

オンラインストレージに基づく SNS システム

鶴岡 行雄^{1,a)} 若井 英之² 神保 直幸³ 多田 好克^{3,b)}

概要：近年，SNS が普及している．既存の SNS では事業者がユーザのコンテンツを一極所持し，大量の個人情報事業者に集まることからユーザのプライバシーへの懸念がある．このため，分散 SNS が複数提案されているが，管理コストやボランティア運用に伴うサービス持続性が課題である．SNS アプリケーションの Frenzy は，コンテンツの保存場所としてオンラインストレージを利用することで上記課題を解いている．しかし，ソーシャルグラフの管理をオンラインストレージに委ねているためユーザコミュニティの拡張性が無いことが課題であった．これに対して，若井らの SNS システムは，ユーザ間で共有するフォルダの構造について，SNS に特化したセマンティクスを定義することで，SNS システム側からのフレンド関係の拡張を可能にしている．ただし，扱えるのはフレンド関係のみでユーザのグループには対応していなかった．本研究では，フレンド関係とユーザのグループに対応し，ユーザコミュニティの拡張性を持つオンラインストレージベースの SNS システムを提案する．提案システムは，若井らのシステムに対して，複数 ID の使い分けを可能とする ID 体系の拡張を行うことで，グループへの対応を可能にしたものである．提案システムによりプライバシー，低管理コスト，拡張性を持ち，より使いやすい SNS を実現できる．また，オンラインストレージをベースにしたオープンなコミュニケーション基盤の構築が可能となる．

1. はじめに

近年 SNS(Social Networking Service) が普及している．SNS はつながりのあるユーザ間でユーザコンテンツ(以下，コンテンツ)を共有する．コンテンツとはテキスト，写真，動画などである．SNS の多くは SNS 事業者がユーザコンテンツを一極所持しているため，必要以上の個人情報 SNS 事業者に蓄積されるなど，プライバシーの問題が懸念されている．この問題を解決するため，ユーザコンテンツやソーシャルグラフを複数のサーバに分散させる分散 SNS が提案されている [1][2][3]．たとえば，VIS[2] では各ユーザが管理する複数のサーバにユーザコンテンツを保存し，サーバ同士がコンテンツを補完し合っ分散 SNS を構成する．しかし，VIS ではサーバの運用がボランティアベースであるため，サービスの持続性が問題となる．

ところで，ブロードバンドアクセス網の普及により，オンラインストレージサービスの利用が増加している．たとえば，Dropbox[4] では，インターネット上のサーバを紹介して，ユーザ間もしくは複数端末間でファイルを同期・共有できる．このようなオンラインストレージサービスでは，

ユーザは利用するデータ量に応じて事業者を利用料を支払い，事業者は収入を得て事業の継続に務めるため，サービスの持続性が期待できる．

なお，オンラインストレージサービスを利用した SNS として Frenzy[5] が提案されている．Frenzy では Dropbox で同期したファイルを SNS コンテンツとして扱うことで SNS を実現している．しかし，Frenzy ではグループメンバー以外とのコミュニケーションができずユーザコミュニティが閉鎖的になること，またソーシャルグラフの管理を Dropbox に委ねているため，ユーザコミュニティの拡張性が無いことが課題であった．

これに対して，若井らのオンラインストレージベースの SNS システム [6][7] は，ユーザ間で共有するフォルダの構造について，SNS に特化したセマンティクスを定義することで，SNS システム側からのフレンド関係の拡張を可能にしている．ただし，扱えるのはフレンド関係のみでユーザのグループ(以下，グループ)には対応していなかった．

本研究では，フレンド関係とグループに対応し，ユーザコミュニティの拡張性を持つオンラインストレージベースの SNS システムを提案する．提案システムは，若井らのシステムに対して，複数 ID の使い分けを可能とする ID 体系の拡張を行うことで，グループに対応させたものである．またグループの管理をオンラインストレージ側ではなく SNS システム側で行うため，フォルダ構造を持つ SNS

¹ 日本電信電話株式会社

² レッドフォックス株式会社

³ 電気通信大学

^{a)} tsuruoka.yukio@lab.ntt.co.jp

^{b)} tada@is.uec.ac.jp

のセマンティクスも拡張した。提案システムによりプライバシー、持続性、拡張性を備え、より使いやすいオンラインストレージベースの SNS が実現できる。

本稿の構成は以下のとおりである。2章では関連研究について、3章ではオンラインストレージベース SNS について、4章では提案システムの設計について、5章では評価について、それぞれ述べる。6章はまとめとする。

2. 関連研究

オンラインストレージを利用した SNS に Frenzy と若井らのシステムがある。Frenzy は Mac 端末上で動作する SNS アプリケーションであり、オンラインストレージサービスとして Dropbox を利用している。Frenzy では Dropbox でフォルダを共有するユーザの間で SNS コンテンツを共有する。以降、このフォルダを共有フォルダと呼ぶ。共有フォルダの下には“.frenzy”というフォルダが作られ、SNS で共有される SNS コンテンツ (以下、コンテンツ) はこの下に置かれる。コンテンツは共有フォルダの下にあるため、Dropbox により他の端末と同期・共有される。共有フォルダの設定は、Frenzy とは独立に、Dropbox の UI により予め行っておく必要がある。

