

まねきねこプロジェクト

最終成果報告書

Hanavi システム製作記

松尾壮紘

松田航
福田宙音
荒木恵

目次

1	はじめに	4
1.1	まねきねこプロジェクトとは	4
1.2	プロジェクト定義書	5
2	まねきねこプロジェクトのあゆみ	4
2.1	まねきねこプロジェクトの発足	6
2.1.1	案件選択	6
2.1.2	メンバー	6
2.2	企画	8
2.2.1	アイデアの持ち寄りと選択	8
2.2.2	顧客とのミーティング	17
2.2.3	Hanabi システムの誕生	18
2.2.4	企画フェーズの総括	18
2.2.5	システム提案書（プロジェクト提案書）	19
2.2.6	プロジェクト計画書	19
2.3	設計	38
2.3.1	技術調査合宿	38
2.3.2	入出力仕様（プロトコル）の決定	39
2.3.3	それぞれの設計－検索部編	44
2.3.4	個々の設計－表示部・サーバサイド Java・リンクデータ作成部編	45
2.3.5	設計フェーズの総括	45
2.3.6	画面表示仕様書	47
2.3.7	Hanabi システム全体設計	48
2.4	実装	57
2.4.1	検索部の実装	57
2.4.2	表示部・サーバサイド Java・リンクデータ作成・設定ファイルの実装	58
2.4.3	モジュールテスト	59
2.4.4	検索部テスト	59
2.4.5	表示部テスト	60
2.4.6	結合テスト	60
2.4.7	実装フェーズの総括	60
2.5	品質保証テストチェック	62
2.6	評価	78
2.6.1	評価フェーズの要約と総括	78
2.6.2	ユーザーテストの方法	79

2.6.3	ユーザテストの結果.....	80
2.6.4	ユーザーテストに利用したアンケート	85
3	おわりに	87
3.1	プロジェクトの総括	87
3.2	プロジェクトマネジメント報告書.....	88

1 はじめに

1.1 まねきねこプロジェクトとは

まねきねこプロジェクトは、慶應義塾大学デジタルメディアコンテンツ統合機構（DMC 機構）の検索システムである、DMC システムの拡張機能を作成するために結成された。

DMC 機構では、慶応の学術コンテンツの利用促進を目指して、検索システム「DMC システム」を開発している。この検索システムは、メタデータの自動/手動付加、メタデータによる検索、地名から位置情報を算出し、地図上にコンテンツをマッピングする、などの「新しい検索」の機能を持っている。

まねきねこプロジェクトでは、このような DMC 機構の目的である「慶応の学術コンテンツの利用促進」を目指して、ターゲットを、「慶応の研究者に出会いたいベンチャー企業の人、および学生」とし、新しい検索システムを企画、開発した。

開発したシステムは、Hanavi システム（Human Association Navigation System）という、慶応の研究者と研究分野をグラフ表示するシステムである。検索語を1つ入力すると、それに関連する研究者、およびその研究者の関連研究分野を表示することができる。

本報告書では、まねきねこプロジェクトの発足から Hanavi システムの作成、システムの評価までの一連の流れを時系列に報告し、Hanavi システムの問題点と展望を述べる。

1.2 プロジェクト定義書

プロジェクト名 まねきねこ

より多くの人に見てもらいたい、先客万来のイメージから「まねきねこ」に決定
ぼんやり見ているとゲシュタルト崩壊を起こして ねこまねき と書いてしまう状態に陥
るので注意

顧客

DMC 嶋津恵子 先生

ターゲット層

大学(研究室)

企業

受験生

在校生

卒業生

プロジェクトの目的

DMC の調査により、慶應大学の研究資料へのアクセスが非常に少ないことが分かった
大学の本質である研究へのアクセスが少ないのは非常に嘆かわしいことである。
本プロジェクトにより、学内の貴重な資料をいかに効率的に見せるかを追求するものとする。

現行システム

DMC システムがあり、内部には Yahoo や Google といった大手検索エンジンによる
学部内コンテンツクローラーがある。

2 まねきねこプロジェクトのあゆみ

2.1 まねきねこプロジェクトの発足

2.1.1 案件選択

コラマネでは、まずはじめに PM が、どの案件を扱うかを選ぶ。PM である松尾さんは、複数の案件の中で、どれを選択するか迷っていた。

このとき残っていた案件は、アルバム Web 作成システムと、DMC の案件であった。アルバム Web 作成システムは、画像の入ったフォルダと、HTML テンプレートから、自動で画像アルバム Web ページを作成するというもので、難易度も低く、興味を抱いている学生も 4 名ほどいた。もうひとつの、DMC（慶應義塾大学デジタルメディアコンテンツ統合機構）の案件は、作るものが決まっておらず、ただ、DMC システム（慶應義塾内の各種コンテンツ検索システム）に関連したシステムを作ること、顧客が DMC 機構の嶋津先生であることしか決まっていなかった。

松尾さんは、あえて難易度が高く、読めないプロジェクトである、DMC システムを選ぶことに決めた。

2.1.2 メンバー

PM によるプレゼンの結果、メンバーが決定した。何を創るのかイメージがしにくいことから、このプロジェクトを希望した学生は少なかった。プロジェクトメンバーは以下の 3 名である。

1. 松田航

第一希望は水槽共有システムの開発（高橋 PM : nico2 ライブプロジェクト）。このプロジェクトは第二希望である。コラマネの履修は 2 期目で、先学期のコラマネで、DMC システムの一部モジュールである g-mod システム（地名→地理情報検索・記録モジュール）の開発プロジェクトに参加していた。今回のプロジェクトを第二希望に入れた理由は、もう一度 DMC 関係のシステム開発に携わり、今度こそ、結果を残したい、という気持ちがあったためである。

2. 福田宙音

第一希望はアマゾンジャパンのアンケートシステムの開発（石塚 PM : firstChoice プロジェクト）であり、このプロジェクトは第二希望である。コラマネには初参加で、プログラミング経験もない。もともとはある企業のシステムを開発してもらうために、顧客として参加予定であったが、その企業がシステム開発を行わないことになったため、プロジェ

クトメンバーとして参加することになった。

3. 荒木恵

大岩研究室所属の修士1年である。コラマネの履修者が若干少なく、プロジェクトメンバーが足りなかったため、このプロジェクトに参加することになった。コラマネ参加は4期目であり、以前、DMCシステムのインターフェース開発プロジェクト「DDSS」に参加していた。プログラミングは嫌いではないが、得意でもなく、今までのコラマネプロジェクトではあまりプログラムを書いていなかった。

このようなメンバーで、まねきねこプロジェクトはスタートした。

2.2 企画

2.2.1 アイデアの持ち寄りと選択

当初、クライアントから要求を聞いたが漠然としていて、クライアントからもシステムの提案をして欲しいとの要求があったため、質問し続けても迷惑だと考え、チームからシステムを提案することにした。

だが、DMC とクライアントが何をしていた、何をこちらに求めているかがわからなかったため、クライアントに関連する『利用者のコンテキストに従ったコンテンツ検索・編纂システム』と『学内のデジタルコンテンツの公開と「知の編纂と再編」のチャレンジ』という資料を読んでみた。

以上の資料を基に、どの方針が正しいかわからなかったため可能な限り案を出し、その結果、以下のような18案が出揃った。企画詳細があるものについては、企画紹介も掲載した。

*おすすめDMC (4)

一般 (?)

検索キー：一般ワード

検索結果：関連ワード

*ヒトモノ検索 (5)

学生・研究者

検索キー：全般

検索結果：(人) (論文)

*ニューストラベラー (5)

*受験の足跡 (2)

*歴史表検索 (5)

その年度の情報が知りたい人

検索キー：年度 (西暦)

検索結果：その年度の情報

*逆場所検索 (5)

検索キー：緯度・経度

検索結果：その場所に関連したコンテンツ

*KeioNavi (7)

一般の研究に興味がない人

*InRelation (1 ?)

産学連携に興味がある企業の人が使うか?

検索キー：(企業名・大学名・研究者の名前) (研究していること)

検索結果：(研究内容) (研究分野) (研究主催者名)

*EyeEyeEye (7)

受験生が使うか?

検索キー：(クリックブルマップ=写真の裏にすでに検索キーが存在する)

検索結果：検索キーに応じた検索結果が出てくる (絵なども含む?)

■企画名

カドオリ — コレのココが役に立つ! —

■コンセプト

図書館で角を折ったページを見つけると、いい情報があったりする。

コレのココが、コレに役立つ! というタグを簡単につけられるシステム。

■作るもののカテゴリ

ソーシャルタギングシステム
 (「紙 copi」の CoRich システムに類似)

■企画概要 (What)

巨大な Web ページや、ブログの記事から一部を選択し、
 「○○に役立つ」というタグをつけられるようにするシステム。
 タグはページを部分選択→右クリックメニューから「カドオリ！」を選ぶような
 簡単な操作でつけることができる。
 タグは共有され、タグを検索キーとして「部分的な情報」を検索できる。

■背景 (Why)

ブログのある 1 記事が研究に役立つ情報だったり、
 巨大なページの中のある 1 節が役立つ文章だったりすることがあるが、
 「お気に入り」や「アドレスの記録」ではこれらの情報だけを取り出して
 記録しておくことはできない。
 かといって、ページの一部を切り取って保存しておいても、
 ページの文章の前後関係が多少は分からないと、使い物にならない
 ということもある。
 したがって、「ページの中のココ！」をお気に入りにしたり、
 それにタグをつけたりしたくなる。

■対象者 (Who)

自分の研究テーマに役立つ情報を集めている学生・研究者

■実施体制・スケジュール (How・Where・When)

 ■企画名

InformationCockroach ―ゴミになった情報は……

■コンセプト

情報が見つけにくいのはなぜか？それは、「捨てられない」から。
 どれがイライライものか分からない。自分の部屋だったら、
 ホコリをかぶっている雑誌なんかすぐに捨てるのに。
 みんなが使っていない情報には、「ゴキブリ」がついてしまい、
 掃除をしないと見られないようにしてしまうシステム。

■作るもののカテゴリ

ページ閲覧ランキングシステム

■企画概要 (What)

皆から閲覧されていない情報には、自動的にゴキブリが湧くシステム。
 ゴキブリ〇匹以上のページは検索しない、など、
 ゴキブリの数で検索対象を指定することができる。
 その情報を見てもらいたい人や、役に立つと思う人は
 「掃除」をすることでゴキブリを追い払うことができる。

■背景 (Why)

「役に立つランキング」の上位 100 件の情報だけを検索対象にしても
 役立つ情報は得られないが、
 できれば「ゴミのような情報」にはわずらわされたくない。
 ページ閲覧ランキングの下位を検索対象からはずすことで、
 この要求にこたえられないだろうか？

■対象者 (Who)

ゴミのような情報にわずらわされたくないユーザ

■実施体制・スケジュール (How・Where・When)

 ■企画名

ReserchersEye ―○○先生、お邪魔します！―

■コンセプト

慶應の先生の研究室にある情報を根こそぎ検索できるシステム。
 これらの情報が全部頭に入れば、あなたも○○先生になれる！

■作るもののカテゴリ

検索履歴追跡システム

■企画概要 (What)

慶應の先生が研究のために検索を利用した履歴を公開することで、研究者が見ているページや、検索のプロセスを学ぶことができるようにするシステム。

■背景 (Why)

先生は必ず、新しい情報をいち早くつかんでいる。
また、世間の流れも、とてもよく見ているようだ。
そんな先生のネット検索・閲覧履歴を見ることができたら、
きっと新しい情報や新しい時代の流れをつかむことができる。

■対象者 (Who)

先生の知識や知恵に追いつきたい学生

■実施体制・スケジュール (How・Where・When)

■企画名

Today our Teacher says... ーあなたはだれの弟子？研究室のうわさ話ー

■コンセプト

あの研究室、興味ある…けれど、誰が入っているんだろう？
同年代なら話やすいんだけど…誰か、具体的に何をやってるところなのか、
ちょこっと教えてくれる人いないかな…？

■作るもののカテゴリ

ヒューマンネットワーク検索システム

■企画概要 (What)

自分とかかわりのある人をたどって、自分が興味を持った
研究室に所属している人を探すことができるシステム。

■背景 (Why)

学部2年生の秋は、所属研究室を探す時期。
しかし研究会のシラバスを読んでみても、いまいちピンとこないことが多い。
実際に研究会に所属している人の話を聞いてみたいが、
知らない人には声をかけづらい。
友達の友達が、あるいはサークルの先輩の彼女が、研究室にいたら、
話が聴けるかもしれない。けれどどうやって「その人」を探せばいいんだろう？

■対象者 (Who)

所属研究会を決定したい学部生

■実施体制・スケジュール (How・Where・When)

■企画名

DreamBlock ー僕らの夢の積み重ねー

■コンセプト

受験生は大学生活に向けて夢いっぱい。勉強してみたいこと、議論してみたいことを
いっぱいもっている。
夢にあふれる受験生が「大学生になったらこんな研究に関わってみたい！」と
思う情報をコレクションし、共有できるシステム。

■作るもののカテゴリ

ページ公開・閲覧システム

■企画概要 (What)

大学で学びたいことがある受験生が、自分の興味ある研究や
活動の情報に対して、書評ならぬ「ページ評」を書いて、
公開できるようにするシステム。

■背景 (Why)

受験生は「大学で学びたいコト」へのモチベーションが高い。
そして情報を探すためにネットに費やす時間も、長い。
自分が大学でやりたいこと、興味のあることを調べ、

「お気に入り」にコレクションして、受験へのモチベーション維持に役立てている受験生も少なくない。
受験生が興味を持っている情報が何か知ることができれば、研究室と新入生のマッチングや、新入生同士の出会いに活かせる。

■対象者 (Who)

慶應を受験する受験生

■実施体制・スケジュール (How・Where・When)

■企画名

コトバの赤い糸ーコトバでつなぐ、私とあなたの興味対象ー

■コンセプト

こんなコトバで検索する人、いないだろう…と調べてみたら、実はあの教授も、コレで調べてた！
そして、その検索結果から、これをクリックして読みにいったらしいぞ。どれどれ、私も読んでみようかな…

■作るもののカテゴリ

検索履歴検索サービス

■企画概要 (What)

検索ワードが同じだったユーザを表示し、そのユーザがその検索結果からのページを選んだか、を表示するシステム。

■背景 (Why)

検索ワードが同じであるということは、同じような情報が知りたいということ。同じ検索ワードで検索をかけた人は誰か、どのページにアクセスしたかを知ること、同じ興味を持つ人と出会うことができ、役に立つページがどれか分かる。

■対象者 (Who)

同じ興味を持つ人と出会いたいと思っている人

■実施体制・スケジュール (How・Where・When)

■企画名

SearchStoryTeller ーコレを見て、この絵を見てからコレを見ろ！ー

■コンセプト

検索して得た情報は、「それだけ」で存在するものではなく、「検索」というストーリーの中の一部。
「検索結果」を単体ではなく、いくつかの「検索結果」が時系列でつながったストーリーだと見なして、「検索結果ストーリー」を公開するシステム。

■作るもののカテゴリ

検索履歴公開サービス

■企画概要 (What)

ある人がある情報を探して歩いた足跡を記録し、その中で役立つと思われるものを時系列にならべて公開できるシステム。公開された「検索結果ストーリー」には。

■背景 (Why)

検索で得られる情報は断片的なものが多く、単体では理解が難しい。そのため、人は検索に検索を重ねて、知りたいコトを理解しようとする。しかし「お気に入りに追加」などでは、その検索の文脈が失われてしまう。検索のプロセス自体を保存して、プロセスごと振り返りたいときがある。

■対象者 (Who)

■実施体制・スケジュール (How・Where・When)

■企画名

Idea Tree ー検索ワードの根っこをつなぐー

■コンセプト

自分が検索の「根っこ」にした言葉から、
どんどん検索ワードを増やしてゆく。
たとえば「花」「チューリップ 育て方」「園芸 方法論」
他の人は「園芸」「園芸店運営」「経営 地域密着」
たくさんの人の検索ワードがつながる。「花」から「地域密着」までが。

■ 作るものカテゴリ

検索履歴記録サービス

■ 企画概要 (What)

はじめの検索ワードと、その次に入力された検索ワードを記録し、
多くの人の「検索記録」をつなげることで、
検索ワード同士をつなぐ「木」を作るシステム。
この木は、好きなワードを選んで「根っこ」にして表示することができる。

■ 背景 (Why)

検索ワードには人の思考過程や、人がどんな言葉とどんな言葉を
近いと見ているかが反映される。
他者と自分が感じている「言葉の意味」の差を視覚的に表示でき、かつ、
他者の思考過程をたどることができたら面白い。

■ 対象者 (Who)

言葉の意味や解釈に興味のある人

■ 実施体制・スケジュール (How・Where・When)

18案という数は、クライアントに提示するにはあまりに多いため、これらをカテゴリ分けして、違う性質の3案に絞った。その3案は以下のものである。

企画提案書（3案）

本文書の目的

私たちの作るシステムの目的（なぜシステムを開発しなければならないのか？）について理解を深めるために、我々の考える「システムの目的」を嶋津先生に提示し、嶋津先生に、「どの部分が嶋津先生（DMC 研究機構）のニーズと合致している」「どの部分が合致していないか」を指摘していただき、システムの目的を正しく理解するための手がかりとすること。

【SearchStoryTeller】

システム定義：

コンテンツ間の関連性を示すデータを集めるために、ユーザが複数コンテンツ間の関連を示すタグをつけられるようにすることによって、ユーザによる関連性データの提供を促進する、ソーシャルタギング型システム

