



## 学内情報提供システムの構築

メタデータ	言語: jpn 出版者: 公開日: 2013-12-20 キーワード (Ja): キーワード (En): 作成者: 福本, 隆弘, 大久保, 智史, 窪田, 哲也 メールアドレス: 所属:
URL	<a href="https://doi.org/10.24729/00007615">https://doi.org/10.24729/00007615</a>

# 学内情報提供システムの構築

福本隆弘\*, 大久保智史\*\*, 窪田哲也\*\*\*

Development of Campus Information Service System

Takahiro FUKUMOTO\*, Satoshi OKUBO\*\*, Masanari KUBOTA\*\*\*

## ABSTRACT

本校では、学生への連絡は掲示板を用いて行っている。しかし、多数の情報が一つの掲示板に収まるために情報の取捨選択が面倒であり、確認のための手間も多い。そこで本研究では、学内における各種の情報のうち、各個人が必要とするものを個人認証によって提供し、また、情報の登録・管理も容易に行えるシステムについて開発を行う。

**Key Words:** 掲示板, 個人認証, 一元管理, 学生サービス

### 1 はじめに

現在、本校では休講や時間割の変更、学生の呼び出し等の情報伝達手段として、掲示内容を貼り付ける掲示板を使用している。掲示板の性質上、その大半が自分に関係のない情報や、放置されたまま残ってしまっている古い情報であり、必要な情報が埋もれてしまっていることがある。そのため、既存の掲示板では、閲覧者にとって必要な情報のみを容易に見つけ出すことが困難となっている。また、情報の確認や、掲示内容を更新する場合、設置場所に行く必要がある。さらに、掲示板が教室に近いとは限らないため、学生が頻りに掲示の確認を行わないと思われる。

富士通エフ・アイ・ピー株式会社が提供している、「モバイルキャンパス」[1]を導入している大学や短大がある。このモバイルキャンパスとはインターネットを利用して、PCと携帯電話から学校側が発信する休講情報や学校行事などの情報を確認できるサービスである。このシステムでは、個人を対象とした情報は個人用掲示板に掲示するとメールでも通知を行ってくれるので、情報を検索する手間がかからず、必要な情報のみ確実に入手できる。しかし、このシステムの利用には年額百数十万円が必要となる。

自校にて開発した例もある。福島大学では、学校側か

らの情報を PowerPoint 等を用いて電子化を行い、大型プラズマディスプレイに表示する「お知らせ配信システム」[2]を導入している。同様に、津山高専でも掲示板全ての上部の壁に液晶モニターを備え付け、情報を画像化したものを一定時間毎にスライドショーで学生に提示している[3]。これらのスライドショーの優先順位は情報提供者がフレキシブルに変更可能である。しかし、情報を順次切り替えながら表示しているので個人を対象とした情報の表示には適していない。また、目的の情報が表示されるまで画面の前で待たなくてはならないという問題点がある。

そこで本研究では、ユーザ認証が行われていない場合は「お知らせ配信システム」のように大多数の学生向けの情報を順次表示し、ユーザ認証後は、モバイルキャンパスのように個人を対象とした情報の表示が行えるといった双方の特徴を生かしたシステムの提案・開発する。

### 2 現状分析と提案システム

現在の掲示板は教室から遠い場所にあるため、ほとんどの学生が登下校時にのみ掲示内容の確認を行っている。高学年の自転車通学者にいたっては、掲示板の前を通らずに駐輪場に行ってしまうため、ほとんど掲示内容の確認を行っていないのが現状である。これらの問題は、学内のさまざまな場所に掲示板を設置することで解決が可能だと考えられる。しかし、掲示板の設置場所が増えることで、情報の掲示・更新作業に必要な労力が多くなる。これは現状の掲示板の管理状態を見る限り現実的な手段ではない。さらに更新作業が

2008年4月9日受理

\* 総合工学システム専攻 電気電子工学コース

(Course of Advanced Industrial Systems Eng. : Electrical and Electronic Eng. Course)

\*\* 電子情報工学科 2007年度卒業生

(Graduation of Dept. of Electrical Eng. and Comput. Sci.)

