

2011年度 卒業論文

# プロジェクト管理ウェアの機能拡張

大阪産業大学 工学部 情報システム工学科  
情報教育システム研究室

08H081 津川翔丞

# 目次

1	はじめに	2
2	本研究の目的	3
3	プロジェクト管理ウェア	4
3.1	種類	4
3.1.1	Web ベース型 と ローカル型	4
3.2	Redmine	4
3.3	類似ウェア	5
4	本研究で提案するシステム	6
4.1	動作環境	6
4.2	開発環境	7
4.3	仕様・動作	7
4.3.1	Twitter Plugin	7
4.3.2	SmartPhone UI Plugin	11
5	考察	14
5.1	利点	14
5.1.1	スマートフォン向けページの提供	14
5.1.2	Twitter Plugin による TimeLine の抽出と保存	14
5.2	問題点	14
5.2.1	SmartPhone plugin の機能	14
5.2.2	TimeLine 抽出方法	14
5.3	今後の改善点	14
5.3.1	SmartPhone Plugin への機能追加	14
5.3.2	TimeLine 抽出方法を自動化へ	15
6	まとめ	16
付録 A		19
A.1	SmartPhone Plugin	19
A.2	Twitter Plugin	23

# 1 はじめに

プロジェクト管理システムを用いてチームプロジェクトを進行をする事が増えている傾向である。システム開発を成功させるためにはプロジェクトを適切に管理する必要がある。プロジェクト管理システムは多種多様であるが、近年は Web 型オープンソースのシステムを用いることが多い。このようなシステムは、導入コストの低さ、Web を用いて遠隔地でも利用できるメリットがある。

近年、誰もがスマートフォン<sup>\*1</sup>と呼ばれるモバイル端末を手にし、日常生活の必要ツールの一つになってきた。そこで、遠隔地でプロジェクトに参加するためスマートフォンを用いる機会が増えると考えられる。Web 型のプロジェクト管理システムの多くは PC 向けブラウザ<sup>\*2</sup>に最適化しており、スマートフォンブラウザ<sup>\*3\*4</sup>で多くのページは、文字や図が小さく表示され利用しにくい問題点がある。スマートフォンを生かした直感的かつシンプルな操作<sup>\*5</sup>な UI を提供することにより遠隔地でのプロジェクト参加を目指す。

近年、Twitter [1] や Facebook [2] など SNS<sup>\*6</sup>の発展により、タイムリーに情報を引き出すことが可能になった。プロジェクト管理システム環境下では、メンバー内での情報共有がメインになり閉鎖的である。プロジェクト内でも SNS からの情報をタイムリーに提供すれば、ホットな情報を開発メンバー同士で共有し、情報共有の視野が広がると考えた。本研究システムはプロジェクト内で Twitter 連携を目指し、プロジェクトツールの一部として利用を促進していく。

---

\*1 AndroidOS 搭載端末や iPhone

\*2 Firefox や Google Chrome など。

\*3 Android 標準ブラウザ

\*4 iOS の Safari

\*5 タッチパネル操作が簡易

\*6 Social Network Service

## 2 本研究の目的

本システムでは、Redmine [3-5] という Ruby on Rails<sup>\*7</sup> [6] で動作しているプロジェクト管理ウェアを用いる。Redmine の拡張プラグインを作成することによって、既存の Redmine を拡充していく。Redmine の初期インストールの状態では、スマートフォン用のブラウザでの表示には最適化していない。まず始めに、Redmine に本研究で開発したプラグインを用いて、スマートフォン用ブラウザに最適化したページを提供する。次に、Twitter との連携を測りタイムラインから情報を収集し Redmine プロジェクト内での共有を目指した。

スマートフォン用最適表示プラグインは、スマートフォンのタッチパネルで簡易的にチケット<sup>\*8</sup>が情報操作できるようにするのが目的である。Twitter 連携プラグインでは、公開 TimeLine から任意の検索ワードとプロジェクトに関連するツイートを取得し、プロジェクト参加ユーザが必要な情報を保存し、表示し、TimeLine を共有することが目的である。プロジェクト内で TimeLine の共有をすることによって、チケットや wiki に関連している情報を拡充していくことが期待できる。この 2 つの Plugin の作成し、既存<sup>\*9</sup>の Redmine でも利用出来るように互換性に配慮してシステムの構築を行う。

