

だれでも構築運営できるコラボレーションシステムの実現

— qwikWeb を用いたコミュニケーション・パターンの実践 —

江渡 浩一郎*、濱崎 雅弘**、西村 拓一**

だれでも構築運営できるコラボレーションシステムの実現のために、ユーザがグループ活動形態に適したシステムを容易に構築することを可能とするコミュニケーション・パターンを設計思想とした qwikWeb を提案する。また、本システムをデザイン、実装、運用改良し運用データの分析を行うことで本システムの妥当性と有効性を示す。

1 はじめに

現在、我々は通信技術やインターネット技術の進歩により、時空を超えた情報共有手段を手に入れている。2005年時点でも携帯電話等の移動端末の利用者数が推計6,923万人、パソコン利用者数が推計6,601万人^[1]であり、組織内も含め気軽に情報共有し協働作業するシステムを提供できたら、組織内活動、サークル活動、地域活動など様々な創造的活動や経済活動を活性化できると考えられる。しかし、既存のシステムでは、管理が困難であり、ユーザが知識の蓄積・構造化を効率的に進められないという問題があった。

そこで、本論文では、表1で示すような、複雑なアクセス制御を必要とせず、管理や習得が容易で知識を蓄積・構造化できるシステムを希望するユーザグループを対象としたコラボレーションシステム：qwikWeb を提案する^[2]。この設計思想として、ユーザ自身が好みのシステムを構築することを可能とするコミュニケーション・パターンを採用している。また、本システムをデザイン、実装、運用改良し運用データの分析を行うことで本システムの妥当性と有効性を示す。

本論文では、次章にて Web システムの本格研究について論じ、この開発プロセスに応じて、3章にてコラボレーションシステムの現状分析を行う。4章で qwikWeb のサービス

設計、5章で実装した qwikWeb を説明する。6章にて運用した qwikWeb のデータ解析と考察を行う。7章で関連研究を紹介して本研究の位置づけを明確にし、8章にてまとめと今後の方針を述べる。

2 Webシステムの本格研究

Web システムを開発し実用化するための本格研究プロセスでは、ユーザの利用形態を把握しユーザからのフィードバックを活用したシステム開発が重要である。基本的な設計開発手順は図1のように設計者、システム開発者、ユーザを含めたユーザ参加型のものとなる。

まず、現状分析フェーズにおいてサービスの概要や対象とするユーザを設定し現状の問題点を分析する。サービス概要やターゲットユーザは、現状の問題点を見極めることによって決定できるため、この現状分析における作業は同時並行的に進められる。また、既存システムの問題点の洗

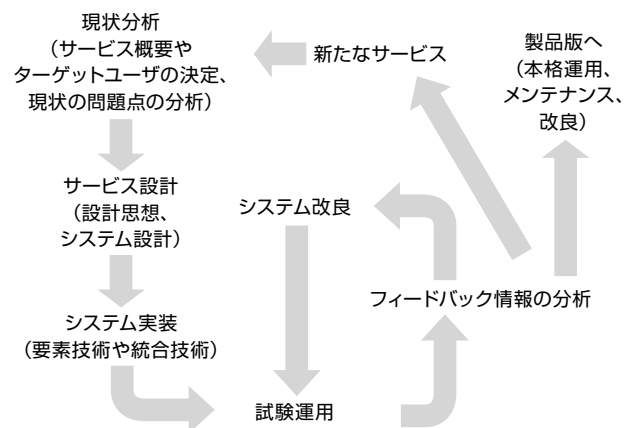


図1 Webシステムの開発プロセス

表1 既存のコラボレーションシステムとqwikWebとの比較

	管理の容易さ	知識の蓄積・構造化	複雑なアクセス制御
サイボウズ ^[3] など	×	△	○
qwikWeb	○	○	×

い出しは、機能レベルだけでなくコンセプトレベルで行う。これらの結果をもとに、新たなサービスの設計を行う。この設計ではサービスのビジョンを含む設計思想を決定し、要素技術の選択や改良、将来の拡張性や他システムとの連携を配慮したシステム設計を行う。システム実装はシステム設計に従い、ソフトウェア工学上コストパフォーマンスを配慮して開発する。次にある程度の規模のユーザを対象として試験運用を行う。このフェーズではユーザからのフィードバックを収集できるようにシステムのログを蓄積するだけでなく、ユーザからの直接の声を収集する。これらの情報を分析して随時設計を修正し、引き続き試験運営によりシステムの改良を進めて製品版を完成させる。これらの分析改良時に隠れたニーズを発見して新たなサービスを構築する知見を収集できる場合もある。製品版は運営会社に販売またはライセンスされて本格運営されるが、この中でも絶えずメンテナンス、システム改良が進められる。

以上が Web システムにおける本格研究の代表的なユーザ参加型プロセスである。qwikWeb でもこのプロセスに従って開発し、現在試験運用がほぼ終了した段階である。次章以降 qwikWeb の開発プロセスについて述べる。

3 コラボレーションシステムの現状分析

企業内やサークルなど組織内での連携や協働作業では、電子メールが重要な役割を果たしており、特にメーリングリスト（以下、ML）による連絡や協働が盛んに行われている。しかし、一連の協働作業においては、情報が構造化されておらず最新の情報の確認にも手間がかかる場合が多い。一方、Lotus Notes^[3] やサイボウズ^[4] などのような、スケジュール調整やテーブル機能、1つの文書を協働で作成する機能を持つコラボレーションツール、グループウェアが利用されている。しかし、このようなシステムの立ち上げ手続きや作業は煩雑であり、メンテナンスの手間も管理者に集中する。また、グループウェア内での連絡手段は、通常利用するメールアドレスと異なるため、メールでのやり取りを基に協働で文書や知識を構築する場合には、情報の移動の手間が発生する。組織外のメンバとの柔軟な連携も容易でない。

立ち上げや運営コストを低く抑える必要があり、少数の管理者に負担が集中しないコラボレーションシステムを欲しているユーザにとって、特にシステムに関するノウハウを習得しなくても、そのシステムを見ただけで利用方法が分かるくらい分かりやすいことが必要である。また同時に、すでに慣れ親しんだ環境を出発点として、徐々に自分の理想とする情報環境を実現できるような発展性を備えた構造にする必要がある。一方、このようなユーザにとっては、各文