\*\*\* 総合工学システム学科 電子情報コース (Dept. of Industrial Systems Eng. : Electrical Eng. and Comput. Sci. Course)

煩雑になり, 今以上に情報の管理ができなくなる可能性が否めない。

そこで, 掲示板を電子化することでこれらの問題を解決する。図 1 に示すように, 学内には LAN が張り巡らされており, これを利用することで掲示内容の管理・投稿を一元的に取り扱うことが可能となる。また, 各教室・廊下などに設置することで, 従来は掲示板を見る回数の少なかった学生も容易に情報の取得が可能となる。さらに, 今回提案する学内情報提供システムでは, 自らが利用する PC から情報を得ることが可能となる。教員も自室の PC を用いて学生への連絡・呼び出しを行うことが可能となる。しかし, 掲示板の代わりとなる情報端末を廊下や図書館等に設置する必要があり, そのためのコストが発生してしまう。だが, このシステムの導入による学生サービスの向上は計り知れないものがあると考えられる。

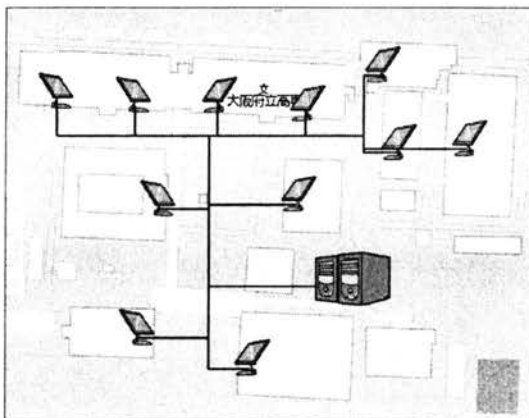


図 1: 学内ネットワーク

### 3 学内情報提供システムの開発

#### 3.1 システムの概要

本システムの利用形態を図 2 に示す。本システムは大きく 4 つに分けられる。Web ブラウザを用いて投稿・閲覧を行うための Web アプリケーション (以下 Web アプリ), 記事投稿を行うための Windows 用アプリケーション (以下投稿アプリ), タッチパネル式の情報提供専用端末 (以下専用端末とする) にて閲覧する際に用いる Windows 用アプリケーション (以下閲覧アプリ), 携帯電話のメール機能を用いて情報の投稿・閲覧を行うためのメールアプリケーション (以下メールアプリ) である。投稿・閲覧共に 3 種類の利用形態を用意することでアクセシビリティの向上を図った。

システムの利用手法を示す。投稿者は専用のアプリケーションまたは Web ブラウザを用いて掲示内容を作成する。この時, 表示期日や対象者・対象団体 (学年・学科など) を絞り込むことが可能である。また, 休講・呼びだしなどのカテゴリも設定され, カテゴリ分けによる掲示内容の明確化が行えるようになっている。掲



図 2: システムの利用形態

示対象者を掲示データに付随させることにより, 閲覧者がユーザ認証を行って Web ブラウザや専用端末を利用する場合には本人に関連した情報のみを取得することが可能となる。また, 携帯電話のメールアドレスを事前に登録しておくことでメールを利用した情報投稿・取得が可能となる。

#### 3.2 データベースの設計

今回 Web アプリ・メールアプリの作成には Ruby On Rails (以下 Rails) を, 投稿アプリ・閲覧アプリには C# と Microsoft .Net Framework を用いた。C# は基本的に直接 SQL 文を用いてデータベースを操作するが, Rails では ActiveRecord という機構を用いて隠蔽化される。この Rails の機能を活かすため, データベースの設計は Rails の命名規則に従って行った。図 3 に ER 図を示すと共に各テーブルの役割を示す。

