に簡単なアンケートにご協力下さい。お願い致します。

※ 該当する番号を○で囲んでください。

1. あなたの職業をお聞かせください。

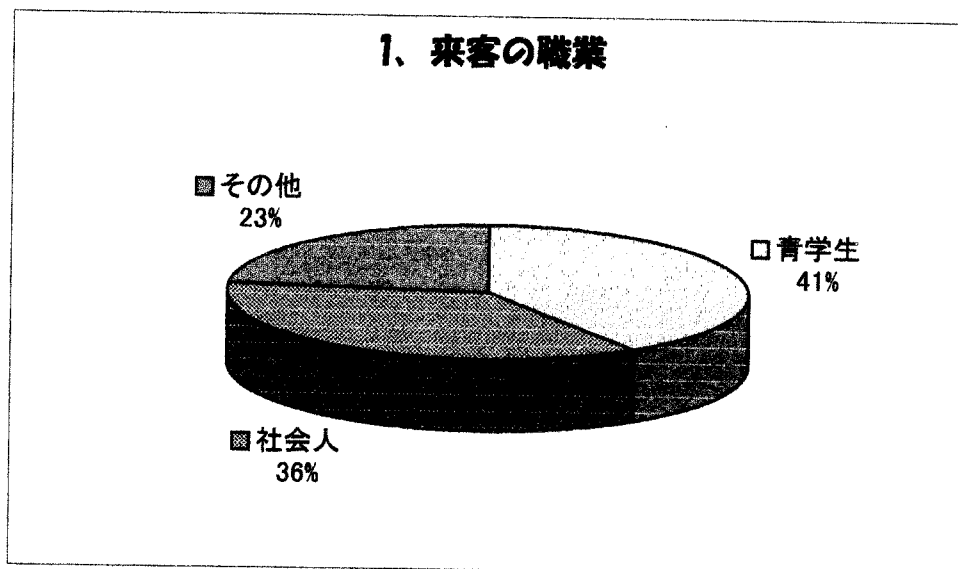


図 7-3

上記図 7-3 の通り、土・日・月と週末あわせて 3 日間行われた青山祭の展示では、青学生だけでなく、企業からのご来訪や保護者の方々など、さまざまなお客様を迎えることができました。

2. あなたの性別をお聞かせください。

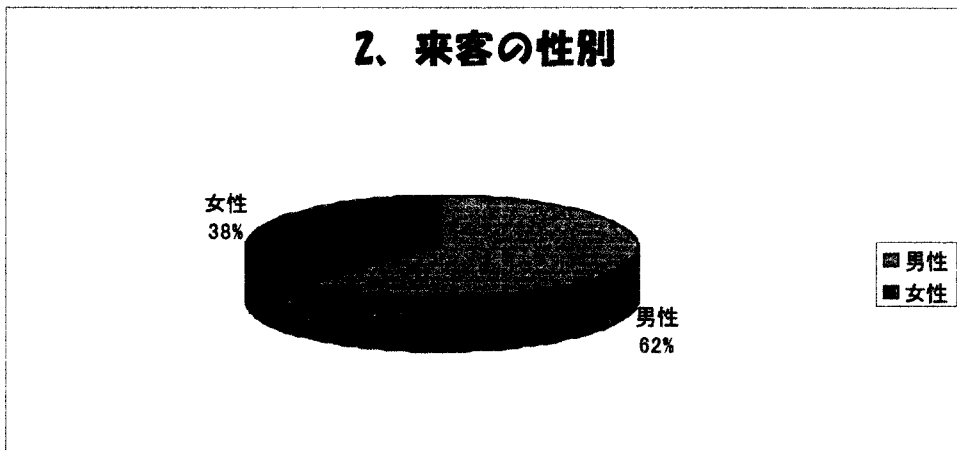


図 7-4

コンピュータやインターネットに興味を持つ方々は、やはりどちらかというとなら男性の方が多いのでしょうか…？

3. 「こみゅぽ」があると便利だな！又は、今後使ってみたい！と思いましたか？

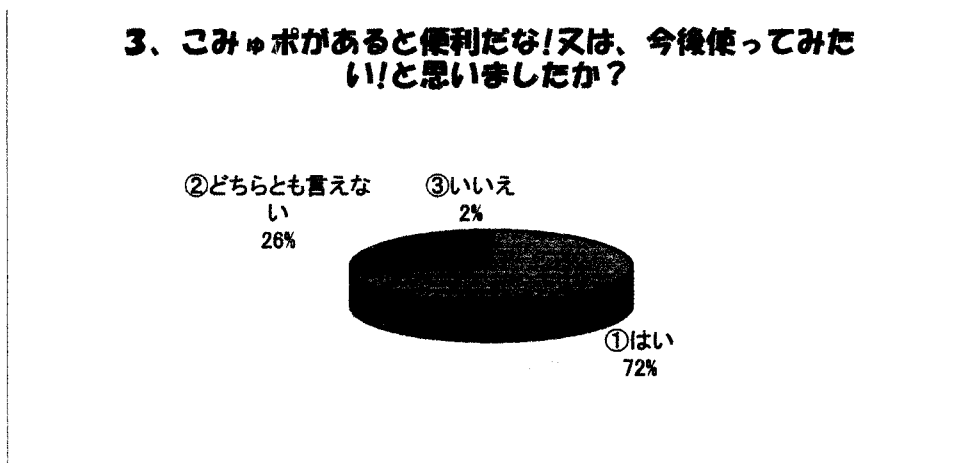


図 7-5

図 7-5 のように、70%以上のお客様に、「今後使ってみたい」という回答をいただきました。この「こみゅぽ」をさらに改良、拡張し、将来他の学校や会社内、サークル内などで使えるようになるといいです。

4. 「こみゅぽ」の中で、面白かったツールはありますか？

#### 4. こみゅボの中で、面白かったツールはありますか？

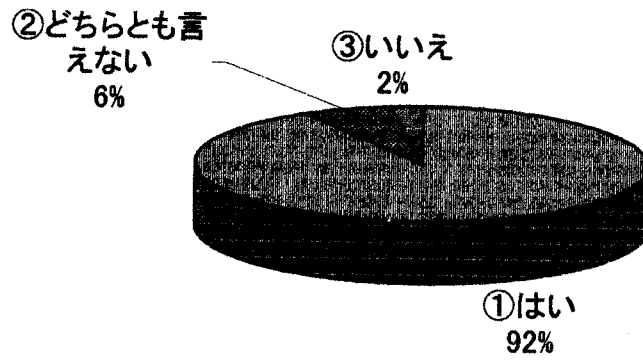


図 7-6

\* 「はい」と答えた方にご質問します。

面白かったツールは以下の何でしょうか？（複数回答可）

#### \*面白かったツールは何でしょうか？

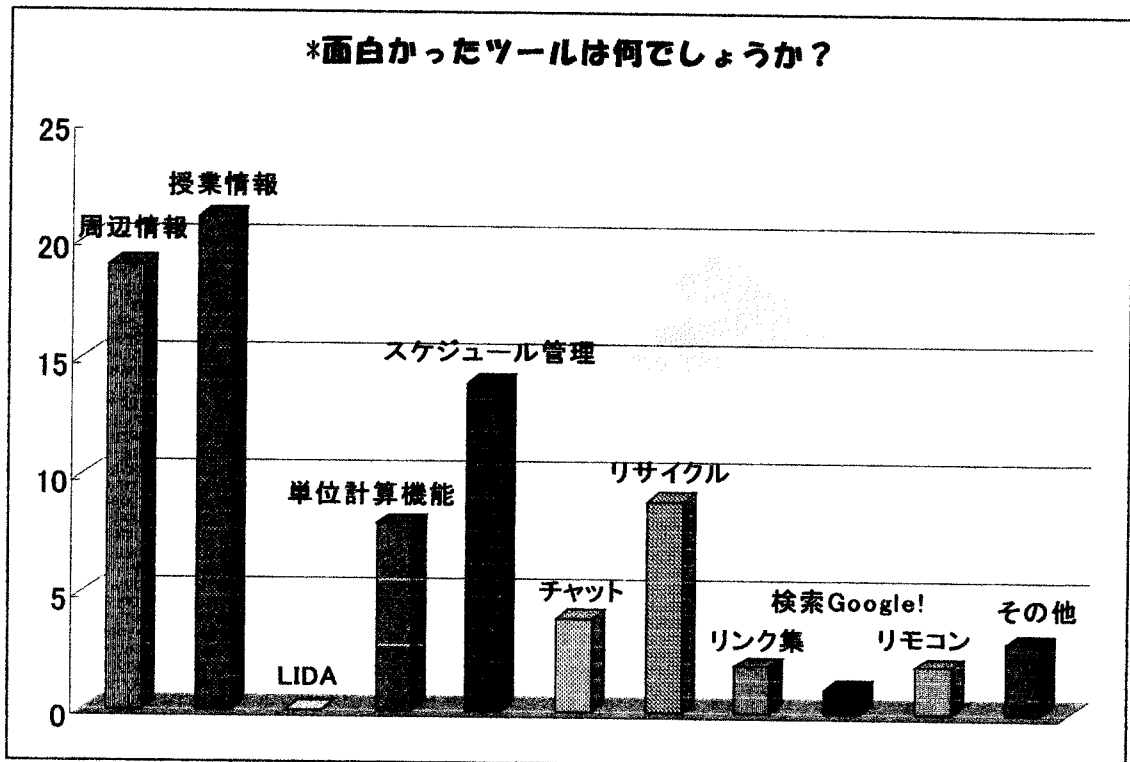


図 7-7

- 授業情報・周辺情報… やはり青山キャンパスに通う青学生には必須！



- 授業情報・周辺情報… やはり青山キャンパスに通う青学生には必須！  
今日の授業も知りたいけれど、休講なら遊べる場所も知りたいよねっ？
- スケジュール管理… プライベートなスケジュールを手帳がわりに書きとめちゃおう！  
でも、それと一緒に学校の行事予定とか、ゼミ・サークルのスケジュールなんかも自動的にお知らせ&管理してくれたらすてき。
- リサイクル… 重たくて高い教科書を、先輩から譲り受けられたらなあ…？  
お互い、このページの中で古教科書や実験器具などをリサイクルしあえたらいいよね。
- 単位計算機能… これは死活問題か！？卒業年度の予想がつく…。
- その他… ごく少数だけど、電話会議ができる機能の評価を受けました。コレが電話と違うところは、同時に3人以上で使用できる点。キャッチホンでは出来ない優れワザ。

5. 「こみゅぽ」を使うことによって、コミュニティ内でのつながりが深められると思いますか？

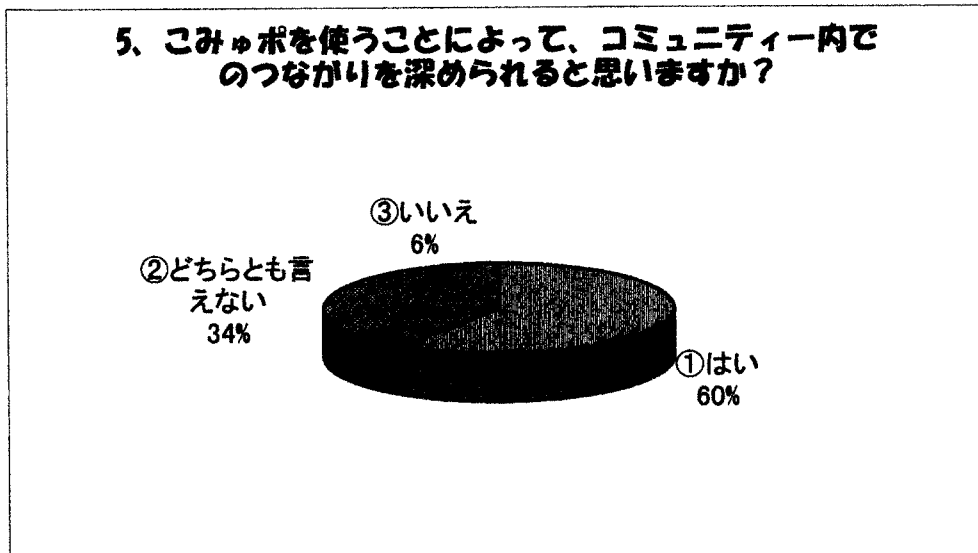


図 7-8

「つながり」と言うとき少し抽象的で良く分からないかもしれませんが、「こみゅぽ」によって、コミュニティ内の連絡などが円滑になり、より身近に感じられることは確かだと思います。(図 7-8 より)

6. 実際にあなたの会社、友達、サークル間などで「こみゅポ」を使ってみたいと思いますか？

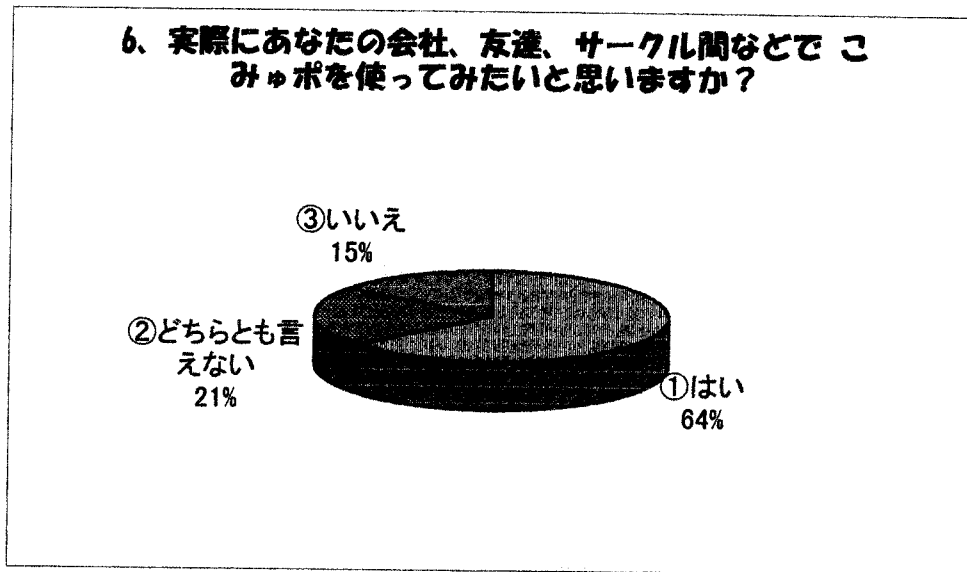


図 7-9

7. 実際にあなたご自身が「こみゅポ」を使用するとしたら、ブラウザ画面とiモード端末用では、どちらがいいですか？

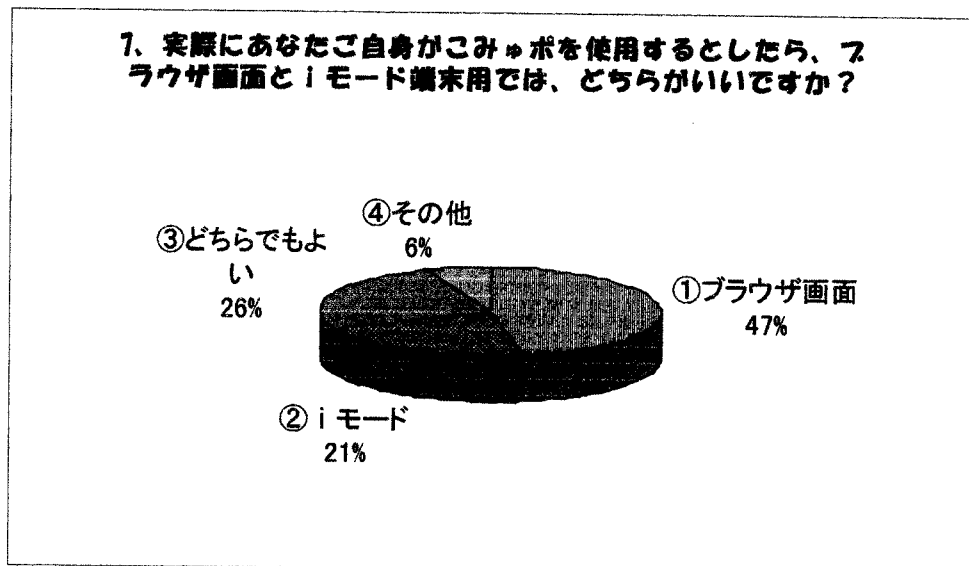


図 7-10

お客様の中には、J-PHONE を使用されている方も多く、そのお客様にとっては、「ブラウザか iモードか？」という質問自体が受け入れられなかったようだ。この質問項目は、初めからドコモ利用者を前提とした質問だったので、そこに問題がありました。(図 7-10) 実際、ドコモ利用者は携帯電話利用者の中で最も多くを占めているが、学生などの若年層には比較的 J-PHONE 利用者が多い。また、同じドコモ利用者であっても、主婦の方や、

それぞれのライフスタイルによって、全くiモードを必要とされていない利用者も多かった。このような青祭当日の現状から、この質問項目は、「ブラウザ画面か携帯端末画面か？」という質問に変更するべきだったと思う。

8. あなたは今までに、eグループ、intranets、dosule!など、他の情報共有ポータルサイトを利用したことがありますか？

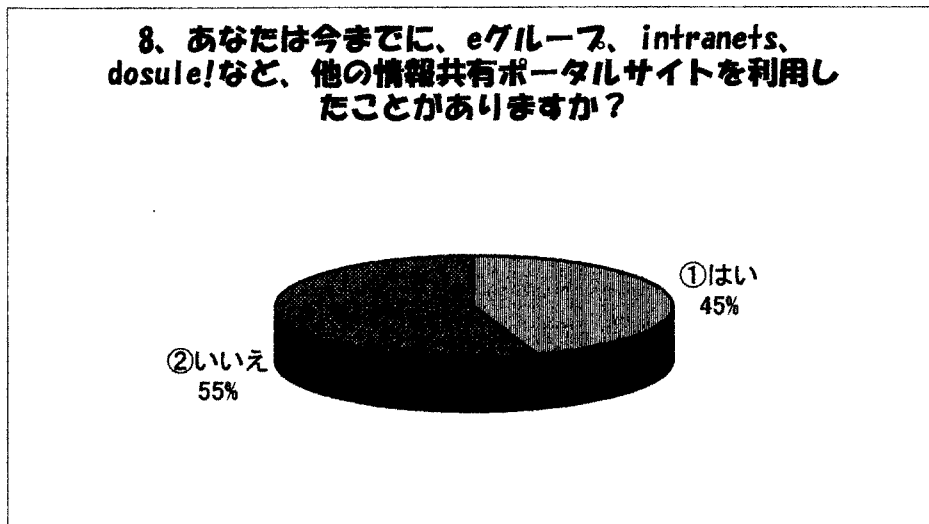


図 7-11

IT 業界からご来訪されたお客様のなかには、さまざまポータルサイトの利用を経験されている方がいました。しかし、一般の学生や、その他主婦の方などは、ポータルサイトという言葉もはじめてという方が多く、利用状況はそれほど多くはないようです。

\* 「はい」と答えた方にご質問します。

他の情報共有ポータルサイトと比べて、「こみゅポ」はいかがでしたか？

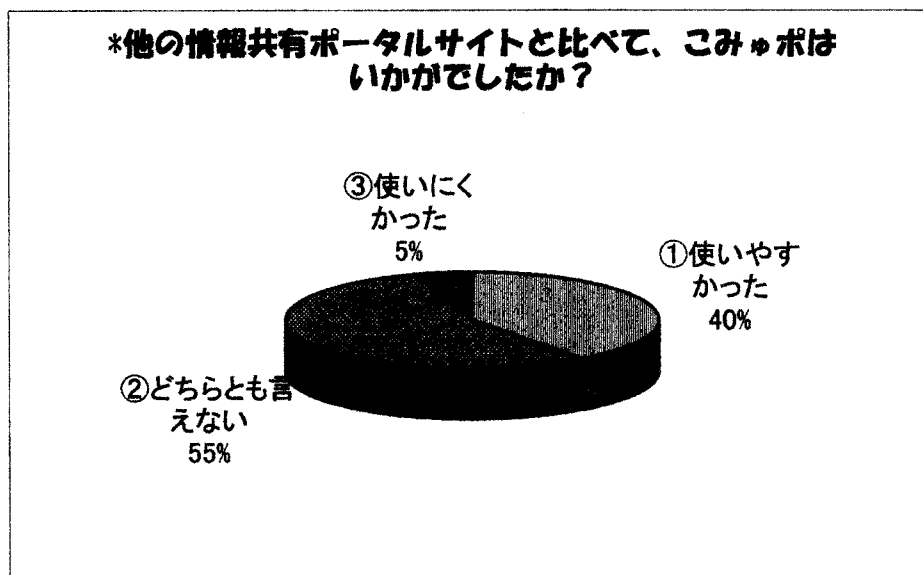


図 7-12

ポータルサイトを利用したことがある方からのご意見としては、励ましの言葉や、学生向けに良いというお褒めの言葉を頂きました。それと同時に、基本的なナビゲーションをもう少し充実させるように示唆を頂いたり、インターフェイスがかわいいなどのコメントもいただきました。(図 7-12)

