慶應義塾大学学術情報リポジトリ
Keio Associated Repository of Academic resouces

Title	BufferMan : 遠隔での課題発見型協調作業を支援するテキストチャットシステム
Sub Title	BufferMan : Text chat system for agenda finding collaborative work in remote
Author	竹内, 冠太(Takeuchi, Kanta) 杉浦, 一徳(Sugiura, Kazunori)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2011
Jtitle	
JaLC DOI	
Abstract	本論では、プロジェクトにおける課題発見型協調作業を支援するための、テキストチャットシステムBufferManについて述べる。近年遠隔での協調作業においてテキストチャットの有用性が再度注目されている。しかし既存のテキストチャットは同期的なコミュニケーションをとること自体を目的としており、課題発見型協調作業に必要な機能を備えていない。BufferManはテキストチャット機能に加えて、テキストチャットでのコミュニケーション内で出てきた懸案事項とタスクを即時的に保持することが出来、また、テキストチャットのログを電子メールで共有することが出来る。BufferManを使って遠隔協調作業を行うことで、より効果的な課題発見型協調作業が実現できる。 本研究ではBufferManの全ての開発工程を述べると共に、実験、評価についても言及する。そして遠隔での課題発見型協調作業におけるBufferManの有効性を考察するものである。
Notes	修士学位論文. 2011年度メディアデザイン学 第174号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002011-0174

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

2011 年度　　修士論文

BufferMan:遠隔での課題発見型協調作業を支援するテキストチャットシステム

KEIO MEDIA DESIGN

慶應義塾大学大学院
メディアデザイン研究科

竹内　冠太

本論文は慶應義塾大学大学院メディアデザイン研究科に
修士 (メディアデザイン学) 授与の要件として提出した修士論文である。

竹内　冠太

指導教員：

杉浦 一徳 准教授　　(主指導教員)

植木 淳朗 特任講師　(副指導教員)

審査委員：

杉浦 一徳 准教授　　(主査)

植木 淳朗 特任講師　(副査)

稲見 昌彦 教授　　(副査)

BufferMan:遠隔での課題発見型協調作業を支援するテキストチャットシステム

内容梗概

本論では、プロジェクトにおける課題発見型協調作業を支援するための、テキストチャットシステム BufferMan について述べる。近年遠隔での協調作業においてテキストチャットの有用性が再度注目されている。しかし既存のテキストチャットは同期的なコミュニケーションをとること自体を目的としており、課題発見型協調作業に必要な機能を備えていない。BufferMan はテキストチャット機能に加えて、テキストチャットでのコミュニケーション内で出てきた懸案事項とタスクを即時的に保持することが出来、また、テキストチャットのログを電子メールで共有することが出来る。BufferMan を使って遠隔協調作業を行うことで、より効果的な課題発見型協調作業が実現できる。

本研究では BufferMan の全ての開発工程を述べると共に、実験、評価についても言及する。そして遠隔での課題発見型協調作業における BufferMan の有効性を考察するものである。

キーワード

課題発見型作業, 協調作業, チャット, 創発

慶應義塾大学大学院　メディアデザイン研究科

竹内　冠太

BufferMan:Text Chat System for Agenda Finding Collaborative Work in Remote

Abstract

In this thesis I write about text chat system called BufferMan. BufferMan supports agenda finding collaborative work in projects. Existing chat systems are focusing on the synchronous communication itself, but they don't have functions needed for agenda finding collaboration work. BufferMan enables users to keep agendas and tasks immediately, and share text chat log easily using e-mail. BufferMan achieves efficient agenda finding collaboration work in projects.

In this paper I introduce whole developing process of BufferMan, its experiment, and its evaluation.

Keywords:

Agenda finding work, Collaborative Work, Text Chat, Emergence

Graduate School of Media Design, Keio University

Kanta Takeuchi

目　　次

第 1 章　序論 **1**

第 2 章　テキストチャットを使った課題発見型協調作業の現状と課題 **4**

2.1. チャットとは 4

2.1.1 チャットとは 4

2.2. 課題発見型協調作業支援のための取り組み 5

2.2.1 協調作業とは 5

2.2.2 協調作業が取り組む課題の種類 6

2.2.3 協調作業を支援するツール 9

2.3. テキストチャットによる課題発見型協調作業 15

2.3.1 課題発見型協調作業におけるテキストチャットの優位性 . . 15

2.4. テキストチャットを使った課題発見型協調作業の現状分析と問題点 17

2.4.1 既存のテキストチャットシステムの種類と特徴 18

2.4.2 テキストチャットを使った課題発見型協調作業の分析と問題点 22

2.5. ２章まとめ 24

第 3 章　BufferMan の提案 **25**

3.1. BufferMan の提案 25

3.1.1 想定する対象ユーザー 25

3.1.2 BufferMan が扱う協調作業 25

3.1.3 課題発見型協調作業に必要とされるモデル 25

3.1.4 課題発見型協調作業に必要とされる機能 28

3.2. Google Person Finder Bot 制作チームへのフィールドワーク . . . 30

3.2.1 フィールドワークの目的 30
3.2.2 Google Person Finder Bot とは 30
3.2.3 Google Person Finder Bot 制作チームの構成 31
3.3. Google Person Finder Bot 制作チームでのコミュニケーションモデル 32
3.3.1 情報共有の方法 33
3.3.2 情報共有の内容 34
3.3.3 情報を共有する対象 34
3.3.4 情報を共有するタイミング 34

第 4 章 BufferMan の設計・実装 36
4.1. BufferMan の画面構成要素 36
4.1.1 トップ画面 36
4.1.2 アカウント作成画面 36
4.1.3 新規プロジェクト作成画面 38
4.1.4 管理画面 38
4.1.5 テキストチャット画面 38
4.2. ログメール送信機能 46
4.2.1 テキストチャットログ送信機能の設計 46
4.2.2 テキストチャットログ送信機能の実装 46

第 5 章 BufferMan の実験、評価 48
5.1. BufferMan の性能実験環境 48
5.1.1 テキストチャットログ送信機能の実験方法 48
5.1.2 BufferMan の実験対象 48
5.1.3 テキストチャットログ送信機能の実験環境 49
5.2. テキストチャット発言表示時間の測定実験と評価 49
5.2.1 テキストチャットログ送信機能の発言表示時間の測定実験 50
5.2.2 テキストチャットログ送信機能の発言表示時間の評価 . . . 50
5.3. テキストチャットログ送信機能の実験評価 51
5.3.1 テキストチャットログ送信機能の発言表示時間の測定実験 51

5.3.2　テキストチャットログ送信機能の発言表示時間の評価 . . . 52
5.4. テキストチャットログ送信機能評価のまとめ 52

第 6 章　結論と今後の展望 53
6.1. 結論 53
6.2. 今後の展望 53

図　目　次

2.1 Skype 画面 19
2.2 microsoft messenger 画面 20
2.3 IRC 画面 21
2.4 ichat 画面 22

3.1 Google Person Finder 31
3.2 Google Person Finder Bot 32
3.3 コミュニケーションモデル分析 33

4.1 システム概要図 37
4.2 テキストチャット画面 39
4.3 プロジェクト選択画面 40
4.4 参加者一覧画面 41
4.5 トピック一覧画面 43
4.6 テキストチャット画面 44
4.7 タスク一覧画面 45
4.8 メール送信機能 47

5.1 実験結果　単位＝横軸；発言　縦軸；秒 50
5.2 実験結果　単位＝横軸；発言　縦軸；秒 51

表　目　次

5.1　実験環境 49

第1章
序　　論

本研究ではデザイン思考を用いて、課題発見型プロジェクト支援に特化したチャットシステム BufferMan[1] の提案、実装を行い、その安定した動作とプロジェクト進行における有用性を評価した。

近年インターネットを使った協調作業支援ツールにおけるテキストチャットの有効性が再び注目されている。

かつてはインターネットを使った遠隔地での協調作業においては、インターネットの帯域幅や PC の性能の制限から、IRC などに代表されるテキストのみを用いたテキストチャットが主流であった。[2] しかし近年ではインターネットの帯域増強や PC の性能の向上により、テキストのみならず音声を使ったボイスチャットや、それにそれぞれのユーザーの動画も加えたビデオチャットなどのツールが充実している。

これらのツールは表情や声色などをリアルタイムに伝えることができるため、感情のやり取りを含めたコミュニケーション自体を目的とするコミュニケーションには有効であった。また、同様に課題解決型もしくは課題発見型の協調作業支援のためのツールとしても有効だと考えられてきた。しかし実際は、既成の決定事項が少ない課題発見型の協調作業においては即興性とを通したタスクの明確化が必要であるため、相手の立場や周囲の目を気にしないコミュニケーションが可能であり、また外部からの情報に邪魔されること無く作業に集中することができるテキストチャットの方が適したツールである。実際に、課題発見型プロジェクトの多くが作業の最中に Skype[3] や Messenger[4] などのテキストチャットを併用している。またサービスとしても chatwork[5] などのテキストチャットとタスク管理システムを統合したものが発表されている。

プロジェクトにおいてそれらテキストチャットは様々な目的で使用される。それらの目的とは雑談、情報共有、議題に関するタスクの明確化などがある。本研究ではその中でもテキストチャットのタスク明確化のための使用方法に注目して議論する。

テキストチャットを使った議論の中では、即興性と発話の交換を通じて、次にすべき議題や進行中の議題にまつわるタスクを明確化していく。多くのテキストチャットは各ユーザーの発言をリアルタイムでタイムラインに表示することが主な機能となっており、同期的なコミュニケーションを成立させるということに主眼が置かれている。この設計は人間も計算機と同様の処理手順で問題を処理していくという思想が背景にある。すなわち、出てきた議題に関して議論する、議題に関してのトピックを明確にするという作業を、それぞれが明らかになった順番に逐次的に処理するという思想に基づいている。

しかしこの設計は本来の人間の問題処理手順に適応していない。人間は新しい議題が明らかになっても、現在進行中の議題についての議論やタスクの明確化を終えてから、先に明らかになった議題に移行すると考えられる。つまり、議論の中で明らかになった議題等の次に議論すべき話題を後で参照するために一時的に蓄積しておく場所が必要であると考えられる。

本研究では上記の問題を解決するためのテキストチャットシステムを設計、実装するにあたりデザイン思考と呼ばれる手法を使用した [6][7]。これはデザインの上流に哲学・ビジョンの構築、技術の棚卸とフィールドワーク、コンセプト／モデルの構築、デザインの４ステップを、プロセスの下流には実証、ビジネスモデル構築、ビジネスオペレーションの３ステップを設定したもので、これを何度も繰り返すことでよりよい成果物を製作する方法論である。

ユーザーに提供したい経験を実現するために、実際のフィールドワークによる参与観察とプロトタイプを対象者に実際に使ってもらうことによって得られるフィードバックを用いるため、より人間にとって自然に使える成果物を実現するものである。

フィールドワークは Google Person Finder Bot 制作チームを対象に行なった。Google Person Finder Bot は 2011 年 3 月 11 日に発生した東北関東大震災時に製

作された、被災者の安否確認サービスである [8][9]。Google や Twitter[10] などとの連携を取りながら震災発生後 1 日でプロトタイプを製作し、最終的に１５万件の安否確認の情報を Twitter 上に tweet した。これは課題発見型の協調作業として有効なモデルを実現していたと言える。フィールドワークではこの Google Person Finder Bot チームでのコミュニケーションモデルを分析した。

Google Person Finder Bot チーム内は主に 3 つの情報をメンバー全体で共有するために蓄積していた。それらは誰かが何かタスクを終えたという情報、チームを取り巻く状況についての外部情報、そしてメンバーによる提案の 3 つである。

本研究ではこれらの情報を蓄積し、確認しながら使用することのできるテキストチャットシステム BufferMan を提案、実装し、評価した。

第2章
テキストチャットを使った課題発見型協調作業の現状と課題

2.1. チャットとは

2.1.1 チャットとは

チャットはインターネット上の同期的コミュニケーション手段のひとつである。同期的とは情報の送信者と受信者の間に情報共有の時間差がないもの、リアルタイムであることを指す。また、逆に非同期的とは情報の送信者と受信者の間に情報共有の時間差があるものを指す。コミュニケーションとは複数の人間がテキスト、音声、動画、感情等の情報を双方向に伝え合い、共有する作業のことを指す。リアルタイムでユーザーが文字入力と閲覧をし、複数のユーザーが同時に会話形式のコミュニケーションをとるものであったが、インターネットの帯域の拡大とpcの性能向上により、文字に限らず音声、映像など様々な情報を双方向にやりとりする同期的コミュニケーションツールである。ここでは文字を中心に使用するチャットをテキストチャット、音声も用いるものをボイスチャット、動画像を用いたものをライブチャットと定義する。

2.2. 課題発見型協調作業支援のための取り組み

2.2.1 協調作業とは

インターネットが発達する以前は協調作業は基本的に対面で行われるものであった。これは情報は人に付いて回るものだという前提条件があったためで、人々は対面で直接共同作業をすることで情報やモノをやり取りした。その後インターネット技術が発達していく中でインターネットを通して様々な情報のやりとりが可能になり、人は遠隔地にいることによる制約、時間的制約から解放された上での協調作業が可能になった。ここではインターネット上での協調作業モデルの概要を確認する。現在インターネット上での協調作業支援として使用されている支援環境の一つとしてグループウエアがある。これは1978年にPeter and Trudy Johnson-Lenzにより作られた単語である。Clarence Ellisの定義によれば、グループウエアとは共通の仕事や目的のために働く利用者のグループを支援し、共有作業環境のためのインターフェースを提供するインターネット上のシステムである[11]。また、これと同時によく使われる語としてCSCW(Computer-Supported Cooperative Work)という語がある。これは1984年にIrene GreifとPaul Cashmanによって作られたもので、グループワーク内でのコンピューターの役割に注目する概念や基本姿勢を表したもので、コンピュータ支援という支援手段と共同作業という支援対象の二つの異なる視点を合わせ持った概念である[12][13]。グループウェアは共同作業支援システムであり、CSCWとは技術的な事柄のみでなく、共同作業に関する組織的、社会的な事柄を包含した上位概念である。これは電子メールやデータベースを基盤とした、蓄積型の通信手段を用いた情報共有システムのみではなく、情報共有空間の提供や、その上でのコミュニケーション、調整、データ管理の支援を機能として持っている。グループウエアが対象としているグループは２つの概念の中間に位置するものである。ひとつは比較的構造の明確な定型的業務を対象に構築されるようなトップダウン的システム、もうひとつはパソコン上の個人用ツールである。グループウエアはこの中間に位置し、ダイナミックに形成される小規模なグループを対象としたシステムである[14]。グループウエアを整理、分類するときには時間（リアルタイム同期型、蓄積・非同期型)、空間（対面型、非

対面型）という二軸からなされることが多い [15]。リアルタイム同期型は複数のユーザーが音声や画像通信チャネル、あるいは共有ウィンドウや共有スクリーンを介して同時に作業を行うタイプのシステムである。蓄積・非同期型は電子メールや電子掲示板のような蓄積（非同期）型通信を基本としたシステムである。また対面型は複数のユーザーが会議室等に集合して対面で使用するタイプ、遠隔分散型は地理的に分散した複数ユーザーがリモート通信機能を用いて使用するタイプである。このようにグループウエアの種類は多岐にわたり、それを応用した協調作業モデルも多岐に渡る。

