

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

サーベイに基づいたコミュニケーションを
提供する研究活動支援システムの開発

宮崎裕久

(コンピュータサイエンス専攻)

指導教員 三末和男

2009年 3月

概要

本プロジェクトの委託元教員が指導するグループでは、学生に調査した論文のリスト（以下サーベイリスト）を公開させている。これはサーベイリストを見ることで学生の興味の対象を知り、卒業論文や修士論文のテーマ決めに役立てるためである。また、リストされた論文に近い分野の論文を薦めるなど、研究調査の指導にも役立てている。サーベイリストの公開には、現状では Wiki を活用しているが、よりサーベイリストの作成に特化した使いやすいシステムが求められている。

委託元教員が希望しているシステムは、単なる Wiki の置き換えではなく、様々な発展の可能性を持つ機能を持ったシステムであった。そこで、筆者らは反復型システム開発プロセスを採用することによって、基本的な機能の開発を進めながら発展的な機能を明確にしていくことにした。1 回のイテレーションを 2 週間と短く設定し、毎回のイテレーションの最後に必ず動作するプログラムを仕上げることによって、リスクを分散し、発展的な機能が固まっていくことを狙った。

本プロジェクトでは、サーベイリストの管理およびサーベイ指導を支援する Web ベースの CMS を開発した。サーベイリスト管理機能を持ち、従来の Wiki を用いたシステムに置き換えることが出来る。それだけではなく、論文情報管理機能により、論文の詳細情報や論文のファイルを共有することも可能である。さらに、論文推薦機能により、教員のサーベイリストの論文を直接学生のサーベイリストに送ることが出来るようになった。これらの機能により、このシステムはサーベイに基づいたコミュニケーションを提供することが出来る。

目次

第 1 章	緒言	3
1.1	動機と目的	3
1.2	研究開発経過	3
第 2 章	現行システム・業務の課題	5
第 3 章	開発システムの概要	6
3.1	システム構成	6
3.2	開発機能の説明	6
第 4 章	システムの開発	10
4.1	要件定義	10
4.2	外部設計	11
4.3	内部設計	12
4.4	実装	14
第 5 章	失敗およびあい路事項	16
第 6 章	結言	17
謝辞	18
参考文献	19

図目次

図 1 件数が多くなったサーベイリスト	5
図 2 システム構成図	6
図 3 サーベイリストのページ	7
図 4 論文の詳細情報	8
図 5 ビュー仕様書	12
図 6 クラス図	13

第1章 緒言

研究開発プロジェクトのカテゴリ 3 は、顧客の要求するシステムを開発する実践的カリキュラムである。教員が顧客となってシステム開発のリクエストを提示し、学生はその中の一つを受託し、開発体制を作ってプロジェクトを遂行する。筆者は三末和男准教授が提示したリクエストを受託した。筆者 1 名による開発体制で 2008 年 9 月から 12 月の間、開発に取り組んだ。

1.1 動機と目的

本プロジェクトの委託元教員が指導するグループでは、学生に調査した研究論文のリスト（以下サーベイリスト）を公開させている。教員はサーベイリストに含まれる論文の傾向を見ることで、学生が興味を持っている研究の分野を知ることが出来る。学生の興味を知るとは卒業論文や修士論文のテーマを指導する助けとなる。また、サーベイリストから学生の研究調査の進捗状況を知り、リストされた論文に関連する論文を薦めるなど、研究調査の指導にも役立っている。

サーベイリストを公開させる方法として、現状では Wiki を活用している。グループの学生は各々個人の Wiki を持っており、そこにページを作成してサーベイリストを公開している。サーベイリストに記載する情報には論文タイトルや著者、出典の他に、「読んだ論文」・「pdf を入手した論文」・「興味のある論文」という 3 つの分類がある。この分類を見て教員は学生の研究調査の進捗状況を知ったり、学生が持っていない論文の pdf ファイルを渡したりすることが出来る。学生への論文の推薦はメールで行っているが、教員はもっと便利に学生へ論文の推薦が出来るシステムを求めている。

本プロジェクトの目的は現在の Wiki に替わるサーベイリストを管理するシステムを開発することで研究調査の指導を支援し、サーベイに基づいたコミュニケーションを活発にすることで学生の研究へのモチベーションを高めることである。