書や知識のきめ細かいアクセス制御（どのユーザがどの文書を編集できるかなど）は不要である。これは、サイボウズ^[4] などは数 1,000 人のユーザを想定しているのに対し、管理者に負担をかけられないユーザは、数 10 人程度のコラボレーションシステムを必要としているからである。

筆者らは、ターゲットユーザの一例として、グループウェアの管理に時間をとることができない身近な研究者の協働作業に着目した。彼らは日常的にメールを主とする連絡手段を利用している。しかし、イントラでのグループウェアを利用した協働作業は、上司からの命令により文書を作成する場合などを省いて、自発的に行っていない。これは協働作業により文書等を作成した方がよい状況においても同様である。なぜこのようなことが起こるのであろうか。岡田^[5] は協調の階層モデルとして、コラボレーションの直下の階層にコミュニケーションを位置づけ、その重要性を示した。北川^[6] はオンラインコミュニティを、情報を得る、関係を結ぶ、協働を行う、という 3 つに機能分類し、大半は情報を得るだけのコミュニティだが、幾つかはメンバ間で親密になり、さらにその中の幾つかは協働を行うようになるのではないかと述べた。このように協働作業（コラボレーション）はその背景に密なコミュニケーションが不可欠であり、コミュニケーションによる情報交換や合意形成を経て、ようやく協働作業が生まれうると考えられる。実際に、報告文書の作成などのような協働作業はメールをベースに進められているものの、すべてのメールを読んで修正や統合をしないと最新の状況が把握できず、知識の構造化や共有上問題があった。そこで、グループ内で自然な連絡を基にグループ内知識の構造化、協働文書構築も容易に行えるシステムが求められていると考えた。

つまり、「管理の容易さ」「習得の容易さ」「知識の蓄積・構造化」が課題となっていたと言える。

4 qwikWebのサービス設計

4.1 qwikWebの設計思想

4.1.1 基本思想

3章で述べたコラボレーションシステムにおける課題をユーザの立場から捉えなおした結果、「ユーザがシステムやその利用形態を、思い通りに設計できること」が根本であると考え、これを qwikWeb の基本思想と設定した。

従来の Lotus Notes^[3] やサイボウズ^[4] などでは、一般ユーザがシステムを改変することはできず、この基本思想を取り込むことが困難である。一方、ウォード・カニングが開発した Wiki を調査した結果、本提案で目指す基本思想と密接な設計思想をもとに開発されたことがわかった。そこで、この Wiki の設計思想を「コミュニケーション・パターン」^{[7][8]}

と名付けて次項で紹介し、さらに qwikWeb の設計思想について論じる。

4.1.2 Wikiの設計思想「コミュニケーション・パターン」

Wiki には建築家クリストファー・アレグザンダーによるパターン・ランゲージ^[9]と呼ばれる思想が影響している。パターン・ランゲージとは、ある建築を実現するにあたって繰り返し表れる建築形態を集め、言語のようにまとめたものである。これを用いることによって、建築物の利用者が設計に関与することができるようになり、結果として良い建築が実現できるとされている。Wiki の開発者であるウォード・カンニングは 1987 年にこのパターン・ランゲージに興味を持ち、特に利用者が設計に関与するという観点から、ソフトウェアの実現にパターン・ランゲージを応用することを試みた。ユーザインタフェースにおけるいくつかのパターンを抽出し、そのパターンをユーザに提供することにより、比較的短期間に優れたソフトウェアの設計を行うことができたという^[10]。この実験は後にデザイン・パターンと呼ばれる開発者間の共通言語を実現する大きな潮流へと発展していった^[11]。カンニングは同じ 1987 年に発売された Apple 社の HyperCard を用い、発見したパターンを記録・編集するためのパターン・ブラウザを作成した^[12]。このパターン・ブラウザが後に Wiki へと発展した。つまり Wiki とはパターン・ランゲージを記録・編集するための基盤として考案されたものである。

カンニングは 1995 年にそのパターン・ブラウザを、それまでに蓄積されたコンテンツを元に、Web 上のシステムとして作り直し、それに WikiWikiWeb という名前を与えた。これが現在につながる Wiki の起源である。それまでの HyperCard によるパターン・ブラウザから Wiki への大きな違いは、インターネットにリアルタイムに接続されたという点である。そのため、Wiki は単に情報を蓄積する場としてだけでなく、蓄積された情報について議論するための場としての性格も同時に持つことになる。

一般にコミュニケーションに用いられるシステムは、情報がある定型的な形式にあてはめて管理する。典型的なものが掲示板であろう。1つの記事は、執筆者名、日付け、文章のタイトル、本文などといった項目に分かれて保持される。記事には権限が付与され、一般に執筆者のみが記事を書き換えられる。Wiki における情報の保持はそれとはまったく異なり、いわば 1 枚の紙を共有しているような状態になる。情報には定型的な形式はなく、どのような文章でも自由に書ける。権限という概念もないため、どの文章を誰でも編集できる。そのような環境で複数の執筆者が共同で文章の執筆を進めていくと、必然的に混乱が起こる。その混乱を解消するために、ローカルなルールが生まれる。

例えば「自分の意見を述べる際には文末に自分の名前を残す」などである。そのように当初は何もルールがない状態だった Wiki に、徐々に自然発生的にルールが生まれていった。そのようなルールはコミュニケーションに関わるルールが利用者間のつながりにおいて自然発生的に生まれたものであり、建築におけるパターンの発生と同様な構造を持つと言える。そのため、そのようにして生まれたコミュニケーションに関わるルールを、ここでは「コミュニケーション・パターン」という言葉で表すこととする。

Wiki は、当初の目的通り情報システムを実現するために必要なパターンを収集するだけではなく、そのようなパターンに興味を持つ人同士のコミュニケーションの場でもあり、またコミュニケーションを行うにあたってのルール（コミュニケーション・パターン）について議論する場でもあり、そのようにして自然発生的に作り出されたコミュニケーション・パターンを記録・収集する場でもあった。単にパターンを収集するだけではなく、情報を収集するためのルール（コミュニケーション・パターン）もまた同時に収集するというメタな意味を同時に持つ場として Wiki は成長していったのである。