9. 今回「こみゅポ」を使ってみて、あなたの関心のある情報にうまくたどり着くことができましたか？

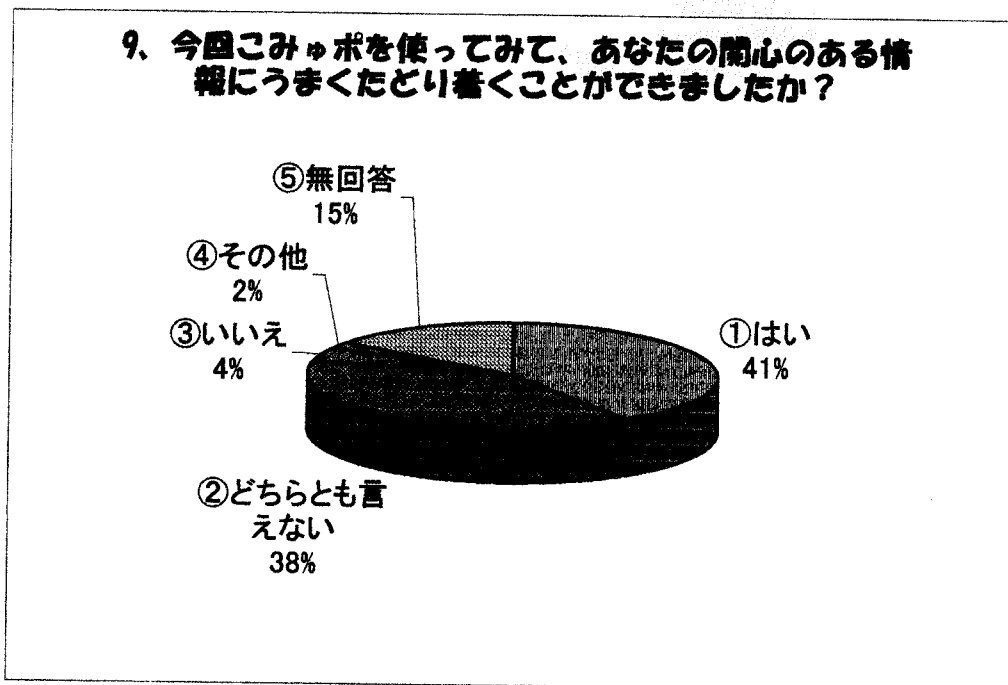


図 7-13

10. 「こみゅポ」画面は見やすかったですでしょうか？

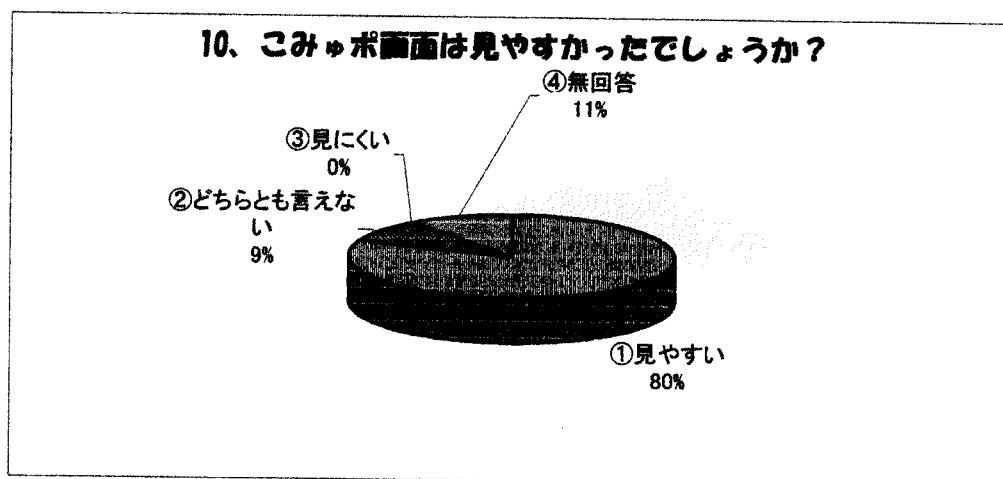


図 7-14

※但し、奥に入っていくと分かりにくいというご意見も頂きました。

11. 「こみゅぽ」画面の使い方は簡単に分かりましたか？

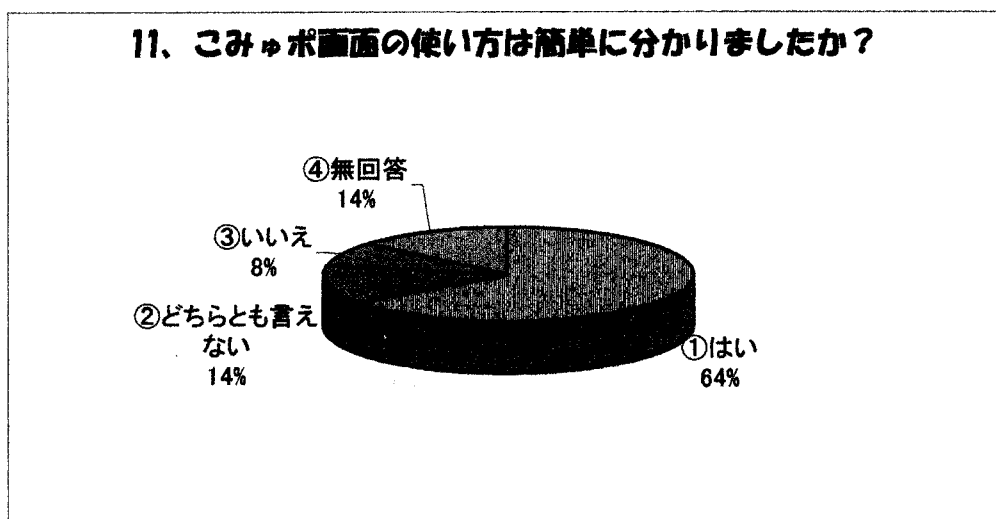


図 7-15

※ なかでも多く頂いた意見は、クリックすると別画面に出てくるリモコンを、メイン画面の横・右上につけたらどうだ？という意見でした。確かに、そうですね。マウスは右手で動かしている人が多いので、右上にリモコンを置くのが一番使いやすいと思います。

12. 「こみゅぽ」画面の中での、さまざまなリンクやダウンロードはスムーズにできましたか？

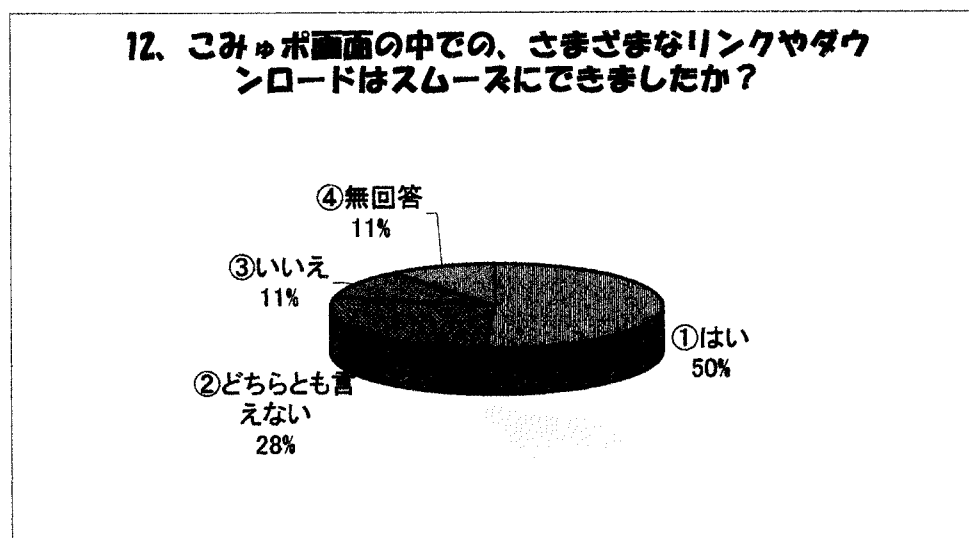


図 7-16

13. 今後、「こんな機能／ツールが欲しい！」と思うものがありましたらお書き下さい

- ・ 過去問題集などがあればいいのでは？
- ・ 自分が関係するコミュニティーに情報が入力されると、その情報がメール配信されるようにしては？スケジュール情報、休講情報など。
- ・ 購買 Sales 情報が自分の興味分やにあわせて届く仕組み
- ・ 簡易機能の必要性
- ・ 特になし
- ・ PDA と Web 関連系ツール
- ・ Java、win32、SyneML を使ったツール
- ・ 機能で使い方についての説明があると使い易くなると思います
- ・ 休講情報をメールで配信
- ・ 利用者が自由にグループを設定して使えるように開放されるとよい
- ・ 自分の単位取得状況を知りたい
- ・ 青学周辺の遊べるところ
- ・ 周辺情報で、居酒屋などについて、電話番号、道順を示して欲しい
- ・ 教室変更とかもわかって今日はどこへ行けばいいのか分かっていいか分かっていいと思います
- ・ 携帯へのメールが簡単にいせるとか、i-mode でワンクリックで他の人のスケジュールに予約できるとかもおもしろいかも
- ・ 図書館とのつながりは結構重要だと思います。
- ・ 白馬ヒュッテとのつながりも重要だと思います
- ・ 今のところ別にありません
- ・ 店の紹介部分で、簡単な略地図などがあるといいかも。(全体図に番号とかふっておくのもいいと思います)

14. その他「こみゅポ」の展示についてご意見・ご感想・改善点などありましたらお書き下さい。

- ・ 頑張ってください
- ・ すごく役に立つものだった
- ・ すごいと思います
- ・ 説明がわかりやすかった
- ・ 直前まで苦労していたわりにはちゃんと動くものになっていて安心しました。どんどん発展させて次の学年にも引き継げるようにしていったらよいと思います。
- ・ よく解りました。

- ・ 電話で web を見ることが可能の機会があることを知って、今日、文化祭に来た価値があったと思いました。
- ・ もっと派手に宣伝すべし
- ・ 学校の発信する情報とマッチしたらいい
- ・ 面白かったです
- ・ 皆さん力を合わせてよく考えがなばったと思います。感心させられました。
- ・ 技術もあってはじめておもちやおもちじゃなくなると思うので、ツール・技術そして考え方みたいのがさらに統合されるといいプレゼンになっていくのではないのでしょうか。
- ・ 電話でネットできるやつが興味深かったです。
- ・ インターネットといってもあまり使用したことがない（検索とかメールとか…）のですが、大学関係のことなどを手軽に知ることが出来るのはいいことだと思います。
- ・ 初心者にも分かるように簡単に説明していただけるとありがたいです。
- ・ 将来性があると思うので、極めて欲しいと思います。
- ・ バナーが無いのがよい
- ・ 説明をお聞きして、ここまで発表するのにとても大変だったなあと思います。分かりやすく説明もしてありよかったです。
- ・ 江泉ゼミ見に来てくださって、ありがとうございました。ここもすごく興味深いものでした。頑張っているいろいろ作ってってください。

## 【実際のパンフレット】

当日来ていただいた方にお配りしていた私たちの概要を載せたパンフレットと、それに付属していたアンケートが以下の資料7-1と資料7-2です。当日はB4サイズに両面印刷し、3つ折りにした形で配布しました。また”MAP”の部分は手書きで書いたものが別に挿入されました。

1. 「こめぽ」とは？

<http://210.177.commerce/index.html>  
「こめぽ」とはポータルサイトの一種です。  
ポータルサイトとは単独が、インターネットに接続して一連の初めに開くサイトのことをいいます。  
「こめぽ」はグループ内でのコミュニケーションに重点をおいた新しいカタチのポータルサイトです。

従来の提供する「こめぽ」ではインターネットを利用して専学で学生生活をおくる上で便利な情報を交換、共有できたり個人のプライベートな情報を管理できたりするだけでなく、サークル、ゼミ、友達などのグループ内で情報を共有できます。

2. なぜ「こめぽ」を作ったのか？

近年、インターネットが普及し、個人として様々な情報が得られるようになりました。そこで、その機能を利用して、いつでもどこでも専学の情報を得られたり、グループ内でのコミュニケーションをスムーズにする事ができたら、便利でより良い生活が送れるのではないかと思います。「こめぽ」の発想が生まれました。

3. 「こめぽ」でこんなことが出来る！

- ①専学として：専学の学生生活より良いものにするためのもの
  - ・周辺情報—商店、交通、飲食関連のレストランなどのお便り情報を得られます。
  - ・授業情報—学生による授業評価を見られます。
  - ・分科情報—分科の傾向や見られます。
  - ・施設情報—校内の施設の利用状況を見られます。
  - ・図書情報—図書館の蔵書を見られます。
  - ・イベント情報—行われるイベントを見られます。
  - ・リサイクル—廃らなくなったテキストをリサイクルできます。
  - ・ゼミクラブ—サークル紹介
  - ・チャット

- ②グループとして：グループ内でのコミュニケーションをよりスムーズにするためのもの
  - ・スケジュール管理—メンバーのスケジュールを管理できます。
  - ・得意先—投票できます。
  - ・各人の投票—メンバーの所在が分かります。
  - ・メール—メンバーにメールを送れます。

- ③個人として：個人の情報をいれておくことができます。
  - ・スケジュール管理機能—手帳がわりに使えます。
  - ・家庭情報—今までの成績や単位を管理できます。
  - ・時間管理—自分の時間管理を管理できます。
  - ・検索エンジン—必要な情報を検索できます。

<http://210.177.commerce/index.html>



4. モードでも使える「こめぽ」!

ブラウザの欠点といえば、やはりその機能性にあるといえます。そのリレーションが制限された状態にあるのでは？という点から、「モード用「こめぽ」は誕生しました。

<http://210.177.commerce/> 今すぐ行ってみよう!!

5. アンケート

0011080-  
本日は井田ゼミも開学プロジェクト「こめぽ」の誕生にご協力ありがとうございました。お客様からのご意見、ご感想を反映させて、さらによりよい「こめぽ」を作りたいので、貴重なご意見をアンケートにご協力下さい。お願いします。  
※回答する順番を口頭記入してください。

1. あなたの用途をお聞かせください。  
①専学 ②個人 ③その他
2. あなたの性別をお聞かせください。  
①男性 ②女性
3. 「こめぽ」があると便利だな! と思ったり、今使っていますか? 思いましたか?  
①はい ②どちらとも考えない ③いいえ
4. 「こめぽ」の中で、便利だったツールはありますか?  
①はい ②いいえ  
\*「はい」と答えた方にご質問します。  
便利だったツールは以下の何でしょうか? (複数回答可)  
①周辺情報 ②授業情報 ③分科情報 ④施設情報  
⑤スケジュール管理 ⑥チャット ⑦リサイクル  
⑧リンク集 ⑨投票機能 ⑩メール  
⑪その他 [ ]
5. 「こめぽ」を使うことによって、コミュニティ内でのやり取りが楽になると感じますか?  
①はい ②どちらとも考えない ③いいえ
6. 将来にあなたの会社、友達、サークルなどで「こめぽ」を使ってみたいと思いませんか?  
①はい ②どちらとも考えない ③いいえ
7. 将来にあなた自身が「こめぽ」を使用するとしたら、どのようなモードが便利かは、どちらがいいですか?  
①ブラウザ用 ②モード ③どちらでもよい
8. あなたは今までの、グループ、Intranet、domainなど、使ったポータルサイトを利用したことがありますか?  
①はい ②いいえ  
\*「はい」と答えた方にご質問します。  
私の所属するポータルサイトと比べて、「こめぽ」はいいかありましたか?

資料 7-1

- Map -

- ① 変いやかった ②どちらとも考えない
- ② 変いにくかった
- 9. 今回「こめぽ」を使ってきて、あなたの関心のある情報にうまくなりやすくなったと感じましたか?  
①はい ②どちらとも考えない ③いいえ
- 10. 「こめぽ」画面は見やすかったですか?  
①はい ②どちらとも考えない ③いいえ
- 11. 「こめぽ」画面の使い方は簡単に分かりましたか?  
①はい ②どちらとも考えない ③いいえ
- 12. 「こめぽ」画面の中で、さまざまなリンクやダウンロードはスムーズにできましたか?  
①はい ②どちらとも考えない ③いいえ
- 13. 今後、「こんな機能/ツールが欲しい!」と思うものがありましたらお書き下さい。

14. その他「こめぽ」の誕生についてご意見、ご感想、改善点などありましたらお書き下さい。

ご協力ありがとうございました。  
2000年 井田ゼミナール4期生一同

連絡先: 郵局 池田市役所2階5号室  
<http://210.177.commerce/index.html>

~ 井田ゼミナール4期生 ~  
22 / 22

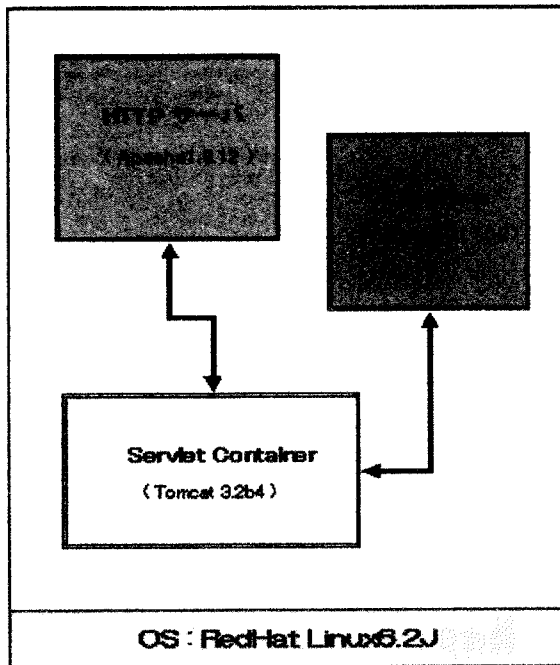
資料 7-2



## 7-5 【パネル】

以下の資料 7-3 から資料 7-11 までが青山祭でパネルブースに設置したパネルです。

### こみゅポのサーバってどんな感じ？



資料 7-3

※ サーバってなんだろう。  
サーバとは、「サービス」を提供するコンピュータのことを意味します。  
そして、サービスを利用する側のことは「クライアント」といいます。

### 家の電話からでもこみゅポ!...

#### Web Touch One



携帯電話からインターネットにアクセスできるようになって、しばらく経ちました。しかしこれからは、家の電話からも、インターネットに接続できるようになります。それを実現するのが、この **Web Touch One** という機械です。

**Web Touch One** は普通の電話機です。しかし大きなタッチパネルがあり、しかも下のほうにはキーボードが内蔵されています。この電話機を使ってインターネットに接続できるのです。この電話機を使えば、わざわざ難しい設定に悩まされることはありません。より簡単にインターネットに接続することができます。

私たちが考えた「こみゅポ」は、「ブラウザさえあればどこでも使える」というのが特徴のひとつです。つまり、**Web Touch One** でも「こみゅポ」は利用できます。**Web Touch One** は家の電話からインターネットに接続するための便利な機械なのです。

(つかいかた)

- 1)接続するには、パネルにある地球儀のボタンを押します。
- 2)文字を入力するには、入力したいフィールドのところを押して、下にあるキーボードを押して引を出し、そこから入力します。
- 3)最初に戻るには、一番左上のボタンを押します。

何かご不明な点がありましたら、お気軽に担当の者にお尋ねください。

資料 7-4

## 青学生・「こみゅボ」に期待ランキング

人とのコミュニケーションを、インターネットや携帯電話、Eメールなど、直接顔の見えない媒体によって取ることが多くなった現代。「友達つきあいはどうなるんだろう?」、「人間関係はどうなっていくんだろう?」という疑問を私たちは持ちました。そこへ「こみゅボ」を提供したら、青学生の生活や人間関係はどのようになるでしょうか?もっとコミュニケーションをスムーズに進めるのに、青学生はどんなものがあつたらいいと思っているのか、そして青学生が「こみゅボ」に必要としている

ものは一体何なのかを知るべく、アンケートを実施しました。(※2000年9月対象青学生約400人)

Q、「あつたらいいな、便利だな」と思うものは何ですか?

A 群:青学生として...

① 授業(新卒) (各授業の進捗支援) 20%

B 群:グループのメンバーとして...

