**研究中間報告**

NetCommons3プラグイン開発における機能提案および評価

　　　　指導教授 国立情報学研究所　社会共有知研究センター

新井　紀子　　教授

　　　　担当教授 (日工専) 情報工学科

清水　富門　　教授

　　　　報告者 (日工専) 情報工学科

第 54 期　研究科生　外田　浩太朗

研究期間 2014 年 4 月 1 日　～　2014 年 12 月 11日

報告年月日 2014 年 12 月 12 日

# **要旨**

　近年、Webサイトの脆弱性を突く攻撃が増えてきており、2014年6月19日にはIPAから「管理できていないウェブサイトは閉鎖の検討を」という注意勧告がなされた。この主な攻撃対象となっているのが、コンテンツマネジメントシステム（以下、CMS）である。その理由の一つとしてCMSの殆どがオープンソースソフトウェアであり、ソースコードの参照が可能であることが挙げられる。

　国立情報学研究所の本研究室で開発されたCMSがNetCommons（現行バージョン2.4.2.0）である。NetCommonsはe-Learningやグループウェアとしての側面も持っており、小中高や大学等の教育機関が主なユーザされている。

　NetCommonsプロジェクトでは一昨年度からバージョン3（以下、NC3）の開発に着手している。NC3の開発では、メンテナンス性の向上やコンプライアンスの徹底等、ユーザが脆弱性やコンプライアンスを気に掛けることなく利用できるというビジョンがある。

　本研究では、NC3の機能開発においてユーザが利用するフォームに対し、リアルタイムのエラーチェック機能を実装し、最適化していくことでユーザビリティの向上を目指す。

　現在は実装したコードの最適化を行っている段階であり、3月末までの課題として「フォーム機能の評価」「iframeプラグインレビュー結果反映」「掲示板プラグインの開発」がある。

# **目次**

[**第 1 章** **緒言** 1](#_Toc403483534)

[1.1. 背景 1](#_Toc403483535)

[1.2. 目的 1](#_Toc403483536)

[**第 2 章** **NetCommonsプロジェクト** 2](#_Toc403483537)

[2.1. NetCommons 2](#_Toc403483540)

[2.1.1. NetCommonsの特徴 2](#_Toc403483541)

[2.1.2. NC2の実績 3](#_Toc403483542)

[2.2. NC3のビジョン 3](#_Toc403483543)

[2.2.1. NC2での問題点 3](#_Toc403483545)

[2.2.2. NC3の基本理念 4](#_Toc403483549)

[2.2.3. 開発スケジュール 7](#_Toc403483553)

[2.2.4. ユーザNC3を利用することによる効果 7](#_Toc403483554)

[2.3. NC3開発方式 8](#_Toc403483558)

[2.3.1. アプリケーション／ライブラリ構成図 8](#_Toc403483560)

[2.3.2. 各アプリケーション／ライブラリの機能概要 10](#_Toc403483561)

[2.3.3. 開発環境および、開発フロー 13](#_Toc403483562)

[**第 3 章** **NC3プラグインにおける入力支援機能の提案** 14](#_Toc403483563)

[3.1. 開発機能 14](#_Toc403483567)

[3.1.1. iframeプラグイン 14](#_Toc403483568)

[3.1.2. 開発スケジュール 14](#_Toc403483569)

[3.1.3. 画面遷移図 14](#_Toc403483570)

[3.1.4. 規模 14](#_Toc403483571)

[3.2. 提案 14](#_Toc403483572)

[3.2.1. 提案内容 14](#_Toc403483573)

[3.2.2. 実現方法 14](#_Toc403483574)

[**第 4 章** **性能評価** 15](#_Toc403483575)

[**第 5 章** **結言** 17](#_Toc403483576)

[5.1. 結論 17](#_Toc403483577)

[5.2. 今後の課題 17](#_Toc403483578)

[**第 6 章** **謝辞** 18](#_Toc403483579)

[**第 7 章** **参考文献** 19](#_Toc403483580)

