

修士論文 2005年度 (平成17年度)

複数の人格を使い分けるソーシャルネットワークモデルの提案

慶應義塾大学大学院 政策・メディア研究科

複数の人格を使い分けるソーシャルネットワークモデルの提案

オンライン上のコミュニケーションにおいて、その主体を抽象化することはその端点を表すという意味から非常に重要な問題である。人間社会における名前や肩書きに置き換わるメールアドレス等の識別子による抽象化が行われ、この識別子を端点とするコミュニケーションによってインターネット上の様々なアプリケーションは普及を遂げた。

さらに次の段階として、人を直接指す識別子だけでなく、人間同士の知人関係を抽象化した知人ネットワークモデルを、インターネット上のコミュニケーション基盤として用いるようになった。しかし、ある人を表すメールアドレスが複数存在し得るように、知人関係にも様々な形態があり、人間は相手や状況に応じた多層的な関係を築いている。

本研究では、複数の人格を相手や状況に応じて使い分けることができるモデルを提案する。それらは完全に乖離した別の人格ではなく、ある人間を複数の側面から見たものに過ぎず、関係性を持つ複数の人格を実現しなければならない。知人関係に基づいた自己情報コントロールを可能とする知人ネットワークを、より人間の本来のコミュニケーションモデルに見合った形で再定義する。

このモデルの有意性を示すため、上記モデルに基づくソーシャルネットワークキングサイト (SNS) を構築し、実環境にて実験運用を行った。この運用結果から、モデルが本研究の目的を達成していることを示した。

キーワード

1. ソーシャルネットワーク, 2. アクセス制御, 3. 自己情報コントロール,
4. 人格の使い分け, 5. 信頼

Social Network Model based on Multiple Personality

Abstracting the party of communication over network is very important problem. A lot of applications on the Internet is deployed in terms of the communication that points to the identity such as name in real world, and e-mail address in cyber space.

In the next phase, we use social networking model which abstracts acquaintanceship as the infrastructure of online communication. However, we have a variety of friendship that is influenced by context, or other part.

This research suggests the model that we can use multiple personality. There are not different completely because there are derived from one person. The social networking model is re-defined to realize the control of myself.

The Social Networking Site (SNS) is implemented and operated for confirming the significance of this model. This model is significant.

Keywords :

1. Social Networking, 2. Access Control, 3. Self Information Control,
4. Multiple Personality, 5. Trust

Keio University Graduate School of Media and Governance

NAKAYAMA, Masahiro

目次

第1章	序論	1
1.1	背景	1
1.1.1	多種多様なコミュニケーション	1
1.1.2	知人ネットワーク	2
1.1.3	自分の使い分け	4
1.2	本研究の目的と意義	4
1.3	用語等の定義	4
1.4	本論文の構成	5
第2章	先行事例	6
2.1	コミュニケーションモデル	6
2.2	相手の指定	6
2.3	既存の SNS についての考察	8
2.3.1	mixi	8
2.3.2	livedoor フレンドパーク	10
2.3.3	wyco	12
2.3.4	imeem	12
2.4	複数の人格の使い分け	12
2.4.1	使い分けを行わない	12
2.4.2	公開範囲のグルーピング	12
2.4.3	複数の ID の併用	13
第3章	設計	15
3.1	要求分析	15
3.1.1	単純な事例	15
3.1.2	弱い使い分け	16
3.1.3	強い使い分け	16
3.1.4	強い使い分け+	16
3.2	要求分析	17
3.2.1	内容	17
3.3	複数人格モデルの概要	17
3.4	複数人格モデルの詳細	18

3.4.1	人格と文脈	18
3.4.2	一方向的な同一人物関係の開示	18
3.5	既存のモデルとの比較	19
3.6	リレーションによるモデルの表現	19
3.6.1	複数的人格の管理	21
3.6.2	知人リンクの構築	21
3.6.3	同一人物リンクの構築	22
3.6.4	ソーシャルネットワークによるアクセス制御	24
第4章	プレ実験	25
4.1	プレ実装の目的	25
4.2	プレ実験の概要	25
4.2.1	実験結果	26
4.2.2	アンケート調査	27
第5章	アプリケーションの設計と実装	30
5.1	実装の方針	30
5.2	実装環境	30
5.3	全体の概要	31
5.4	サイト構成	34
5.4.1	利用者自身のトップページ	36
5.4.2	人格ごとのページ	38
5.4.3	リンク申請	39
5.4.4	日記の登録	39
5.4.5	プロフィールの編集	41
5.5	各モジュールの解説	42
5.5.1	利用者の管理	42
5.5.2	人格の管理	42
5.5.3	知人ネットワークの管理	42
第6章	評価	45
6.1	事例との比較	45
6.1.1	単純な事例	45
6.1.2	弱い使い分け	45
6.1.3	強い使い分け	45
6.1.4	強い使い分け+	46
6.1.5	まとめ	46
6.2	モデルの考察	46
6.2.1	複数的人格の分離	46
6.2.2	一方向的な同一人物関係の開示	46

6.2.3	既存の SNS モデルとの親和性	46
6.3	他方式との比較	47
6.3.1	他の SNS	47
6.3.2	UNIX group	47
6.3.3	アクセス制御リスト	47
6.4	運用実験	48
6.4.1	実験概要	48
6.4.2	実験結果	49
6.4.3	アンケート調査	49
第 7 章	結論	50
7.1	本研究の成果	50
7.2	今後の課題	50
7.2.1	分散化	50
7.2.2	細かい制御	50
7.3	発展	51
	謝辞	52
	参考文献	53
	付 録 A アンケート集計結果	54

目次

1.1	知人ネットワークに基づく情報の配布	3
2.2	知人ネットワークによる範囲の限定	8
2.3	mixi ホームページ	9
2.4	livedoor フレンドパーク 友達の一覧	11
2.1	宛先による範囲の限定	14
3.1	ER 図によるソーシャルネットワークの定義	20
3.2	複数の人格の管理	21
3.3	知人リンクの構築	23
3.4	同一人物リンクの構築とその影響	23
3.5	ソーシャルネットワーク上の距離に基づく公開範囲の指定	24
5.1	システム全体のモジュール構成	32
5.2	各テーブルの関係	33
5.3	Perso ウェブサイト構成	35
5.4	利用者のホームページ	37
5.5	人格ごとのページ	38
5.6	知人リンクの要求	39
5.7	知人リンクの承認・拒否画面	40
5.8	日記の登録画面	40
5.9	プロフィールの編集	41

表 目 次

4.1	運用実験の利用状況	26
4.2	所有する人格の数による度数分布	26
4.3	同一人物関係の設定状況	29
5.1	実装環境	30
5.2	RDBMS 上のテーブルの抽象化	44

第1章 序論

本章ではまず、人間のコミュニケーションのあり方が多岐にわたることを明らかにし、実世界およびインターネット上においてそれをどのように実現してきたかを述べる。その中から、コミュニケーションの当事者である人間同士の知人関係について考察する。この考察を元に、多様な知人関係を表現できる新しい抽象化を提案する意義を述べる。

1.1 背景

1.1.1 多種多様なコミュニケーション

人類文明において、人々が互いを別個の存在として認識し、その意思を他者に伝えるコミュニケーション手段の発達によりその発展が促されてきた。特に、一定の規則に基づく確実な意思伝達手段である文字の役割は大きく、紀元前5000年頃には伝書鳩のような遠隔地とのコミュニケーションも実現した。

コミュニケーションは三つの要素によって分類することができる。そのうちもっとも重要なものは言うまでもなく伝えるべき「内容」である。その内容は、たとえば何らかの事実であったり、思考であったり、あるいは感情であったりする。これらを共有された言語によって表すことで、できる限り曖昧さを避けて伝えることができる。

次に重要なものが、「誰に」伝えるのか、つまり他者を識別することである。他者を識別するために生まれたのが、それぞれの個体に付けられた名前である。名前の歴史は有史以前にまで遡り、名前の付け方は様々な文化的な事情と結びついてきた。特に、血縁関係により姓や氏族名と呼ばれる名前の一部を引き継ぐことにより、家柄を名乗る習慣が多く、現在の日本でもこの姓名制度を採用している。最近では、1989年に国連総会で児童の権利に関する条約^[1]が採択され、その第7条第1項において、「児童は、出生の時から氏名を有する権利及び国籍を取得する権利を有する」ことが定められた。このように、全ての人類はお互いを名前によって識別し、名前に基づいた社会を構築してきた。