① 部 課活動(同部(同サークル)内での活動) 25%

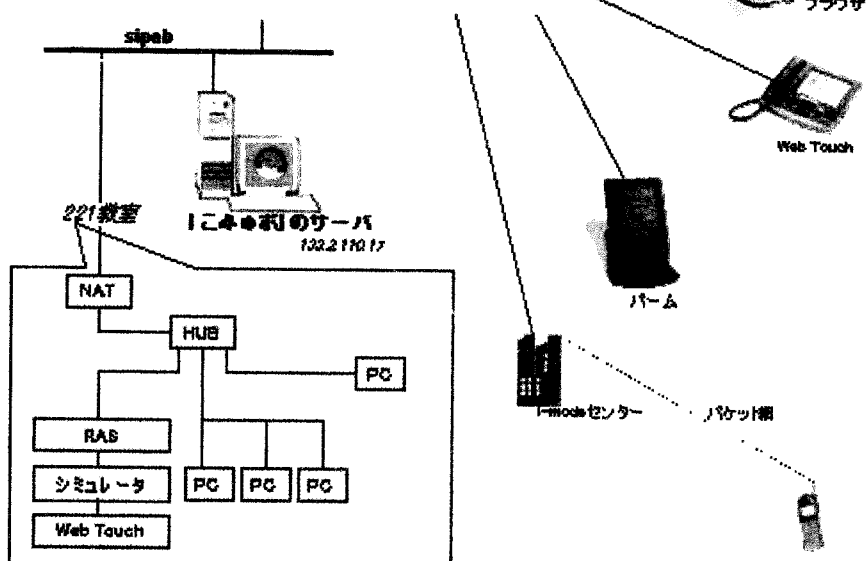
C 群:個人として...

① メール(個人間のコミュニケーション) 30%

資料 7-5

## こみゅボのネットワーク

インターネット



資料 7-6

<コミュニティポータルを作るに当たって>

私達が目指すコミュニティポータルを提供する目的の1つとして実際に学生が Web 上でコミュニケーションを  
 ることのできる場を提供する事です。それは青山学院の公式ホームページのように学校側からの一方的な情  
 報の伝達ではなく私たち学生からでも情報を発信できる場を提供しようと言う事になります。そのために掲示板、チ  
 ャットのようなコンテンツを盛り込み、インタラクティブでダイナミックな動きのあるものを作ろうと決めました。  
 それを実現するに当たってどのような技術が必要かを私たちが学生の中で考えた末、候補として CGI、ervlet  
 (サーブレットの二つが挙がりました。

<なぜ一般に使われているCGIではなくてサーブレットを使ったのか？>

サーブレットがJavaで書かれていることプログラミング言語として学んだ経験があるのはJavaだけであり、  
 CGIを書くためのPerlは学んだ経験はありません。つまり一番身近なプログラミング言語がJavaであったから  
 こ、サーブレットを使う事に決めました。

<どんな勉強をしたのか？(どんな勉強をした方がいいんだろう?)>

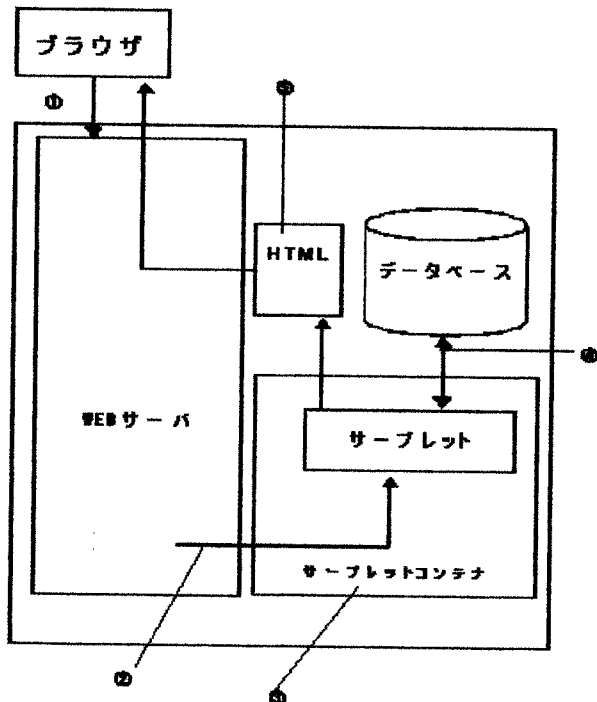
- サーブレットに関する書籍・資料は圧倒的に少なく、レベルの高さも高かったです。Javaをよく知っている人で  
 ないと理論の理解は難しいです。
- 一番の勉強法は、「真似をする」ことだと思います。書籍・資料の理論が分からなくても、そこに掲載されて  
 いるサンプルの真似をして、その動作を自分の目で確認する事で理論ではなく「やり方」を理解することが  
 出来、自分で実際に作る時に応用できます。そして、身近によく知っている人がいれば、自分で作って  
 困った時に聞くことができ、また書籍等にはない情報・やり方をその人から得るのも可能だと思います。

<参考資料>

- JAVÄ SERVLET 最新サーバプログラミング
- JAVÄ PRESS Vol.13
- まるごと図解 JAVÄ がわかる

資料 7-7

サーブレットについて



- ①: ブラウザがInternet Explorer などからWEBサーバに対して要求を出す。
- ②: WEBサーバは、その要求がサーブレットを起動させるものならば、それをサーブレットコンテナに送る。
- ③: サーブレットコンテナは、サーブレットプログラムを起動する。
- ④: サーブレットは、必要ならばデータベースに問い合わせをする。
- ⑤: データベースから返ってきた結果をサーブレットはHTML化してブラウザに送る。

資料 7-8

ここで「ユーザー登録」の画面を説明し、人が何の目的でこのサイトにアクセスしているか、その目的を把握して、適切な対応をとる。

私達がコミュニティポータルを作成する際、上のようなことを目指しページにアクセスしている人が誰であるのかを特定しその人にあった情報を提供しようと考えました。

#### ① 必要な機能を考える

アクセスしている人を特定する為以下の三つの機能をこみゆりに持たせました。

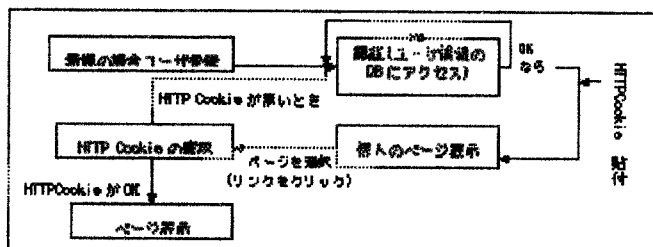
- ① 最初にポータルにアクセスした時に、サイト利用に必要なユーザに関する情報とアクセスする為のIDとパスワードの設定してもらう機能(ユーザ登録)
- ② 他人の情報があるページにアクセスする前にIDとパスワードを入力してもらい確認する機能(認証)
- ③ 認証後各ページにアクセスする際に、アクセスしている人を特定する機能(認証を一度すればその後は認証などが必要なくなる)

#### ② データベースとCookieの活用

私達は上記の3つの機能を行う為、登録したデータを保存しておくデータベースとサイトにアクセスしている間ブラウザに今アクセスしている人の情報をPC上に保存する必要をまにサーバが読み取ることが出来るHTTP Cookie を用いました。

#### ③ 認証とデータベースの連携

私達が目指していた機能の三つの機能を、データベースとHTTP Cookie を使用し、図のように使用しています。



- ① まず、初めてこみゆり訪れた人は、ユーザ登録をしてもらいます。そのとき、各自、IDとパスワードを指定します。その情報をサーバはユーザ情報のあるデータベースに格納します。
- ② 2回目以降の人及びユーザ登録を済ませた人は、登録してあるID&パスワードを入力し認証を要します。このときサーバは、ユーザ情報のあるデータベースにヒットするデータがあるかないかを検索します。
- ③ サーバはデータベースにヒットするユーザ情報があれば、その人であると認証を行い、その人のデータをもとに個人ユーザ向けのトップページを作成するとともに、その人の情報を入れたHTTP Cookieを発行します。(データベースにヒットがなければ認証のページを再表示します)
- ④ 次にユーザ側がアクセスされたページのリンクを辿ると、リンク先はクライアントPCがユーザ情報を持ったHTTP Cookieを持っているかを検索します。
- ⑤ クライアント側のPCにHTTP Cookieがあれば、HTTP Cookieにあるユーザの情報を元にページを作成します。(HTTP Cookieが見つからなければ、再度認証の画面に移動します)
- ⑥ コミューンを見終わったとき、ブラウザを閉めます。そうするとクライアント側が持つHTTP Cookieは削除されます。

資料 7-9

## < “こみゅぽ” の将来と可能性 >

### 「こみゅぽ」の対象を決める経緯

私たちが今回、コミュニティーポータルを作るに当たってまずゼミ生の間で一番議論を重ねたのがこのコミュニティーポータルを誰に向けて発信し、使ってもらうかという事でした。それはこのコミュニティーポータルを作っていく上で一番重要な要素の一つでした。

色々議論を重ねていく上で、今回はまず私たちは大学生という大きな集団をターゲットにコミュニティーポータルを作っていくと話が固まってきました。ところが、それでは余りにもターゲットが大きすぎ、中に盛り込む内容を絞りきれず、既存のポータルと差別化できないという問題にいき当たりました。そこであくまでも最終的にこのコミュニティーポータルを使っていくのは大学生であるという理想を掲げつつ、まずは実験的に大学生の代表として私たち青山学院大学の学生をターゲットにコミュニティーポータルを作ることに決めました。実際に作業を進めていく上で、この決定は良かった事に気がきました。

サイトの中に組み込むツールは青学生に絞る事でより学生の意見を反映することができ、より学生にとって身近なものに出来上がったと私たちは思っています。

資料 7-10

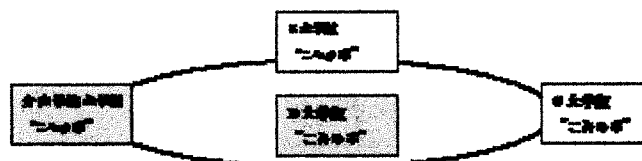
### “こみゅぽ”の夢その1:「プロトタイプ」の提供

今回の青山祭での展示は1大学としてのコミュニティーポータルでありこのプロトタイプをそれぞれの大学に提供し、その各大学で中身をカスタマイズしてもらう事によって、その大学独自のコミュニティーポータルを作り上げていてもらいたいです。

### “こみゅぽ”の夢その2:「sharing the information」

そして、各大学で独自の“こみゅぽ”が形成されたのなら、それらの“こみゅぽ”間での情報の共有を実現したいのです。例えば、今回の青山祭の日程などの情報は決して青山学院大学内の“こみゅぽ”だけでなく、他大学の“こみゅぽ”においても公開されていていいものなのです。より多くの情報を共有することにより学生たちが生活をより豊かなものにし、コミュニケーションを円滑に行うことができるのではないのでしょうか。

私たちがこのコミュニティーポータルを作った理由は大学生による大学生のための情報提供の場、コミュニケーションの場を提供する事です。それを実現するために、私たちの「こみゅぽ」が貢献できる可能性を秘めていると私たちは確信しています。



資料 7-11

## 7-6【ポスター】

宣伝のために青山祭期間中は特定の場所を除き、校内の壁にポスターやビラを貼ることができます。私たち4期生が貼り出したポスター（資料7-12参照）は全て統一し、一つのデザインで多くを貼りだしました。

青学生のためのコミュニティーポータルサイト

# こみゆポ!!

開催日: 2000年10月28~30日

場所: 2号館221教室

内容: ①デモンストレーション

百聞は一見に如かず!実際に体験してみましよう!!

②プレゼンテーション

私たちが目指すもの、それは……!!

青山学院大学国際政治経済学部

井田ゼミナール4期生

<http://133.2.110.17/commupo/>

資料7-12

# 第 8 章

## アンケート

8-1 【アンケートを作るまで】	p. 183～
8-2 【アンケート集計結果】	p. 184～
8-3 【アンケートの考察】	p. 187～
8-4 【アンケートの反映】	p. 188

# 第8章 アンケート

〈〈文責：土橋りつこ

\* 事前アンケート編 \*

## 8-1【アンケートを作るまで】—こみゅぽ！をつくるにあたって…—

前述したように、青山学院の学生が青学生として必要な情報を共有できたり、グループ内でコミュニケーションをとるのに便利なサイトがあったらいいな…という構想が私たちの間にありました。そして、作るなら本当に学生に使ってもらえて、青学生が大学生活の上で本当に役に立つものをつくろうと考え、そのためには絶対に事前調査が必要だと思いました。

そこで、私たちは、どのツールが本当に学生に必要とされているのか、また必要のないツールは何なのか、そして学生はこの情報交換サイトに何を求めているのかを調べるためにアンケートを作成し、実施しました。

こみゅぽ！は青学生として使うのに便利なツール、グループ内でコミュニケーションをとるのに便利なツール、そして個人が手帳がわりに使えるツールの3種類を用意していたので、項目をその3つにわけてそれぞれいいと思うツールを3つずつ選んでもらいました。

\* アンケートを行った対象は青山学院大学に通う1年生～4年生までの学生500人です \*

次の図8-1及び図8-2がそのアンケートです。

### 青学生の情報システムに関するアンケート

私たち国際政経井田ゼミナールでは、青学生のための情報交換サイトを作っています。

そこで、皆さんの意見を反映させるため、アンケートにご協力をお願いします。

学科（ ） 学年（ ） (男・女)

I. 次にあげる機能（または情報）の中で、あったらいいなと思うもの、便利だなと思うものを各群からそれぞれ3つを選んで、マルをつけて下さい。

A群（青学の学生として）

- |                          |                                  |
|--------------------------|----------------------------------|
| 1. 学食ランキング               | 学食の情報を交換します。                     |
| 2. 授業評価                  | 各授業の意見の交換をします。                   |
| 3. 教員ランキング               | 教員の学生レベルでの評価をします。                |
| 4. 渋谷・表参道（または世田谷・厚木）周辺情報 | 学校周辺の情報を交換します。                   |
| 5. 休講情報                  | 休講情報を提供します。                      |
| 6. 礼拝情報                  | 礼拝の情報を紹介します。                     |
| 7. 学校施設利用状況              | セミナーハウスや教室など<br>学校施設の利用状況を紹介します。 |



図 8-1

B 群 (グループのメンバーとして)	
1. 個人・グループスケジュール管理機能	スケジュールを一覧できます。
2. 授業時間割	メンバーの履修状況を一覧できます。
3. プロフィール紹介	メンバーの紹介です。
4. 住所録・連絡先	メンバーの連絡先を一覧できます。
5. メンバー所在検索	メンバーの所在を確認できます。
C 群 (個人として)	
1. 文章作成ツール	Word のようなものです。
2. 表計算ツール	Excel のようなものです。
3. メール	メールの送受信ができるようにします。
4. 単位計算機能	単位の計算が出来る計算機です。
5. 検索エンジン	ホームページの検索に使います。
Ⅲ. 上にあげたもの以外で、あったらいいなと思うもの、また使いたいと思うものを自由に書いてください。	
<div style="border: 1px solid black; height: 60px; width: 100%;"></div>	
ご協力ありがとうございました。集計結果は今年度の青山祭にて発表いたします。 青山学院大学 国際政治経済学部 井田ゼミナール	

図 8-2

## 8-2 【アンケート集計結果】 —実際にアンケートを採ってみて…—

アンケートの結果は表のようになりました。

A 群「青学の学生として」の項目では…

1 位	授業評価	28%
2 位	休講情報	24%
3 位	教員ランキング	24%
4 位	周辺情報	19%
5 位	施設利用状況	7%
6 位	学食ランキング	5%
7 位	礼拝情報	1%

このように授業評価が全体の 28% と最も高い人気でした。続いて、休講情報、教員ランキングとなっています。

B群「グループのメンバーとして」の項目では…

1位	授業時間割	23%
2位	スケジュール管理	22%
3位	住所録・連絡先	20%
4位	プロフィール紹介	18%
5位	メンバー所在検索	17%

ここでは授業時間割が全体の23%と最も高い人気でした。次にスケジュール管理、住所録・連絡先となっています。

C群「個人として」の項目では…

1位	メール	30%
2位	検索エンジン	26%
3位	文章作成ツール	22%
4位	単位計算機能	13%
5位	表計算ツール	9%

メールが全体の30%と最も高い人気で、続いて検索エンジンという結果でした。学部学科別の全体の表は図8-3を参照してください。

【事前アンケート集計結果】

有効回答数：373 (人)

A群	国際政治		国際経済		国際経営		英米文		日文		教育		経済		経営		私法		公法		芸術		合計	
	男	女	男	女	男	女	男	女	男	女	男	女	男	女	男	女	男	女	男	女	男	女		
1 学食ラッキンダ	2	7	7	7	12	6	1	3	1	2	0	0	2	0	1	2	0	0	0	0	0	0	1	54
2 授業評価	22	28	23	36	71	49	9	27	4	1	2	3	4	3	13	3	1	1	1	1	0	0	0	302
3 教員ラッキンダ	15	23	22	24	64	35	7	19	4	0	2	3	5	1	10	2	1	2	1	1	0	0	0	240
4 渋谷・表参道周辺情報	10	20	10	21	47	36	4	21	1	2	1	0	4	2	3	0	1	3	0	0	1	1	1	199
5 休講情報	17	23	11	27	67	43	5	30	3	0	1	1	4	2	6	2	0	2	1	1	1	1	1	246
6 礼拝情報	1	0	0	1	2	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	8
7 学校施設利用状況	2	6	5	10	19	7	3	14	2	0	0	2	1	1	2	3	0	1	0	0	0	0	0	78
B群																								1121
1 スケジュール管理	15	25	17	31	64	39	8	22	4	1	2	2	4	2	10	3	0	0	1	1	1	1	1	251
2 授業時間割	17	25	18	30	64	38	9	23	4	2	1	3	6	3	11	3	1	2	0	0	0	0	0	261
3 プロフィール紹介	13	17	20	19	50	31	6	24	3	0	1	1	1	1	7	2	1	1	0	0	1	1	1	199
4 住所録・連絡先	13	20	16	27	55	42	2	24	3	1	1	1	4	0	8	3	1	3	1	1	0	0	0	225
5 メンバー所在情報	11	21	6	19	49	26	5	21	1	2	1	2	6	3	6	1	0	3	1	1	1	1	1	185
C群																								1121
1 文章作成ツール	19	28	16	27	55	40	7	29	4	1	0	2	4	2	9	3	0	1	0	0	1	1	1	248
2 表計算ツール	4	9	9	9	30	20	4	6	1	1	0	0	3	0	2	2	0	0	0	0	0	0	0	100
3 メール	21	30	25	39	79	54	9	34	4	2	2	3	7	3	12	4	1	3	1	1	1	1	1	334
4 単位計算機能	6	10	9	17	41	13	6	18	3	0	2	1	3	1	7	0	1	3	1	1	0	0	0	142
5 検索エンジン	19	31	19	34	77	49	4	27	3	2	2	3	4	3	12	3	1	2	1	1	1	1	1	297
	207	324	234	378	846	528	90	342	45	18	18	27	63	27	126	36	9	27	9	9	9	9	9	2517

人数	23	36	26	42	94	58	10	38	5	2	2	3	7	3	14	4	1	3	1	1	1	1	373
----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	----	---	---	---	---	---	---	---	-----

	男	女
人数	182	191
割合	49%	51%

図 8-3

### 8-3 【アンケートの考察】 -アンケート結果から……-

このアンケートの結果をもとに、私たちの創ろうとしているこみゅぽ！に学生達が何を求め、期待しているのかを考察してみました。

A群の青学生として必要な情報を提供するというツールの中では1位の授業評価、2位の休講情報、3位の教員ランキングなど圧倒的に授業に関係しているツールが人気を集める結果となりました。これはいかに学生が授業に関する情報を手に入れにくいかという実情を表していると思います。実際、休講情報などは学校の掲示板（学校の一番奥まったところにあり、めったに見ることがない）で見ると以外に情報を事前に手に入れることが出来ないため、「ぜひ自宅のパソコンやiモードで見れたらいい！」という強い要望が目立ちました。逆に学食ランキング、礼拝情報など授業に関係のないツールは学生にとって手に入れにくい情報ではないのではないのか、また情報がそれほど欲しいものではないのではないのではと思えました。

B群のグループがコミュニケーションをとるためのツールではあまり人気に大差はなかったものの、やはり1位のグループのメンバー授業時間割は授業に関してのツールであり、学生がいかに授業情報を重視しているかが分かる結果になったと思います。予想外に人気のなかったのはメンバー所在情報でした。これは昨今、携帯電話が普及してきて同じグループのメンバーとも、簡単にいつでも連絡がとれる状態にあるので、いつでも所在が明らかになっているということが関係していると思えました。

C群の個人として使うためのツールではメールが高い人気を示しました。やはりメールは学生が頻繁に使うツールであるということを反映しての結果だと思われます。表計算ツールや文章作成ツールは大体の人が各人のパソコンに持っていて、メールほど頻繁に使わないということで、必要性はあまり高くなかったようです。単位計算機能は名前だけではどういうものなのかがいまいち分からなかった人も結構いたようですが、履修登録の際には絶対に必要という強い要望も多かったです。

#### 4. その他の意見（あったらいいもの・使いたいもの）

- ・ 就職系情報（多数）
- ・ 友達作り情報
- ・ 教科書入荷情報・在庫情報
- ・ 学割目玉商品・購買のセール情報
- ・ 学校周辺の交通情報
- ・ 試験（TOEFL・TOEICなど）の受付期限
- ・ 学校近辺の病院情報