1.2 研究開発経過

プロジェクトの開始に先駆けて、委託元教員にヒアリングを行った。ヒアリングではサーベイリスト公開の現状と、求めるシステムのイメージについて話を聞くことが出来た。ただ Wiki に置き換わるシステムではなく、様々な発展の可能性を考えていることがわかった。そこで、本プロジェクトでは反復型開発プロセスを採用し、要件定義を繰り返しながら基礎的な機能から発展的な機能へとシステムを拡張していく方法を取ることにした。

本プロジェクトで採用した反復型開発プロセスでは、1 回のイテレーション（反復）の長さを 2 週間という比較的短い期間に設定した。この期間の中でソフトウェア開発プロセスの工程を 1 通り行い、毎回実際に動作するプログラムを仕上げることを目標とした。このよう

に短い期間のイテレーションにした理由は、アジャイルソフトウェア開発[1]の考え方に習い、開発におけるリスクを分散・最小化するためである。アジャイルソフトウェア開発では、開発を細かな機能に区切り、顧客との打ち合わせを頻繁に行うことで、開発したシステムが顧客の要求と大きくずれてしまうことを防いでいる。その代わりに小さな変更を多く受け入れることで、顧客の満足度を高めることを目指す。しかも、紙に書いた成果物に基づいたやりとりではなく、実際に動作するプログラムを顧客に見せるため、勘違いによるリスクを回避しやすい。本プロジェクトでは基本的な機能から順に開発し、動作するプログラムを委託元教員とレビューすることで、その上に乗る発展的な機能のイメージが徐々に固まっていくことを狙った。本プロジェクトはチーム体制ではなく個人での開発であるため、メンバー間の意識合わせの時間が必要なく、このように小回りの利いた開発プロセスが実現できた。

通常ソフトウェア開発では、各工程の成果物としてドキュメントを仕上げることで、次の工程に移ることができる。しかしながら、本プロジェクトは個人による体制であり、1回のイテレーションで扱う規模も大きくないため、分析や設計をドキュメントにまとめて「誰かに伝える」必要性が低い。しかも、繰り返し委託元教員のレビューを受ける中で、プログラムの変更を多く受け入れることを考えていたため、逐ードキュメントを作成していたのではドキュメント修正に大きなコストが掛かってしまう。そこで本プロジェクトでは、ドキュメント作成を開発の最後（イテレーション内の最後ではなく、複数回のイテレーションを回した後である）に行い、本システムの開発を別の技術者が引き継げるような資料を作成することを目標にした。実際に委託元教員は本システムの開発を次年度以降も続けることを考えており、そのためにはプロジェクトの引き継ぎを行う必要がある。作成したドキュメントは、プロジェクト引き継ぎのための重要な資料となる。実装を基にドキュメントを作成するため、設計書ではなく「仕様書」と呼ぶことにした。

第2章 現行システム・業務の課題

現在サーベイリストを公開するために利用されている Wiki は、その用途において完璧なシステムであるとは言い難い。Wiki を用いたサーベイリストは、あくまで手作りのページであるため、Wiki の操作に不慣れな学生が十分に使いこなせない恐れがある。サーベイリストの作成を困難に感じたら、サーベイへのモチベーションは下がってしまうであろう。また、十分に使いこなしている学生であっても、リストの件数が増えるにつれて管理が困難になり、過去のリストを十分に活用できない状態になり得る（図 1）。このような状況を解消するためには、Wiki よりもサーベイリストの作成に特化した使いやすいシステムが必要である。

論文の推薦はメールで行われているが、メールを受け取った学生は推薦された論文の情報を Wiki ページに打ち直さなければならない。Wiki であるため、学生のサーベイリストに直接推薦する論文の情報を入力することも可能であるが、スタイルがまちまちであるため入力が難しかったり、入力されても学生が気づかなかったりとうまく機能していない。ただリストを公開するだけでなく、論文を推薦し合ったり論文の内容について議論したりといった、サーベイに基づいたコミュニケーションを活発に行うことが出来るシステムが求められている。



図 1 件数が多くなったサーベイリスト

第3章 開発システムの概要

本システムはサーベリストの管理およびサーベ指導を支援する Web ベースの CMS である。利用者として、委託元教員と教員が指導する学生 12 名を想定している。