名前には、その識別子としての必要性から、用途によって様々な種類がある。法律で認められる公式な本名だけでなく、例えば本名の簡略形や、その人の特徴を表すようなあだ名や、相手の立場・役職名を名前のように扱うことが多い。また、

洗礼によって与えられるミドルネームや、歌舞伎役者のように歴代の先人の役者名を襲名するなど、名前はきわめて多様性のあるものとなっている。

人類が複雑なコミュニケーションを実現する上で、その話し手、聞き手を識別できる名前の存在は必要不可欠なものである。言い換えると、自分以外の全ての人に対してではなく、誰か特定の人に相手を限定することにより、複雑な意思伝達が可能となった。

このように、名前という識別子を通して人間や、あるいは人間でないものを抽象化することで、コミュニケーションの相手を正しく識別することができるようになった。

コミュニケーションにおける最後の要素が、その手段である。最初に紹介した伝書鳩や、煙による狼煙などが初期の通信手段であったが、これらには大きな欠点を抱えている。伝書鳩には、鳩が相手にたどり着けるかの保証が無く、気候などに大きく左右されるほか途中で捕食される危険すらある。狼煙信号は、立ち上った煙が見える範囲に相手が居なければならぬだけでなく、誰からもその通信の存在が見られてしまうという点も場合によっては問題となるだろう。また、笛などの音響的な手段も存在したが、これも狼煙と同様の問題がある。このような問題点から、人間の記憶や書面により、直接人が移動することによる通信手段が、電気通信技術が登場する 19 世紀まで利用され続けた。

1832 年に電信機が発明されたことで状況は大きく変化する。遠隔地とのコミュニケーションに要する時間が激減したことで、世界中の情報を時差無くやりとりできるようになった。1837 年に考案されたモールス信号ではアルファベットや簡単な記号のみであった通信内容は、1875 年には音声信号を直接やりとりできるようになり、1930 年代のテレビ放送、1970 年代のデータ通信の実用化を経て、現在では多種多様な情報を送ることが可能となっている。

さらに、現在の基盤となっているインターネットは、その階層化されたアーキテクチャにより、特性の異なる方式を用いたアプリケーションを選択し使い分けることができる。電話のように相手と時間を共有して音声や映像をやりとりする事もできれば、電子メールのような時間差のある方式もある。また、このように相手を指定したコミュニケーションもあれば、掲示板のように不特定多数を相手としたものなど、様々な形態を選ぶことができる。

このように、多種多様なコミュニケーションが可能になり、その上で人が様々なやりとりをすることで社会が形成されている。

1.1.2 知人ネットワーク

様々なコミュニケーションが行われる中で、人がそれぞれ持つ人間関係を利用して、コミュニケーションの範囲を決定することが一般的に行われている。つまり、人間にとってもっともコミュニケーションの機会が多いのは、常日頃から共に活動をしている立場にある人であり、すなわち知人である。

このもっとも重要な知人に対して何かを伝えたいとすると、「知人」という曖昧なものを相手としたコミュニケーションを抽象化することが必要となる。これをもっとも単純な形として、知人であるかどうかという真偽に基づいてネットワークを構成したのが、知人ネットワーク(ソーシャルネットワーク)である。元々は、社会分析の手法のモデルであり、図 1.1 のように知人ネットワーク上を辿ることで、伝えたい相手、すなわち知人に対して情報を届けることができる。この知人ネットワークという考え方の登場により、人はより自然な形でのコミュニケーションが可能となった。

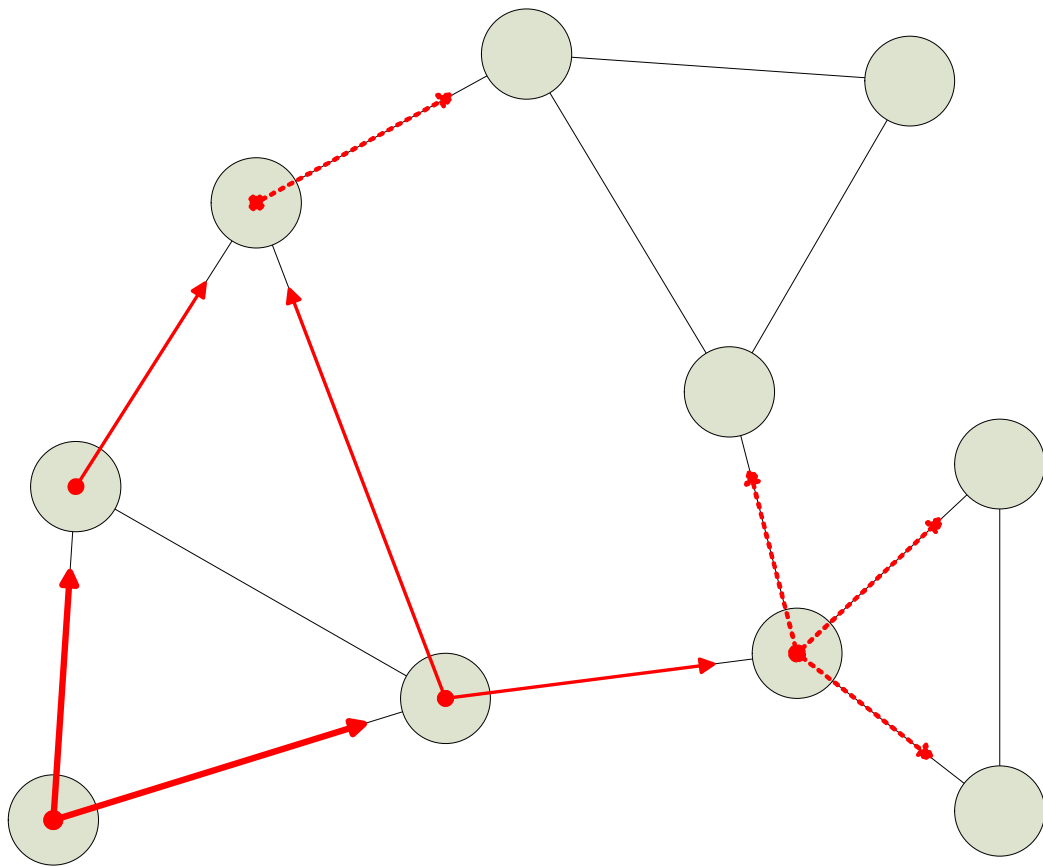


図 1.1: 知人ネットワークに基づく情報の配布

さらに、この知人ネットワークの一点に注目し、その周囲の知人関係を見ることで、どういう人であるかを知ることができる。また、その人が確かにあるオフラインでの誰かであることを確認できる証拠ともなりえる。

このように、知人ネットワークはこれまで人間が行ってきたコミュニケーションをうまくネットワーク上で実現するものとして注目されている。

1.1.3 自分の使い分け

人間社会においては、いわゆる「公私混同は避ける」という考え方として、その時その立場に応じて人格を使い分けることが要求される。自分が他人からどう思われているか、どういう振る舞いが期待されているかをともに、それに対応する人格をもって行動を行うわけである。

周囲が期待するキャラクターをふまえた上で、自分の人格をどのように合わせ、コントロールするかがいわばその人格のキャラクター性といえる。このように、自分の見せ方をコントロールするということは、周囲の期待に反するような情報は見せないということであり、広義の自己情報コントロールとも言える。

これまでは、公私という二つの大きな人格が大まかにこの使い分けに対応していたが、オンライン上では容易に仮想的な人格を作り上げることができる。公として期待された自分の見せ方から切り離された「ネット人格」を持つということである。このように、インターネットの持つ多様なコミュニケーション方法によって多様な人格を使い分けることが可能になった。

しかし、知人ネットワークでは、知人関係の有無という単純な形で抽象化したため、相手とはどういう関係なのかという文脈を無視している。その結果、様々な状況で自分が期待されるものの積集合として、きわめて限定された自分しか表現できなくなってしまう。この現象は、「SNS内の知人が多岐にわたるほど書けることが減っていく」として知られている。

1.2 本研究の目的と意義

このような背景から、人間の活動の中でインターネットの重要性が増すに従って多様なコミュニケーションのあり方を実現することが必要とされている。そのためには、その相手となる知人関係という抽象化はより人間の要求に即した形で様々な状況を表現することができなければならない。なおかつそれを実現するコストがコミュニケーションを阻害しない程度に低くなければならない。本研究の目的は、このような社会的な要求を満たす知人関係モデルを提案し、人間にとってあるべきコミュニケーションを実現することにある。