- ・ 学院内お見合い用掲示板
- ・ 図書館・テブライブラリ貸し出し状況
- ・ 教授の居場所検索
- ・ テスト・レポートの傾向と対策
- ・ 講演会などのイベント情報
- ・ 大学院・ダブルスクール情報
- ・ 青学周辺の天気予報
- ・ OB・OG と交流できるもの
- ・ 他大学情報
- ・ 試験の過去問・予想問題
- ・ アルバイト情報
- ・ バス時刻表
- ・ タイムリーな話題特集

特に就職情報やゼミ情報（次年度にゼミをとる2年生のための）などの進路関係の要望が多く見られました。他には、学院内の知らない人とコミュニケーションをとり友達を増やしたいという人も多かったです。一番下に用意していた自由に要望をお書きくださいの欄には色々な意見やこうした方がいい、などというアドバイスを書いてくれた人が思いのほかたくさんいたことに驚き、それと同時に学生の web やポータルサイトへの高い関心を感じる事が出来ました。

#### 8-4 【アンケートの反映】 ―取ったはいいけど…―

このアンケートによって分かったこと、得たものはたくさんありましたが、実際のところこのアンケートの結果や学生さんの意見を私達のつくったコミュニティーポータルサイトに反映させるということはまだ出来ていない、というのが現状です。このことに関する反省点としては、やはりアンケートをつくる段階でかなり手間取ってしまって、アンケートを取り、集計した時点でもうツールがかなり出来上がってしまっていて、ツールを改良したり、皆の要望にあるようなツールを追加することも難しくなっていたということが挙げられます。今後、このポータルサイトを続けていくのならば（今のところ未定）、このサイトが学生の要望に答えることのできる、本当にみんなに使ってもらえるようなものになっていくために、このアンケートはとても有用だったと思います。

## 第9章

### 将来のこみゅぽ

9-1【私たちの夢の実現、将来の「こみゅぽ」】

p. 190～

## 第9章 将来のこみゅぽ

〈文責：黒田洋史、大西淳

### 9-1【私たちの夢の実現、将来の「こみゅぽ」】

私たちは、コミュニティポータルというテーマに着目し、3ヶ月という長い時間をかけて取り組んできました。みんなで、コミュニティポータルをどういうものにすれば多くの人に使ってもらえるかという話し合いを何度も行い決めてきました。それが、大学という大きな枠のコミュニティとゼミやサークルなどの小さなコミュニティを融合するという概念です。他にも、管理者がいなくても自動的に情報の更新ができることなども概念として考えてきました。私たちは、話し合った概念を青山祭という場で多くの人に共感していただけたらと考え、サイト作りを開始しました。私たちの概念をサイトにするには、多くの技術が必要であることがわかり、勉強を平行し行いました。その勉強の中で様々なことがわかりました。そして、多くの技術を利用しながらも青山祭で何とか形のあるものとして公開できるまでに至りました。

青山祭に来ていただいた方は、実際にデモを触っていただいたりプレゼンテーションを見ていただいたりしていただく中で私たちの考えやコンセプトに共感していただきました。

しかしながら、操作性の問題・セキュリティーの問題など多くの点において助言をいただきました。私たちはこれらの助言を、私たちを応援していただいている上でいただいているのだと考え、うれしく思いました。まだまだ私たちのコミュニティポータルという概念自体も足りないことも多いですし、技術的な面でも実現を諦めたものもあります。しかし、私たちの夢はまだまだ進化していて終わりはありません。私たちは、このコミュニティポータルを多くの方とともにこれからも作っていきたくて、そうすべきであると考えています。今回のプロジェクトで、多くの方にこみゅぽにご関心を持っていただき、新しいこみゅぽ作りに参加していただけたらと考えています。そうやって多くの方が力を合わせてコミュニティポータルを作っていただくことがこれからの「こみゅぽ」の第一歩ではないかと思えます。

#### 「こみゅぽ」の潜在能力

人は、一人では生きていけません。人は、人と一緒に生活しています。人が人とかかわっています。それがコミュニティの形成の原点です。人は、多からず少なからずコミュニティに所属して生活しています。昔から、「井戸端会議」などというような形でコミュニティ

ィでの情報交換が行われてきました。そして、人はコミュニティ内で情報交換し共有することで自分の価値を高めていったといえます。それが文明や文化であるというのはいまでもありません。

しかしながら、スピードや能率が重視され「個人主義」という言葉が乱用される現代、人と人が顔を合わせて話す時間すら取れなくなり、「井戸端会議」という言葉はなくなりつつあります。しかしながら、人と人は協力なくして生きていけませんしコミュニティに所属せずに生活はできません。そこで、人と人が同じ場を共有しなくても実現できるサイバースペースを利用しようということになります。その概念を持つのが、こみゅぽです。そういう意味で、人と人の情報交換や人と人の文化形成そして「井戸端会議」の場の一つの手段としてコミュニティポータルいや、「こみゅぽ」の存在意義、ポテンシャルが存在するのではないかと考えます。

コミュニティポータルが有効に利用されるのはその人一人一人が自分の所属しているコミュニティに参加したいというモチベーションが必要です。それはコンピュータという機械では作ることはできません。よって、もし人が自分だけで生きていこうとし始めたとき、このコミュニティポータル自体不要なものになるのです。つまり、たとえコミュニティポータルがあっても自分がどういうコミュニティに所属しているか考える必要は残ります。でも、自分の所属するコミュニティについて考える瞬間を持つことは、他人のことを考えることにもつながります。そうすることで、また人間や文化は進歩するのではないのでしょうか。例え、どんなに文化が進歩しても人とともに生きるということを考えていくべきですし、そうするための一つの手段としてコミュニティポータル、「こみゅぽ」が使われればと思います。



# 第 10 章

## 借用機器

10-1【各借用機器について】

p. 193～

10-2【企業への訪問】

p. 196～

# 第 10 章 借用機器

◀◀文責：森健蔵

## 10-1【各借用機器について】

私たちが今回のプロジェクトを行うにあたり、学生の私たちの力では揃えることが出来なかった機器を企業の方々にお借りいたしました。ここにその一覧と使用目的等を記したいと思います。

### 1. お借りした企業先

株式会社東芝

### 2. お借りした機器

ノートブック型 PC (計 5 台)

- ・ Satellite2210 SA50C/2C8(PA-SA50C2C8)× 1 台  
OS : Windows98SE CPU : Celeron500MHz メモリ : 64MB  
ディスプレイ : 12.1 型 TFT(SXGA) HDD : 6GB  
CD-ROM : 24 倍速 FDD : 3.5 型 LAN : 100Base-TX/10Base-T
  
- ・ DynaBook SS 3480(PA-DS60P1N8L)× 2 台  
OS : Windows98SE CPU : PentiumⅡ600MHz メモリ : 64MB  
ディスプレイ : 11.3 型 TFT(XGA) HDD : 12GB  
LAN : 100Base-TX/10Base-T
  
- ・ Satellite4360 SA70P/5CA(PA-SA70P5CA)× 2 台  
OS : Windows98SE CPU : Pentium700MHz メモリ : 64MB  
ディスプレイ : 15.0 型 TFT(XGA) HDD : 12GB  
CD-ROM : 24 倍速 FDD : 3.5 型  
LAN : 100Base-TX/10Base-T

デスクトップ型 PC (計 2 台)

- ・ EQUIUM9000(PA-EQ70PTC8B)×2 台

OS : Windows98SE CPU : Pentium□700MHz メモリ : 64MB  
ディスプレイ : 15 型液晶フラットパネル(XGA) HDD : 15GB  
CD-ROM : 40 倍速 FDD : 3.5 型 LAN : 100Base-TX/10Base-T

### 3. お借りした期間

00/10/16~00/10/30

- ・ Satellite2210 SA50C/2C8(PA-SA50C2C8)× 1 台
- ・ DynaBook SS 3480(PA-DS60P1N8L)× 2 台

00/10/26~00/10/30

- ・ EQUIUM9000(PA-EQ70PTC8B)× 2 台
- ・ Satellite4360 SA70P/5CA(PA-SA70P5CA)× 2 台

### 4. お借りした理由・使用方法

今回の学生向けのコミュニケーションポータル“こみゅぽ”を実演するには当たり前ですが PC が必要なのは言うまでもありません。私たちゼミ生は各自ノート PC をそれぞれ持っていますが、画面が小さい、キーボードが小さい、スペックが十分ではないなど展示する時に大きな支障となる要因が幾つもありました。このような理由からより画面が大きくハイスペックな PC をお借りする事にしました。使用方法としては Satellite2210 を仮のサーバーを組むことに使い、後の PC は当日の自宅ブース、体験コーナーに展示しました。

### 1. お借りした企業

株式会社オレンジソフト

### 2. お借りした機器

携帯端末 P 209is ×1 台  
携帯端末 N502i ×1 台

### 3. お借りした期間

00/10/27~00/10/31

### 3. お借りした理由・使用方法

私たちがこの学生向けのコミュニケーションポータル“こみゅぽ”を作っていく過程で最初からあるコンセプトを掲げていました。それは何処からでもこの“こみゅぽ”に接続できるというものです。それを実演として見せるには携帯端末を使うのが一番

良い方法だと考えたからです。今、現在、携帯端末からインターネットへ接続する人口はPCからそれへ接続する人口を追い抜かんとする勢いで増加していること。また今回のコミュニティポータルターゲットが大学生という最も頻りに携帯を使っている世代であることなどを考え携帯端末をお借りする事にしました。

使用方法としては教室内に作った外ブースでの展示、また実際にキャンパス内に出て、“こみゅぽ”の実演を行いました。

#### 1. お借りした企業

アルカテル・インターネットワーキング株式会社

#### 2. お借りした機器

LAN Rover Plus (RAS=Remote Access Server)

28,8K モデム 8 個内蔵・Ethernet(10B-T,10B-2,)内蔵 ユーザ管理、DHCP Server

製造元: INTEL/Shiva

電話機テスター (シミュレータ)

モジュラージャック 2 口

#### 3. お借りした期間

00/10/26~00/10/30

#### 4. お借りした理由・使用方法

私たちの手元には井田教授から貸していただいた WebTouch という電話とブラウザーが一つになった機器がありました。ちょうど私たちのコンセプトの一つに、携帯端末をお借りした理由のように“こみゅぽ”にどこからでもアクセスできるようにするというものがありました。そしてこの WebTouch も将来今の電話機に取って代わり主要なアクセス手段になりうるだろうことから青山祭の展示で使うことを決めました。そのために WebTouch の電話回線を LAN のネットワークにアクセスさせる機器が必要なために RAS を WebTouch に擬似的に電話番号をふるためにシミュレータをお借りしました。

青山祭当日は自宅ブースの一角に WebTouch と共に展示し、実際に来場された方々に画面をタッチして頂きいろいろとネットサーフィンをして楽しんでいただきました。

## 10-2【企業への訪問】

〈文責：森健蔵

今回のプロジェクトの中で企業訪問を通じて色々なことを学び、感じました。それをここに記します。

私たちが機器を提供していただくために企業訪問を開始したのはもう9月の下旬でした。

8月下旬の夏休みの合宿において先輩の方々から早く企業を訪問した方がいいと指摘されたにも関わらず、この時期まで伸びてしまったのには大きな理由があります。それは企業を訪問した時に私たちのプロジェクトの売り、ここが他のコミュニティーポータルと違うという差別化が説明できていなかった事にあります。また私自身もインターン先の方々に初版の企画書を見せた時に“このプロジェクトって既存のものとは何処が違うの？”と聞かれ、言葉に詰まってしまいました。そして9月中にようやく私たちが作るコミュニティーポータルの売り、コミュニティーの中にコミュニティーを作るといった2重構造の概念が明確化されました。そして最終的な企画書がついにできました。

そして企業訪問が始まりました。今までにない経験の連続でした。何を如何したらいいか全く分かりませんでした。それは名刺の渡し方、受け取り方ひとつとってもそうでした。

どのようにすれば私たちのプロジェクトについて簡潔で、分かりやすく、かつ魅力的に説明できるのかを常に心がけていました。事前にここと、ここの場所を中心に話そうと訪問する担当のもの同士で打ち合わせをしておいても、いざ企業の方々にお会いして、話して見ると極度の緊張で自分が何を言っているのかが理解できないこともありました。それは企業の方々が凄く真剣になって私たちのプロジェクトについて耳を傾けて下さり、色々な側面から質問、ご指摘をして下さったからではないでしょうか。

時には企業を訪問する際に余りの勉強不足にご迷惑をおかけした事もありました。特にWebTouchに関してはそれを実際LANにつなげるには何が必要か全く分からない状態で企業の方にお会いしていただきました。そしてそのお会いした場で、お話を伺いながらに本当には何が必要なのかを学んだりしていました。今、思うと本当に失礼な事をしていたと深く反省しています。

今回のプロジェクトでは私たちの計画性のなさもあり企業訪問がかなり時間との戦いになってしまいました。そのために企業の方々には時間的に無理なお願いをしてしまったことを深く反省しています。また時間にもっと余裕があれば、企業にもものをお借りするだけでなく私たちが目指す“こみゅぽ”の手本とすべく、コミュニティーポータルを作っている企業にも訪問するべきでした。このような活動を夏休みに入った頃に始めていればメンバーのモチベーションもより上がり、もっと良い“こみゅぽ”が出来上がっていたのかもしれない。

<企業に対しての活動歴>

- 9/26 株式会社東芝、キャノンにメールを出す
- 9/29 電話で株式会社東芝と面会の約束を交わす
- 9/30 NTT データにメール、キャノンに再度メール
- 10/1 NTTDoCoMo にメールを出す
- 10/2 株式会社東芝を訪問
- 10/3 NEC にメール
- 10/4 NEC との面会の約束を交わす
- 10/11 NEC の方々と面会
- 10/14 株式会社オレンジソフトへメール
- 10/16 東芝より PC 3 台が青学へ搬入
- 10/19 株式会社オレンジソフトより携帯端末の提供をとりつける
- 10/24 アルカテル・ネットワーキング株式会社訪問、株式会社オレンジソフト訪問
- 10/27 株式会社オレンジソフトへ期待端末を取りに行く
- 10/30 株式会社東芝へお借りした機器を手渡して返却  
アルカテル・ネットワーキング株式会社からお借りした機器を郵送で返却
- 10/31 株式会社オレンジソフトからお借りした機器を手渡して返却
- 11/15 ご協力していただいた企業に御礼及び報告会への招待状の手紙を送る

# 第 11 章

## プロジェクト後記

11-1 【荒金啓子】	p. 199～
11-2 【大西淳】	p. 201～
11-3 【岡田牧男】	p. 203～
11-4 【姜秀珍】	p. 206～
11-5 【黒田洋史】	p. 209～
11-6 【清水淳】	p. 210～
11-7 【土橋りつこ】	p. 213～
11-8 【森健蔵】	p. 214～
11-9 【矢島達也】	p. 217～
11-10 【山崎淳子】	p. 220～

## 11-1 【荒金啓子】 as keiko

〈〈文責：荒金啓子

### ◆担当の仕事を通して◆

プロジェクトの中で特に大きな仕事だったのは、やっぱり青山祭でのプレゼンテーションを作ることでした。プロジェクトは「コンセプトが先だ!」「いや、早く具体的にツールを作っていないと!」「すでに作られているツールを探し出す作業が大切なんだ!」とみんなであーでもないこーでもないで散々議論しあってきました。そんなことをしているうちに青祭の時間がどんどんせまり、技術的にもほぼ初歩からのスタートだったこともあって、ツール・サイト作りと実際に青祭で何をどう展示するかということで大忙しという感じでした。もう、いっぱいいっぱいのパニクリ状態という感じです。そんなこんなで最後まで最終的なコンセプトとなるものが、みんなの頭の中ではそれぞれ着実に暖められてきているのに、それをはっきりと文章化したものがないという状態でした。みんなの考えを汲みつつ整理されたかたちで、これまでの私たちの考えてきたこと、やってきたことを何も知らない外部の人達に伝えるにはどうしたら一番良いかということに頭を悩ませました。

青祭での展示も担当しました。展示をするという経験がほとんど皆無といってもよく、まず教室の寸法を測らなければいけないということさえ知りませんでした。物品調達では、限られた資金内でどこに、どんなものを借りてくるべきか、買って来るべきかを練らなければなりません。しかし、練ったほどの効果が出なかったり（しきりに使ったテーブルクロス。）、結局借りてこなくてもいいものを借りてしまったりしました（まったく使わなかった白熱灯。）。i-modeの展示で臨場感を出すために学校にあるベンチを使いたかったため、施設課、学生課と飛び回って可能かどうかを打診したことがありました。しかし、正式な文書で申請するときに、青山祭実行委員会（青実）という青祭を取り仕切る組織で「非常に例外的なお願い」ということでストップをかけられてしまったときは残念でした。文化祭の参加を通して、イベントを行うにも多くの役割とそれに伴う責任・立場があるということを知りました。また、前日の準備日まで教室のセッティングを試すチャンスがないことと、青実を通した物品のレンタルは3週間ほど前が申し込み期限ということから、実際にセッティングした時、どんな感じか、どんなものが必要か、どんな問題が起こるか、をすべて事前に予測しなければならなかったことが大変でした。

### ◆こみゅぽについて◆

5月、井田先生から「Community Portal」と今まで聞いたことも見たこともない言葉を投げかけられ、途方にくれました。まず、良く言う「ポータル」自体、その言葉の意味って何なんだろう?というところから始まりました。そのときの私にとって、WebTop - ブラウザさえあれば、いつでもどこでもデスクトップのような環境を提供できる - という概念がとても新鮮でコミュニティーポータルはWebTopに近いものなのではないかと考えていました。



が！プロジェクトを進めていくうちに、コミュニティポータルでは、ある人の集まりを対象としてその中でのコミュニケーションに役立つような機能が主役だろうという声が上がってきました。最もだと思いました。WebTopにあらう機能（例えば、WordやExcelのようなもの、メールの機能）を作ったり、既にある機能を発展させて作るなどということが私にとって技術的な壁が高かったことやプログラミング初心者な私たちにはコミュニケーションにかかわるツールを作っていくことで手一杯であって優先するべきことでした。こうして「コミュニティポータル」から「デスクトップ環境がいつでもどこでも」という見方が消えていってしまうような感がありました。第2章でも書きましたが、将来は、その両方が実現すれば、人が、「便利」「楽しい」と感じるようなサイトができて、人が集まるサイトを作り出すことができると思います。

青祭でプレゼンをしたということもあって、様々な質問を受けました。その中でも印象に残った質問が、学生向けのサイトは大きな市場であって様々な企業が狙っているが、なぜそれがあまり実現していないのかということでした。この質問に対して私が思うのは、学生のためのサイトは一社独占のかたちではなく、多くの人の手で作り上げていくことが良いのでは、ということです。こみゅポにあるすべてのソースを「コピーレフト」にして、サイトにあるツールを多くの人の手で改良していくということです。

実現が可能であると思われる理由として以下が挙げられます。

- 学生は何かに参加・貢献していると実感するのが好き
- 自分で作った功績が残る（文系の学生も能力を示す機会を持てる）
- 作ったものを多くの人に試してもらえる・感想が聞ける機会がある
- 自分自身のために使うサイトだから何が欲しい、何が要らないがわかる

これを本当に現実的なはなしにするには、もちろんまだまだ考えを練らなければいけません。また、こみゅポがそのたたき台の役割となるには、予めもっと改良が必要です。また、一番求められているものは何かがよくわかったり、後から作られたツールがうまくサイトに汲みこまれていく仕組みも用意しなければいけません。例えば、感想を書くだけでなく、出された意見に対して投票ができて、一番多く投票されている意見が一番目にとまるようになる仕組みを用意する。また、集まったツールを検索して自分たちのコミュニティで使うツールとしてドラッグアンドドロップぐらい簡単に採り入れられる。でも、多くの学生に使ってもらい、プログラミングができない、興味がない人達からの意見もプールして、そこから能力のある人がより良いものを作ろうとチャレンジできるサイトになっているととても楽しいと思います。いろいろなアイデアのいろいろな能力が集まるように。

#### ◆プロジェクトを通して◆

まず、具体的な私に関する反省点です。もっとプログラミングやサーバのことに理解を深

めれば、どんなことが可能かわかって、そこからアイデアが沸いたり、取り合えず自分のアイデアをデモ的にどんどん提示していくことができたと思います。何かを企画する側としても重要な能力だということがわかりました。また、プレゼンテーションを作った時に、私たちの取り組んでいることをあまり外部に発信していなかったことがかなり良くなかったことに改めて気付きました。自分だけで取り組むことに埋没するだけではだめで、物事を進めながら、途中でも外部にも伝えていかなければいけないことが時にはとっても大事なんだと気付いたのです。そうした方法をとってれば、技術的に至らない部分は、自分の力で実現していくことも大切だけれど、私たちは私たちだからこそ出せる新鮮なアイデアを提示していくことにもっと力を注いで、できないことをもっとできる人を巻き込んで協力していってもらったり、わからないことについて既にわかっている人から知恵を借りたりすることができたと思います。そのような方法を採用の方がもっと良いものが効率良くできただろうと痛感しました。

