

# 超小型SNS「s4」による教育用 情報システム導入過程の効率化

広瀬 雄二

東北公益文科大学総合研究論集第33号 抜刷

2017年12月20日発行

# 超小型SNS「s4」による教育用 情報システム導入過程の効率化

広瀬 雄二

## 概要

近年教育現場では、インタラククションなどを通じた受講者の積極的な参加や、蓄積しておいた成果物を見返してその後の成長の方針策定に役立てることが重視されている。とくに後者に関しては、電子的に成果物等の蓄積と参照等を行なえるシステムが「eポートフォリオ」という呼称で一般化し、著名なシステムも広まりつつある。

しかしながらそれらのシステムが、組織にとって本当に必要とするものなのかは実際に体験してみないと分からない上に、導入も容易でない場合が多い。導入できた場合でもその後のシステム、依存ライブラリ、データベースすべての更新に留意が必要で、作業コストを上まわる利点が見込めない場合には導入を見送らざるを得ないこともある。

本稿では、そうした支援ツール導入前の事情をふまえ、必要とする依存ライブラリが最小限で、特定環境に合わせたカスタマイズを行なう際にもデータベースの再構築が不要な機構を持った支援ツールs4について提案する。

## Abstract

ICT aided education has been focused on these days. Many educational institutions have introduced systems which can be classified into 'SNS', 'LMS' (Learning Management System) and 'ePortfolio' to improve the efficiency of class work. Although there are many outstanding systems for those purpose, some systems, especially useful ones, require rich and complex library dependencies which discourage users those who do not have confidence of necessity for systems from trying to deploy such systems.

In this paper, we propose a highly miniaturized SNS named 's4', which is a collection of shell scripts, and prove the flexibility of database design s4

adopts that enables indecisive construction of an information system.

キーワード：SNS, LMS, ePortfolio, 情報システム, シェルスクリプト, SQLite3, プロトタイプ  
イピング

Keywords：SNS, Shell Script, SQLite3, Prototyping

## 1 背景

教育に関する手法は、より効果の高いものを目指して時代により様々なものが提唱されていて、近年ではそうした手法に情報技術が一翼を担うケースが増えている。たとえば文科省では平成26年度に大学教育再生加速プログラム[1]の中で、(1)アクティブラーニング、(2)学修成果の可視化、(3)入試改革・高大接続を行う取組を支援するとしているが、このうち最初の2つに対する取り組みとして、大学講義内で情報通信技術を利用した学習管理システム（Learning Management System; 以下 LMS）や、電子ポートフォリオ（Electronic Portfolio; 以下 ePF）システムの導入が進んでいる。

しかしながら、LMSやePFはそれ自体の導入に手間暇費用がそれなりにかかることや講義の進め方自体を変革する必要があるなどして、既存の授業にすぐに取り込むといったことは容易でない。このことは、普及させることへの工夫が複数の文献にあることから読み取れる[2, 3]。また、これらツールは電子的データでの情報授受を前提とするため、その入出力インタフェースでは伝達しづらいものを対象とする科目や、電子デバイスの使用が困難な科目などには適さない。「適さない」と分かれば導入しない決定が下せるが、そのことは実際に導入して試用してみなければ分からない、といったいわば鶏と卵の関係があるため、導入に対する資金的コストがかけられない場合には要不要の判断すら困難である。

そこで本稿では、導入および運用に関するコストや必要資源を極力なくし、電子的教育支援ツールが必要かどうかの導入試験が気軽に行なえ、かつある程度の規模内での実用が可能なシステム“s4”を提案し、東北公益文科大学（以下本学）で運用した実績とともに効果を検証する。

## 2 授業進行の補助となる電子的支援ツール

情報技術で進行を支援するツールにはいくつか分類がある。一般的な3種について、オープンソースであるLMSとSNSとePFの各代表例を1つずつ挙げる。なお、これら著名なツールの特徴についてまとめた文献も存在する[4]。

### 2.1 LMS

授業等での講師と学習者間の教材配付、答案回収、採点、評価などの工程支援に重きを置くのがLMSである。LMSの代表格であるMoodle[5]では、利用者の立場を「ロール」として定義し、利用者は自身のロール内で可能な操作を行なうことができる。たとえば「教師」というロールでは提出された課題の評点などを行なえるが「TA」というロールでは講義資料は編集できるが課題の評点はできない、といったことを設定できるのが特徴である。

### 2.2 ePF

近年有用性が謳われるようになってきたのがePFで、オープンソースでの代表的な実装がmahara[6]である。maharaでは「ポートフォリオ」としての機能が重視され、学生自身の学びの成果の蓄積と振り返りのしやすい設計となっている。課題などの提出された成果物のことを「アーティファクト」と呼び、単に一箇所に溜めるだけでなく「誰に見せるか」を制御する「ビュー」を各自定義して公開範囲を設定できるようになっているのが特徴である。