Frenzy における処理の流れを図 1 の例を用いて説明する。この例では予め Dropbox 上で、ユーザ A,B,C がフォルダ“G1”を、ユーザ C,D がフォルダ“G2”をそれぞれ共有しているものとする。またユーザ A,B,C,D の端末をそれぞれ T_A, T_B, T_C, T_D と呼ぶ。すなわち T_A, T_B, T_C 上の“G1”が同じ内容となるよう常に同期される。同様に T_C, T_D 上の“G2”が同じ内容となるよう常に同期される。

次に各ユーザはそれぞれの端末上で Frenzy アプリを実行し、Frenzy で使用するフォルダを、共有フォルダの中から選択して初期設定する。具体的にはユーザ A,B は“G1”を、ユーザ C は“G1”と“G2”を、ユーザ D は“G2”をそれぞれ選択して設定する。選択された共有フォルダの下には“.frenzy”というフォルダが作成される。

Frenzy はユーザの ID をランダムな文字列として選び、その ID と同名のフォルダを、全ての“.frenzy”フォルダの下に作成する。さらに ID 名を持つフォルダの下にフォルダ“feeds”、ファイル“info.json”、ファイル“avatar.png”を生成する。“info.json”にはユーザ名が格納され、“avatar.png”にはアイコン画像が格納される。図 1 の例では、ユーザ A は Frenzy で使用するフォルダとして“G1”を選択したので、Frenzy は A の ID として文字列“jzuxmt8d42sk”を生成し、“G1”の下にフォルダ“jzuxmt8d42sk”を作成する。他のユーザ端末についても同様に設定される。

次に Frenzy によるコミュニケーションの例を以下に示す。ユーザ C が Frenzy アプリケーション上で日記を投稿したとする。日記は Frenzy により、フォルダ“G1/.frenzy/x79gtey76w3u/feeds/”およ

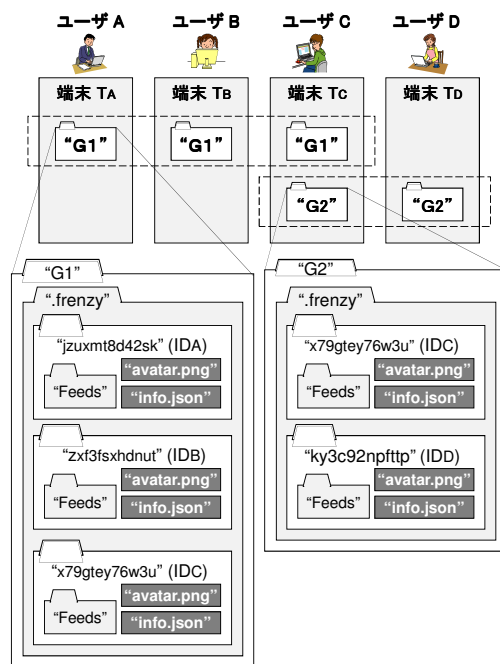


図 1 Frenzy の利用例

Fig. 1 Usage example of Frenzy

び“G2/.frenzy/x79gtey76w3u/feeds/”の下にそれぞれファイルとして格納される。ここで“x79gtey76w3u”は C の ID，“G1/.frenzy/x79gtey76w3u/feeds/”はフォルダ“G1”の下のフォルダ“.frenzy”の下のフォルダ“x79gtey76w3u”の下のフォルダ“feeds”という意味である。ユーザ D の端末 T_D では、“G2”が Frenzy から使うフォルダとなっていることから、Frenzy は“G2/.frenzy/x79gtey76w3u/feeds/”の下の C の日記のファイルを読み込んで GUI で表示する。さらに、ユーザ D が C の日記を読んで Frenzy に返信を書き込むと、返信は“G2/.frenzy/ky3c92npfttp/feeds/”の下にファイルとして格納される。ここで“ky3c92npfttp”は D の ID である。D の返信は、フォルダ“G2”を共有している C からは見ることができる。しかし、フォルダ“G2”を共有していない A と B は D の返信を見ることができない。A, B が D の返信を見るためには、C または D が Dropbox 上でフォルダ“G2”の共有設定に A, B を追加する必要がある。

以上のように、Frenzy では Dropbox で事前設定を行う必要がある。Frenzy アプリケーションからは共有の設定を行うことができない。よって Frenzy ではコミュニケーション相手が固定となり、コミュニティを広げるという SNS としての機能が欠如している。

一方、若井らのシステムは、ユーザ間で共有するフォルダの構造について、SNS に特化したセマンティクスを定義することで、SNS システム側からのフレンド関係の拡張を可能にしている。ただし、扱えるのはフレンド関係のみでユーザのグループ (以下、グループ) には対応していなかった。

本研究では、若井らのシステムに対して、複数 ID の使い分けを可能とする ID 体系の拡張を行うことで、グループに対応させた SNS システムを提案する。提案システムと若井らのシステムの差分は 4 章で説明する。

3. オンラインストレージベース SNS

本研究では、SNS の一実現形態として、オンラインストレージベース SNS を定義する。オンラインストレージベース SNS は以下に述べる要件を満たすものとする。