コンセプト：

検索して得た情報は、「それだけ」で存在するものではなく、「検索」というストーリーの中の一部分。「検索結果」を単体ではなく、いくつかの「検索結果」が時系列でつながったストーリーだと見なして、「検索結果ストーリー」にタイトルを付けて公開するシステム。

概要：

ユーザは複数のコンテンツから部分を抜き出し、それを並べたものにタイトルをつけ、ページとして公開することができる。

ユーザがつけたタイトルは、参照元のコンテンツにも反映され、参照元の複数のコンテンツの関連を示すデータとして蓄積される。

何が嬉しい：

ユーザがコンテンツ間の関連を示すデータの作成に協力することによって、関連データの精度が上がり、目的の情報をすばやく得ることができるようになる。

背景：

コンピュータによって、情報の内容を判断することは可能である。（スパムメールのベイジアンフィルタのように）

そのためには質の良い学習データが必要であるが、しかし、インターネット上に散らばっているデータは、関連性を示すことを目的として作られたデータではない。

そのため、コンテンツ同士が関連しているかどうかの判定の精度が高くないと考えられる。

ユーザに、関連を示すためのタグ付けを促すことによって、コンテンツ同士が関連しているかどうかを判断するための質のいい学習データが得られるのではないかと考えた。

企画立案の背景：

（私の考えた前提）

嶋津先生は、現在一般的なデータ蓄積・検索サービスは、人間の行う分類・関連の判断・連想などの高度な情報処理に対応できるものではないと考えている。

嶋津先生は、人間の行う高度な情報処理を支援するような、新しいデータ蓄積・検索サービスを研究している。

このような検索サービスを構築するためには、

- データ蓄積の方法論
- 既存のデータから、人間が判断するような「意味」を抽出する方法論
- 抽出した「意味」をわかりやすく表示するインターフェース

などが必要である。

今回のシステムは、「意味」を抽出する方法論を支援するために、「コンテンツ間の関連データ」をユーザに提供してもらうシステムである。

（企画のアピール）

コンピュータが人間と同じように分類や判断を行うためには、質のいい学習データが必要であるが、Web上にあるコンテンツを解析するだけでは

- コンテンツの種類が多様多様であること
- 判断すべき関連や分類も多様多様であること
- Web上のコンテンツにはノイズが多いこと

などから、質のいい学習データが取れないと考えられる。

一方、ユーザは、面倒なフォーム入力などは嫌いだが、関連を示すデータはあちこちに投稿している。(ex. ニュースサイトの URL の羅列、知恵袋の中の URL 羅列など)
この、ユーザの投稿を、コンテンツ間の関連性を示すデータとして蓄積しないのはもったいない。

(利用例)

たとえば学生 A が「情報教育」で検索をかけたうち、「大岩研究室」「情報教員の Blog」「教科情報カリキュラム実践集」「CE 研究会」を有用な情報だと感じたとする、その学生は「情報教育」というタイトルを付けて、これら 4 つのページから情報を抜き出して作った検索ストーリーページを簡単に公開することができる。

次に、学生 B が「情報教育」で検索をすると、学生 A が作成した検索ストーリーページが表示され、簡単に有用な情報に飛ぶことができる。

また、ある学生 C が「教科情報」で検索をかけると、「教科情報カリキュラム実践集」が検索結果として表示されるだけでなく、「大岩研究室」「情報教員の Blog」「CE 研究会」「情報教育 (検索ストーリーページ)」が関連ページとして表示される。学生 C は、これらの検索結果を組み合わせ、新たに「教科情報」という検索ストーリーページを作ることができる。

これを繰り返すことによって、コンテンツ間の関連データが強化されてゆく。

検索キー：ユーザーの付けたタグ

検索結果：ユーザーが再編集してタグ付けした検索結果

利用者：学生・研究者を想定

【In Relation】**システム定義：**

企業や研究者の方の、慶應の研究情報に関する検索率を増やすために、検索した言葉とそれに関連する言葉の関係を樹形図で示すことによって、どこで共同研究がしやすいかをわかりやすくする、インターフェース改善系システム

コンセプト：

ある企業や研究者の方が、慶應大学と共同研究したいと考えた時に使用される。自社名や大学名を打ち込むと、それと関連する研究を行っている研究会やメンバーが樹形図形式で見られる。それにより、慶應と企業や他大学との共同研究自体が増える可能性が高くなる。

関連検索に特化した検索システムだといえる。

概要：

検索した言葉とそれに関連する言葉の関係を樹形図で示してくれる検索システム。

何が嬉しい：

慶應としては、情報を発信する目的である、「この大学のこの人と何かやってみたい」ということを達成しやすい状況になる。

企業や他大学としては、共同研究ができる機会が増えるという嬉しい状況になる。

背景：

慶應義塾大学は共同研究などにより、リソースを確保したいと考えているが、慶應ネットへの研究情報に関するアクセスは、0.2%と低い。大学が情報発信をする目的である、「この大学のこの人と何かやってみたい」ということを達成しきれていない現状がある。

企画立案の背景：

嶋津先生にアピールするポイント：

おそらく、企業や研究者の方が慶應の教授と出会うのは、学会であったり、何かしらのパーティだと考えられる。今までリアルな現場で出会っていたからこそ、慶應ネットでの検索率があれほど低かったのだろう。だが、ネットでの研究情報に関する検索率を増やすことができれば、今後、共同研究が増える可能性が高くなる。

今までの検索システムでも、確かに、関連検索で関連する単語が何かはわかった。だが、視覚的にも見づらいうし、関連性のある単語の数は制限されていた。そこを改善し、例えば、ある企業と大岩研の関連性が視覚的にわかれば、積極的に調べてみようという気になるのではないだろうか。

検索キー：(調べたい単語)

検索結果：(調べた単語と一緒に打ち込まれた一般ユーザーの検索履歴を基にした樹形図)

利用者：(企業や研究者)

【celebrity search】

システム定義：

ある分野に関連している人物を見つけるために、
検索キーから人物だけを表示することによって、
その分野に関連する人物と、その人物に関連するコンテンツを表示するシステム

コンセプト：

あの分野ではどのような人が、どのような研究をしているのだろうか？

もしその人達の一覧を見つけることができれば便利だろう。

その人達に関連する研究室、発表した論文も簡単に検索できると嬉しい。

背景：

何かを検索すると、多くの情報を検索することができる。しかし、実際にはそこから必要な情報を自分で探さなくてはならない。そこで、検索される情報をあらかじめ絞ってしまうことを考えた。

企画立案の背景：

DMC システムの自然言語処理を活かすことができるありそうでなかったものである

外部の研究者が、慶應内部の研究者を探すのに役立つ

研究者だけでなく、在学生在が研究室を探すのに役立つ

他のシステムにも転用できる（地名など）

検索されるものを絞ることで、精度の高い検索をすることができる

概要：

検索キーから人物一覧を表示する。表示された人物は、その人物の関係するコンテンツ（自身が所属する研究会のページ、発表した論文など）へのリンクが表示される。

検索キー：調べたい分野に関連していると意味を持つ文字列

検索結果：検索キーに関係する分野と関わりのある人物の一覧が表示される

利用者：特定の分野に関連している人と、その人物に関連しているコンテンツを探したい人

この3案を出した後、分析・調査はそれぞれ別々に行っていたことが原因で、各自イメージするシステムの目的がずれていることが判明した。そこで、チーム内でシステムの目的を「慶應義塾に蓄積されている研究成果や学術成果を、学外にも広く開放し、専門家だけでなく一般の方々にも自由に利用できる仕組み」を支援するという目的に絞った。

2.2.2 顧客とのミーティング

チーム内で考えたシステムの目的が、クライアントの考える目的から外れていないかを聞くために、クライアントに連絡を取る必要が出てきたが、非常に多忙で連絡が取れないことがあったため、対策を練った。以下が対策項目である。

- 連絡窓口を決める。
- クライアント側の担当者を変更して頂くことも視野に入れる。
- 返事を待っているのではなくこちらから連絡をする。
- 話しが出来るときにできるだけ多くのことを話す。

また、クライアントから、結論が先になく、メールの仕方が下手であるのご指摘を頂いたため、メールのチェックリストを作成することにした。以下がそのチェックリストの項目である。

- 表題(Subject)は、あいさつではなく用件をズバリ書く
ex. 「○○という件について」「△△と○○の違いについて」
- 誰宛に書いているのかを本文上部で明示する
- 最初に要旨を書く
- 関連する話題なら「返信」する。新規の話題を振るときは、MLのメールにリプライせず新規でメールを書く
- 本文はできるだけ35～40文字程度（全角）で改行することを心がける
- 3～5行で1段落にして、改行を空ける
- 署名が入っているか確認する。署名の内容は氏名、大学名（会社名）、メールアドレス、電話番号、FAX番号などである
- 送信先アドレスと件名、本文を再確認する

その後、クライアントと連絡が取れ、直接会って、3案をそれぞれクライアントの前でプレゼンする機会を得た。その結果、どれも甲乙付けがたく良いが、実現性が見えないという問題点をそれぞれが抱えているため、松尾PMが1案に絞り、きちんとした提案の形にしてくるようにご指摘をいただいた。

そこで、案を1つに絞ることにした。

2.2.3 Hanabi システムの誕生

「研究者を検索できる」という実用性 (CelebritySearch) と「関連図表示」というインターフェース (InRelation) の2つの案を良いところ組み合わせ、「人の関連を図で表示するシステム」にすることを、松尾 PM が決定した。

その後、「慶應のコンテンツの学術利用促進に役立つ」という目的に当てはまるものか考え、また、DMC システムを使って実際に (人力で) 検索を行って実現可能性を確かめ、「検索ワードを入力すると、その分野に関連のある人物 (慶應の教員・研究者) の一覧および、その人物の関連研究分野 (慶應の教員が教員検索システムに自ら登録している研究分野) に関係のある人物の一覧を表示するシステム」という提案に詰めなおした。

その後、11月1日に DMC (三田) を訪問し、提案書をクライアントに提出し、①DMC の担当範囲 (負担) を明示することと、②用語定義をきちんと書くことという2点の「ここを修正すれば承認する」という修正点を指摘していただき、提案書を修正した。その後、修正案を提出し、クライアントから承認を頂いた。

2.2.4 企画フェーズの総括

システムの企画をチームの側で立て、クライアントに提案し、承認を頂くという戦略でこの企画フェーズは推し進めたが、この戦略は、クライアントと綿密に連絡が取り合える場合に関しては成功する可能性が高いが、今回のプロジェクトでは厳しかったといえる。

上記の戦略を取ってよかったのは、要求分析の必要性に気付き、それに関わることが学習できたことだ。また、連絡が取れない場合の対応策を練れたのも学びとして良かった点として考えられる。

今回作成したシステムを決めるまでもにも紆余曲折があったが、決定の要因は、①慶應義塾に所属する専任教員・研究者についての研究業績・プロフィールなどの情報を検索できる、慶應義塾研究者情報データベース (K-RIS) があったという理由と、②DMC の検索システムが使えるという理由、また、③メンバーのうち2名が JAVA でプログラムを組んだ経験があるという、以上の3点の理由による。

2.2.5 システム提案書（プロジェクト提案書）

2.2.6 プロジェクト計画書

2.2.7 中間発表資料

Maneki Neko Project

まねきねこプロジェクト提案書

更新日 2007/12/18

まねきねこ@三田DMC

Maneki Neko Project

システム提案

まねきねこ@三田DMC

Maneki Neko Proj

システムの目的・前提(1/2)

- 現状
 - 研究分野と人のつながりを知りたい場合、キーワード検索→ページ閲覧→新たなキーワードの抽出→キーワード検索...というステップを何度も繰り返さなければならない
- 目的
 - 本システムでは、ユーザがキーワードを入力すると、関連する人名と研究分野をグラフとして表示する。これによって、研究分野と人のつながりを知るために必要な検索ステップ数を減少させる。
- 前提
 - 特定のキーワードから関連する人名のみを抽出し、その数を数えることができるモジュールがDMCシステム内にあることを前提とする。

まねきねこ@三田DMC

Maneki Neko Project 2007 Fall

Maneki Neko Proj

システムの目的・前提(2/2)

- いままでの×
 - 多くの検索ステップ数
 - ユーザの高度な検索スキルが必要
- これからの○
 - 検索ステップが1回に
 - ユーザの検索スキルは必要ない

まねきねこ@三田DMC

Maneki Neko Project 2007 Fall

Maneki Neko Proj

システム概要(1/3)

- 任意のキーワードで検索すると、関連する人名と研究分野の繋がりを図のようにグラフ表示する。

まねきねこ@三田DMC

Maneki Neko Project 2007 Fall

Maneki Neko Proj

システム概要(2/3)

- 検索キーワードから研究者と研究分野を表示するシステム
 - 人名を抽出するためDMCの検索システムを利用する
 - 研究分野を抽出するため各学部の教員リストをあらかじめファイル化したものを利用する

まねきねこ@三田DMC

Maneki Neko Project 2007 Fall

システム概要(3/3)

Maneki Neko Proj

- 検索ステップ1
 - 検索キーワード「プログラム」で検索。
 - →「大岩・萩野...」と人名のリストが出る
- 検索ステップ2
 - 人名「大岩」で教員検索
 - →研究分野が「認知工学・情報教育・ソフトウェア工学」と出る
- 検索ステップ3
 - 研究分野「認知工学」で検索
 - →「川島・大門」と人名が出る。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

用語定義

Maneki Neko Proj

- 「検索キーワード」
 - 検索キーワードとは、はじめにユーザが入力する語のことです。
 - この語は何でもかまいません。
- 「人名」
 - 人名とは、「慶應の教員名」もしくは「その他の人名」のことです。
 - 慶應の教員の場合のみ、研究分野へのリンクが表示されます。それ以外の場合には、人名のみが表示されます。
- 「研究分野」
 - 研究分野とは、慶應の教員が教員情報の「研究分野」欄に登録した語を指します。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

懸案事項

Maneki Neko Proj

- 以下の懸案事項に関しては、設計・実装フェーズに技術調査を行い、対策を考える予定です。
 - DMCの人名抽出モジュールの精度
 - 研究者名の精度よく抽出するには
 - (人名/研究分野) + 「研究室」というキーワードだと、研究者の名前が出やすいので、これを対策として使うことを考えている。
 - 人名辞書による人名抽出補助を追加で実装する
 - 速度の問題
 - 3段階検索し、1ノードに対して4リンクあるとすると、1+4+16で最大21回の検索をバックグラウンドで行う。
 - 何段階まで検索ができるか、速度を考慮して決定する必要がある。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

プロジェクト定義

Maneki Neko Project

まねきねこ@三田DMC

作業範囲

Maneki Neko Proj

まねきねこ

- 要求分析(10/4~11/8)
 - プロジェクト提案書
- 設計(11/9~11/21)
 - 画面表示仕様書
 - 表示モジュール仕様書
 - 検索モジュール仕様書
 - モジュール間入出力仕様書
 - 外部入出力仕様書
- 実装(11/21~1/10)
 - クラス図
 - 単体テスト仕様書
 - 結合テスト仕様書
 - ユーザテストアンケートの作成
 - 設定・構築マニユアル
 - システム仕様書
- テスト(1/11~1/17)
 - 品質保証テスト仕様書(1/10までに)
 - 品質保証テスト
 - テストを通過したシステム
- 検証、査収(1/18~1/31)
 - 品質向上したシステム
 - 報告書 発表会資料

DMC様にお願すること

- 要求分析(10/4~11/8)
 - プロジェクト提案書の承認
- 設計(11/9~11/21)
 - DMCシステムテスト環境提供
- 実装(11/21~1/10)
 - IndexDBから人名を抽出するモジュールの作成に必要な環境、情報の提供
- テスト(1/11~1/17)
 - 品質保証テスト仕様書の不備チェック
- 検証、査収(1/18~1/31)
 - クライアント検証、フィードバック
 - 査収、結果報告

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

開発範囲

Maneki Neko Proj

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

スケジュール

Maneki Neko Proj

- 要求分析 (10/4~11/8)
 - DMC様による、提案書承認をもって要求分析フェーズを終了する
- 設計 (11/9~11/21)
 - 要求分析フェーズの終了を受け、設計フェーズを開始する。
 - 提案書をもとにシステム仕様書を完成させた時点で、設計フェーズを終了する。
- 実装 (11/22~1/10)
 - 設計フェーズの終了を受け、実装フェーズを開始する。
 - システムが単体テスト・結合テストを通過した時点で、実装フェーズを終了する。
- 品質保証テスト(1/11~1/17)
 - 実装フェーズの終了を受け、テストフェーズを開始する。
- クライアント検証、査収 (1/18~1/31)
 - クライアントからの検証結果報告を受け、プロジェクトの報告書を作成し、報告書作成終了時点でフェーズを終了する。
 - テストユーザに使ってもらい、アンケートで使用感を調査する。

詳しいスケジュールは別紙「開発マイルストーン.xls」を参照してください。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

作業分担

Maneki Neko Proj

	まねきねこ	DMC
プロジェクト提案書	○	△
調査仕様書	○	
モジュール仕様書	○	
DMCテスト環境提供		○
コーディング	○	
単体テスト仕様書	○	
単体テスト	○	
品質保証テスト仕様書	○	△
品質保証テスト	○	
クライアント検証		○
ユーザテスト	○	
査収		○
レポート作成	○	

作業スケジュールは別紙「開発マイルストーン.xls」を参考にしてください。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

DMC様への技術的作業依頼

Maneki Neko Proj

- IndexDB データ抽出モジュールの作成のために必要な資料・環境の提供
 - サンプルプログラム
 - 動作環境 (dcs3になるかは未定)
 - IndexDB にアクセスするための資料
- ※現在交渉中
- テスト環境の構築
 - dcs3 の提供を既に頂きました
- 品質保証テスト仕様書のチェック
 - 漏れがないか嶋津先生のほうでチェックしたいとのことでしたので、テスト前に仕様書を提出しチェックを行っていただきます。
- クライアント検証
 - 嶋津先生に実際にシステムを使っていただき、使用感のフィードバックをいただきます。
- 査収
 - システム、ドキュメント一式を納品致しますので、ご査収ください。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