# **緒言**

## 背景

　本研究室ではNC3プロジェクトに参画し、iframeというプラグイン（CakePHPのアプリケーションの単位）開発を担当することになった。機能によるが、殆どのプラグインがフォームを利用する。例えばiframeであれば、URLを設定するためのフォームがある。

　上記のようなフォームを利用する機能の開発は、筆者が研究室内で先行して開発していたため、このフォームに関する提案および評価を本稿のテーマとして纏める。

## 目的

　本科生時代の卒業研究ではNC2を利用して、PLATONという（日工専）の情報共有基盤の開発に取り組んだ。その中で、NC2のインターフェースに関する質問を受けることがあった。しかし、卒業研究では見た目以外でNC2のコードに手を加えることをせずに、運用方法を定めることでカバーするとしていたため、NC2の仕様であるとしか回答することができなかった。

　今回の開発ではユーザが入力／操作するインターフェースとなるフォームを提案できる機会を得た。ユーザ目線で入力がしやすい、エラーの内容が分かりやすい等のフォームを意識して提案／実装し、その評価を行う。

# **NetCommonsプロジェクト**



## NetCommons

　NetCommonsの特徴と実績を2.1.1以下に示す。

### NetCommonsの特徴

　NetCommonsはCMSとLMSとグループウェアを統合したコミュニティウェアである。



図 2.1　コミュニティウェアとは

またNetCommonsは外部配信向けのポータルサイトの機能（パブリックスペース）、個人のバーチャルオフィスとしての機能（プライベートスペース）、グループの情報共有のための機能（グループスペース）が一つのシステムの中で統合されている。



図 2.2　各スペースによる機能

　システムの管理者は短時間で主要ブラウザで閲覧可能な美しくデザインされたサイトを構築することが可能である。また利用するユーザは短時間で操作方法を取得できる。

### NC2の実績

　TODO

　後援団体など。採用している団体、教育委員会、企業など

## NC3のビジョン

　NC2での問題点、NC3の基本理念および開発スケジュールを2.2.1以下に示す。



### NC2での問題点

### プログラム改修

　日々の問合せの中で、改修で対応するものがある。過去に承認フローの見直しや会員検索のバグ修正等の改修があり、膨大な量のテスト項目、テストデータを作成し、Webブラウザから一つ一つ操作して潰していかなければならなかった。

またコーディング規約は現状バラバラであるため、開発者によって書き方が違い、一括で共通のコードを置換しようとした場合に、置換漏れが出てしまうといった問題がある。

### 動作環境

　現在はバージョンを固定するmetaタグが入れられているが、IE等のWebブラウザやPHPのバージョンアップが発生した場合は対応が必要となる。その際は機能を限定して動作確認を行っていた。この際もプログラム改修と同様に、Webブラウザから操作してテスト項目を潰す作業が必要となる。このようなアプリケーションやライブラリのバージョンアップは突発的に発生するため、前もって作業員を当てることは難しく、無理にでも捻出して対応しなければならないといった問題がある。

### 開発環境

　開発用サーバー（Webサーバー、DBサーバー）、バージョン管理サーバー（SVN）を研究室内にい構築し、開発していた。そのため、オープンソースソフトウェアなのに研究室内でなければメンテナンスができないといった問題があった。

### NC3の基本理念

　前項のように様々な問題点がNC2には存在している。そこで発足したのがNC3プロジェクトである。

NC3にある3つの基本理念を下記に示す。

### ドライな経済性

1. 費用対効果が見積もれるもの
2. 環境が変わった時の安定性・頑強性
3. パソコンだけでなく、スマートフォンやタブレットから見ても崩れない（レスポンシブデザイン）
4. メンテナンスの向上