私は、このプロジェクトを通して、やってみること、失敗してみることが大事だということと学びました。あの時点でああすれば良かったということはたくさんあります。うまくいかなかったなあと思うところもあります。もっとがんばれたんじゃないかということもあります。グループで仕事をする上でもっと上手に協力ができたんじゃないかということもあります。それが今の自分。でもこのプロジェクトを経験する前の自分はどうでしょう？今後悔することも、失敗だったと思うことも、実体験を伴って今の自分を知ること、この様なかたちではなかったかもしれません。先日 NHK の番組で、シリコンバレーでの起業についての特集がありました。そこで、エンジェル（投資家）の一人が投資先を選ぶときに、「同じ能力の人間がいて、一人は起業の失敗を経験した人、もう一人は失敗を経験したことがなかったとしたら、私は十中八九失敗した経験をもった人を選ぶ」と言っていました。私はこのプロジェクトを通してみんなで経験したことはすべて、今後の私次第でプラスに変わっていく材料となると思います。どうせうまくいきこない、とそこにとどまってしまうのは何も始まらなくなってしまいます。始めてみることの大切さを知りました。将来どんな仕事に携っても、仕事とは全然関係のない事柄でも役に立つ貴重な経験だったと感じています。

## 11-2 【大西淳】 as jon

〈〈文責：大西淳

6月からプロジェクトが始まって、いま思えば本当にあっという間でした。これは、それだけ熱心に活動していたからなのか、はたまた実行動として何もしていなかったから印象が少なく短く感じたのか。いろいろ考え方はあると思います。これまでの期間にいろいろなことを学びました。集団で作業することの難しさ、皆と協調してどこまで成果を挙げら

れるか。正直、学校の課題や試験は自分が満足できた地点に立ったとき、そこであきらめがつくかもしれない。でも、集団、いいえ、コミュニティの場合にはその達する地点、山頂が自分より高い場合が多いということ。これは決して山が高いから登りたくないということではなく、「もっと上を行こうよ!」という刺激になって、そこからまた意欲が湧いて、時にはもう辞めたいと思っていた人たちをも引っ張れる原動力を生み出すのだと思う。皆、それぞれが目標を持つところが違うからこそ、こうしてよりよいものへの探求が長く続いたのかもしれない。こういうところにゼミという同じ目的に向かった人たちの良いところがあるのだと思います。

7月頃に班分けがあって、サーバ班いわゆる技術班とむしろ芸術的志向の多くあるべきアプリケーション班とに分かれた。自分はどちらともいえませんでした。技術班ではないことに早くから気が付いていました。すごく消極的な感じですが、このころ ASP のベンチャー企業に入社したのもあったからなのです。その会社で身に付けた技術を発揮できればいいのではという、大変安易な発想でアプリケーション班（後にアプリ班と呼ばれる）に入部したのです。

アプリケーションを探ることから全てが始まりました。つまりネタとして、全世界にどのようなものがあるか。まず世界を知ることになった。世界のサイトを見ることで英語の勉強もできたり、また新たな Java の側面をみることができました。日本は欧米諸国に比べて遅れていると良く耳にしますが、この時それを痛感いたしました。Java に関する記述だけでもこんなにも違う、そうしたら全体数もこんなものではない。まだ、完成は先のことと甘んじていた日々のせいかもしれませんが、こうして定期試験の終わる 8 月頭まではほとんど作業の進まない日々が続きました。ようやく 8 月、この頃になって英語サイトにも慣れ、どんどんネタを探ることができるようになりました。しかし、残念なことに探してきた英語サイトを皆は敬遠してか、まったく読んでくれませんでした。これはその後になって気づいたのですが、誰一人としてその内容について返答がなかったのでもってショックでした。ゼミ生誰も不満をこの頃持っていたと思います、それは私たちのビジョンが見えないこと、方向性にばらつきがある。でもこれは逆にいえば、何も規制のない自由な夢を自分達で、独自に考えられるのだと思いました。既にある何かに向かっていくのではなく、自分の信じた道をただひたすら走ることが必要だったのです。みな勝手に創造することでプロジェクトは大きくなっていくのです。だから、とにかく自分の意見は多くさなければいけない、そうでなければ自分の構想が呑みこまれてしまう、構想に限ったことでは無く、自分までもがゼミに圧倒されてしまうと思ったからです。

9 月の中旬、私はヨーロッパへ 2 週間行きました。ゼミへの反逆と知りながらも、広い視野を持たなければと出国しました。ヨーロッパに滞在する間も 4 日に一回くらいはみんなの顔を思い出しました。また、スイスの景色を見ながらビールを飲んでいる時も日本では同じ時間が流れているのだと実感しました。” Junko ” もこの頃タイに渡っていて、帰ってき

てからこの感情は彼女にもあったことを聞きました。最先端の技術に触れること、世界の先進に行くことがこんなにも辛く思ったことはありませんでした。このヨーロッパへの旅は自分の人生をも変えてくれたような気がします、いろいろなことを考えて、自分が見ていた世界がインターネットの世界というなんて小さな枠の中だけのものだったのだろう。

10月に入りようやく私たちの考えていたこみゅポの形が統一され、いよいよ携帯電話にも着手する時がやってきました。時間が時間だっただけに、毎日試行錯誤で作成しました。また、技術も勉強し、携帯電話の社会的勉強もしました。そして、それによって様々なことが現段階でもできるのだと感心しました。しかし、私たちの作ろうとしているこみゅポに比べれば全く時限の低いものばかりでした。こみゅポは完成できませんでしたが、そして技術的な面でもどのように解決すればいいのかもわかりませんでした。しかし、私たちが目指すものはしっかりと心の中にあります。言葉にするのが難しいので、ゼミ生全員が同じ構想を持っているかということもわかりません。今回の作品がまだほんの途中であること、まだ簡単に完成させてはいけないというような気持ちもあります。サグラダ・ファミリアやディズニールランドと同じような感覚なのでしょう。その反面で、本当に技術が進歩して便利なものが出てきて、ますますデジタル製品に頼った時代に、果たして本当のコミュニケーションが取れるのかということ、そんな人間の寂しさ、暖かさを忘れたくないというのも理由の一つです。

この報告書を作成するにあたり、いままでとは一味違った団体行動の難しさ、印刷の難しさを改めて実感しました。普段なにげなく目にしているものが実はとてつもなくすごいものだったりとか、身近な物も観察しなければ分からないという発見もありました。コピーライトが印刷機の具合によって大きくずれたこと、プリントアウトをインクジェットでやったことも良い思い出として、忘れないようにここに記します。

最後にプロジェクトに携わっていただいた多くの方々、至らぬ点も多くありました、私たちの夢に耳を傾けていただき本当にありがとうございました。

2000/12/1/金

## 11-3【岡田牧男】as macci75

〈〈文責：岡田牧男

私はこのページを書くことになって、はじめてプロジェクトをやったんだという実感を持ちました。6月から約4ヶ月の間、「コミュニティーポータルサイト」というテーマでプロジェクトを進めてきましたが、本当にこれがプロジェクトだったのか、という疑問がついてまわりました。

悩みました。幾度となく迷いました。何度も何度も選択を迫られつづけた4ヶ月でした。このページに関して、「何を書くべきか」ということでまたしても悩んでいます。それは今までやってきたことや議論があまりにも多岐にわたっていて、なおかつ複雑に絡み合っているからだと思います。

私はこのプロジェクトをやって良かったと思います。それはこの機会を通してグループで作業するときに必要なものは、はっきりと感じ取ることができたからです。

私はこのプロジェクトリーダーとして、全般的な統括とこまごまとした事務的手続きと、他のサーバのインストールやアンケート用紙作成、ツール作成などという作業の補佐的役割を担ってきました。全般的にまんべんなく関わっていったので、実質的には自分がこれをやったという形が残っていません。かろうじていくつかツールが残っていますが、本当にそれだけです。こう考えると、自分にとってのプロジェクトは、「実体のない」プロジェクトだったようにも感じました。

私たちが作った一つの大きな成果である「こみゅぼ!」が完成したのが10月の下旬でした。それまでは成果がぜんぜん目に見えてはこなかったもので、大変な作業でした。議論ばかりは先行していましたが、その結果に付随してくる「モノ」がぜんぜん出てこなかったのです。何か一つモノを作るときには、それを一人で作るにしても、何人かのグループで作るにしても、完全に動かなくてもいいから見た目だけでも作りながら、徐々に完成度を高めていく作業をしていかないと、大変であるということを痛感しました。いきなり自分たちが思う完全な形での「モノ」は絶対に一回では出来上がらないからです。自分たちが思い浮かべる理想的なモノは、一発で出来上がるものではなく、いくつかの試行錯誤の末に、「自然に」出来上がるものだと思います。

私はプロジェクトリーダーとして、全般的な統括を担当しました。メンバーのここの作業を把握し、プロジェクト全体の進行のコントロールを担当しました。しかし実際きちんと進行できたかという点、果たしてできたかちょっと自信がありません。「こみゅぼ!」ができたから、それでよしとしてもいいのかもしれませんが、「開発経緯」の項でも述べましたが、「こみゅぼ!」開発にはいくつもの<sup>うよきくせつ</sup>紆余曲折を経してきました。同じ内容が二転三転していったこともありますし、ミーティングが建設的かつ論理的に進行できたかという点、そうでもありません。結局メンバーのみんなが「自由に」意見を出し合って、それだけで終わってしまったということも多々ありました。今ここでプロジェクトを振り返ってみて、自分たちの持つ「理想」が「こみゅぼ!」で満たされているのか、私はちょっと考えます。チーム全体としての意思を決めるミーティングが、単なるディスカッションに終始してしまった、きちんとした進行ができなかったのは統括担当の私として一番の反省点になります。しかもこのミーティングにおいて、はっきりとした意思決定をしていかなかったのが、「こみゅぼ!」開発を多少なりとも遅らせた主たる要因になってしまったようにも思えました。ミーティングとディスカッションをはっきり区別することが、大事ではないかと思いました。

プロジェクトのメンバーとしては相当特徴的な、そしてそれぞれが異なる資質を持つメンバーになったと思います。単純にスキルがあるとかではなく、それぞれの性格や考え方に関してみんなばらばらでした。よくもまあこれだけの違うものだと少し驚いてもいます。このような異なる資質を持つメンバーをどのようにコントロールしていくのか、ばらばらのキャラクターを持つ人たちを、どうお互いに抑制せずに高い次元でのコラボレーションを実現するかということもひとつの課題であったと思います。

価値観や考え方が「違う」ことに関しては決して悪いことではないと思います。問題はどうお互いに受容しあって、その上の「次元」での「融合」をするか、またどれくらい高い次元で「融合」できるか、という点にあったと思います。

私は統括という立場で、敢えて「統制」を行いませんでした。できるだけ個々のアイデアを尊重したかったし、やりたいと思ったことを止めてしまうとそこからの発展が望めないと思ったからです。あくまでも自発性を期待しました。しかしその結果、全体が進行しなくなりました。それははじめからみんなの考え方が違うので、お互いに受容するという作業がなかったからです。プロジェクトの全体という枠を外れて動いてしまったのです。そのため、チーム全体としての意思決定がなかなか取れなくなりました。私はこのことを通して、グループを通して何か一つのことに向かって活動するときには、ある程度の全体的な方向付けは必要であると思いました。

また、コミュニケーションが極端に少なかったようにも思います。私たちが「コミュニティーポータルサイト」を作るというのに、「コミュニケーションが少ない」とはいかがなものかと思いますが、複雑なもろもろの事情から、グループ全体として意思の疎通に困難な事態が多々ありました。特に夏期休暇中のミーティングでは、全員が集まったという機会はほとんどなく、そのうえ問題だったのはそのときの欠席者が重要な役割を担当していたということでした。直接的な担当者不在で意思決定することは不可能であって、結果として無駄に時間を費やすことにつながってしまうこともしばしばありました。あまりにもひどすぎたことがあって、深夜に徹夜で表参道のあるファミリーレストランで話し合いをしたこともありました。グループ全体で動くときの一番大切にすべきところは「協調性」であると思います。しかしこれは決して「無理やりに自分の考えを曲げてまでもあわせる」というわけではなくて、「きちんとした議論の末にお互いが納得した形に合わせる」というのが、本来の形であり、そのためには十分なコミュニケーションが必要だったと思いました。これは今後にも生かされる重要な点であると思うし、また何かグループを作って活動するときには絶対欠かすことのできない要素であると思います。

同時に私は意思の疎通をすることの難しさを感じました。同じ人間で、同じ国に生まれてきた人間であるけれども、それでもこれだけ考え方や価値観が違うのです。同じ言語体系を

持っているとはいえども、同じ単語や文法が話の相手によって解釈の仕方が違うんだ、ということ痛切に思いました。自分のアイディアをいかに相手に一番伝わりやすい方法で伝えるか、そしてそのときにいかに自分のアイディアが「意図していない方向に解釈されないように」相手に伝えるか、この2点が重要だと感じました。

プロジェクト全体を通して、私が思ったことは、「全体」として動くことがどれだけ難しいかということです。たかだか10人という小さなグループでしたが、それでもそれぞれの持つ考え方や価値観の違い、「こみゅぼ!」に対してのイメージや夢、理想の違いに対して、どのように「チームとしての」答えを出していくか、それが本当に大変な作業だったように思います。もちろん互いに「抑制」してしまえば、個人が生きてこなくなるので意味がないのです。いかにお互いを「殺さず」、全体を形作るのかが大変でした。別にひとつの「こみゅぼ!」を作る必要はなかったのかもしれませんが、しかしそれでも、私たちが「コミュニティー」を作っている以上、「私たち=チーム」の必要なものを作り上げる必要があったのではないかと、考えています。

私は今回のプロジェクトを通して、さまざまな課題や教訓を残していったと思います。しかし一番私が思ったのは、その教訓をいかに今後に生かしていくかだと思います。私はそのような学びの機会を得たことに対して感謝したいと思います。

## 11-4 【姜秀珍】 as sussy

〈〈文責：姜秀珍

プロジェクトの期間中は、あまりにも忙しくてじたばたしていたため、青山祭の打ち上げと共にこの約6ヶ月間の起こった事柄をまるで記憶喪失のように一気に忘れてしまいました。記憶喪失のおかげで(?) プロジェクトメンバーではなく、去年の私の普通の生活を2週間ぐらい満喫していたところ、災難(?) はもう再び訪れました。それは、12月に行われる報告会と報告会にお渡しする報告書の準備です。報告書の原稿を書くために、この6ヶ月間たまった、1000通以上のゼミのメールを読み返してみました。こういうこともあったのか!!と思わず感動です。サーバ構築記を書きながらみんなで揉めていたことや、精神的にダウンしていたこと、合宿とジョナサンでの徹夜会などなど今から思い出せば、なんか笑ってしまいます。(たぶん今だからの話でしょうが) 報告書を書きながら思い出した、プロジェクトの中での出来事、自分がこの6ヶ月間になにに挑んできたのかを記してみたいと思います。

〈公開ソース探し〉

当初、私はHardware関係の仕事にはあまり関わりたくないと思っていました。最新技術の

リサーチを行って、その技術をどのように私たちが考えるコミュニティーポータルに取り入れたら良いのか。を考えた上で、アイデアを出す、企画寄りの仕事をしてみたいと思っていたので、土台班-アプリ班に分かれたときには、迷い無くアプリ班の方で活動することにしました。アプリ班は、現在インターネット上にフリーで公開されているソースを探し出し、それを自分たちが使えるようにカスタマイズして使うという原則を基に進めてきていたので、デジタルキャッツのサイト [www.javacats.com](http://www.javacats.com) や、IBM の alphaworks などでスケジュール管理のソース、掲示板のソース、GuestBook の Java ソースを入手することが主な仕事でした。公開されているソースは、ほとんどが class ファイルだけだったり、または jar で圧縮されていて解凍の方法もわからなかったのが、戸惑っていました。そうした中、逆コンパイラーの存在や IBM の Visual Age や Borland の Jbuilder 等の Java 開発アプリケーションの存在にも気づき、プログラミングは、必ずテキストエディターで行うものではないことを発見しました。インストールはしたものの、このプロジェクトの中でプログラミングには関わっていなかったのが、実際このようなツールを活用するまでには至りませんでした。ちょっと残念です。

#### <モバイル機器への対応>

私たちの考えるコミュニティーポータルは、Anytime、Anywhere、with any device の考え方に基づいていたので、対応するデバイスがパソコンだけであったら、移動中には、使えませんということで、モバイル機器への対応の案が登場しました。当時、インターン先で KVM を勉強していた私は、是非フリーで公開の java ソースを入手し、それを PDA や携帯で動かせるように J2ME\_CLDC、J2ME\_MIDP を使ってソースを修正するかたちで、PDA、I-mode 対応の Java アプリケーションを作りたいと思いました。しかし、公開されているソースを修正するのは、思ったより難しかったので、この話は流してしまいました。

#### <井田文庫の本リストアップ>

最初、私たちが考えたコミュニティーポータルにおけるツールの中の一つとしてあげられたのが、井田文庫 (LIDA) です。一番、始めに cafeuro がサーブレットの勉強の成果を出したのも井田文庫です。このツールを完成させるためには、もちろん先生の研究室の本をリストアップしてそれを DB 化することが必要不可欠です。2 日間の作業で時間的には短い作業でしたが、ritz との本に関する説明項目が一致していなくて、やり直し?の羽目に落ちたりもしました。私にとってはプロジェクトの中で一番最初にした仕事なのでごく簡単な単純な作業にもトラブルは必ず起こりえるということを見据えた上でのスケジューリングが必ず問われるのだ。と大変良い教訓になったと感心です。

#### <サーバーセットアップ及び運用>

サーバーセットアップに関しては、本人も驚きです。時間的にみんな迫っていたので、インターンをしている私は、インターン先で早くインストールを終えてしまおうと思っていました。インストールという役割は、本当にサーバソフトのインストールだけだと信じていましたが、その役割には運営も含まれていたのが、急にサーバ班に変身してしまいまし



た。墓穴を掘ったよ…

サーバに関しての知識が全く 0 状態で、(実際、サーバという言葉は今年の 5 月にはじめて知りました。) JAVAPRESS や月刊 JavaWorld のサーバ構築に関する特集を読みながら勉強をしましたが、それでも初心者にとってはわかり難かったのです。結局、五十嵐さんに助けを求めました。五十嵐さんは Servlet の Specification (仕様書) をみんなにコピーしてサーバレットって一体なんだろうっていう概念を教えたあと、全面的にサーバ構築を手伝ってくれました。本当に地獄から救ってくださったようなものですね。ありがとうございます。なににせよ製品にはマニュアルっていうものがついているということは、常識ですが、Tomcat、Apache のようなソフトや Servlet や JSP という技術にも User's Guide や仕様書があり、まずそれを理解しなければならない。英語のサイトを毛嫌いして最後の最後まで日本語で訳されたサイトを探し回っていましたが、やはり作った人たちが記している文書そのものが一番信頼性が高かったです。最後になって、私はようやく日本語訳をあきらめて、英語の文書を読んでいました。もっと早くわかってたらな…

コミュニティーポータルサーバがインターネット上に公開されなければ、I-mode のページも WebTouch を展示することも無意味になってしまいます。それ故、学校所有の空いている IP を借用することにして、公開してもらいました。公開後には、それ程大きい問題は起こりませんでした。サーバ公開の数日後に繋がったり繋がらなかったりの異常が一回だけありました。このトラブルは、マシーンを再起動するだけで解決できたのでサーバ運営に関しては、思ったよりスムーズにいきました。自分の希望としては、サーバのセキュリティの面をもっと強化していきたいというのがあります。

#### <パネル作り>

パネルづくり…これは実を言うと青山祭の 3 日まえから考え出しました。ディスプレイについてや、レンタル、またツール作りなど他の仕事でみんな忙しかったため、意外と肝心なこみゅポの裏の仕組みを披露してくれるパネルブースについては、あまり気を配れなかったようです。わたしは、1 週間ほどまえにサーバのセットアップを終えていたので、ちょうど余裕がでるようなでないような状態でしたので、パネル規格を作りことになりません。時間的にすごく迫っていたので青山祭 2 日前はパネル作りの人々は、徹夜しっぱなしです。原案を書いてこれでどうかな? というメールが出すと、3 分以内に返事がきたり、返事が来なかったら、“〇〇ちゃん、寝ちゃったのか。裏切られたよ～～” などと嘆きながらまだ起きているひとたちと電話もしながら結構たのしい(?) 徹夜でした。あ、そういえば、徹夜の晩に、日本のサッカー試合があって、それを見てたためどうしようもなく徹夜のひともしましたね。

#### <最後に>

日本の大学で日本人だけのメンバーの中でプロジェクトを経験した。これは、留学生の私にとって他ではできないとても貴重な経験をしたと思いました。よく日本人は何事にせよ注意深く、韓国人は大雑把? という言葉を耳にします。大雑把すぎるわたし、みんなと共