3.1 システム構成

本システムのシステム構成図を図 2 に示す。

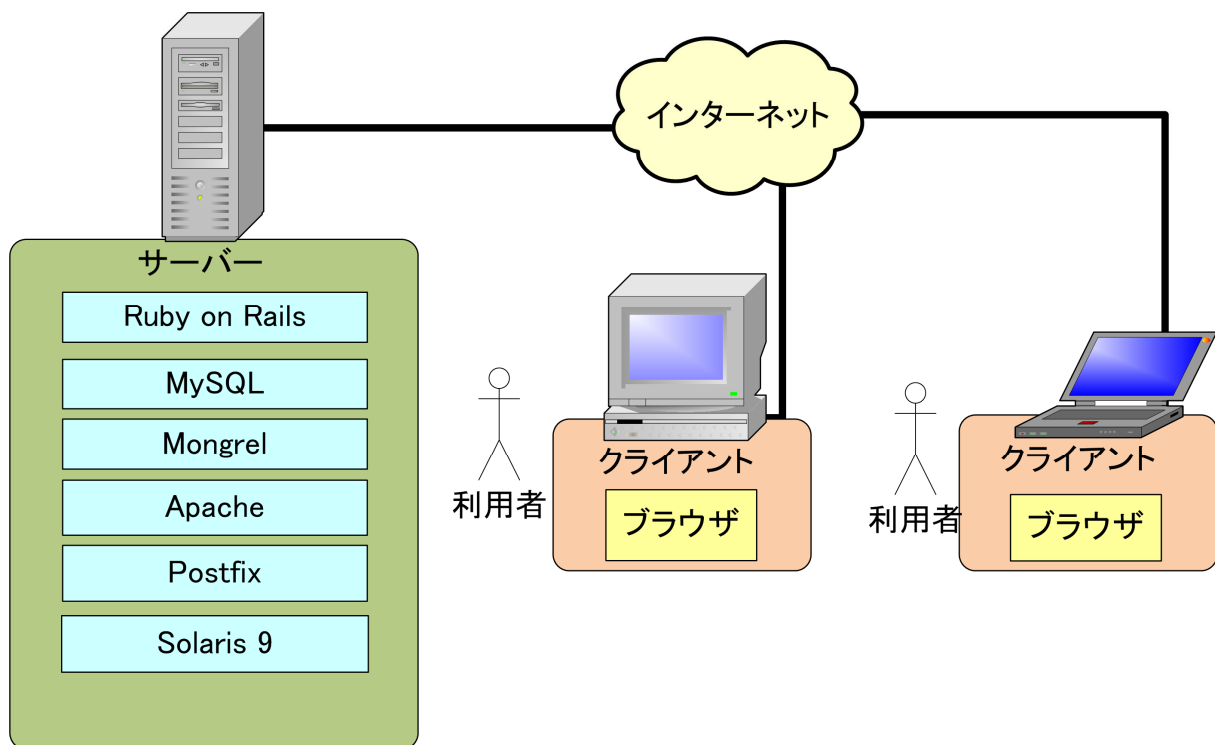


図 2 システム構成図

フレームワークには Ruby on Rails を使用する。DB には MySQL を、Web サーバには Apache と Mongrel を組み合わせて使用する。システムからのメール通知のために、Postfix を使用する。利用者はクライアントマシンのブラウザからインターネットを通じてシステムを利用する。

3.2 開発機能の説明

本プロジェクトで開発したシステムは次のような機能を持つ。

- ・ サーベイリスト管理機能

サーベイした論文のリストを管理する機能である。システムを利用するユーザはそれぞれサーベイリストのページ（図 3）を持ち、このページは他のユーザからも閲覧できる。サーベイリストにはサーベイした論文のタイトル・著者・出典とユーザのコメント、ユーザがその論文を読んだかどうか（既読・未読）が記録される。サーベイリストの論文は既読・未読によって分けて表示される。サーベイリストを編集するにはそのリストの持ち主のユーザアカウントでログインする必要がある。図 3 はリストの持ち主のユーザアカウントでログインした状態のサーベイリストである。

サーベイマラソン支援システム

[トップページ](#) [サーベイリスト](#) [ログアウト](#)

ログインしました

user1のサーベイリスト

読んだ論文

[論文タイトル1](#) [file](#) [編集](#)

著者名1, 会議名1

コメントです。

2009-01-20 01:14:04 +0900 更新

[論文タイトル3](#) [file](#) [編集](#)

著者名3, 出典3

興味深いです。

2009-01-20 01:16:02 +0900 更新

興味のある論文

[論文タイトル2](#) [編集](#)