* セキュリティ上の脆弱性が発見されたときに、短期間でバージョンアップが可能

1. RESTなサービス



図 2.3　レスポンシブデザイン1（Webブラウザの場合〈Firefox〉）



図 2.4　レスポンシブデザイン2（スマートフォンの場合〈シミュレータ〉）

### 遵法であること

1. 個人情報保護法への対応

* 会員管理やアンケート、問い合わせフォーム等の個人情報を入力させる機能での個人情報の扱い。
* 暗号化の仕組みとルール化。

### 権限の委託

1. 役割に応じて権限を委譲し、仕事を分散する

* たとえば、校長先生は学校の責任者であり、管理者である。しかし、校長先生がITから遠い人の場合、学校の業務が滞ってしまう。その管理者の業務を役割に応じて権限委譲することにより、業務を円滑に進められる。
* コンテンツには承認の機能を設け、コンテンツを書く人、公開する人に役割を分散することによって、仕事を一人に集中させない。

表 2.1　権限による表示制御

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 項番 | パターン | A | | B | C |
| 1 | 編集権限 | なし | | あり | あり |
| 2 | 公開権限 | なし | | なし | あり |
| 3 | 表示 | 公開コンテンツ　あり |  |  |  |
| なし | フレームが表示されない |

* パターンA

　編集／公開権限を持たないユーザの場合で、編集するための管理ボタン等は何も表示されない。

この場合は公開されているコンテンツが表示される。公開されているコンテンツが無い場合は、フレームの枠そのものが表示されない。

* パターンB

　編集権限のみ持つユーザの場合で、管理ボタンおよび、コンテンツの状態を表すラベルが表示される。

この場合は管理ボタンからコンテンツの編集等が可能である。このユーザが「決定」を押下し、保存されたコンテンツは「承認待ち」となる。

* パターンC

　編集／公開権限を持つユーザの場合で、管理ボタンおよび、コンテンツの状態を表すラベル、承認待ちの場合、承認ボタンが表示される。

　パターンBとは違い承認権限を持ち、承認ボタンが押されたコンテンツは公開状態となる。また、パターンBの「決定」とは意味が変わり、このユーザが「決定」したコンテンツは公開される。

### 開発スケジュール

　筆者がプロジェクト参画する期間内のNC3開発スケジュールを示す。

**表　NC3開発スケジュール**

●は実績、○は予定

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 項番 | 年月  作業項目 | | 2014 | | | | | | | | | 2015 | | |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 1 | 2 | 3 |
| 1 | 管理プラグイン | |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  | 仕様検討 | ● | ● | ● | - | - | - | - | - | - | - | - | - |
| 3 | 設計・実装・テスト | ● | ● | ● | ● | ● | ● | ● | ● | ● | - | - | - |
| 4 | レビュー※ | - | - | - | - | - | - | - | - | ● | - | - | - |
| 5 | 一般プラグイン（研究室内） | |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  | 仕様検討 | ● | ● | ● | ● | ● | ● | ● | ● | - | - | - | - |
| 7 | 設計・実装・テスト | - | - | - | - | - | ● | ● | ● | ● | ○ | ○ | ○ |
| 8 | レビュー※ | - | - | - | - | - | - | - | ● | - | - | ○ | - |
| 9 | 一般プラグイン（外部委託） | |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  | 設計・実装・テスト | - | - | - | - | - | ● | ● | ● | ● | ○ | ○ | ○ |
| 11 | レビュー※ | - | - | - | - | - | - | - | ● | - | - | - | ○ |
| 12 | α版リリース | | - | - | - | - | - | - | - | - | - | - | - | ○ |

※仕様の相談等の細かなレビューは不定期に実施しており、ここの○は最終レビューの大まかな時期を示す。

### ユーザNC3を利用することによる効果

　NC3とNC2を比較した場合、NC3を利用することによって得られる効果を下記に示す。

### 独自デザインの反映が容易になる

　NC2でもデザインのカスタマイズは可能であったが、NC2の構造を理解しなければ難しい。NC3では画面上で一括してサイト全体のデザインを変更できる。またオリジナルで作成したテーマをアップロードして全体に反映させることも可能となる。

### 追加機能（プラグイン）の開発の敷居が低くなる

　NC2ではMapleというPHPフレームワークを使用している。このMapleは開発者が日本人であることもあり、日本語の開発ドキュメントが豊富で多くの開発で利用されていたが、TODO○○に開発が終了していた。NC3ではCakePHPというフレームワークを使用し、今でも盛んに開発が進められているためサポートは十分に受けられる。CakePHPはアプリケーションをプラグインという単位でパッケージ化できる。そのため、CakePHPやNC3の規約に沿って開発を行えば、独自でプラグインを開発しNC3本体にその機能を取り込むことが可能となる。