このようなモデルを実現するために、本研究では知人関係を利用した自己情報コントロールを基盤とするソーシャルネットワークモデルに着目した。これまでのソーシャルネットワークモデルをより人間のコミュニケーションモデルに近い形で再定義し、実環境での実験により評価することでモデルの有意性を検証する。

1.3 用語等の定義

利用者 本モデルを実装したシステムの利用者、一人の人間に相当

人格 一人の利用者は、一つ以上の人格を持つ。

本モデル上での全ての活動は、人格をその主体として行われる。

知人リンク お互いが何らかの信頼関係をもつ知人であることを相互に同意した利用者どうしの関係。ただし、人格をその端点とする

ソーシャルネットワーク (知人ネットワーク) 一つの行動主体である人格を節 (ノード) とし、知人リンクによる紐帯 (リンク) で結んだネットワーク

公開範囲 知人ネットワーク上における情報の到達範囲。

直接隣接するノードまでであれば「知人まで」、隣接ノードを経由してさらにその隣接ノードまで、つまり 2hop 先までであれば「知人の知人まで」と表現する

人格の使い分け 知人リンクを結ぼうとする際や、何らかの発言をする際に、その行動者を自分の持つ複数の人格の中から状況に応じて選択すること

同一人物 ある一人の利用者が使い分けている複数の人格の関係

第三者からは、それらの人格は別人として見える

1.4 本論文の構成

本論文は7章から構成される。

第2章では本研究の先行事例として、既存のソーシャルネットワークモデルについて述べる。それらが本研究の目的を満たしていないことを明らかにし、実現すべき要件を定義する。

これらの要件を元に、第3章において具体的に設計を行う。

第5章では、この設計を実装するアプリケーションについて述べる。第6章では、本システムの評価方法を決定し、それに基づいて評価を行う。

最後に、第7章において本研究の総括と、今後の展望について述べる。

第2章 先行事例

本章では、本研究で提案するコミュニケーションモデルについての理解を深めるため、これまで実現されてきたコミュニケーションモデルについて取り上げる。特に、知人関係に基づくアクセス制御を利用する知人ネットワークについて深く考察を加える。

2.1 コミュニケーションモデル

人間のコミュニケーションモデルは、どのような相手に、どの情報を伝えるかという二つの要素の組み合わせで分類することができる。本節では、まず他の誰かに伝えたい情報の種類について取り上げる。

- 個人情報、コミュニティ(属性)
- 自分が発信する情報(日記/weblog等)
- 知人関係(知人の一覧、紹介文)

2.2 相手の指定

次に、どのように相手を指定するかという点から、コミュニケーションモデルを分類する。各々のモデルを分かりやすくするため、主要な構成要素を図??にて紹介する。

文字の書かれた円で示されるのが、コミュニケーションの当事者となる人間を示している。自分と書かれたものがコミュニケーションを開始しようとする側になり、アルファベットが書かれたものはそれを受ける側となる。アルファベットが名前もしくは識別子を示し、このうちの A、B、C に対してコミュニケーションを取りたいものとする。

小包は相手に見せるべき情報の実体を示す。

個別に指定した宛先に基づくユニキャスト アクセス制御のもっとも単純な形態では、自分がある情報を伝えたい相手をその都度個別に指定して、それぞれに直接その情報を送る。その通信手段は、宛先を指定できるものであれば何でも良く、一般的には電子メールが使われ、必要な秘匿性に依じて暗号化メール等が併用される。この形態を図示したものが図 2.1 である。

人間が全てをコントロールする以上、この形態がもっとも柔軟性も高い。その反面、必要な人数だけ個別に送るため、相手の数に比例してコストが増加する。

また、送り手主導であるため、ウェブサイトのような閲覧者主導のコミュニケーションとは相容れない。

全ての人に対する送信 broadcast

特定の相手群を指定したマルチキャスト 特定の属性や秘密情報(パスワード等)を持つ相手全て

相手のリストアップによる(XCAST等)

情報の配布コストを下げるために、パスワードを利用することができる。これを図??に示した。

ある情報を限定して配布したい場合、あらかじめその情報をパスワードで保護してウェブサイト上に掲載する。その方法として、パスワードによって暗号化するやり方や、HTTPの基本認証などがある。このパスワードのみを、電子メール等で直接公開の対象者に配布する。

知人ネットワーク 特定の相手群を指定する際に、あらかじめ構築された知人関係に基づくことができる。

このモデルでは、図 2.2 のようにあらかじめ自分の知人関係をシステムに登録し、その知人関係に基づいて情報の公開範囲を決定する。



図 2.3: mixi ホームページ

- 現住所
- 誕生日
- 生まれた年
- 出身地
- 職業
- 所属
- 日記
- フォトアルバム

原則として、それぞれの項目ごとに友人まで公開、友人の友人まで公開、全体に公開から範囲を選択することができる。ただし、入力が必要の誕生日、生まれた年については非公開を選択することができる。また、フォトアルバムは特殊になっており、全体に公開、友人までに公開と、既に述べたパスワード方式である閲覧キー認証型を選ぶことができる。これは、この両者のそれぞれの特徴の違いから、どちらも場合に応じて使い分けが必要だと考えられたためであろう。

mixi においては、知人関係はその有無以上の属性を持たない。そのため、一度知人関係を結んだ相手は全て同様の知人として扱われる。知人関係の差異による情報の振り分けなどを行うことは一切できず、あくまで既に述べたように知人ネットワーク上の距離のみがアクセス制御の判断基準として使われる。

2.3.2 livedoor フレンドパーク

livedoor フレンドパークも mixi と同様に一般向けの SNS だが、図 2.4 で示すように知人リンクにグループという属性が付加され、知人関係のカテゴリ化が可能となっている。

livedoor フレンドパークで特徴的な点は、この知人グループに基づいたアクセス制御を実現していることである。mixi と同様に住所や職業を公開することができるが、その場合に、「フレンドパーク参加者全員に公開」、「友達全員に公開」に加えて、その情報を公開する知人をグループ単位で指定することができる。また、日記についてもその公開先を知人グループ単位で指定できるが、日記の記事単位ではなくあくまで日記そのものに対して公開設定が行われる。この点において柔軟性を欠いている。

livedoor が実現する知人関係の区別は、あくまで自らが発信する情報の公開先指定にのみ使われるため、どのような知人が居るかは誰でも知ることができる。この点は mixi と同様である。

ようこそ、[\[ユーザー名\]](#)さん

[マイページ](#) | [友達を招待する](#) | [ログアウト](#)

livedoor フレンドパーク 8版

[トップ](#) [コミュニティ](#) [あなたのプロフィール](#) [マイページ](#)

マイページ: [トップ](#) [プロフィールの変更](#) [ネットワーク](#) [コミュニティ](#) [日記](#) [マイレビュー](#) [メッセージボックス](#)

- メニュー
- [友達の一覧](#)
 - [申請中のリンク](#)
 - [グループの編集](#)

友達の一覧

あなたがお友達に書いた印象の変更や、グループの設定ができます。

- | | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |

1

	<p> [ユーザー名]</p> <p>あなたへの印象が登録されていません</p>	
<p>あなたが招待したメンバーは削除できません。</p>		

	<p> [ユーザー名]</p> <p>クール&ラノバマスターあきたん。いろいろ釋りにしてます! この印象を削除する</p>	
<p>あなたが招待したメンバーは削除できません。</p>		

	<p> [ユーザー名]</p> <p>あなたへの印象が登録されていません</p>	
<p>招待してもらったメンバーは削除できません。</p>		

	<p> [ユーザー名]</p> <p>あなたへの印象が登録されていません</p>	
<p>このリンクをネットワークから外す</p>		

	<p> [ユーザー名]</p> <p>あなたへの印象が登録されていません</p>	
<p>このリンクをネットワークから外す</p>		

	<p> [ユーザー名]</p> <p>あなたへの印象が登録されていません</p>	
<p>このリンクをネットワークから外す</p>		

1

図 2.4: livedoor フレンドパーク 友達の一覧

2.3.3 wyco

「名刺」
立場による見せ方の変化

2.3.4 imeem

imeem はオーバーレイネットワーク上に構築された分散型の SNS である。
知人関係そのものの秘匿性

2.4 複数の人格の使い分け

2.4.1 使い分けを行わない

既存の SNS のほとんどは多層的な知人関係の存在を無視することにより、単純な形での知人ネットワークを構築している。このモデルでは全ての知人関係は一律なものとして扱われる。前述した mixi がこのモデルを採用している。