2.2.2 協調作業が取り組む課題の種類

協調作業が取り組む課題の種類

プロジェクトにおける協調作業が扱う課題に関して議論する。協調作業が取り組む課題には以下の課題が考えられる。

- 課題解決型協調作業
  課題解決型協調作業は明確になっているタスクをコミュニケーションを取りながら解決する作業である。例えば「〜のプログラミングをすること」「〜のイラストを書くこと」「〜に関して調べること」などの具体的な作業を、必要があればチームメンバーとコミュニケーションをとりながら解決していく作業である。

- 懸案事項に関するタスク明確化の作業
  懸案事項に関するタスクを明確化するための作業とは、その議題に付随するタスクは明確になっていないが、議論する必要がある懸案事項について議論する中で、その懸案事項を解決するためのタスクを明確にするものである。これは「〜の HP のインターフェースどうしようか」「来週のミーティングどうしようか」などの抽象度の高い議題である。これらの懸案事項に関して議論する中で、その懸案事項に付随する、「〜の部分を php で実装する」「〜の UI をフォトショップでモックアップで作る」などの具体的なタスクが明確化され、前述の課題解決型協調作業が行われる。

- 懸案事項の細分化、新しい視点を持った懸案事項の生成
  タスクの明確化の作業の中では、新しい懸案事項が出現することがある。これは「〜の HP のインターフェースどうしようか」という懸案事項を、「〜の HP の、ヘッダーの部分のデザインをどうしようか」などの懸案事項にさらに細分化する作業であったり、「そもそも〜はこうしたらどうだろうか」、「ふと〜と思った」などの新しい視点を持った懸案事項を生成する作業である。つまり、とある懸案事項に関する議論はふたつの種類の成果物を生成する。ひとつはその懸案事項を解決するための具体的なタスク、もうひとつはその懸案事項を細分化した、いわばその懸案事項の下部に属する新しい懸案事項、もしくは新しい視点を持った懸案事項である。これら、懸案事項から新しいタスクを明確化する作業、懸案事項からさらに細分化された、もしくは新しい視点を持った懸案事項を生成する作業は課題発見型の作業ということができる。

このように、懸案事項には、議論を通じてそれを解決するための具体的なタスク、その懸案事項を細分化した懸案事項という成果物が生成される。そしてそのようにして生成されたタスクを実行、また細分化された懸案事項に関してはそれに関してまた議論する、という形でプロジェクトは進行する。

協調作業における課題発見型作業

プロジェクトでは、扱う課題ごと、課題における局面ごと、協調作業が使用する協調作業モデルごとによって必要とされる協調作業が異なる。そのなかでも、前日したように、解決すべき課題自体の発見をするための協調作業、すなわち課題発見型の協調作業が必要となる場面がある。ここでは、課題発見型の協調作業を要求する課題、課題における局面、協調作業モデルについて議論し、プロジェクトにおける課題発見型協調作業について明らかにする。

- 課題発見型協調作業を要求する議題
  プロジェクトにおける、解決策が提示されるまで問題だと認識されない、正しい解が存在しない、一度しか取り組むことのできない、既存の解決策が

存在しない、などの特徴を持った、未解明なことや予測不能な要素が多い課題がある。これは課題発見型の協調作業を要求する議題である。

- 課題発見型協調作業が要求される局面
  課題発見型協調作業を要求する課題において、課題発見型協調作業が必要となる局面について議論する。こういった課題を取り扱うプロジェクトは、始めるにあたっての前提条件や既存の決定事項が少ない。そのため、課題解決の作業以外に、そのつど何を解決するかの課題自体を明確にする局面が必要となる。課題を明確にするための局面とはすなわち、開発における課題を適切な部分課題へと分割する作業、課題の解決方法を考える作業、そして実際にその解決方法が有効であったかを確かめるために解決方法に従い作業した結果の考察に基づき、次の課題を発見する作業のことである。[16]

- 課題発見型協調作業に適応した作業モデル
  最後に、こういった課題解決の方法を実現する協調作業モデルはアジャイル型開発モデルと呼ばれる [17]。これはウォーターフォール型開発モデルと対比される開発モデルである。ウォーターフォール型開発ではプロジェクトに付随する作業は事前に明確になっている。また各局面での成果物は完全であることが前提とされており、工程の成果物に見直しが入ることを原則前提としていない。それに対しアジャイル型開発は変更への迅速な対応を優先させ、短期間の開発工程を反復させることで開発の初期の段階からユーザーが実際に動くシステムを目でで確認できるという長所を持つものである。

このように、事前の決定事項の少ない不明確な課題におけるアジャイル型開発において、課題発見のための協調作業が存在する。

2.2.3 協調作業を支援するツール

協調作業における環境の変化

インターネットを使った協調作業が普及している背景として、コンピューターの高性能化と帯域が十分に確保されたインターネットの普及がある [?]。インターネットの普及は地理的に分散した作業者同士のインターネットを介したコミュニケーションを実現した。

またそれにより、単なるコミュニケーションのみならず、作業者同士が協調して課題解決を行う機会が増加した。これにより人は遠隔地にいることによる地理的制約、時間的制約から解放された上での協調作業が可能になった [?]。

協調作業支援のためのツール群

協調作業は様々なツールによって支援されてきた。ここでは既存のグループウエア等に実装されてきた協調作業支援ツールを大きく分類する。ツールを分析するに当たり、以下にあげる４つの要素を中心に分析する。その４つはやり取りされる情報の形式、情報を共有する対象、情報のやりとりが行われるタイミング、やり取りされた情報の事後的な参照方法である。また、それぞれのツールに関して、課題発見型協調作業に最適化された情報の最適化はどういった形式を取るのか、人がなにか行動を起こすときの脳内での思考過程を、外部から情報を受ける、情報を解釈する、それをうけて何をするか考える、実際に行動するという４段階のモデルとした上で考察する。なぜなら、課題発見型協調作業のためのアジャイル開発の最大の特徴はプロトタイプ制作とその評価、対応を非常に短い周期で繰り返すことにあり、そのサイクルの短さ、繰り返し回数の多さはすなわちよりよい成果物を生み出すと考えられるためである。そのため、これを先の人の行動の脳内の４段階モデルに当てはめると、そのサイクルを短くすること、サイクルを多く繰り返すことがより効率的な課題発見型協調作業を行う際に求められると考えるためである。ここではこれに則って先にあげた既存の協調作業支援ツールを分析し、それらが実現してきた協調作業モデルを考察する。

1. 電子メール

- 電子メールとは
  電子メールはインターネットを使った非同期的コミュニケーションツールのひとつである。インターネットを通じて、メールアドレスを知っている特定のユーザー同士が文字や画像、ファイルを送受信することができる。
- 情報形式
  メールでやり取りされる情報は基本的に文字が中心であるが、参照URLを添付やファイルの添付ができる。また、内容としては大きく二つに分かれる。その2つとは報告、要求である。
- 対象
  メールが送信される対象は、一般に公開されているメーリングリストなどの特殊な例を除けば、チームメンバー全体もしくはチームの中の誰かである。しかしメールは転送できるため、受信者が元の送信者の知らない第三者に転送することも可能である。メールにおいて起きがちな問題としては、外部要因や誰かの成果報告がチーム内の特定の誰かを対象にせず、報告や提案という形で全体に送信された場合、全員がそのメールの送信相手が自分ではないと考えてしまうことで情報が誰の入力にもならずに終わってしまうことである。
- タイミング
  非同期型である。送信者が送ったメールは直ちに受信者がわのサーバーへ届くが、受信者がそのメールを確認するのはいつになるかわからない。
- 情報の参照方法
  受け取ったメールは基本的に受信日時の新しいものから順に保管される。差出人、件名などを元にソートしたり、後からわかるようにチェックマークなどをつけておくこともできる。
- 考察
  ネットワーク上での協調作業支援に欠かせないツールとして電子メールがある。協調作業においてはメーリングリストという形が最も使用されており、これは協調作業メンバー全員にメールの一斉送信を可能

にする。ここでは進捗状況の報告、質問、リマインド、外部情報の共有などが行われる。メーリングリストに送信されたメールという成果報告の形は、先の脳内モデルと照らし合わせた考えると受け手の実際の行動に結びつくまでに、受け手が情報を受ける、解釈する、何をするか考える、というス テップを踏まなければならないために、アジャイルな開発手法を用いた協調作業においては効率のいい情報共有ツールとは言いがたい。しかし、非同期的な情報共有ツールとしては、名前、日時、件名、キーワード検索など様々な形での検索をかけることができるため有用なツールであると言える。

2. チャット

- チャットとは
  チャットはインターネット上の同期的コミュニケーション手段のひとつである。リアルタイムでユーザーが文字入力と閲覧をし、複数のユーザーが同時に会話形式のコミュニケーションをとるものであったが、インターネットの帯域の拡大と pc のマシンパワーの増大により、文字に限らず音声、映像など様々な情報を双方向にやりとりする同期的コミュニケーションツールである。ここでは文字を中心に使用するチャットをテキストチャット、音声も用いるものをボイスチャット、動画像を用いたものをライブチャットと呼ぶことにする。
- 情報形式
  チャットでやり取りされる情報は文字が中心ではあるが、参照 URL を添付したり、 ファイルを添付することができる。メールと違い、同期的なコミュニケーションツールであるため、何かのトピックをもとに会話を始めるのこそ誰かひとりのメンバーたが、そのトピックに関して会話が生まれ、一方通行な情報共有に比べてそのトピックについて深い理解を共有できるとともに、その会話の中からまったく新しい発見などの創造が生まれる可能性がある。
- 対象

チャットでのコミュニケーションの対象はチームメンバー全体もしくはチームメンバーの誰かである。会話に新しいメンバーを呼ぶこともできるが、新たに呼ばれたメンバーはそれまでにやり取りされた会話のログを見ることができない。

- タイミング
  チャットはリアルタイムでのコミュニケーションツールであるため、同期型である。

- 情報の参照方法
  チャットでのやりとりのログは発言時間が新しい発言順に並ぶ。ウィンドウは会話の対象者ごとに開くが、そのなかでの会話は単に発言の時系列順に並んでいるだけであるため、日時や題名などによるソートができず、過去の特定のやりとりを探すには時間と手間がかかる。また、扱われているトピックは同じでも、会話に参加しているメンバーが違うと別のウィンドウでのやりとりになるため、一本化して確認することが難しい。そして会話のログという形式での情報ログであるため、そこで行われた会話の要点の一覧性は低い。

- 考察
  チャットは同期的協調作業において最もよく使われているツールである。文字や、音声、映像を使った同期的なコミュニケーションが可能で、グループでの同時チャット、音声、映像通話も可能である。チャットでのやりとりは同期的であるために、情報の受け手は確実に情報を受け、また、やりとりの中で 情報の解釈が可能である。また、会話の中で次にとるべき行動が明確になることもあるため、情報の受け手は相手からの入力を受けてすぐに実際の行動に取り組める可能性が高い。しかし過去にやり取りされた情報の一覧性が低いことや、グループでチャットでは全員が出席していない状態でも先に進めてしまうために、自分が同期状態にない状態でも会話が進んでしまう問題などが考えられる。そのため、チャットは 同期的な作業には力を発揮するが、非同期的な使用法には難がある。

3. マイクロブログ

- マイクロブログとは
  マイクロブログとはインターネットを使ったコミュニケーションツールのひとつで、同期的なコミュニケーションツールとしての側面と非同期的なコミュニケーションツールとしての側面を併せ持ったツールである。かつてブログは投稿者の体験等を日記のような形で私的に投稿するものから、ニュースなどをまとめ議論の場とするものなどが中心であった。これらはある程度まとまった量での投稿が主で非同期的なコミュニケーションの場であった。それに比べマイクロブログでは投稿する文章の文字数に上限を設けられるなど投稿内容が短い。そのため普通のブログに比べて更新が簡単で、結果的にほぼ同期的なコミュニケーションツールとしても機能する。2006 年にサービスを開始した Twitter の普及などにより、近年認知度が高くなったツールである。
- 情報形式
  マイクロブログでやり取りされる情報は文字、参照 URL、添付画像等である。
- 対象
  自分の知り合いが主であるが、情報を受けた人間が転送という形で簡単に自分の知り合い全体に情報を拡散することができるため、発信された情報は他のメディアに比べて広く拡散する。
- タイミング
  非同期的である。しかし一度発信された情報でも、転送された場合にはまた新しく受信側の情報の一番上に表示される。であるので非同期型ではあるが複数回に渡って受信されるタイミングは考えられる。
- 情報の参照方法
  マイクロブログのタイムラインに乗っている情報は題名や受信日時などによってソートすることができず、単語検索やお気に入りのチェックを付けたものを一覧することしかできないので事後的な情報の確認

性は低い。

- 考察
  マイクロブログは協調作業においては不特定多数への呼びかけを目的に使用され、グループ内だけでなくグループ外の人的資源に接触できるという特性を持っている。チームに欠けている知識、ノウハウなどが明確であるが、それを補完できる資源の存在の在り処が不明であるときにマイクロブログ上で情報を流すと、それを見た誰かがその知識を補完してくれそうな第三者に転送という形で情報を回すことが可能である。しかしここで共有される文字ベースの情報は受け手がすぐに行動を想起できるような情報ではないため、その受け手にとっては、情報を受取る、解釈する、その上で何をする、というプロセスを踏む必要がある。であるのでこれはグループ以外の人的資源への呼びかけ、広報の第一段階としては有効であるが、その効果の確実性、操作性には改善の余地がある。

4. 掲示板

- 掲示板とは
  電子掲示板はインターネットを使った非同期的コミュニケーションツールのひとつである。特定の話題に関する書き込みはスレッド、もしくはトピックという名前でそれぞれの話題ごとにまとめられ、ユーザーはそれぞれのスレッドやトピックに沿った内容の書き込みをする。個人的に管理される小規模な掲示板から、２ちゃんねるなどの大規模な掲示板までその種類と規模は様々である。
- 情報形式
  グループ内で運用される掲示板運用はメーリングリストに似ており、でやり取りされる情報は基本的に文字が中心であるが、参照 URL を添付したり、ファイルを添付したりすることができる。また、内容としては大きく二つに分かれる。その 2 つとは報告、要求である。投稿者のさまざまな考えを文字や画像などの形式知に落とした形での情報

の投稿になるため、送信者の考えが完全な形で共有されるには至らない。掲示板は自動的に通知が来るメールと違い、アクセスしないと情報を確認できず、また、非同期的な情報共有ツールであることから、そこで頻繁なやりとりが行われづらいため、暗黙知を形式知化したときに抜け落ちる情報を補完することは難しい。

- 対象
  掲示板へのアクセスが許可されているメンバー
- タイミング
  非同期である。投稿が行われた際にメンバーにメールでの通知が送信される。メールを受けたメンバーは記載されている URL から掲示板にアクセスする。
- 情報の参照方法
  掲示板はそれぞれトピックにわかれた情報が記載されているため、事後的な情報の参照性は高い。

2.3. テキストチャットによる課題発見型協調作業

本研究では不明確で予想不可能な要素の多い課題における課題発見型作業のためのテキストチャットの提案と評価を行う。ここではテキストチャットの課題発見型協調作業における優位性について議論する。