---

<sup>\*7</sup> Ruby で書かれている Web Application Framework である。

<sup>\*8</sup> プロジェクトごとのチケットシステム

<sup>\*9</sup> Redmine Ver 1.2.0 以降

## 3 プロジェクト管理ウェア

プロジェクト管理ウェアは、いろいろな機能が提供されている。様々な機能が存在しているが、主な機能として以下が列挙できる。

- 人員管理  
プロジェクトに携わる人を登録し、管理するための機能である。これはタスク管理と連携して作業を割り当てたり、課題管理の報告者や担当者の割り当てに使う。
- マイルストーン  
開発プロジェクトの水準を決められたスケジュールまでに達成するための目標を設定し、リリースする段階を決めたり、デバッグの完了時期を定めるのにも使う。大枠なスケジュールを策定するのに利用する機能である。
- チケット管理
- スケジュール・進捗管理  
主に、メンバーの開発予定工程を設定し、カレンダー形式で表示できる。マイルストーンよりも事細かくスケジュールを設定し、各メンバー同士で進捗管理にも役に立つ機能である。
- タスク管理
- リポジトリビューワ<sup>\*10</sup>  
プログラムソースコードの共有を行う。プロジェクトに関連しているリポジトリを登録することによりソースコードを共有でき具体的な開発状況を把握できる。
- ファイル共有
- コンテンツ (wiki・フォーラム)

### 3.1 種類

#### 3.1.1 Web ベース型 と ローカル型

プロジェクトウェアを大きく分けて、Web ベース型とローカル型の 2 種類の動作するものがある。最近では Web ベースのものが人気があり、ローカルマシンでのインストールが不要であること、インターネットを通じて複数の拠点間でデータを共有できる。本研究で用いる Redmine は Web ベース型に分類される。

### 3.2 Redmine

本研究では Redmine を用いる。Redmine は、初期インストールの時点では以下の機能が提供されている。

- チケット管理
- ガントチャート
- スケジュール管理
- フォーラム
- Wiki
- リポジトリビューワ
- プロジェクトメンバーの管理
- ファイル共有

---

<sup>\*10</sup> Git や Subversion など。

### 3.3 類似ウェア

プロジェクト管理ウェアとして以下のシステムが存在する。

#### Basecamp [7]

37signals [8] が提供するプロジェクト管理ウェアである。

マイルストーン・Todo・スケジュール・ファイル共有・ノートなどが主な機能である。リポジトリビューワーは提供されていない。

#### ZoHo Projects [9]

Zoho Projects は、有償のクラウドサービスでありマイルストーン・タスク管理・タイムシート・フォーラムを備えている。

ドキュメントをオンラインで編集できるのが特徴である。アップロードしたファイルは独自のバージョン管理を備えている。MS Project で作られたプロジェクトファイルをインポートできる機能がある。

#### Microsoft Office Project [10]

商用プロジェクト管理ウェアである。スケジュール管理・レポート作成・財務管理・ガントチャートといった機能がある。

主にプロジェクトマネージャが上層部への報告書を含めてレポートングするために利用されている。

#### OpenProj [11]

OpenProj は Linux/Mac OSX/Windows 用のオープンソースプロジェクト管理ウェアで、MS Project のファイルを開覧、編集できるのが特徴である。ガントチャートや PERT 図<sup>\*11</sup>、WBS<sup>\*12</sup>など多彩なグラフが描けるのも特徴である。予算の関係上、全員に MS Project をインストールできない場合にオープンソース・ソフトウェアの OpenProj で代用する場合もある。

#### The Trac Project [12]

Python で作られたオープンソースの Web ベースプロジェクト管理ウェアである。タイムラインやロードマップ、リポジトリブラウザ、チケット管理、Wiki といった機能がある。最近、Trac から Redmine ヘシステムに移行していく傾向がある。

---

<sup>\*11</sup> Program Evaluation and Review Technique

<sup>\*12</sup> Work Breakdown Structure

## 4 本研究で提案するシステム

この章では動作環境と開発環境、システム仕様について説明する。

### 4.1 動作環境

本システムは、さくらインターネット株式会社<sup>\*13</sup> [13] のVPS<sup>\*14</sup>上で動作している。

- Server 動作環境
  - OS .....Debian 5.0
  - Httpd .....Apache2
  - Database .....MySQL 5.0.51a
  - Language .....Ruby 1.8.7
- gem<sup>\*15</sup> [14] 一覧
  - actionmailer (2.3.11)
  - actionpack (2.3.11)
  - activerecord (2.3.11)
  - activeresource (2.3.11)
  - activesupport (2.3.11)
  - addressable (2.2.6)
  - faraday (0.7.5)
  - faraday\_middleware (0.7.0)
  - hashie (1.1.0)
  - i18n (0.4.2)
  - json (1.6.1)
  - multi\_json (1.0.3)
  - multi\_xml (0.4.1)
  - multipart-post (1.1.3)
  - mysql (2.8.1)
  - oauth (0.4.5)
  - rack (1.1.2, 1.0.1)
  - rails (2.3.11)
  - rake (0.9.2)
  - rdoc (3.9.4, 2.5.11)
  - rdoc-data (2.5.3)
  - simple\_oauth (0.1.5)
  - twitter (1.7.2)