著者2, 出典2

まだ読んでいません。

2009-01-20 01:15:21 +0900 更新

おすすめの論文

[論文タイトル4](#) [file](#) [編集](#)

著者名4, 会議名4

読んでみてください。

2009-01-20 01:42:40 +0900 に ユーザ2さんから推薦されました

推薦した論文

[論文タイトル1](#) [file](#) [編集](#)

著者名1, 会議名1

関連すると思います。

2009-01-20 01:47:46 +0900 に ユーザ1さんに推薦されました

[論文を追加](#)

図 3 サーベイリストのページ

- 論文情報管理機能

サーベイリストに追加される論文の情報を管理する機能である。システムに登録されている論文の情報は、他のユーザが同じ論文に登録する際に利用することができる。システムは論文毎に詳細情報のページ（図 4）を作成する。このページは論文のタイトル・著者・出典といった基本的な情報を表示する他に、ファイルのアップロードやリンクの貼り付けといった機能も提供する。アップロードするファイルは論文のファイルの他に、公開されている発表資料や、ゼミで使われた資料などを想定している。リンクには著者のホームページや論文に関するデータが掲載されているページへのリンクを想定している。

サーベイマラソン支援システム

[トップページ](#) [サーベイリスト](#) [ログアウト](#)

論文の詳細情報

タイトル: 論文タイトル1 著者: 著者名1 出典: 会議名1 論文ファイル: 論文ファイル.pdf この論文はサーベイリストに追加されています。 <input type="button" value="この論文を他のユーザに推薦"/>	編集
関連ファイル 関連ファイル.pdf - 補足資料です。 発表資料.pdf - 発表資料です。	編集
リンク 著者のホームページです	編集
ユーザ user1 - コメントです。	

図 4 論文の詳細情報

- 論文推薦機能

このシステムを利用する他のユーザに論文を推薦する機能である。推薦される論文はシステムに論文情報が登録されている必要がある。論文を推薦したいユーザは、推薦する論文と対象のユーザを指定し、推薦のコメントを入力する。推薦が完了す

るとシステムは推薦を受けたユーザに確認のメールを送信する。過去に推薦を受けた論文と、他のユーザに推薦した論文のリストは各ユーザのサーベイリストのページ（図 3）に表示される。このリストはサーベイリストの持ち主のユーザアカウントでログインしたときのみ見ることができる。

- ユーザ情報管理機能

ユーザの名前やメールアドレスを管理する機能である。論文がユーザに推薦されるときにメールが送られるが、その宛先としてこの情報が利用される。ユーザ情報を編集するには、そのユーザのアカウントでログインする必要がある。

第4章 システムの開発

4.1 要件定義

プロジェクト開始前にヒアリングを行った時点で、委託元教員の要求には基本的な機能と発展的な機能が混在していることがわかっていました。基本的な機能とは、Wiki を使って実現していた機能をより便利に置き換えるものである。発展的な機能とは、基本的な機能を基礎として、新しい情報や活動を提供する機能である。そこで、反復型開発の中で基本的な機能を作りながら、発展的な機能の要求を固めることにした。イテレーションを繰り返すうちに、委託元教員がシステムに求めている機能を整理することが出来た。大きく分けて以下の6点である。

- ・ サーベイリストの管理
これまで使用していた Wiki の代替となる機能である。これまでは「興味がある」・「pdf を手に入れた」・「読んだ」で分類していたが、これは「既読・未読」と「ファイルの有無」が表現出来ればよいことになる。
- ・ 論文情報の管理
論文のタイトル、著者、出典の情報や、論文のファイル、論文に関連するファイルを1カ所に集約して管理する機能である。これまではたとえ同じ論文であっても学生がそれぞれにサーベイリストに情報を入力していたが、システムが1元的に論文情報を管理すれば、他のユーザのサーベイリストにある論文を自分のサーベイリストに入れるときは改めて情報を入力する必要がない。さらに、ファイルのアップロード機能が論文のファイルや、会議での発表資料、英語論文の和訳テキストなど論文の理解を助けるファイルの共有を容易にする。この機能によって、サーベイに基づいたコミュニケーションを活発にすることが出来る。
- ・ 論文の推薦
これまではメールで行っていた論文の推薦を、システムを通して行える様にする機能である。メールで論文を推薦する場合、教員が論文の情報をメールに書き出す作業と、それを受け取った学生がメールの論文の情報を自分のサーベイリストに入力する作業が発生する。システムを通して論文を推薦することが出来れば、教員のサーベイリストに入っている論文ならば情報を書き出しなくても学生のサーベイリストに追加することが出来る。また、個人への推薦だけではなく、特定のグループへ論文を推薦する機能があれば、多数の学生のサーベイ指導を手軽にすることが出来る。ここまでの基本的な機能に該当する。
- ・ 論文の評価
ユーザによる論文の評価を管理する機能である。サーベイリスト内の論文について、その論文が役に立ったかどうか、自分の研究との関連性はどれくらいかを記述すること