### 2.3 SNS

コミュニケーションが主目的のツールがSNS (Social Networking System) で、SNSを学習環境で利用しているケースも多い。その代表的な実装OpenPNE[7]では、ユーザ間の1対1の繋がりである「フレンド」という関係を築ける他、同じ興味を持った人同士で作れる「コミュニティ」というグループを作成でき、コミュニティ内での掲示板やイベント進行補助機能が提供されているのが特徴である。

表1 パッケージが依存するソフトウェア

|           | Moodle (2.x)  | mahara (15.10)   | OpenPNE (3.6)  |
|-----------|---|--|--|
| RDBMS     | MySQL 5.1.33 以上<br>PostgreSQL 8.3 以上  | MySQL 5.0.25 以上<br>PostgreSQL 8.3 以上                                   | MySQL 4.1 以上   |
| 言語        | PHP 5.3.2 以上  | PHP 5.1.3 以上   | PHP 5.2.11 以上  |
| PHP モジュール | iconv, GD, FreeType2,<br>mysql or pgsq, zlib,<br>mbstring, curl, tokenizer,<br>openssl, | gd, FreeType2, curl,<br>pgsql or mysql, mcrypt,<br>zip, imagick, adodb | mbstring, xml, pcre,<br>pdo, mysql, exif, mcrypt,<br>apc |

## 2.4 各ツールの得失

授業進行補助、という観点では課題の提出や回収・評価を主眼としたLMSが適しているように思われるが、実際には授業形態や進行形態で変わってくる。たとえば本学の科目「基礎プログラミング」では、課題提出と評価に関することはそれ以前からWebと電子メールベースで不便なく進めていたため、そうした支援はとくに必要なかった。支援を必要としていたのは、学生同士でのグループ活動の相互連絡の場と、それを評価者から見えるようにすることであり、そのためにはコミュニケーション主眼のツールであるSNSが最適と判断したため、2010年度にOpenPNEを導入した。もし別の授業形態・進行形態であればまた違った主眼のツールが必要になるであろう。

ある現場において、その場に適したツールがどれを主眼とするものなのかは一概に決められないことが多い。そのため、複数のツールを融合させるようなモジュールも開発されている。たとえば前述のMoodleとmaharaを統合するMahoodle<sup>\*1</sup>などがその好例である。

## 3 既存のシステム導入時の問題点

上述のツールを使いたいとしても、それだけ導入すれば済むわけではなく、稼動要件を満たすために複数のソフトウェアを事前導入する必要がある。さら

<sup>\*1</sup> <https://docs.moodle.org/29/en/Mahoodle>

には依存関係にあるソフトウェア\*2 (表1) に不具合が発見された場合の更新作業も少なからぬ負担を強いているのが現状である。いずれのツールも多数のPHPライブラリに依存しているため、どれか1つに不具合が見つかっただけで更新の必要が発生する。この問題は、PHPを利用するツール全般に共通したものである。

本学でもOpenPNEを導入した当初は、想定通りの効果だけでなく、消極的だった学生の自己表現が見られるなどの、予想外の満足できる効果も得られ、継続利用のため更新作業を重ねた。しかしながら作業には以下のような負担があった。

- PHP+Apache httpd運用の習熟度不足からくるシステム更新の効率の悪さ
- 本体システム更新による既存データ破壊への不安
- RDBMS更新による既存データ破壊への不安

支援ツールの機能から得られる恩恵も大事だが、こうした「更新苦業」からの解放も重要な要素と捉え、以下のような条件を満たす新規システムが望まれた。

- SNSタイプであること  
新規に構築するのであれば、LMSやePFからのアプローチも考えられたが、それらの提供する主な機能の一部は他のツール代替できている状態であり、学生間コミュニケーションのための機能の必要性が最も高いことから、SNSをベースとするものが望ましい。
- シンプルであること  
依存するソフトウェアや言語処理系の種類が最低限であり、かつそれら自身のセキュリティ問題発生頻度が高くないものが望ましい。
- 可搬性にすぐれること  
様々なOS環境で動かすことが容易なものが望ましい。
- コンパクトであること

---

\*2 とくにPHPによるソフトウェアは、Webサーバ上のファイル書き換え可能な権限を持たせて動作させることが一般的で、PHP処理系のセキュリティホールは深刻な被害に至りやすい。

ソフトウェアの複雑さはそのサイズに連動して増す。必要最低限の機能を必要最低限のコードで実装することが望ましい。

## 4 システム要件

先行利用環境であるOpenPNE（3.8.12）で利用度の高かった機能、用途との違いから改善の必要を感じていた仕様、その他の講義・演習を進める上で必要性を感じていたもののうち、必須と思われる部分を抽出し、設計仕様にまとめていった。