Wiki において最も重要とされるコミュニケーション・パターンは、スレッド・モード（議論状態）とドキュメント・モード（文章状態）の区別である。スレッド・モードとは、そのページを用いて議論が進んでいる状態のことを指し、そこでは各利用者の意見が掲示板のように並んでいる状態となる。それに対しドキュメント・モードとは、ページに客観的な記述のみが置かれた状態であり、主観的な意見の記述は排除される。実際にはほとんどのページはハイブリッド状態となり、ページ上方には客観的な記述があり、その下の方で議論が進められる。各々のページは、そのページの位置付けを示す短い文章から始まり、それについて各々の意見がスレッド・モードとして追加され、議論が進んで個々の意見の違いが吸収されていき、それが最終的に 1 つのまとまった客観的な記述（ドキュメント・モード）へと成長していくものとされた。つまり、全てのページが最終的にはドキュメント・モードへ成長することが目標であるとされた。

このように、あらかじめルールを決めてそれに沿ってコミュニケーションを進めるのではなく、コミュニケーションをどのように進めていくかというルールそのものをコミュニケーションを進めながら決めていくという自然発生的なルールを重視するという姿勢をとっており、その意味でこのような姿勢こそが Wiki の最も特徴的な点であると考えられることができるだろう。Wiki の設計思想の根本的な部分を一言で表現するとすると「コミュニケーション・パターンの重視」とであると言える。

4.1.3 qwikWebにおけるコミュニケーション・パターン

前節で再設定したWikiの設計思想「コミュニケーション・パターン」を基にqwikWebの設計思想を検討すると、「ユーザが各種のコミュニケーション・パターンを独自に組みあわせることで当該グループに適したコラボレーションシステムを構築していくことが可能なシステム」となる。その上で、「今までの知識をもとに説明なしで使うことができる」というユーザの立場を加味すると、すでに慣れ親しんだメールによるコミュニケーション環境を出発点として、徐々に自分の理想とする情報環境を実現できるような発展性を備えたコミュニケーション・パターンを実践することができるようなWebシステムとなる。

4.2 qwikWebのシステム設計

4.2.1 基本設計

今まで議論してきた「管理の容易さ」「習得の容易さ」「知識の蓄積・構造化」を考慮し、我々のターゲットユーザが利用している日常的なコミュニケーション手段を検討する。それは、主に電話、FAX、携帯電話、メール、Webブラウジングである。この中でテキスト情報などのインターネットで流通させる場合に整合性の良いデジタル情報によるコミュニケーション手段は、メール、Webブラウジングである。前節で設計思想の面で採用を決定したWikiはWebブラウジングの拡張であり、メールの使い方には何の変化ももたらさないため、ユーザは従来通りの手段でメールを使い続けると想定される。そこで、メールと知識の構造化、協働構築に利用されるWikiを連携することとする。つまり、Wikiの入口の画面を含めて、利用者が好みの利用規則を設定し、自由に編集することができるコミュニケーション・パターンを実践することができるWebシステムとして設計する。具体的には以下の点を重視して設計する。

- ・「管理の容易さ」

メールを送って、システムからのメールに返信するだけで開設できる。メンバの追加や削除など誰でも可能とし、従来のコラボレーションシステムでは困難であった組織外のメンバとの連携もすぐに行える。システムのメンバとWikiのメンバを一致させ、誰でも容易に管理できるようにする。

- ・「習得の容易さ」

メールによる手軽なコミュニケーションを入口として、徐々に自分が望む情報環境へと進めていくことができる。

- ・「知識の蓄積・構造化」

メールのやり取りがWikiに蓄積され、それを基に必要な知識構造化や協働作業を行うことができる。Wikiに対して誰がどんな操作をしたか通知される（メールで配信される）。

このため、グループにメールを配信するMLとWikiを密接に連携する設計とする。MLとは、複数の人に同時に電子メールを配信する仕組みである。ML宛にメールを送信すると、そのメールは登録されている全てのメンバに転送される。MLは非同期型のコミュニケーションシステムに分類できる。ML用のサーバソフトウェアにはfmlやmajordomoなど数多くあるが、設置の手間などから、現在はYahooグループ^[13]やfreeML^[14]などのホスティングサービスが多く用いられている。

MLを開設する際には、まずMLの管理者となるユーザがMLのメールアドレスを設定し、さらにメンバの登録を行う。一般にメンバの参加および脱退はMLの管理者によって管理されており、管理者へ依頼を出すことで参加/脱退が行われる。これには管理者が権限を持つことで、問題のある人物の排除が容易に行える一方、管理者に多くの負荷がかかるという問題がある。

そこで、MLとしてQuickML^[15]を採用する。このQuickMLは、平易な操作でMLの立ち上げや運用が行えるMLシステムである。

4.2.2 QuickML

QuickMLは高林哲と増井俊之らによって開発された、容易に開設・運用が行えるML管理システムである。QuickMLではメンバ全員がメンバの参加/脱退操作の権限を持つため、管理者1人に負荷が集中するという問題が発生しない。メンバが投稿したメールは各メンバに転送されるため、メールは各メンバがクライアント側で管理することになる。

5 qwikWebの実装

5.1 実装手法

qwikWebの実装では、要素技術としてQuickMLとWiki技術を用いているが、ユーザにとっては、MLとWikiの機能が巧みに連携したシステムと見え、QuickMLや従来のWikiを意識しないよう実装した。qwikWebでは、MLに送られたメールをすべてWikiページとして登録していくことで、知識の構造化を促進する。

ユーザはqwikWebシステムにメールを送るだけでMLとWikiサイトを開設できる。まずMLの名前を決め、「MLの名前@qwik.jp」にメールを送信する。その名前が未使用であれば確認メールが送られてくる。このメールに返信することで、MLとWikiサイトが開設される（図2）。もしMLとWikiサイトが1ヶ月以上使われなかった場合、警告の後に自動的に削除される。このように開設と閉鎖に際してユーザにほとんど負荷をかけない仕組みであり、ユーザはメールを投げる感覚でMLおよびWikiサイトの立ち上