3.1 オンラインストレージに対する要件

オンラインストレージベース SNS では、全ての SNS ユーザがオンラインストレージを利用できることを前提とする。本研究で考えるオンラインストレージは以下を満たすものとする。まず、オンラインストレージサーバ（以下、サーバと呼ぶ）がインターネット上で動作し、端末間でのファイル同期と共有の機能を提供する。端末上の一つのフォルダ（以下、共有フォルダと呼ぶ）以下の指定したファイルやフォルダを、端末とサーバ間で同期できるものとする。複数の端末が同一のファイル F をサーバと同期した場合、結果として端末間でファイル F が共有される。

オンラインストレージは複数のサービス提供者により提供されてもよいが、ここでは簡単のため全ての SNS ユーザが同一のオンラインストレージサービスを利用するものとする。

同期・共有は各端末においてどのような手段で行ってもよいが、ここでは端末上で同期アプリケーションが動作し、インターネットを介して随時サーバと通信するものとする。

たとえばユーザ A の端末 T_A 上でファイル X を X' に更新したとする。端末上の同期アプリケーションはこの更新に関する情報をサーバに送信し、サーバはサーバ上のファイル X を X' に更新する。ユーザ B の端末 T_B 上の同期アプリケーションは定期的にサーバにアクセスし、 X の X' への更新に関する情報を取得する。そして、端末 T_B 上で X を X' に更新する。この結果、端末 T_A と端末 T_B 間でファイル X' が同期される。

また、上記の同期・共有はローカルファイルへのアクセスに対してトランスペアレントであるものとする。すなわち、端末上のアプリケーションは他端末と共有するファイルを、通常のローカルファイルと同様に扱うことができる。また、サーバはアクセスするユーザを認証し、その結果に基づきフォルダの共有やアクセス制御を行えるものとする。さらに、アクセス制御の設定を外部から変更するインターフェースを持つものとする。

上記に加え、オンラインストレージサービスは、SNS に必要な拡張機能を持つものとする。拡張機能の詳細については 4 章で述べる。

3.2 SNS としての要件

以下に本研究で考える SNS の要件を示す。SNS によってはより多くの機能を提供することが考えられるが、ここではオンラインストレージベース SNS を定義し、特徴づけるために必要な機能に限定して SNS を定義する。

SNS ではユーザ間のつながりを管理する。ユーザ間のつながりは、一般にフレンド関係とグループがある。フレンド関係は、2 者間の対称的、非推移的、非反射的な 2 項関係であり、2 者がそれぞれの情報を共有することを互いに合意した場合にのみ成り立つものとする。

グループはユーザの集合である。

本研究で考える SNS は、フレンド関係とグループを扱えるものとする。なお Frenzy はグループのみ扱え、[6] はフレンド関係のみ扱える。

3.2.1 SNS の機能

本研究で考える SNS は、ユーザ間でコンテンツを共有する枠組みとして、以下の機能を持つものとする。

- コンテンツの新規投稿
UI を通じてコンテンツをシステムに入力し、特定のユーザから閲覧可能とする。たとえばフレンドやグループに対して日記を投稿するなどである。
- コンテンツの閲覧
自分や他のユーザが投稿したコンテンツを画面に表示する。
- コンテンツに対するコメントの投稿
自分や他のユーザが投稿したコンテンツに対するコメントを、元のコンテンツに関連する新たなコンテンツとして投稿する。
- メッセージの送信
特定のユーザに向けてコンテンツを送信する
- メッセージの受信
自分に向けて送信されたメッセージを受け取る
- フレンド関係の追加と削除
ユーザが他のユーザにフレンド関係となることをリクエストする。リクエストを受け取ったユーザがリクエストを承認した場合にのみフレンド関係が追加される。またユーザがフレンド関係を解消することをリクエストする。フレンド関係の解消は相手の承認を必要としない。
- グループへの参加と脱退
ユーザがグループへの参加や脱退をリクエストする。参加はリクエストを受け取ったグループオーナーがリクエストを承認した場合にのみ許可される。脱退はグループオーナーの承認を必要としない。
- 通知
自分宛てにリクエストがあった時に通知される。

3.2.2 拡張性に関する要件

SNS の利点の一つに、システムの利用を通して新たな

ユーザとのつながりができる点がある．このため，SNS システムは未知のユーザに接する機会を設け，フレンド関係を結んだりグループに参加するきっかけを提供する必要がある．本研究で考える SNS システムは以下の機能を持つものとする．

- フレンドのフレンドのコンテンツの閲覧
フレンドだけでなく，フレンドのフレンドのコンテンツを閲覧可能にする．具体的には，ユーザ A, C が共にユーザ B のフレンドであったとき，B の投稿に対する A のコメントを C が閲覧できるようにすることで，C は (フレンド B のフレンドである) A を知ることができ，A とフレンド関係となる機会を得る．
- グループ
グループへの投稿は，グループの全てのメンバーに共有される．グループのメンバーは投稿者を知り，投稿者とフレンド関係となる機会を得る．
- ソーシャルグラフを管理する機能
システムからグループメンバーやフレンド関係の追加や削除ができること．Frenzy のように，外部のオンラインストレージサービスでソーシャルグラフを管理するのではなく，管理に必要な情報と機能を全て備えた自己完結型のシステムとする．