### 1. 招待制

一般的なSNSはインターネット経由で自由に「自分で」アカウントを作成して利用できるいわば解放空間になっているが、学習に関する情報を交換するためのものは、誰でも自由に加入できてはまずく、閉じた空間である必要がある。その点OpenPNEの持つ、招待制アカウント作成の仕組みは、

- つながりのあるものだけが加入できる（部外者が入りにくい）
- 管理者のアカウント作成の手間を省ける

という2点において優れているため、これを踏襲することとした。

### 2. グループ作成機能とチーム作成機能

演習科目ではグループワークを課すことがある。その際に、グループ間での連絡が必要になっているが、学生同士はとくに指定をしないとLINE<sup>\*3</sup>でのID交換を始める。しかしながら、本当に親しい相手にしかIDを教えたくない学生やそもそもLINEを避けたい者が尻込みする様子を見たり、LINEのみにコミュニケーションを頼ることの危険性の指摘[9]などがあることから、目的に応じた空間とツールの使い分けを教育する機会も必要と判断した。そのために、OpenPNEにも存在したグループ機能が必須である。

OpenPNEだけでなくmaharaにもグループ作成機能があるが、いずれ

---

\*3 本稿執筆時時点で国内約7000万人[8]の登録ユーザをもつ電子的コミュニケーションアプリ。

も同一の名前空間での命名であるため名前の衝突が起り得るため命名に注意が必要になる[2]。実際の演習科目等では、「Aチーム、Bチーム」あるいは「1班、2班」といった曖昧な名前付けで済ませたいこともあるため、グループ内でのみ通用する「チーム」の名前を付ける機能があると有用である。

### 3. 掲示板機能

先行利用環境にもあったが、SNS特有の掲示板をグループごと、あるいは個人ごとに作れる機能が必要である。

### 4. 書き込み通知機能

前項と関連し、チーム分けした活動では、メンバー相互に連絡・報告が必要である。チーム所有の掲示板に書き込みがあったときに、必要であれば所属者に通知メールを送る機能が必要である。

### 5. 全文検索

先行利用環境ではSNS全体からの検索機能がないため、せっかく書いた有用な記事も、書いた場所を忘れると探すのに苦勞することがあった。SNSに有益な情報が集まるようになることを見越すと、全文検索の機能は必要と言える。

### 6. 任意の種類ファイル添付

先行利用環境では投稿記事に添付できるのは写真のみであり、PDF文書やプログラムソースなど、他の形式を許可するにはカスタマイズする必要があった。不特定多数の利用者を想定する場合はセキュリティ重視でファイル種別を限定するのは妥当だが、講義等では自由である方がよいと考えた。

### 7. 提出物の集計

レポート等をファイルで受け取る場合は、グループ管理者（たとえば講師など）から未提出者一覧が一瞥できるようになっていると採点作業の助けとなる。

これらを保有するものを設計した。



## 5 システムの設計と実装

4で述べた要件を満たすためのシステム設計について、データベース設計の方針とUI画面、実装環境について示す。

### 5.1 データベース設計

RDBを利用する一般的なシステム設計では、あらかじめ格納すべきデータを分析し、リレーショナルモデルに適合する形に変形（正規化）し、格納テーブルの設計を確定させる。一度決めたテーブルの設計の変更は、それを処理する部分にも影響が及ぶため、通常はコストを掛けて高度な経験を有するシステムエンジニアと面談を重ねて構築していく。

話を前提に戻すと、教育支援系の情報システムの導入の可否そのものを判断する場合にはそもそも要求仕様導出作業の発注以前の段階であるため、既存パッケージを試験的に導入したいところである。ところが、導入作業が複雑なものである場合はそれ自体が負担となる。

以上のことを踏まえ、本稿で提案するシステムs4（SNS by Shell Script and SQLite）

表2 ユーザの持つ属性

|             |                   |
|-------------|-------------------|
| <u>name</u> | 一意に定まる名前          |
| pswd        | ハッシュ化されたパスワード文字列  |
| gecos       | 実名あるいはニックネーム      |
| skey        | セッションキー           |
| email       | 連絡先emailアドレス（複数可） |
| address     | 住所・居場所            |
| profimg     | プロフィール画像（複数可）     |
| desc        | 自己紹介の一言           |
| profpdf     | 自己紹介となる文書         |

表3 第1正規化後のユーザ情報格納テーブル

|             |                  |               |                      |
|-------------|------------------|---------------|----------------------|
| user テーブル   |                  | emails テーブル   |                      |
| <u>name</u> | 一意に定まる名前         | <u>name</u>   | users テーブルの name     |
| pswd        | ハッシュ化されたパスワード文字列 | email         | 連絡先 email アドレス (複数可) |
| skey        | セッションキー          |               |                      |
| address     | 住所・居場所           | profiles テーブル |                      |
| desc        | 自己紹介の一言          | <u>name</u>   | users テーブルの name     |
| profpdf     | 自己紹介となる文書        | profimg       | プロフィール画像 (複数可)       |