にプロジェクトをしていて、迷惑もいっぱいかけたと思いますが、周りに対する自分の対応の仕方を見ながら自分が見えてきたような気がしてとても面白かったです。このプロジェクトを大学生活の中の大切な思い出の一つとして取っておきたいものです。みんな、お疲れさまでした。

## 11-5【黒田洋史】as caferuro

〈〈文責：黒田洋史

いま、報告書の原稿を書き終え、散歩に出る。冬の風が冷たい。もう冬になったのかと思う。僕たちのプロジェクトが始まったのはまだ梅雨の終わりごろだった。まずはじめに先生からもらったのは「コミュニティーポータル」という言葉だった。コミュニティーポータル？という、なんてミスマッチな組み合わせだろうかと思った。でも、今はすごく自然な言葉になっている。同時にコミュニティーポータルの一つとして「こみゅぽ」が目の見える形であり多くの人に使ってもらった。そう考えると、とても不思議な気がしてしまう。

でも、正直言ってプロジェクトをやっている最中、自分がやっていることの一つ一つが本当にコミュニティーポータルにつながっていくのか不安で不安でたまらなかった。なんか、霧の中何も見えない中を徘徊している感覚を何度となく感じた。でも、自分を信じて進んでいくことしか自分には残されていなかったと思い押し進んだ。あのいろいろと悩む格闘している時、こういうある程度完成した形でプロジェクトを終えることができるのは正直言ってぜんぜん思っていなかったし、逆に青山祭できちんとした展示をお見せできないという夢をいやというほど見てしまっていた。そんなときから考えると今のできはミラクル(奇跡)なのではないかと思う。

どうして、こんなミラクルが起こってくれたのかと思考を巡らす。大きな理由が10人がいい意味でばらばらであったことではないかと思う。一人一人が自分のしたいことを持ち、それがいい意味でばらばらでお互いに専門家になれたのである。例えば、サーバの事だったら sussy、交渉なら moriken、携帯電話なら jon、アンケートなら ritz、青山祭なら arrow、デザインなら aqua などのように自分達の専門を決めて、それに没頭することができた。この専門分野の決定の多くは自分達が実践していく中で見つけ、責任を持つようになりそれが自然に専門になっていったのである。専門を持つことですごくいいものをそれぞれの人が作り持ち寄りすばらしい結果をもたらしたのではないかと思う。

同時に、いろんな人がいたからよく意見がぶつかった。みんな、同じことについて話しているのに考えていることが本当に違うんだなあと感じたことが多々あった。でも、僕自身

人と意見をぶつけるのを避けやすい性格だと思っているのだが、プロジェクトでは自分の思っていることを存分にみんなにぶつけた。それは、ちゃんと自分の持っていることに責任を持ちたいし第一いいものを作りたいと思うからである。いいものを作りたいという気持ちは誰一人変わらないのでみんな意見を投げる。時々、大きな火がつくことがある。それも恐れず投げる。そうすることで、みんなで共有できる一つの答えが見えてきたのではないかと思う。

そうやってみんな一人一人が作業を進め、みんなであった時にそれを見せ合い意見をぶつけあったり軌道修正したり先生や五十嵐さんから助言ももらったりした。そうして一つのものを作っていくことができたのである。

一方、一人一人が夢に実現を真剣に考えることができたからよいものができたのではないかと思う。みんなの一つの概念を出した後それぞれがそれを実現させようと真剣にあきらめずに取り組んだのだ。ぼくは、Java という技術を学び、それを生かすことで、みんなで考えた夢を何とか実現できないかと思い取り組んできました。そのなかで多くのことを学んだ。一番大きかったのは「技術は夢の実現の為にある」ということである。今、技術ばかりに注目を集まっているが技術は夢を実現する為の一つのツールでしかなく実際、重要なのは何を実現するかななどの人の頭で考えることであるということがわかった。

企業に入れば、こうしてみんなでぶつかり合い物を作っていくのは普通のことなのかもしれない。しかし、学生の身分で最初から概念を提示し、それに必要な技術を勉強し概念を形にしそれを展示する。こんな経験をできるチャンスは多くない。私たちのこの経験を多くの人に共感してもらいたいとこの報告書を作りました。でも、僕たちのやったこと考えたこと悩んだことはこの本には入りきれないほどあります。その経験をこれから多くの場で活用していきたいと思います。このプロジェクトを支えてくださった、井田先生、五十嵐さん、雪乃さん、協力してくださった企業の皆様本当にありがとうございました。そして、一緒にやってきた4期生のみんなお疲れ！！そしてありがとう。

追伸：僕のコみゅポへの理想はとどまるところはありません……。

## 11-6 【清水淳】 as aqua

＜＜文責：清水淳

### はじめに

プロジェクトを終えての感想は「大変だった……」の一言に尽きます。授業で少し触れただけの Java、ほんの数ヶ月前に使い始めたばかりの Linux、全くの未知の領域だったサーバの構築、気心を知らない仲間との共同作業、何もかもが中途半端な状態で始まったのが

今回のプロジェクトでした。

### ちょっと気になること

Web サイト製作っていうとカッコいいけど、ちょっと硬い。ホームページを作るという  
と少し幼稚っぽいですよ。友達から、ゼミで何やってるの、と聞かれるたびにモドカシ  
イ思いをしました。「ホームページ作ってる」って答えると、帰ってくる反応は「そうなん  
だー」なんです。その後「Java プログラミングやってる人もいるし、サーバ構築してる  
人もいるよ。i-mode のコンテンツを作ってる人もいる」と付け加えると、「そんなことやっ  
てるの。すごいね!!」とさっきと反応が違います。「ホームページ」というとどうしても  
軽く見られてしまいます。「何でそんなに反応が違うの?!」と大声で叫びたいくらいでした。

みんなにしてみれば、「ホームページを作る」というのは遊び感覚なんですね。「ホー  
ムページ」作成に関するソフトウェアや本が氾濫して、誰にでも簡単に作れる環境がある  
のだから仕方ないんですけど。そんなこと言ってる僕も実はそう思っていました。でも、  
ちゃんと勉強してみると実は難しんです。報告書に書いてあることはほんの一部にすぎな  
くて、本当はもっともっと奥が深かったりします。

何が言いたいかというと、Java プログラミングする人も、サーバを構築する人も、i-mode  
のコンテンツを作る人も、アンケートを作る人も、誰かと交渉する人も、レイアウトをす  
る人も、飾り付けをする人も、プレゼンテーションをする人も、みんな大変さは同じだと  
いうことです。比重の問題はあるけど、それぞれが大変な思いをしてやってきたというこ  
とです。

### 少し話はそれて・・・

「ホームページ」が持つ本来の意味は、みんなの思っている意味とは少し違います。こ  
の文章で「ホームページ」という言葉に「」（カッコ）がついているのもそれと区別するた  
めです。身近な例でいえば、「情けは人のためならず」がそうかもしれません。「情け」は  
「人のため」になるのかならないのか。今ではどちらでもいいというような風潮になっ  
てきているみたいです。でも、正しい使われ方というのがあって、時代の流れとともに間違  
った使われ方も正しいとされるようになってきました。まさに言葉は生き物です。

絶対に間違ってるとか正しいとかを言うことはできません。どっちでもいいじゃん、と  
いう人もいます。僕もどっちでもいいと思います。ただ、その分野を学んでいるのなら、  
その言葉がもつ本当の意味は知っておきたいし、知っておいてほしいと思います。

### Java について少し

Java プログラミングに関しては、ほんとに少しかじっただけ。色々なプログラミング言  
語はあるけど、Java もそのうちのひとつなんだな、くらいでした。「くらす?、めそっど?  
、おぶじえくと指向?」。もう、全てに対して「?」マークです。それでも、本を買ってきて

はせっせとソースを書き写して、動かしてみても、「おっ、動いた。すげえ!!」なんて一人で感動していました。でも、動くのは当たり前なんですね。やっぱり自分で考えてプログラミングしたいと思うようになりました。限られた知識の中で何を生み出せるのか、全く先行きが見えなかったけれど、とりあえずやってみないことには何も始まらないのでエンジンだけはかけておきました。まるで、真冬に暖めるだけにかけてエンジンのような状態でしたけど。

Java プログラミングを続けるうちに、思ったことが1つあります。Java って起動するのにやたらと時間がかかるんですね。ジャストシステムのプロ集団が作った Java アプリケーションの「一太郎 Ark」を試しに使ってみたときは、巨大なプログラムだから仕方ないのかなと思っていました。しかし、僕たちが作ってるほんとに小さなプログラムでもやたらと時間がかかるんです。「いくらプラットフォームを選ばないと言っても、これはちょっと…」なんて思ったりもしました。確かにどのコンピュータでも動くというのはすごいことなんだと思います。でも、アプリケーションを使っている側からみれば、どのプラットフォームでも動作するというのはあんまり気にならないことだと思うんです。2つの OS を同じように使っている人の割合はそんなに多くないと思います。実際に、僕の周りではほんの数人しかいません。その人たちだってコンピュータに熟知している人たちです。ものすごい大暴な言い方をすれば、プラットフォームを選ばないというのはプログラマーの理想であって、ユーザーの理想ではないような気がします（プログラマーのみなさん、ごめんなさい!!）。Windows ユーザーの僕としては、「Macintosh では動かなくてもいいから Windows ではだけは完璧に動いてくれ」というのが切実な願いです。別に Java を否定しているわけではありません。僕も Java にはとても興味があります。動作が遅いという問題も将来はきっと解決することだと思います。ただ、素人目から見たら、そういうところが気になったというだけの話です。

### 共同作業って大変だ・・・

今さらながら、共同作業って難しいです。約半年間のプロジェクトを終えて改めて思いました。自分も含めて、人間のイヤな部分を見てしまいました。もちろん、イイ部分もですけど。これ以上言うとボロが出そうなのでやめておきましょう。

全然関係ないんですけど、多分プロジェクトが始まったばかりの頃の僕に対するみんなの印象って、「この人、大丈夫かな？」だったんじゃないかと思います。金髪だし、うるさいし、モノワカリが悪いし、ちょっとヒネクレ者だし・・・。一言でいえば、トラブルメーカーですね。僕のみみんなに対する印象は全く反対で、「こいつら、すげえなあ」です。それは今でもあまり変わりません。

色々な人が集まって、1つのことを成し遂げるってとても難しいけど、それ以上に面白かったです。みんなは僕にないものをたくさん持っていて、たくさんのことを僕に与え

てくれました。

## おわりに

このプロジェクト期間、良いことも悪いこともたくさんありました。感動させられたことも、がっかりさせられたこと、嬉しかったこと、悲しかったこともたくさんありました。全部ひっくるめて、いい思い出です。みんなには感謝してもしきれません。みんなに会えて本当に幸せです。ほんとにありがとう!!

## 11-7【土橋りつこ】as ritz

＜＜文責：土橋りつこ

### 1・このプロジェクトで私がしたこと

- ・ 勉強
- ・ アンケート集計
- ・ パネルづくり
- ・ パンフレットづくり
- ・ 井田文庫 (LIDA) 情報集め
- ・ 買出し
- ・ 教室の装飾

### 2. 感想

まず、私が最初にプロジェクトをやる上で考えたのが、他の人より知識も実践経験もない自分がどうやってこのプロジェクトに携わっていくかということでした。実際、このプロジェクトでコミュニティーポータルサイトを作るという具体的な話が出たときも、ポータルって何??? どんなものをつくらたいの??? どうやったらつくれるの??? とハテナマークだらけで自分が何をしたいのか、なにをすればいいのかがなかなか見えてこない状態でした。だから毎回毎回のミーティングもとても辛くて、なんで私がここにいるんだろう...? と思っていたことも多々ありました。しかし、なににもできないから何もしなくてもいいなんて思っはられないし、やるからには楽しみながらやりたかったので、自分なりに勉強してみたり、とりあえず自分の出来るところから頑張ってみよう! と思いました。技術的なことは好きな人に任せてみて私は何もわからないなりの立場で客観的にこのプロジェクトを見、純粋な視点でアイデアを出せばいいな一と思い始めたのです。

しかし、実際サイトの形ができるまでは本当にどんなものを作ろうとしてるのが見えてきませんでした。しかし、段々とサイトのツールやデザインが具体的に決まってくると作ろうとしているものがはっきりと見えてきて、凄いいものをつくっているんだという意識が出てきました。そんな中で私はアンケートの集計、井田文庫の情報集め、パンフレット作

成などに携わりました。アンケートはいかに簡潔で答えやすいものにするか考えるのに苦労したし、パンフレットはいかに来てもらった人たちに自分たちがやっていることを分かりやすく説明するかでかなり悩みました。

沢山のミーティングを通して学んだことも多々ありました。みんな本当に自分の意見をしっかり持っていて、それを言葉に表すのがうまくて、自分の思っていることをなかなか言葉にできない私にとってはとても勉強になりました。そしてなによりも難しいのは皆で意見をたたかわせて一つのもので作り上げていくということだと思いました。社会に出たときに必ずこういう経験が自分にとって役立つ貴重な経験ができたと本当に思っています。

このプロジェクトは精神的にみんなとても大変だったと思うけど、最後まで頑張って成し遂げたみんなを見ていて、このメンバーでこのプロジェクトをやれて本当によかったなと思っています。いろんなことを学ばせてくれたみんなに感謝しています。

そして、最後にお世話になった五十嵐さん、雪乃さん、そして先生、本当にありがとうございました。

## 11-8 【森健蔵】 as moriken

〈文責：森健蔵

プロジェクトが始まって6ヶ月。色々な事がありました。今思い返せばどれもいい思い出です(!?)

このプロジェクトで私がやったことは大きく分けて2つ。1つはツールの一部である青山周辺情報、授業評価の企画、設計。もう1つは機材を借りるための企業への訪問、及びそれらに関する雑務一般です。それぞれの仕事から色々な事を感じ、学びました。

ツール作りも一生懸命にやりましたが、私自身、より印象的で楽しみながら活動できたのが企業訪問でした。ですから企業訪問を通して感じたこと、学んだこと、更には生意気にも後に続く後輩たちへアドバイスをここに記しておきます。

### 企業訪問

合宿が終わり皆のなかに焦りが出始めた頃、ようやく各自の分担がハッキリしてきました。その中で私が担当したのが機材を提供していただくための企業訪問でした。私がこのゼミに入った理由は扱っている内容が今の時代にとっても合っていると言う事もありましたが、3年次のプロジェクトで企業を訪問し、色々とお話を伺えるチャンスあると聞いていたからでした。

ですからその役を任された時はとても嬉しかったです。また、同時に、機材が無ければ青山祭で展示できないという責任も同じくらいに感じていました。

まずは何を借りるのかをメンバーの間で洗い出し、その後、直ぐに Web 上で手当たり次第に色々な企業にメールを出しました。もちろんどの企業にも知り合いなどいませんでしたし、どこのどういう部署にメールを出したらいいか全く分からなかったのも、とりあえず、“お客様相談室”などこちらから出したメールを読んでくれそうな所へは一通り出しました。今、思えばかなり無謀で計画性の無い事をやっていたのだと思います。メンバーには“そんなところに出しても無駄でしょ。”などと言われましたが、企業の方はやさしかった……。全部とは言いませんがその幾つからはちゃんとした返事が帰ってきました。(断りのメールも含めて。)

たまには良い事っていうものも続いて起きるものです。PC 提供を前提に私たちのプロジェクトについてのお話を伺ってくださると言う企業があったその次の日に、井田教授のほうから同じ会社の違う方の名刺を頂きました。(私の努力って……) 何はともあれ、まずは面会の約束が取れ、いざ企業訪問へ。

企業訪問の初日、cafekuro と二人で一応、作戦会議のようなことをしました。挨拶の仕方、名刺の渡し方、どの点を一番伝えたいか、もし何か質問されたら、どのように答えるなどなど。やはり人生において企業を訪問するという事は初めての経験で、不安と期待でいっぱいでした。そして企業に伺うとまずは受付の綺麗な受付の女性の方が笑顔で迎えてくれました。何となく緊張もほぐれたと思いきや、通されたのはワンフロア-全てが商談室になっていて、周りはビジネスマンたちが真剣に交渉しているではありませんか。もう私の緊張はピークに達していました。そしてお約束のように名刺の交換。テレビでは何度も見ていた光景ですが、自分でやるとなるとチョッとぎこちなかったかもしれません。

名刺の交換も順調に済ませ、本題に入りました。cafekuro との予定ではまず最初に私がプロジェクト全般の話をして、次にシステム関係のことを彼が話す予定でした。ところが何を思ったか、いきなり彼が話し始めてしまうではないですか。これにはかなり動揺してしまいました。(おいおい話が違うよ!! 後で聞いたら彼も相当緊張していたようでついつい話してしまったようで)。それでも何とか話を進めていきました。他人が話していることは、ここはこういったほうが良い、良くない、など冷静に聞けるのですが、自分の言っていることは冷静には聞けません。ちょっと気を落ち着かせようと思い、さりげなく、先方の方に出していただいたアイスコーヒーを飲もうと思うと、ストローがcafekuro の肘の下にあるではないですか。ああ近くて、遠いアイスコーヒー!! 必死にストローを彼の肘の下からひっぱってもなかなか抜けず、そのことには彼は話していて気付きません。あまりに強くひっぱっていたので企業の方がそれに気付き“どうぞ、お飲みください”の優しい一言。かなりはずかしかったです。冷たいものも頂き、何となく落ち着きを取り戻し、



その後は何事も無く、私たちが言いたい事は全部、話せました。その後、企業の方々から厳しいながらも、的を得た意見が帰ってきました。“このプロジェクトは少し、的が大きすぎない。もっと絞ったほうがいいよ。時間的にもかなりきついんじゃないの。” 私たちの中では自覚していた事なのですが、やはり第三者に指摘されるとかなりショックでした。

帰り道、機材は貸していただけることになり嬉しい反面、その指摘された事に二人でブルーになって学校に戻りました。せっかくみんなの士気が高まっていたところに水を差すようなことは言いたくなかったのですが、包み隠さず他のメンバーにも同じように伝えました。なせならその理由は企業訪問とはただ機材を借りに行くだけでなく企業の方々から意見を聞ける貴重なチャンスであり、やはりそれをプロジェクトに生かさなければいけないという事だと思ったからです。家に帰ってきて、自分なりにその日の反省をしてみました。(その反省が次の企業訪問に活かされたか否かは、定かではないですが)

ある企業の方にはわざわざ本校まで足を運んでいただきました。私たちの力不足で学校の施設が借りられずに、学食でお話を伺うことになってしまいました。(本当にすみませんでした。) その時にお話した内容は WebTouch を使うために一体何が必要であるかが中心でした。事前の私たちの調べでは WebTouch を LAN につなげるには PBX なるものが必要があるということだったので私たちは PBX をお借りしたい旨を伝えると、実際に必要なものは PBX ではなく RAS とシミュレータであり、その主な機能までいろいろと教えていただきました。本当に勉強になりました。

その後も企業訪問は順調に進んでいきました。(そう思っているのは私だけ?) 時にはプロジェクトの活動に疲れてしまい、訪問先の近くの公園で、ベンチに座り込んで日向ぼっこをしていたこともありました。追い込み時期だというのに、今までの活動の思い出を他のメンバーと懐かしく語ってしまい、こんなことをやってる場合じゃないと気合を入れ直した覚えもあります。

また訪問先ではプロジェクトの話を済ますとまったく関係無いのこと、たとえば就職のこと学校生活のことなどをプロジェクトの内容と同じほど、もしかしたらそれ以上話していたかもしれません。(お話好きの性格がもろに出てしまった一面です。)

最後に私が企業訪問を通してこれから続く後輩たちに企業訪問をする際のアドバイスを一つしたいと思います (こんな私の意見がアドバイスと呼べるかは疑問ですが)。それは自分たちの夢を“これでもか”というほど語ることです。企業の方々は普段から第一線で働いている方で、日夜いろいろな方々から新しい話を聞いている方です。そのような方々にただ平凡とプロジェクトのことを語っても何の反応も示してくれません。自分たちの理想を高く掲げ自分たちのプロジェクトがいかに素晴らしいかを話すことです。ほんの少しぐらい自分たちが作っている物と話すことが違ってかまわないのでしょうか。ハツタリを

かますぐらいがちょうどいいのです（根底の部分が変わっていなければいくら装飾して話しても同じ物）。企業の方は学生が少しくらい変なことや、間違っただけをいっても驚きもせず、真剣に耳を傾けてくださいます。むしろ学生の視点から見た新鮮な考え、夢に興味を示してくれます。ぜひ、頭の隅にでもこのアドバイスを残しておき、企業訪問がんばってください。