によって、サーベイリストを単なる列挙からより有機的なものにすることが出来る。また、委託元教員のグループでは、学生に「新規性・有効性・技術的質・貢献」の4つの観点で論文を評価するよう指導している。これらの観点は査読に用いられる一般的な観点であり、他人の論文をこの観点で見ることによって、自分が論文を書くときにも意識出来るようにする訓練である。このような観点に基づいた評価もシステム上で行える様にすることによって、サーベイ活動を活発にすることが出来る。さらに、学生の評価の点数を集計することによって研究室におけるその論文の重要さの目安を知ることが出来る。

- ・ サーベイ状況の可視化
学生がサーベイした論文の数を、グラフによって可視化する機能である。日々のサーベイの積み重ねを視覚的に表現することによって、学生のモチベーションが高まり、教員の指導の助けにもなる。また、グループ全体のサーベイの状況を可視化することで、大局的な情報を知ることが出来るかもしれない。
- ・ 他の研究室への提供
本システムをサービスとして外部へ提供する機能である。

本プロジェクトでは、これらの要求のうち、基本的な機能に該当するサーベイリストの管理、論文情報のアップデート、論文の推薦（ただし、グループへの推薦は除く）の機能を実装することが出来た。システムの開発と共に、残された要求も後継者に引き継ぐこととなる。

4.2 外部設計

Ruby on Rails は MVC の概念に基づいており[2]、ユーザインタフェースとなるビューの部分は ERb (Embedded Ruby) という HTML に Ruby のコードが埋め込まれたコードが担当する。本システムは 15 枚のビューを使用している。システムのインターフェースを理解するためのドキュメントとして、ビュー仕様書を作成した。ビュー仕様書の概観を図 5 に示す。

ビュー名 edit_url
概要 論文にリンクを追加・削除する。

サーベイマラソン支援システム

[トップページ](#)
[サーベリスト](#)
[ログアウト](#)

リンクを追加しました

①

リンクの編集

②

<http://homepage.com/> - 著者のホームページです [削除](#) ③

リンクの追加:

④

説明:

⑤

⑥

追加

- ① メッセージがある場合のみ表示される。
- ② リンクのURLと説明。URLをクリックするとリンク先に遷移する。
- ③ リンクを削除する。edit_urlに遷移する。
- ④ 追加するリンクのURLを入力する。初期値は空白。
- ⑤ 追加するリンクの説明を入力する。初期値は空白。
- ⑥ リンクを追加する。edit_urlに遷移する。

図 5 ビュー仕様書

ビュー仕様書には画面の名前、画面の概要、画面のスクリーンショットおよび画面各部の説明が記載される。画面遷移を発生させる部分の説明には、遷移先画面名が記載されている。

4.3 内部設計

Ruby on Rails のモデルは ORM(Object-Relational Mapping)が利用されている[2]。そのためデータベースのテーブルの構造がそのままクラスに反映される。本システムのクラス図を図 6 に示す。