### 閲覧する媒体に依存しない

　基本方針のドライな経済性でも述べた通り、NC3はレスポンシブデザインを取り入れているため、どのようなWebブラウザのサイズにも対応し、表示することができる。

そのため閲覧する媒体がPCであろうとタブレットであろうとスマホであろうと決して画面構成が崩れることなく、閲覧、操作することができる。

## NC3開発方式

利用するアプリケーション構成図と機能概要、開発フローを2.3.1以下に示す。



### ソフトウェア構成図

NC3を動作させるために利用するソフトウェアの構成図を以下に示す。

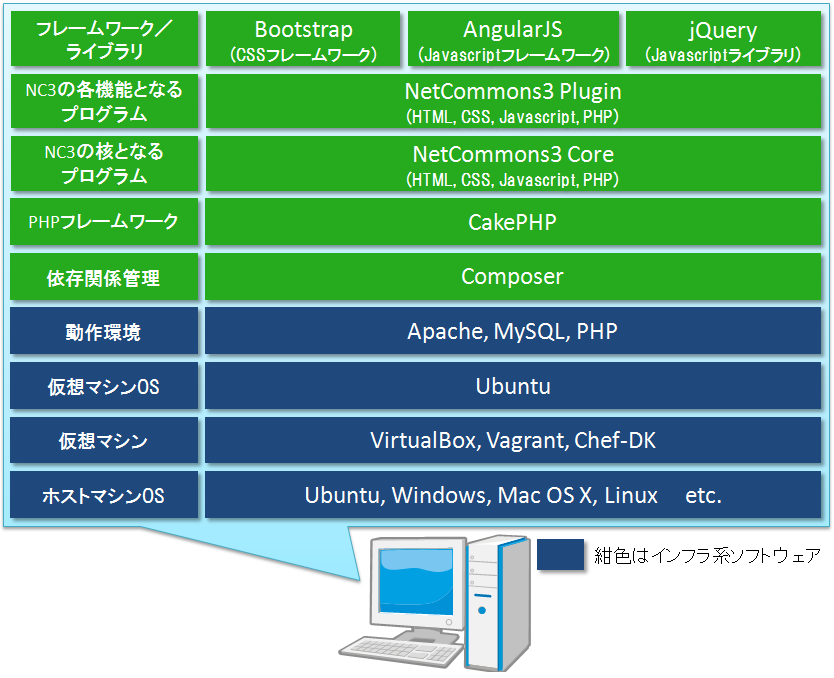


図 2.5　NC3動作に利用するソフトウェア構成図

　上記に加え、NC3を開発するために利用するソフトウェアの構成図を以下に示す。

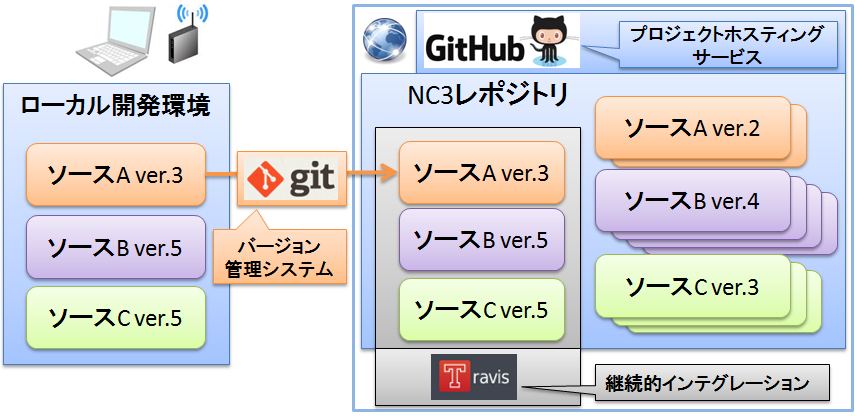


図 2.6　NC3開発に利用するソフトウェア

### ソフトウェアの機能概要

　NC3の動作および開発で利用するソフトウェアの機能概要を以下の表に示す。

表 2.2　利用するソフトウェアの機能概要

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | 大分類 | 小分類 | アプリケーション名 | 概要 |
| 1 | インフラ系 | 仮想マシン | Oracle VM VirtualBox | オープンソースの仮想化ソフトウェアで、コンピュータ上にVMを構築し、その中でOSを起動して操作することができる。 |
| 2 | Vagrant | 仮想環境の雛形を作成し、どこでも簡単に同じ環境を再現することができる。仮想マシンの設定やOSの設定をテキストファイルで与えることで特定の状態でマシンを立ち上げることができる。 |
| 3 | Chef Development Kit | Chefを利用するためにパッケージングされたツール群。  Chefとはオープンソースのシステム統合フレームワークで必要なアプリケーションを自動的に構築および、調整することができる。設定情報の定義はRubyで記述する。 |
| 4 | 動作環境 | Apache HTTP Server | フリーソフトウェアとして公開されるWebサーバ。 |
| 5 | MySQL | オープンソースで利用できるリレーショナルデータベース管理システム。 |
| 6 | PHP | オープンソースでWebページを記述することに特化した、サーバサイドスクリプト言語。HTMLに埋め込むことができる。 |
| 7 | NC3動作 | 依存関係管理 | Composer | ライブラリの依存関係を管理するツール。composer.jsonというファイルに使うライブラリ名、バージョン等の一覧を記述し、コマンドを実行することで必要なライブラリを一括でインストールすることができる。 |
| 8 | NC3 | Core（本体） | NC3を構成するソフトウェアの設定やNC3のプラグインを動作させるために必要となるプログラム群。 |