2.3.1 課題発見型協調作業におけるテキストチャットの優位性

課題発見型協調作業における同期的コミュニケーションの優位性

不明確で予測不可能な要素の多い課題を解決するためには、柔軟なタスク明確化の作業が必要であり、これには創発的なコミュニケーションが欠かせない [18]。これを実現するのは対面での会話や、インターネット上のツールとしてはテキストチャット、もしくはビデオチャットなどのツールによって実現される同期的な

コミュニケーションである [19]。創発的なコミュニケーションとは、情報交換を目的とはしているが、目的指向型の対話とは異なり、ある発話が現在の文脈の中でどれくらい関連があるか、発話者自身もよくわからない対話を通してその議論の文脈を明確にしていく作業である。このようなコミュニケーションを実現するためには相手の割り込みに対して素早く反応したり、自ら相手に確認するなど相手とのやり取りを維持すること、また相手の理解度にあわせて、対話途中でも説明の戦略を適切に変更できる必要がある。これは対話という行為の持つ、即興性と発話の交換を通してお互いに持っている考え以上のものを作り出すという創造性が必要とされる作業である。この対話は「質疑応答の連続」とは異なり、明示的な質問は見当たらず、非常に短い単位での発話の交代や、発話の重複がみられる。また、その対話はその場の思いつきのみで進行しているように見えるが、実は対話の中で次々と新しい視点が導入されており、話者はお互いに自分の主張を明確にしようとするよりはむしろ両者が強調して「場の意見」をつくろうとするものである。この対話においては、お互いがうまく自分の意見を言い合ったほうが、一方的に意見をまとめてしまうメンバーがいるよりもより多様で新しい観点が生まれる。対等な協調が行われれば多様で新しい観点が生まれる [20]。

同期的コミュニケーションにおけるテキストチャットの優位性

前述したように、チャットにはテキストチャット、ボイスチャット、ライブチャットなど様々な種類があるが、ここではテキストチャットに注目する。以下に理由を述べる。

1. デモグラフィの制限が容易：
   デモグラフィとはその人の属性を表す。テキストチャットは相手のデモグラフィを制限することが出来る。すなわち、相手の姿形や声色等が見えない、聞こえないことによって相手の性格、立場といった個人情報を制限することが可能である。

2. 非言語情報の欠落：
   表情・身振り・視線といった言語で表せないコミュニケーション情報を送

受することは困難である。

3. 外部への漏洩がない：
   文字メディアはコミュニケーションをとっている対象以外の人間に対してはその内容を漏らさない。

この特徴は電子メールなどの非リアルタイムコミュニケーションにおける特徴であるが、チャットのようなリアルタイム性を持つ場合でも当てはまると考えられる。これらの特徴から、次の特性が存在すると考えられる。

- (1) と (3) より、チャットにおいては相手の立場や周囲の目を気にすることなくコミュニケーションを取ることが可能になる。したがって自由な発話が許されることになり、このことが発想の増大に寄与していると考えられる。
- (1) と (2) より、相手の感情や周囲の雰囲気、あるいは立場など、発送作業と直接関係のない情報が入りにくい。したがって、チャットを用いた場合では外部からの情報に邪魔されることのないぶん作業に集中することができる。ただし、その分微妙なニュアンスは伝わりにくい。

これらの理由により、コミュニケーション自体が目的のコミュニケーションではなく、議題に沿った会話をすることで議題にまつわる事項を決定していくようなコミュニケーションではテキストチャットに優位性があると考えられる。

2.4. テキストチャットを使った課題発見型協調作業の現状分析と問題点

ここではプロジェクトにおけるテキストチャットを使った課題発見型協調作業の現状分析と問題点について議論する。プロジェクトにおいて使用されるテキストチャットで、特定の使用目的に特化していない汎用的なサービスとしては Skype、MSN メッセンジャー、IRC チャット、ichat などが挙げられる。また、課題発見

型、課題解決型のプロジェクトでの使用に特化したテキストチャットサービスとしては chatwork などがある。以下にそれぞれのサービスの分析を行う。

2.4.1 既存のテキストチャットシステムの種類と特徴

- Skype
  Skype は P2P 技術を使ったインターネット電話サービスのひとつである。Skype ではユーザー同士の無制限の音声チャット、ビデオチャット、テキストチャット（インスタントメッセンジャー）が可能である。これらの機能は１体１での通信から、最大２５人までの同時通信も可能である。これらのチャットには任意の題名をつけることができる。またチャット画面にファイルをドラッグアンドドロップすることで通信中の相手にファイルを送信することも可能である。ユーザーは Skype アプリケーションを PC もしくは mac 等のコンピューター上にダウンロードし、ユーザー名とパスワードを設定し、サービスにログインする。Skype ではユーザーのアイコンが変化することによって相手がオンランかどうかが分かる機能があるが、これは友達リクエストを認証したユーザー同士である必要がある。ユーザー同士はお互いのユーザー名を使って相手を検索することができ、探している相手が見つかれば相手に友達リクエストを送信する。リクエストを受け取った側はリクエストを承認もしくは無視することができ、承認した場合にはお互いに友達関係となる。図 2.1 に skype の画面を乗せる。

- MSN メッセンジャー
  MSN メッセンジャーは P2P 技術を使ったインターネット電話サービスの一つである。MSN メッセンジャーではユーザー同士の音声チャット、ビデオチャット、テキストチャットが可能である。これらの機能は１体１での通信から、最大２５人までの同時通信も可能である。またチャット画面にファイルをドラッグアンドドロップすることで通信中の相手にファイルを送信することも可能である。ユーザーは PC もしくは mac 等のコンピュータ上にアプリケーションをダウンロードし、MSN アカウントを取得してサービス

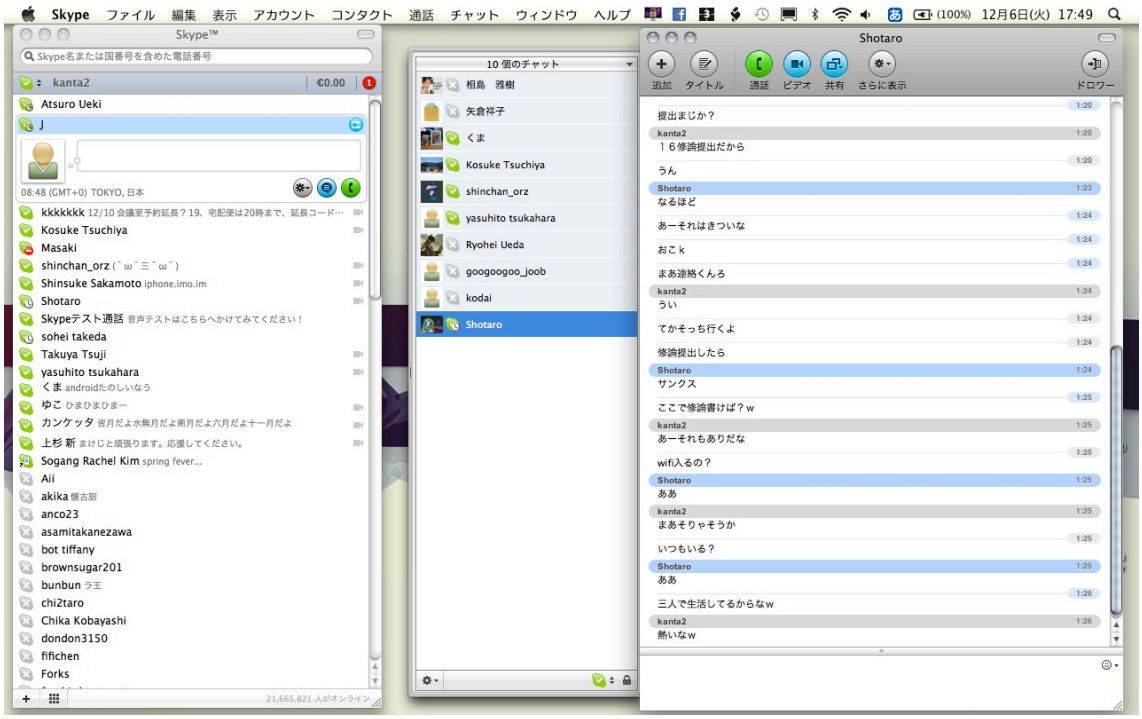

Skype ファイル 編集 表示 アカウント コンタクト 通話 チャット ウィンドウ ヘルプ
(100%) 12月6日(火) 17:49
Skype™
Skype名または国番号を含めた電話番号
kanta2
€0.00
Atsuro Ueki
J
08:48 (GMT+0) TOKYO, 日本
kkkkkkk 12/10 会議室予約延長？19、宅配便は20時まで、延長コード…
Kosuke Tsuchiya
Masaki
shinchan_orz (^ω^三^ω^)
Shinsuke Sakamoto iphone.imo.im
Shotaro
Skypeテスト通話 音声テストはこちらへかけてみてください！
sohei takeda
Takuya Tsuji
yasuhito tsukahara
くま androidたのしいなう
ゆこ ひまひまひまー
カンケッタ 皆月だよ水無月だよ南月だよ六月だよ十一月だよ
上杉 新 まけじと頑張ります。応援してください。
Sogang Rachel Kim spring fever...
Ali
akika 懐古厨
anco23
asamitakanezawa
bot tiffany
brownsugar201
bunbun ラ王
chi2taro
Chika Kobayashi
dondon3150
fifichen
Forks
21,665,821 人がオンライン
10 個のチャット
相島 雅樹
矢倉祥子
くま
Kosuke Tsuchiya
shinchan_orz
yasuhito tsukahara
Ryohei Ueda
googoogoo_joob
kodai
Shotaro
Shotaro
追加 タイトル 通話 ビデオ 共有 さらに表示 ドロワー
提出まじか？
kanta2
１６修論提出だから
うん
Shotaro
なるほど
あーそれはきついな
おこｋ
まあ連絡くんろ
kanta2
うい
てかそっち行くよ
修論提出したら
Shotaro
サンクス
ここで修論書けば？ｗ
kanta2
あーそれもありだな
wifi入るの？
Shotaro
ああ
kanta2
まあそりゃそうか
いつもいる？
Shotaro
ああ
三人で生活してるからなｗ
kanta2
熱いなｗ

図 2.1 Skype 画面

にログインする。アプリケーションが使えない環境にいるユーザーに対してはウェブブラウザ上で起動する MSNweb メッセンジャーというサービスも存在する。また、表示されたウィンドウに何か記入すると相手のウィンドウにもそれが反映されるホワイトボードという機能や、写真を共有する MSN フォト共有やカレンダーを共有する MSN カレンダー共有などの機能もある。図 2.2 に MSN メッセンジャーの画面を乗せる。

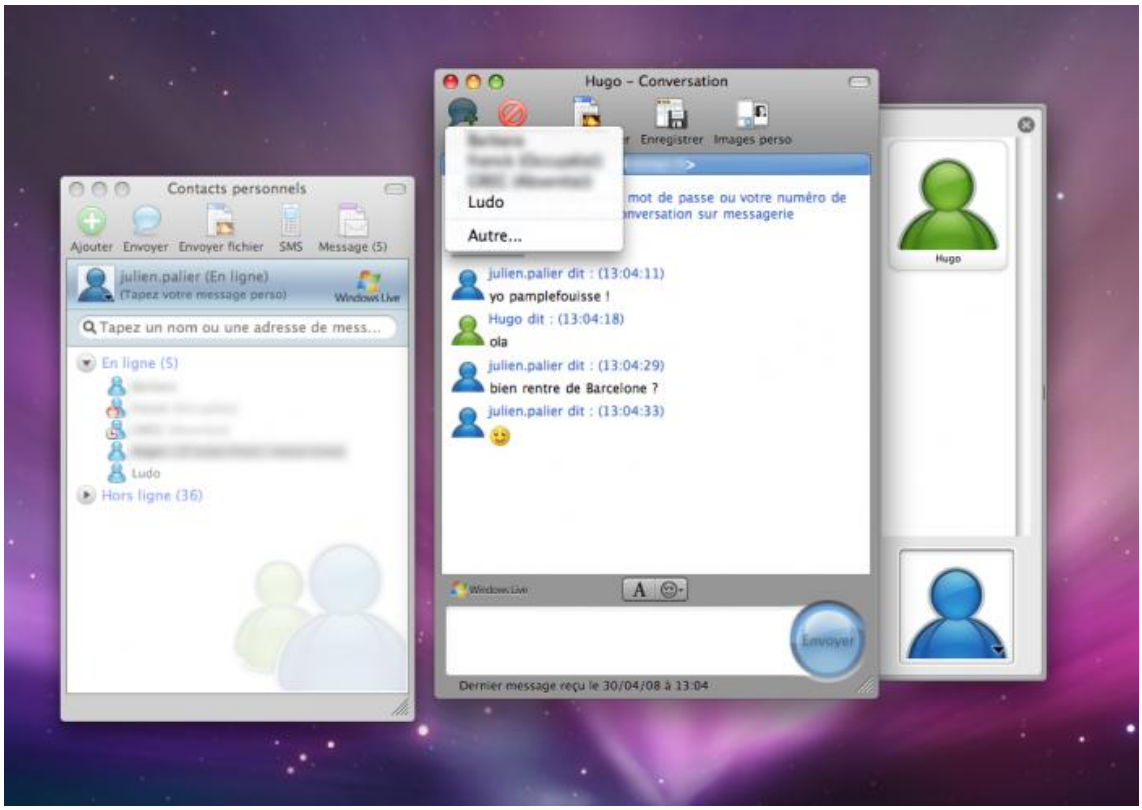


図 2.2 microsoft messenger 画面

- LimeChat（IRC チャット）
  IRC はインターネットを使ったテキストチャット専用のシステムである。ユーザーは IRC クライアントをダウンロードし、IRC サーバーに接続する。ユーザーはニックネームを設定し、接続したい IRC サーバーのアドレスを記述

する。そして入室したいチャンネルの名前を記述することでチャンネルに入室する。そこでIRCサーバーに接続しているユーザー同士でテキストチャットをすることができる。また、サーバー同士を接続して他のサーバーに接続しているユーザーともテキストチャットをすることも可能である。図2.3にIRCチャットの画面を乗せる。