---

<sup>\*13</sup> 個人レンタル

<sup>\*14</sup> Virtual Private Server

<sup>\*15</sup> Ruby の Library

## 4.2 開発環境

開発環境を以下に示す。今回の場合は仮想 OS<sup>\*16</sup>の環境下で開発を行った。

- OS.....Debian 6.0.3
- HDD.....5GB
- RAM.....512MB

## 4.3 仕様・動作

Redmine 上で動作するプラグインを開発した。Redmine と RubyOnRails の機能<sup>\*17</sup>を用いてプラグインの雛形を作成した。

### 4.3.1 Twitter Plugin

Twitter API を用いて Redmine Plugin の開発を行った。TimeLine から特定のキーワードやハッシュタグを検索することによって、プロジェクト参加ユーザーが情報収集ツールとして利用を推進させる。Twitter - Public API [15] を利用して、TimeLine の検索結果を Redmine 上で表示し、ユーザが任意の TimeLine をデータベースに保存することができる。さらに、プロジェクトごとにプラグインの機能を独立しているため、閲覧権限のあるユーザが保存した TimeLine を参照することができる。

- 仕様
  - TimeLine の最大検索結果件数は 100 件である。
  - プロジェクトごとに TimeLine の保存は識別している。
  - Public API を用いているため、API 制限が存在する。
  - 初期の検索ワードはプロジェクトの識別子<sup>\*18</sup>である。
  - プラグインの利用有無を切り替え可能である。
  - TimeLine 上の ScreenName と Image-icon も取得している。
- 動作 (実際の動作画面を図 1 に示す。)
  - プラグインの有効化 or 無効化。  
プロジェクトマネージャ<sup>\*19</sup>がプラグインの使用の有無を設定することができる。実際に設定画面を、図 2 に示す。
  - Twitter Time Line の表示。  
Twitter Api を用いて得た TimeLine の結果を図 3 に示す。検索結果は最大で 100 件を表示することができる。
  - TimeLine の保存。  
ユーザーが検索した Tweet を保存した結果を図 4 に示す。
  - 保存を行った Tweet の削除。
  - プロジェクトメニューに項目を表示。  
プラグインが有効になっている場合 Redmine 上のプロジェクトメニューに表示を図 5 示す。

---

\*16 OracleVirtualBox 上での動作

\*17 plugin\_generate

\*18 プロジェクト作成時に設定した識別子

\*19 Redmine 管理者またはプロジェクト管理者の権限があるユーザー

ログイン中: desucru 個人設定 ログアウト

検索:  プロジェクトへ移動...

redmineの拡張

Home マイページ プロジェクト 管理 ヘルプ

概要 活動 ロードマップ チケット 新しいチケット ガントチャート カレンダー ニュース 文書 Wild フォーラム ファイル リポジトリ Twit4red 設定

Default Search word: touhu-sotsuken

Search

tweetlist		tweet database			
	[AndriyLesyuk]:Want users to be able to use their google accounts on your #redmine or @chiliproject?.. http://t.co/oykenwov	[kanete]:RT @rakuten_tech: 楽天でのRedmineの活用事例です。 Rakuten Redmine http://t.co/PaFav5y3	<input type="button" value="Save"/>	<input type="button" value="delete"/>	<input type="button" value="ツイート"/>
	[nsmr]:Mantisとかもあつたけど現在のメジャーどころはtracがredmineあたりっべえのな	[gantawriter]:RT @akipli: このプラグインはありがたい。No Ticket No Commitの運用ルールは最終的には変更管理につながるから。Redmine Commit Relation Editor Plugin 0.0.1をリリースしましたhttp://t.co/sLc6rQJH	<input type="button" value="Save"/>	<input type="button" value="delete"/>	<input type="button" value="ツイート"/>
	[Vshp]:Tracやmantisのようなツールを、ソフトウェア開発だけでなく、業務管理に使っている企業も増えてきた。今は、tracやmantisのいいとこどりのようなredmineを使っているところが多い。私も使っているが、とても優秀なツール。	[m_e_]f:研究室のサーバ管理にさしあたってRedmineで個人の研究管理することを提案しておいた。ホントは自分の履歴にだけだからはやくどっかにデータ移譲したいだけという)	<input type="button" value="Save"/>	<input type="button" value="delete"/>	<input type="button" value="ツイート"/>
	[blna1204]:Redmineにプロジェクトやチケットを登録するなど。誰も使い方が分からないので、一度勉強が必要か	[YoshiWoods]:@xx_sakai Redmine に関する同僚の共著がベストセラーになっています。 http://t.co/wQ9ZChVK	<input type="button" value="Save"/>	<input type="button" value="delete"/>	<input type="button" value="ツイート"/>
	[avalanche_slide]:Redmine 1.3.0 チケット一覧画面のグラフなど特定の文字がchromeだけ文字化けする！? chromeさん何してますか...		<input type="button" value="Save"/>		
	[howndwolf]:本日はひたすらメールとredmineのチケットを切る機械になる。redmine、使いたい方わかってない人(自分含む)が利用すると、結構ひどい事になるのね...		<input type="button" value="Save"/>		
	[mori_dev]:Trac と redmine は似たようなものだけど、trac と pivotal は、いせやと肉いせやくらい違う。pivotalの方が好み。		<input type="button" value="Save"/>		
	[kyon_mm]:@linda re なるほどー。jenkinsとかgitとかredmineとかやりたいたいけど、開発とて内訳でいっぱいはいす><。tefのmiがいますごいことになっていて、返信がおいつかない><		<input type="button" value="Save"/>		
	[zumabot]:Redmine の新顔の円とは... あのかわいすぎる海女? 大向美咲さんら新人3人、海女を辞めてから wi-fi 同期いいな〜。私ははこーまとめおつです		<input type="button" value="Save"/>		
	[fukuchiharuki]:All-in-oneなredmine/パッケージ / "bitnami :: redmine" http://t.co/y3vfrtd		<input type="button" value="Save"/>		
	[antennae] 2011年12月10日、redmine 1.3.0がリリースされました。				