- users** ユーザ認証に必要な情報を持つ
- user\_infos** 氏名などのより詳細な個人情報を持つ
- notices** 掲示内容 (題名・本文 など) を持つ
- attached\_files** ポスターなどの画像による添付ファイルに関わる情報を持つ
- categories** カテゴリの名称 (休講情報・呼びだし など) を持つ
- tguser\_notices** 掲示内容とユーザとを結びつける多対多テーブル

notice\_has\_categories 掲示内容とカテゴリを結びつける多対多テーブル

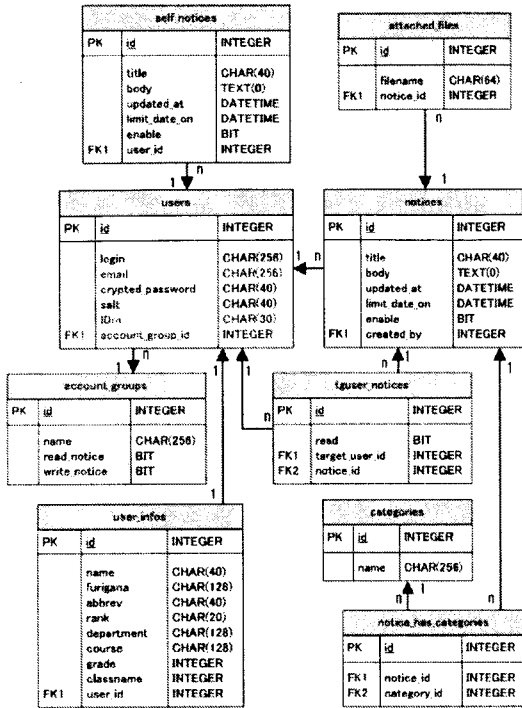


図 3: データベース

しかし掲示内容とユーザを直接関連付けると、学生全体の掲示の場合は約 1000 件の多対多対応用のデータが DB に追加される。運用のことを考えた場合、これは効率が悪いため図 4 に示すように「団体」という概念を導入し、これと掲示内容を結びつけられるようにすることで効率化した。団体には「学年」「クラス」「学科」などのあらゆる団体を含む。ユーザはこれらの団体に複数所属し、各団体を対象とする掲示も自分のものとして表示される。

3.3 ユーザ認証

本システムの利用時にユーザ認証を行うことで、その個人に関連した情報が得られるが、その認証方式のひとつとして Felica を用いた個人認証システムを作成した。これは、Felica に付加される固有番号を用いたもので、専用端末を使用する際に利用できるものである。Felica を用いる利点としては以下のようなものが挙げられる。