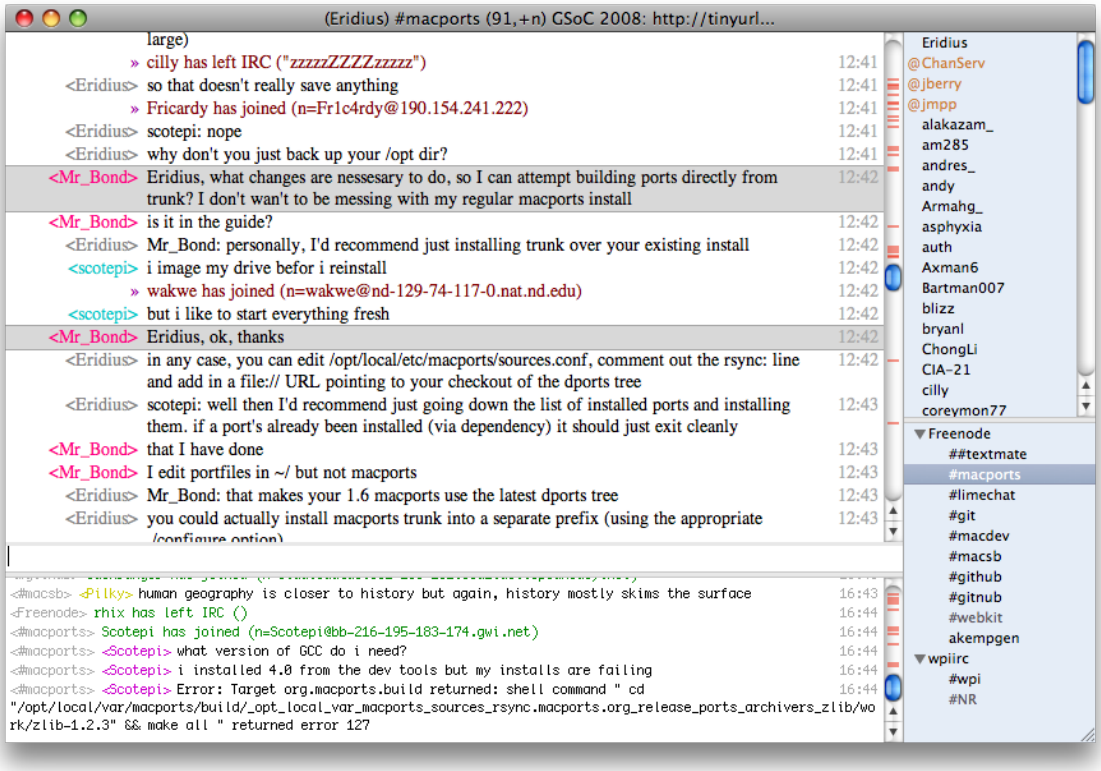


図 2.3 IRC画面

- ichat

  ichatはMacOS Xに標準搭載されているテキストチャットシステムである。ichatは音声チャットならば同時最大１０人と、ビデオチャットであれば同時最大４人と通話ができる。ユーザーはMobileMeアカウントなどを取得し、アカウントとパスワードをichatに登録する。ichatを起動すると自動的にユーザーはオンライン状態になり、ichat上で探している相手のアカウ

ント名を入力して検索し、相手とのチャットを開始する。図 2.4 に ichat の画面を乗せる。

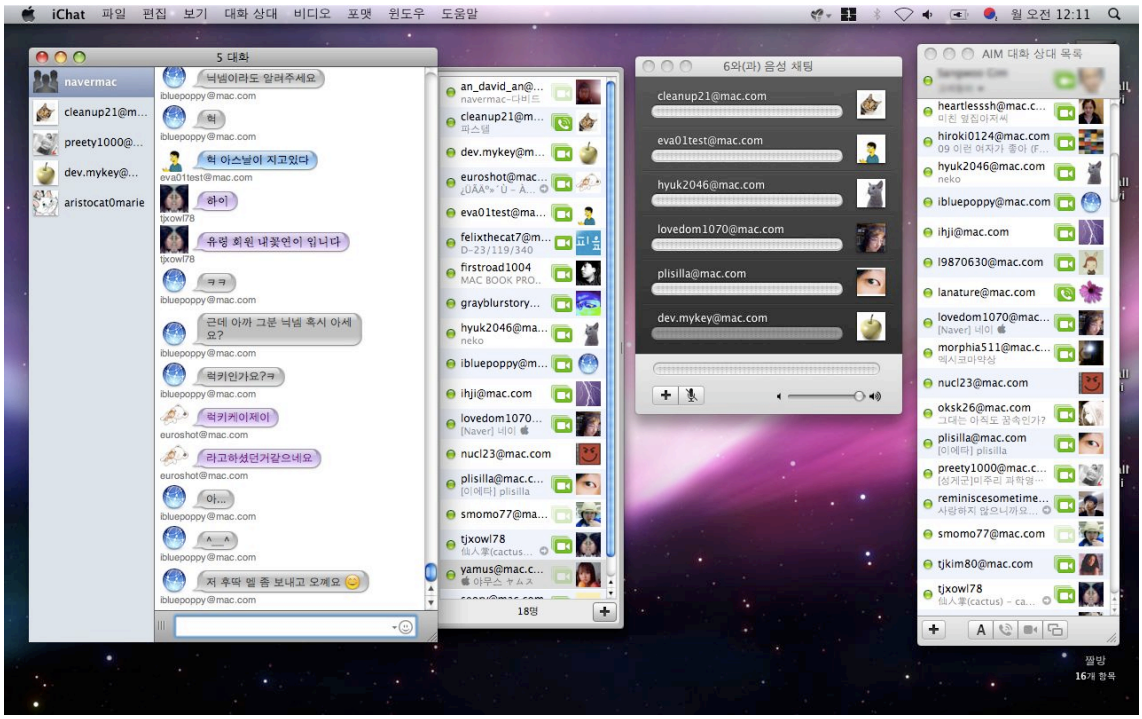


図 2.4 ichat 画面

2.4.2 テキストチャットを使った課題発見型協調作業の分析と問題点

これらのテキストチャットツールは同期的コミュニケーションそれ自体の円滑化を目的としているため、課題発見型協調作業に必要とされる同期的コミュニケーション以外の様々な方策はそれぞれのユーザーが各自で別個のアプリケーションを使用するなどして補う必要がある。

生成物 (新しい懸案事項等) 保持の困難さ

前述したように、課題発見型協調作業における、ひとつの懸案事項に対するテキストチャットではふたつの成果物が生成される。ひとつは懸案事項を解決するための具体的なタスク、もうひとつは細分化された、もしくは新しい視点を持った懸案事項である。人間は全てのタスクや懸案事項を逐次的に処理することができないため、新しい懸案事項や明確化されたタスクはどこかに保持し、後から参照できるようなしくみ等が必要である。また、これらの成果物は生成された瞬間に即時的にどこかに保持されないと、事後的なログの可読性が低いというテキストチャットの特性上無視されてしまう可能性が高い。事実、多くのユーザーはテキストチャットでの課題発見型協調作業で明確化されたタスクの管理には、ical や Google Calendar などの外部アプリケーションを起動して管理しているが、この方法ではタスクの管理はできるものの、創発的コミュニケーションに欠かせないコミュニケーションの即時性が損なわれてしまうため、理想的な管理方法とは言いがたい。また、課題発見型協調作業ではタスク以外にも、細分化された懸案事項や新たな視点を持った懸案事項が生成されるが、これはさらなる議論の余地がある懸案事項にもかかわらずタスクと混同して扱われる事が多い。これを解決する方法として、外部アプリケーションでそれを保持するという方法があるが、課題発見型協調作業が行われる Skype や MSN メッセンジャーなどの同期的テキストチャットツールは、創発的なコミュニケーションを担保する即時性が欠かせないため、外部アプリケーションでそれを保持するという方法はその即時性を失うという意味で理想的ではないと言える。

テキストチャットログの事後的な可読性の低さ

また、チャットやビデオチャットなどの同期的コミュニケーションシステムでは、そこで行われたやり取りの事後的な可読性は著しく低い。これはチャットやビデオ チャットなどの同期的コミュニケーションシステムでは、そのやりとりは論理的ではなく、非常に短い発言の連続や発言の重複があり、そこで明確化されるタスクに比べて会話の総量が莫大になるため、ログを見るだけではどこの部分で重要な発言がなされたのかが ひと目でわからないためである。したがって創

発的コミュニケーションは新しいタスク、新しい意見の明確化を実現するが、コミュニケーションを終えた段階でその内容がタスクとして明確化されずに忘れられてしまったり、たとえ明確化されても事後的な視認性が著しく低い可能性が考えられる。

2.5. ２章まとめ

第２章ではテキストチャットを使った課題発見型協調作業の現状と課題について議論した。まず既存のチャットシステムの分類をし、そして課題発見型協調作業支援の取り組みに関して議論した。次に協調作業が取り組む課題の種類には３つあることを確認した。その３つとはまず、明確化されているタスクを実行するための課題解決型協調作業、そして懸案事項を明確化するための課題発見型協調作業、最後に懸案事項を細分化するための課題発見型協調作業である。そしてその中の課題発見型協調作業に注目し、課題発見型協調作業を要求する議題の種類、プロジェクト内での局面、それを支援するワークモデルについて議論した。また、これらの協調作業は様々なツールに支援されており、次にそれらツールの分析と考察を行った。次にテキストチャットを使った課題発見型協調作業に関して議論した。課題発見型協調作業における同期的コミュニケーションの優位性と、その中でも特にテキストチャットの優位性について議論した。最後にテキストチャットを使った課題発見型協調作業の現状分析と問題点について議論した。現状として様々なテキストチャットシステムは同期的コミュニケーション自体を目的にしており、課題発見型協調作業に必要とされる様々な方策はユーザーの経験則によってそれぞれ実現されている。問題点としては、同期的コミュニケーションは即時的なやり取りがそれを成立させていることによる同期的コミュニケーションの成果物をその都度記録していくことの困難さ、かつ事後的なコミュニケーションログの可読性の低さが挙げられる。

第3章
BufferManの提案

3.1. BufferManの提案

3.1.1 想定する対象ユーザー

BufferManの想定ユーザーについて議論する。BufferManは遠隔地、または異なる時間帯、もしくはその両方で協調作業をするプロジェクトでの使用を想定している。テキストチャットを使った協調作業において、共同作業者が8人を超えると一部のメンバーがリーダーシップを取り問題解決を行う傾向がみられ、また発言の少ないメンバーに対しては相手の姿が見えないことに不安を感じグループ内での協調作業という感覚が低減する [21]。したがってBufferManでは共同作業者数の上限を8人とする。

3.1.2 BufferManが扱う協調作業

BufferManが扱う協調作業の種類について議論する。BufferManは課題解決型作業のみならず、課題発見型作業も内包する課題を扱うことを想定している。

3.1.3 課題発見型協調作業に必要とされるモデル

課題を解決するために、創発的協調作業のモデルを3つの側面から要件定義する。ひとつはやり取りされる情報の形式の最適化、ふたつめは情報を共有する相手の最適化、三つ目は情報を共有するタイミングの最適化である。

1. 情報の形式の最適化

情報の形式の最適化は、やり取りされる情報が受け手の実際の行動に反映されるため、またその時間をできるだけ短くするためのものである。ウォーターフォール型の開発である場合、ひとつの工程が終わった場合には次の行程が明確化されているので、ひとつの工程が終わった場合にはそれを共有し、他のメンバーの行程を確認した上で単に自分の次の工程に進めばいい。しかし、正解がない課題発見型協調作業に取り組む際には、自分の取り組むべき行程も事前には決まっておらず、他のメンバーの進捗や刻々と変わる問題の状況にその都度対応しながら、時には後戻りをしながら進めていかなければならない。ユーザーの画面上に表示された情報が実際に認識されるためには、目に入った情報が、その時点で瞬間的に理解できる、もしくはさらにある程度自分の取るべき行動が想像できるような情報でないと、認識されずに見過ごされてしまう可能性がある。しかし特にこのような課題発見型協調作業ためのチームでやり取りされる情報は複雑であるため、その解釈・認識とそこからの自分の課題設定は容易ではない。また、入力された情報をどういったタスクとして解決するか、という課題も課題発見型作業であり正解はなく、そこには「よりよい解」しか存在しない。したがってこの課題に対しても発案と評価を繰り返してよりよい行動を考える必要がある。これを実現するためには同期的協調作業と非同期的協調作業の複合の複合化が有効と考えられる。チャットのような同期的なコミュニケーションツールを使うことで半強制的に情報共有する側とされる側の相互的なコミュニケーションを発生させ、表示された情報は見えているが認識されていないというような状況を防ぐ。そして同期的なやりとりの中で受け手の入力に対する行動の発案と評価をやりとりの中で繰り返し、よりよいタスクを実現する。しかし既存の同期的協調作業支援ツールではそこで行われたやりとりのログ化が難しいため、そこに参加できなかったメンバーがそこでやりとりされた情報を後から参照するのが難しい。そこで非同期的なメールや掲示板という手段で同期的なコミュニケーションの内容を共有することが必要とされると考える。

2. 情報共有の相手の最適化

人には各人異なった能力、得意分野があり、それぞれが効果的に成果報告を出せる分野は異なる。それは各人がそれぞれそれまでに取り組んできた課題や経験則に基づくものである。この分野は目にする情報に対する反応にも深く関わりがあり、自分にとって馴染みが深かい情報は認識されやすい。例えばある情報を目にしたとき、それに関する関連情報や解決方法を知っていればその情報は認識されるが、知らなければ見過ごされてしまう。であるので同じ情報を複数の作業者に一律に与えたとき、それがちゃんと認識されるかどうかは人それぞれであり、また、認識された情報に対すしてタスクを考えつくかどうか、またそのタスクが実行されて成果報告されるかどうか、またそれにかかる時間や、その成果報告自体の質も人それぞれである。であるから、よりよい課題発見型協調作業を目指す場合、情報を誰かと共有するときにはその情報を受けて考えられる行動がより向いている相手に対して情報共有をすることが効果的である。例えば、誰かが作業をした結果、なにかのデザインをする必要性が出てきたとき、その情報はデザイン系統に関して得意分野をもっている相手に共有されたほうが、マネジメント分野に関して得意分野を持っている相手に共有されるよりも迅速で質の高い協調作業が期待できる。これに関しては、慶應大学大学院メディアデザイン研究科が、創発的社会の到来に備えて人的資源をデザイン、テクノロジー、マネジメント、ポリシーの４軸で捉え、ひとつのチーム内で複数人が各々の特性をもって協調することで４つの軸をチームとして実現し、創発的な協調作業を実現しようとしている [22]。このように情報を人の特性にあわせて共有する相手を柔軟に選ぶ取り組みは重要である。かつては他人の能力を把握する能力の強い人間の経験則にてこれを実現していたが、これはそのメンバーの実際に知っている相手にしか共有することができなかった。しかし、本来であれば実際に知り合いでなくても、時間的、空間的に離れた場所にいるまったくの他人でも、必要とされている能力特性とその人が持つ能力が適合すれば効率的な協調作業ができるはずである。

3. 情報共有のタイミングの最適化

仮に誰かの成果報告した情報と誰かの入力される情報の形式が適合し、潜在的には効率的な創発的協調作業の可能性がある状態でも、情報を受ける側すでになにかの行動に取り組んでいる場合、またはどういった行動をすべきか考えるなどしていた場合、あたらしい情報の入力は後回しにされ、実行されないまま放置されてしまう可能性が高い。これは誰かの作業の成果報告や外部要因の報告などが、送信側の送信したいタイミングで送られた場合、そのとき相手の脳内でのタスクが占有されているとそれは実行されず、効率的な創発的作業に不都合であるということである。であるので、誰かの作業の結果としての成果報告や、外部要因からの情報共有は、共有相手の脳内タスクがあいているとき、作業や思考が一段落したときに共有されると、その情報はちゃんと受信され、解釈され、そのあとの行動につながる可能性が高い。このようなチームメンバーの状況把握の重要性は Awareness として岡田、松下にも強調されている [23]。