一つの SNS の中で多層的な知人関係を分離することができないため、用途の異なる複数の SNS を使い分けることでそれらを分離するほかない。しかし、[4] によると複数の SNS に参加しているユーザは全体の 16.3%にとどまる。これは、複数の SNS を掛け持つと言うことに面倒を感じる人が多いためと推測される。したがって、一つのシステムとして、様々な形態の知人関係を包含することのできるモデルを実現することが必要とされている。

2.4.2 公開範囲のグルーピング

複数の人格はあくまで一つのシステム内の存在として表現し、知人関係をグルーピングすることで、自分の情報を見せる相手を限定するという方法がある。これは livedoor フレンドパークが採用している形態である。この方法は、自分の個人情報や日記のように自分から公開する情報に基づいて自分の見せ方を制御するという目的に適応する方法である。

ただし、この方法ではあくまで自分が発信する情報のみを制御することができる。例えば、知人関係そのものの見せ方は制御できない。知人関係から別の人格がどんな活動をしているかを類推するのは難しいことではないだろう。また、たとえば他人が自分に対して記入した紹介文に対して範囲制限をするというのも、あまり自然なやり方とは言えず、モデレートを取り入れようとすればコストも高い。まとめると、この公開範囲のグルーピングという手法は、きわめて中途半端なものであると言わざるを得ない。

2.4.3 複数の ID の併用

一人が複数の人格を使い分ける場合、これまでは第 1 章で述べたように単純に複数の名前を使い分けるといった方法をとっていた。複数の名前を使い分けているという部分は、その当事者がその使い方を工夫することで実現されている。

課金処理等の理由から、複数の ID を一つの利用者に結びつけて管理しているケースがある。例えば、はてなにおけるサブアカウント等がこのケースに含まれる。システム内での行動はあくまで個別の ID に困っているため、この場合も一般的なケースと同様と考えられる。

この方法はいわばシステム外の運用テクニックに依存したやり方であり、使い分けそのものは実現されていない。したがって、利用者の視点では複数の利用者アカウントをまた、知人の情報を知るためには、使い分けている名前の数だけログイン・ログアウトを繰り返す必要があるなど、二重に管理しなければならず、きわめてコストが高い。

また、親しい相手には全て見せるが、そうでない相手には限定した部分のみを見せる、といった包含的なコントロールも困難である。この方法も、本研究の目的を満たすとは言えない。

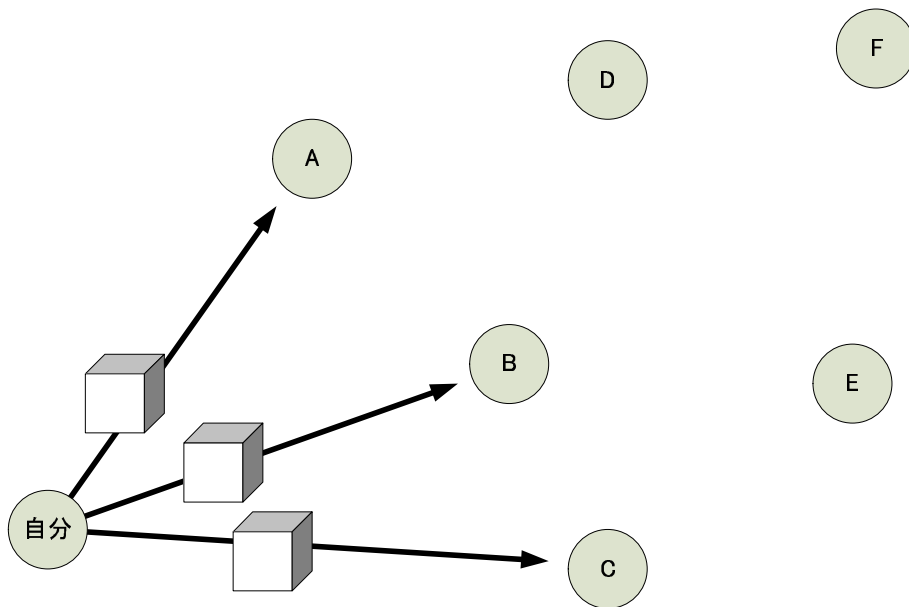


図 2.1: 宛先による範囲の限定

第3章 設計

3.1 要求分析

本研究の目的を実現するために必要とされる、知人関係モデルについて、実際の事例を元に分析を行う。

登場人物を指す代名詞として、以下を用いる

システム 本研究で提案するモデルを実装したシステムを指す。

利用者 システムの利用者となる人間を指す。

主人公 事例の主演となる利用者を指す。

知人関係 あらかじめ主人公との間で双方向で既知の間柄であることを確認した、システムに登録された関係を指す。

知人 主人公と知人関係を持っている利用者を指す。

閲覧者 主人公から情報を得ようとする利用者を指す。

3.1.1 単純な事例

まず、一番単純なモデルとして平坦な知人関係について取り上げる。たとえば、主人公が所属するコミュニティないし組織、たとえば大学の知人だけと知人関係がある場合、この主人公が持つ知人関係は平坦なものであるとすることができる。言い換えると知人の種類は単一である。

この事例では、閲覧者が知人かそうでないか、あるいは、閲覧者と主人公で共通の知人を持っているか、などによる制御が必要となる。これは、知人関係による距離であることから、縦方向の制御とすることができる。

3.1.2 弱い使い分け

次に、様々な背景を持つ知人関係を使い分けることを考える。

主人公は、複数のコミュニティに属しており、それぞれのコミュニティに閉じたコミュニケーションを必要としている。たとえば、勤務している会社の同僚と、大学時代の同窓生というように種類の異なる知人関係を持っている場合、複数の背景となるコミュニティに属していることは公知で構わなくとも、同窓生には内容を知られることなく同僚と仕事上の話題ができることが望ましいだろう。この使い分けは、自分の知人である人の中で制御を行うことから、横方向の制御と言い換えることができよう。

3.1.3 強い使い分け

さらに、より強い使い分けとして、複数の背景による知人関係があり、それぞれの背景での活動をそれ以外の文脈では知られたくないという事例がある。

一つの事例として、オンライン上での情報発信をしている主人公が、その所属組織の立場を隠すことで、発信した情報を色眼鏡で見られることを避けたいというケースがある。この場合、主人公は二つの独立した顔を持ち、閲覧者からはそれぞれが別の人の活動として見えることになる。言い換えれば、ある組織に所属する主人公が存在することを知ることができても、それがオンライン上で情報発信をしている人物と同一人物であることを知ることができないことを保証しなければならない。

これまで実世界の活動では、複数の活動をしていても普通はそれらが結びつけられることはない。しかし、オンライン上では名前に基づく検索などにより容易に別の活動が結びつけられてしまう事が起き得る。従って、これまでは特に気にせずに使い分けることができた状況においても、オンライン上では確実な使い分けを実現することが要求される。

3.1.4 強い使い分け +

強い使い分けによる見え方の分離だけでは満たせないケースが存在する。強い使い分けで紹介した事例では、完全に別個の存在として自分以外の誰からも主人公の持つ複数の活動が同一人物だと知ることができないが、限られた知人に対しては、様々な活動を行っているということを見せたいという状況があり得る。

たとえば、以前の例と同様に主人公はオンライン上での情報発信を行っており、その活動は主人公の所属組織との関連を秘匿している状況において、主人公の知人に対しては詳しい主人公のプロフィールや、知人関係などを紹介したいという場合がある。これは、親しい間柄の相手に対して詳しい背景情報を見せることにより情報発信の助けとなる可能性があるからである。知人の種類によって見せ方

を変える横方向の制御だけでなく、距離の近い人に対してはより深く自己を見せるという縦方向の制御を組み合わせたものと言える。

このような、様々な軸によって自分の見せ方を変化させると言うことを人間は実際に行っており、これらの実現が求められている。

3.2 要求分析

次に、これらの事例からコミュニケーションの要素を抽出し、細かい要求定義を行う。

3.2.1 内容

まずコミュニケーションの内容を定義する。

情報発信 主人公が何らかのトピックに対してウェブサイトやウェブログのような形で発信する一連の情報を指す。

属性の見せ方 主人公が自らの背景について公開する属性情報を指す。

知人関係の見せ方 主人公を取り巻く知人関係、つまり知人の一覧を指す。たとえば主人公自身が全く自分についての情報を公開していなくとも、それぞれの知人もまたその属性情報を公開することから、どういう知人が居るかという点から主人公の社会的な立場を推測することができる。また、SNSによって実装されているような知人からの紹介文も、それを助けている。