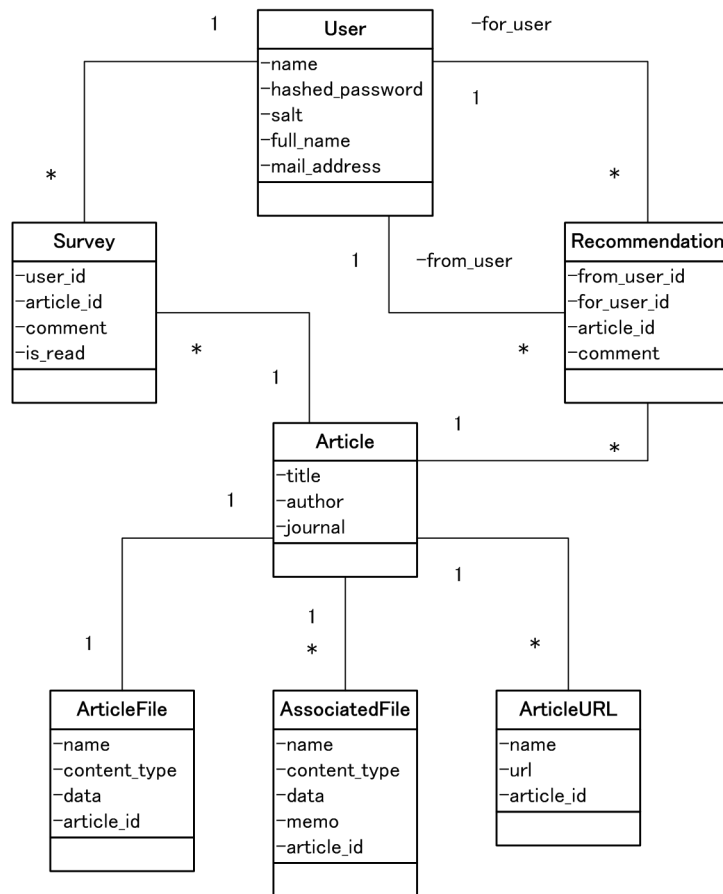


図 6 クラス図

User はシステムにログインするユーザの情報を扱うクラスで、**Article** はシステムが管理する論文の情報を扱うクラスである。**User** と **Article** は **Survey** によって多対多の関係で結び付けられており、これがユーザとユーザがサーベイした論文の関連を表す。この 3 つのクラスがサーベイリスト管理機能において最も重要なクラスである。**Recommendation** は 2 つの **User** のインスタンスと **Article** を結びつけるクラスである。**Recommendation** は論文を推薦するユーザ、推薦されるユーザ、推薦される論文の三者を表す、論文推薦機能を実現するクラスである。**Article** には **ArticleFile**, **AssociatedFile**, **ArticleURL** の 3 つのクラスが関連している。**ArticleFile** は論文そのもののファイルを表すクラスで、**Article** とは 1 対 1 の関係で結び付いている。**AssociatedFile** は論文の和訳や会議での発表資料など、論文に関連するファイルを表すクラスで、**Article** とは多対 1 の関係で結び付いている。**ArticleURL** は著者のホームページなど、論文に関連するサイトへのリンクを表すクラスで、これも **Article** と多対 1 の関係で結び付いている。各クラスの属性やメソッドを詳しく記載したドキュメントは、モデル仕様書という形でまとめられている。

4.4 実装

コードの行数は表 1 コード行数の通りとなった。

表 1 コード行数

	行数
コメント	107
コメントを除くプログラムコード	453
ビューのコード	572
合計	1132

フレームワークの効果により、少ないプログラムコードで機能を実装することが出来た。しかし、プログラムコードに比べてビューのコードは多いように感じられる。HTML の行数は多くなりがちであるが、それだけではなくビューに一部ビジネスロジックが入っていること、コードの繰り返しが存在してしまっていることが考えられる。

第5章 結果の検討及び効果

本プロジェクトでは、委託元教員の要求のうち、基本的な機能の開発を行った。サーベイリスト管理機能は、Wiki に比べて容易にサーベイリストを作成することが出来、サーベイリスト作成のハードルを下げる事が出来る。また、論文の情報はデータベースに構造的に記録されるため、リストの件数が増えた場合の管理も容易である。論文管理機能によって、他のユーザがシステムに登録した論文は簡単にサーベイリストに追加することが出来る。論文に関連するファイルを管理する機能によって、システムが研究室のサーベイの情報タンクとして機能することが出来る。論文推薦機能により、自分のサーベイリストにある論文を、推薦する相手のサーベイリストに直接送ることが出来る。Wiki のサーベイリストの論文を推薦することと比較すると、サーベイリストの論文情報をメールに書き出す部分と、メールの論文情報をサーベイリストに追加する部分の手間が削減されることとなる。これらの機能により、システムは研究調査活動を従来よりも強力に支援することが出来る。

サーベイリストの作成や論文の推薦をスムーズに行えるようになることで、サーベイに基づいたコミュニケーションが発生しやすくなる。論文の詳細情報のページでは、その論文をサーベイリストに追加しているユーザのコメントを見ることが出来る。このコメントを自身の研究調査の助けにしたり、自分の考えを発信したりする場として利用することも出来る。論文の推薦が容易になったことで、教員が学生に推薦するだけでなく学生同士で論文を紹介し合うことも容易になった。コミュニケーションを活発にすることで研究調査に組織的に取り組むことが出来るようになり、効率やモチベーションを高めることに役立つ。

