

## ユーザ間の会話とファイル更新履歴のタイムライン化による 同期的協調作業支援

北田 哲也†  
†立命館大学情報理工学部

塩見 和則‡ 高田 秀志†  
‡立命館大学大学院情報理工学研究所

### 1 はじめに

近年、インターネットの普及により、ネットワーク上での協調作業が一般的となった。ネットワーク上での協調作業の例として、例えばプレゼンテーションの資料作りのために、ネットワークを介してユーザ間で会話を行ったり、必要な資料を送信したりするようなものが挙げられる。このような作業のための会話やファイル送受信の方法として、電子メールを利用する非同期的な方法がある。この方法では、データを送信したい相手を指定して必要な資料を添付し、連絡事項を書いた後に送信するという手間がかかる。また、送信された相手は添付資料と連絡事項を確認し返信する必要がある。

一方で、Dropboxなどのファイル共有サービスが登場してからは、多数のユーザでファイルを共有し、お互いに編集しあうといった同期的な作業が可能になった。しかし、ファイル共有サービスによる協調作業には会話機能が不足しているため、LINEのようなチャットツールをあわせて利用する必要があるが、ファイル更新履歴とチャットツールのタイムラインが分かれてしまい、進捗を確認することが困難になる。そのため、円滑な協調作業が行えない。

そこで、本研究では、ユーザ間の会話とファイル更新履歴を同一のタイムライン上に時系列としてユーザに提示することによって同期的協調作業の円滑化を図る。

### 2 研究背景

#### 2.1 同期型協調作業支援の問題点

協調作業を行う際に必要な要素として、ファイルなどのデータを共有することや、ユーザ間での会話ができることが挙げられる。チャットツールとDropboxのようなファイル共有サービスを用いて協調作業を行う場合、チャットツールでは、会話のやり取りを行う。これに対してファイル共有サービスでは共有されているファイルが自動で同期されてデスクトップ上に通知される。この時、チャットツールにはファイルを更新したという通知は残らないため、ユーザ自身が自主的にファイル更新に関する情報を書き込む必要があり、不便である。ユーザによってはこの作業自体を行わない場合もある。また、ファイル共有サービス上の更新履歴単体では作業の具体的な内容を把握するのは難しい。このような問題点から協調作業が円滑に行われていない現状がある。

同期型協調作業支援の研究としては、ユーザ間のコンピュータの環境の違いによって協調作業ができない問題を、ブラウザで動作するデバイスに依存しないシステムによって解決しようとしたもの[1]があるが、データの更新と会話のタイムラインが別々になってしまう問題については触れられていない。

### 2.2 研究目的

本研究では、上記のような問題点を解決するため、ユーザが会話しているチャットツールのタイムライン上に作業に伴って行われたファイル共有サービス上のファイル更新を自動的に通知することで、作業に関する情報を自動的に通知しチャットツールのタイムラインとファイル更新のタイムラインを一体化するツールを提案する。これによりユーザが作業状況を報告する手間の軽減や、作業の進捗状況の確認を容易化し、協調作業の円滑化を図る。

### 3 タイムライン化機能

#### 3.1 機能

システムの全体構成は図1の通りである。各ユーザの端末には、会話を行うためのTwitterクライアントとファイル共有を行うためのDropboxクライアントがインストールされている。また、Dropbox上のファイル更新をTwitterのタイムライン上に通知するために管理用PCを設け、管理用PC上では本研究で開発する「タイムライン化ツール」が動作している。このタイムライン化ツールは、Dropboxからファイル更新履歴が格納されたRSS情報を受信し、更新情報をTwitterサーバへ投稿する。

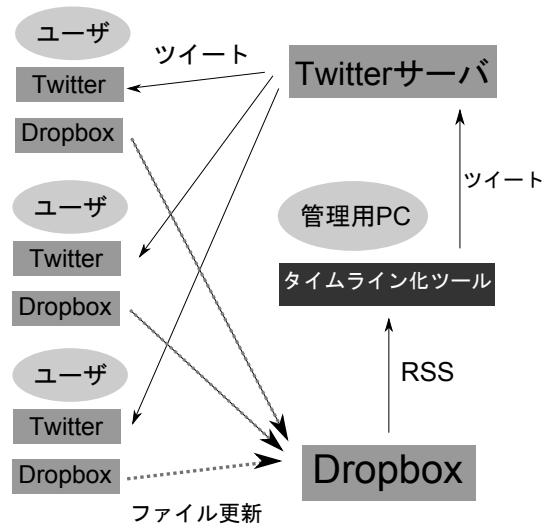


図1: 全体構成

Dropboxはファイルが更新された時、RSSとして更新情報を出力する機能を持つ。図2は出力されたRSSの一例である。

Dropboxから受信したRSS情報を基に、図3の2行目のような「(ファイル名)が更新されました」という形式のツイート文を作成し、Twitterのタイムライン上にとどのファイルが更新されたかを通知する。図2の例では、1行目の作業報告に対して2行目の更新通知が行われたことを示している。

Synchronous Cooperative Work Support with Timeline of Communication among Users and History of File Updates