これらをもとに、表に起こしたものが表 TODO である。

TODO: 表が入る

3.3 複数人格モデルの概要

前節で明らかにした要件を満たすモデルとして、本研究では「複数人格モデル」を提案する。

たとえば同じ組織のつながりなどの特定の背景を共有する知人関係と、その知人とのやりとりを、知人関係における文脈と定義する。主人公は、この文脈を状況に応じて使い分けることにより、多様な知人関係を構築している。

単純な事例では、一つの文脈で全てを満たすことができる。「弱い使い分け」を必要とする状況では複数の文脈を使い分けることになり、文脈ごとに情報を公開するかどうかを定義することで実現する。

さらに、この文脈ごとに個別の行動主体が存在するとみなし、それぞれの行動主体を「人格」として定義する。複数の人格を使い分けて行動することにより、「強い使い分け」を実現することができる。

3.4 複数人格モデルの詳細

3.4.1 人格と文脈

既に述べたように、別人として振る舞う必要があるような状況において、それらを別個の存在として表現したものが「人格」である。このモデル内では、人間は1つ以上の人格をその行動主体として操作する。

また、それぞれの人格が持つ行動や属性情報、知人関係などを文脈と呼び、一つの人格に対して、知人関係などの区分に応じて複数の文脈を使い分けることができる。

原則として、名前ないしニックネームなどは人格ごとに別のものを設定する。同じ名前を名乗ることも可能だが、検索などにより分かれて存在するべき複数の人格が、本人の意思に反して同一人物であることが発見されてしまうことになる。

同様に、誕生日や職業といったその人が公開する情報についても、人格単位で個別に設定する。それらの公開範囲は、人格ごと、もしくはさらに細かく文脈ごとの知人関係に基づいて制限される。

知人関係 知人関係そのものがその当事者がどんな人であるかをあらゆる重要な情報となりえる。そもそも知人ネットワークというモデル自体が、知人関係に基づいて社会的な信頼性を担保するという面が主目的であり、オンライン空間における知人ネットワークにおいても同様である。特に、一般的なSNSにおいて実装されることの多い紹介文機能はその特徴を良くあらわしている。

よって、ある人格において周囲から期待されない面を隠すためには、多層的になっている知人関係そのものを人格ごとに分離しなければならない。

活動の行為者 システム内のグループへの所属や掲示板への書き込み等、オンライン上での活動の行為者がどの人格であるかを、利用者の負担にならない直感的な方法によって使い分けられなければならない。

3.4.2 一方向的な同一人物関係の開示

第??節で述べたように、複数の人格は全く別の存在として使い分けられるだけでなく、知人関係の種類によっては和集合的に別の人格の存在を見せても良いという関係が存在する。この複数の人格の間の実現するため、同一人物の別人格の存在を、一方向的に開示できなければならない。

3.5 既存のモデルとの比較

TODO: ここから下は、ここに書くべき内容として取捨選択し書き直す。

これまでソーシャルネットワークにおいては、各ノードが利用者一人に割り当てられ、リンクをその利用者間の知人関係というのが一般的な定義だった。

本モデルでは、要件を満たすモデルを構築するため、以下のようにこれを拡張し、再定義する。

ノード 各ノードは、各々の利用者が相手や状況に応じて使い分けている複数の人格の一つを指す。利用者は、それぞれ一つ以上の人格を持ち、自分の情報や、発言などはそれぞれの人格に結びつけられる。

リンク 本モデルの定義では、リンクは利用者の間ではなく、利用者が持つ個別の人格の間を結ぶ。このリンクに対し、二つの属性を付加する。

一つは方向性で、双方向か単方向どちらかに属する。双方向なリンクは、起点と終点を区別しないが、単方向なリンクはそれらを区別する。双方向リンクは、方向がそれぞれ逆の二本の単方向リンクであらわすこともできる。

もう一つの属性が可視性である。二つの人格の間に存在するリンクそのものが公開されるかどうかをあらわす。誰からもそのリンクが存在することを知らることができる公開状態と、リンクの当事者の知人にのみその関係が分かる限定状態を取り得る。前述の方向性が単方向の場合、リンクの起点の人格を基準とし、その直接の知人のみに公開される。

3.6 リレーションによるモデルの表現

前節で再定義した新しいソーシャルネットワークを元に、本研究が提案する新しいモデルをER図に表したものが図3.1である。

利用者は複数の人格(Person)を持つ。リンクは、それぞれの人格の間に構築される。通常の知人関係は、双方向かつ公開された可視性を持ち、これはこれまでのソーシャルネットワークと同様のモデルである。

一方、一人の利用者が使い分ける人格のうち、一方向的に同一人物であることを明かしたい場合は、同一人物リンクと呼ぶ特殊なリンクを構築する。このリンクは、単方向かつ限定された可視性を持つ。リンクの起点となっている人格の直接の知人のみが、そのリンクで結ばれたもう一つの人格が同一人物であることを知らることができる。