図1 本システムの稼働全体画面。TwitterTimeLineを左側に表示し、ユーザーが選択したTweetを右側に表示している。Searchしたい単語をフォームに入力するとTimeLineが更新される。

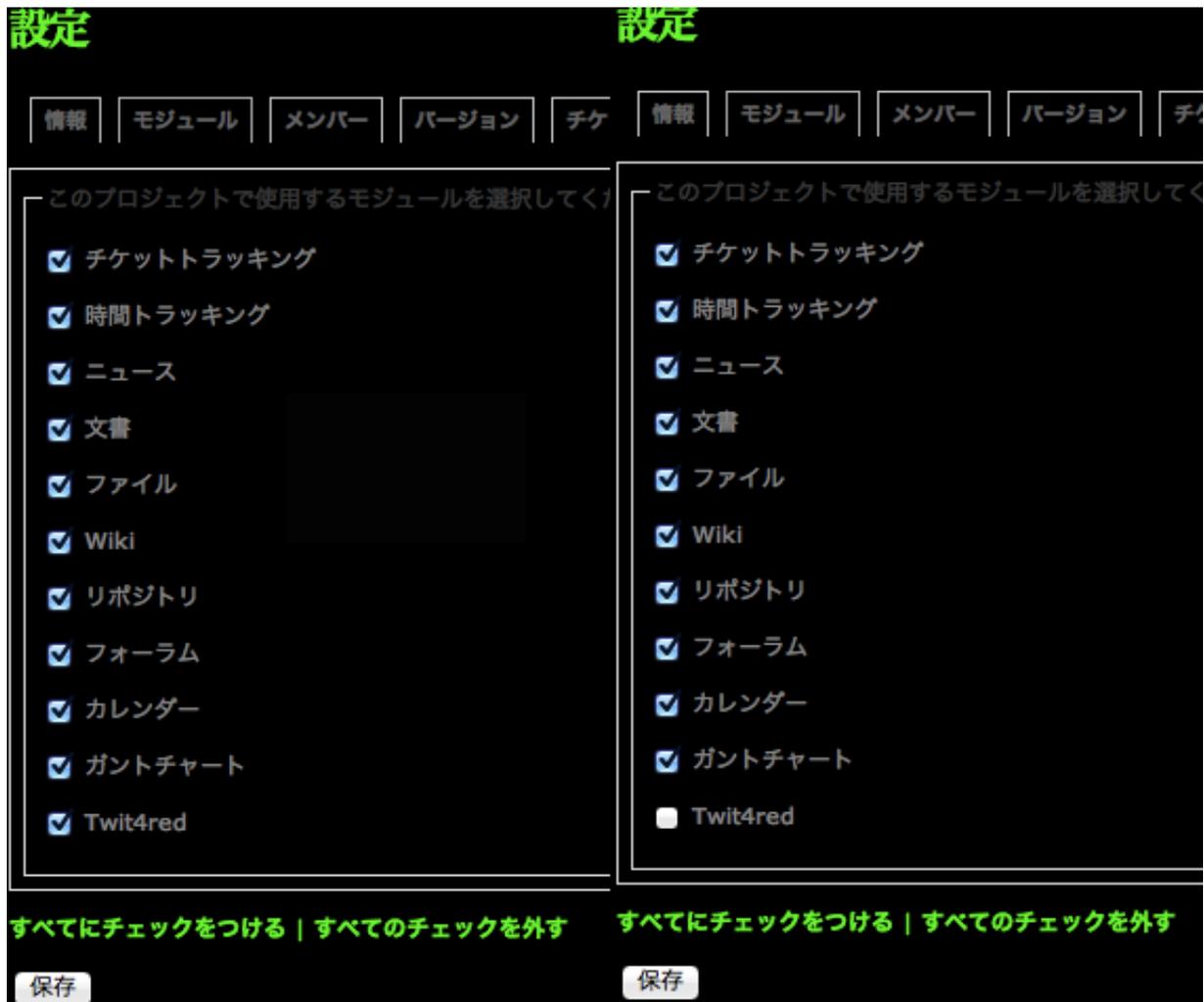


図2 プロジェクトで使用するモジュールの選択画面。プロジェクト管理者または Redmine 管理者がモジュールの使用を有効にすることができる。



図3 Twitter の TimeLine の表示。この例では、「Redmine」という単語を検索している。

tweet database		
[kanetel]:RT @rakuten_tech: 楽天でのRedmineの活用事例です。 Rakuten Redmine http://t.co/PaFav5y3	delete	ツイートする
[gantwitter]:RT @akipli: このプラグインはありがたい。 No Ticket No Commit の運用ルールは最終的には変更管理につながるから。 Redmine Commit Relation Editor Plugin 0.0.1をリリースしましたhttp://t.co/sLc6r0jH	delete	ツイートする

図 4 ユーザーが選択した Tweet の保存リスト。保存した Tweet はユーザーが自由自在に削除することができる。また、その場で Tweet も可能である。



図 5 プロジェクトメニュー一覧にプラグインのメニューを表示 プラグインが有効になっている場合にメニュー項目に表示される。

#### 4.3.2 SmartPhone UI Plugin

SmartPhone UI Plugin 開発では Layout を新たに作成をし、以下の仕様を策定し、jQueryMobile を用いてテンプレートを作成した。

- 仕様
  - jQuery 1.6.2 min
  - jquery.mobile-1.0a3.min.js
  - Ajax 機能は基本的に使用しない。
  - jQuery Mobile 導入時初期 css を利用。
  - OAuth は使用しない。
- 動作
  - ログイン\*<sup>20</sup>画面。  
ユーザー名とパスワードを入力するフォームを図 6 に示す。
  - 各プロジェクト一覧。  
実際に、参加権限があるプロジェクトリストを図 7 に示す。
  - チケット一覧。  
現在作成されている全チケット一覧を図 8 に示す。
  - チケット新規作成。  
チケット一覧画面からチケットの新規作成をすることができる。
  - チケット閲覧。  
チケット一覧画面からチケットを選択すると詳細情報が表示する画面を図 9 に示す。

---

\*<sup>20</sup> Public Project ならば、ログイン無でも利用可能。