納品物

Maneki Neko Proj

- ソースコード一式
- プロジェクト提案書
- システム仕様書
- 品質保証テスト仕様書
- 設定構築マニュアル

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

プロジェクト まねきねこ
プロジェクトマネジメント計画書

作成日：2007/11/08

作成者：松尾 壮紘

プロジェクト概要

当プロジェクトは

慶應義塾大学デジタルメディア・コンテンツ統合研究機構(以下 DMC)の嶋津恵子様をクライアントとし、

クライアントが担当している DMC システム構築の手助けとなるアイデアを提案し、それをシステム化するものである。

ステークホルダー

顧客：

DMC 嶋津恵子様

開発メンバー：

松尾壮紘@ドワンゴ

荒木恵@SFC

福田宙音@SFC

松田航@SFC

監修：

大岩研究所 PMO

開発範囲

ユーザが入力したキーワードに対して、

1. 人名
2. 研究分野
3. 1と2に関連する人名

をグラフにして表示する。

これを実装するため、以下の開発範囲を定める。

やること

ユーザのキーワード入力を受け取る

DMC システムに人名を問い合わせる

K-List に研究分野を問い合わせる

問い合わせた結果からリンクデータを生成する。

画面にグラフを表示する

やらないこと

キーワードと人名の関係性が正しいかについては DMC システムの出力に従い本プロジェクトでは関与しない

人名の研究分野については、K-list(またはそれに準ずる物)に従い、内容の正確性については本プロジェクトでは関与しない。

キーワードに関連する人物を DMC システム以外 (Google 等)から抽出する

研究分野以外の関係性を表示する

研究分野が年代ごとにどういう順序で進んでいったのか表示する

成果物

各フェイズにおける成果物は以下の物とする

フェイズ	成果物
要求分析	プロジェクト提案書
設計	画面表示仕様書 システム仕様書 外部入出力仕様書
実装	単体テスト仕様書 結合テスト仕様書 環境設定構築マニュアル βシステム
テスト	クライアントからのフィードバック 市場導入テスト結果 最終システム 最終報告書

特にクライアントから注文のあった、
環境設定構築マニュアルについては、
今後の保守・拡張性のために視認性の高いものが望まれる。

テスト計画

単体テスト

各独立したモジュールごとに単体テストの方法を確立し、独立して動作に問題がないことを確認する

結合テスト

各モジュールを結合し、システムとして正常に動作するかテストパターンを作り動作を確認する。

クライアント検証

βまで組みあがったシステムをクライアントに実際に使ってもらい感想や要求をフィードバックして頂くことを目的とする。そのフィードバックを元にシステムの改善を行う

市場導入テスト

実際にユーザに使われるシナリオを想定し、ユーザを集めシナリオに沿ってシステムを使ってもらい評価をしてもらう。システムを使ってもらったあとアンケートを行い評点を行うことも想定している。

単体テストと結合テストについては、テスト仕様書をクライアントに納め、システム変更を行った時のテストに役立ててもらおう。

コストマネジメント

開発期間を 2007/10/01 ~ 2008/01/31 までとする。

本プロジェクトでは人員コストを時間の合計で算出する。

	月	期間(4ヶ月)
PM	約 60 時間	約 240 時間
学生	約 30 時間	約 120 時間

算出方法は、

PM が出勤日 20 日に対して 3 時間/日

学生が有効日 30 日に対して 1 時間/日 として算出したもので、

プロジェクトに必要な作業時間に基づいた物ではない。

リスクマネジメント

クライアントとの連絡

クライアントが多忙であり、やりとりに時間がかかることが想定される
対応として、メンバーで DMC へ出向き直接交渉を行うことで連絡のやりとり回数を減らす

実装能力

メンバー全員がプログラムを得意ではないと自覚している。
実装フェイズにおいてスケジュール面でオーバーする恐れがある。
解決方法として、松澤さんの協力要請をお願いする可能性を視野に入れている

学習計画

PM 松尾壮紘 ドワンゴ

PM の立場からプロジェクトを経験することで、今後の業務に生かしたい
具体的には

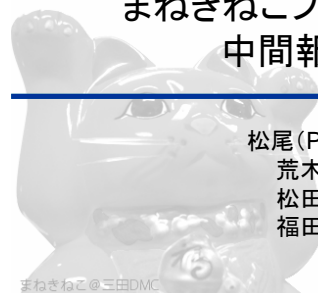
プロジェクト進行の手法(理論) メンバー管理(経験) 資料作成(スキル)
といった面で自身の成長を期待している。

Maneki Neko Project

まねきねこプロジェクト 中間報告

松尾 (PM)
荒木
松田
福田

まねきねこ@三田DMC

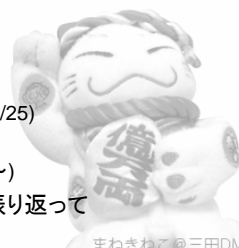


Maneki Neko Proj

目次

- はじめに
- プロジェクトの成果
- システム概要
- プロジェクトの活動
 - 提案型でいこう (10/4~10/15)
 - 要求分析って何? (10/16~10/25)
 - そして提案へ (10/26~11/1)
 - DMC訪問・提案書承認 (11/1~)
- 今までのプロジェクト活動を振り返って

まねきねこ@三田DMC

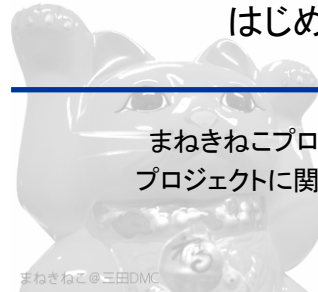


Maneki Neko Project

はじめに

まねきねこプロジェクトとは
プロジェクトに関わるひとびと

まねきねこ@三田DMC

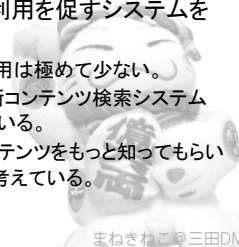


Maneki Neko Proj

まねきねこプロジェクトとは

- プロジェクトの目的
 - 慶應の学術コンテンツの利用を支援する検索システム(DMCシステム)の利用を促すシステムを提案し、開発する。
 - 慶應のコンテンツの学術利用は極めて少ない。
 - DMC機構では、慶應の学術コンテンツ検索システム(DMCシステム)を開発している。
 - DMC機構では、慶應のコンテンツをもっと知ってもらいたい、使ってもらいたい、と考えている。

まねきねこ@三田DMC

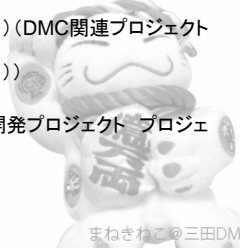


Maneki Neko Proj

プロジェクトにかかわるひとびと

- まねきねこプロジェクトメンバー
 - 松尾 (PM) (株式会社ドワンゴ)
 - 荒木 (政策・メディア研究科1年(4期目) (DMC関連プロジェクト経験者))
 - 松田 (環境情報学部4年(2期目) (DMC関連プロジェクト経験者))
 - 福田 (総合政策学部2年(1期目))
- クライアント
 - 嶋津恵子先生 (DMCシステム開発プロジェクト プロジェクトリーダー)
- 支援
 - コラマネオフィス

まねきねこ@三田DMC



Maneki Neko Project

プロジェクトの成果

まねきねこ@三田DMC



プロジェクトの成果(～11/8)

Maneki Neko Proj

- 「検索ワードを入力すると、その分野に関連のある人物の一覧および、その人物の関連研究分野に関係のある人物の一覧を表示するシステム」を作ることが決まった。
 - クライアントの要望を受け、上記システムを提案し、大枠で承認をいただいた。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システム概要

(提案書から抜粋)

まねきねこ@三田DMC

システムの目的・前提(1/3)

Maneki Neko Proj

- 特定の分野に関連する研究者を知りたい人が、簡単に研究者を検索できるようにする。
 - 共同研究者を探している経営者/研究者
 - 所属研究室を決めたい学生

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システムの目的・前提(2/3)

Maneki Neko Proj

- いままでの×
 - 多くの検索ステップ数
 - ユーザの高度な検索スキルが必要
- これからの○
 - 検索ステップが1回に
 - ユーザの検索スキルは必要ない

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システムの目的・前提(3/3)

Maneki Neko Proj

- 対象
 - 特定の研究分野に関連する人物について知りたい人
 - 共同研究者を探している経営者/研究者
 - 所属研究室を決めたい学生
- 現状
 - 研究分野と人のつながりを知りたい場合、キーワード検索→ページ閲覧→新たなキーワードの抽出→キーワード検索...というステップを何度も繰り返さなければならない
- 目的
 - 本システムでは、研究分野をキーワードとして入力すると、関連する人と関連研究をグラフとして表示することによって、研究分野と人のつながりを知るために必要な検索ステップ数を減少させる。
- 前提
 - 特定のページから人名のみを抽出し、その数を数えることができるモジュールがDMCシステム内にあることを前提とする。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システム概要(1/3)

Maneki Neko Proj

- 任意の検索ワードを入力すると、検索ワードを中心とし、人と研究分野をノードとするグラフを表示する。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システム概要(2/3)

Maneki Neko Proj

- 検索ワードから、関連研究者と関連分野を探すシステム。

The diagram illustrates the system's search process. It starts with a '検索ワード' (Search Word) input, which goes into a '検索エンジン' (Search Engine). The engine outputs a '人名のリスト' (List of Names) and a '研究分野のリスト' (List of Research Fields). These lists are then processed by a 'DMC' (Digital Media Center) and a '推薦システム' (Recommendation System) to generate a 'グラフ' (Graph) and a 'リンク' (Link). The graph and link are then used to find related researchers and fields.

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システム概要(3/3)

Maneki Neko Proj

- 検索ステップ1
 - 検索ワード「プログラム」で検索。
 - 「大岩・萩野...」と人名のリストが出る
- 検索ステップ2
 - 検索ワード「大岩」で教員検索
 - 専門分野が「認知工学・情報教育・ソフトウェア工学」と出る
- 検索ステップ3
 - 検索ワード「認知工学」で検索
 - 「川島・大門」と人名のリストが出る。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

プロジェクトの活動

Maneki Neko Project

- 提案型でいこう(10/4~10/15)
- 要求分析って何?(10/16~10/25)
- そして提案へ(10/26~11/1)
- DMC訪問・提案書承認(11/1~)

まねきねこ@三田DMC

提案型でいこう

Maneki Neko Project

- (10/4~10/15)
- 漠然とした要求調査
- アイデア出しと選定

まねきねこ@三田DMC

漠然とした要求

Maneki Neko Proj

- 鳴津先生から要求を聞いたが漠然としていた。
- ↓
- 鳴津先生から提案して欲しいとの要求があったし、質問し続けても迷惑だと考えた。
- ↓
- 結果、チームからシステムを提案することにした。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

調査

Maneki Neko Proj

- DMCとクライアントが何をしていた、何をこちらに求めているかがわからなかったため、クライアントに関する資料を読みました。
- 『利用者のコンテキストに合ったコンテンツ検索・編纂システム』
- 『学内のデジタルコンテンツの公開と「知の編纂と再編」のチャレンジ』

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

アイデア出しと選定

- どの方針が正しいか探るため、18案出した。
- 同じものを選んでもしかたがないため、カテゴリ分けして、違う性質の3案を選んだ。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

要求分析って何？

(10/16~10/25)
要求分析って何？
目的を決めよう
クライアントとの連絡

まねきねこ@三田DMC

要求分析って何？

- 何のためにシステムを開発するのかを考えていなかった
 - 分析・調査はそれぞれ別々に行っていた。
 - →各自イメージする「目的」がズレていた

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

目的を決めよう

- チーム内でシステムの「目的」が決まっていなかったため、目的を絞った。
 - 「慶應義塾に蓄積されている研究成果や学術成果を、学外にも広く開放し、専門家だけでなく一般の方々にも自由に利用できる仕組み」を支援するという目的に絞った。
(※クライアントのインタビュー記事より抜粋)

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

クライアントとの連絡

- クライアントが非常に多忙で連絡が取れないことがあったため、対策を練った。
 - 連絡窓口を決める。
 - クライアント側の担当者を変更して頂くことも視野に入れる。
 - 返事を待っているのではなくこちらから連絡をする。
 - 話しが出来るときにできるだけ多くのことを話す。
- 結論が先になく、メールの仕方が下手であるとのこと指摘を頂いた。
 - メールをチェックリストを作成することにした。

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

そして提案へ

(10/26~11/1)
3つのプレゼン
良い点を生かそう
-目的に照らし合わせて-
企画提案

まねきねこ@三田DMC

3つのプレゼン

Maneki Neko Proj

- 3案をそれぞれクライアントの前でプレゼンしたら、どれも甲乙付けがたくクライアント側で絞れないという指摘を頂いた。
- 実現性が見えないため、松尾PMが選び、きちんとした提案の形にして頂くようにご指摘をいただいた。
 - SearchStoryTeller
 - ユーザが複数コンテンツ間の関連を示すタグをつけられるようにするシステム
 - In Relation
 - 言葉とそれに関連する言葉の関係を樹形図で示すシステム
 - celebrity search (人物検索)
 - 検索ワードに関連する人物と、その人物に関連するコンテンツを表示するシステム

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

良い点を生かそう

Maneki Neko Proj

- 「人の関連を図で表示するシステム」にすることを、PMが決定。
 - 「研究者を検索できる」という実用性 (CelebritySearch)
 - 「関連図表示」というインターフェース (InRelation)

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

企画提案

Maneki Neko Proj

- 目的と実現性を考えて提案を詰めた
 - 「慶應のコンテンツの学術利用促進」に役立つか
 - DMCシステムを使って実際に(人力で)検索を行い、実現できるかを調査した
- 「検索ワードを入力すると、その分野に関連のある人物の一覧および、その人物の関連研究分野に関係のある人物の一覧を表示するシステム」を提案した。
 - 人: 慶應の教員・研究者
 - 研究分野: 慶應の教員が教員検索システムに自ら登録している研究分野

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

DMC訪問・提案書承認

Maneki Neko Project

(11/1～現在)

DMC訪問・提案書へのレビュー
システム実現のために

まねきねこ@三田DMC

DMC訪問・提案書へのレビュー(11/1)

Maneki Neko Proj

- 11月1日にDMC(三田)を訪問し、提案書を提出した。
- 「ここを修正すれば承認する」という修正点を指摘していただいた。
 - DMCの担当範囲(負担)を明示すること
 - 用語定義をきちんと書くこと

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

システム実現のために


Maneki Neko Proj

- 技術調査
 - XMLパーサの作成
 - DMC、教員システムのデータをXMLでやりとりするので、そこから必要なデータを獲得できるようにする
 - グラフ出力用ライブラリ調査
 - 渡された人物と研究分野のデータを関連グラフとして出力できるようにする

まねきねこ@三田DMC
Maneki Neko Project 2007 Fall

Maneki Neko Project

学んだこと・考えたこと




まねきねこ@三田DMC

Maneki Neko Proj

それぞれが学んだこと・考えたこと

- 荒木
 - (当日昼まで)
- 松田
 - (当日昼まで)
- 福田
 - (当日昼まで)



まねきねこ@三田DMC

Maneki Neko Project 2007 Fall

2.3 設計

2.3.1 技術調査合宿

まねきねこで開発する検索システムは、以下のような仕組みで動作することを想定していた。

- (1) 検索語から、DMC の人名検索 API を用いて人名リストを取得する
- (2) 慶応の教員リストから、人名検索を行い、その教員の研究分野を取得する
- (3) 取得した研究分野を検索語とし、DMC の人名検索 API から人名リストを取得する
- (4) 取得した人名と研究分野をグラフ表示する

この仕組みを実現するためには、

- DMC の人名検索 API について
- 慶応の教員リスト検索システムについて
- XML パーサの仕組み (検索結果の取得および解析のため)
- サーバサイド Java の仕組み (システム本体の動作のため)
- Java アプレット (グラフ表示のため)

についての知識が必要であった。

XML やサーバレット、慶応の教員リスト、DMC の検索 API などのはじめて触れる技術を使う必要があったため、開発メンバーである荒木と松田で技術調査合宿を行った。荒木がサーバサイド Java を担当、松田が XML の技術調査を担当した。

技術調査合宿で行ったことは以下の通りである。

- 開発環境整備
- 慶応の教員リスト検索システム (K-Ris) の調査
- DMC システムの検索 API (人名抽出) の調査
- XML パーサの使い方
- サーバサイド Java
- YahooAPI の調査
- 各学部の教員リスト (HTML) から教員名一分野対応データ (XML) を作成するプログラムの作成

技術調査を行った結果、以下のことが分かった。

- 慶応の教員リスト検索システムは、情報量の少なさと遅さから、使えない
- 教員リストは、各学部のホームページから情報を抽出し、XML 化して使うことで、慶応の教員リスト検索システムより情報量を多くできそうである。
- DMC の人名抽出 API は精度が良くない
- Java 標準の XML パーサは 2 種類ある
- YahooAPI から、検索結果のインデックスを取得し、解析することで、人名を抽出でき

る可能性がある

- サーバサイド Java は、JSP と JavaBeans、サーブレットの連携で造ることができる
これらを踏まえて、PM に相談したところ、とりあえず、検索システムの実装の前に、全体設計をし、担当範囲を決める、ということが決まった。