3.1.4 課題発見型協調作業に必要とされる機能

即時的な情報蓄積機能の必要性

課題発見型協調作業に必要とされる同期的コミュニケーション以外の様々な方策はそれぞれのユーザーが各自で別個のアプリケーションを使用するなどして補う必要がある。２章でも述べたように、課題発見型協調作業における、ひとつの懸案事項に対するテキストチャットではふたつの成果物が生成される。ひとつは懸案事項を解決するための具体的なタスク、もうひとつは細分化された、もしくは新しい視点を持った懸案事項である。人間は全てのタスクや懸案事項を逐次的に処理することができないため、新しい懸案事項や明確化されたタスクはどこかに保持し、後から参照できるようなしくみ等が必要である。また、これらの成果物は生成された瞬間に即時的にどこかに保持されないと、事後的なログの可読性が低いというテキストチャットの特性上無視されてしまう可能性が高い。であるので、課題発見型の協調作業をするにおいて、テキストチャットはその内部に以下の機能を実装するべきである。それれの機能とは、テキストチャットによって

明確化されたタスクを格納する機能や、テキストチャット内で発見された新しい観点、懸案事項などを一時的に保持する機能である。保持されたタスクは実行されたかどうかがわかる必要がある。また、細分化された、もしくは新しい視点を持った懸案事項はそれをもとにさらなる議論が可能となる必要がある。どういった内容が懸案事項として保持され得るのか、また、どの情報が誰に共有される必要があるのかを調べるにあたり、本研究では Google Person Finder Bot 製作チーム [9] にフィールドワークを行い、そのコミュニケーションモデルを調査した。このフィールドワークについては後述する。

可読性の高い事後的な共有機能の必要性

２章における議論のように、チャットやビデオチャットなどの同期的コミュニケーションシステムでは、そこで行われたやり取りの事後的な可読性は著しく低い。しかしテキストチャットに参加していたメンバーは、内容の振り返りのためにその懸案事項に関連するコミュニケーションログを事後的に確認する必要がある。また、そのテキストチャットに参加できなかったメンバーは、事後的に短時間でわかりやすく自分が参加していなかった間のコミュニケーションログを確認できる必要がある。であるので、課題発見型の協調作業をするにおいて、テキストチャットはその内部に以下の機能を実装するべきである。その機能とは、テキストチャット内でやり取りされた情報を事後的に他のメンバーに共有する機能である。この機能では、ユーザーは議題となっている懸案事項ごとにテキストチャットのログを確認できる必要がある。これは事後的な、非同期的な情報共有のツールとして有効な電子メール等を活用する。

3.2. Google Person Finder Bot 制作チームへのフィールドワーク

3.2.1 フィールドワークの目的

課題発見型協調作業を行うにあたり、人間は逐次的に情報を処理しない。であるので、とある懸案事項に関する議論に関する同期的な会話で出てきた新しい懸案事項、明確化されたタスクは会話の最中に一時的に蓄積され、事後的に参照できる必要があるという仮説に関しては前述した。ここでは、どういった形で情報の一時的な蓄積が課題発見型協調作業において有効なのかを調べるために、フィールドワークを行った。フィールドワーク先としては Google Person Finder Bot 制作チームを対象とした。理由としては、Google Person Finder Bot 制作チームの活動モデルがが本研究が対象とする協調作業チームの活動モデルと似ていたからである。具体的にはチームの人数規模が最大１０名前後であること、取り組む課題が課題発見型の協調作業であること、迅速性が要求される短期間のプロジェクトであった。

3.2.2 Google Person Finder Bot とは

ここでは Google Person Finder Bot 制作チームにおける情報共有の方法を分析する。bot とはある特定の事象や行動に反応して動作を行う人工無脳である。２０１１年３月１１日に発生した東北地方太平洋沖地震の際、あるチームが被災地のために貢献しようと行動を始めた。被災という状況を受けての行動に正解はなく、活動は成果物を通しての事後的な判断しかできないのでウィキッドプロブレムの一つと言える。Google person finder は Google によって製作されたウェブサービスであり [8]、被災した方々の生存を確認することを目的としている。図 3.1 に Google Person Finder の画面を示す。これは被災して連絡が取れなくなっている被災地の方の名前、住所、電話番号等を入力するとその人の安否情報が確認できるというサービスである。しかし同時期マイクロブログの一つである Twitter 上にも被災者の安否情報が多く投稿されていたため、Google person finder（以下

GPF）上でやり取りされている情報と、Twitter 上でやり取りされている情報には重複が多く、安否確認の際に手間がかかったり、必要としている情報に到達できないなどの不具合が存在した。それをうけて、チームは GPF を元にしたマイクロブログ bot、Google Person Finder Bot を製作した [9]。図 3.2 に Google Person Finder Bot の画面を示す。GPF は製作過程においてマイクロブログからの公式化や Google とのやりとりの中で API 制限の解除などを受けながら、震災翌日には ver1 を公開するというスピード感のある協調作業を実現し、最終的に１５万件にのぼる安否確認のフィードをツイートした。

図 3.1　Google Person Finder

3.2.3　Google Person Finder Bot 制作チームの構成

ここでは Google Person Finder Bot 制作チームの構成について議論する。Google Person Finder Bot 制作チームは全部で１０名のメンバーから構成されており、その中で各メンバーは分業しながら協調作業をしていた。このメンバー間で、このプロジェクト以前に知り合い同士だったメンバーは少なく、プロジェクト以前に

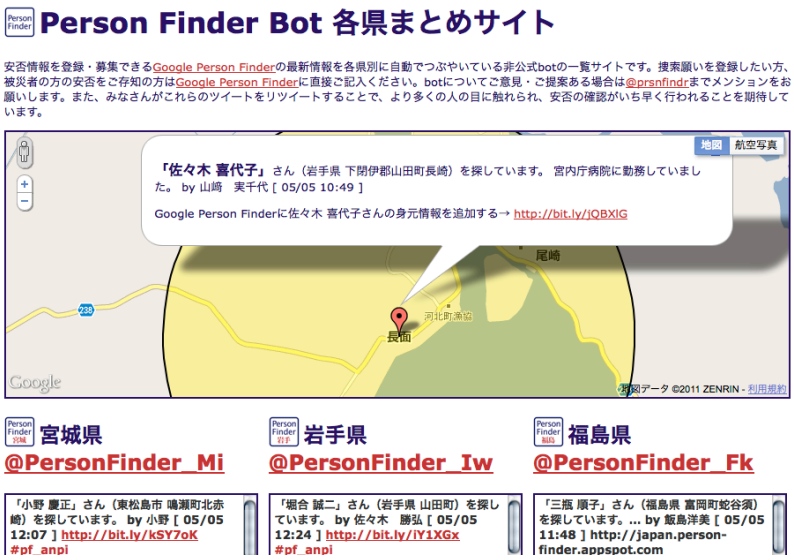


図 3.2　Google Person Finder Bot

全員と知り合いだったメンバーはひとりしかいなかった。また、プロジェクト終了後にもお互いを認識せずに終わったメンバー同士もいた。ここではプロジェクト以前に全員と知り合いだったそのメンバーをメンバー A と呼ぶ。このプロジェクトではメンバー A がそれぞれのメンバーのタスク管理や情報共有のハブとなっていた。その方法はメンバー A の経験則によるものであった。また、メンバー全員がプロジェクトの全容を把握していたわけではなく、その程度はメンバーごとに差があった。それぞれのメンバーは共有された情報や懸案事項をもとに、そのつど自発的にタスクを生成し、実行しては全体で共有していた。

3.3. Google Person Finder Bot 制作チームでのコミュニケーションモデル

図 3.3 に Google Person Finder Bot 制作チームでのコミュニケーションモデルを乗せる。この図において、紫の破線はひとつのタスクを表す。ここではどの情報がそのメンバーへの入力になって、その入力をもとにした議論によってどういっ

たタスクが明確化され、誰に共有されたかを表している。オレンジの字は行われたタスクを表す。黒い実線は明示的に共有相手を特定した情報共有であり、黒の点線は参加メンバー全員に共有された情報を表している。また、水色の実線は外部からの情報、状況変化の報告等を表す。

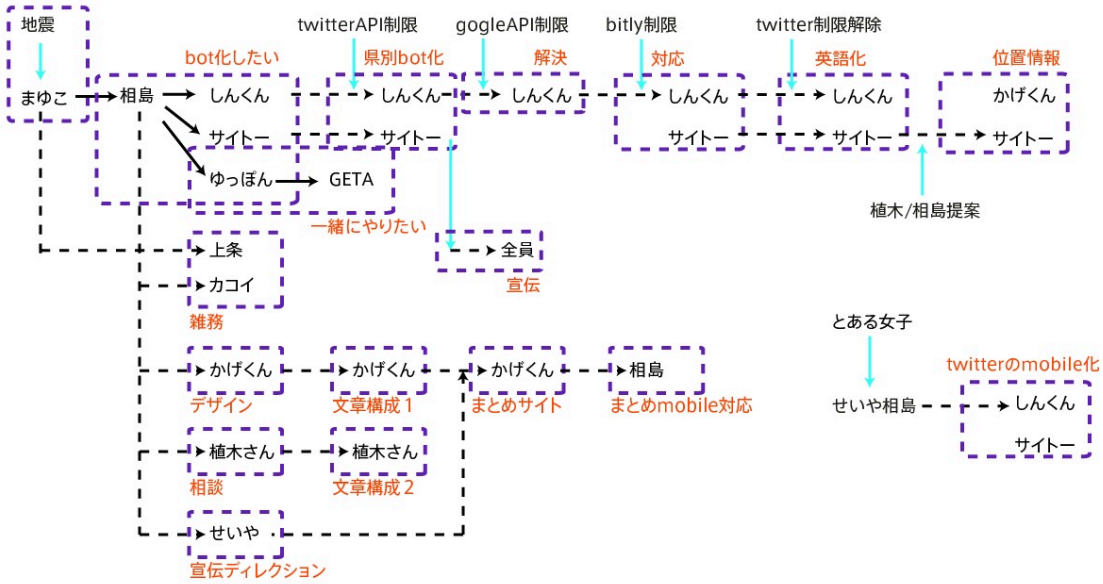


図 3.3　コミュニケーションモデル分析

3.3.1　情報共有の方法

チームは１０名のメンバーによって形成されていた。作業中は常にテキストチャットを立ち上げており、テキストチャットのウィンドウは全メンバーが参加しているものと、それぞれの作業班が少クラスタチームとして開いているものが複数個同時に開いているという状態だった。また、全員参加のメーリングリストも併用していた。

3.3.2 情報共有の内容

共有されている情報は大きく分けて 3 種類あった。ひとつは外部からの情報、状況変化等の報告。例えばこれは「google からの API 制限がかかった」などの情報である。もうひとつの種類はプロジェクトへの提案である。例えばこれは「Google Person Finder Bot を英語化したらどうだろうか？」などの提案である。そしてもうひとつの種類はなにかのタスクを終了した、という成果報告である。

3.3.3 情報を共有する対象

チームを結成する際のメンバー集めを主に務めたメンバーがいる。これは前述のメンバー A である。メンバー A は、Google Person Finder Bot を製作するにあたり、制作物のデザイン、画面構成などを考えるデザイン系のメンバー、実際に実装をするテクノロジー系のメンバー、チーム内外のマネジメントや対外活動、広報を行うマネジメント、ポリシー系のメンバーが必要と考え、それに則ってメンバーを招集した。例えば画面デザイン、文章校正などはデザイン系メンバーが担当し、bot の実装、マイクロブログや Google との技術的なやりとりはテクノロジー型のメンバーが担当し、bot の英語化や著名人への RT 依頼などはマネジメント、ポリシー系のメンバーが担当した。

3.3.4 情報を共有するタイミング

bot 製作中、あるメンバーがツイートされた安否情報の位置情報をもとに Google マップ上に情報を描画しようという提案を行った。しかし、そのタイミングでは全員のメンバーが他の作業に取り組んでおり、その情報は誰にもに認識されずに放置され、実際に安否情報が地図上に描画されたのは数日後であった。しかし別の時、メンバー B がメンバー A に bot の英語文の文章構成の改善を提案した際、メンバー A はその時点で他のメンバーが他の作業中であることが分かっていたために一時的にその提案を保持し、英語文章の校正が得意そうなメンバー C に、手

が空いているときを見計らってそのタスクの実行を打診したところ、メンバーCはすぐに英語文章の校正に取り組むことができた。

第4章
BufferManの設計・実装

4.1. BufferManの画面構成要素

ここではBufferManの画面構成用について議論する。BufferManはウェブブラウザで使用できるウェブアプリケーションである。開発環境はphp version5.3.3-7、mysql version5.1.49-3である。システムの概要を図4.1に示す。

4.1.1 トップ画面

ここではBufferManのトップ画面について議論する。BufferManのアカウントを持っておらず、このウェブページに始めてアクセスするユーザー、もしくは登録しているがサインアウトしている状態にあるユーザーにはこのページが表示される。「アカウントを作る」のボタンをクリックするとアカウント作成画面に移動する。既にアカウントを持っているユーザーは右上のフォームにユーザ名とパスワードを入れることでログインする。

4.1.2 アカウント作成画面

ここではBufferManのアカウント制作画面について議論する。BufferManはメールアドレス、ユーザー名、パスワード、Twitterアカウントを登録することで登録でき、使用することが出来るようになる。このページではそれぞれのフォームに対応した情報を入力することで新規ユーザーはBufferManに登録できる。