The image shows the Redmine login interface. It features a blue header with the text "Redmine". Below the header, there are two input fields: "Username:" and "Password:". A large, rounded button labeled "ログイン" (Login) is positioned below the password field. At the bottom of the form, there is a blue footer containing the copyright notice "©Desucru".

図6 ログイン画面。Redmine に登録している User はログインが可能である。ログイン処理に成功するとプロジェクト一覧ページに遷移する。ログイン処理に失敗すると再度ログインフォームを表示する。



Redmine	
My Projects	
ひろぼん攻略プロジェクト	➤
とうふプロジェクト	➤
おがじゅん応援プロジェクト	➤
redmineの拡張	➤
kitazato - 卒研	➤
ひろきんプロジェクト	➤
skypebot	➤

図7 プロジェクト一覧。ユーザーが閲覧・編集権限があるプロジェクトリストを表示する。各プロジェクト毎のリンクからチケット一覧ページに遷移する。



図 8 チケット一覧。新しく作成されたチケットの順で表示を施している。各チケット毎のリンクからチケット詳細ページに遷移する。



図 9 チケット情報詳細表示。題名・チケットの種類・チケット作成者・チケットの説明を表示している。

## 5 考察

本研究で提案するシステムによって、Twitter 連携とスマートフォン向けページを提供した。

Android OS 搭載端末や iPhone を所有している研究室メンバーからのレビューを得た。プロジェクト管理ウェアに対する導入<sup>\*21</sup>の理解等の箇所は本研究では実施していない。

### 5.1 利点

本研究で開発した 2 種類のプラグイン毎に利点を示す。

#### 5.1.1 スマートフォン向けページの提供

本プラグインではチケットの閲覧・作成とチケット管理機能の最低限の機能を提供した。

スマートフォンを用いて遠隔地の環境<sup>\*22</sup>で、シンプルかつスマートな直感的な UI を使用することができる。

#### 5.1.2 Twitter Plugin による TimeLine の抽出と保存

各プロジェクトごとに関連するキーワードをユーザー自身が Twitter で検索し抽出し、プロジェクト内で TwitterTimeLine を共有することができた。実際にプロジェクト内で関する Tweet<sup>\*23</sup>を保存することによって、情報共有ツールのプラグインとして完成することができた。

### 5.2 問題点

本研究で開発した 2 種類のプラグイン毎に問題点を示す。

#### 5.2.1 SmartPhone plugin の機能

Redmine は標準の状態、様々なツール<sup>\*24</sup>を利用することが出来る。本プラグインはチケット管理部分の導入しか行っていないため、Redmine の機能をフルに活かすことが出来ていない。