では「試験導入段階では集計すべき項目が未確定」との前提に立ち、テーブル設計を確定させなくてもよい定義形式を採用する。説明のため、本システムで取り扱う「ユーザ」を例に示す。

### 5.1.1 正規化前提の難しさ

本システムの利用者を表す「ユーザ」はいくつかの属性を持ったものである。表2中の「(複数可)」のある部分は、同一属性に複数の値を持たすものでありこのままではテーブルには入らないため正規化して別のテーブルに格納する。この例であれば表3のように複数のテーブルに分ける。

問題は、このようなテーブル設計をしたとして、項目追加などの修正の必要が出た場合である。「(複数可)」の項目追加をする場合はテーブルを増やす必要があり、それ以外の項目追加の場合にはusers テーブルの設計を変える必要がある。この必要をなくすため、本システムではRDBMSとしてテーブル定義でのカラム型の自由度が高いSQLite3<sup>\*4</sup>を採用することとし、あいまいなテーブル設計のままシステム構築できる以下の方式を提案する。

### 5.1.2 ハッシュ + リストのエミュレーションテーブル

表3中のuser テーブルの主キーnameは必須カラムで一意に定まる値を持つ。その他の「パスワード」や「住所」は利用状況により必須か任意かが分かれる

<sup>\*4</sup> <https://www.sqlite.org/> 世界で最も利用されていると推測されるデータベースエンジン。スマートフォン (Android/iPhoneともに)、Windows10、MacOS、各種Linux、NetBSD、Firefox、chrome などありとあらゆるOS、ソフトウェアに内蔵されている。

が、いずれにせよ主キーに対応する値は1つのみという想定である。また email テーブル、profiles テーブルについては1つの主キーに対応する各値は0個以上が重複なく入るとの想定である。これらの関係を、以下のような3テーブルに分解して格納する。

### 1. 主キーテーブル

主キーと（もしあれば）外部キー制約を持つカラムのみを格納するテーブル

### 2. 単一値テーブル

主キーが持つ属性のうち、0個または1個の値を持つものを格納するテーブル

### 3. 複数値テーブル

主キーが持つ属性のうち、0個以上任意個の値を持つものを格納するテーブル

2と3のテーブルには「属性」を示すカラムと「その属性の値」を保持するカラムが必要になる。一般的なRDBMSではカラムには必ず固定的な型指定が必要だが、SQLite3には厳密な型指定はなく、「文字列優先」あるいは「数値優先」のようなAffinityが存在するのみである[10]。また、本システムの言語処理系であるshでは、基本的に原始型として文字列しか持ち得ないため、属性値をデータベースに格納する場合もすべて文字列として格納することで不都合がない。ただし、添付ファイルを扱う場合には、shが変数に保持できないバイナリ形式だけは区別する必要がある。以上を総合して2と3のテーブルにはテキストの場合に値を保持するカラムと、バイナリの場合に保持するカラムを設定し、格納すべき値によって使い分ける設計とした。

以上のような構成に組み直して設計したものを表4に示した。FKは外部キー制約、(U)はname, key, val 3つの組み合わせでのUNIQUE制約を示している。user\_s テーブルではname, keyの組み合わせでの複合主キーとなっているので、それらは同時にUNIQUE制約も満たす。この2つのテーブルの制約で格納できる値をプログラミング言語のハッシュ + リスト風に表現すると以下の

表 4 ms 展開した属性保持テーブル群

| user テーブル            |                        |                    |
|----------------------|------------------------|--------------------|
| <u>name</u>          | 一意に定まる名前               |                    |
| user_m テーブル (複数値格納用) |                        |                    |
| カラム                  | 意味                     | 制約                 |
| name                 | users テーブルの name       | (U), FK(user.name) |
| key                  | 属性名                    | (U)                |
| type                 | 値の型                    |                    |
| val                  | 文字列の場合の値/バイナリの場合はファイル名 | (U)                |
| bin                  | バイナリの場合の値              |                    |
| user_s テーブル (単一値格納用) |                        |                    |
| カラム                  | 意味                     | 制約                 |
| <u>name</u>          | users テーブルの name       | FK(user.name)      |
| <u>key</u>           | 属性名                    |                    |
| type                 | 値の型                    |                    |
| val                  | 文字列の場合の値/バイナリの場合はファイル名 |                    |
| bin                  | バイナリの場合の値              |                    |