げを行える。

ML に送られたメールは、新たな Wiki ページとして Wiki サイトに自動的に蓄積される。メールの件名がそのまま Wiki ページのページ名となり、同じページ名が既に存在していた場合はその Wiki ページの最後に新しいメールが付け加えられる。そのメールに対して返信したメールも同様に親メールの Wiki ページの最後に付け加えられる。そして同じ Wiki ページ内のメールは、日付と投稿したユーザ名で区切られて表示される。このように関連するメールは 1 つの Wiki ページに蓄積される。作成された Wiki サイトは ML のメンバだけがアクセス可能である。メンバの追加はメールでも Wiki でもいずれからでも可能である。ML 宛のメールに追加したいメンバのメールアドレスを cc するだけでメンバ追加は完了する。Wiki からは、メンバリストの Wiki ページを直接編集することでメンバの追加が行える。このメンバ管理操作はメンバであれば誰でも行うことができ、特定の管理者だけに負荷がかかることはない。

qwikWeb において Wiki はアーカイバの役割を果たすが、一般的な ML のアーカイバと異なり、Wiki であるために編集できる。例えば、わざわざメールで連絡するほどでもない小さな誤植も Wiki ページを直接編集すれば容易に修正できる。他には「日程調整するので、下記の候補日のうち都合の良い日に丸を付けてください」といったメールを ML に投稿した場合、返信せずとも Wiki ページにてそのメールに直接自分の都合の良い日を書き込むといったこともできる。メールにファイルが添付されていた場合は、Wiki 上にてそのファイルが共有される。

qwikWeb では、このように ML で行われたやりとりからシームレスに共同編集作業へと移行できる。なお、メールと違って Wiki は全員がチェックしているとは限らないため、このような編集が行われたことが他のユーザに気づかれない可能性がある。そこで qwikWeb では定期的な

Wiki ページの更新情報（更新したユーザおよび更新時刻）を ML にシステムが自動投稿する機能を備えている。また、全ての Wiki ページはバージョン管理されているため、誰かが誤って削除しても復帰可能である。さらに Wiki ページの編集状況を容易に閲覧できるタイムマシーン機能も付いている。これはスライドバーを左右に動かすことで、Wiki ページの時間変化を連続的に見ていくことができる機能である。

ユーザ参加型の場合は、セキュリティの問題が懸念される。そこでこの点も細心の注意を払って設計した。例えば、ID をメールアドレスとし、パスワードはシステムが自動生成したものにしているため、他人のメールアドレスを入力してもパスワードを入手できないため、なりすましは困難である。パスワードを忘れた場合は、メールアドレスを入力すると送られてくるが、そのメールアドレスに送られるため基本的に本人しか受け取れない。もちろん、パスワードを盗まれたり誤って他者に送ってしまった場合は、他の多くのシステムと同様になりすましが可能となる。しかし、なりすまして Web サイトを修正した場合には、その修正結果がメンバーに通知されるため、本人は他者が自分の ID とパスワードで不当な操作を行ったことが分かることになる。すべての Web ページは過去の履歴を保有しているため、希望の状態まで復帰可能であり、ID 登録を削除すれば今後なりすましをされることはなくなる。誤った入力をしてしまった場合にも過去の履歴を保有しているため容易に復旧可能である。

5.2 実装上の工夫

qwikWeb の開発は、単に実用的な Web システムとしてではなく、「Web システムとして理想的な解は何か」を考え、その理想にできるだけ近付けるよう開発を進めていった。具体的には、URL の設計において、Roy Fielding による REST Architectural Style に準拠した。qwikWeb 開発当初は使いやすい Web アプリケーションフレームワークは存在していなかったため、テンプレートやプラグインなどといったフレームワークに相当する部分から含めて自作した。インストールや保守を簡単にするために、あえて DB ではなくファイルシステムのみをストレージとして用いる仕組みとした。

実用に供するシステムとして安定度を高めることと、研究のためのシステムとして随時機能追加を行うことは、一般には相反する行為である。機能を追加すれば自ずとバグも発生し、システムは不安定になる。しかしその両者を同時に実現しなければ、研究であることと実用であることを同時に満たすことはできない。この相反する要求を実現するために、徹底した機能テストを記述することで極力バグを



図2 開設された Wiki MLでのやりとりがそのまま Wiki ページとなっている。

少なくすることにした。今回は、Webアプリケーションフレームワークを自作しており、その際にテストフレームワークも自作しているため、通常よりもWebシステムの機能テストを容易に記述することができるようになった。開発を進める際にも常に機能テストを先に記述するようにしたため、機能テストの伴わない機能はほとんど存在していない。このような工夫によって、長期間にわたってほとんどダウンタイム無しに運用し続けられる体制を実現することができた。開発・運用共に筆者（江渡）1人だけで、4年半に渡って運用を継続することができた。

6 qwikWebの運用

6.1 qwikWebのアクセス解析

実装したqwikWebを用いて2003年8月から研究ベースの運用を開始し、現在も稼働中である。ユーザは利用時に利用規約（個人を特定できない形でデータが研究上利用されることなど）を承認する必要がある。

図3には、ML数、閉鎖ML数、Wiki利用ML数の累積値を示した。2007年5月現在でMLの数は3,110、ユーザ数は18,519人である（開発者が開設したMLのデータは除く）。各MLのユーザ数は、最大が648人、平均は7.7人であった。半数以上が10人以下のMLであり、少人数でも手軽にMLを利用するという設計意図に合った利用状況といえる。また約70%のMLにてWikiが利用されており多くのユーザはWikiも利用できる点を考えてqwikWebを選択していることがわかる。Wikiを利用したML（2,235個）の平均活動期間は145.1日であるのに対し、利用していないML（875個）は42.4日であった。長期間利用するものほどWikiが利用されていることがわかる。閉鎖ML数は現状で約10%となっており、QuickMLにおける閉鎖ML数の50%と比較して低い水準となっており、知識蓄積に貢献していると考えられる。