#### 5.2.2 TimeLine 抽出方法

本プラグインでは、TimeLine の情報を抽出する部分ではユーザー自身が Tweet を選択する必要がある。TimeLine 選別方法が各ユーザそれぞれ異なる可能性があるため、効率的に必要な情報を取り出せるかは曖昧である。

### 5.3 今後の改善点

本研究で開発した 2 種類のプラグイン毎に改善点を示す。

#### 5.3.1 SmartPhone Plugin への機能追加

wiki やフォーラム、リポジトリビューなど標準で備わっている機能を追加していく。jQuery や jQueryMobile のバージョンアップやスマートフォン用ブラウザの更新や開発が頻繁に行われているため、HTML 部分の互換性を保ちながら機能を提供する必要がある。

---

\*21 プロジェクト管理ウェアの重要性や必要性

\*22 屋外でのチケットの閲覧など

\*23 Twitter のつぶやき

\*24 wiki や forum

### 5.3.2 TimeLine 抽出方法を自動化へ

プロジェクト内に関連する重要キーワードやメンバーが興味のある Tweet を自動で保存できる機能を実装したい。ユーザが漏らしている情報を効率的に収集させることが今後の課題になるであろう。

## 6 まとめ

本研究では、Redmine Plugin の開発によって UI の利便性の向上と外部 SNS との連携をし、プロジェクト管理ウェアの機能拡充を行った。実際にスマートフォン向けプラグインを研究室内で公開すると、シンプルな操作感でチケットシステムの仕組みを理解することができ、利用してもらえた。Twitter プラグインを利用することで、Twitter からのタイムライン共有が可能になり、情報共有ツールの一部として利用ができる。スマートフォン向けプラグインの利用で、遠隔地でスマートフォンブラウザを用いてプロジェクトに参加が可能になった。本システムを用いることでスマートフォンに最適化した UI 環境の提供と、SNS 連携を用いたタイムリーな情報の引き出しは、今後のプロジェクト進捗に貢献することが期待できる。

## 謝辞

本研究を進めていく上で、大垣 斉講師から御指導及び御協力をいただきました。また、情報教育システム研究室のメンバーならびに、卒業生の方々に深く感謝します。

## 参考文献

- [1] Twitter. <http://twitter.com/>.
- [2] Facebook. <http://www.facebook.com/>.
- [3] redmine.jp. <http://redmine.jp/>.
- [4] 前田剛. 入門 Redmine 第2版 Linux/Windows 対応. 秀和システム, 2010.
- [5] 小川 明彦 阪井誠. Redmine によるタスクマネジメント実践技法. 翔泳社, 2010.
- [6] Ruby on rails. <http://rubyonrails.org/>.
- [7] Basecamp. <http://basecamp.com/>.
- [8] 37signals. <http://37signals.com/>.
- [9] Zoho projects. <http://www.zoho.jp/projects/>.
- [10] Microsoft office project. <http://www.microsoft.com/japan/project/default.aspx>.
- [11] Openproj. <http://sourceforge.net/projects/openproj/>.
- [12] The trac project. <http://trac.edgewall.org/>.
- [13] Sakura internet. <http://www.sakura.ad.jp/>.
- [14] Rubygems.org. <http://rubygems.org/>.
- [15] Twitter developer. <https://dev.twitter.com/docs/api/1/get/search>.

## 付録 A

### A.1 SmartPhone Plugin

```
-- init.rb

require 'redmine'

Redmine::Plugin.register :redmine_smart_phone do
  name 'Redmine Smart Phone plugin'
  author 'Desucru'
  description 'SmartPhone UI '
  version '1.0.1'
  author_url 'http://redmine.desucru.info/'
end

-- config/routes.rb

ActionController::Routing::Routes.draw do |map|

  map.mobileui '/mobileui', :controller => "mobileui"

  map.namespace :mobileui do |mobileui|
    mobileui.resources :projects , :member => { :activity => :get }
  end
end

-- app/controllers/mobileui_controller.rb

class MobileuiController < ApplicationController
  unloadable
  layout "mobileui"

  def index
    @projects = Project.visible.select {|proj| proj.root?}
  end

  def login
    if request.get?
      self.logged_user = nil
    else
      passwd_auth
    end
  end

  def logout
    self.logged_user = nil
    redirect_to :action => 'index'
  end

  def clear_passwd(user)
    self.logged_user = user
  end
end
```

```

        call_hook(:controller_account_success_authentication_after, {:user => user })
        redirect_to :action => 'index'
    end

    def error_login
        flash.now[:error] = l(:notice_account_invalid_credentials)
    end

    def passwd_auth
        user = User.try_to_login(params[:username], params[:password])
        if user.nil?
            error_login
        else
            clear_passwd(user)
        end
    end
end

-- app/controllers/mobileui/projects_controller.rb

class Mobileui::ProjectsController < ApplicationController
    unloadable
    layout "mobileui"
    before_filter :find_project

    def show
        @issues = @project.issues.all(:conditions => ["project_id = ?", @project.id])
    end

    def new
        @issue = @project.issues.build
    end

    def create
        @issue = @project.issues.build(params[:issue])
        @issue.author_id = User.current.id
        @issue.parent_issue_id = params[:parent_issue_id]
        if @issue.save
            flash[:notice] = l(:notice_successful_update)
            redirect_to :action => 'show'
        else
            render :action => 'new'
        end
    end

    def list_view
        @tickets = Issue.find_by_id(params[:ticket_id])
    end

    def find_project
        @project = Project.find(params[:id])
        rescue ActiveRecord::RecordNotFound
    end
end