何らかの情報を公開する場合には、自分のどの人格を使って公開するのか、どの範囲に限定して公開するのかを指定する。

一方、閲覧する場合には、自分が持つどれかの人格が相手が指定した公開範囲に含まれるかによってアクセス制御が行われる。

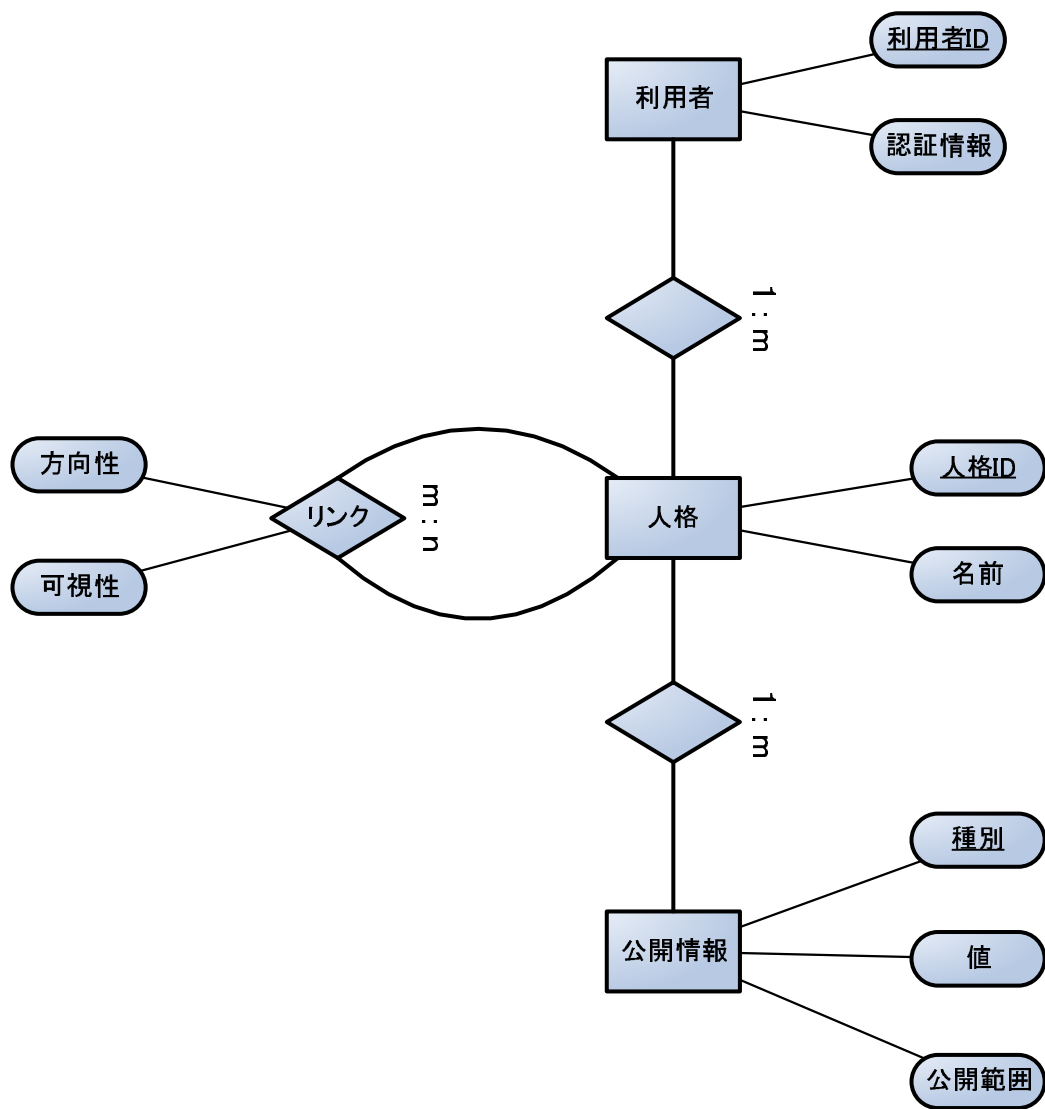


図 3.1: ER 図によるソーシャルネットワークの定義

3.6.1 複数の人格の管理

利用者は、それぞれに割り振られたユーザIDとパスワード等の認証情報を用いてログインし、自分が持つ複数の人格を管理する。

それぞれの人格は別個に個人情報を持っており、さらにその情報単位で公開範囲を設定する。図 3.2 に示す。

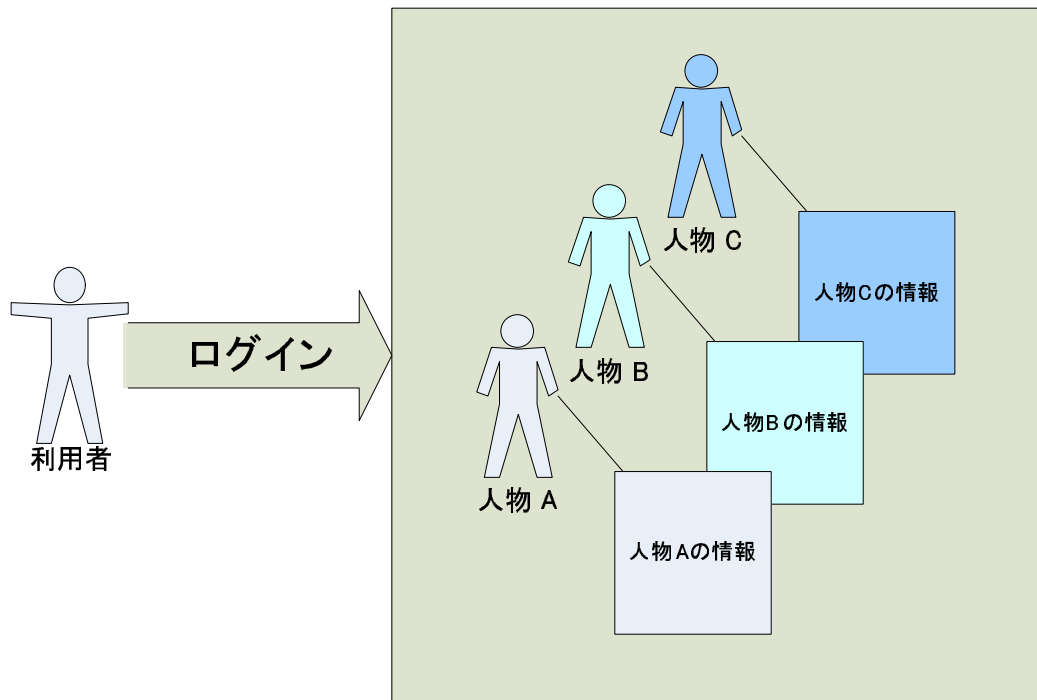


図 3.2: 複数の人格の管理

3.6.2 知人リンクの構築

図 3.3 は、知人リンクの構築プロセスを UML シーケンス図によってあらわしたものである。

原則は、知人リンクの構築には双方の合意が必要である。これを実現するため、片方向のみ同意している構築途中の段階では知人リンク要求として別のリレーションによって管理される。この知人リンク要求は、知人リンク要求を受けた側に要求が来ていることを通知する以外、アクセス制御や知人の一覧等には何も影響を及ぼさない。

3.6.3 同一人物リンクの構築

図3.4では、同一人物リンクを構築することで、知人関係がどのように見えるかをあらわしている。

四角で囲った二人が同じ人間である二つの人格をあらわし、赤い点線によって、同一人物関係を示した。この例では、右の人格から左の人格に同一人物関係を設定し、右の人格は左の人格という別の面を持っているということを、その知人にもみ見せている。

左の人格が持つ知人関係や、自己紹介等が右の人格に加えて知ることができ、右の人物に対する知人の評価にフィードバックされる。

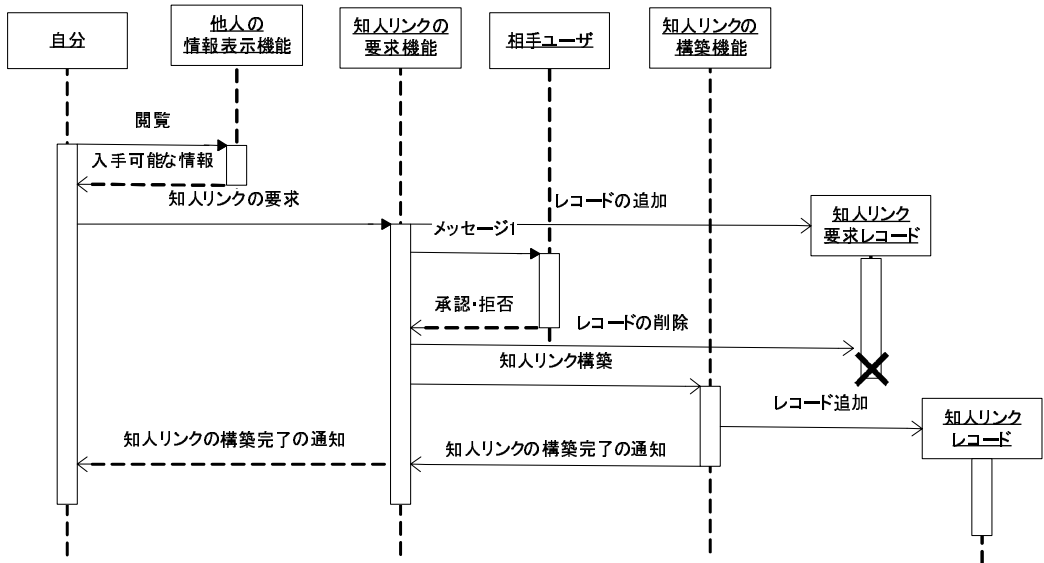


図 3.3: 知人リンクの構築

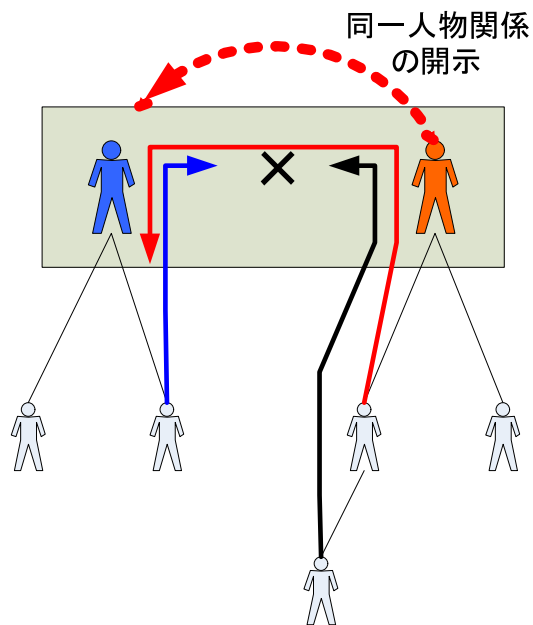


図 3.4: 同一人物リンクの構築とその影響

3.6.4 ソーシャルネットワークによるアクセス制御

知人ネットワークでの経由ノード数すなわち距離に基づいて、情報の配布範囲を決定する。もっとも基礎的な「知人まで」「知人の知人まで」での場合を図 3.5 に示した。

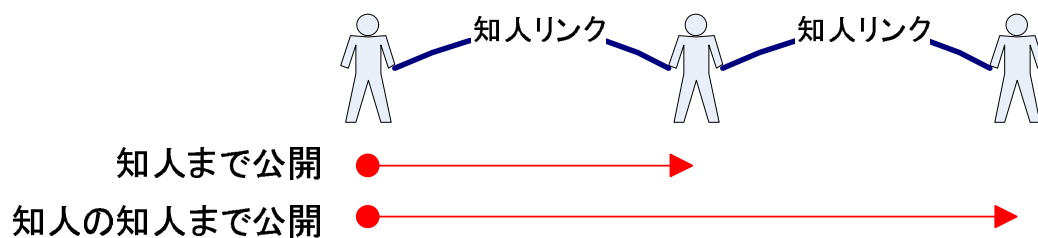


図 3.5: ソーシャルネットワーク上の距離に基づく公開範囲の指定

単純な知人ネットワークとの違いは、人格どうしの距離を算出する際に、同一人物リンクは距離に数えないということである。これは、同じ人物の別の側面として見える以上、知人関係の距離に対してもあくまで側面であり知人関係ではないからである。