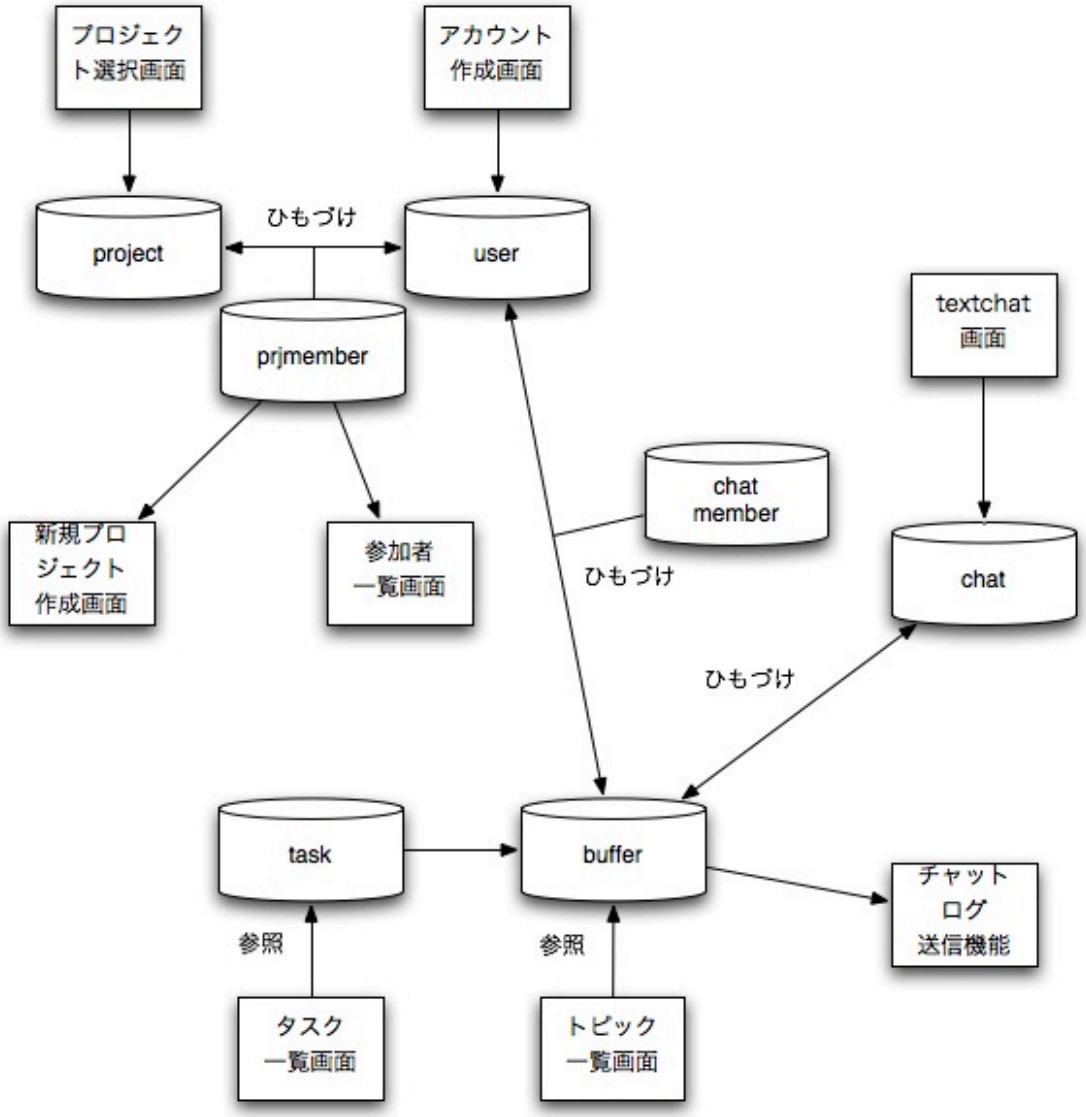

プロジェクト選択画面
アカウント作成画面
project
ひもづけ
user
prjmember
textchat画面
chat member
新規プロジェクト作成画面
参加者一覧画面
ひもづけ
chat
ひもづけ
task
buffer
チャットログ送信機能
参照
参照
タスク一覧画面
トピック一覧画面

図 4.1 システム概要図

4.1.3 新規プロジェクト作成画面

ここではBufferManの新規プロジェクト作成画面について議論する。BufferManでは登録されているプロジェクトに参加する、もしくは新しくプロジェクトを作成することでそこに参加する。この画面は新たな課題発見型プロジェクトについて新しいプロジェクトを製作する画面である。これは個々の懸案事項やタスクではなく、プロジェクトそれ自体をタイトルとする。例えば、「〜日のイベント」「〜の開発」などのタイトルがつけられる。プロジェクトのタイトルとパスワードをそれぞれ対応するフォームに入力することで新しいプロジェクトが作成される。

4.1.4 管理画面

ここではBufferManの管理画面について議論する。BufferManでは登録されているプロジェクトに参加する、もしくは新しくプロジェクトを作成することでそこに参加する。この画面では既存の課題発見型プロジェクトを検索することが出来る。また、目当てのプロジェクトが見つかった場合は、パスワードを入力することでそのプロジェクトに参加することが出来る。

4.1.5 テキストチャット画面

ここではBufferManのテキストチャット画面について議論する。ユーザーは協調作業の大半をこの画面で行う。図4.2にテキストチャットの画面を示す。

プロジェクト選択画面

ここではプロジェクト選択画面について議論する。テキストチャット画面内のプロジェクト選択画面では、参加しているプロジェクトをタブの中から選ぶことによって画面を切り替えることが出来る。プロジェクト選択画面のタブ内に表示されるプロジェクトは、プロジェクト検索によって検索した、もしくはプロジェクト作成画面で自ら作成したものであり、ユーザーが参加しているものである。タブの中から参加しているプロジェクトを変更すると、参加者一覧画面、トピッ

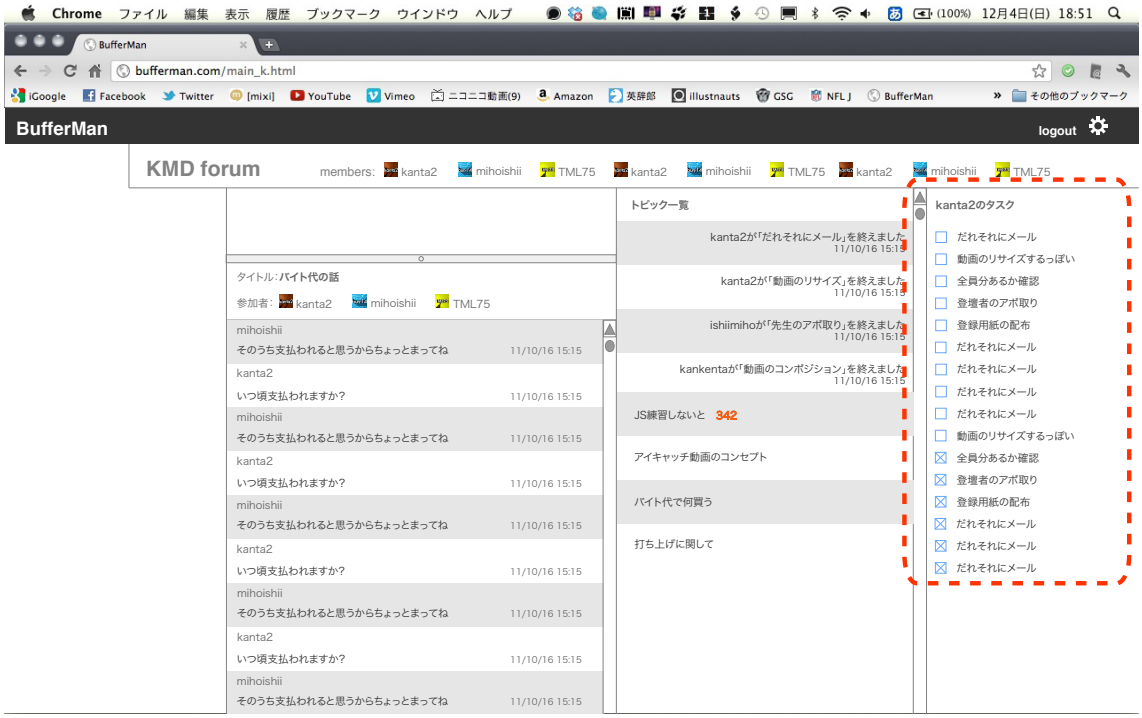
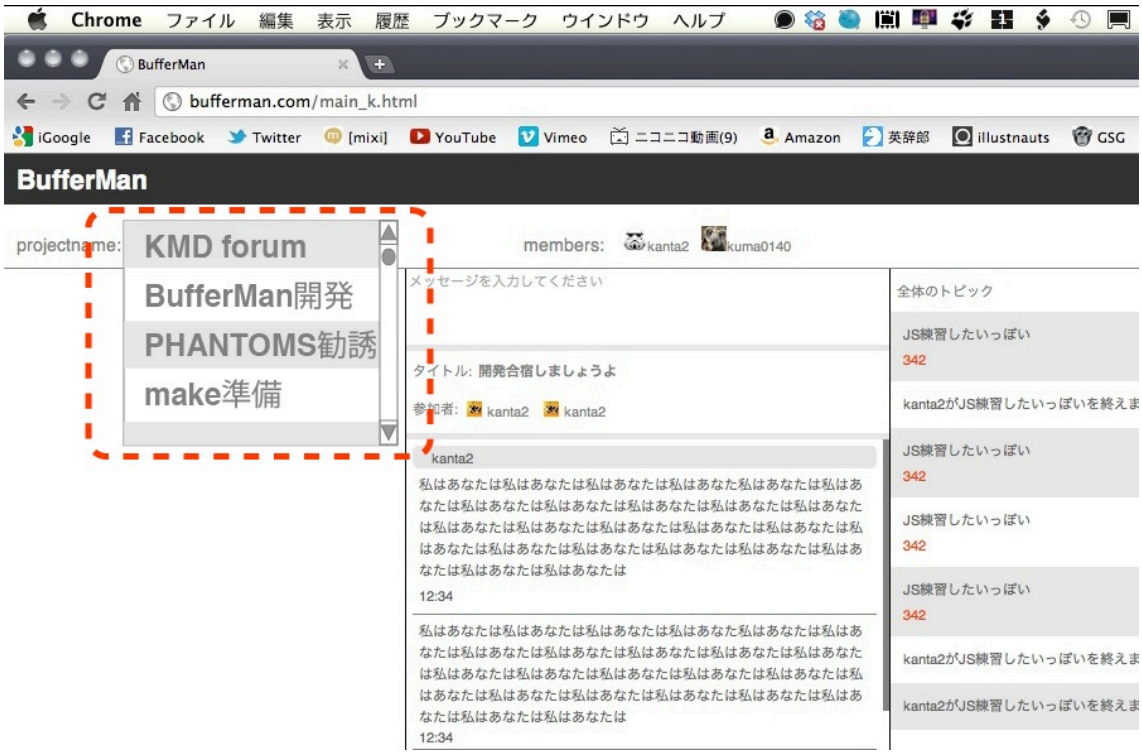


図 4.2 テキストチャット画面

ク一覧画面、テキストチャット画面がそれぞれのプロジェクトに対応したものに変更される。図 4.3 にプロジェクト選択の画面を示す。


Chrome ファイル 編集 表示 履歴 ブックマーク ウインドウ ヘルプ
BufferMan
bufferman.com/main_k.html
iGoogle Facebook Twitter [mixi] YouTube Vimeo ニコニコ動画(9) Amazon 英辞郎 illustnauts GSG
BufferMan
projectname:
KMD forum
BufferMan開発
PHANTOMS勧誘
make準備
members: kanta2 kuma0140
メッセージを入力してください
タイトル: 開発合宿しましょうよ
参加者: kanta2 kanta2
kanta2
私はあなたは私はあなたは私はあなたは私はあなた私はあなたは私はあなたは私はあなたは

ンバーが同じタスクに取り組んでしまうリスクは低減する。また、進捗を確認するためにメンバー同士が直接連絡をとる必要がなく、手軽に確認することができるため、メンバー同士の不安感も低減する。図 4.4 に参加者一覧の画面を示す。

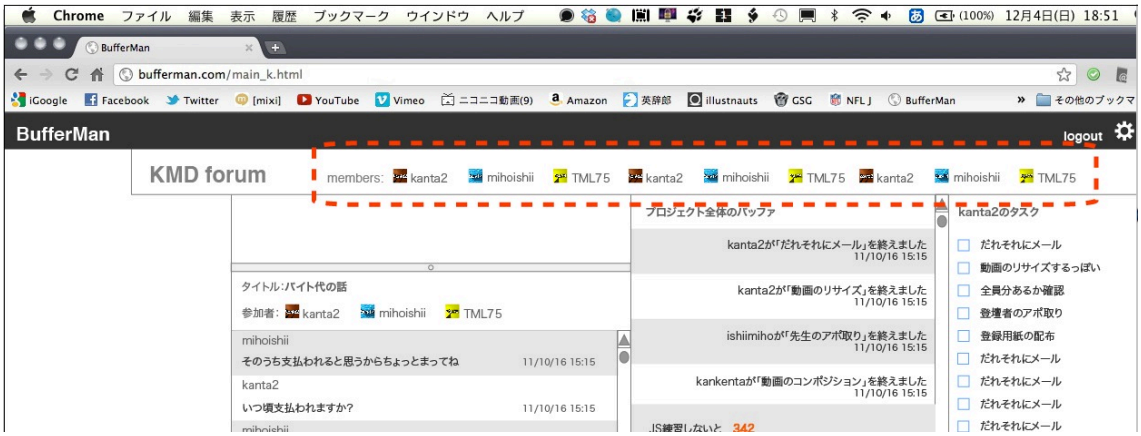


図 4.4 参加者一覧画面

トピック一覧画面

ここではトピック一覧画面について議論する。ここには議論されるべき懸案事項が一覧で表示される。ここに表示されている懸案事項は、それをタイトルにしたテキストチャットが開く。ここに表示される懸案事項の種類は大きく分けて 3 つある。ひとつは誰かがなにかタスクを終えたという報告である。これは「(メンバー名) が (タスク名) を終えました」という表記で表示される。もうひとつは外部からの状況報告である。これは「〜のサービスの API 規制がかかった」「〜が脱退した」などのものである。そして最後は提案である。これは「〜について考えるべきだ」などの抽象度の高いものである。これらの懸案事項は 2 通りの方法でこのトピック一覧に格納され、表示される。まずひとつは誰かが何かを終えた、という報告のためにトピック一覧画面のラジオボタンを押す入力方法である。これはメンバーが自分のタスク一覧画面に表示されているタスクを終え、そのタスクの先頭に表示されているラジオボタンをクリックすると、そのタスクが対応

しているプロジェクトのトピック一覧画面に、自動的に「（メンバー名）が（タスク名）を終えました」という表示がなされるものである。もう一つはテキストチャットの発言の下にある「トピックに送る」ボタンを押す方法である。これはテキストチャット内での即興的な同期的コミュニケーション内で生まれた、新しい観点を持った発言、懸案事項を手軽な動作で即時的に保持するための方策である。これによって各メンバーは今やり取りされている会話の没入感や創発感を維持したまま新しい懸案事項をトピック一覧に保持することができ、そのときやり取りされている会話が一段落した所で、会話の中で出てきた新しい懸案事項について検討することが出来る。また、表示されている懸案事項は、三角タブを押すとその懸案事項から作成されたタスクが一覧として懸案事項の下に表示される。これによってその懸案事項からどういったタスクを誰が作成したのかがわかる。懸案事項の横にはその懸案事項をタイトルとしたテキストチャットにおいてどれだけの議論がなされたのかがわかりやすいように、テキストチャットでの発言数がオレンジ色の数字で表示される。これは自分が関わっていない懸案事項において他のメンバーがどれだけ議論をしているのかがひと目でわかるものであり、これもメンバー同士のお互いの進捗に対する不安を低減するものである。トピック間を移動するためには移動したい先のトピック名をクリックする。するとテキストチャット画面が移動した先のトピックと対応したものになる。各トピックの横には「メールで送信」というボタンが設置されており、このボタンを押すとそのトピックに関して行われたテキストチャットのログがメールでプロジェクトに参加しているメンバーに送信される。この機能に関しては後述する。また、それぞれのプロジェクトはプロジェクトが進行していく中で作成されるものとは別に、全てのプロジェクトが全体チャットというテキストチャットを備えている。全体で議論したい話題、共有したい話題などは全体チャットを使って議論する。図 4.5 にトピック一覧の画面を示す。