```

```

        render_404
    end
end

-- app/views/layouts

<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Redmine for SmartPhone</title>
<%= stylesheet_link_tag 'jquery.mobile-1.0a3.min.css', :plugin => :redmine_smart_phone %>
<%= javascript_include_tag 'jquery-1.5.min.js', :plugin => :redmine_smart_phone %>
<script type="text/javascript">
$(document).bind("mobileinit", function(){
    $.extend( $.mobile , {
        ajaxLinksEnabled: false,
        ajaxFormsEnabled: false
    });
});
</script>
<%= javascript_include_tag 'jquery.mobile-1.0a3.min.js', :plugin => :redmine_smart_phone %>
</head>
<body>
<div data-role="page" data-theme="b">
    <%= yield %>
    <div data-role="footer">
        <h4>&copy;Desucru</h4>
    </div>
</div>
</body>
</html>

-- app/views/mobileui/index.html.erb

<div data-role="header">
    <h1>Redmine</h1>
    <% if !User.current.logged? %>
        <a href="/mobileui/login" data-rel="login" data-icon="plus" data-iconpos="right">Login</a>
    <% else %>
        <%= link_to 'Logout',{:action => 'logout'},:method => :post %>
    <% end %>
</div>
<div data-role="content">
<ul data-role="listview">
<% if User.current.logged? %>
    <li data-role="list-divider">My Projects</li>
<% else %>
    <li data-role="list-divider">Projects</li>
<% end %>
    <% if @projects.any? %>
        <% @projects.each do |project| %>

```

```

                <li>
                    <%= link_to("#{h(project.name)}", mobileui_project_path(project)) %>
                </li>
            <% end %>
        <% end %>
    </ul>
</div>

```

```
-- app/views/mobileui/login.html.erb
```

```

<div data-role="header">
    <h1>Redmine</h1>
</div>
<div data-role="content">
<% form_tag({:controller => "/mobileui", :action=> "login"}) do %>
    <ul data-role="listview">
        <li>Username:<%= text_field_tag 'username' ,nil %></li>
        <li>Password:<%= password_field_tag 'password', nil %></li>
    </ul>
    <p><input type="submit" value="<%= l(:button_login) %>"></p>
<% end %>
</div>

```

```
-- app/views/mobileui/projects/show.html.erb
```

```

<div data-role="header">
    <a href=".." data-rel="back">Back</a>
    <h1>Redmine</h1>
</div>
<div data-role="content">
<ul data-role="listview">
    <li data-role="list-divider">MENU</li>
    <li><%= link_to l(:label_new),{:action => 'new', :id => @project} %></li>
    <li data-role="list-divider"><%= @project.name %> - Ticket List</li>
<% @issues.each do |issue| %>
    <li><%= link_to "#{issue.id}:#{issue.subject}" ,
        {:action => 'list_view', :id => @project , :ticket_id => issue.id} %></li>
<% end %>
</ul>
</div>

```

```
-- app/views/mobileui/projects/new.html.erb
```

```

<div data-role="header">
<a href=".."<%= @project %>" data-rel="back">Back</a>
<h1>Redmine</h1>
</div>
<div data-role="content">
<% form_for @issue , :url => {:action => 'create', :id => @project} do |f| %>
<ul data-role="listview">
    <li data-role="list-divider"><%= l(:label_new) %></li>
    <li>Subject:<%= f.text_field :subject, :placeholder => "題名", :size => 10 %> </li>