2.3.2 入出力仕様（プロトコル）の決定

モジュール間入出力仕様書、外部入出力仕様書

モジュール内外の入出力仕様を決定した全体設計では、そこに至るまでの紆余曲折があった。最終的に PM によって作成・整理された入出力仕様に関しては、下記「モジュール間入出力仕様書」と「外部入出力仕様書」を見ていただきたい。ただしこの仕様が作成された段階では既にメンバーは自分たちで策定した仕様に従って実装を進めてしまったため、実際には PM が作成した仕様書は利用されなかった。

【モジュール間入出力仕様書】

・ 研究分野→人 検索モジュール

テーマや研究分野から人物を得るモジュールである。テーマなどを DMC システムで検索し、自然言語解析で得ることのできた人物名を DMC システムから得た XML データから人物名を抽出する。

<入力>

String 型文字列（研究分野、テーマなど）

<出力>

String[][]型文字列配列（人物名、研究分野、人物名）

ArrayList 型可変長配列

・ 人→研究分野 検索モジュール

人物名から研究分野を得るモジュールである。人物名を k-list に問い合わせ、そこで得た研究分野を用いる。k-list から得た XML データから研究分野を抽出する。なお、k-list に名前が無い場合は null が返却される。

<入力>

String 型文字列（人物名）

<出力>

String[][]型文字列配列（人物名、研究分野、人物名）

ArrayList 型可変長配列

・ リンクデータ作成モジュール

研究分野→人 検索モジュール、人→研究分野 検索モジュールで得ることができ

た関連データをグラフに表示するのに適したデータに変換するモジュールである。

<入力>

String[][]型文字列配列 (人物名、研究分野、人物名)

ArrayList 型可変長配列

<出力>

HTML

・ グラフ表示モジュール

リンクデータから関連グラフを出力するモジュールである。

<入力>

リンクデータ

<出力>

関連グラフ

【外部入出力仕様書】

・ DMC システム

テーマや研究分野から人物名を得るのに使用するシステムである。テーマや研究分野を検索した際に得ることができる XML データを、人物名を抽出するのに使用する。

<入力>

String 型文字列 (人物名)

<出力>

XML データ

・ k-list

慶応義塾大学の教員検索システムである。人物名を検索した際に得ることができる XML データを、専門分野を抽出するのに使用する。

<入力>

String 型文字列 (専門分野)

<出力>

XML データ

入出力仕様決定までの顛末

ここでは、上記仕様を PM が策定するまでにメンバー間で試行錯誤した過程を記す。まず、技術調査の後、全体設計を行うことになった。はじめ、私たちは全体設計とは何をしたいのか分からなかったため、システム全体図をもとに、このシステムを「検索部」と「表示部」に分けて、それぞれ検索部は松田が、表示部は荒木が担当することに決め、実装に入ろうとしていた。

この図は荒木が企画の段階で作成したもので、人マークはユーザを示し、紙マークがデータを、黒矢印がデータの流れを、太枠が Hanabi システム範囲を示す。曲矢印は、外部システムを使う、ということを示す。

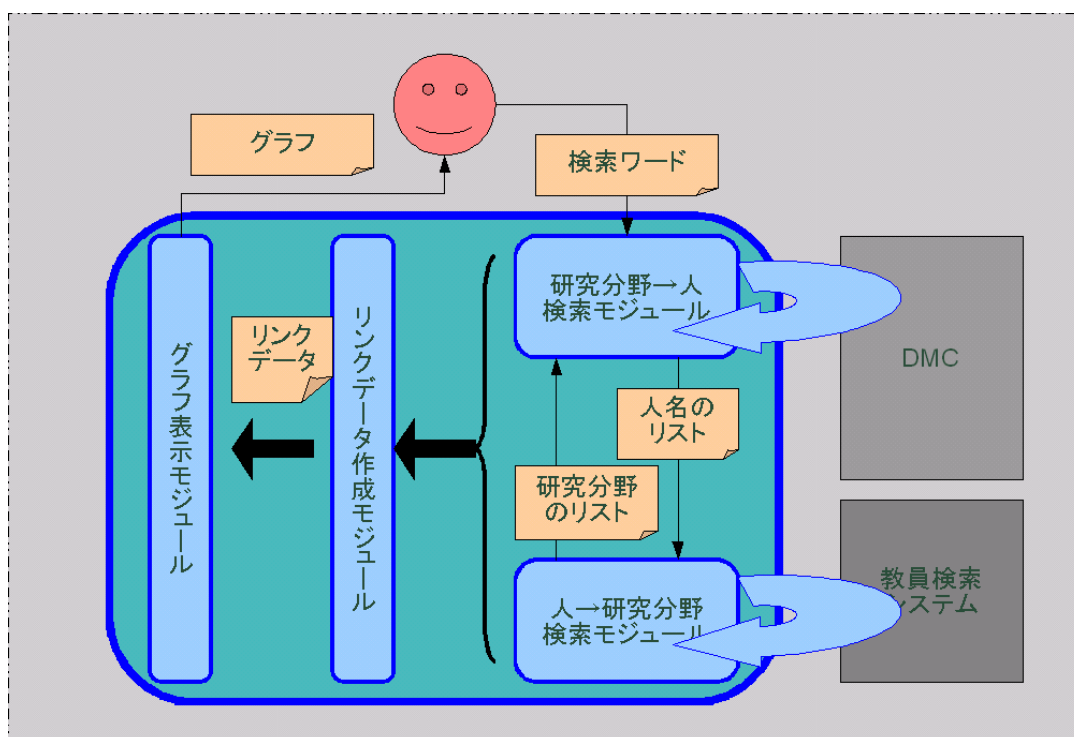


図 2-1 「全体設計図」

しかし、PMO の松澤さんから、「サーバ-クライアント間のプロトコルを決める」ことが必要だ、というレビューをいただいた。この時点では、松田と荒木は本システムを「サーバ-クライアント」という見方よりも、「モジュールの集合」という見方で見ており、実際、役割分担もモジュールごとになっていたため、これをモジュール間のプロトコルと呼び、このうち、検索部の出力データ仕様（リンクデータ）を以下のように決定した。

「大岩-情報教育,大岩-認知科学,大岩-コンピュータ,情報教育-斉藤…」

これは、語間の関連をハイフンで表した文字列である。この仕様で決定した理由は、今回使用するグラフデータ表示モジュール「GraphLayout」が、引数として取る値がこのような文字列だったためである。

このプロトコルを松澤さんに見ていただいたところ、「これでは汎用性がない」というレビューをいただいた。

この時点では、汎用性、とはどのようなことをさすのかが分からなかったが、とりあえず、レビューにしたがって、モジュール間のプロトコルを変更し、LinkData というクラスにした。

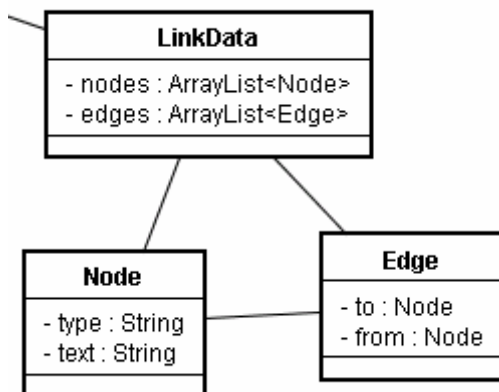


図 2-2 「プロトコル出力仕様」

そして、このプロトコルをもとに、全体を設計しなおした。

この図は、角丸四角形が Hanabi システムを指し、左下にある「XML 作成モジュール」は、Hanabi システムで使う教員リスト (XML) を、慶応の教員一覧ページ (HTML) から生成するために作成したプログラムである。「検索」部分と「リンクデータ」「表示」部分の色が違うのは、松田と荒木の担当範囲を示している。

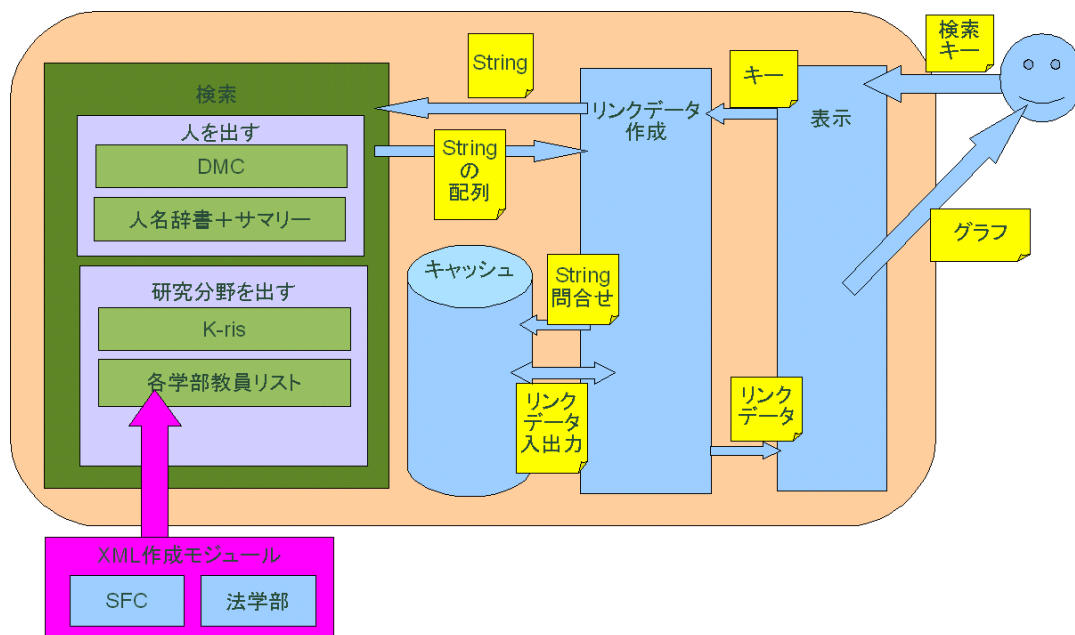


図 2-3 「修正後の全体設計」

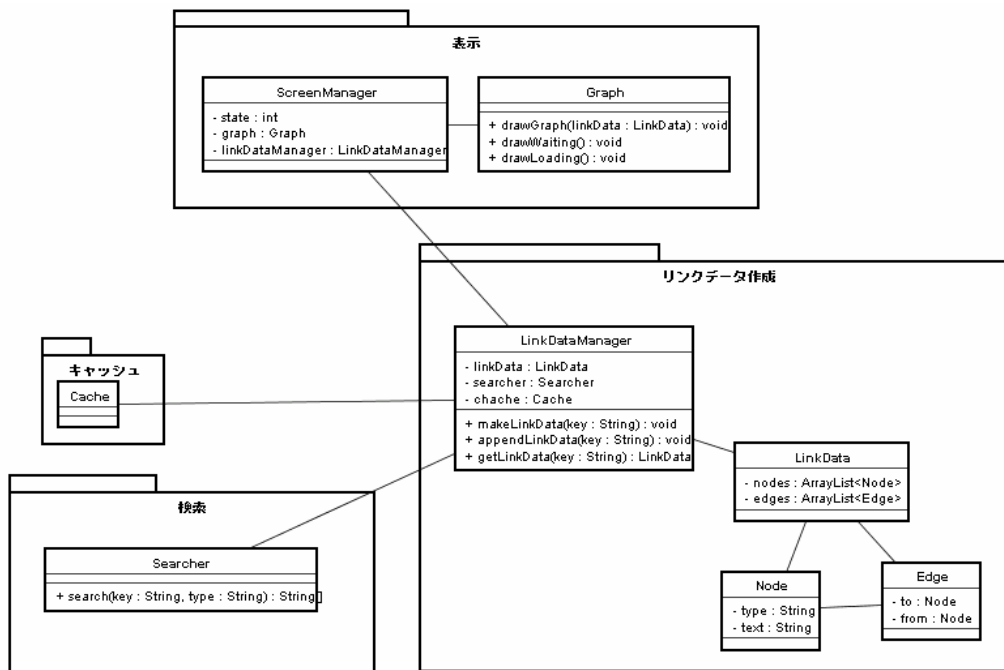


図 2-4 「全体クラス図」

この全体設計をもとに、検索部分を松田が担当し、それ以外を荒木が担当することになった。検索部分の呼び出しの仕様は、以下のように決定した。

この図では、外部システムとのやり取りが矢印であらわされている。円筒形は外部システムを指し、ファイルアイコンは、事前に用意しておくデータファイルをさす。この時点では、「検索語からそれに関連する人名を出す」部分の実装方法として、DMC システムのインデックス DB を使う方法と、Yahoo 検索システムの検索結果ページのサマリーから人名を抽出する方法のいずれを採用するかが決まっていなかった。「人名から研究分野を出す」部分に関しては、K-ris(慶応大学研究者検索データベース)ではなく、あらかじめ用意した教員リストファイルを使うことに決定していた。

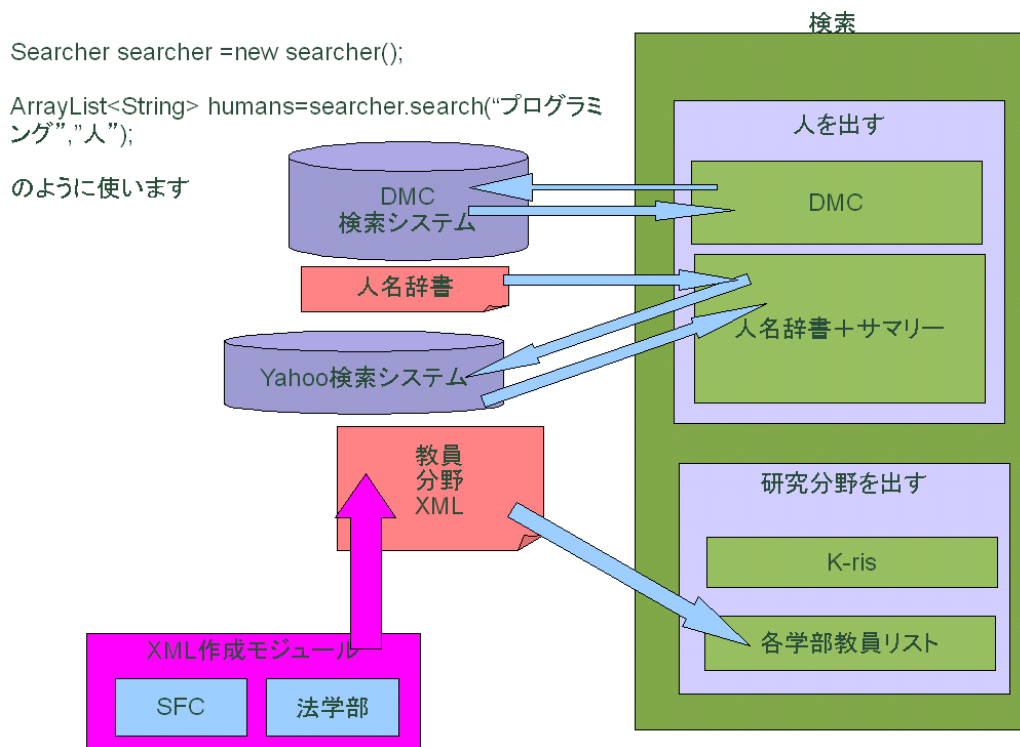


図 2-5 「検索部の呼び出し仕様」

2.3.3 それぞれの設計—検索部編

検索部の設計は大きく二つに分けることができる。

設計① : 11/15~11/27

この時点での設計では、検索部はキーワードに対してそれに関係する人物リストとその専門分野の情報を返すという仕様になっていた。設計で重視したことは再起処理を利用して手間を掛けずに実装できるかを重視した。その結果 11/22 までにそれを実現したプログラムを作ることができた。しかし、11/26~11/27 に於ける荒木宅での合宿では教員検索システムが使い物にならにことが発覚したため、専門分野抽出に必要な XML の教員リストを検索部が持つ仕様になり、それによって仕様も教員検索システムに問い合わせるのではなく、XML の教員リストを読む設計に変更になった。

この時点での設計は、設計フェーズの資料の「モジュール間入出力仕様書」と「外部入出力仕様書」を参考にしてほしい。

設計② : 11/28~1/17

チームミーティングによって全体の設計を変更することとなった。検索部はキーワードに対しては人名を、人名に対しては専門分野を、専門分野に対しては人名を返すという設計に変更になった。他の変更はクライアントの都合により、人名を抽出するモ

ジュールを作成することとなったことである。全体の設計がより明確になったことで、各モジュールの設計もどのような役割を持たせることにするかも明確になった。

2.3.4 個々の設計ー表示部・サーバサイド Java・リンクデータ作成部編

表示部の設計の解説

表示部には、Sun が提供している既存のグラフ表示プログラムである「GraphLayout」(<http://java.sun.com/applets/jdk/1.4/demo/applets/GraphLayout/example1.html>) を使用したため、まず、どのようにして動いているのかを解説する必要がある。

この解説はさほど難しくなかった。グラフ表示プログラムは、GraphPanel という、実際の描画を担当するクラスと、パラメータの読み込みを行う Graph というクラス、Node クラスと Edge クラスに分かれていた。

今回のシステムでは、語の種類によって色分けする必要があったため、Node クラスに Type 属性を追加し、Type によって色を分けるように修正した。

サーバサイド Java

技術調査合宿で、サーバサイド Java にはサーブレットと JSP、JavaBeans があることが分かった。しかし、これらをそれぞれ、どのように使っているのか分からなかったため、他プロジェクトの PL である熊谷さんに相談し、以下のように設計を行った。

HanabiMainServlet

POST のデータを受け取り、リンクデータを作成し、JSP ページへの遷移を担当する

Hanabi.jsp

HanabiMain からリンクデータを受け取り、グラフページを表示する

Error.jsp

HanabiMain でエラーが起きた場合、エラーメッセージを受け取りエラーを表示する
そのほかの機能(LinkDataManager や Searcher)などは JavaBean として、HanabiMain から読み出すことにした。