- おサイフケータイ・ICOCA など普及しており、認知度も高い
- リーダが安価に入手可能

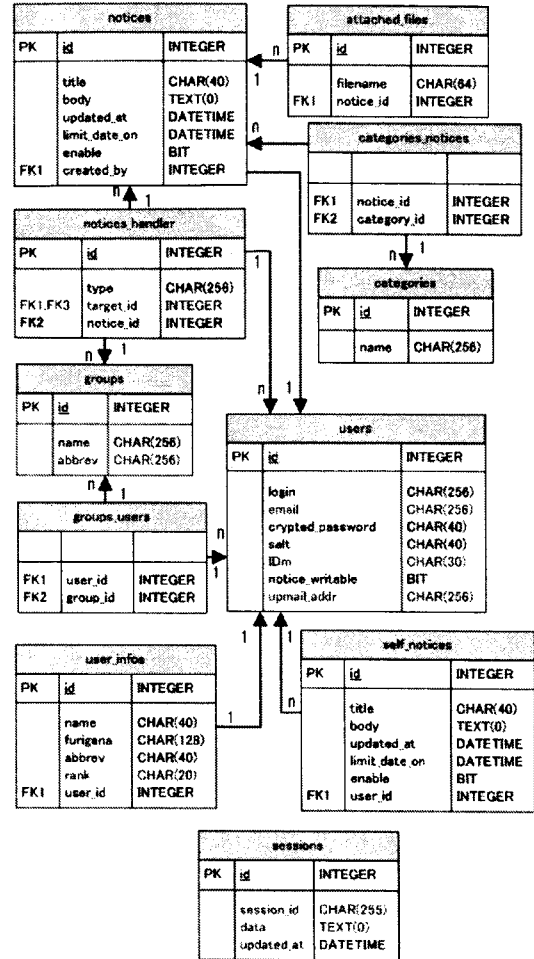


図 4: 改善されたデータベース

- 開発キットに無料のものがある

また、使い方も定期として ICOCA を用いる場合と同様に端末にかざすだけであり、初めて利用する場合でも利用法の習得は容易であることも挙げられる。

なお、認証部分はシステムとは別個に作成することが可能であり、現在 Felica 以外の認証方式として Web アプリにおける ID・パスワード入力によるチャレンジ&レスポンス方式、メールアプリにおけるメールサーバのアドレスや、メールに付加される一意の MessageID を用いた認証方式を備えている。メールアプリにおける認証は 3.5 節にて詳細を述べる。

3.4 Web アプリ

学内には Windows だけではなく、Mac や Linux を使用している人も多い。本システムを異なる OS 間で利用可能とするために、Web を用いた投稿・閲覧アプリを作成した。このアプリケーションの作成には MVC

モデルを採用し, さまざまな DB と容易に接続可能な Rails を用いた。

### 3.4.1 閲覧アプリ

ユーザが閲覧アプリにログインすると, 図 5 に示すような記事一覧が表示される。この時に右側のメニューを使用することでカテゴリを絞り込んだり, 数日以内に更新された情報だけを得る事が可能となる。また, 図 6 に示すようにメールアプリでの記事閲覧に使用するメールアドレスの設定や, 氏名・所属などを確認することもできる。

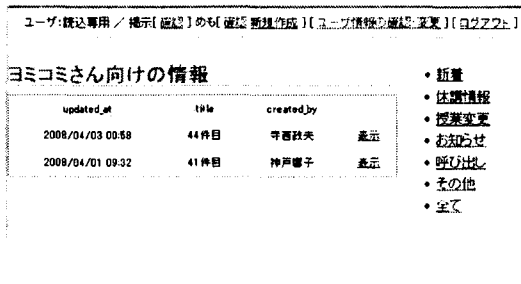


図 5: Web 版閲覧アプリ

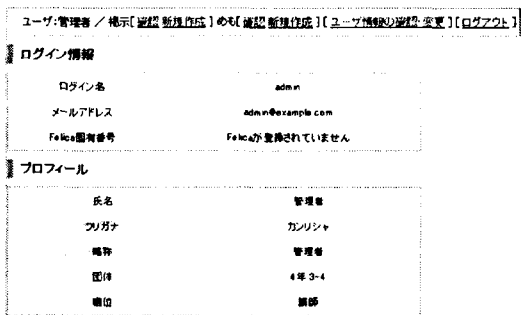


図 6: 個人の情報確認

### 3.4.2 投稿アプリ

上部のメニューより新規作成を選択することで記事投稿の画面に移る。記事投稿画面を図 7 に示す。この投稿アプリでは掲示対象を選択する部分に Ajax という技術を用い, ページ遷移を行うことなく対象の選択・削除が行えるようになっている。一連の流れを図 8 に示す。掲示対象を選択するには, まずその対象の団体名を指定する (図 8(a))。すると, その団体に所属する個人の一覧が表示される。この時に追加したい個人 (図 8(b)), あるいは「全て」のリンクをクリックすること (図 8(c)) で対象として追加される。