```

```

      <li>Tracker:<%= f.select :tracker_id, @project.trackers.collect
      {|tracker| [tracker.name, tracker.id]}, :required => true %></li>
      <li>Description:<%= f.text_field :description, :placeholder => "概要", :size => 15 %></li>
    </ul>
    <p><%= submit_tag l(:button_save) %></p>
  <% end %>
</div>

-- app/views/mobileui/projects/list_view.html.erb

<div data-role="header">
  <a href=".." data-rel="back">Back</a>
  <h1>Redmine</h1>
</div>
<div data-role="content">
  <ul data-role="listview">
    <li data-role="list-divider"><%= @tickets.id %>&nbsp;&nbsp;&nbsp;<%= @tickets.subject %></li>
    <li>題名:<%= @tickets.subject %></li>
    <li>種類:<%= @tickets.tracker.name %></li>
    <li><%= l(:field_status) %>:<%= @tickets.status.name %></li>
    <li><%= l(:field_priority) %>:<%= @tickets.priority.name %></li>
    <li>作成者:<%= @tickets.author %></li>
    <li>
      <%= l(:field_assigned_to) %>:
      <%= avatar(@tickets.assigned_to, :size => "14") %>
      <%= @tickets.assigned_to ? link_to_user(@tickets.assigned_to) : "-" %>
    </li>
    <li><%= l(:field_description) %>:<%= @tickets.description %></li>
    <li>作成日:<%= format_date(@tickets.created_on) %></li>
    <li>更新日:<%= format_date(@tickets.updated_on) %></li>
  </ul>
</div>

```

## A.2 Twitter Plugin

```

-- init.rb

require 'redmine'

Redmine::Plugin.register :redmine_twit4red do
  name 'Redmine Twit4red plugin'
  author 'Desucru'
  description 'This is a plugin for Redmine'
  version '1.0.1'
  url 'http://redmine.desucru.info/'
  author_url 'http://redmine.desucru.info/'
  project_module :twit4red do
    permission :twit4red, {:tweets => [:index]}, :public => true
  end
end

```

```

    menu :project_menu, :twit4red, {:controller => 'tweets', :action => 'index'}
end

-- app/controllers/tweets_controller.rb

require 'rubygems'
require 'twitter'

class TweetsController < ApplicationController
  unloadable
  before_filter :find_project, :find_twitter

  def index
    @twitters = Tweetline.find(:all, :conditions => ["project_id = #{@project.id}"])
    if request.post?
      words = params[:words]
      @search = Twitter::Search.new.containing(words).per_page(100)
    else
      @search = Twitter::Search.new.containing(@project.identifier).per_page(100)
    end
  end

  def new
    @twitter = Tweetline.new(params[:twitter])
    @twitter.project_id = @project.id
    if request.post? and @twitter.save
      flash[:notice] = 'Tweet Saved'
      redirect_to :action => 'index', :id => @project, :twitter_id => @twitter_id
    end
  end

  def destroy
    Tweetline.destroy(params[:delete_id])
    flash[:notice] = 'Tweet Deleted'
    redirect_to :action => 'index', :id => @project, :twitter_id => @twitter_id
  end

  def find_project
    @project = Project.find(params[:id])
  end

  def find_twitter
    @twitter = Tweetline.find_by_id(params[:twitter_id])
  end
end

-- app/views/tweets/index.html.erb

<div id="tweetmain" style="width:100%;">
<p>Defalut Search word: <%= @project.identifier %></p>
<p>
<% form_tag :action => "index", :id => @project ,:project_id => @project_id do %>

```

```

<%= text_field_tag 'words' %>
<%= submit_tag "Search" %>
<% end %>
</p>
<div id="left" style="float:left;width:50%;">
<table>
<thead><tr><th colspan="3">tweetlist</th></tr></thead>
<% @search.each do |line| %>
<tr>
<td style="border:solid 1px #444;"></td>
<% form_tag :action => "new", :id => @project, :project_id => @project_id do %>
<td style="border:solid 1px #444;"><%= line.from_user %>:<%= label_tag line.text %></td>
<td style="border:solid 1px #444;">
<input type="hidden" name="twitter[timeline]" value="<%= line.from_user %>:<%= line.text %>">
<%= submit_tag "Save" %>
</td>
<% end %>
</tr>
<% end %>
</table>
</div>

<div id="right" style="float:right;width:50%;">
<% if (@twitters.blank?) %>
  <p class="nodata"><%= l(:label_no_data) %></p>
<% else %>
  <table>
    <thead><tr><th colspan="3">tweet database</th></tr></thead>
    <tbody>
      <% @twitters.each do |twitter| %>
        <tr>
          <td style="border:solid 1px #444;"><%= twitter.timeline %></td>
          <td style="border:solid 1px #444;">
            <%= link_to 'delete',{:action => 'destroy',
              :id => @project, :delete_id => twitter.id}, :confirm=> :text_are_you_sure) , :method => :post %></td>
          <td style="border:solid 1px #444;">
            <a href="https://twitter.com/share" data-text="<%= twitter.timeline %>"
              class="twitter-share-button" data-count="none" data-lang="ja">Tweet</a>
            <script type="text/javascript" src="//platform.twitter.com/widgets.js"></script>
          </td>
        </tr>
      <% end %>
    </tbody>
  </table>
<% end %>
</div>
</div>

```

```
-- db/migrate/create_timelines.rb
```

```
class CreateTweetlines < ActiveRecord::Migration
  def self.up
```

```
create_table :tweetlines do |t|
  t.column :project_id, :integer
  t.column :timeline, :text
end
end

def self.down
  drop_table :tweetlines
end
end
```