ユーザ数の累積値を図4に示した。ユーザ数、Wiki利用ユーザ数、ML作成ユーザ数、いずれも純増している。

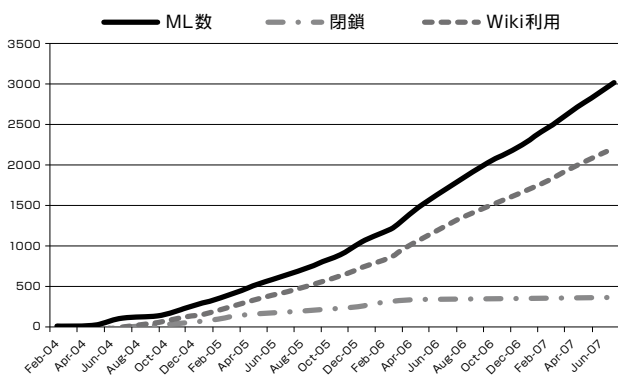


図3 ML数の時間遷移

Wiki利用MLが全体の約70%であり、Wiki利用ユーザは約40%となった。これは多くのユーザがMLから誘導されてWikiを利用していたためと考えられる。MLを作成したユーザ数は約10%（1,909人）となっている。1人が作成したML数の最大は40個、平均は1.6個であった。1人で最大40個ものMLを立ち上げるユーザがいることから、管理者のコストが比較的安く抑えられていることが示されていると考える。約25%にあたる505人が2つ以上のMLを作成している。一方、参加しているML数を見ると、18,519人のうち参加ML数最大は48個、平均は1.3個であった。こちらも約20%にあたる3,515人が2つ以上のMLに参加していた。複数のMLに参加するということは、qwikWebをある程度評価しているためと考えられる。

WikiとMLの違いを、時間帯ごとの利用状況の違いから調べた。図5は時間帯ごとの投稿数、Wikiアクセス数、Wiki編集数である。MLへの投稿には、9～11時、12～13時、17～19時、そして23～24時にピークがあることがわかる。それぞれ就業開始、昼休み後、就業終了、就寝前に当てはまると考えられる。対してWikiアクセス数や編集は、午後の就業開始から就業終了まで増加していき、17～18時にピークを迎える。これは、MLへの投稿はコミュニケーションであるため、生活における区切りの開

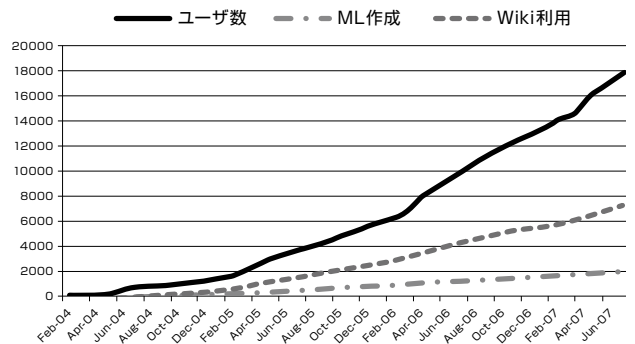


図4 ユーザ数の時間遷移

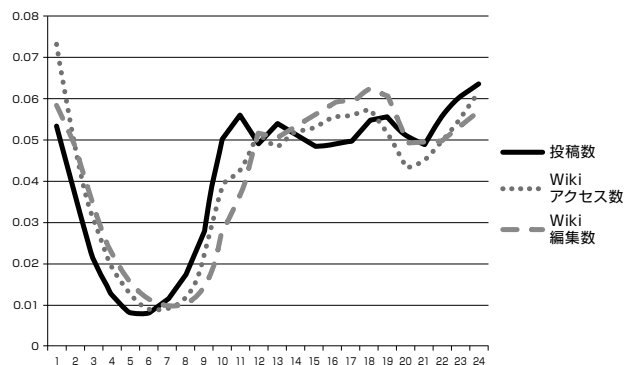


図5 時間帯ごとのMLとWikiとの利用状況の違い
グラフの実線は1日全体の投稿数を1.0としたときの時間帯ごとの投稿数が占める割合を示している。Wikiアクセス数、Wiki編集数も同様である。

始と終了時に頻繁に行われるが、Wiki へのアクセスや編集は作業と密接に関わるため午後の就業時間に頻繁に行われたり、就業終了時や就寝前などの区切りがついた時点でまとめ作業の場として用いられたりしたのではないかと考えられる。

次に、Wiki も ML も利用しているグループでの利用状況について見てみる。ここでは投稿数が 50 件以上、Wiki ページ編集回数 50 回以上、継続期間が 1 年以上の ML に注目した。この条件を満たす ML は全 3,110 個のうち 64 個あった。各月の投稿数と編集回数の大小を比べてみたところ、8 割にあたる 54 個の ML が編集回数よりも投稿数の方が多いう月が多かった。図 6 は幾つかの ML における投稿数、編集回数の累積値を示した値である。図 6-a は ML 中心、図 6-b は Wiki 中心の例である。全体的に最初の数ヶ月が最も活動的（投稿数も編集回数も多い）でそれ以降は少なくなるという傾向が多く見られた。特に Wiki はその傾向が強く、最初に編集があった後は小康状態に落ち着くことが多い。しかし幾つかの ML では突然 Wiki 編集回数が増えるという例もみられた。図 6-c1 および c2 はコンスタントに増加する投稿数に対して、Wiki の編集回数は一時期急増した沈静化したという例である。日常業務での連絡、協働知識構造化として利用している場合には ML も Wiki も定常的に利用され、特定のイベントのために利用

している場合にはそのイベントの前後での利用が多くなっていると考えられる。

以上、qwikWeb の利用状況と ML と Wiki の利用のされ方の違いについて運用結果データを基に述べた。研究ベースの運用にも関わらず qwikWeb 設計と実装の有効性が示せた。

6.2 qwikWebの実践例

技術評論社「Software Design」誌 2005 年 8 月号より 2006 年 7 月号まで、12 回に渡って Wiki に関する記事が連載され、連載の 1 つとして「qwikWeb 徹底解説」^[16] が掲載された。同時に連載を支えるコラボレーションシステムとしても qwikWeb は利用され、最終回において qwikWeb を利用した編集作業の実態^[17] が紹介された。連載において利用された qwikWeb のグループは、同様な利用者の参考になるように一般公開された^[18]。記事によれば、qwikWeb が選ばれたのは「後から追加されたメンバにもプロジェクトの見通しが良いこと」、「メールとドキュメントの両方が閲覧可能であること」などの条件があげられ、qwikWeb の目標が実現されていることがわかる。このことから、qwikWeb が研究ベースの運用であるにもかかわらず、実運用にも使えるクオリティを発揮していたことが確認できる。