4. 提案システムの設計

本研究では，3 章で定義したオンラインストレージベース SNS に基づく SNS システムとして OS-SNSv2(Online Storage based Social Networking System Version 2) を提案する．OS-SNSv2 の設計を以下に示す．

4.1 ユーザ ID とグループ ID

SNS のユーザはオンラインストレージサービスのユーザでもある．オンラインストレージサービスは，そのユーザを識別するため，ユーザに Online Storage Service User ID (以下，OID と呼ぶ) を付与する．OID はオンラインストレージサービスにローカルな ID である．

また，オンラインストレージサービスプロバイダを識別するためのグローバルユニークな ID として OSSP-ID を定義する．OSSP-ID は，たとえばオンラインストレージサービスプロバイダのドメイン名"x.com"などである．

また，SNS のユーザは互いを識別するためのグローバルユニークな SNS User ID (以下，SID と呼ぶ) を持つものとする．SID は他のサービスの ID，たとえばメールアドレスを流用することも出来るが，ここでは簡単のためオンラインストレージサービスプロバイダがそのユーザに SNS のユーザ ID である SID を付与するものとする．

一般に，サービスがそのユーザを識別する際，通常は名寄せされない．たとえば一人の人が一つのプロバイダから複数のメールアドレスを取得して使い分けことが運用上可

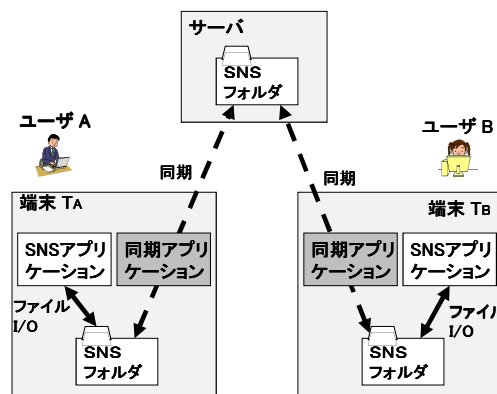


図 2 OS-SNS のシステム構成

Fig. 2 Structure of OS-SNS

能である．ID の使い分けはペルソナと呼ばれ，ユーザに利便性をもたらす．SNS においても複数の人格を使い分けたいという要求がある．本研究で考える SNS においても SID の使い分けを可能とする．すなわち OID で識別されるオンラインストレージサービスの 1 ユーザが，複数の SID を取得して，SNS 上の別人としてそれらを使い分けができるものとする．オンラインストレージサービスは SID から OID への一意の対応関係を保持する．各 ID の例として，たとえば OSSP-ID が"x.com"，OID が"A@x.com"，対応する SID が"A1@u.x.com"と"A2@u.x.com"などになる．

グループを識別するためにグローバルユニークな SNS Group ID (以下，GID と呼ぶ) を定義する．GID も SID と同様にオンラインストレージサービスプロバイダが付与するものとする．OID で識別されるオンラインストレージサービスのユーザが，GID を取得してグループを作成できる．GID を取得したユーザはグループのオーナーとなる．ユーザは複数の GID を取得できる．オンラインストレージサービスは GID から OID への一意の対応関係を保持する．

4.2 構成

提案システムの構成を図 2 に示す．提案システムは複数の端末と，オンラインストレージサービスを提供するサーバで構成される．サーバについては 3 章で述べた通りとする．端末上では同期アプリケーションと SNS アプリケーションが動作する．同期アプリケーションは 3 章で述べた通りサーバに対して共有フォルダの同期・共有を行う．共有フォルダ内には SNS で共有するファイルを集めた SNS フォルダを置く．SNS アプリケーションは SNS フォルダ以下のファイルを SNS コンテンツとしてユーザに提示し，またユーザから入力された SNS コンテンツをファイルとして SNS フォルダに格納する．

ユーザは複数台の端末を利用してもよく，それらの SNS フォルダは同期される．以下では簡単のため，各ユーザの

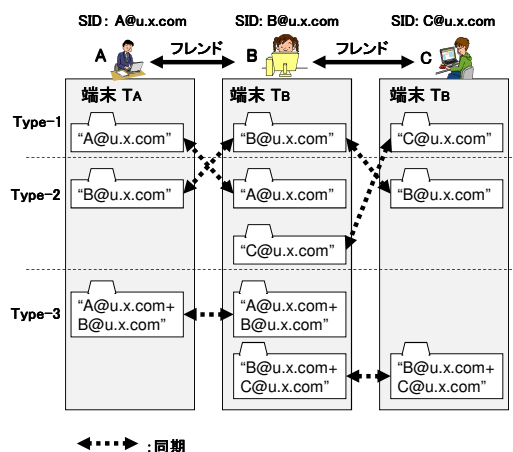


図3 SNS フォルダの構造 (フレンド関係)

Fig. 3 Structure of SNS folder (friend relation)