†Tetsuya Kitada ‡Kazunori Shiomi †Hideyuki Takada

†College of Information Science and Engineering, Ritsumeikan University

‡Graduate School of Information Science and Engineering, Ritsumeikan University

```
<link> https://www.dropbox.com/home
</link>
<pubDate> Tue, 06 Jan 2015 06:59:14 GMT
</pubDate>
<description>
<![CDATA[
<a target='_blank' href='/event_details/368764674/781152885/454085459/0'
?#q37504;#q39307;#q23546;.odp'
</a>
ファイルを追加しました。<br />
]]>
</description>
<guid isPermaLink='false'>
DROPB0X_270888F0B7EB3BD1073B5024B216B8A8
</guid>
```

図 2: RSS の内容

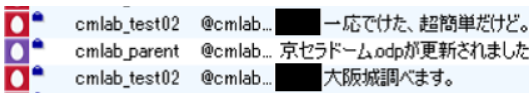


図 3: 会話中での通知

### 3.2 実装

実装は Java で行い、ライブラリとして Twitter にアクセスするための twitter4j と RSS を読み込むための informax を使用している。また RSS 内のテキストの抜き出しには jsoup を利用している。

タイムライン化ツールが起動されると、まず、Twitter にログインする。ログイン後、twitter のタイムライン上にツールを起動した旨の通知を行う。また、Dropbox の RSS の URL を設定する。ログインが完了したら、5 秒おきにファイルが更新されたかどうかを Dropbox から取得する RSS 情報の pubDate を基に判定する。この情報が変化していた場合更新が行われている。更新が行われていた場合、RSS 情報の description からファイル名を抜き出し、抜き出したファイル名と「が更新されました。」の文字列と結合し「(ファイル名)が更新されました。」という文章を作成する。このようにして作成した文章を twitter に送信する。

## 4 評価実験

### 4.1 実験目的

本実験の目的は、ユーザの会話のタイムライン上にファイル更新履歴を表示することにより、ネットワーク上で行う協調作業が円滑に行われたかを検証することである。また、本システムを利用することにより発生する、メンバが行う協調作業の変化について調べる。

### 4.2 実験内容

本実験は情報系学生 3 人一組で構成される 2 つのグループで実施した。実験上の都合により、今回は同一の部屋で実験を行ったが、実際にはネットワーク上での作業を想定しているため、各ユーザが直接会話できないように配置した上、会話手段を Twitter に限定した。本実験では、Twitter のアカウントとしてユーザが作業を行うためのユーザ用アカウントと更新情報の通知と実験を観測するための管理用アカウントを用意した。また、Twitter クライアントとして tween を利用した。

作業では、ユーザ同士である地域に対する紹介資料をスライドとして作成してもらった。比較実験をするために、タイムライン化ツールによる更新通知がある場合とタイムライン化ツールがなく更新通知がない場合の 2 種類を用意する。片方のグループでは更新通知がない場合の作業を初めに行い、更新通知がある場合の作業を後に行った。もう一方のグループでは、これを逆順に行った。所要時間は更新通知がある場合と更新通知がない場合でそれぞれ 20 分程度である。また、

ユーザの行動の変化を確かめるためユーザが使用している PC の画面をキャプチャした。実験終了時には簡単なアンケートを実施し、利便性を調査した。実験の流れは以下のとおりである。まず、被験者全員に実験の説明を行った。説明後、被験者は、Twitter 上に投稿される作業開始のツイートに従って初めの作業を開始する。スライドが完成したら作業を終了する。次にもう一方の作業を同様に行う。その後、被験者に対してアンケートを行う。アンケートの内容は表 1 のとおりである。

表 1: アンケート内容

項目	形式
通知がある場合に通知がない場合と比べて便利であるように感じたか	5 段階評価 5(とても良い)~1(とても悪い)
ファイルの更新データのほかに通知してほしいものはあるか	記述
通知がない場合と比べて変わったと感じたことは何か	記述
自由記述	記述

### 4.3 実験結果

ファイル更新通知があった場合とない場合を比較した結果を以下に示す。

ツイート文では、ファイル更新通知がない場合はファイル更新通知がある場合と比べて「スライドをアップロードしました」のように自分の作業状況に関するツイートが比較的多く見られたため、ファイル更新を通知することにより負担が軽減されていると考えられる。キャプチャ動画の比較においては、ユーザがブラウザ上からスライド上に写真や文章をコピーしている様子などが確認されたが、ユーザの行動として差は見られなかった。

アンケートでは「通知がある場合に通知がない場合と比べて便利であるように感じたか」について、5 が三人、4 が二人、2 が一人という結果が得られた。また、「通知がない場合と比べて変わったと感じたことは何か」については、「誰がファイルを更新したかが分かりやすい」といった回答が多かった。「ファイルの更新のほかに通知してほしいものはあるか」については、「他ユーザの具体的な作業状況を把握したい」という意見が多く寄せられた。その他の意見として、自由記述では、「何度もファイルが更新されると通知が多くて邪魔である」という意見が得られた。

上記のように、ツイート文の分析やアンケートから良い結果を得られたことから、ユーザの心理的な利便性が上がっていると考えられる。

## 5 おわりに

本研究では、ユーザの会話とファイル更新履歴のタイムライン化による同期的協調支援について述べた。今後の課題として、アンケートの意見から、頻繁にファイル更新が行われた場合について通知回数を減らす方法について検討していく予定である。

## 参考文献

[1] Felix Albertos Marco, Victor M. R. Penichet, Jose A. Gallud : Drag & Share: A Shared Workspace for Distributed Synchronous Collaboration, Distributed User Interfaces, pp.125-132, Springer, 2011