プロジェクト開始の段階では曖昧だった発展的な機能を、具体的な機能に固められたことも本プロジェクトの成果である。本プロジェクトで開発したシステムと要求定義を引き継いだシステムには、革新的な方法で研究調査活動を支援することが期待される。本プロジェクトではそのようなシステムの基盤となる機能を開発することが出来た。さらに、システムの引き継ぎを助けるためのドキュメントを作成し、今後の開発につなげることができた。

第6章 失敗およびあい路事項

本システム開発における失敗として、本番サーバでの稼動が遅れてしまったことが挙げられる。イテレーション毎に動くプログラムが出来上がるといっても、開発環境での稼動であるため、委託元教員や利用者がいつでも触れる状態にならなかった。そのため顧客のフィードバックを得られるという反復型プロセスの利点を十分活かすことができなかった。原因として、サーバ管理に関する知識の習得が追いつかなかったことが考えられる。**Web** アプリケーションの開発は、様々な技術の集合であるため難しいと言われている。今回はフレームワークを利用することによって開発の大部分が簡単に開発できるようになったが、デプロイの部分ではつまづいてしまった。サーバ管理の知識は今後の開発においても重要だと考えられるので、ぜひとも身に着けたい。

十分なテストが行えなかったことも本プロジェクトの失敗事項である。フレームワークではユニットテストから統合テストまでがサポートされており、テストコードを書きながら開発を進めていくことでバグが少ないだけでなく、変更に強いコードを作成することができる。しかしながら本プロジェクトでは動作確認程度のテストしか行えなかった。システムの引き継ぎの前に、十分なテストの実施が必要である。本来の反復型開発ではイテレーション内でテストも完結させなければならない。短いサイクルの反復型開発には開発者のスキルが求められるという理由を知ることができた。

第7章 結言

委託元教員は指導する学生にサーベイリストを公開させている。リストの公開には Wiki を利用している。しかしながら、現在の Wiki ではサーベイに基づいたコミュニケーションを活発にするには不十分であった。

筆者はヒアリングを行うことで、委託元教員は単なる Wiki の代替ではなく、もっと発展的な機能をもつシステムを求めていることがわかった。その要求はまだ具体的なものになっておらず、様々な発展の可能性が考えられた。そこで、反復型システム開発プロセスを採用し、基本的な機能を開発しながら徐々に要求を固めていくことにした。イテレーションを繰り返すうちに、委託元教員の要求は明らかになった。

開発したシステムは委託元教員の要求のうち、基本的な機能の部分を持つもので、従来の Wiki を用いたサーベイリストに置き換えることが出来た。それだけではなく、論文情報や関連するファイルを共有することが出来、論文の推薦もよりスムーズに行うことが出来た。これらの機能により、サーベイに基づいたコミュニケーションを提供する基盤が整った。

本システムは今後も引き継がれて開発が進んでいく予定である。本プロジェクトで作成した成果物は開発の引き継ぎを助けることを目標に作成された。今回開発した機能に加えて、要件定義で明らかになった発展的な機能が開発されることで、さらにサーベイ活動を盛り上げることが出来ると考える。

謝辞

本プロジェクトの遂行にあたり、三末和男准教授には委託元教員として魅力的なアイデアを数多く賜りました。また指導教員として終始暖かいご指導を頂き、プロジェクトをやり遂げることが出来ました。心から感謝申し上げます。

田中二郎教授には多くのご助言を頂きました。研究室での日々は今後の人生の宝となります。心より御礼申し上げます。

専修プログラム担当の菊池純男教授、駒谷昇一教授には大変貴重なアドバイスを頂きました。ここに感謝の意を表します。

ヒアリングにご協力頂いたインタラクティブプログラミング研究室 NAIS チームの皆様には有り難いご意見を頂きました。厚く御礼申し上げます。

最後に、物心両面で支えてくれた両親に感謝いたします。有り難うございました。

参考文献

- [1] Beck, et. al., Manifesto for Agile Software Development,
<http://www.agilemanifesto.org/>
- [2] Dave Thomas, David Heinemeier Hansson. Rails によるアジャイル Web アプリケーション開発 第 2 版. オーム社, 2007.