テキストチャット画面

ここではテキストチャット画面について議論する。ユーザーはテキストチャット内のフォームに発言を入力し、エンターボタンを押すことで発言する。発言は時

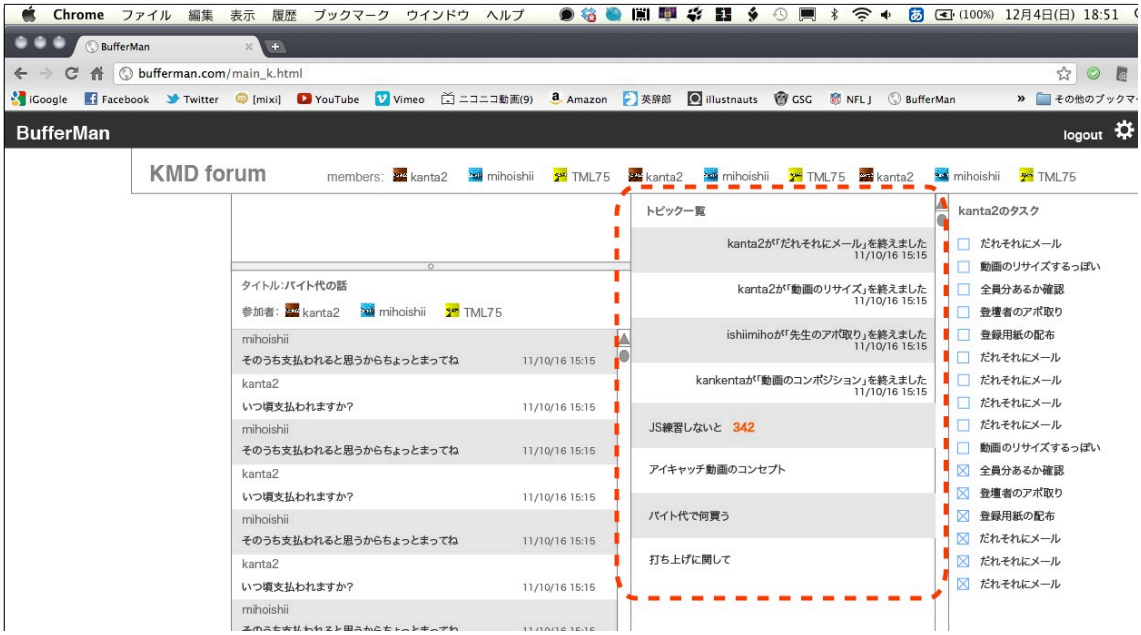


図 4.5 トピック一覧画面

系列順に表示され、画面の上部にいくほど新しい発言が表示される。各発言は、マウスオーバーすると「トピックに送る」、「タスクに送る」のふたつのボタンが出現する。それにまつわるタスクは明確ではないが、議論すべきと考えられる懸案事項がチャットの中で生まれた場合、「トピックに送る」ボタンを押すことでその発言はトピック一覧画面に格納される。この懸案事項はふとした思いつきや、提案、協調作業者全員が知っておくべき情報などの種類がある。格納された懸案事項はそれを題名としたテキストチャットにおいてその後議論することができる。次に、「タスクに送る」ボタンは、懸案事項に関する議論の後に明確化されたタスクがテキストチャット内で生成された場合にボタンを押すことで、発言をタスク一覧画面に格納することができる。また、前章でも指摘したとおり、テキストチャットは事後的なチャットログの可読性が非常に低い。BufferMan ではこのテキストチャットの事後的なログの可読性の低さを解決するために、テキストチャットでやり取りされた会話をメールで共有する機能を持っている。この機能に関してはログメール送信機能のところで議論する。図 4.6 にテキストチャットの画面

を示す。

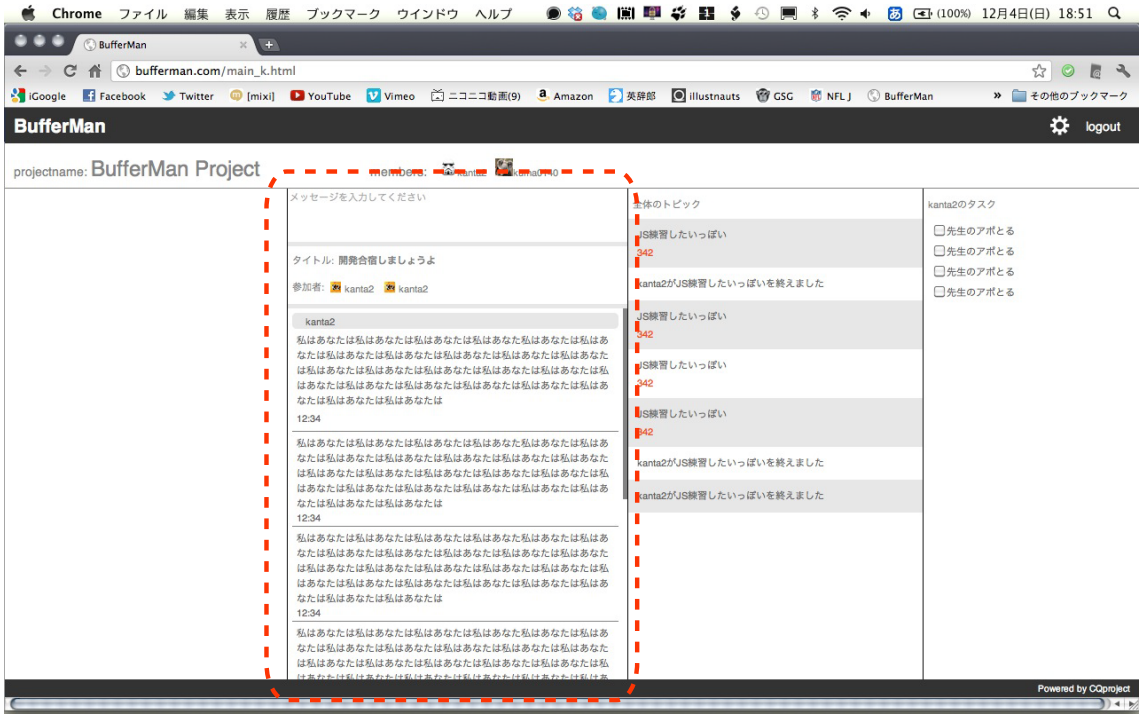


図 4.6 テキストチャット画面

タスク一覧画面

ここではタスク一覧画面について議論する。この画面には懸案事項に関する議論の中で明確化されたタスクが格納される。格納するにはテキストチャット画面で、発言の上にマウスオーバーすることで出てくる「タスクに送る」ボタンを押す。表示されているタスクにはそれぞれ先頭にラジオボタンがついており、ユーザーはそのタスクを完了するとそのラジオボタンにチェックを入れる。するとそのタスクが完了したフラグが立ち、トピック一覧画面に「(タスクを終えたユーザー) が (タスク名) を終えました」という表示がなされる。タスクを格納するのはタスクを実行する本人であり、議論なしにユーザーが他のメンバーにタスクをふることはできない。図 4.7 にタスク一覧の画面を示す。

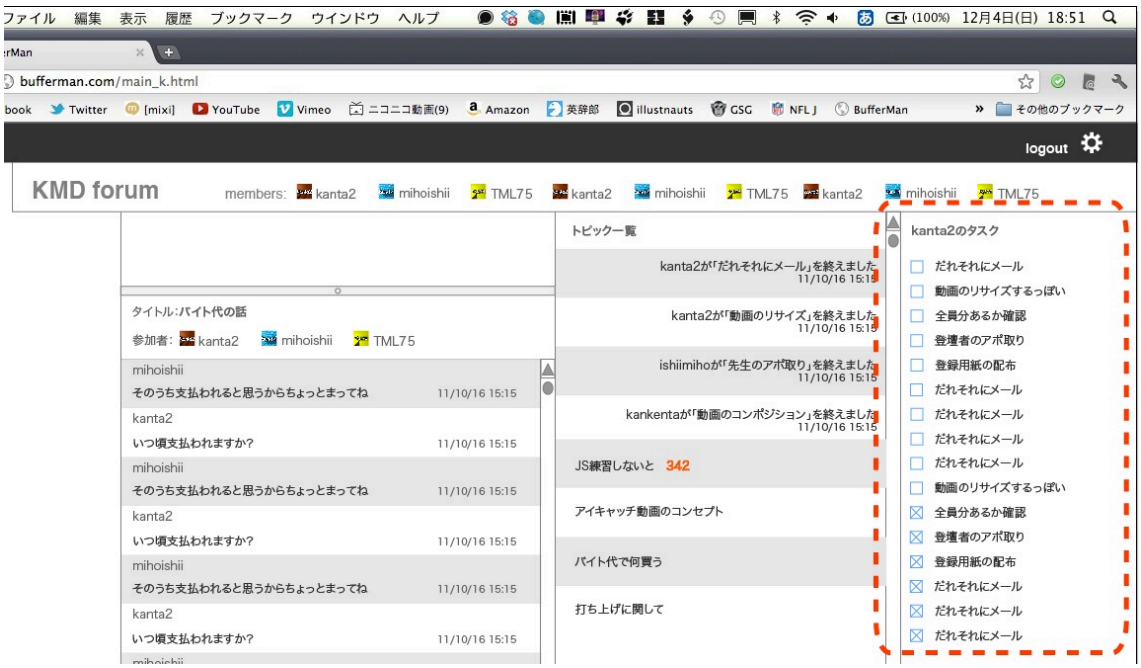

ファイル 編集 表示 履歴 ブックマーク ウインドウ ヘルプ
(100%) 12月4日(日) 18:51
bufferman.com/main_k.html
Twitter
[mixi]
YouTube
Vimeo
ニコニコ動画(9)
Amazon
英辞郎
illustnauts
GSG
NFL J
BufferMan
その他のブックマーク
logout
KMD forum
members: kanta2 mihoishii TML75 kanta2 mihoishii TML75 kanta2 mihoishii TML75
タイトル:バイト代の話
参加者: kanta2 mihoishii TML75
mihoishii
そのうち支払われると思うからちょっとまってね 11/10/16 15:15
kanta2
いつ頃支払われますか? 11/10/16 15:15
mihoishii
そのうち支払われると思うからちょっとまってね 11/10/16 15:15
kanta2
いつ頃支払われますか? 11/10/16 15:15
mihoishii
そのうち支払われると思うからちょっとまってね 11/10/16 15:15
kanta2
いつ頃支払われますか? 11/10/16 15:15
トピック一覧
kanta2が「だれそれにメール」を終えました 11/10/16 15:15
kanta2が「動画のリサイズ」を終えました 11/10/16 15:15
ishiimihoが「先生のアポ取り」を終えました 11/10/16 15:15
kankentaが「動画のコンポジション」を終えました 11/10/16 15:15
JS練習しないと 342
アイキャッチ動画のコンセプト
バイト代で何買う
打ち上げに関して
kanta2のタスク
だれそれにメール
動画のリサイズするっぽい
全員分あるか確認
登壇者のアポ取り
登録用紙の配布
だれそれにメール
だれそれにメール
だれそれにメール
だれそれにメール
動画のリサイズするっぽい
全員分あるか確認
登壇者のアポ取り
登録用紙の配布
だれそれにメール
だれそれにメール
だれそれにメール

図 4.7 タスク一覧画面

4.2. ログメール送信機能

ここでは BufferMan のテキストチャットログの送信機能について議論する。

4.2.1 テキストチャットログ送信機能の設計

ログ編集機能はチャットに入力された会話のログを、メールも用いて共有するに際し、作業者同士がわかりやすいように編集するための機能である。チャットでやり取りされるコミュニケーションは即興性の応酬が多く論理的ではない。また、非常に短い発言の連続や発言の重複がみられ、明確化されたはずのタスクが他の会話にまぎれて事後的に確認できない可能性が高い。そのためログを事後的に確認しやすいかたちに編集する必要がある。文脈として有益な部分、明確化されたタスクを可読性高く共有するため、ログを送信する際に必要と思われる部分をドラッグで選択する。選択された発言は色が変わり、その他の部分は破棄される。JavaScript によって実装する。

4.2.2 テキストチャットログ送信機能の実装

テキストチャットによって行われていたやりとりのログを各ユーザーにメールで送信する処理をする。

トピック一覧画面には、それぞれのトピックの横に「メールで送信」ボタンが設置されている。このボタンを押すと、そのトピックに関してやりとりされたテキストチャットのログが送信される。テキストチャットのログが送信される宛先は、そのプロジェクトに参加しているメンバー全員の、アカウント作成時に登録されたメールアドレスである。

メールはそのチャットに参加しているメンバーが、登録時に登録したメールアドレスに送信される。この画面に関しては図 4.8 で示す。

差出人: notification@kanta2.com
件名: @@@今日のミーティングの議題
日時: 2011年8月4日 16:00:50JST
宛先: 竹内 冠太

@@@今日のミーティングの議題
えっと何から始めますか
まずはM1の論文
それからM２の修論
それから次回のプレナリーミーティングの
発表内容とか決める
なるほど
誰が出席すんのかな
M1は全員来れる？
いや結構怪しい
なるほど
M2は？
全員来れると思う
場所はどうしますか
02でやろうか。
アジェンダ投げときます
了解
アジェンダもM1に書いてもらわないとそろそろ
まあそれもミーティングで話そうか

図 4.8 メール送信機能

第5章
BufferManの実験、評価

ここではBufferManの性能実験と評価を行う。

5.1. BufferManの性能実験環境

ここではBufferManの性能実験の環境について議論する。

5.1.1 テキストチャットログ送信機能の実験方法

BufferManシステムはphpで実装されている。phpの処理速度のベンチマークの取得にはPEARのbenchmarkを使用する。benchmarkはphpプログラム内の測定したい場所にマーカーを記述することでマーカー間の処理にかかる時間を測定する。BufferManシステムは送信ボタンを押すと画面が再読み込みされる。まずプログラムの先頭にマーカーを記述し、次に読み込んだdbのデータを画面に表示させるHTMLの場所にマーカーを記述することで、送信ボタンを押してから発言が画面に反映されるまでの時間を測定した。

5.1.2 BufferManの実験対象

チャットを使ったコラボレーションにおいて、共同作業者が8人を超えると一部のメンバーがリーダーシップを取り問題解決を行う傾向がみられ、また発言の少ないメンバーに対しては相手の姿が見えないことに不安を感じグループ内での協調作業という感覚が低減する[21]。したがってBufferManシステムでは共同作

業者数の上限を７人とする。ここでは最大共同作業謝数の７人での使用を想定し、データベース内の members テーブルに７名分の情報を入力し、実験を行った。また、テキストチャットでのひと発言における平均文字数を１９文字と仮定してベンチマークを測定した。BufferMan システムを使ったミーティングの時間を６０分とした。チャット内での発言が 3 秒に 1 発言更新されると仮定すると、ミーティングの最中には１２００件の発言があると推定できる。db の message テーブルに１２００件の発言が格納されていることを仮定する。そのため発言数は１００から１２００までを１００発言ごとに測定した。

5.1.3 テキストチャットログ送信機能の実験環境

BufferMan システムの実験環境を以下に示す。

表 5.1 実験環境

項目	詳細
CPU	QEMU Virtual CPU 2.80GHz
メモリ	512MB
OS	Linux version 2.6.32-5-amd64

5.2. テキストチャット発言表示時間の測定実験と評価

作業者がテキストエリアに発言を入力し、送信ボタンを押してから発言が画面に反映されるまでの反応速度が遅いと、創発性を担保するための同期的なな協調作業におけるコミュニケーションに支障をきたす。そのため送信ボタンを押してから画面に発言が表示されるまでにかかる時間を評価した。