今回のプロジェクトで企業訪問の担当につけたのは本当によかったと思います。自分の強みを十分にこのプロジェクトに生かせたと思います。人とコミュニケーションする能力。それが私の強みではないでしょうか。先生はこのプロジェクトが始まる時にこうおっしゃいました。“自分にしかできないことでプロジェクトに貢献しなさい。みんなはそれぞれが違う能力をもっているんだから。それを早く見つけることだよ。”プロジェクトの初期、正直言って私しかできないことが全く分かりませんでした。こんな私がこのプロジェクトに貢献できるのだろうか。しかし、企業訪問を通じて自分にしかできないことが明確になっていきました。

今は、プロジェクトをやり終えた満足感でいっぱいです。このプロジェクトを通して学んだ、自分の強みを今後の就職、更には人生に大きく役立てていきたいと思っています。

## 11-9【矢島達也】as arrow

〈文責：矢島達也

6月中旬から8月下旬まで

6月中旬（だったと思う）にプロジェクトが開始しました。プロジェクトをする上で考えたことは、「何でもチャレンジしてみる」ことでした。知らないことばかりだったので、何でも吸収してやろうという気持ちで望みました。

最初の班分けとしてサーバ半とアプリケーション班に分かれたのですが、自分が選んだのはサーバ班でした。その理由はサーバのことやネットワークなど裏方の仕事に興味があったからです。またその当時、ネットワークやサーバ関連の会社にインターンとして行く可能性もあったので、そのための勉強としての意味合いも大きかったです。（インターンは結局落選しました、トホホ…）

また、実務的な分担として会計係に就任しました。会計係として考えたことは、自分はただの金庫番にならず、会計の立場からプロジェクトを引っ張る、あるいはフォローをすることが大切だ、ということです。そのために、班別ミーティング費と資料費に予算の大半を割くこととしました。「支出が増える＝活動が活発化する」ということを実

践しようとした結果です。そして、会計規則を作り、お金の管理をきちんとし(会計規則の内容は付録をご覧ください)、適切なお金の流れを目指しました。しかし、班別ミーティングはあまり開かれず、資料費も一部の人だけに使われただけになって、結果として、費用の大半は青山祭の準備費用に消えてしまい、当初の目的とは違った方向に事は進んでしまいました。各自が会計規則を見ていなかったことが一因となっていたのかもしれないです。

サーバ班としてまず最初に取り組んだことは、自分がサーバやネットワークの本当の初心者であったので、サーバの基礎の基礎の学習でした。その勉強の過程で Servlet の存在を知り、win 版の JSWDK のインストールをやってみました。ダウンロードはすぐに終わり、実際に使おうと CLASSPATH の設定も順調に終わったのですが、動かない。cafekuro の助け(どうやったのかは覚えてない)を得て、(DNS サーバが動かなかったが)ようやく実際に使うことができるようになりました。20 分の予定が 2 時間もかかってしまったのでした。世の中そんなにうまくいくもんじゃない。

7 月の下旬から Java の勉強にも取り組みました。今回のプロジェクトで Java は必須の技術であると思っていたことと、他のメンバーが既に Java の学習を去年しており、自分が遅れていて何とか追い付きたかったことがその理由です。始めのうちは苦痛だったのですが、慣れるに従って学習、特に何かを作ることに楽しみを覚えるようになりました。まるでプラモデルを作っているような気持ちで作っていました。結果として生みの苦しみと作ることの楽しみの両方を経験できたと思います。

## ○夏の合宿

合宿のプレゼンテーションにおいて私は Servlet の概念の説明をしました。その時の写真を見て、自分が予備校の講師のように熱弁を振るっていたことを初めて知りました。まさか自分がこんなに熱血していたとはビックリです。

今回の合宿は、自分たちの理念を再確認する場として最適であったと思います。自分たちが本当にしたいことは何かという方向性を導くよいきっかけにもなったと思います。この合宿中、徹夜で会議をしましたが、徹夜に強い人は少なかったです。(ちなみに私は徹夜に強いのです。)

## ○9月～青山祭

合宿後、ジョナサンにて徹夜ミーティングがあり、その場で企画書の作成、そして分担を再び行いました。私は「個人/グループスケジュール管理ツール」の作成に転向することとなりました。サーバに関してはもっと知っている人もいるし、時間もなかったためです。そして、Java で何かを作りたいという欲求もそこにはありました。

Java を勉強した期間があまりにも短く、しかも複雑な機能をもつプログラムの経験がなかったので、本当にプログラミングできるのか不安でした。そこで、書店に何かサンプルがないか探しにいったところ、Swing についての本があり、中身を見てみると、そこにはスケジュール管理ツールに必要だと思われるノウハウがたくさんかかっているではないか！ということで安易に Swing に手を出し、作成に入ることとなりました。

3 週間の苦労の後に個人用のスケジュール管理ツールは完成しました。Swing は、自分にとって実装をする過程が分かりやすかったこともあり (最初に枠を作っておいて、そこに機能をパズルのように組み込んでいく)、学習はすんなりとはいったのですが、エラーが何十個も出てくると、もう何処が違うのかが分からなくなり、もう一度作り直してみたことが何度かあって、産みの苦しみを味わいました。

ツールはとりあえず完成しましたが、今回のプロジェクトでは、双方向的 (インタラクティブ性) と動的 (ダイナミック) なサイトを目指すことになり、自分が作った Java アプリケーションがその方向性に反するものであること、そしてコンピュータを起動してサイトにつなぐたびにアプリケーションをダウンロードしなければならないということがあって (アプリケーションの起動に時間がかかる)、このアプリケーションをサイトに使うことを断念しなければならなくなりました。そこで次に取り組んだのは 2、3ヶ月前に直に触れた Servlet でした。CGI ではなく Servlet を使用した理由は、Servlet が慣れ親しんだ Java で書かれていること、そして、CGI は使うプログラミング言語がややこしいと聞いたためです。

Servlet に関する書物は現在のところ一冊しか出ておらず、しかもその本は Java をマスターした人向けの本であると感じました。要は読んでまったく理論が分からなかったのです。その中でしたことはサンプルを真似することでした。真似をすることで理論を理解することが少しであったが出来、そのサンプルの、スケジュール管理ツール作成への応用の方法を模索することも出来ました。

自分である程度のところまではいったのですが、あるところで行き詰まってしまいました。ある日、入力内容が反映されなくなってしまったのです。そこで、五十嵐さんに助けを乞うこととしました。結局簡単な HTML エラーが原因でした。その後もすぐ近くに迫った締め切り日に向かって自分なりに頑張りましたが、グループスケジュール管理ツールの実装があまりにも難しく、終わらないうちに cafekuro に託しました。cafekuro はきちんと実装をしてくれたので、とても感謝したい気分だった。

ツール作成と並行して青山祭出展会場のレイアウトを考えました。考える際に留意したことは、以下の 3 点です。①プロジェクトのコンセプトを具現化させること。②プレゼンテーションと、一般の人々に実際に使ってもらって率直な感想を聞くことに一番の

重きを置く。③出入り口が一箇所しかないので、如何にスムーズに人の流れをつくるか。  
(実際のレイアウトは青山祭のセクションをご覧ください)

青山祭当日は思った以上にたくさんの方が来てくれて、説明する方としてもうれしかったです。接客をしていて感じたことは、学生の人、特にサークルの幹事をしている人は、「すぐにでも使いたい」、とサークルでこのシステムを使うことに興味を持ってくれたこと、そして、社会人の方が「これをビジネスにしないのか」とビジネスの話をしていました。立場の違いが発言の違いを生んでいることを感じました。

### ○全体を通しての感想

今回のプロジェクトを通して感じたのは、理想を現実とすることの難しさでした。私たちのサイト作りもそうですが、自分の携わった仕事に特にそのことを感じました。会計の場合、結果として赤字は出なかったのですが、自分で考えたお金の流れとは全く違った方向へお金は流れていってしまいました。青山祭の場合、自分の頭の中で考えたレイアウトのイメージを忠実に再現させることが如何に難しいかがよく分かりました。例えば、家の中ブースを作る際、本当の部屋の雰囲気を出そうと考えていましたが、実際は、失敗してしまい、“WebTouch One”の機能のアピールの場のみになってしまいました。ブースの意味がなくなってしまったのです。

もう一つ思ったことは、「各自が担当の仕事をきちんとこなさなければ、他人にその分の負担がいく」ことでした。私の場合、ツールを自分で全部作ることが出来ていれば、負担をcafekuroにかけることもなかったのではないかとずっと思っています。時間的な問題もあってしょうがないこともありましたが、やはり自分の仕事はきちんとこなしたかったという気持ちは今でも持ちつづけています。ただ達成感を得ただけなのかもしれないのですが。

最後にゼミ4期生全メンバーへの感謝と共に、五十嵐さんにも大いに感謝したいと思います。彼がいなかったら、このプロジェクトの成功度はかなり低かったと思うし、もしかしたら成功しなかったかもしれません。それだけ大きな存在だったと思います。

## 11-10【山崎淳子】as junko

＜＜文責：山崎淳子

- 1、時系列編
- 2、心境編
- 3、宴のあと…？編

---

### § 1、時系列編

こみゅぽ製作に関して、わたしがやったこと…って何??  
時系列で並べてみよう!

① 前期～夏休み

- Java の勉強 (JDK1.3、『初体験 Java』)
- HTML の勉強 → 夏合宿 HP 作り
- アプレット探し on WEB
- 議事録 (夏合宿、ジョナサンオールナイト MTG)

② 青祭準備段階 (渡タイ前)

- 事前アンケート原案作り

③ 青祭準備段階 (帰国後)

- パンフレット作成&当日アンケート作り→印刷
- 事前アンケートに関するパネル作り
- 買い物・買い出し

④ 青祭当日

- 風船作り、こみゅぽ看板作り
- 客あんない

⑤ おまけ

- タイにて

---

## §2、心境編

○ HTML 勉強 → 夏合宿 HP 作り

昨年 C&C で半期ちよろっと齧ったことしかない HTML。それから1年も年月が過ぎ、とっくのおに忘れていたので MTG 後、東急文化会館の三省堂で『HTML タグ辞典』なるゴツツイ本を購入。夏合宿の写真 from macchi75 をもとに、何とかフレーム付きの HP を作成。というか、この時期は1週間後に渡タイを控えていて、気持ち的に不安定かつ焦っていた。とりあえず、1ヶ月も空白ができてしまうから、何とか自分のやった軌跡を少しでも残していかなきゃ!みたいな切迫感で HP を作った。

○アプレット探し on WEB

「とりあえず何でも言いから WEB 上でフリーのやつを見つけてきて!」というお達しのもと、フリーなアプレットを手当たり次第 Javacats などで探した。フリーでもソースはオープンじゃないのも多く、よく分からなかった。実際、ソースを見ても、さっぱり分からず、将来これをダウンロードしてみたところで、改良したりなんてこと、できるのだろうか?などと思った。フリーなアプリケーションは、どれも機能が似たり寄ったりで、時計とか、スケジュール管理系のものがうじゃうじゃ転がっていた。でも、こんなに同じ物ばかりでどうするんだろう?と思った。そんな中でもわたしの興味をひいたものは、女性

向けのバイオリズムを管理するものや、インドネシア語のジャワ暦（どうやって使うか分からないが…。）だった。同時複数エンジンサイト検索の‘Bingo!’などは非常に機能的だった。発見したものの多くを、noaに入れておいたが、後になって、タイから帰国後、そのような発見モノは何一つ使われておらず、みな手作りするという方針になっていて、ちょっとショックだった(笑)。

#### ○議事録

夏合宿 in 八ヶ岳と、ジョナサンで行ったオールナイトの議事録を細かく付けた。こういうメモや、話し合った事柄は、後にどんなに方向転換しようとも、付けておくに越したことはない。しかし、書記とか、議事録付けというのは、細かくて几帳面な作業だなあ…と思った。ちょっと他のことを考えていると、その部分のメモを取り忘れるので、油断ならない仕事。書記と違って、頭使わなそうだし、誰でも出来そうだけれど、かなり神経を使うゾ。寝てられないもん。

#### ○事前アンケート作り

アンケートというのは、予め欲しい答えや、欲しい反応をどうした得られるか？ということから作るものだということを、初めて知った。(それを知らずに作った最初の原案は、見事にひっくり返された。) さらに、集計のしやすさ、答えてもらいやすさ…などなど、綿密に詰めて作らなければいけない。「どうやって、私達の望むように答えさせるか??」コレにアンケートの真髄があるんだなあ…。とりあえず、あとは印刷するだけ!というところまでアンケートを完成させ、フロッピーを macci75 に委ねて、わたしはタイへ発った。あとはよろしくね!

#### ○帰国後…

タイで楽しく過ごした 1 ヶ月間を胸に、まだ心は真夏のまま、帰国後ゼミに戻った。私は他のメンバーには誰にも出来ない、「タイ」へ行ったからこそ感じられる視点(?) をうまく活用できないかな? などと高尚なことを考えていたが、実際戻ってみたら、青祭を 2 週間後に控えた緊迫感と忙しさで、ゼミの雰囲気はとても恐ろしく、それどころではなかった。

また、夏休みに図書館のマルチメディア室であれほど探したフリーのアプレットは全く採用されず、すべて技術班が手作りする…という方針になっていたので、少しばかり悲しかったことも否めない(笑)。

1 ヶ月も離れた世界に行っていて、突然ピョコンと帰ってきた者に使い道が無いのは分かっているつもりだった。しかし急加速しているゼミのなかで、私は自分が今、何をやっていいのかさっぱり掴めず、しばらくつらかった。状況を飲み込むのにも少し時間があった。やっと仕事が見つかり、意気揚揚として臨んだ青祭に関する説明会でも、「っていうか来なくていいのに。」などという言葉に「えっ、そんなあ。」ズシーン。(not スジン)。

### ○パンフレット作成+当日アンケートそして印刷

気を取りなおして、よいしょっと。りっちゃんがB4をたてに三つ折するという、斬新なアイデアを出してくれた。当日来てくださるお客様に対して、どうしたら簡単に分かるように説明できるか？どうしたらマニアックに引かれなくて楽しんでいただけるか？どうしたらある程度知識のある方々に理解・興味を持っていただけるか？など、幾層ものお客様を想定して、パンフレットやアンケートを作るのは、思いのほか大変だった。しかも画像を取り込む、取り込まない、色はカラーでモノクロで…など、最終的に詰めることも多かった。というか、企画の段階で、もうすでに青祭 3 日前…というラストスパートすぎたので、焦った、焦った。(物事は前倒しにしてやるべし。)

わたしは当日アンケート作りに際して、普段、買った本の中に挟まっていたり、就職情報誌のDMなどで頻繁に家に送られてくるハガキについているようなアンケートをたくさん思い返してみた。企業のプロ(?)が作るアンケートって、どれも項目に番号がついている。質問項目の文章が短い。選択項目がかなりたくさんある。どの質問にも必ず「その他」という選択肢がある。複数選択可能の項目が多い…などなど。そして、どんなアンケートにも、最後の項目などに、自由に書き込める欄がついている。さらに、事前アンケートを作ったときのことも参照にして、今度は機能的な工夫を凝らして作ってみた。そうしたら、原案をメンバーに見せたところ、前回のように、原案そのものがひっくり返されるほどの批判はなく、ある程度の承認を得られて、わりとスムーズに印刷にこぎつけることができた。よかった、よかった。

### ○買い物・買い出し

青祭を直前に控えた夜、渋谷の東急ハンズまで買い物・買い出しに行った。パネルやかごなど、重くはないけれどかさばるものばかり。さらに学校への帰りがけには、ダンボールをありったけコンビニからもらってくるという仕事があり、青山通りをダンボール抱えて歩いたのには、何となく寂しさが募ってきた。

### ○青祭当日～風船作り、看板作り

いよいよ青祭当日。朝から風船を幾つも膨らますと、さすがに中学時代、「男子並みの肺活量ね!」と褒められた私でも頭がクラクラしてくる。昨日、寂しい思いをして運んだダンボール達も、切って貼って楽しい看板になってくると少し嬉しかった。こみゅぽのかわいらしい看板。吊り下げた矢印。自分の作ったものはどれもすばらしく見えてきた。青祭準備段階に比べて、青祭当日の余裕のあることといたら!!和やかに青祭当日の展示は進んだ。

### \* おまけのけ

#### ○タイにて…

わたしが今回、ゼミのプロジェクトで大変なことも知りながら、総務庁のタイ派遣に申し込んでしまったことには、昨年度、締め切りを過ぎてから気付いて、悔しい思いをし



たことと、自分は、国際協力系井田ゼミ生(?)として、生きていこうと思っていたからである。1ヶ月間、ゼミをお留守にしまして、使えないながらも、メンバーに迷惑をかけてしまった。それでも、帰ってきてからもすんなり活動の波の中に入れてくれて、みんながいたから、がんばれたことも多い。本当にありがとう。

で、実際にタイでやっていたことは、コンピュータとは全くかけ離れたこと。でも、地域ディバイドを考えさせられる場面があった。そして同じミレニアムを生きているという感じさせられる場面もあった。

タイに行って、わたしが最も強く感じたことは、「わたしは何で傲慢だったんだろう?」ということ。タイに行く、ほんの数日前までは、学校のマルチメディア室で、Javacatsの中なんかから、フリーのソースを探したり、JDK1.3をインストールしたりという、俗に「イマドキ」なことをやっていた。しかし、その数日後には、13軒しかない集落にホームステイして、電気も郵便も住所もない村で、ろうそく1本握り締めて生きている。コンピュータのコの字もなければ、インターネットのイの字もない。時計もなくて、ただ毎日、水牛の鳴き声で一日が始まり、お寺への托鉢から朝が始まる。ホストマザーと一緒に滝へ遊びに行った。ホストマザーは慣れなくて疲れているわたしを、膝枕して寝かせてくれた。

わたしは今まで、家族や周囲の人々に愛され、親切にしてもらって生きていることを、当然のように思ってこなかったか?今、自分がこうしてゼミで元気に勉強できたり、整えられた環境で生きていられることに、全く何も感じてこなかったのではないか??

自分が毎日元気にやってこれているのも、多くの愛情を注いでくれ、いつもご飯を用意してくれ、あらゆる面でサポートしてくれる家族がいてこそだし、ゼミでがんばれるのも、一緒に支えあってくれるメンバー、先生、そして大学院生の五十嵐さんなどがいてこそだ。そんなことを普段から考えてこなかった私は、自分だけが辛い思いをしているとか、自分だけが頑張っているなんて、大きな勘違いや錯覚に陥りやすい。

自分ひとりだって、ぐらぐら揺れる不安定な感情を支えきれない未熟者だけれど、他人の気持ちや、他人の状況を考える又は察するということの大切さを学んだ。

これが、今回のこみゅぽとその製作過程に反映されたかは、甚だ疑わしい。しかし、他人の気持ちや他人の状況を考えるということは、こみゅぽと全く無関係だとは思わない。そして、10人のメンバーでやってきた過程の中で、最も難しかったのは、この、他人の気持ちや他人の状況を考えるということではなかったかな?と思う。

---

### §3、宴のあと…?

ゼミの打ち上げでははじめてしまって、多少のみ過ぎました。そして、ご迷惑をおかけしました。申し訳ありません。東急東横線の桜木町駅で車掌さんに起こされたときはビックリしました。

# 第 12 章

## 付録

12-1 【コミュニティーポータルの各自の構想】	p. 226～
12-2 【Community Portal の Wish List】	p. 229～
12-3 【会計規則】	p. 235

## 第12章 付録

〈文責：大西淳

これまでの活動の中で様々なものが多く残っています。それら全てを記すのは無理ですので、そのなかでもゼミ生一同強く掲載しておきたい、これまでの過程を思い出す上でも重要だと思われるものをここで取り扱うことにしました。

### 12-1【コミュニティーポータルの各自の構想】

プロジェクトが始まって、右も左も分からない頃それぞれが自分たちの中で思い描いていたコミュニティーポータルの意識です。この頃の新鮮な気持ちをいつまでも忘れずという意味で、またこれまでのこみゅぽの土台ともなったこれらのメモを残したいという意味で掲載しました。

#### ●macci75 案

##### 1.(自分が欲しい)と思う機能(コミュニティーポータルとして)

- ・ portable なポータルサイト (携帯でチェックできるということ)
- ・ メンバーが今どこで何をしているか  
などのチェックができる、ということ。
- ・ goo のような検索エンジン  
情報を手に入れるところ
- ・ (学生として)参加可能なイベント  
(講演会とか、勉強会とか、学生活動とか)の案内。(ぴあのようなものの学生限定版)
- ・ ニュースの一覧機能(朝刊ななめ読み)
- ・ メンバーの status(自己紹介みたいなもの)

#### ●aqua 案

- ・ 学習機能を持ったページ  
始めはいろいろなカテゴリを網羅してるんだけど、しばらく使っている間に、訪れた回数が多いサイトを含むカテゴリだけを表示するようなページ。
- ・ 自分で作るページ  
自分でホームページを作るような感覚で、自分独自のサイトを簡単に作れるようなページ。部品はあらかじめ用意しておいたうえで、これだけは必ず使用してもらいたい部品 も指定しておく。
- ・ 携帯端末でも見られるページ  
i mode などでも見られるページ。もちろん、その分情報は精選されたものにする