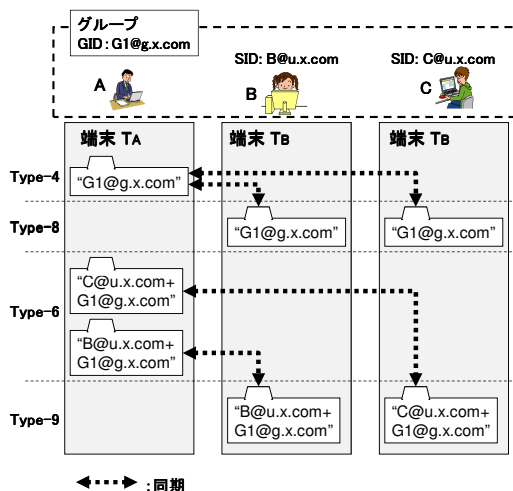


図4 SNS フォルダの構造 (グループ)

Fig. 4 Structure of SNS folder (user group)

端末が1台であると仮定する。

4.3 SNS フォルダの構造

提案システムにおいては、SNS フォルダの構造に SNS に固有の意味を持たせている。SNS フォルダの構造とその意味について以下に述べる。各ユーザの SNS フォルダは、その下に以下の9種類のフォルダを持つ。

Type-0: フォルダ”control”

Type-1: ユーザ自身の SID と同名のフォルダ

Type-2: フレンドの SID と同名のフォルダ

Type-3: ユーザ自身の SID とフレンドの SID を組み合わせた名前のフォルダ。組み合わせは2つのIDを辞書順にならべ”+”で繋いだもの。

Type-4: ユーザ自身が作成したグループ (GID) と同名のフォルダ

Type-5: ユーザ自身の GID とユーザ自身の SID を組み合わせた名前のフォルダ

Type-6: ユーザ自身の GID とグループメンバーの SID を組み合わせた名前のフォルダ

Type-8: 他のユーザが作成したグループ (GID) と同名のフォルダ

Type-9: 他のユーザの GID とユーザ自身の SID を組み合わせた名前のフォルダ

Type-3 は Type-1 と Type-2 の組み合わせ、Type-5 は Type-1 と Type-4 の組み合わせとなっている。このような関係があるため Type-7 は欠番とした。なお、Type-0 から Type-3 までは若井らのシステム [6] で定義したものであり、Type-4 から Type-9 までは本研究で拡張した部分である。

二つの端末が SNS フォルダの下に同名のフォルダを持つ場合、オンラインストレージサービスは、それらを端末間で共有するよう動作するものとする。

フレンド関係の例を図3に示す。実線矢印は SNS ユーザ間のフレンド関係を表し、破線矢印はフォルダの共有を表す。例ではユーザ A (以下, A) とユーザ B (以下, B)、B とユーザ C (以下, C) がそれぞれフレンドである。A, B, C の SID はそれぞれ”A@u.x.com”, ”B@u.x.com”, ”C@u.x.com”とする。A は、Type-1 フォルダ”A@u.x.com”, Type-2 フォルダ”B@u.x.com”, Type-3 フォルダ”A@u.x.com+B@u.x.com”をそれぞれ持つ。B, C も同様である。なお同名のフォルダは自動的に共有されるため、A から見たとき、フォルダ”A@u.x.com”はフレンドである B と共有され、フォルダ”B@u.x.com”はフレンドである B およびフレンドのフレンドである C と共有され、フォルダ”A@u.x.com+B@u.x.com”は B と共有される。

すなわち、Type-1 フォルダはフレンド全員と共有され、Type-2 フォルダは一人のフレンドと、そのフレンドのフレンド全員と共有され、Type-3 フォルダは一人のフレンドのみと共有される。

したがって、Type-1 フォルダは、フレンド全員に向けて情報を発信するとき、もしくはフレンドのいずれかから情報を受け取る時に用いることができる。Type-2 フォルダは、フレンドの発信した情報を閲覧するときや返信するとき、さらにフレンドのフレンドとのコミュニケーションに用いることができる。Type-3 フォルダは特定の相手との一対一のコミュニケーションに用いることができる。Type-0 フォルダはサーバのみと同期・共有され、サーバとの制御情報のやり取りに用いられる。具体的な利用例は 4.6, 4.7, 4.8 節で述べる。

グループを実現するフォルダの例を図4に示す。グループもフレンド関係とほぼ同様の仕組みで実現できる。

4.4 フレンドとのコンテンツ共有の手順

フレンドへのコンテンツの投稿から閲覧までの処理の流れを示す例を、図3を用いて説明する。

例 1 : B がコンテンツを投稿し、それをフレンドの A が閲覧し、 B に返信するものとする。 B は端末 T_B の SNS アプリケーションにフレンド宛ての日記を入力する。 SNS アプリケーションはその日記をフォルダ " $B@u.x.com$ " に格納する。 端末 T_B, T_A の同期アプリケーションとサーバの動作により日記は端末 T_A のフォルダ " $B@u.x.com$ " に同期される。 A が端末 T_A の SNS アプリケーションを起動すると、フォルダ " $B@u.x.com$ " に格納された B の日記が画面に表示される。 A が日記に対するコメントを返信として SNS アプリケーションに入力すると、 SNS アプリケーションは返信を、日記と同じフォルダ " $B@u.x.com$ " に格納する。 フォルダ " $B@u.x.com$ " は A, B, C 間で共有されているため、 B の日記と A の返信は、 A, B, C から等しく閲覧できる。 これにより C は B の日記に返信した A の存在を知ることができる。