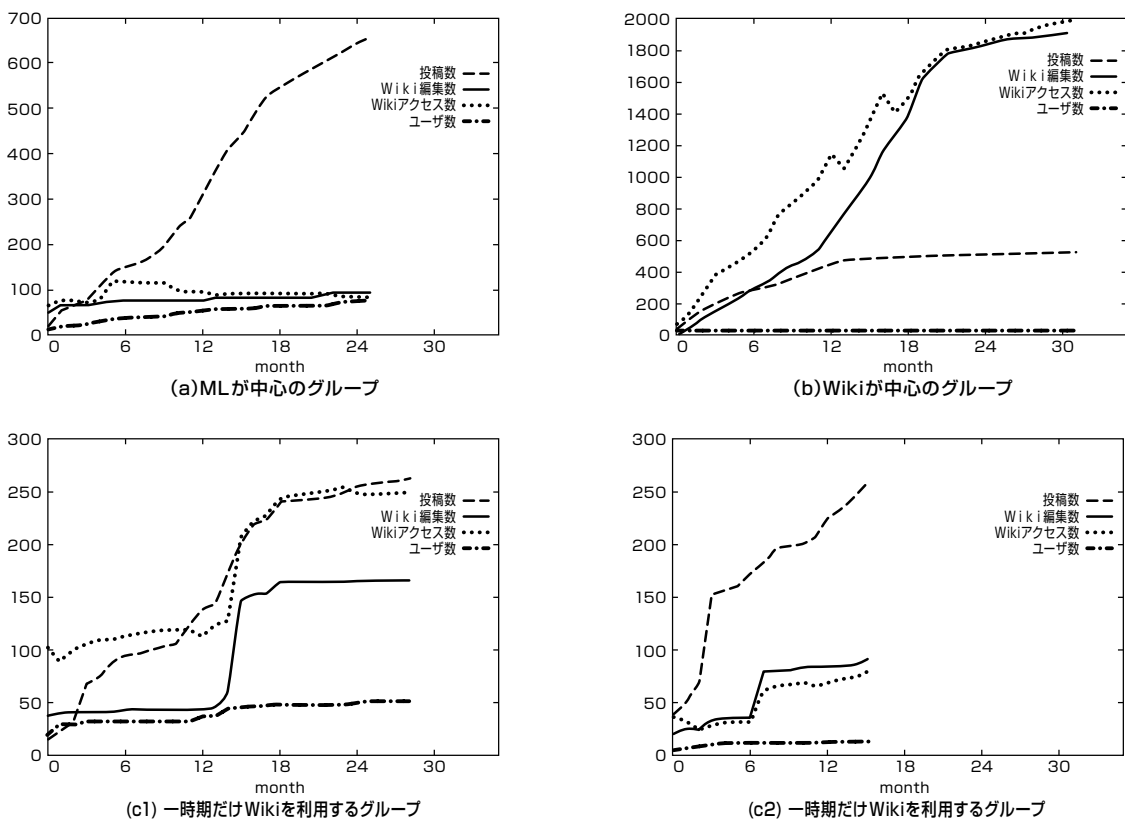


図6 グループによる利用状況の違い

7 関連研究

Wikiの基本コンセプトは誰もが容易に内容を編集できることであり、このコンセプトを踏襲したWikiクローンと呼ばれる同様のシステムが多数存在する。Wikiをより便利に使うために、メールとの連携機能を備えたWikiエンジンもある。JotSpot^[19]では、それぞれのWikiページに対応するメールアドレスにメールを送ることによって、Wikiページを更新できる。PukiWiki^[20]は、受け取ったメールをページへの書き込みとして扱う機能を追加できる。Hiki^[21]は、更新をメールで通知する機能を備えている。いずれもメールをWikiの補完的な機能として用いているが、qwikWebはMLと密に統合されているという点で大きく異なる。

目的を特化したWikiも多く存在する。おそらく最も著名なものはMediaWikiであろう。これはオンライン百科事典のWikipediaに用いられているWikiエンジンである。他にもSemantic Webデータを容易に作成するツールとしてSemanticWikiと呼ばれるWikiエンジンもいくつか提案されている^{[22][23]}。教育用途を目的としたWikiエンジンもある。Guzdialらは、ジョージア工科大学において、CoWebというWikiエンジンを開発し、教育環境において使用している^[24]。2年間の運用において、学内には120を超えるWikiが立ち上げられ、10台のサーバで運営されている。Wangらは、Wikiにページ所有者、書き込み権限、不可視モードの設定などの学習環境に特有な拡張を施した^[25]。Breretonらは、Wikiをベースとした学習支援環境を構築している^[26]。しかし、qwikWebのようにMLとのシームレスな融合システムは提案されていない。

8 まとめ

誰でも構築運営できるコラボレーションシステムの実現に向けて、情報伝達と知識の集約や構造化技術をスムーズに融合したqwikWebを提案した。また、このようなコラボレーションシステムの本格研究を実現するプロセスを明らかにした。本システムをデザイン、実装、運用した結果、約18,000人のユーザが約3,000のMLを利用した。運用データの分析を行うことで本システムの妥当性と有効性を示した。

今後の課題としては、企業による本格運用、ターゲットユーザの絞込みによる改良、携帯電話など、多様な情報環境に向けての改良などが挙げられる。

筆者（江渡）が手掛けている他のプロジェクトとして、仮想生物構築環境Modulobeがある。Modulobeでは、ユーザが作った仮想生物をネット上で共有し、他のユーザはそのモデルを改造して新しいモデルを作り上げることができる。モデルのパーツレベルでつけられたIDによって、モデ