ようになる。

- user\_m テーブル → val がベクトル (配列)  
name, key, val の val が異なれば何個でも可:  

$$\{name \Rightarrow \{key \Rightarrow [val_1, val_2, val_3, \dots]\}\}$$
 例:  $\{ "yuuji" \Rightarrow \{ "email" \Rightarrow [addr_1, addr_2] \} \}$
- user\_s テーブル → val がスカラ (単一値)  
name, key に対して val は 1 個のみ:  

$$\{name \Rightarrow \{key \Rightarrow val\}\}$$
 例:  $\{ "yuuji" \Rightarrow \{ "pswd" \Rightarrow "$2a$10$x4zkd225aU..." \} \}$

この関係を利用すると、ハッシュ (連想配列/辞書) とリスト (配列) の多

```

1 ユーザ名(英数字):name:p:user:length="20" maxlength="40"
2 パスワード:pswd:s:password:length="20" maxlength="40"
3 なまえ:gecos:s:text:length="20" maxlength="40"
4 セッションキー:key:s:session
5 通知送信メールアドレス(複数可):email:mh:text:length="20" maxlength="40"
6 住所:address:m:textarea:maxlength="400"
7 プロフィール画像:profimg:m:image:maxlength="400K"
8 ひとこと:desc:s:textarea:maxlength="1024"
9 履歴書(4MBまで):profpdf:m:document:maxlength="4M"

```

図1 格納情報定義ファイル user.def

重組み合わせで表現できる構造は表4のような3つのテーブルの多重組み合わせで格納できる。なお、上記のように1つの主体に関する属性値を、「主キーテーブル」、「複数 (multiple) 値用テーブル」、「単一 (single) 値用テーブル」の3つのテーブルに展開して格納する方式を本稿では便宜上「ms展開」と表すことにする。

ところで、このように属性と値の組をカラム組に持たせる方法をEAV (Entity Attribute Value) と呼び、本格的設計には用いないのが望ましいとする議論もある[11]。主な理由としては「必須属性が設定できない」、「SQLの型が使えない」、「値に対する制約が使えない」、「行の再構築に多重結合が必要」があるが、今回のSNSを目的としたものでは、「主キー以外は絶対に必須というわけではない」、「今回のSQLite3+shではそもそもすべて文字列」、「制約が必要なものはラジオボタン入力等で対応可能」、「行の再構築は定式化されたMAX関数、CASE式と集約による文で結合なしに可能<sup>\*5</sup>」といった理由から不利益はほぼないといえる。

### 5.1.3 ms展開テーブルを想定したデータ構造定義記法

格納したいデータの主キーは何か、その他の属性の値は1つに限られるものか否か、などを列挙することで自動的にRDBのスキーマに変換できるように

<sup>\*5</sup> 「縦持ちから横持ちへの変換」などと表現される効率的手法が知られている。

表5 抽象エントリタイプの例（抜粋）

| タイプ           | 意味                          |
|---------------|-----------------------------|
| user          | 利用者を表す。認証を行なう単位である。         |
| group         | userの集合体を保持する。              |
| password      | userのパスワードとして振舞うもの。         |
| session       | 認証状態継続のためセッションキー。           |
| owner         | 何かの所有者を表す値でuserを外部キー制約とする値。 |
| text          | HTMLのtextフォームでの入力文字列。       |
| image         | 画像として処理すべき入力値。              |
| radio, select | 択一入力処理されるべき値。               |
| *tbl          | 別のms展開テーブルtblを値として持つ。       |

構造定義記法を設計する。その際、データ入力がWeb画面から行なわれることを考え、ある属性値のWeb画面での入力時の情報も持たせるような書式とする。まず表2に示したようなユーザ情報の入出力・保存に関する定義ファイルの実例を図1に示す。これは、1行1単位の半角コロン（:）区切りの書式であり、カラムの並びは左から順に、

1. Web画面での値入力時のプロンプト文字列
2. データベース格納時のカラム名
3. 値の保持形態（p=主キー、m=複数值属性、s=単一値属性）
4. 抽象エントリタイプ
5. Webフォーム入力時の追加パラメータ

を意味する。4の抽象エントリタイプは、その値がシステム中で持つ特別な典型的機能（表5）を表すもので、変数の入力法と処理方法をまとめて指定できるいわばプリセット型である。

実際のデータベースでは、定義ファイルのベース名がテーブル名になる。この例（`user.def`）の場合`user`、`user_m`、`user_s`とms展開された3テーブルが適切な制約とともに作成される。もし、本システムを利用しているうちに、ユーザの属性情報を増やしたいとなった場合は、`user.def`にそのための行を追加するだけでよい。

表6 s4の実装環境

|              |                                  |
|--------------|----------------------------------|
| Webサーバ       | Apache HTTP Server (CGIが利用できるもの) |
| 言語処理系        | Bourne Shell, Perl               |
| データベース管理システム | SQLite3 (外部キー制約の利用できるバージョン)      |