例 2 : B が A のみに宛ててメッセージを送信し、 A が B のみに返信するものとする。 B は端末 T_B の SNS アプリケーションに A 宛てのメッセージを入力する。 このとき、宛先として A の SID である " $A@u.x.com$ " を指定する。 SNS アプリケーションは宛先が指定されたそのメッセージをフォルダ " $A@u.x.com+B@u.x.com$ " に格納する。 以降は上記の例と同様に、メッセージは端末 T_A のフォルダ " $A@u.x.com+B@u.x.com$ " に同期される。 A が端末 T_A の SNS アプリケーションを起動すると、フォルダ " $A@u.x.com+B@u.x.com$ " に格納された B のメッセージが画面に表示される。 A がメッセージに対する返信を SNS アプリケーションに入力すると、 SNS アプリケーションは返信を、元のメッセージと同じフォルダ " $A@u.x.com+B@u.x.com$ " に格納する。 フォルダ " $A@u.x.com+B@u.x.com$ " は A と B のみで共有されているため、 B のメッセージと A の返信は、 A, B のみから閲覧できる。

4.5 グループ内でのコンテンツ共有の手順

グループへの投稿は、フレンドへの投稿とほぼ同じである。 ユーザ C がグループ " $G1@g.x.com$ " へ投稿しユーザ B が閲覧する例を以下に示す。 なお B と C とは予めグループ " $G1@g.x.com$ " に参加しているものとする。 C は端末 T_C の SNS アプリケーションでグループ " $G1@g.x.com$ " 宛てにメッセージを投稿する。 SNS アプリケーションはそのメッセージをフォルダ " $G1@g.x.com$ " に格納する。 端末 T_B, T_C の同期アプリケーションとサーバの動作によりメッセージは端末 T_B のフォルダ " $G1@g.x.com$ " に同期される。 B が端末 T_B の SNS アプリケーションを起動すると、フォルダ " $G1@g.x.com$ " に格納された C のメッセージが画面に表示される。 B がメッセージに対するコメントを返信として SNS アプリケーションに入力すると、 SNS アプリケーションは返信を、元のメッセージと同じフォルダ " $G1@g.x.com$ "

に格納する。 フォルダ " $G1@g.x.com$ " は B, C を含めグループのメンバー間で共有されているため、 C の投稿と B の返信は、グループメンバーから等しく閲覧できる。 これによりグループのメンバー、たとえばユーザ D はユーザ B や C の存在を知ることができる。 なお、フレンドとグループの違いは以下のとおりである。 フレンドとのやりとりの場合は、相手の SID 名のフォルダに直接投稿はできず、相手の投稿にコメントする場合にのみ相手の SID 名のフォルダに投稿される。 最初の投稿（スレッドの生成）は自分の SID 名のフォルダに対してなされる。 一方、グループへの投稿の場合は、参加するグループであれば、他のユーザが作成したグループであっても最初の投稿ができる。

フォルダ " $C@u.x.com+G1@g.x.com$ " は、グループメンバー C とグループオーナーとの直接のやりとり、たとえばグループへの参加や脱退などの連絡に用いる。

4.6 フレンド関係の追加の手順

提案システムにおける、フレンド関係の追加の処理の流れの例を図 3、図 5 を用いて説明する。

例 3 : 上記の例 1 の後に C は A に対してフレンド関係となるようリクエストするものとする。 C は SNS アプリケーションの GUI を通して A にフレンド関係となるようリクエスト（以下、フレンドリクエスト）する。 SNS アプリケーションは C のリクエストを受け、端末 T_C の SNS フォルダの下にフォルダ " $A@u.x.com+C@u.x.com$ " を作成する。 なお " $A@u.x.com$ " はユーザ A の SID である。同期アプリケーションはフォルダ " $A@u.x.com+C@u.x.com$ " の生成を検知し、これをサーバに同期する。サーバはフォルダ " $A@u.x.com+C@u.x.com$ " の生成をフレンドリクエストと解釈し、SID として " $A@u.x.com$ " を持つオンラインストレージサービスのユーザ A ($OID="A@x.com"$) を特定する。そして、 A のフォルダ " $control$ " 下のファイル " $req.txt$ " に、 C からフレンドリクエストがあったことを追記する。この更新は A の端末 T_A の " $req.txt$ " に同期される。端末 T_A の SNS アプリケーションは " $req.txt$ " の更新を検出し、GUI を通して、 A に、" $C@u.x.com$ " からフレンドリクエストが来た事を通知する。 A がフレンドリクエストの承諾を入力すると SNS アプリケーションは、承諾を表すステータスコードを " $req.txt$ " に書きこむ。 端末 T_A の同期アプリケーションはこの更新をサーバに伝える。サーバは A の " $req.txt$ " の内容から A の承諾を読み取り、 A と C のフレンド関係に必要な設定を行う。具体的には端末 T_A の SNS フォルダ下にフォルダ " $C@u.x.com$ " と " $A@u.x.com+C@u.x.com$ " を、端末 T_C の SNS フォルダ下にフォルダ " $A@u.x.com$ " をそれぞれ作り、 A, C 間で同名のフォルダ " $A@u.x.com$ "、" $C@u.x.com$ "、" $A@u.x.com+C@u.x.com$ " の共有設定を行う。