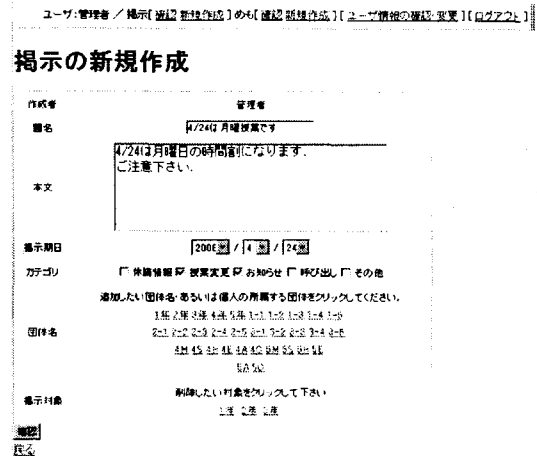
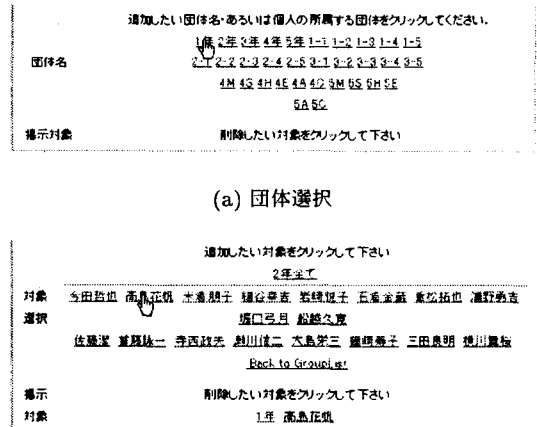
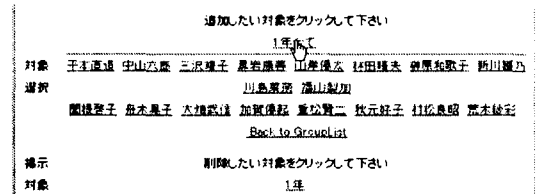


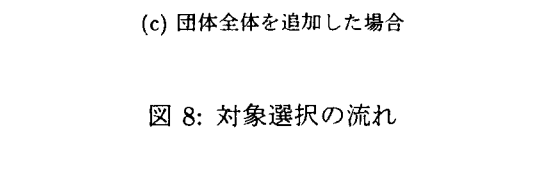
図 7: Web 版投稿アプリ



(a) 団体選択



(b) ある個人を追加した場合



(c) 団体全体を追加した場合

図 8: 対象選択の流れ

### 3.5 メールアプリ

近年の携帯電話の機能向上は留まるところを知らず、Webブラウザの機能ひとつをとってもPCと遜色ない機能を持つものまで出現し始めている。しかし、その一方で携帯電話でWebブラウジングを行う人は少なく、ほとんどがメールを利用するのみという実情もある[4]。本システムにおけるメールアプリは、そのような現在の利用形態に適した情報提供手段を与えるものである。このメールアプリの作成にはWebアプリと同様にRailsを用いた。Railsにはメールを扱うためのActionMailerという機能が用意されており、SMTPサーバを通じてのメール送受信やメールヘッダの情報取得などが容易に行える。

本アプリケーションではメールの送信先と本文の内容を変更することで新着情報の取得や記事の投稿を行うことができる。なお、メールアドレスを詐称することによる「なりすまし」を防ぐために(3.5.1)に示す対策を講じている。また、最大で1000人程度が同時にアクセスすることが見込まれるため、(3.5.2)に示すように動作の効率化を図っている。

#### 3.5.1 メールサーバのアドレス逆引きによる詐称回避

メールの送受信で用いられるメールアドレスというものは本来一意のものであるが、現在のメール送受信に用いられるプロトコルでは送信元がこれを詐称することが可能となっている[5]。つまり、送信元メールアドレスによって個人認証を行う場合は、そのアドレスが本来のアドレス所有者から送信されているかを検証する必要がある。今回、この問題を送信元メールサーバのIPアドレスの逆引きによって限定的にはあるが解決した。メールアドレスを詐称することは可能であるが、送信したサーバのアドレスを詐称することは困難である。そこで、送信したサーバのIPアドレスをDNS逆引きを行うことでドメイン名を取得し、これがメールアドレスのドメインと一致するかを検証することで本来のサーバから送られていないメールを弾くことにした。