第4章 プレ実験

第5章にて述べた実装を用いて実験運用を行った。本節ではその結果を示し、考察を行う。

4.1 プレ実装の目的

- 方向性の評価
- 実装要件の洗い出し

4.2 プレ実験の概要

実験目的 本実験の目的は、大きく二点に分けることができる。

一点目は、本研究で提案するモデルが実装可能なものであるとともに、その使い勝手の変化について確認することである。

二点目は、人格を使い分けるモデルを採用する SNS を運用し実際の利用者に触れてもらうことにより、社会的に本研究が要求され意義が有るものであることを示すことである。

実験の参加者 今回の実験では、モデルの有意性を明らかにするという目的から多層的な知人関係を再現することが望ましく、単純に参加者を増やすだけでは実験の意義が薄れてしまう。そこで、以下の二点を前提に実験の参加者を募った。

- 知人ネットワークが著しく分断することを防ぐため、著者を起点とする招待制を採用した。
- 知人ネットワークの広がりを重視するため、著者の知人を媒介に、著者と直接の知人を持たない利用者を増やした。

具体的には、以下の手順で参加者を応募した。

システムにはメールアドレスを指定して実験への参加依頼を送信する機能を実装した。この機能を利用して実験に参加してもらうことを本実験において招待と呼ぶことにする。

1. システムに初期から存在する管理者ユーザから、著者自身を招待した。
2. 著者は、二つの人格を用意し、既存の知人関係に基づいてそれぞれ5人程度を招待した。
3. 実験の参加者となった被験者に対し、さらにその知人関係をもとに別の被験者を募り、招待を行った。

4.2.1 実験結果

実験状況 本実験はその性質から継続的に運用を続けており、本章にて示す数値はあくまで執筆時点での数値である。

平成17年7月5日時点での利用状況を表4.1に示す。

表 4.1: 運用実験の利用状況

項目名	状況
利用者数	28人
登録されている人格数	43人

知人ネットワークの分析 本システムに蓄積された知人関係情報から判明するものとして、表4.2に各利用者が所有する人格の数による度数分布を示す。

表 4.2: 所有する人格の数による度数分布

所有する人格の数	人数
1つ	17人 (60.7%)
2つ	7人 (25.0%)
3つ	4人 (14.3%)
合計	28人

この表から4割前後の利用者が複数の人格を所有していることが分かる。さらに、同一人物関係の設定状況を集計した物が表4.2である。

同一人物関係が設定されたのは合計9組で、うち7組が片方向、2組が両方向の同一人物関係が設定された。後者は、両方向で設定を行うことでどちらの知人に対してももう片方の人格の存在が見えることになり、実質可視範囲が知人までに限られた知人リンクと言うこともできる。

4.2.2 アンケート調査

利用者を対象として、アンケート調査を行った。アンケート調査の結果については付録 A に載せる。

アンケートは、システムにログインした状態で表示される利用者自身のホームページのおしらせ欄にて告知し、そこからアンケートシステムにリンクを設置した。

被験者の特徴

SNS の利用状況 (Q101-102) 今回、著者が直接招待した被験者は既存の SNS に慣れていると思われる人を対象とし、さらにその招待者も既存の SNS の知人関係を利用して被験者を増やしたため、回答者全員が毎日以上の頻度で SNS を利用しているという回答を得た。

SNS への参加形態 (Q103-104) 日本国内の SNS の特徴として、実名によらない利用者が少なくない [5] という点がある。今回のアンケート調査においても、実名参加者が 5 人、それ以外が 4 人と [5] における実名登録者 43.22% という結果と近い回答を得た。

また、実名を使わない理由として、以下のような意見が得られた。

- 検索によって SNS に参加していることが分かってしまう。
- 必ずしも、自由に意思を表明できない。
- 実名で参加していたときにつきまとわれた。
- ハンドルネームの方がよく知られている。

実名で参加している場合には、言論が阻害されたり、ストーキング被害が実際に起きているということが判明した。

異なる自分の使い分けについて(Q201) Q201において、異なる自分を使い分ける事にたいする需要を調査した結果、使い分けが不要であるという意見は皆無であった。ただし、完全に別人格として振る舞うという本研究のモデルまでは必要とせず、「(日記等の)内容の公開にコントロールがかけられれば、それで足りる」という回答もあった。その一方で、「実名を明らかにしない方がうまくいく関係もある」回答もあった。したがって、第2.4.2節で紹介した知人関係のグルーピングとそれによる公開範囲の選択を本モデルの上で併用することが望ましい。

複数の人格を設定したかどうか(Q202-203) Q203において、複数の人格を使い分けるコストについて調査したが、手間とは感じない人が多数であった。そもそも人格を使い分けている時点で、コストと比較して人格を使い分けるメリットを取ったと考えられる。

複数の人格というモデルについて(Q204) 本研究の目的である、複数の人格を使い分けるというモデルに対しては比較的好意的な意見が得られたが、複数の人格がある場合に、その共通部分をどのように扱うかという点に対して不安を抱いていることが分かった。

表 4.3: 同一人物関係の設定状況

同一人物関係の種別	人数
片方向	7組
両方向	2組
合計	9組

第5章 アプリケーションの設計と実装

5.1 実装の方針

本研究で提案するモデルは、あくまで人間のコミュニケーションの抽象化であり、様々なアプリケーションによる実現が考えられる。

今回は、実環境で広く利用してもらうことで評価を行うため、既に普及している形態に近い形での実装が好ましいと判断し、ウェブサイト上での公開サービスとして実装を行った。さらに、mixi や GREE[6] のような既存のサービスと競合することで、それらとの比較により定性的に評価することができる。

5.2 実装環境

本実装は、表 5.1 の環境で行った。

表 5.1: 実装環境

項目名	スペック等
ハードウェア	PC/AT®互換機
CPU	Intel®Celeron®CPU 2.40GHz
OS	FreeBSD®5.4-RELEASE
開発言語	Perl 5.8.6
フレームワーク	Catalyst 5.22 mod_perl 2.0.0 / Apache 2.0.54
RDBMS	MySQL 4.0.24

今回の実装ではフレームワークとして Catalyst を採用した。Catalyst は Perl モジュールとして構築され、MVC アーキテクチャの Web アプリケーションに必要な枠組みを提供する。さらに本実装では、Class::DBI による O/R マッピングと Template-Toolkit によるテンプレートエンジンを、それぞれ MVC アーキテクチャにおけるモデルとビューに利用した。

RDBMS には MySQL を採用した。これは、ストアードプロシージャのような機能面と、パフォーマンス面から PostgreSQL と比較した結果、利用者が増えた場合のパフォーマンスを優先したためである。既に述べたように、データベースへのアクセスは `Class::DBI` による O/R マッピングを経由しているため、今後何らかの理由で PostgreSQL を含め他の RDBMS に移行することは容易である。

今回は比較的小規模な実験を想定したためシステムを一台のホスト上に全て実装したが、利用者の増加や、コミュニケーションの活発化に応じてアプリケーションや RDBMS の分散化が必要となるだろう。

5.3 全体の概要

図 5.1 にシステム全体のモジュール構成を示す。

`Perso` で始まるモジュール群が、今回実装した部分であり、それ以外は既存のフレームワークやモジュールとなっている。

全体は、大きく三つに分かれている。それぞれ、アプリケーション層、高レベルモジュール層、低レベルモジュール層である。これを MVC アーキテクチャと比較すると、コントローラとビューがアプリケーション層に含まれ、モデルが高レベルモジュール層、低レベルモジュール層の二層に分かれる。

`Perso::C` MVC アーキテクチャにおけるコントローラに相当し、Catalyst フレームワークから呼び出され、利用者からのリクエストを処理する。

必要に応じて高レベルモジュールを呼び出し、要求された更新を行った上で `Perso::V::TT` に遷移する。

`Perso::V::TT` 同じくビューに相当する。コントローラから処理を遷移され、Template-Toolkit によるテンプレートエンジンを用いて利用者に処理結果を出力する。