## 5.2 UI画面

利用者が目にする情報入力ユーザインタフェースは標準的なHTML5[12]の入力フォームのみを用いた。また、入力画面のデザインはCSS Level3の規則のみを用い、JavaScript等他の言語処理系要素は組み込んでいない。したがってテキストブラウザ等でもアクセスできるデザインとなっている。

## 5.3 実装

複数人が同時アクセスすることを前提としたWeb上で動くシステムとするためには、少なくとも「Webサーバ」、「言語処理系」、「排他機構を持つデータベース管理システム」が必要である。POSIX<sup>\*6</sup>準拠のシステムであれば動かせる程度のもとするため、表6のような実装環境に定めた。

言語処理系として最低限を目指すため、あらゆるOSに標準装備されたBourne Shell (以下sh) を選択した。shはUnix系OSの初期から存在し、新たな機能追加がほぼ為されず完全に「枯れた」極めて信頼できる処理系である。当初、shのみでの実装を目指したが、標準的機能だけではバイナリデータの読み書きができないため、これを補助するためだけにPerlを用いることとした。なお、shを利用するということは主な処理を外部コマンドに負うこととなる。使用しているのは head, tail, cat, sed, (f)grep, m4, awk, tr, cp, mv, rm, basename, dirname, date, file, mktemp などの標準的コマンドと、nkt<sup>\*7</sup>, ImageMagick<sup>\*8</sup>な

<sup>\*6</sup> <http://standards.ieee.org/develop/wg/POSIX.html>- OSの基本的な機能を共通化して定めた規格のこと。

<sup>\*7</sup> Network Kanji-code Filterの略で、1987年当初は漢字コードのみの変換装置であったが多くの人が開発に参加し、後にMIMEストリームへの変換機能などを備え、文字のエンコードに関する処理をほぼ一任できるようになった。

<sup>\*8</sup> <https://www.imagemagick.org/> - 画像の変換処理を行なうコマンドとライブラリ群。

表7 各システムの量的規模の比較

| システム             | 総容量 <sup>*9</sup> | ファイル数 | ソースプログラム行数 |
|------------------|-------------------|-------|------------|
| Moodle (3.3.2+)  | 168MB             | 16813 | 1116271    |
| Mahara (17.04)   | 76MB              | 7015  | 678501     |
| OpenPNE (3.8.25) | 29MB              | 5890  | 466915     |
| s4 (r475)        | 0.3MB             | 48    | 5351       |

どの歴史を持ち複数のOSに移植されているものである。その他、バイナリ・ASCII文字列相互変換やダイジェスト文字列生成用の外部コマンドも利用しているが、いくつかの候補からシステムに存在するコマンドの組み合わせを選ぶようにしている。

## 6 システムの評価

先述の検討をふまえて実装したs4について評価を試みる。ここでは、当初の目標で重視していた

- シンプルな実装となっているか
- プロトタイプ的な実装で実用できるのか
- 格納データの属性追加等に対応可能か

といった観点に絞って評価する。

本稿で設計・実装したシステムs4は、依存ライブラリ・モジュールが極力少なくなることを目指した。シンプルであるがゆえ当然のことではあるが、実装プログラムの本体サイズは、本稿で取り上げた他のシステムの100分の1未満程度に小さくなった（表7）。もちろん、長い歴史を持ち貢献者も多いシステムとの比較であるから圧倒的な差がつくのは明らかであり、比較自体は意味がとぼしい。重要なのは、万人の要望に応えることを目指して作られたソフトウェアはすべからく大きなものになるということと、逆に単一の組織が必要と

<sup>\*9</sup> ファイルを展開するとファイルシステムのブロックサイズで容量に差が出るため、非圧縮状態のtarアーカイブのサイズを調査した。





図2 s4のホーム画面

表8 s4 利用状況の推移

| 月日        | ユーザ数 | グループ数 | 掲示板数 | 記事数   | DB 容量(MB) |
|-----------|------|-------|------|-------|-----------|
| 2015-0901 | 15   | 4     | 26   | 284   | 66        |
| 2015-1201 | 45   | 8     | 54   | 672   | 177       |
| 2016-0301 | 45   | 11    | 88   | 1250  | 355       |
| 2016-0601 | 55   | 15    | 121  | 1943  | 549       |
| 2016-0901 | 79   | 17    | 193  | 4295  | 1124      |
| 2016-1201 | 88   | 22    | 296  | 6395  | 1494      |
| 2017-0301 | 89   | 23    | 359  | 8961  | 2308      |
| 2017-0601 | 139  | 26    | 400  | 10449 | 2440      |
| 2017-0901 | 293  | 35    | 515  | 14246 | 2857      |