| 9 | Plugin（プラグイン） | CakePHPのアプリケーションの単位。開発はこのPlugin単位に行う。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | 大分類 | 小分類 | アプリケーション名 | 概要 |
| 10 | NC3動作 | フレームワーク | CakePHP | オープンソースで公開されているRAD（Rapid Application Development）型のPHPフレームワーク。MVCモデルが採用され、プログラムのメンテナンス等が容易になる。 |
| 11 | AngularJS | Googleが中心となってオープンソースで開発されているJavascriptフレームワーク。MVCモデルが採用されている。双方向データバインディング等の特徴を持つ。 |
| 12 | Bootstrap | Twitterが公開するCSSフレームワーク。TwitterライクなデザインでWebページを作成でき、レスポンシブデザインを実現することができる。 |
| 13 | ライブラリ | jQuery | オープンソースのJavascript用ライブラリ。Ajaxによる非同期通信、DOM操作、ユーティリティの利用、プラグインによる機能拡張、CSS操作、ブラウザに依存しない等の特徴を持つ。これにより、Javascriptのコード量を簡素化することができる。 |
| 15 | バージョン管理  システム | | Git | 分散型のバージョン管理システム（VCS）で開発者それぞれがローカル環境にリポジトリを持つことができる。そのためネットワークにアクセスできない環境でも、自らが作業した履歴の調査や変更の記録等、ほとんどの作業をローカル環境で行うことができる。 |
| 16 | プロジェクト  ホスティング  サービス | | GitHub | Gitのリポジトリをホスティングするサービスで、Gitはコマンドラインツールであるのに対し、GitHubはWebでグラフィカルなユーザインターフェースを提供する。アカウントを登録し、制限内は無料で利用することができる。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | 大分類 | 小分類 | アプリケーション名 | 概要 |
| 17 | テスト | CI | TravisCI | オープンソースコミュニティのためのCI（継続的インテグレーション）※サービス。GitHubと連携しており、CIしたいリポジトリを接続することでTravisCIがコミットを取得して設定通りにビルド／テストを実行する。失敗するとメール等で結果が送信される。 |
| 18 | TravisCIによるテスト | PHPUnit | xUnit系テスティングフレームワーク。 |
| 19 | phpcs | コーディング規約のチェッカ。 |
| 20 | phpmd | 一般的なコーディング作法チェッカ。 |
| 21 | phpcpd | コピーペーストディテクタ。 |