#### 3.5.2 dRubyによるActionMailerの効率化

ActionMailerを用いてメールを受信する際には、あるメールアドレスのエイリアスとしてRailsの起動スクリプトを渡すようになっている。しかし、この方式ではメールを受信するたびに新しいRailsのプロセスが立ち上がるためオーバーヘッドが大きく、最大1000人程度が利用すると考えられる本システムには不向きである。

そこで、Railsのプロセスは常時起動させておき、そこにメールのデータを流し込む事で処理を行うような仕組みを作成した。

この機能の作成にはdRubyと呼ばれる機能を利用した。dRubyは、すでに起動しているRubyのプロセスにあるメソッドをネットワーク越しに呼び出せるようにする機能であり、Railsのメール受信部をdRubyによってネットワーク化、これをSMTPサーバにより起動される簡易なスクリプトから呼び出すことでRails環境立ち上げに伴うオーバーヘッドの軽減を行った。機能概略図を図9に示す。

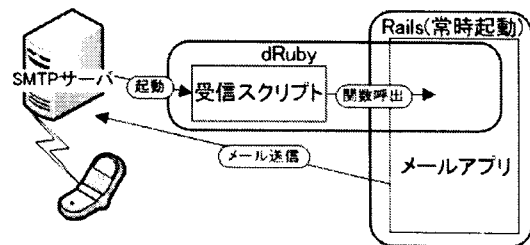


図9: dRubyとRailsの関係

### 3.6 Windows用アプリケーション

Webアプリ版の投稿機能との機能差は無いが、Windows上での利便性を考慮して作成されている。このアプリケーションの作成にはGUIの構築と検証が容易であり、ガベージコレクタを標準で備えるC#を採用している。

#### 3.6.1 投稿アプリ

投稿アプリはWindowsを利用している教職員が記事を投稿する際に利用できるアプリケーションである。インターフェースは異なるが、Web版投稿アプリと同等の機能を持っている。

#### 3.6.2 閲覧アプリ

閲覧アプリは、学内に配備する専用端末での情報表示に用いるアプリケーションである。そのため、専用端末での使い勝手を重視して作成されている。また、構造上キーボードを介した入力ができないのでユーザ認証には専用のものを用意している。

以下に本アプリケーションとWebアプリとの差を列挙する。

**操作ボタンの大型化** タッチパネルでの操作なので, 各ボタンを大きくして操作性を向上した

**ポップアップ式の記事表示** ひとつの記事を大きく表示し, 見やすさを向上した

**Felica によるユーザ認証** ユーザ認証には, Web アプリで用いているユーザ名・パスワード入力によるものではなく, Felica を用いた認証を採用した

#### 4 現状と今後の方針

現在は Windows 用閲覧・投稿アプリおよび Web アプリの閲覧・投稿部の開発を終えている。また, メールアプリも情報取得部分は開発を終えている。しかし, Windows 用の各アプリはデータベースの変更に対する追従ができておらず, これを行う必要がある。

今後は実際に本システムを配備し, 実運用における問題点の把握・解決を図ると共に, 使い勝手の向上なども行っていく予定である。専用端末の設置に際しては盗難・破壊への対策も講じる必要があると考えられる。また, メールアプリの認証方式の工夫や, 専用端末における Felica 以外のより容易に利用可能な認証方式の構築も行っていく。

#### 参考文献

- [1] モバイルキャンパス,  
<http://jp.fujitsu.com/group/fip/services/education/mobilecampus/>
- [2] 福島大学お知らせ配信システム,  
<http://kyoumu.adb.fukushima-u.ac.jp/fibs.htm>
- [3] 宮岡樹, 衣笠昭弘, 岡田正, Web ベース業務効率化システムのオープンソースソフトウェアによる開発, 津山工業高等専門学校紀要, 第 49 号, 2005, pp.59-66
- [4] 株式会社アイシェア, アイシェア意識調査「携帯電話でたまに使って便利な機能は?」,  
<http://blog.ishare1.com/press/archives/2005/05/271235.html>
- [5] RFC2821, <http://tools.ietf.org/html/rfc2821>