ル間の再利用の関係が可視化される。ユーザ間のコミュニケーションを支援するシステムを運営し、そこから知見を引き出そうとしている点で共通点がある。

qwikWebとModulobeには、また「引き算によるデザイン」という共通点がある。ユーザにとって本質的に重要な機能は何かを考え、それ以外の機能は徹底的に削る。それによってターゲットとなるユーザがシステムを把握しやすくする。qwikWebでは、管理者を設定する機能がない、ページ毎に編集権限を変える機能がないなど、通常存在する機能を削ることでわかりやすさを実現した。これにより利用できない状況も出てくるが、新しいユーザがすぐに理解して使いこなせるようなシステムが実現できた。

今後はよりユーザ間のコミュニケーションを支援し、ネットワーク上の共有知の実現に寄与するような研究を続けていきたい。

謝辞

本研究の一部は、科学技術振興事業団「JST」の戦略的基礎研究推進事業「CREST」における研究領域「デジタルメディア作品の制作を支援する基盤技術」の研究課題「情報デザインによる市民芸術創出プラットフォームの構築」の支援により行いました。

キーワード

コラボレーションシステム、グループウェア、パターン・ランゲージ、Wiki、メーリングリスト

参考文献

- [1] 総務省: 平成17年「通信利用動向調査」の結果, http://www.soumu.go.jp/s-news/2006/060519_1.html (2006).
- [2] 江渡浩一郎, 高林 哲, 増井俊之: qwikWeb: メーリングリストとWikiを統合したコミュニケーション・システム, *インタラクティブ2005*, 13-20 (2005).
- [3] Lotus Notes: <http://www-06.ibm.com/jp/software/lotus/>
- [4] サイボウズ: <http://cybozu.co.jp/>
- [5] 岡田謙一: 協調作業におけるコミュニケーション支援, *電子情報通信学会誌*, 89 (3), 213-217 (2006).
- [6] 北山 聡: コミュニティを計量する, *人工知能学会誌*, 18 (6), 668-674 (2003).
- [7] B.Leuf and W.Cunningham: *The Wiki Way: Quick Collaboration on the Web*, Addison-Wesley, Reading, MA (2001).
- [8] 江渡浩一郎: なぜそんなにもWikiは重要なのか, *Mobile Society Review 未来心理*, 7, 50-57 (2006).
- [9] C.Alexander, S.Ishikawa and M.Silverstein: *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, New York (1977).
- [10] K.Beck and W.Cunningham: Using pattern languages for object-oriented programs, Technical Report No. CR-87-43, <http://c2.com/doc/oopsla87.html> (1987).
- [11] E.Gamma, R.Helm, R.Johnson and J.Vlissides: Design patterns: Abstraction and reuse of object-oriented

design, *Proceedings of ECOOP'93*, Kaiserslautern, Germany, 406-431 (1993).

- [12] <http://c2.com/cgi/wiki?WikiWikiHyperCard>
- [13] Yahoo! グループ: <http://groups.yahoo.co.jp/>
- [14] freeML: <http://www.freeml.com/>
- [15] 高林 哲, 増井俊之: QuickML: 手軽なグループコミュニケーションツール, *情報処理学会論文誌*, 44 (11), 2608-2616 (2003).
- [16] 江渡浩一郎: qwikWeb徹底解説, *Software Design 2006年5月号*, 102-111 (2006).
- [17] しばむらしのぶ: qwikWeb上のWikiつまみぐい—Wikiつまみぐい運用の舞台裏, *Software Design 2006年7月号*, 113-115 (2006).
- [18] <http://qwik.jp/wikibana-gihyo/>
- [19] JotSpot: <http://www.jot.com/>
- [20] PukiWiki: <http://pukiwiki.org/>
- [21] Hiki: <http://www.namaraii.com/hiki/>
- [22] 武田英明, ムリアディ・ヘンドリー: Semantic MediaWikiの構築に向けて, *人工知能学会研究会資料*, SIG-SWO-A404-06, 06-01-06-03 (2004).
- [23] 河本健作, 北村泰彦: Semantic WikiによるRDF自動生成, *人工知能学会研究会資料*, SIG-SWO-A501-02, 02-01-02-06 (2005).
- [24] M.Guzdial, J.Rick and B.Kerimbaev: Recognizing and supporting roles in CSCW, *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 261-268 (2000).
- [25] C.Wang and D.Turner: Extending the Wiki paradigm for use in the classroom, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2, 255-261 (2004).
- [26] M.Brereton, J.Donovan and S.Viller: Talking about watching: Using the video card game and wiki-web technology to engage IT students in developing observational skills, *Proceedings of the fifth Australasian conference on Computing education*, 197-205 (2003).

(受付日 2007.12.26, 改訂受理日 2008.3.10)

執筆者略歴

江渡 浩一郎 (えと こういちろう)

1997年 慶應義塾大学大学院政策・メディア研究科修了。同年、国際メディア研究財団に所属。メディアアーティストとしてネットワーク上のコミュニケーションをテーマとした作品制作活動を行う。1996年「WebHopper」を制作。1997年 アルス・エレクトロニカ賞グランプリを受賞 (sensorium プロジェクトとして)。1998年 Canon ARTLABとの共同制作として「SoundCreatures」を制作。1999年 アルス・エレクトロニカ賞 Honorary Mentionを受賞。2002年 産業技術総合研究所に所属。2003年より、「qwikWeb」の開発・運用を継続する。2005年、仮想生物構築環境「Modulobe」を発表する。ネットワーク上の共有知を支援する情報共有環境に興味を持つ。修士 (政策・メディア)。情報処理学会会員。本論文では主として qwikWeb の設計、開発および試験運用、論文作成を行った。

濱崎 雅弘 (はまさき まさひろ)

2000年 同志社大学工学部知識工学科卒業。2002年 奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。2005年 総合研究大学院大学数物科学研究科博士後期課程修了。博士 (情報学)。同年より、産業技術総合研究所情報技術研究部門勤務。情報推薦やオンラインコミュニティの研究に従事。人のネットワークを活用した情報システムに興味がある。人工知能学会、電子情報通信学会、

ACM、各会員。本論文では主として qwikWeb の試験運用データ解析および論文作成を行った。

西村 拓一 (にしむら たくいち)