リンクデータ作成部

リンクデータ作成部は、全体設計で作成したとおりの仕様にした。

2.3.5 設計フェーズの総括

設計フェーズで全体設計を行ったことにより、検索部とその他の部分の結合は、問題なく行うことができた。また、リンクデータを、GraphLayout に特化した文字列でなく、クラスにしたことで、データ仕様の変更や、検索アルゴリズムの作成、出力形式の変更などに柔軟に対応できるようになった。

とくに、検索部とサーバサイド Java は、何度か書き換えを行ったにも関わらず、入出力

仕様がはっきりしていたため、書き換えによって結合に問題が起きることはなかった。

設計文書などはほとんど残さず、最低限の仕様を決めただけであったが、設計をしなかったら、結合で大きな問題が起きていたと思われる。

設計フェーズで残念だった点は、個人作業に移ってから、設計をほとんど文書に残さなかったため、最終的なシステムの全体設計を示す文書が残らなかったことである。全体的に分かりやすいソースにすることを心がけてはいるが、今後プログラムを変更したりするためには全体設計を示す文書があったほうが便利である。

今後のことを考えて、システム作成後に全体設計と仕様を示す文書を作成した。「資料」の節に添付してある、「Hanabi システム全体設計」を参照していただきたい。

2.3.6 画面表示仕様書

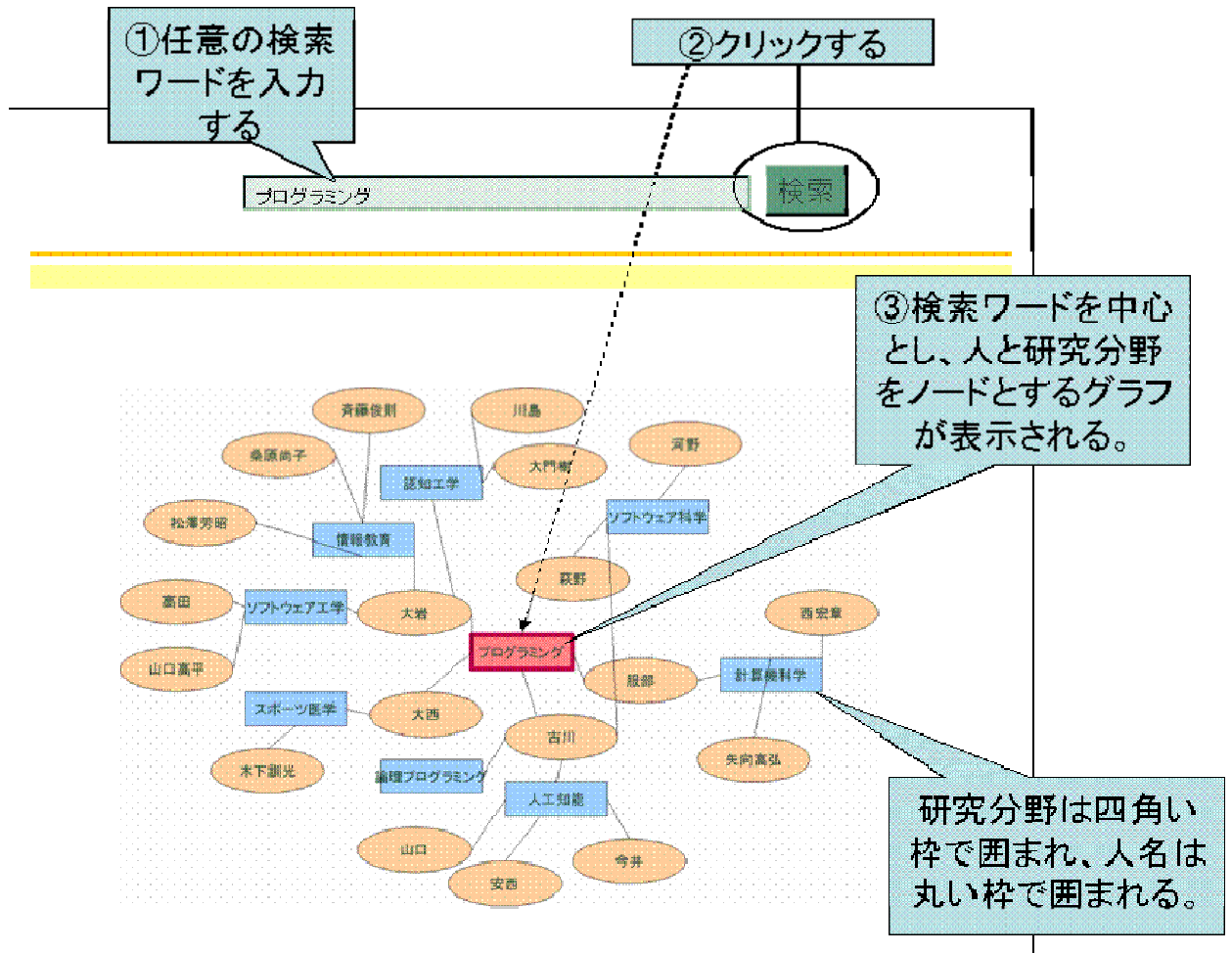


図 2-6 「画面表示仕様書」

2.3.7 Hanabi システム全体設計

Hanavi システム全体設計

1. 全体設計

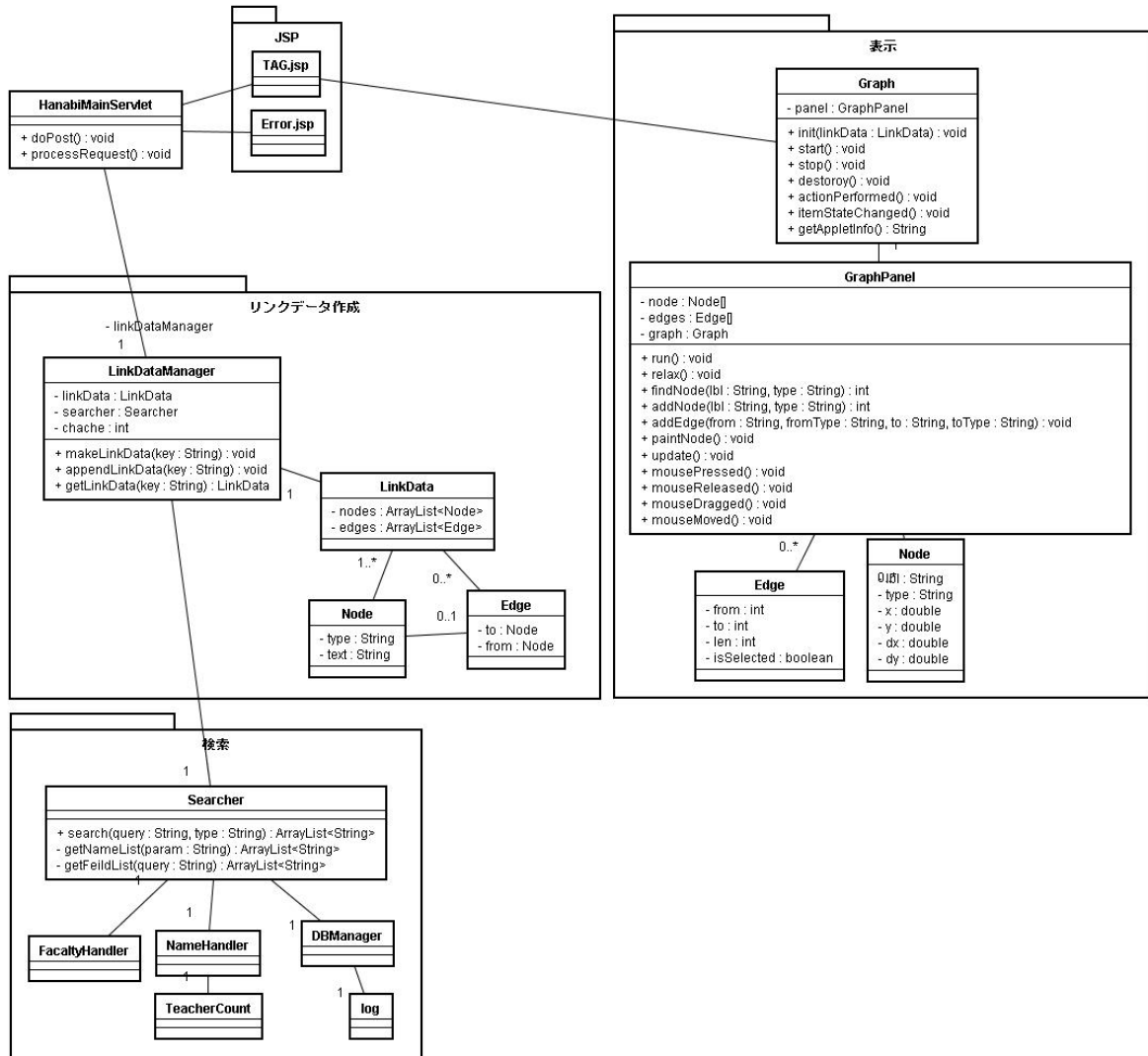


図 2-7 「全体設計」

2. システム全体の挙動を示す HCP チャート

Hanavi グラフを出力する

検索語の入力を受け付ける

リンクデータを作成する

研究分野から関連する人名リストを検索する

人名-キーワード DB に問い合わせを行う

(キャッシュがあったら)

人名リストを返す

(キャッシュがなかったら)

Web 検索エンジンに問い合わせを行う

検索結果から人名を抽出する

人名リストを返す

人名から関連する研究分野リストを検索する

教員リストを検索する

研究分野リストを返す

グラフを出力する

3. 詳細仕様

WEB-INF/src 内は 4 つのパッケージに分かれている。

defaultpackage : メイン部・表示部

searcher : 検索関連

linkdatamanager : リンクデータ作成

hanabisetting : 設定部

そのほか、Hanabi ディレクトリの直下に、以下のファイルがある

HTML

index.html (フレーム表示 HTML)

SearchTop.html (はじめに表示される左フレーム)

JSP

TAG.jsp (Hanabi グラフ表示 JSP)

Error.jsp (エラー表示 JSP)

APPLET (グラフ表示アプレット)

Graph.class

GraphPanel.class

Edge.class

Node.class

hanabisetting/HanabiSetting.class

(ア)メイン部 (デフォルトパッケージ)

① サーブレット部

HanabiMainServlet

POST でデータを受け取り、文字のチェックを行い、リンクデータを作成し、作成したリンクデータを文字列に変換したものを Hanabi ディレクトリ直下の TAG.jsp に渡す。リンクデータが作成できなかった場合、文字にエラーがあった場合は Hanabi ディレクトリ直下の Error.jsp にエラーメッセージを渡す。

② アプレット部

Graph

GraphPanel

Node

Edge

グラフを描画する。

(イ)検索部

Searcher

サーチキー（文字列）とそのタイプ（キーワード、人名、専門分野）の組み合わせを受け取って、サーチキーに対するそれぞれのタイプに関連する情報を返す。

FacultyHandler

教員リストを参照し、教員の専門分野を Searcher に渡す。

NameHandler

人名辞書を参照し、検索ページのサマリから人名をみつけ、それを Searcher に渡す。

DBManager

データベースを操作する。一度取得した人名リストを保存・参照を行う。

教員リスト

教員名とその専門分野が XML 形式で登録されているファイル。

人名辞書

人名が記載されているテキストファイル。

① 検索部の挙動を示す HCP チャート

searcher へのサーチキー・タイプの入力を待つ

サーチキーに対する情報を抽出する

タイプがキーの場合：キーワードに関する人名を返す

キーワードに対する人名がデータベースにあるか参照する

ある場合：データベースを参照し、人名を抽出する

無い場合：検索ページのサマリから人名辞書を参照し、人名を抽出する

タイプが研究分野の場合：人名に関する研究分野を返す

教員リストを参照し、研究分野を抽出する

タイプが人名の場合：研究分野に関する人名を返す

キーワードに対する人名がデータベースにあるか参照する

ある場合：データベースを参照し、人名を抽出する

ない場合：検索ページのサマリから人名辞書を参照し、人名を抽出する

検索ページのサマリから人名辞書を参照し、人名を抽出する

サーチキー・タイプに従った情報を返す

(ウ)設定部

HanabiSetting

Hanabi システム全体の設定ファイル。
デフォルトの設定は以下の通りである

```
/**
 * 検索システム初期設定
 */
// 検索用データファイルのあるディレクトリの絶対パス
// public static final String DATA_DIR = "C:\¥¥Documents and
// Settings¥¥manekineko¥¥workspace¥¥Hanabi¥¥hanabisetting¥¥";
public static final String DATA_DIR = "";

// 検索結果の種別
public static final String TYPE_KEY = "key";
public static final String TYPE_HUMAN = "name";
public static final String TYPE_FIELD = "field";

/**
 * 検索システム動作設定
 */
// 検索する順番と段階数
// (最後の1つはダミーデータで、検索を行わない。SEARCH_ORDER.length-1回検索する)
public static final String[] SEARCH_ORDER = {TYPE_KEY, TYPE_HUMAN,
        TYPE_FIELD, TYPE_HUMAN, TYPE_FIELD};
public static final int SEARCH_NUMBER_OF_TIMES = 4;// 1回の検索で返ってくる結果数
public static final int SEARCH_WORD_MAX_LENGTH = 20;// 検索語の最大文字数

/**
 * 検索 URL 設定 (Yahoo 検索用) 実際の検索 URL は
 * http://api.search.yahoo.co.jp/WebSearchService/V1/webSearch?appid=awl_247&query= 【 検 索 語
 * &results=50&site=keio.ac.jp
 * となる
 */
public static final String YAHOO_API = "http://api.search.yahoo.co.jp/";
public static final String WEB_SEARCH = "WebSearchService/V1/webSearch";
```

```
public static final String INIT_APPID = "?appid=";
public static final String YAHOO_ID = "awl_247";
public static final String INIT_QUERY = "&query=";
public static final String INIT_RESULTS_AMOUNT = "&results=";
public static final int RESULTS = 50;
public static final String INIT_DOMAIN = "&site=";
public static final String DOMAIN = ".keio.ac.jp";
public static final String REVISION_WORD = "研究室";

/**
 * グラフ表示用設定
 */
// 背景色
public static final Color GRAPH_BGCOLOR = Color.black;

// ノードの色
public static final Color KEY_NODE_COLOR = new Color(255, 204, 102);
public static final Color HUMAN_NODE_COLOR = new Color(102, 255, 153);
public static final Color FIELD_NODE_COLOR = new Color(102, 255, 255);

// クリックして選択されたノードの色
public static final Color SELECTED_NODE_COLOR = Color.pink;

// マウスが乗っているノードの色
public static final Color MOUSE_ON_NODE_COLOR = Color.red;

// 関係線の色
public static final Color BASIC_EDGE_COLOR = Color.white;
public static final Color SHORT_EDGE_COLOR = Color.white;
public static final Color MEDIUM_EDGE_COLOR = Color.white;
public static final Color LONG_EDGE_COLOR = Color.white;

// 関連強度表示文字の色
public static final Color STRESS_COLOR = Color.white;

/**
```

```
* ノードクリック検索用設定 実際の URL は
* http://search.yahoo.co.jp/search?p=【URL エンコード済み検索語】&ei=UTF-8&vs=keio.ac.jp
* となる
*/

public static final String NODE_CLICK_SEARCH_URL = "http://search.yahoo.co.jp/search?p=";
public static final String NODE_CLICK_SEARCH_ENCODING = "&ei=UTF-8";
public static final String NODE_CLICK_SEARCH_DOMAIN = "&vs=keio.ac.jp";

/**
 * デバッグプリント
 */
public static final boolean DEBUG = false;
public static void debugPrint(String text) {
    if (DEBUG) {
        System.out.println(text);
    }
}
```

4. ソースコードに変更を加える際の注意点

設定ファイルもしくは表示部 (デフォルトパッケージの `Graph, GraphPanel, Edge, Node`) に変更を加えた場合、**Hanabi** ディレクトリ直下にアプレット関係の `class` ファイルを配置するために、コマンドラインから `WEB-INF/src/GraphModuleMaker` バッチファイルを実行すること。

2.4 実装

2.4.1 検索部の実装

検索モジュールの実装は六段階に分かれていた。

第一段階：11/15~11/22

どのようなものを作るかが決定したが、メンバーの一人が12月からの活動が難しいことが判明し、各モジュールの入出力を決定することで設計を済ませ、実装を前倒しで開始した。この時点での検索モジュールの仕様は、人物を抽出するのにDMCシステムの持つ自然言語解析による人名抽出機能、専門分野を抽出するのに慶應の教員検索システムを使用することでグラフを表示するのに必要なデータを作成するというものであった。DMCのWEB APIが未完成、教員検索システムの調査をしていなかったため、仮のデータで動作確認をしたところ想定通りの動作をした。

技術調査：11/25

11/22のPMとクライアントの電話での話し合いで、DMCシステムの名抽出機能は使えない物にならないとクライアントから指摘され、DMCシステムのINDEX DBに直接問い合わせるという方式に変更になったことを受け、検索にどれほどの時間が掛かるかの調査を行った。Yahoo APIを使い、実際に必要な情報を得るのに掛かる時間を計測するプログラムを作成し、時間を計測したところ、必要な情報を得るのに10秒程度かかることがわかった。