※CI（継続的インテグレーション）

　主にプログラマーのアプリケーション作成時の品質改善や納期の短縮のための習慣のこと。XP（エクストリームプログラミング）のプラクティスの一つで、ビルドやテスト等を継続的に実行していくことを意味する。

Jenkins

### 開発環境および、開発フロー

### 開発環境

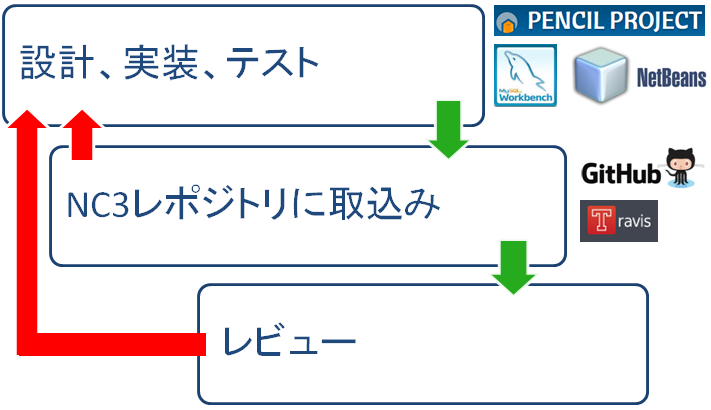
　ローカル開発環境上で使用するソフトウェアを以下に示す。

表 2.3　ローカル開発環境上で使用するソフトウェア

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 項番 | 分類 | ソフトウェア | バージョン | 概要 |
| 1 | 設計 | Pencil Project | 2.0.5 | 画面遷移図作成に使用する。 |
| 2 | MySQL Workbench | 6.1 | ER図の閲覧／編集に使用する。 |
| 3 | 実装 | NetBeans IDE | 8.0 | プラグインをインストールし、Vagrant, Git, データベースへの接続／操作等で使用する。またコーディングの際にPHPやJavascriptの入力補完により、効率的に開発することができる。 |
| 4 | テスト | ― | | TravisCIのビルド実行時に流れるシェルと同等のシェルをローカル開発環境で実行し、NC3レポジトリに取り込む際のエラーを局所化する。 |