するものを最小セットから組み上げたものは圧倒的に小さくすむということである。

## 6.1 運用状況

本学においては、2015年度9月に始まる講義からs4の運用実験を開始した。実際に稼動している各ユーザのホーム画面が図2で、この画面に「自分の日記(ブログ) 一覧」、「参加しているグループ」、「自分の書いた新着記事」、「全体の最新記事」などが一覧でまとめられている。

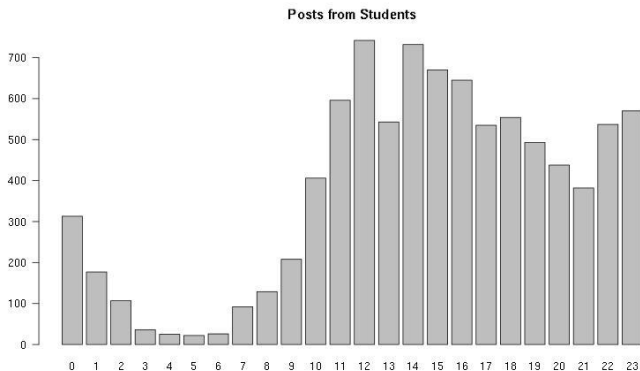


図3 一演習における時間帯別の受講者書込数

2017年9月1日現在、本学で稼働中のs4は登録学生281名を含む293名のユーザによる書き込み記事14246件を保持している（表8）。元来講義進行の補助目的での導入であるから、進行そのものが円滑化されたかの評価が必要とされるところであるが、システムなしの場合との比較などの条件不足で検討できる段階には至っていない。その中で一点、受講者（学生）のs4上での活動の様子が明らかになるデータを示す。

図3は2016年度本学の演習科目「プロジェクト型応用演習（さかたまっぷ）」における、対応するs4グループ内の掲示板書き込みを時間帯ごとにグラフ化したものである。この演習は時間割上は4限（14:40-16:10）に置かれていたが、グラフを見ると午後の時間帯に幅広く書き込みが分布していることが分かる。このことより、講義・演習の時間外にも自宅等でそれに対するなんらかの作業をしていることから、学外学習の時間が保たれていたといえる。主観的な記述にはなるが、s4の利用が演習進行の円滑化に繋がったことは確かで、シンプルな設計ではあったが十分に実用的であったといえる。

## 6.2 仕様変更追従性

s4でユーザが記事を書き込む場所である「掲示板」は2015年度の当初運用時の仕様では以下の4つの属性値を持つものであった。

# 修正

このエントリの削除

|          |  |
|----------|--|
| 用途       | 普通の掲示板   |
| タイトル     | 2017春学期成果発表会賞  |
| コメント書込通知 | 管理者のみに通知(グループの場合のみ)  |
| 稼働状態     | 稼働   |
| 序文       | 移動<br>凍結(新規書込停止) 置いて下さい。<br>添削+修正して新しいものに置き換えて行きま<br>  *テーマ *レジュメ *発表スライド <br> malフケゾー キップコーナー ○ ○ |

OK Reset

図4 掲示板の属性編集画面

用途 以下のいずれか:

- 一般的な記事の書き込みのみ
- レポート提出用 (相互参照可能)
- レポート提出用 (相互参照不可)

タイトル 掲示板自体につけるタイトル

コメント書き込み通知 書き込みがあったことを通知するメールを:

- 管理者 (掲示板の作者) に送る
- グループ加入者に送る
- 誰にも送らない

序文 掲示板の見出しとなる短い段落

しばらくこの4属性で運用していたが、本来書くべき掲示板ではなく使用済みの古いものに間違えて書き込む事例が散見されるようになり、これを防止するため、特定の掲示板に対して参照は可能だが新規書き込みはできないようにする機構が必要になった。書き込みの許可不許可自体の挙動はプログラム修正で対処することになるが、どちらのモードにあるかの情報を属性として掲示板自体に持たせる必要がある。そのために掲示板の属性定義ファイルに以下のような1行を追加した。

稼動状態:state:s:select:稼動=active凍結(新規書込停止)=frozen

この1行の追加により、自動的に掲示板そのものの属性を編集するUI画面にも変更のためのフォームが追加された(図4の「稼動状態」欄)。このように、データベース構造の再構築なしで機能追加や仕様変更等が行なえることが確認できた。

## 7 結論と展望

依存性を徹底的に排除し、処理系言語も仕様変更とは無縁の言語(Bourne Shell)を選び、データ保存に対する安全性・完全性を確保できるリレーショナルデータベース(SQLite3)を採用しつつも保存すべきデータ属性の仕様変更にも柔軟に対処できる構造を採用した本稿提案システムs4は、実用にも値する速やかな試行環境の提供と、格納データの仕様確定の助けとなる機能を提供できることが確認できた。