第二段階：11/26~11/27

メンバー荒木宅にて合宿を行った。この合宿はグラフ表示モジュール、検索モジュールを連動させることで想定通りのグラフが表示されるかを確認する技術調査である。また11/22のクライアントとの電話での話し合いの中で、検索モジュールに人名辞書を持たせ検索結果のサマリーから人物を抽出するという機能の提案もしたので、同時進行で調査おした。未調査だった教員検索システムを調査したときは驚愕した。学部毎に仕様が異なり、人によって専門分野の区切りの文字がバラバラであったりするので、ここから自動的に人物から専門分野を抽出するのは不可能だということが判明し、教員検索システムを使わずXML形式の教員リストを読み込むことで専門分野を取得できるような仕様となった。他にもサマリーから人名を抽出するのにかかる時間が膨大だったことやXMLの扱い慣れていなかったために、必要なデータを作成することが出来ず、この合宿ではグラフを表示させるという目的は果たせなかった。

第三段階：11/29~12/12

チームミーティングで仕様が変更になる。今までの検索モジュールでは、キーワードに対してそれに関係する人物リストとその専門分野を付加したデータをグラフ表示モジュールに渡していたが、キーワードを渡されたら人物の配列、人物を渡されたら専門分野の配列、専門分野を渡されたら人物の配列を返すといった仕様に変更となった。この間で XML パーサの勉強をしたため、XML から情報を取得できるようになった。これによって XML 形式の教員リストから、教員の専門分野を取得できるようになった。あとは DMC の INDEX DB さえ動けば稼働可能な状態となった。

第四段階：12/13~12/20

DMC の INDEX DB の仕様がクライアントと話をする度に変更になり、これ以上話し合いをすると余計な混乱が生じてしまうので、INDEX DB を使用しないことで合意をし、Yahoo API と人名辞書を使った人名抽出機能を追加することにした。実装第二段階での調査では人名抽出に 30 分程度掛かっていたが、1~3 秒にまで縮めることに成功した。また人物抽出の精度も上昇した。

第五段階：12/20~1/16

機能に変更はないが、設計が変更になった。他にもバグフィックスなどが行われた。今まで一つのクラスで全てを処理していたが、XML パーサのハンドラーをクラス分けし、メンテナンスが容易となった。インターフェース、グラフ表示モジュールとの結合も行き、設定ファイルを共有した。また、ファイルのエンコードや、辞書ファイル、教員リストの確認、ネットワークのタイムアウト処理などが加わった。

第六段階：1/17~

松澤さんのソースコードレビューを受けたことで若干の変更を加えた。またレビューの際に受けたアドバイスにより、前回関わった Geocoding Module の データベース管理システムを本システムに移植することにした。そうすることでキーワードや、専門分野から人物リストを抽出する際に Yahoo API に問い合わせる必要がないので、より高速グラフを表示することができる。

2.4.2 表示部・サーバサイド Java・リンクデータ作成・設定ファイルの実装 表示部の実装

表示部は、既存の GraphLayout プログラムを使って行ったため、楽に作ることができた。一部、MouseEvent や、Panel の仕様などが分からない部分もあったが、ちょうど他のプロジェクトで、MouseEvent、KeyEvent を受け取り、処理を行うプログラムの作成をしていたため、同時に勉強することができた。

後に、グラフモジュールのノードが重なって見にくい、ということが分かってから、マウスオーバーで関連したノードの色を変える、というサンプルを作ってみたが、このサンプルを作るのにもあまり時間はかからなかった。

サーバサイド Java の実装

まず、環境構築方法もよくわかっていなかったため、Tomcat のインストール、サーバサイド Java 開発環境のインストールに時間がかかった。

サーバサイド Java ははじめて使う技術だったため、まず技術調査が必要であった。はじめは、サーブレットと JSP、JavaBeans の違いを理解するのに時間がかかったが、本や Web の解説などを読み、概要がつかめたところで実装をしてみると、わりとすんなりと実装を進めることができた。はじめの学習に時間がかかったが、実装する機能が極めてシンプルであることもあり、実装自体には時間はかからなかった。

リンクデータ作成部の実装

リンクデータ作成部は、技術調査合宿の時点で松田くんに「このようなリンクデータ作成メソッドがあるとよいと思う」ということを示すためのサンプルとして提示したものをほぼそのまま使った。このため、後のコードレビューで、「再帰を使っている分りにくい」「読みにくい」といわれる原因になった。

設定ファイルの実装

実装がほぼ完了した時点で、設定ファイルを作ることにした。これは、実際に動作を確認する段階になると、やはり 1 つずつソースコードを書き換えるのは面倒だ、と感じたためである。設定ファイルの実装は、独断ではじめたため、2 回ほど松田くんの作業とコンフリクトを起こし、コンフリクトの解消に若干時間を取られた。

2.4.3 モジュールテスト

検索部のモジュールのテストは実装の第四段階から行われた。コーディングが終了した後に、各モジュールが想定通りの動作がするかどうかを正常系、異常系のデータを送って行ったが多くのエラーを出した。それをもとに修正を重ねた。

2.4.4 検索部テスト

検索部のテストは大きく分けると Yahoo API との通信関係、人名辞書・教員リスト関係、表示モジュールに渡すデータ関係の三つに分けることができる。

Yahoo API

テスト項目は API に接続できない、API からの応答が遅い、API からエラーが返ってく

るといったことについてのテストが行われた。主に実装の第五段階で追加された機能についてのテストである。それぞれに対応したメッセージがでるようになっており、それぞれが機能したことを確認した。

人名辞書・教員リスト

両方ともファイルが存在するか、ファイルを読み込むができるかといったことができるかというテストである。ファイルが存在しない場合、読み込めない場合はそれぞれメッセージがでることを確認した。

データ

キーワード、人名、専門分野それぞれに対して設定された大きさの配列を返すことのテストを行った。検索部モジュールでも同様のテストを行ったため、特に心配をすること無くテストを終了させることが出来た。

2.4.5 表示部テスト

表示部のテストでは、20文字以上の文字のカット、記号のカットなど、入力文字数に関する仕様や、検索中に「検索中」表示を出すという仕様が盛り込まれていたが、その部分を実装していなかったため、テスト仕様を松尾さんからいただいてから実装をし、テストを通した。そして、S-Jis以外の文字列が来たら、自動的にはじく、というテスト以外は、通すことができた。詳しいテスト項目は、添付のテスト仕様書を参考にしてほしい。

2.4.6 結合テスト

実際に結合をして動かして分かったことは検索のスピードの遅さ、人名の検索精度が悪いことが挙げられる。検索スピードは二月以降のチューニングで改善していくことができるが、人名の検索をしようとしてもそれが人名なのか地名なのかそれ以外なのかが判別することができないというものが多い。改善策として曖昧な名前を人名辞書から削除するということを考えたが、クライアントから「それは合理的ではない」との指摘を受けたため、NG機能をできるところまで実装するというところで妥協した。

2.4.7 実装フェーズの総括

実装プロセスの妥当性

実装プロセスが妥当だったかと言われれば妥当だったとは言えない。理由として技術調査の遅れが挙げられる。特に大きく影響したのが教員システムの技術調査である。ただできえ実装期間に制限があったにも関わらず、今回のシステムの中核を担うはずであった教員システムが全く使い物にならないということは、もっと早期に知ることができたはずである。今回のプロセスで良かったと思われることは、反復して実装をしたことで

ある。クライアントの都合で仕様が変わったのでそうせざるを得ない状況であったが、その影響で繰り返し実装したことにより、妥当なソースコードができたことはひょうかできるであろう。

何ができて、何ができなかったか

今回のプロジェクトでできたものはテーマに関係している人物とその専門分野の関連をグラフとして表示するシステムである。できなかったことはそれをユーザの使用に耐えるものにするることである。実際に使用してもらっているところを見せてもらおうと、**JAVA**をインストールするところから始まり、キーワードを入力して出てきたグラフが何を意味しているのかをよく理解してない、関連すると予想していたものが出てこないといったことがあった。デザイン、基幹システムともにデモレベルであるということである。しかし、視覚的に関連性を表すことができたことは評価されると思われる。デモレベルではあるが、クライアントに新たな可能性を見せることができるものができた。

2.5 品質保証テストチェック

プロジェクト まねきねこ
ハナビシステム

品質保証テスト チェック方法

作成 松尾壮紘 初版 2008/01/11

— 目次 —

対象ブラウザ

テスト方法

1. 表示メイン (No.1～No.11)
2. 検索メイン (No.12～No.19)
3. 負荷試験 (No.20～No.23)

対象ブラウザ

ブラウザによるテストは下記の2種類でそれぞれ行う。

Windows Internet Explorer 6.x

Windows Firefox 2.0.x

テスト方法

それぞれのテストについてテスト方法と、判定基準を記載する

1. 表示メイン (No.1~No.11)

No.1

テスト内容

- 1 周目のノードが 0
- 検索の結果、1 個も結果がない

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、入力フォームに 1 個も結果が出ないキーワード「※」を入力し検索を行う

結果

- 正：グラフにキーワードのみが表示される
- 誤：それ以外

No.2

テスト内容

- 2 周目のノード数が 0 の場合

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、2 周目のノード数が 0 になるキーワード「※」を入れる

結果

- 正：1 週目までのノードが描画される
- 誤：それ以外

No.3

テスト内容

3周目のノード数が0の場合

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、3周目のノード数が0になるキーワード「※」を入れる

結果

正：2周目までのノードが描画される

誤：それ以外

No.4

テスト内容

重複するノードがある

確認環境

ブラウザ

確認方法

単体テスト用の html を用意しブラウザで Hanabi にアクセスし、重複するノードの表示を確認する

オプション 複雑なパターン

重複数 1 種類 2 個 2 種類 2 個 3 種類 3 個

重複数 1 種類 3 個 2 種類 3 個

重複数 1 種類 4 個

結果

正：重複するノードが1つにまとまって表示される

誤：それ以外

No.5

テスト内容

長い文字のノードがある

確認環境

ブラウザ

確認方法

単体テスト用の `html` を用意しブラウザで `Hanabi` にアクセスし、長い文字の表示を確認する

結果

正：枠内に文字が収まる

誤：文字が枠からはみ出る、見づらい

No.6

テスト内容

ノード MAX 値を完全に満たす結果を描画する

確認環境

ブラウザ

確認方法

単体テスト用の `html` を用意しブラウザで `Hanabi` にアクセスし、すべてのノードを完全に満たして表示できるか確認する

オプション

ノード数 1～10

結果

正：すべてのノードが全部表示される

誤：それ以外

No.7

テスト内容

入力フォームに制限以上の文字数を入れる

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、入力フォーム欄に文字がいくつまで入るか調べる

結果

正：制限されている最大値まで入力できる
誤：それ以外

No.8

テスト内容

POST で入力文字制限以上を強制的に入力する

確認環境

ブラウザ

確認方法

テスト用の html を作りブラウザでテスト html にアクセスし、テストフォームから長い文字列を入力し POST する。

結果

正：制限されている最大値まで検索対象のキーワードとして拾っている。
誤：それ以外

No.9

テスト内容

キーワードに記号を含む

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、記号 `!#$%&'()*+,-“¥<>{}@[].,` をキーワードにいれ検索を行う。

結果

正：記号が `html` タグやメタ文字として実行されないこと
誤：それ以外

No.10

テスト内容

SJIS 固有の問題のある文字列を入力する

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、“表”や”一”など文字コードに“¥”を含むものをキーワードに入力し検索を行う。

結果

正：正常にキーワードが認識される
誤：それ以外

No.11

テスト内容

SJIS 以外の文字コードを入力する

確認環境

ブラウザ

確認方法

ブラウザで Hanabi にアクセスし、SJIS 以外の文字コードをキーワードに入れ検索を行う。

結果

正：エラーメッセージを表示「Sorry , not supported your encoding . Please use sjis .」

誤：それ以外

2. Search (検索) メイン (No.12~No.19)

No.12

テスト内容

Yahoo に接続できない

確認環境

ブラウザ

確認方法

Hosts ファイルに Yahoo の IP を別な IP を設定し、HTTP で接続できない状態にしたうえで、ブラウザで Hanabi にアクセスし、通常の検索を行う

結果

正：指定秒数が過ぎたら「Yahoo に接続できません」のエラーを返す

誤：それ以外

No.13

テスト内容

Yahoo の応答が遅い

確認環境

ブラウザ

確認方法

応答を返さない sleep.cgi を作成し、hosts ファイルで Yahoo へのアクセスをテスト CGI に問い合わせるようにした上で、ブラウザで Hanabi にアクセスし、通常の検索を行う

結果

正：指定秒数を過ぎたらエラーメッセージが表示される「Yahoo の応答がタイムアウトしました」

誤：それ以外

No.14

テスト内容

Yahoo がエラーを返している

確認環境

ブラウザ

確認方法

テスト用の CGI を作成し、HTTP200 を返すが、内容は HTML でエラーを返す false.cgi を作成し、hosts ファイルを書き換え Yahoo へのアクセスを false.cgi へ行ったうえで、ブラウザで Hanabi にアクセスし、通常の検索を行う

結果

正：エラーメッセージが表示される「Yahoo の出力が解析できません」
誤：それ以外

No.15

テスト内容

人名.xml がないとき

確認環境

ブラウザ

確認方法

人名.xml を消し、ブラウザで Hanabi にアクセスし、通常の検索を行う。

結果

正：エラーメッセージが表示される。「人名辞書がありません」
誤：それ以外

No.16

テスト内容

人名.xmlがあるけどデータ形式がおかしくて読めない

確認環境

ブラウザ

確認方法

人名.xmlを文字コードを変えて保存しブラウザで Hanabi にアクセスし、通常の検索をおこなう。

結果

正：エラーメッセージが表示される。「人名辞書が正しくありません」

誤：それ以外

No.17

テスト内容

教員.xmlがないとき

確認環境

ブラウザ

確認方法

教員.xmlを消し、ブラウザで Hanabi にアクセスし、通常の検索を行う。

結果

正：エラーメッセージが表示される。「教員辞書がありません」

誤：それ以外

No.18

テスト内容

教員.xml があるけど読めない(データがおかしい)

確認環境

ブラウザ

確認方法

教員.xml を文字コードを変えて保存しブラウザで Hanabi にアクセスし、通常の検索をおこなう。

結果

正：エラーメッセージが表示される。「教員辞書が正しくありません」

誤：それ以外

No.19

テスト内容

ノード MAX 値どおりに描画されるか

確認環境

ブラウザ

確認方法

単体テスト用の html を用意しブラウザで Hanabi にアクセスし、ノード MAX 値を超えて表示されないか確認する

オプション

ノード MAX 値を変更する (1~10)

結果

正：ノード MAX 値の数だけノードが描画される

誤：それ以外

3. 負荷試験 (No.20～No.23)

No.20

テスト内容

1 秒間に 1 アクセスを 10 分続ける

確認環境

jmeter

確認方法

jmeter でアクセス頻度、秒数を指定し実行する

オプション

Java と Apache のエラーログを監視する

結果

正：エラーログの出力がなく負荷試験後も正常にブラウザから検索が行える
誤：それ以外

No.21

テスト内容

1 秒間に 10 アクセスを 10 分続ける

確認環境

jmeter

確認方法

jmeter でアクセス頻度、秒数を指定し実行する

オプション

Java と Apache のエラーログを監視する

結果

正：エラーログの出力がなく負荷試験後も正常にブラウザから検索が行える
誤：それ以外

No.22

テスト内容

1 秒間に 100 アクセスを 30 秒続ける

確認環境

jmeter

確認方法

jmeter でアクセス頻度、秒数を指定し実行する

オプション

Java と Apache のエラーログを監視する

結果

正：エラーログの出力がなく負荷試験後も正常にブラウザから検索が行える
誤：それ以外

No.23

テスト内容

100 種類のキーワードを順次入れる

確認環境

jmeter

確認方法

jmeter でアクセス頻度、秒数を指定し実行する。キーワードファイルを作り各アクセスごとに違うキーワードでリクエストを投げる。

オプション

Java と Apache のエラーログを監視する

結果

正：エラーログの出力がなく負荷試験後も正常にブラウザから検索が行える
誤：それ以外

Hanabi

品質保証チェックシート

		IE	Firefox	○△×	コメント	
表示メ	1	1週目のノード数が0	○	○	○	
	2	2週目のノード数が0	○	○	○	
	3	3週目のノード数が0	○	○	○	
	4	重複するノードがある	○	○	○	オプション未チェック
	5	長い文字のノードがある	○	○	○	
	(検索部19)	ノードMAX値のとおり描画される	-	-	-	MAX値の変更は検索側の設定
	6	ノードMAX値を完全に満たす結果を描画する	○	○	○	
	7	入力フォームに制限以上の文字数を入れる	○	○	○	
	8	POSTで入力文字制限以上を強制入力	○	○	○	
	9	キーワードに記号を含む	○	○	○	記号はすべて消す
	10	SJIS固有の問題のある文字列を入力する	○	○	○	○表、表示、一、キーの4種類がOK
11	SJIS以外の文字コードを入力する	△	△	△	そのまま動いてしまうので対応考え中	
検索メ	12	Yahooに接続できない			○	
	13	Yahooの応答が遅い			○	
	14	Yahooがエラーを返している				未チェック
	15	人名.xmlが無いとき			○	
	16	人名.xmlがあるけどデータ形式がおかしい			○	
	17	教員.xmlが無いとき			○	
	18	教員.xmlがあるけどデータ形式がおかしい			○	
	19	ノードMAX値のとおり描画される				未チェック
負荷試	20	1秒間に1アクセスを10分			○	YahooAPIの規制にかかるため200秒に短縮
	21	1秒間に10アクセスを10分			○	300秒に短縮
	22	1秒間に100アクセスを30秒			△	負荷クライアント側がハングアップ
	23	100種類のキーワード			○	