### ●keiko 案

- ・各ゼミ、各部活、各サークル向けに、大学版 Intranets を提供する
- ・ウェブメール
- ・参加すると抽選で1名に昼食（学食）が当たる投票
- ・図書館の貸出状況がわかる

### ●arrow 案

- ・自分でカスタマイズできるサイト作り

自分らしいポータルサイトを自分で作る。(サイトのなかに自分の興味ある分野の情報を組み込んでいく→パズルの要素)例えば、intranets のカレンダーの部分を時刻表にするなど、自分の都合のいいようにサイトを構成する。

- ・組み入れたいもの

JR 時刻表→特に南武線かなんか。 休講情報 株価情報 WEB メールのようなもの  
お気に入りサイトへの一発リンク(いちいちお気に入りを開くことなくそのサイトへいく) など

### ●moriken 案

- ・声の出るサイト。学習機能があるならそれについての情報を世間話のようにしてくれるもの。ポータルサイト版アイボまたはシーマンのようなもの
- ・サイトを②分割または④分割して同時に見る
- ・特定の人が今現在 Online してるかしてないかを自動的に教えてくれるもの (ICQ はある)

### ●cafekuro 案

- ・いろんな掲示板へのリンク

自分の好きな BBS を事前に登録してもらいそこに書き込みがあったら知らせるサービス

- ・E-MAIL 着信通知サービス

自分のメールアカウントを事前に登録しておき、メールがきていたら知らせてくれるサービス

- ・ブラウザのようなメニューバーがついたもの

自分のブラウザの左側かどこかにどこのページにいても開いているメニューバーをおき最新情報（ニュース）や仲間の状態を逐一見ることができるもの。(邪魔なら出したり消したりできる)

- ・自分のお気に入りのページの更新情報

自分のお気に入りのページを登録してもらい情報が更新されたら知らせるサービス

### ●ritz 案

- ・ニュースをすぐに見られる。政治、経済、社会、芸能とかチャートなど

- ・ i モードとかの端末で見られる
- ・ 学校の掲示板。休講情報
- ・ イベント情報とかコンサート情報
- ・ 映画の情報、講演会情報

### ●junko 案

あったらいいもの

- ・ 人の居場所が検索できるもの
- ・ ページの在処が検索できるもの
- ・ 自分の所在地がわかるもの
- ・ 自分の歴史（保存とか）がわかるもの
- ・ 今日は何をしよう？っていうスケジュール
- ・ お天気
- ・ 時計と日にちが内臓されてて、開けるとピコーン！とその日締切りの宿題や小テスト情報が自動的に表示されるようなプログラム

ページから感じたい(?)ものやこと

- ・ 人のぬくもり

(Web や PC 上で、人のぬくもりとか所在とかを感じられる仕組みがあったら)

- ・ 季節
- ・ 天気

やれたらいいなと思うしくみ

- ・ 郵便事業

(そのページから直接メールを送ったり書いたり受信したりできるもの)

- ・ メモ帳

(紙のメモ帳のように、文字も絵も自由自在に書いたり消したり破って保存したりできるもの)

- ・ 自分のお金を管理しておくところ  
(これはパスワードとかで保護する。)
- ・ 今日の献立を自分で考えて、冷蔵庫の中にあらかじめあるものを入力しとくと、何を買い足せばいいかピコーン！と算出してくれるプログラム

### ●sussy 案

- ・ Dosule!のサービスみたいなもの。

自分のスケジュール管理が出来て、To Do リストやメールの送受信も出来るもの。PDA が Web 上にあるみたいなもの

- ・ 学生向けのポータルとして

学校の休講・教室変更や学校内の講演会のお知らせみたいなものをメールボックスには

もちろん携帯端末にも発信してもらえるサービス

- ・同窓会のページみたいなもの
- ・学生にとってお得な情報や学生が知りたい新製品の情報。学校の周りのお勧めのお店の情報などを広告として載せる

### ●jon 案

- ・ブラウザを立ち上げた時間などを計算し具体的にかかる時間を計算するもの。学校、バイト先、成田空港、JFK、試験会場、etc までにかかる時間をダイヤと照らし合わせ、詳細に計算する。時には電車の事故を発見し、タクシーでの transportation を薦めたりする。
- ・個人設定も出来るようにして、個人差による delay などを修正出来たり。
- ・いままでの 2 次元的な「表示」をやめ、データ量に依存しない自由に移動ができる仮想 3 次元の「空間」を開発。
- ・ポータルはたくさんの方がいるだけ、それだけたくさんのニーズに答えないといけないので一様ではいけない。
- ・例えばどんなに良いポータルがあっても必ずやこれは要らない、これあったらいい系の需要がでるはず。
- ・そこで、各人が自由に設定できる環境が整わなければいけない。
- ・知識のある人、無い人、若いひと、老いた人老若男女に対応しうるシステム（一人一人にあった設計）
- ・金融のポータルとか何とかのポータのように目的別になっているのが現状。作りやすいのかも。

## 12-2 【Community Portal の Wish List】

このメモも 7 月の中旬ごろに採集したもので、ゼミ生がコミュニティーポータルに一体何を求めているのかというのを調査するものでした。この指針によって今までの方向性が決まったといっても過言ではありません。まだまだ未熟だったこの頃の方向性がこの後、八ヶ岳でのいろいろな意味で悲劇的な夏合宿へとつながっていったのでした。

(<http://noa.sipeb.aoyama.ac.jp/semi4/smrcmp/summercamp.html> を参照してください。)

【macci75】

どうも.macci75 です。

ポータルサイトに関しての自分の意見です。

(かなりかぶってしまうけど、許してくださいね)

1. (自分が欲しい) と思う機能

コミュニティーポータルとして…

(はじめにコミュニティーがあることが前提ですよ?)

- a. portable なポータルサイト  
(携帯でチェックできるということ)
- b. メンバーが今どこで何をしているか  
などのチェックができる、ということ。
- c. goo のような検索エンジン  
情報を手に入れるところ
- d. (学生として) 参加可能なイベント  
(講演会とか、勉強会とか、学生活動とか)  
の案内。(ぴあのようなものの学生限定版)
- e. ニュースの一覧機能(朝刊ななめ読み)
- f. メンバーの status (自己紹介みたいなもの)

今思いつくあったらいい機能はこれくらいです

【aqua】

- あったらいいな・・・シリーズ

学習機能を持ったページ

始めはいろいろなカテゴリを網羅してるんだけど、しばらく使っている間に、訪れた回数の多いサイトを含むカテゴリだけを表示するようなページ。

自分で作るページ

自分でホームページを作るような感覚で、自分独自のサイトを簡単に作れるようなページ。部品はあらかじめ用意しておいたうえで、これだけは必ず使用してもらいたい部品も指定しておく。

携帯端末でも見られるページ

i-mode などでも見られるページ。もちろん、その分情報は精選されたものにする。

【keiko】

- 各ゼミ、各部活、各サークル向けに、大学版 Intranets を提供する
- ウェブメール
- 参加すると抽選で1名に昼食(学食)が当たる投票

●図書館の貸出状況がわかる

【arrow】

●自分でカスタマイズできるサイト作り●

自分らしいポータルサイトを自分で作る。(サイトのなかに自分の興味ある分野の情報を組み込んでいく→パズルの要素)例えば、intranets のカレンダーの部分を時刻表にする。こんな感じです。aqua の第 2 案と似ているがまあいいか。また浮かんだら出します

【moriken】

ポータルサイトのアイデア

声の出るサイトシー君のをヒントに、学習機能があるならそれについての情報を世間話のようにしてくれるもの。ポータルサイト版アイボまたはシーマンのようなもの

サイトを②分割または④分割して同時に見ていけたらいいなあ

特定の人が今現在 Online してるかしてないかを自動的に教えてくれるもの (これってすでにありそう)

ICQ はあるんだけど。

【cafekuro】

実行可能性は少ないかも知れないけどブレインストーミングとして出します。

・いろんな掲示板へのリンク

自分の好きな BBS を事前に登録してもらいそこに書き込みがあったら知らせるサービス

・E-MAIL 着信通知サービス

自分のメールアドレスを事前に登録しておき、メールがきていたら知らせてくれるサービス

・ブラウザのようなメニューバーがついたもの

自分のブラウザの左側かどこかにどこのページにいつでも開いているメニューバーをおき最新情報 (ニュース) や仲間の状態を逐一見ることができるもの。  
(邪魔なら出したり消したりできる)

・自分のお気に入りのページの更新  
情報

自分のお気に入りのページを登録してもらい情報が更新されたら知らせるサービス



こんなところでは、またあったら書きます。

#### 【ritz】

そうだなあー今思いつくのは…

1. ニュースをすぐに見られる。政治、経済、社会、芸能とかチャートとかもあったらいいなあ。
  2. iモードとかの端末で見られるのは良いと思う。かぶってるけど。
  3. 学校の掲示板。休講情報とか。
  4. イベント情報とかコンサート情報とか。
- あと映画の情報、講演会情報。

#### 【junko】

「自分がインターネットと接続したときに一番初めに見たいページ」っていう概念を、わたしは、「自分がドアを開けて家を出ようとするときに、一番始めにそろっていたら嬉しい景色」または、「朝起きたときに一番始めに見たいもの」って考えて見ました。そしたら…、

1. あったらしいもの
  - ・人の居場所が検索できるもの
  - ・ページの在処が検索できるもの
  - ・自分の所在地がわかるもの
  - ・自分の歴史（保存？とか）がわかるもの
  - ・今日は何をしよう？っていうスケジュール
  - ・お天気
  - ・時計と日にちが内臓されてて、開けるとピコーン！とその日締切りの宿題や小テスト情報が自動的に表示されるようなプログラム
2. ページから感じたい(?)ものやこと
  - ・人のぬくもり

(ちょっとキモチ悪いかもしれないけど、自分が外の世界に出たり、朝目覚めたときなど、やはり人間の存在みたいのを感じられた方が人間的。Web や PC 上で、人のぬくもりとか所在とかを感じられる仕組みがあったら、いいと思う。)

  - ・季節
  - ・天気

### 3. やれたらいいなと思うしくみ

- ・郵便事業

(そのページから直接メールを送ったり  
書いたり受信したりできるもの)

- ・メモ帳

(紙のメモ帳のように、文字も絵も自由自在に書いたり消したり破って保存したりできるもの)

- ・自分のお金を管理しておくところ (これはパスワードとかで保護する。)
- ・今日の献立を自分で考えて、冷蔵庫の中にあらかじめあるものを入力しとくと、何を買い足せばいいかピコーン! と算出してくれるプログラム (こんなのムリか。)

まだまだ考えたらあるかもしれないけれど、ひとまず今日はこのくらいに・・・

Junko でした

【sussy】

sussy です。

あ、21日になっちゃった。ごめんなさいね。メールのチェックしてたら20日が過ぎちゃいました。

私があつたらいいなと思うものは、みんなも知ってると思うけれど、Dosule!のサービスみたいなもの。

自分のスケジュール管理が出来て、To Do リストやメールの送受信も出来るもの。  
PDA が Web 上にあるみたいなものですね。

それに加えて、学生向けのポータルにするのだから、学校の休講・教室変更や学校内の講演会のお知らせみたいなものをメールボックスにはもちろん携帯端末にも発信してもらえ  
るサービスが欲しいと思います。

また、同窓会のページみたいなものを作っても面白いと思います。

学生にとってお得な情報や学生が知りたい新製品の情報? 学校の周りのお勧めのお店情報?

などを広告として載せるのもいいのでは。

それでは

【jon】

今までのエキスパートとは違って、具体的な時間を返してくれるもの。結構ソフトウェアっていう感じになってしまうけど、いままであったものを複合したってなんの意味もないからね。

考えたのが、ブラウザを立ち上げた時間などを計算し具体的にかかる時間を計算するもの。学校、バイト先、成田空港、JFK、試験会場、etc までにかかる時間をダイヤと照らし合わせ、詳細に計算。

今までにこんな経験はありませんか??せっかく早く駅に行ったのに 5 分あとに家を出ても同じ電車だったってこと??もしくはその後の乗換えで結局特急に抜かされたりとか。

これはそんな無駄な時間を省き、スムーズで計画的な行動が出来るはず。

時には電車の事故を発見し、タクシーでの移動の方が良いとかも指摘したり。

個人設定も出来るようにして、個人差による遅延などを修正できたり。

## 12-3【会計規則】

研究に使うお金をいかに効率よく使うか、計画性を立てるためには会計規則が必要です。以下の文章はプロジェクトを通じて根幹となった井田ゼミ公認会計士矢島君の2000年度会計規則です。

### 2000年度井田ゼミプロジェクト会計に関する規則(第1版)

2000年7月22日発行

- [1] 予算はアルバイト料112,000円とBBQ費だった3,000円を足した115,000円である。
- [2] 基本的に交通費と資料費に使用する。
- [3] 請求・清算は週一回(あるいは二回)に行なわれるミーティングの際行なう。
- [4] 具体的規則
  - ①交通費(予算30,000円)
    - －交通費の支払いは、各自が所有する定期券の範囲外において行なう。
    - －公のイベント(例:EXPO)の時の交通費は、1000円を上限とし、最終的な決算の後に余裕が生じた場合、余剰分を支払う。
    - －学校でのミーティングでは交通費は支給しない
    - －交通費の請求・清算は「交通費請求書(兼領収書)」で行なう。
  - ②資料費(予算30,000円)
    - －レシートを必ず提出する。
    - －レシートに書名を出来るだけ併記する。
    - －請求・清算は[2]に基づく。
  - ③ミーティング補助費(予算45,000円)
    - －各班のミーティングに関しては、補助金として一人一律500円を支給する。(月3回まで、合計1500円/月)用途は自由であるが、必ず「交通費請求書(兼領収書)」の(兼領収書)の部分を○で囲み、それをもって請求・清算する。(一か月分まとめて請求することも可能)
- [5] 補足
  - －残りの10,000円は、青山祭の準備費用(機材以外)とする。その請求・清算については後日説明する。

\*本規則は2000年8月1日をもって施行するものとする  
(但し、7月18日、19日のEXPOの清算は7月28日に行なう)

# 第 13 章

## 参考文献・URL

13-1 【Java 関連】	p. 237～
13-2 【Linux 関連】	p. 238
13-3 【サーバー関連】	p. 238～
13-4 【Web、i-mode 関連】	p. 239
13-5 【その他】	p. 240

## 第 13 章 参考文献・URL

### 13-1【Java 関連】

Suleiman Lalani, Kris Jamsa (1997) 『JAVA プログラミングケーススタディ』、東京、株式会社スリーエーシステムズ

阿藤しづ枝, 長島実 (2000) 「超入門サーブレット/DB システム構築術 (前編及び後編)」  
『Java PressVol. 12』、東京、株式会社技術評論社

井田昌之 (1997) 『New はやわかり Java』、東京、共立出版株式会社

井田昌之、清水康司、中里宗敬、堀内正博 (1997) 『情報リテラシー教科書 ネットワーク世代の知的生産術』、東京、共立出版株式会社

大村忠史 (1998) 『Swing による Java GUI プログラミング』、東京、株式会社カットシステム

桑原恒夫 (2000) 『3 日で解る Java』、東京都、共立出版株式会社

ジョセフ・オニール (1999) 『独習 Java』、東京、株式会社翔泳社

鈴木克典 (2000) 『Java PRESS Vol. 13』、東京、株式会社技術評論社

鈴木克典 (2000) 『Java PRESS Vol. 14』、東京、株式会社技術評論社

スティーブン・ホルズナー QUIPU LLC 遠藤美代子訳 (1999) 『テクニック・ライブラリ シリーズ Java2』、東京、株式会社 IDG コミュニケーションズ

辻秀典、渡辺高志 アクロバイト&インプレス編集部 (2000) 『できる Linux サーバー構築』、東京、株式会社インプレス

原田洋子 (1999) 『Java Servlet 最新サーバ・プログラミング』、東京、株式会社秀和システム

藤田一郎 (1999) 『Java がわかる』、東京、株式会社技術評論社

丸の内とら (2000) 『初体験 JAVA』、東京、株式会社技術評論社

結城浩 (1999) 『Java 言語プログラミングレッスン(上)』、東京、ソフトバンクパブリッシング株式会社

結城浩 (1999) 『Java 言語プログラミングレッスン(下)』、東京、ソフトバンクパブリッシング株式会社

IDG Japan Inc. (2000) 『月刊 JavaWorld10 月号』、東京、IDC コミュニケーションズ株式会社

石井達夫 (1996) 「PostgreSQL mailing list in Japan」  
<http://www.sra.co.jp/people/t-ishii/PostgreSQL/>

128K-Communications Ltd. (1996) 「Programmers Heaven - Java Zone - Unsorted Filelist」  
<http://www.programmersheaven.com/zone13/cat216/index.htm>

javaboutique.com (2000) 「JAVA BOUTIQUE」  
<http://javaboutique.internet.com/index.html>

javapowered.com (2000) 「Java - Applets - Free Java Applets - javapowered.com」  
<http://www.javapowered.com/werks.html>

javashareware.com (1997) 「JavaShareware.com」  
<http://www.javashareware.com/>

Sun Microsystems (1995) 「java.sun.com - The Source for Java(TM) Technology」  
<http://www.java.sun.com/>

VCUBE INTERNET INC. (2000) 「www.stylec.com」  
<http://www.stylec.com/>

### 13-2 【Linux 関連】

ASCII Corporation (1999) 「日刊アスキーLinux」  
<http://www.linux24.com/linux/>

### 13-3 【サーバー】

小泉修 (1997) 『図解でわかる サーバーのすべて』、東京、株式会社日本実業出版社

Apache Software Foundation (1999) 「The Jakarta Project」

<http://jakarta.apache.org/>

Apache Software Foundation (1999) 「Tomcat Documentation」

<http://jakarta.apache.org/tomcat/jakarta-tomcat/src/doc/index.html>

Apache Software Foundation (1999) 「Tomcat- 簡易ユーザーガイド」

[http://servlet.good-day.net/tomcat\\_ug\\_j.html](http://servlet.good-day.net/tomcat_ug_j.html)

原田洋子 (2000) 「Servlet Garden ～Servlet のお庭～」

<http://www.t3.rim.or.jp/~yoko-k-h/java/servlet/>

### 13-4 【Web、i-mode 関連】

アंक (2000) 『カラー版 ホームページ辞典』、東京、株式会社翔泳社

アंक (2000) 『i-mode タグ辞典』、東京、株式会社翔泳社

内田広由紀 (2000) 『わかりやすく効果的 Web デザイン基礎講座』、東京、株式会社視覚デザイン研究所

川崎和哉 (1999) 『オープンソース・ワールド』、東京、株式会社翔泳社

河西朝雄 (2000) 『i モード対応 HTML と CGI』、東京、株式会社ナツメ社

ヤコブ・ニールセン著、グエル訳 (2000) 『ウェブ・ユーザビリティ 顧客を逃さないサイトづくりの秘訣』、東京、株式会社エムディエヌコーポレーション

石野恵一郎 (1998) 「Another HTML-lint gateway」

<http://openlab.ring.gr.jp/k16/htmlhint/htmlhint.html>

梅本 肇 (1998) 「ネットワーク時代の知的所有権入門」

<http://www.imasy.or.jp/~ume/copyright-ml/inetmag/>

神崎正英 (1995) 「The Web KANZAKI -- Japan, music and computer」

<http://www.kanzaki.com/index.html>

こころリソースブック編集会、社団法人日本電子工業振興協会 「こころ Web」

<http://www.jeida.or.jp/document/kokoroweb/>



社団法人 著作権情報センター「社団法人 著作権情報センター」

<http://www.cric.or.jp/>

Jakob Nielsen (1995) 「useit.com: Jakob Nielsen's Website」

<http://www.useit.com/>

Tim Berners-Lee (1997) 「The Implications of Links -- Axioms of Web architecture」

<http://www.w3.org/DesignIssues/LinkLaw>

W3C (1997) 「Web Accessibility Initiative (WAI)」

<http://www.w3.org/WAI/>

W3C (1994) 「The World Wide Web Consortium」

<http://www.w3.org/>

W3C (1999) 「XHTML? Basic」

<http://www.w3.org/TR/xhtml-basic/>

### 13-5 【その他】

藤沢晃治 (1999) 『分かりやすい表現の技術』、東京、株式会社講談社

好川哲人 (1998) 『[最新] TCP/IP ネットワーキング』、東京、株式会社技術評論社

ALCATEL (1999) 「Alcatel Webtouch」

<http://www.webtouch.alcatel.com/home.htm>

impress (1999) 「学生生活のニーズを満たす学生用ポータル」

<http://www.watch.impress.co.jp/Internet/www/article/1999/0823/egenda.htm>

Freewarejava (1999) 「Freewarejava.com Java Applets」

<http://freewarejava.com/applets/index.shtml>

WorldHot (2000) 「WorldHot.com - World Hottest Web Sites」

<http://www.worldhot.com/>