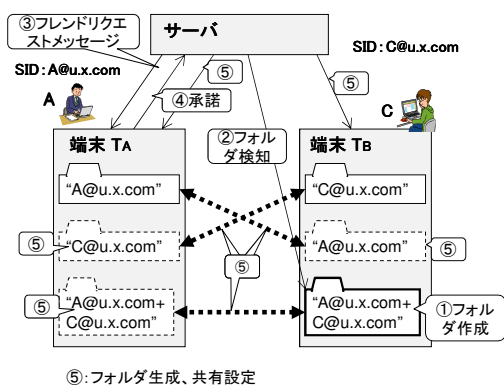


図5 フレンド関係追加の手順
Fig. 5 Sequence for friend-request

なお、SNS アプリケーションはフォルダを作成することでフレンドリクエストができる。また、オンラインストレージサービスがフォルダ構造から SNS のセマンティクスを理解し、上記の動作を行うことで SNS アプリケーションからのフレンド関係の拡張が可能になる。A がフレンドリクエストを拒否した場合は、サーバは C の端末 T_C からフォルダ $A@u.x.com+C@u.x.com$ を削除する。

なお、若井らのシステムでは、フレンドリクエストを行う際に $A@u.x.com+C@u.x.com$ ではなく、 $A@u.x.com$ を作成する。提案システムにおいては、C の SID は複数あり得るので、サーバが C の SID を特定できるようにするため上記の仕様に変更した。

4.7 グループへの参加の手順

グループへの参加の手順は、フレンド関係の追加とほぼ同じである。以下、ユーザ C がグループ $G1@g.x.com$ (GID) への参加をリクエストする例で説明する。

C は SNS アプリケーションの GUI を通して $G1@g.x.com$ への参加をリクエスト (以下、グループリクエスト) する。SNS アプリケーションは C のリクエストを受け、端末 T_C にフォルダ $C@u.x.com+G1@g.x.com$ を作成する。同期アプリケーションはフォルダ $C@u.x.com+G1@g.x.com$ の生成を検知し、これをサーバに同期する。サーバはフォルダ $C@u.x.com+G1@g.x.com$ の生成をグループリクエストと解釈し、GID が $G1@g.x.com$ であるグループを持つオンラインストレージサービスのユーザ A (OID= $A@x.com$) を特定する。そして、A のフォルダ $control$ 下のファイル $req.txt$ に、C からグループリクエストがあったことを追記する。この更新は A の端末 T_A の $req.txt$ に同期される。端末 T_A の SNS アプリケーションは $req.txt$ の更新を検出し、GUI を通して、A に、 $C@u.x.com$ からグループ $G1@g.x.com$ へのグループリクエストが来た事を

通知する。A が C の参加を認め、グループリクエストの承諾を入力すると SNS アプリケーションは、承諾を表すステータスコードを $req.txt$ に書きこむ。端末 T_A の同期アプリケーションはこの更新をサーバに伝える。サーバは A の $req.txt$ の内容から A の承諾を読み取り、C のグループへの参加に必要な設定を行う。具体的には端末 T_A にフォルダ $C@u.x.com+G1@g.x.com$ を、端末 T_C にフォルダ $G1@g.x.com$ をそれぞれ作り、A、C 間で同名のフォルダ $G1@g.x.com$ 、 $C@u.x.com+G1@g.x.com$ の共有設定を行う。なお、グループリクエストの場合はフレンドリクエストと異なり、端末 T_A に $C@u.x.com$ は作成されず、共有もされない。A がグループリクエストを拒否した場合、サーバは C の端末 T_C からフォルダ $C@u.x.com+G1@g.x.com$ を削除する。

4.8 SID と GID 取得の手順

提案システムでは、SNS を利用する準備として SID の取得が必要である。オンラインストレージサービスのユーザ A (OID= $A@x.com$) が、SID として $A@u.x.com$ を取得する手順を以下に示す。

A は SNS アプリケーション上で SNS ID (SID) の追加を実行し、GUI を通して SID のプレフィックス、たとえば A を入力する。SNS アプリケーションは A のフォルダ $control$ 下のファイル $req.txt$ に、SID として $A@u.x.com$ を追加するリクエストがあったことを追記する。この更新はサーバ上の $req.txt$ に同期され、サーバは $A@u.x.com$ の一意性を確認する。過去に $A@u.x.com$ の取得が無い場合、サーバはユーザ A (OID= $A@x.com$) へ SID $A@u.x.com$ の付与を許可し、取得の成功を表すステータスコードを $req.txt$ に追記する。 $A@u.x.com$ が過去に取得されていた場合は、取得の失敗を表すステータスコードが $req.txt$ に追記される。取得が成功した場合、サーバは端末 T_A の SNS フォルダ下にフォルダ $A@u.x.com$ を作る。

グループを利用する場合は、グループの代表となるユーザ (グループのオーナー) が事前に GID を取得する。その手順は SID の場合と同様である。なおグループオーナー自身もグループのメンバーとして自身の SID をグループに登録 (参加) する必要がある。その手順は前節で述べたとおりである。グループオーナーはオーナーとしての ID、たとえば $G1@g.x.com$ 、とグループメンバーとしての ID、たとえば $A@u.x.com$ を使い分ける。