2.6 評価

2.6.1 評価フェーズの要約と総括

2008年1月25日～1月29日にかけて、研究室を探しているSFCの大学生（以下、学生）6人と、ベンチャー企業の社長2名と社員1名（以下、社長）に、SFC校内と藤沢イノベーションビレッジで、HANABIシステムの使い方を説明した後、実際にHANABIシステムを10～15分程度使ってもらい、前述したアンケートに答えて頂いた。パソコンに関しては、アンケート調査実施者の福田宙音（まねきねこチームメンバー）のパソコンを使い、ユーザーに対し1人ずつ実施した。この時のシステムのバージョンは、バージョン番号を全体ではつけていないため不明である。

結論から言うと、学生にとっては使えないシステムで、社長にとっては使える可能性があるシステムであることが、今回のユーザーテストの中のアンケートとヒアリングを通してわかった。今回のこのシステムは、「慶應義塾に蓄積されている研究成果や学術成果を、学外にも広く開放し、専門家だけでなく一般の方々にも自由に利用できる仕組み」を支援するという目的のために作られたシステムだが、学生から得られた評価としてはビジュアル面でのよさに限られ、研究会を決めるのに有効な情報が得られないという結果に終わり、当初の目的を満たせないで終わった。学生からのヒアリングで、大量に出てくる関連図の人名をいちいちクリックし、すべて調べるのは面倒くさく思ったという意見を頂いたが、どれが打ち込んだキーワードと関連性が近いのかのガイドがないことが原因だと考えられる。それに対して、社長の側としては、前述したような使いにくさがあっても本気で提携先を探そうとするため、代表的な教授の論文や、教授の連絡先が表示されれば使っていただけという評価を得られた。

このシステムはユーザーにとってまだまだ親切なシステムだとは言いがたいので根気がいり、その根気を持つベンチャー企業の社長や社員の方々にとっては使える可能性があるものになったと考えられる。

2.6.2 ユーザーテストの方法

ユーザーテストに関し、クライアント側に以下の内容で大丈夫かので了承を、実装完成前に頂いた。ただ、テスト仕様の作成（2）の、所属研究室を決めたい外部からの大学院受験生は集めるのは無理なため、研究室を探している SFC の大学生を対象にした。また、検索ステップ数を記録するという成果の測定方法は、検索ステップ数を計測するのが困難であるという理由から、行わないことになった。以下、クライアントに了承を頂いたテスト仕様案である。

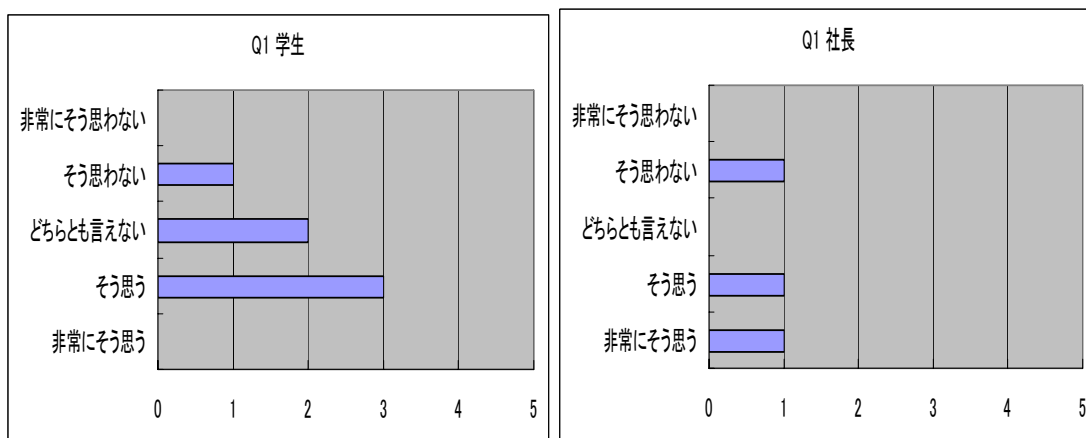
- テスト仕様の作成（1）
 - 対象者は誰か
共同研究者を探したい、若手起業家
※こちらで準備する予定です。
 - 規模はどれほどか
3組程度（6名）
 - いつ行うか
完成後（2007年12月20日以降）
 - シナリオはどんなものか
各企業で興味のある検索単語を3つ用意してもらおう。
そして、1人は一般の検索で、もう1人は本システムを使って、誰と研究を行いたいかが決定してもらおう。
 - 何で成果を測るか
アンケート
検索ステップ数を記録し、一般の検索システムの場合と比較
- テスト仕様の作成（2）
 - 対象者は誰か
所属研究室を決めたい、外部からの大学院受験生
 - 規模はどれほどか
8名程度
 - いつ行うか
完成後（2007年12月20日以降）
 - シナリオはどんなものか
興味のある研究分野に関する単語を5つほど用意してもらい、所属研究室を決めてもらう。
 - 何で成果を測るか
アンケート
検索ステップ数を記録し、一般の検索システムの場合と比較

2.6.3 ユーザテストの結果

2008年1月25日～1月29日にかけて、研究室を探しているSFCの大学生（以下、学生）6人と、ベンチャー企業の社長2名と社員1名（以下、社長）に、SFC校内と藤沢イノベーションビルで、HANABIシステムの使い方を説明した後、実際にHANABIシステムを10～15分程度使ってもらい、前述したアンケートに答えて頂いた。パソコンに関しては、アンケート調査実施者の福田宙音（まねきねこチームメンバー）のパソコンを使い、ユーザーに対し1人ずつ実施した。この時のシステムのバージョンは、バージョン番号を全体ではつけていないため不明である。その集計結果が以下のものになる。

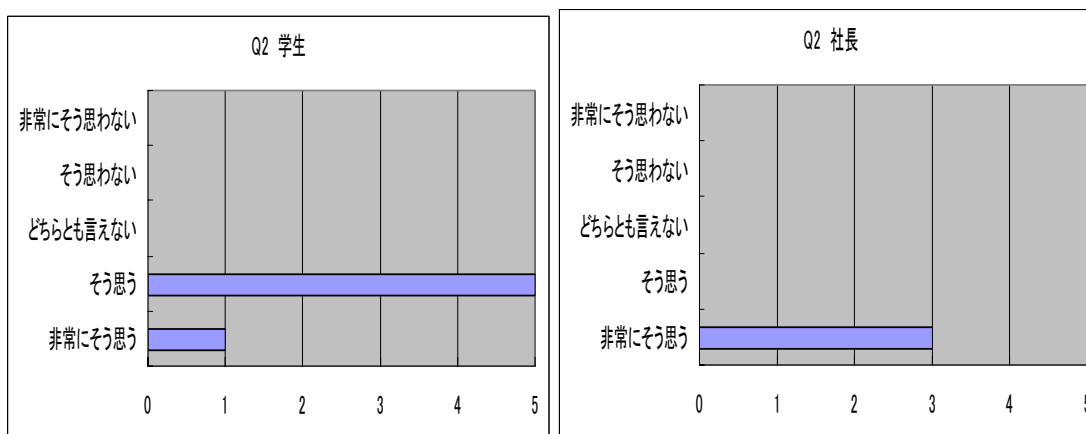
■ サイトのデザインについて

Q1 見た目はわかりやすいですか？



見た目に関しては中～高評価が得られた。

Q2 操作はしやすかったですか？

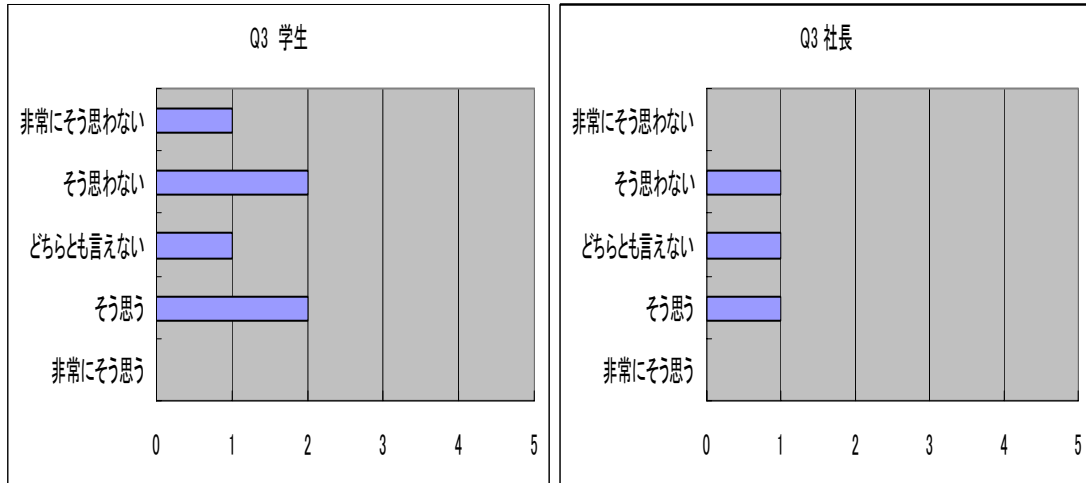


学生からの評価は高評価だが、この回答に関しては、結果を疑う必要がある。なぜなら、Q7の回答において、「検索速度が遅い」「目的のところへいけないことがある」「文字が重

なって見づらいところがある」といった、操作性に関するマイナスの評価が記述されているためだ。このような操作性のしやすさを高評価にしてしまった要因として、ユーザーテストをはじめる前に、操作に関する説明を毎回詳細にしすぎたことが考えられる。

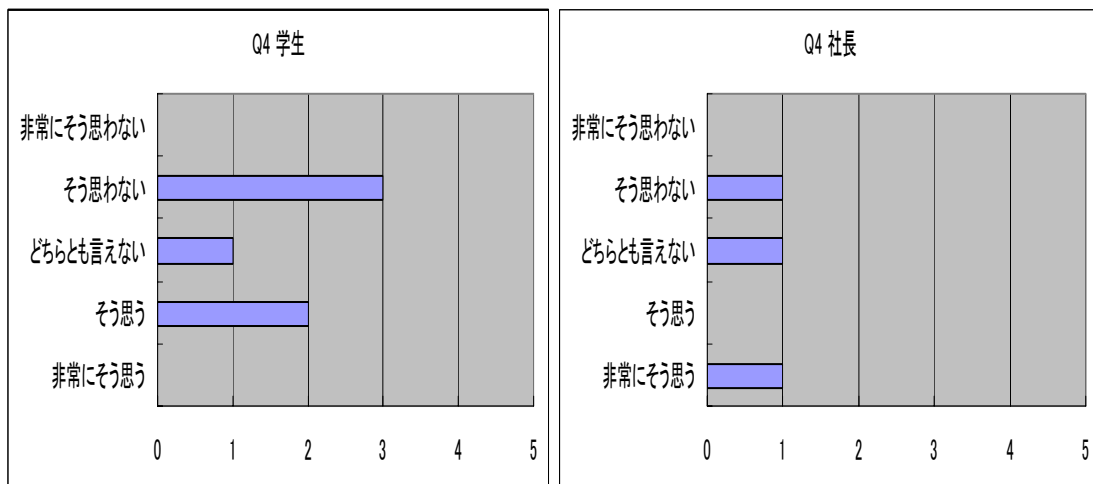
■ 検索結果の情報精度について

Q3 関連図形式で検索結果を示すことで、目的の情報にすばやく到達できましたか？



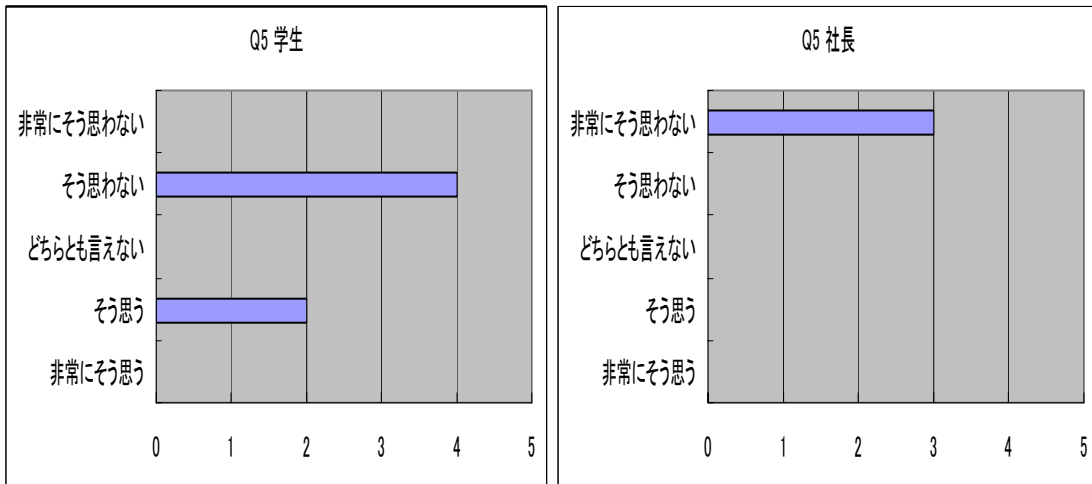
学生に関しても社長に関しても評価にばらつきがでた。これは、各個人によって検索したキーワードが違い、キーワードによっては欲しい情報が得られるが、あるキーワードに関しては何も出てこないという状況があったためだと推測される。例えば、「プログラミング」と打ち込んだ場合、それに関わる研究者の名前は精度が高く表示されるが、「介護」と打ち込んだ場合、全く関連のないドイツ語の教授が出てくることなどだ。これは人名辞書を500件という少数で作ってしまったことに起因する。

Q4 キーワードと人、人と研究分野の繋がりは適切な情報が出ていたでしょうか？



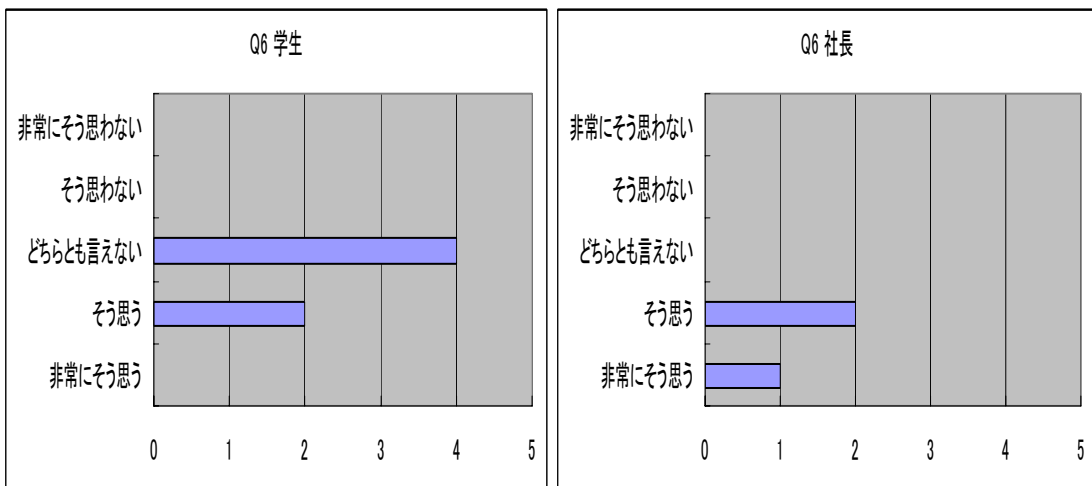
これは Q3 と同じ、人名辞書の問題に起因する。

Q5 バグやスペルミス等の不具合は気になりましたか？



バグやスペルミスはほとんどなかった。

Q6 この検索システムをまた使ってみたいですか？



社長に関しては、また使ってみたいという意見が多かった。

■気になった点を聞かせてください

Q7 この検索システムの問題点を2つ挙げて下さい。

学生から得られた回答

- キーワードで出てくるべき人が出てこなかった
- インターネットアーキテクチャをクリックしても出てこない
- 検索速度が遅い

- どうやって関連キーワードを作成しているかが不明
- 目的のところへいけないことがある
- 文字が重なって見づらいところがある
- 検索結果が釈然としない（以外な人がひっかかる）
- 部分一致で引っかかる場合がある
- 似たようなシステムが既にある

社長から得られた回答

- キーワードの意味がよくわからない
- 複数の研究室のどれがよいかわからない
- なんとなく機能的にさみしい感じがする
- 「検索システム」とよんでいる点
- 関係図が動く意味がわからない

Q 8 この検索システムのいい点を3つ挙げて下さい。

学生から得られた回答

- 見やすい
- つながりが見えてよい
- シンプルなところ
- ほしそうなのが出てくる
- めずらしい
- 関連キーワードが出る
- 感覚的に操作できる
- 見た目に楽しい
- 意外な人の研究内容がわかってよい
- ビジュアル的にわかりやすい

→学生からは見た目の目新しさは評価されたが、Google で検索した方が早いといわれるケースが多かった。

社長から得られた回答

- 研究室からしぼりこめる
- 分割画面がよい
- ビジュアルとカラーリング
- ワードからの発想のつながりが見えてとてもクリエイティブ
- 一度に関連する人名・カテゴリを検索できる点
- クリックするだけで、詳細情報が検索でき、複数ウィンドウを開く手間が省ける

→社長からは部分的に改善すれば今後も使う余地はあるという評価が得られた。

■今後の拡張性について

Q9 今回(人名)-(研究分野)-(人名)の関係図を表示しましたが、他にも「会社、企業」「場所」「時間」といったものともリンクが可能です。もし自由に関係性を結べるとしたら、どのようなグラフを表示してみたいですか？

例：(ゲーム会社)と(ゲームジャンル)と(ゲームタイトル)

学生

- food price place
- game price place
- 女 デート 場所
- 映画 スケジュール レビュー
- 年齢 性別 研究テーマ
- 難易度 充実度 分野
- 本のジャンル 著作 著者
- 電車 ダイヤ 目的地