1992年 東京大学工学系大学院修士 (計測工学) 課程修了。同年、NKK (株) 入社。X 線、音響・振動制御関係の研究開発に従事。1995年 技術研究組合新情報処理開発機構つくば研究センタに向向、1998年 NKK (株) 復帰。1999年 技術研究組合新情報処理開発機構つくば研究センタに所属。2001年 産業技術総合研究所サイバーアシスト研究センターに所属、2005年 同情報技術研究部門実世界指向インタラクショングループ長、同年筑波大学大学院知能機能システム専攻 助教授 (連携大学院) を併任、現在に至る。博士 (工学)。時系列データ検索・認識、実世界情報支援に興味を持つ。電子情報通信学会、情報処理学会、人工知能学会ヒューマンインタフェース学会、赤外線学会、ACM、各会員。本論文では主として本研究の位置づけおよび論旨展開の構築、論文作成を行った。

査読者との議論

議論1 ユーザ参加型の開発改良について

質問 (大蒔 和仁)

「システム設計には、利用者である末端ユーザも直接関与する仕組みを用意することが大変重要である」という観点が重要だと思いますので、その点を明確に記載されたらどうでしょうか。

回答 (江渡 浩一郎)

ご指摘の通り、情報システムではこのユーザ参加型の開発改良プロセスが重要な点ですので、「2. Web システムの本格研究」をより分かりやすくしました。末端ユーザの直接関与は非常に重要な点なのですが、今回の開発においては、ユーザにシステムを使っていたいただき、その様子を細かく観察し、観察結果を元に必要な機能を考えるという手順をとりました。ユーザはシステムに関しては素人なので、本当に必要な機能を提案できるかという点、必ずしもそうではありません。逆に末端ユーザは彼ら自身の仕事については専門家なので、彼ら自身の仕事のプロセスを元に開発者が必要な機能を決定いたしました。この点は2章に関連部分を記述いたしました。

議論2 ソフトウェアの権利関係について

質問 (大蒔 和仁)

qwikWeb の権利関係はクリアになっているのでしょうか。

回答 (江渡 浩一郎)

qwikWeb に関しては、直接開発に関与した人員は産総研職員のみであり、他者との権利関係は明確となっております。システムの一部として QuickML を利用しておりますが、こちらは GPL に基づいてフリーソフトウェアとして公開しているもので、qwikWeb も同様に GPL を選択していますので、権利関係上問題はありません。

議論3 ソフトウェアの学習について

質問 (大蒔 和仁)

査読者が、Wiki を使って感じるの、簡単だとはいつても「パターンランゲージ」をある程度学ばなければならない点です。このトレードオフをどう考えるか、という点の言及が欲しいと思います。

回答 (江渡 浩一郎)

良い御指摘ありがとうございます。まずパターンランゲージは Wiki の背景となる思想で、それを学ばないと Wiki を使えないわけではなく、簡単な使い方を学習すればまず使い始めることは誰にでもできます。その上で Wiki を Wiki らしく使うためには御指摘の通りある程度はパターンランゲージを学ばないといけないのですが、それは Wiki の背景となる思想を身につける必要があるということであり、

徐々に修得していくことができるという種類のものです。逆に言えば、そのように Wiki を使いこなしていくことで、徐々にパターンランゲージの思想に触れることができるようになっており、それが Wiki の利点の1つであると考えています。

Wiki では、Wiki 記法という独自のマークアップランゲージを使って文章の構造を明示する必要があるのですが、たしかにその学習のための負担も無視することはできません。しかし、Wiki 記法を使って文章を書いていくことによって、文章の構造に自覚的になるという利点もあります。実際に大学の授業で Wiki を教える際には、そのように文章を構造化して書く利点と共に Wiki 記法を教えることによって、自然と Wiki 記法を修得させることができました。トレードオフですが、Wiki 記法にはそのような利点もあります。

この点を御指摘の通り、説明を加えました。

議論 4 開発の目標とシナリオと要素技術の選択プロセスについて

質問（赤松 幹之）

「2.3 節」の最後にポイントとして4点を掲げていますが、1点目を除いて、ML と Wiki の連携が前提となった説明になっていました。できれば、この4点をシステム設計上の目標とする4点として記載して、それぞれを満たすために検討した結果、ML と Wiki の連携という方針で行くことにした、というシナリオを書けると良いと思います。（「シナリオ」の記載）同時に、要素技術の選択プロセスがあったと思いますので、ML や Wiki 以外を検討した結果、それらは採用しなかった、という「選択のプロセス」を書いていただければと思います。

回答（江渡 浩一郎）

シナリオに関して、ご指摘の通り元の論文では明確になっておりませんでしたので、図1のプロセスに則った章だてにし、現状のコラボレーションの問題点と解決策（ご指摘くださった「管理の容易さ」と「知識の蓄積・構造化」）、その解決のための基本思想、その基本思想に

近い Wiki を選択、システム設計における基本設計で QuickML を採用という流れに修正いたしました。選択のプロセスについては、4.1.1 にて要素技術として Wiki を選択した理由を述べ、4.1.2 にて Wiki について説明し、4.2.1 にて要素技術として QuickML を選択した理由を述べ、4.2.2 にて QuickML について説明しております。

議論 5 運用における改良プロセスについて

質問（赤松 幹之）

「4. qwikWeb の運用」において、ユーザの利用データの分析の結果を示していますが、これが図1のなかの、フィードバック情報の分析、システム修正、システム運用、のサイクルに対応したセクションであると思われますが、このサイクルによってシステムの修正がどのように行われたかが書かれていません。そういった改良とのつながりが分かるようにできますでしょうか？

回答（江渡 浩一郎）

「6.2 qwikWeb の実践例」を設けて代表例を説明いたしました。この「フィードバック」に関しては図1の説明と対応させました。

議論 6 研究開発の困難性について

質問（赤松 幹之）

この論文を読むと、簡単に qwikWeb が実現できたかのような印象を持ちます。実際には種々の苦勞があり、それらを解決した上で実現できているのではないかと想像します。そういった技術的な困難さがどこにあって、それをどのように解決したのかの記述があると、読者の大いなる参考になると思います。

回答（江渡 浩一郎）

5.2 節に実装上の苦勞と運営上の苦勞を記述いたしました。