5.2.1 テキストチャットログ送信機能の発言表示時間の測定実験

通常の発言をテキストエリアに入れ、送信ボタンを押してから発言が画面に表示されるまでの時間を測定した。実験の結果を図 5.1 に示す。

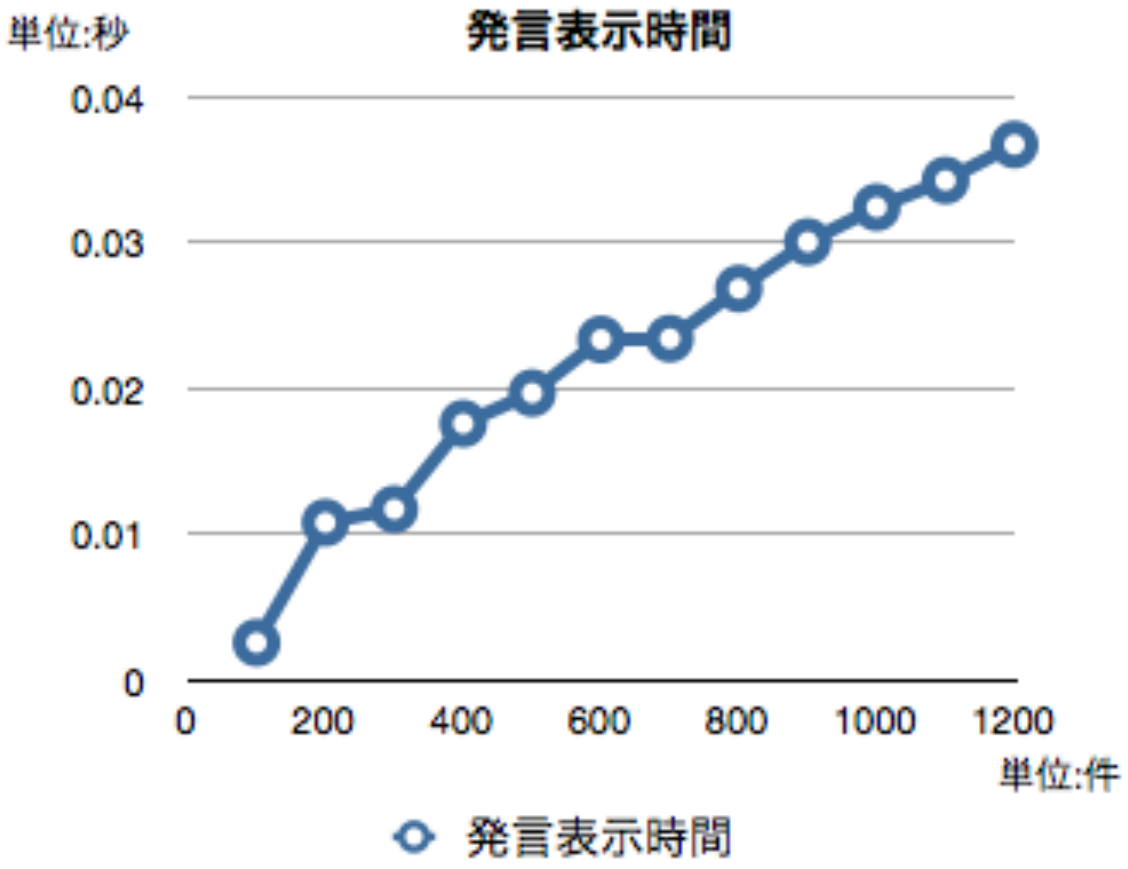


図 5.1 実験結果　単位＝横軸；発言　縦軸；秒

5.2.2 テキストチャットログ送信機能の発言表示時間の評価

発言数が１００から１０００までの発言表示時間の平均値は 19.8142 ミリセカンドであった。これは入力が画面に反映されるまでの時間として無視出来るほど短

く、創発性のための同期的コミュニケーションを阻害するものではないと言える。

5.3. テキストチャットログ送信機能の実験評価

5.3.1 テキストチャットログ送信機能の発言表示時間の測定実験

共同作業者にチャットのログをメールで共有するため、トピックの横の「メールで送信」ボタンを押してからメールが送信されるまでの時間を測定した。実験の結果を図 5.2 に示す。

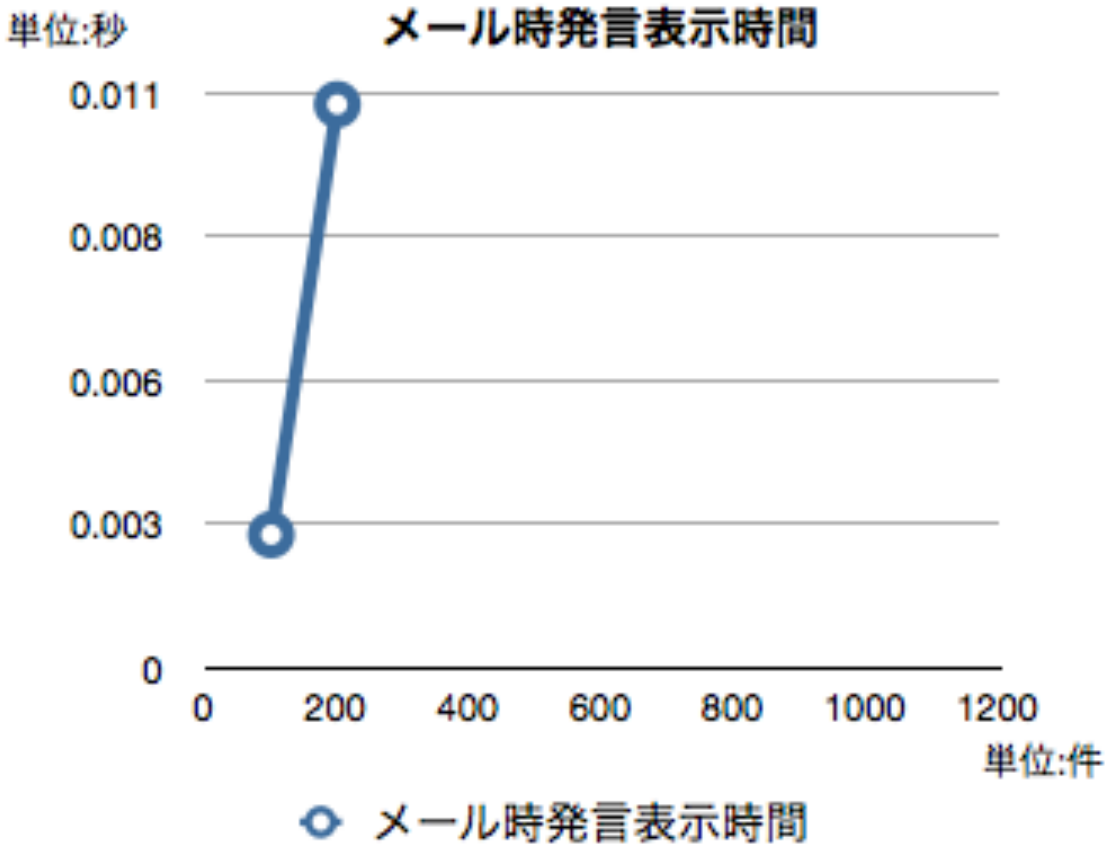


図 5.2 実験結果　単位＝横軸；発言　縦軸；秒

5.3.2 テキストチャットログ送信機能の発言表示時間の評価

メール時発言表示時間は発言数が１００件、２００件の平均時間が 4.2 ミリセカンドとなった。これは入力が画面に反映されるまでの時間として無視出来るほど短く、創発性を阻害するものではないと言える。

5.4. テキストチャットログ送信機能評価のまとめ

テキストチャット表示時間、テキストチャットログメール評価時間の平均は、想定される使用環境下ではそれぞれ 19.8 ミリセカンド、4.2 ミリセカンドであった。実験結果は創発のための同期的なコミュニケーションに支障をきたさないことを示した。

第6章
結論と今後の展望

6.1. 結論

本論では、プロジェクトにおける課題発見型協調作業を支援するための、テキストチャットシステム BufferMan について述べた。BufferMan はテキストチャット機能に加えて、テキストチャットでのコミュニケーション内で出てきた懸案事項とタスクを即時的に保持することができ、またテキストチャットのログを、懸案事項ごとにタイトルをつけて電子メールで送信する事ができる。通常テキストチャットで生成された懸案事項やタスクはユーザーの経験則よって外部アプリケーションに保持されていた。この方法ではコミュニケーションにおける創発性を担保するためのコミュニケーションの即時性が失われていた。また即時性を失わないために懸案事項やタスクの保持を後回しにし、結局保持されない場合もあった。BufferMan はこれを解決するために懸案事項やタスクをアプリケーション内で保持することを目的にしている。また５章で示したように、BufferMan が遠隔での課題発見型協調作業において有効なコミュニケーションプラットフォームとして機能していることが実証できた。実証実験では想定される使用状況下でテキストチャットの表示に遅延がないこと、またメールでの情報共有に遅延がないことを示した。評価の結果、BufferMan は想定環境で安定稼働することを示した。

6.2. 今後の展望

本論では、プロジェクトにおけるコミュニケーションツールとしての BufferMan を実装し、評価した。しかしインターネットを使った協調作業において、コミュ

ニケーションの方法はプロジェクトの一部に過ぎない。コラボレーションを行うメンバー、またそのプロジェクトで取り組む課題のマッチング等を行うにおいてもインターネットは活用できると考える。CQ Project の同期である相島雅紀の開発しているプロジェクトマッチングサービス「Implify」との連携を実現することでインターネットを使った遠隔コラボレーションのプラットフォームとして発展していくことを願っている。

謝　　辞

本研究の指導教員であり、幅広い知見から的確な指導と暖かい励ましやご指摘をしていただきました慶應義塾大学大学院メディアデザイン研究科の杉浦一徳准教授に心から感謝いたします。

研究の方向性について様々な助言や指導をいただきました慶應義塾大学大学院メディアデザイン研究科の植木 淳朗特任講師に心から感謝いたします。

研究指導や論文執筆など数多くの助言を賜りました慶應義塾大学大学院 メディアデザイン研究科の稲見 昌彦教授に心から感謝いたします。

研究活動、学生生活全般にわたり数多くの貴重な助言、ご指導をいただき、自分自身の可能性に気づかせて下さった慶応義塾大学大学院 メディアデザイン研究科の奥出 直人教授に心から感謝致します。

研究の方向性や論文執筆などについて様々な助言や指導をいただきました慶應義塾大学大学院 メディアデザイン研究科博士後期課程の塚原康仁さんに心から感謝いたします。

時に苦楽を共にし、様々な面において助けてもらった CQ Project 同期の相島雅樹君、荒川裕紀君に心から感謝致します。三人という少ない人数でしたが、お互い支え合う事でここまで研究を行うことが出来たと思います。

さまざまな面から研究活動を支えていただき、時に苦楽を共にした慶應義塾大学大学院メディアデザイン研究科 Media Telescope Project の皆様に心から感謝致します。

さまざまな局面で刺激と勇気を与えてくれた麻布高校の皆様に心から感謝致します。

創部初の二部昇格を遂げた東京外国語大学アメリカンフットボール部 PHANTOMS の皆様に心から感謝致します。その真摯な取り組みはいつも私に勇気を与

えてくれました。
　最後に、研究活動に関するご理解とともに、経済面や生活面において支援していただきました家族に心から感謝いたします。

参 考 文 献

[1] BufferMan. http://bufferman.com/.

[2] irc. http://www.irc.org/.

[3] skype. http://www.skype.com/intl/ja/home/.

[4] MSNmessenger. http://messenger.live.jp/.

[5] chatwork. http://www.chatwork.com/ja/.

[6] A. Cooper, R. Reimann, and D. Cronin. About Face 3 インタラクションデザインの極意:. アスキー・メディアワークス, 2008.

[7] デザイン思考の道具箱: イノベーションを生む会社のつくり方. 早川書房, 2007.

[8] google person finder. http://japan.person-finder.appspot.com/.

[9] google person finder bot. http://gpf-bot.jizmo.co.jp/.

[10] twitter. http://twitter.com/.

[11] C.A. Ellis, S.J. Gibbs, and G.L. Rein. *Groupware: the research and development issues*. MCC technical report. Microelectronics and Computer Technology Corp., 1988.

[12] cscw. http://cscw2012.org/.

[13] 図解グループウェア入門. オーム社, 1991.

[14] 石井裕. コンピュータを用いたグループワーク支援の研究動向 (¡特集¿ヒューマンインタフェース). コンピュータソフトウェア, Vol. 8, No. 2, pp. 110–122, 1991-03-15.

[15] 岡田謙一. グループウェアはどう進化するのか?(コンピュータと通信). 情報処理, Vol. 39, No. 6, pp. 540–544, 1998-06-15.

[16] 信彦小倉, 晴美渡辺. ロボットコンテストを利用した組込み教育の実践. 情報処理学会論文誌, Vol. 49, No. 10, pp. 3531–3540, 2008 年 oct 月.

[17] 金子美和. 2108 アジャイル開発におけるプロジェクトマネジメント : ビジネス目標を達成するための効果的な管理 (一般セッション). プロジェクトマネジメント学会研究発表大会予稿集, Vol. 2009, pp. 107–112, 2009-03-10.

[18] 博之矢野, 昭伊藤. 協調型タスクにおける非言語情報の使われ方. Technical Report 21(1995-SLP-010), 郵政省通信総合研究所関西支所, 郵政省通信総合研究所関西支所, 1996 年 feb 月.

[19] 昭伊藤, 博之矢野. 「共話」－創発的対話の対話モデル. Technical Report 12(1997-SLP-020), 郵政省通信総合研究所, 郵政省通信総合研究所, 1998 年 feb 月.

[20] 長谷川隆明, 中野有紀子, 加藤恒昭. 反射と熟考の相互作用に基づく協調的対話モデル. 全国大会講演論文集, Vol. 52, No. 2, pp. 411–412, 1996-03-06.

[21] 到倉本, 純宗森, 勝首藤. チャットに注目した発想支援グループウエアのコミュニケーションに関する検討. Technical Report 8(1997-GN-026), 大阪大学大学院基礎工学研究科, 大阪大学大学院基礎工学研究科, 大阪大学大学院基礎工学研究科, 1998 年 jan 月.

[22] 相島雅樹, 塚原康仁, 植木淳朗, 杉浦一徳. Dtmp メソッドを用いたグループ編成支援システムの提案. Technical report, 情報処理学会, 2011.

[23] 謙一岡田, 温松下. 協調の次元階層モデルとグループウェアへの適用. Technical Report 95(1993-GN-004), 慶應義塾大学理工学部, 慶應義塾大学理工学部, 1993 年 oct 月.

[24] 非ウォーターフォール型開発に関する調査結果. http://sec.ipa.go.jp/reports/20100330a.html.

[25] ichat. http://www.apple.com/jp/support/ichat/.

[26] google. http://google.com/.

[27] 金子美和. 適応的反復型開発におけるプロジェクトマネジメント : ビジネス目標を達成するための効果的な管理. プロジェクトマネジメント学会誌, Vol. 11, No. 5, pp. 22–27, 2009-10-15.