今後の展望としては機能や操作性を改善することが考えられる。

### 7.1 機能に関する検討

s4はSNSを基本としているため、LMS由来の機能とePF由来の機能には欠けている部分が多い。これらは必要性があれば順次実装して行くという流れしかないが、s4の設計目標の原点に立ち帰ると支援ツールの必要性があるかどうか明確になっていない段階での試用環境の提供という目標を損なわない範囲、つまり初めて導入し、初めて使う人に「複雑さ」を感じさせない範囲での機能追加を進めなければ元の黙阿弥といえる。手軽に利用したい場合に必要となる機能は実装し、それ以上のものは本格的な既存システムに誘導するのが望ましいといえる。

### 7.2 操作性に関する検討

機能追加に関連するが、s4はUIの部分の設計・実装をすべてHTML、CSS、CGIの範疇に納めている。近代的なWebアプリケーションでは非同期通信と

JavaScript を利用した“Ajax”と呼ばれる機構の組み合わせで作成されているものが多く、それらと比べるとひけを取る部分が多い。とくにCGIのみでは、表示されるWebページ画面のすべての内容を受け取る必要がある。このことは、掲示板講読における非効率性に繋がる。つまり、既存の掲示板の書き込み内容をすべて表示している状態で、1つの書き込みを追加しただけでも、それを読むためには過去の書き込みすべてを受け取る必要がある。Ajaxを利用すれば最新の書き込みだけを受け取って、現在見ている画面に追加できるため、高速化され通信量の節約にも繋がる。しかしながらそうした改善は依存環境の追加に繋がるため、追加によるデメリットに留意しながら進める必要がある。

## 8 おわりに

s4での進行に協力してくれた該当科目の履修者すべてに感謝する。なお、本稿で設計・構築したシステム s4 は以下の URL にて公開、配付している。

Webページ <https://www.gentei.org/~yuuji/software/s4/>

Mercurialリポジトリ <https://www.yatex.org/hgrepos/s4/>



## 参考文献

- [1] 文部科学省. 平成26年度「大学教育再生加速プログラム」公募要領. [http://www.mext.go.jp/component/a\\_menu/education/detail/\\_icsFiles/afieldfile/2014/04/08/1346355\\_1.pdf](http://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/afieldfile/2014/04/08/1346355_1.pdf), April 2014.
- [2] 大嶋良明. 学部科目への ePortfolio の活用について：立ち上がりの実施報告. 異文化. 論文編, No.15, pp. 137-153, April 2014.
- [3] 松本亜実. 「シラバスの可視化」及び「自己評価ルーブリック」利用による学習意欲と修得内容向上のための取り組み. *Bulletin of Okazaki Women's*

University, Okazaki Women's Junior College, Vol.49, pp. 65-74, March 2016.

- [ 4 ] サイエнтиフィック・システム研究会 eポートフォリオ研究 WG. eポートフォリオ研究 WG成果報告書. Technicalreport,サイエнтиフィック・システム研究会, 2015.
- [ 5 ] M. Dougiamas and P. C Taylor. Moodle: Using learning communities to create an open source course management system. In *Proceedings of the EDMEDIA 2003 Conference, Honolulu, Hawaii*, 2003.
- [ 6 ] Mahara Project. Mahara ePortfolio System. <https://mahara.org>, (Accessed 2017-09-12).
- [ 7 ] 手嶋屋. OpenPNE. <https://www.openpne.jp>, (参照 2017-09-14).
- [ 8 ] LINE株式会社. 2017年12月期第2四半期決算説明会資料. [https://scdn.line-apps.com/stf/linecorp/ja/ir/library/Q2\\_presentation\\_JP.pdf](https://scdn.line-apps.com/stf/linecorp/ja/ir/library/Q2_presentation_JP.pdf), July 2017.
- [ 9 ] 若本純子. 高校生の LINE使用と, いじめ, 友人関係, 心理面との関連: 仲がよい3人の友人とのコミュニケーションに着目して (臨床, ポスター発表E). 日本教育心理学会総会発表論文集 日本教育心理学会第56回総会, p. 634. 一般社団法人 日本教育心理学会, 2014.
- [10] SQLite.org. Datatypes In SQLite Version 3. <https://sqlite.org/datatype3.html>, (Accessed 2017-09-11).
- [11] Bill Karwin. *Avoiding the Pitfalls of Database Programming*. Pragmatic Bookshelf, 2010. (和田卓人 和田省二 監訳, 児島修 訳. SQL アンチパターン 第5章. オライリー・ジャパン. 2013年).
- [12] The World Wide Web Consortium. HTML5 A vocabulary and associated APIs for HTML and XHTML W3C Recommendation 28 October 2014. <https://www.w3.org/TR/html5/>.