### 開発フロー

　開発フローを以下に示す。



図によって示す。

　GitHub上にあるNC3リポジトリをクローン（複製）し、

　　　プロトタイプ、仕様検討、先生の同意、実装（テスト含み）、レビュー

　　　プラグイン毎に開発依頼が出されるため、機能規模によるがおおよそ数か月単位のアジャイル開発となっている。

# **NC3プラグインにおける入力支援機能の提案**



## 開発機能

### iframeプラグイン

### 開発スケジュール

### 画面遷移図

### 規模

　　　　　現在の最新版iframeの各フォルダのステップ数などを記載

　　　　　　PHP

　　　　　　　Model　○行

　　　　　　　View　○行

　　　　　　　Controller　○行

　　　　　　　その他設定ファイル　○行

　　　　　　Javascript　○行

　　　　　　合計　○行

## 提案

### 提案内容

　　・サブミットブロック

　　・初期文言表示

　　・正常／エラーの色識別

　　・リアルタイムエラー表示

　　フォームのバリデーション即時表示を提案

　　　AngularJSのモデルバインド、Bootstrapによる正常、エラー表示の操作

### 実現方法

# **性能評価**

　これから

機能要件的評価

　　処理時間(ブラウザ表示、登録処理など)比較　NC2,NC3の比較

　非機能要件的評価

　　入力フォームの使用性比較（ベンチマーク、検証対象は考えないと）

　　①一つ目の入力項目は入力しやすい項目にするべき。

　　　ダメな例：ユーザにとって、入力内容の自由度が高いもの

　　　　　　　　使えない文字があるなど、入力にあったっての注意が必要なもの。

　　　　　　　　エラーが出やすいもの。

　　②何のためのフォーム化伝える

　　　4点セット：フォーム名（左上）

　　　　　　　　ユーザへの指示。たとえば「入力項目に入力して確認ボタンを押して確認ページに進んでください」など。

　　　　　　　　送信完了後「保存が完了しました。」など

　　　　　　　　ゴールまでの道のりと現在地。入力へのモチベーション（NetCommons3であれば、会員登録フォームなどで意識すべき。）

　　③ファーストビューに収める

　　　入力フォームが縦に長いと、どれだけ入力項目があるのか分からず、ユーザは不安に感じてしまう。

　　　ファーストビューに収まりきらない場合でもせめて、2画面程度に分けるようにする。

　　④

　　　エラーメッセージがリアルタイムに表示され、周囲の要素とかぶらない。

　　　余白を多めに取ることで、該当の項目に集中しやすい。

　　　入力するときの視線の動きを少なくするように配置するよう工夫して、入力のための必要な情報を過不足なく記載する。

　　　（同じないようの記載を避ける。入力項目と注意書きの距離を近くする。など）

　　　入力例、入力形式を明記する。

　　　入力のルールはなるべく少ないのが理想。

　　※リアルタイムアラート機能：https://f-tra.jp/feature/feature\_detail02.html

　　　ユーザが最もストレスを感じる瞬間のひとつがエラー表示。

　　　とくにミスなく入力したと思っていたのに、予想外のエラーが出ることは大きなストレスが生じる。

　　　　「どこどこは全角[半角]で入力してくだいさい。」など

　　　登録などのボタンを押下後にズラリの並んだエラーを表示するには、あらためて全ての項目に１から取り組む必要がある。

　　　　スマホの場合、誤って違うボタンに触れてしまった場合、登録中の情報が全て破棄されることも。

　　　　入力回数、タッチ回数を減らすこと！！！

　　　一方で、ボタン押下後ではなく、入力の都度、間違い箇所を指摘してくれるとこのストレスは格段に小さくなるはずです。

　　　その項目に意識が集中している間に入力形式の違いなどのアドバイスをもらえることは、ユーザにとって小さな方向転換に過ぎず、

　　　特にストレスを感じることなく、修正に応じてもらえる。

　　※ユーザにストレスを感じさせないことの重要性を語る。スマホでの会員登録であったりを例として。

# **結言**

## 結論

　あ

## 今後の課題

　今後、本研究を進めていく上で、次の課題があげられる。

1. フォームバリデーション箇所のコードの最適化
2. 掲示板プラグイン開発時のフォームバリデーション機能実装

　　今後、本研究を進めていく上で、次の課題があげられる。

　　①フォームバリデーション箇所のコード最適化

　　　ぱっとコードを見たときに人目で概要をつかめるようなコード体系か？

　　　AngularJSの使い方を含め、コードを整理する。

　　②掲示板プラグイン開発

　　　12月着手～3月末納品

　　　機能数がiframeと違いかなり多くなる

　　　iframeの開発経験を生かし、手戻りが極力少なくなるように着手する。

# **謝辞**

最後に、本研究を進めるにあたり、データストリームに関する基礎知識から、研究者としての心構えなど、様々なご指導を北川博之教授，川島英之講師より頂いた。また、Semi-Active Standbyの先行研究者である塩川浩昭氏を始めとして、北川データ工学研究室の皆様には、研究に関することからその他に至るまで、大変お世話になった。ここに、心から感謝の意を表する。

# **参考文献**

[1] cakeソフトウェア財団．”CakePHP CookBook 2.Xドキュメント”．

CakePHP CookBook 2.X（オンライン），入手先〈<http://book.cakephp.org/2.0/ja/index.html>〉．

参照2014/4/12 – 2014/12/10．

[2] @tomof．“AngularJS 1.2 日本語リファレンス|js STUDIO”．js STUDIO（オンライン），

入手先〈<http://js.studio-kingdom.com/angularjs>〉．参照2014/4/12 – 2014/12/10．

[3] Mark Otto, Jacob Thornton．”Bootstrap”． Bootstrap（オンライン），

入手先〈<http://getbootstrap.com/>〉．参照2014/4/12 – 2014/12/10．