社長

- 人名 分野 代表的な論文
- 人名 分野 製品例
- 女優 写真 動画
- 夢 記憶 現実
- 営業担当 営業先 マーケティング担当
- 自分 興味分野 友人

2.6.4 ユーザーテストに利用したアンケート

以下のアンケートを使用し、ユーザーテストを行った。

HANABIシステムに関するアンケート調査

※以下の項目の中で当てはまるもの1つだけに丸をお付け下さい。

■サイトのデザインについてお聞きます

Q 1 見た目はわかりやすいですか？

非常にそう思う	そう思う	どちらとも言えない	そう思わない	非常にそう思わない
---------	------	-----------	--------	-----------

Q 2 操作はしやすかったですか？

非常にそう思う	そう思う	どちらとも言えない	そう思わない	非常にそう思わない
---------	------	-----------	--------	-----------

■検索結果の情報精度についてお聞きします

Q 3 関連図形式で検索結果を示すことで、目的の情報にすばやく到達できましたか？

非常にそう思う	そう思う	どちらとも言えない	そう思わない	非常にそう思わない
---------	------	-----------	--------	-----------

Q 4 キーワードと人、人と研究分野の繋がりには適切な情報が出ていたでしょうか？

非常にそう思う	そう思う	どちらとも言えない	そう思わない	非常にそう思わない
---------	------	-----------	--------	-----------

■システムについてお聞きします

Q 5 バグやスペルミス等の不具合は気になりましたか？

非常にそう思う	そう思う	どちらとも言えない	そう思わない	非常にそう思わない
---------	------	-----------	--------	-----------

Q 6 この検索システムをまた使ってみたいですか？

--	--	--	--	--

非常にそう思う そう思う どちらとも言えない そう思わない 非常にそう思わない

■気になった点を聞かせください

Q 7 この検索システムの問題点を2つ挙げて下さい。

[]

[]

Q 8 この検索システムのいい点を3つ挙げて下さい。

[]

[]

■今後の拡張性について

Q 9 今回 (人名)-(研究分野)-(人名) の関係図を表示しましたが、他にも「会社、企業」「場所」「時間」といったものともリンクが可能です。もし自由に関係性を結べるとしたら、どのようなグラフを表示してみたいですか？

例：(ゲーム会社) と (ゲームジャンル) と (ゲームタイトル)

- () と () と ()
- () と () と ()

☆ご協力ありがとうございました。差し支えない範囲内で以下にご記入下さい。

性別： 男性 女性 年齢： _____ 歳 ご職業： _____

※アンケート結果に関してももちろん厳重に管理しますが、レポートに載せることをご了承下さい。

3 おわりに

3.1 プロジェクトの総括

本プロジェクトでは、研究者間の関連を図示するシステム「Hanavi システム」が、慶応の学術コンテンツの利用、および産学連携の促進に役立つという仮説を立てた。

今回、この仮説に基づいて Hanavi を実際に作ってみて、実際に複数のベンチャー企業社長と、学生に使ってもらった結果、とくに、ベンチャー企業の社長から「可能性を感じる」「これで教授の代表論文と連絡先が出れば、ぜひ使ってみたい」という意見をいただき、その仮説が正しそうだという感触を得た。

しかし、今回開発したシステムは、教員リストと、人名辞書の準備不足による精度の悪さ、ユースケースの分析不足によるユーザインターフェースの悪さから、実用レベルには至っていない。

今回のプロジェクトでは、「動くもの」と「使ってもらえるもの」の差を甘くみていた。アイデアをクライアントに承認していただいた時点で満足してしまい、エンドユーザのことを考えることができなかつたのが、プロジェクトとしての反省点である。

最後に、プロジェクト全体の活動について述べる。今回作ったのは、あくまで Hanavi システムのプロトタイプであり、実用に耐えるものではなかつた。しかし、度重なるスコープ変更と、実装力不足のメンバー、という状況の中、少なくともプロトタイプとして動くものを、比較的綺麗なソースコードで書くことができた。そして、まねきねこプロジェクトでは、システムの提案から、実装、評価までのすべてのフェーズを行った。はじめから最後まで、トラブルもありながらプロジェクトをやりきった、という意味で、このまねきねこプロジェクトの活動は、とても有意義だったと思う。荒波の上の小船のようなプロジェクトを、うまく陸まで（死者も出さずに）舵取りしてくれた、PM の松尾さんに感謝したい。

プロジェクトマネジメント報告書

PM 松尾壮紘
株式会社ドワンゴ
matsuo@dwango.co.jp

まず最初に、このプロジェクトは完了しなかったということを断り書きしなくてはならない。

それなりに動く物はできた。

ユーザーテストの反応も思っていたよりはいくぶん良いものであったが、

WBS で完成していないままの作業や、EVM による実績値を見ると

果たして終わったのだろうか？と大いに疑問が残る。

クライアントに提示した納品予定のものも出来上がっていないものがあり、未完のままである。

PM が作成物の進行管理をきっちり把握し、未完の物を洗い出し、完了まで作業をすすめることが出来なかったことを反省している。

プロジェクト計画書からみた QCD

初期に立てた計画と、実際の値がどうなったかを比較検討していく

・ Cost 面の考察

作業時間 (単位 時間)

	予定	総時間	直接	間接
松尾	240	190	128.5	14.5
荒木	120	127	87.5	21.5
福田	120	99	54	32
松田	120	186.5	122.5	42
チーム	600	602.5	392.5	110

※総時間には授業時間含む

コスト面では見積りの値にとっても近いものとなった。

時間見積りは、

過去のコラマネの報告を元に1日1時間がんばってもらえればいいかな、
という単純計算によるものだったが、それに近い値になったのは、
メンバーが最後までモチベーションを保って作業を行ってくれたおかげではないか。

ただし、その裏には

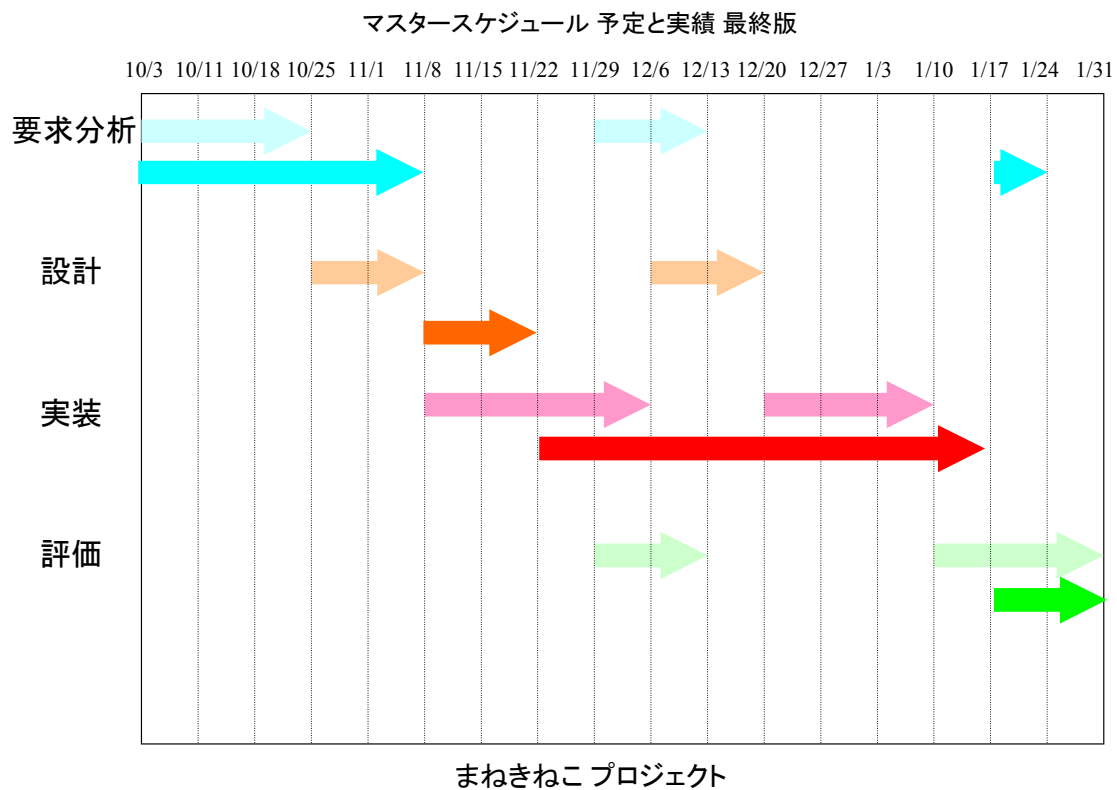
その時間でやれるところまでしかやらなかった。という部分が含まれている。
予定していた作業が出来ていないのに、PM から作業完成までやるよう追い込みをかけられ
なかったという部分があり、作業時間が増えなかっただけであるとも考えている。
スケジュールは伸び期限内(1/31)に終わらなかっただけで、2/26 まで開発を継続して行う
予定になっているため、その時間を加算すると大幅なコスト超過となることが確実にな
っている。

メンバーによって作業時間に倍近い差が出来ているのは

プロジェクト進行途中での大幅な仕様変更により、余計に時間がかかったことが要因の一
つ、またメンバーのスキル差により担当範囲に大きな偏りがあった為である。

・ Delivery 面による考察

当初予定していたスケジュールを薄い色、
実際のスケジュールを濃い色で示してある。



当初は反復型でクライアントの希望するものを反映させたものにする予定であったが、大幅なスケジュール変更により行えなくなった。

スケジュールを大きく変更しなければならなかった理由は、

- ・ メンバーの実装能力の把握不足
- ・ 大幅な実装変更

集まったメンバーは、皆がコーディングが得意ではないと言っていた為ある程度折り込み済みではあったが、それでもスケジュール超過した感じが否めない。だが結果からすればよく調べて動いてくれたと PM は感じている。

影響が大きかったのは外部サイトにあるデータを頼っていたが、そのデータが使えないことが分かり実装を大幅に変更せざるを得なかったこと。

早い段階で気付けばよかったのだが、実装を行ってしてからサイトのデータが使えないこ

とに気付いた為、後手の対応となり大幅に遅れる原因となった。これは検証段階で調査すれば気付く部分であったので下調べが甘かった。

さらにクライアントとの意思疎通がうまくいってなかったために、仕様変更や追加タスクが発生した。追加分においてはメンバーのタスクが既にいっぱいであったため PM が実作業を行う部分があった。

自分達がどのように実装して、どのように使ってもらうのかをクライアントと確認できていなかった為に起こったことで、最初の段階でイメージしているものがきちんと伝わるモックのようなものを作っていれば回避できていた部分かもしれない。

またスケジュールの遅れから、焦って実装に手を出した為、設計が満足にできておらず仕様書が残らない原因になった。

Quality 面からの考察

Web の検索システムであることから

サービスの質を、速度・精度・面白さ の3つの点から計測してみる

・ 速度

検索ボタンを押してから結果が返るまで

目標：5 秒以内

実測：5.36 秒

目標に僅かにとどかなかったものの、実際に使ってみてストレスの無い応答速度であった。初期の段階では1回の検索に数分を要し、実用に耐えられるものではなかったがパフォーマンスチューニングにより劇的な速度改善を遂げた。

またこの計測には含まれていないが、過去の **g-mod** から結果をキャッシュするものを取り入れ高速化を実施した。ただし、高速化後のテストが出来ていない為、その測定値はない。

・ 精度

検索制度に関してはユーザテストを行い、アンケートに回答してもらったものから抜粋

キーワードと人、人と研究分野の繋がりには適切な情報が出ていたでしょうか？

非常に良い	1
良い	2
どちらとも言えない	2
悪い	4
非常に悪い	0

精度は非常によくない。大きな原因は分かっているがクライアントからもその部分の修正案があがっているが、間に合わせる事が出来なかった。改良のため 2/27 まで継続開発を行う。

・ 面白さ

サービスを継続してユーザに使ってもらうためには、また使いたいと思う面白さがなければ人はすぐ去ってしまう。

同様にユーザテストから抜粋

この検索システムをまた使ってみたいですか？

非常に良い	1
良い	4
どちらとも言えない	4
悪い	0
非常に悪い	0

結果はまずまずだが、
ユーザを飽きさせない演出のアイデアはあったが、これも期限に間に合わせる事が出来なかったため、2/26まで継続開発を行うこととなっている。

他の点からみた Quality

- ・仕様書

仕様書が非常に少ない、
ソースをクライアントに納め、今後クライアント側で保守していくことを考えると、仕様書がほとんど残っていないのは非情に不安の残る部分。

- ・コードの質

コードレビューを行い PMO の松澤さんに見てもらったところ、意外に好評であった。
普段コーディングをしない私からみても、見やすいと感じるものであった。

- ・バグ

品質保証テスト仕様書を作成し、どのようなテストをどのようなチェック方法で行ったかの提示を行った。

作成担当者に担当箇所のテストを行ってもらい確認と修正を行ってもらった。

また PM は負荷テストを行い、速度面の測定と、高負荷時のトラブルがないか調査を行った。

WBSによる作業成果

フェイズ	要求分析	設計	実装	テスト	合計
松尾	100	50	100	33	283
荒木	366	100	383	200	1050
福田	166	100	0	150	416
松田	166	100	183	150	566
チーム	800	350	666	533	2315/3200

※各フェイズにおける成果合計は 800

最初から最後まで広範囲に渡って作業の中心を担ってくれた荒木さんの成果が目立つ。途中で大幅な仕様変更が入った松田くんは作業時間を最も費やしてくれたのに成果が少ないという残念な数字になってしまった。

福田くんはコーディングができなかったが、限定した作業を独立してやってくれたことでプロジェクトの成果に貢献してくれている。

ただし、1/31 時点で作業が完了しておらず、合計の値は足りていない。

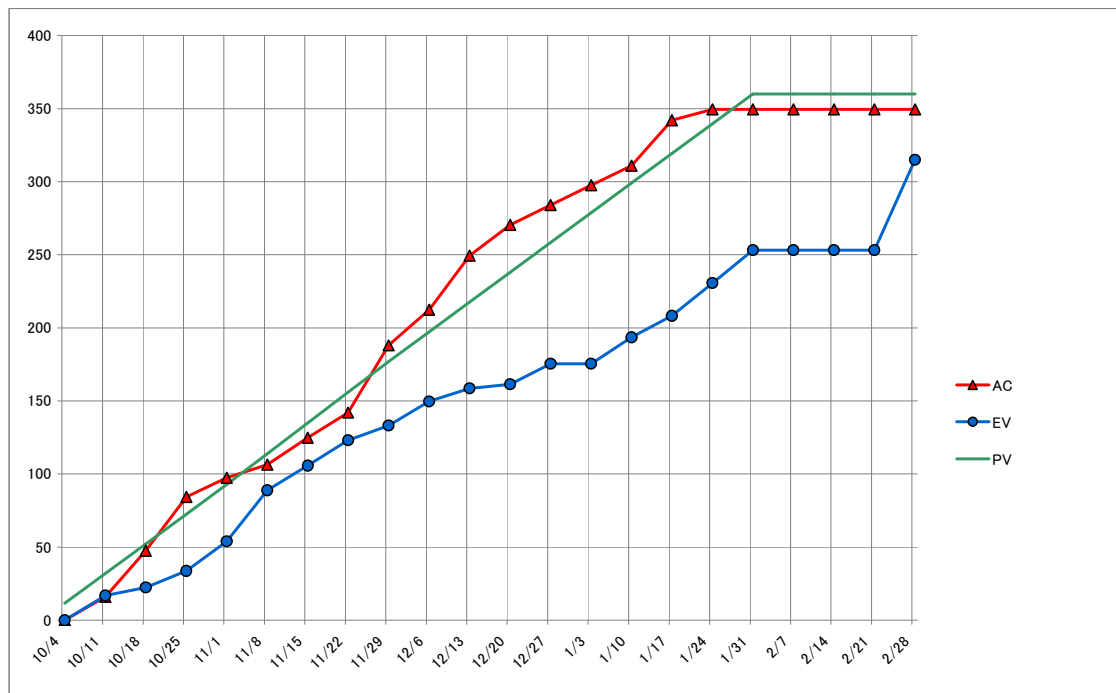
2/26 まで継続して開発を行う予定である。

・ステップ計算 (使用ソフト: コロ助)

荒木 1101 行

松田 342 行

EVM グラフ



全体において予定していたスケジュールをオーバーし、

AC については、メンバーが最初から最後までモチベーションを保って作業してくれたことが、順調な線として出ていると PM は感じている。

EV については、出だしから順調とはいえないが、問題は大きく乖離を始めた 12 月、この時期に大幅な仕様変更があり、それまで作成していたものが無駄になったため EV にカウントされないこととなった。

年明けから順調な伸びを見せるが、結局目標としていた品質までは届かなかった。

成果物

各フェイズ		1月31日	2月27日
要求分析	プロジェクト提案書	○	○
設計	画面表示仕様書	○	○
	モジュール仕様書	×	×
	システム仕様書	○	○
	外部入出力仕様書	○	○
	モジュール間入出力仕様書	×	×
実装	単体テスト仕様書	○	○
	結合テスト仕様書	○	○
	環境設定構築マニュアル	×	○
	βシステム	○	○
テスト	クライアントフィードバック	○	○
	市場導入テストの結果	○	○
	最終システム	×	○
	最終報告書	○	○

1月31日

この時点で成果物が完成しておらず、主にクライアントフィードバックを受け修正すべき点がほとんど手付かずの状態であったため、期間を延長し最終発表会まで開発を継続することにした。

モジュール仕様書、モジュール間入出力仕様書は、設計の段階で作成していなかったため、保守に必要な情報を納品物のシステム仕様書に盛り込むことで対応とした。