4.9 グループとフレンド関係の相互作用

SNS の利用によりユーザコミュニティを広げていくためには、(1) SNS のアクティビティから未知のユーザの SID や未知のグループの GID を知ること、(2) それらの ID からユーザやグループに関する情報を取得できること、(3) SID や GID を用いてフレンドリクエストやグループリクエ

表 1 関連技術との比較

Table 1 Comparison among other systems

	プライバシー	サービスの持続性	つながりの拡張性	フレンド関係	グループ
Facebook, mixi	× コンテンツを一極所持	広告収入等	フレンド申請, ユーザ検索等		
VIS[2] diaspora[3]	コンテンツ所有権の分散	× ボランティアベースの運用	フレンド申請, ユーザ検索等		
Frenzy[5]	コンテンツ所有権の分散	ストレージ使用料	× SNS 外で設定が必要	×	
若井らのシステム (OS-SNSv1)	コンテンツ所有権の分散	ストレージ使用料	フレンドリクエスト		×
提案システム (OS-SNSv2)	コンテンツ所有権の分散	ストレージ使用料	フレンド, グループリクエスト		

トが行えることが必要である。(1) について, まずフレンドについては予め SID が分かっている。また, 同じグループのメンバーやフレンドのフレンドについては投稿を閲覧することで SID を知る機会がある。さらにフレンドとグループ, もしくは複数のグループにクロスポストし, 受け取ったコメントも含めたスレッドを複数のコミュニティで共有することで, 異なるコミュニティに属するユーザ同士が互いの SID を知る機会が得られる。クロスポストはコンテンツを複数のフォルダに入れることで可能となる。この際, コピーせずにハードリンクでファイルを共有すれば複数の投稿先でスレッドを共有できる。(2) についてはゲストユーザへのプロフィールの公開で実現できる。SID からユーザのプロフィールを閲覧することで, そのユーザに関する情報の他, 所属するグループの GID やフレンドの SID 等が得られる。グループのプロフィールからはそのグループに関する情報の他, グループメンバーの SID や関連するグループの GID が得られる。(3) について, 提案システムではファイル操作に基づく API を用いてアプリケーションから容易に実現できる。まとめると, クロスポストとプロフィールの公開により, グループとフレンド関係の相互作用でユーザコミュニティの拡大が図れる。

5. 評価

提案方式を従来方式との比較を表 1 に示す。Facebook, mixi 等の既存の SNS サービスはユーザコンテンツを一極所持するためプライバシーの懸念がある。分散 SNS について, VIS 等では個人サーバの運用コストが課題となり, diaspora 等のボランティアベースの運用ではサービスの持続性に問題がある。Frenzy は上記の問題を解決しているが, ユーザコミュニティの拡張性が無いことが課題である。Frenzy の課題を解決した若井らのシステムではプライバシー, 持続性, 拡張性を満たす SNS システムとなっているが, グループに対応していなかった。提案システムは, 若井らのシステムに対して, 複数 ID の使い分けを可能とする ID 体系の拡張を行った。このアプローチにより, SNS

システム側からソーシャルグラフの管理を可能とする利点を生かしたまま, グループのサポートが可能となった。結果としてフレンド関係とユーザのグループに対応し, ユーザコミュニティの拡張性を持つオンラインストレージベースの SNS システムが実現できる。

6. まとめ

本研究では, オンラインストレージを用いたソーシャルネットワークシステムを提案した。提案システムは, 若井らのシステムに対して, 複数 ID の使い分けを可能とする ID 体系の拡張を行うことで, グループへの対応を可能にしたものである。これにより, フレンド関係とユーザのグループに対応し, ユーザコミュニティの拡張性を持つオンラインストレージベースの SNS システムが実現できる。提案システムによりプライバシー, 低管理コスト, 拡張性を持ち, より使いやすい SNS を実現できる。また, オンラインストレージをベースにしたオープンなコミュニケーション基盤の構築が可能となる。

参考文献

- [1] Sonja.B et al.: A Casefor P2P Infrastructure for Social Networks - Opportunities & Challenges, 6th Wireless On-Demand Network Systems and Services (WONS), pp. 161-168, 2009.
- [2] Ramn.C et al.: Virtual Individual Servers as Privacy-Preserving Proxies for Mobile Devices, ACM SIGCOMM Conference (SIGCOMM), pp. 37-42, 2009.
- [3] diaspora*: <https://joindiaspora.com/>, 2013 年閲覧.
- [4] Dropbox: <http://www.dropbox.com/>, 2014 年 3 月閲覧.
- [5] Frenzy 1.6: <http://aptonic.github.io/frenzy/>, 2014 年 3 月閲覧.
- [6] 若井英之, 鶴岡行雄, 多田好克: オンラインストレージを用いた分散 SNS の設計と実装, 情報処理学会 第 76 回全国大会予稿 2014 年 3 月.
- [7] 若井英之: オンラインストレージを用いた分散 SNS の設計と実装, 修士論文, 電気通信大学 大学院情報システム学研究科 (2014).