`Perso::M::Network` 高レベルモジュール層に位置する。低レベルモジュール層を細かく制御し、ソーシャルネットワークに必要な様々な機能をアプリケーション層に提供する。

`Perso::M::CDBI` 低レベルモジュール層に位置する。`Class::DBI` による O/R マッピングを実装し、高レベルモジュール層が必要とするデータベースへのアクセスを抽象化する。

表 5.2 に各テーブルとクラスの間関係をまとめた。

RDBMS 既に述べたように RDBMS には MySQL を採用している。各テーブル間の関係を図 5.2 に示す。

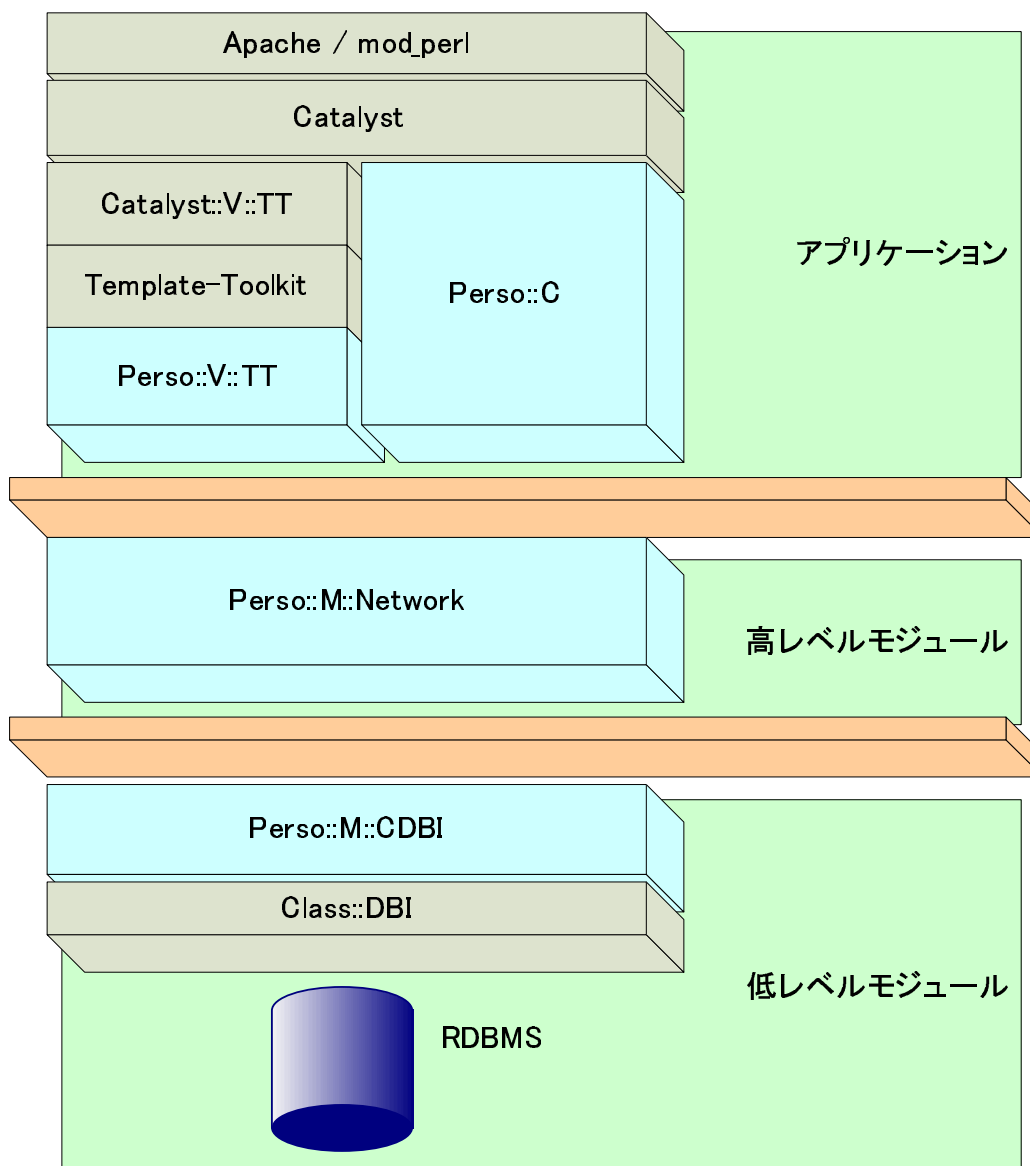


図 5.1: システム全体のモジュール構成

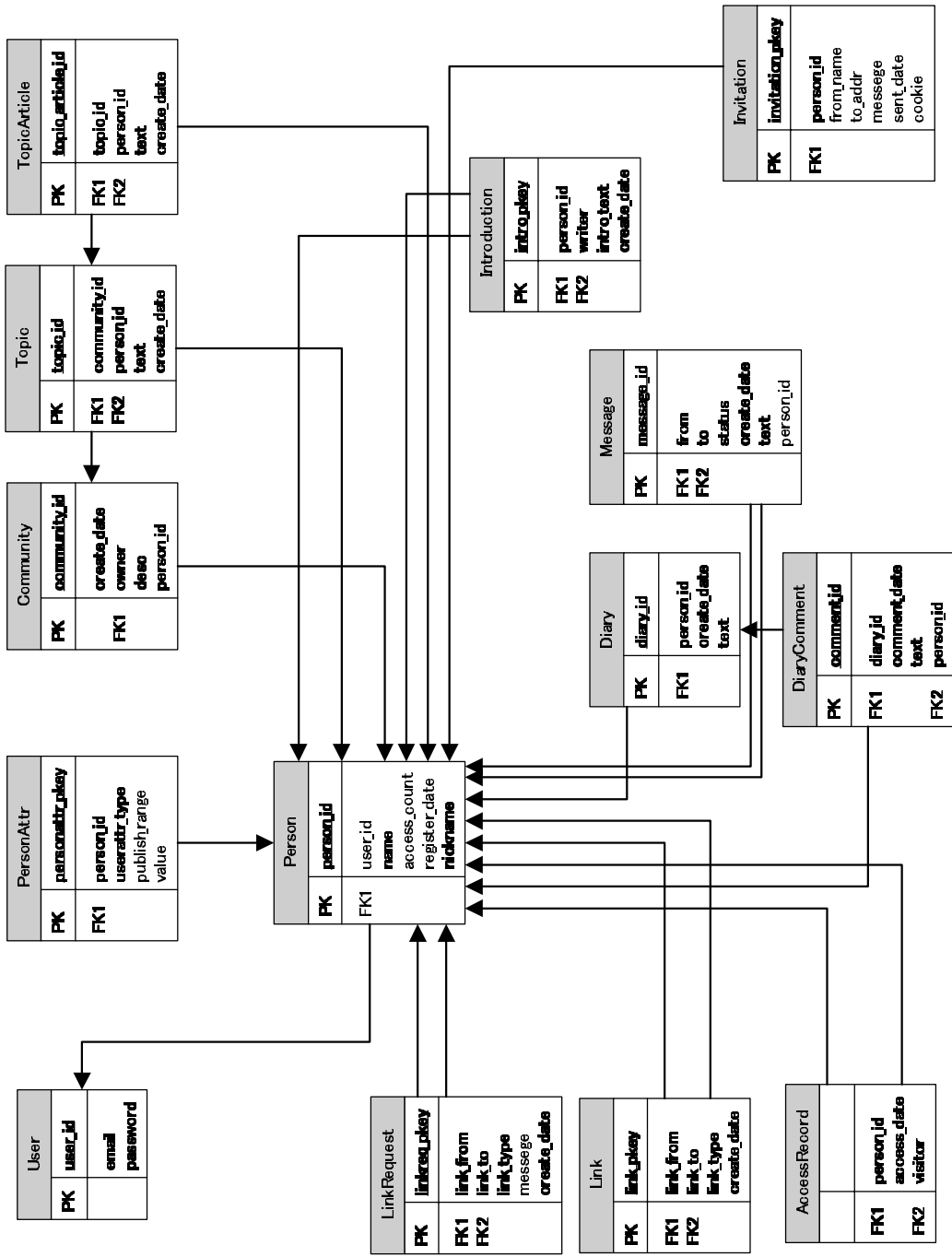


図 5.2: 各テーブルの関係

5.4 サイト構成

図 5.3 に、本システムのサイト構成を示す。

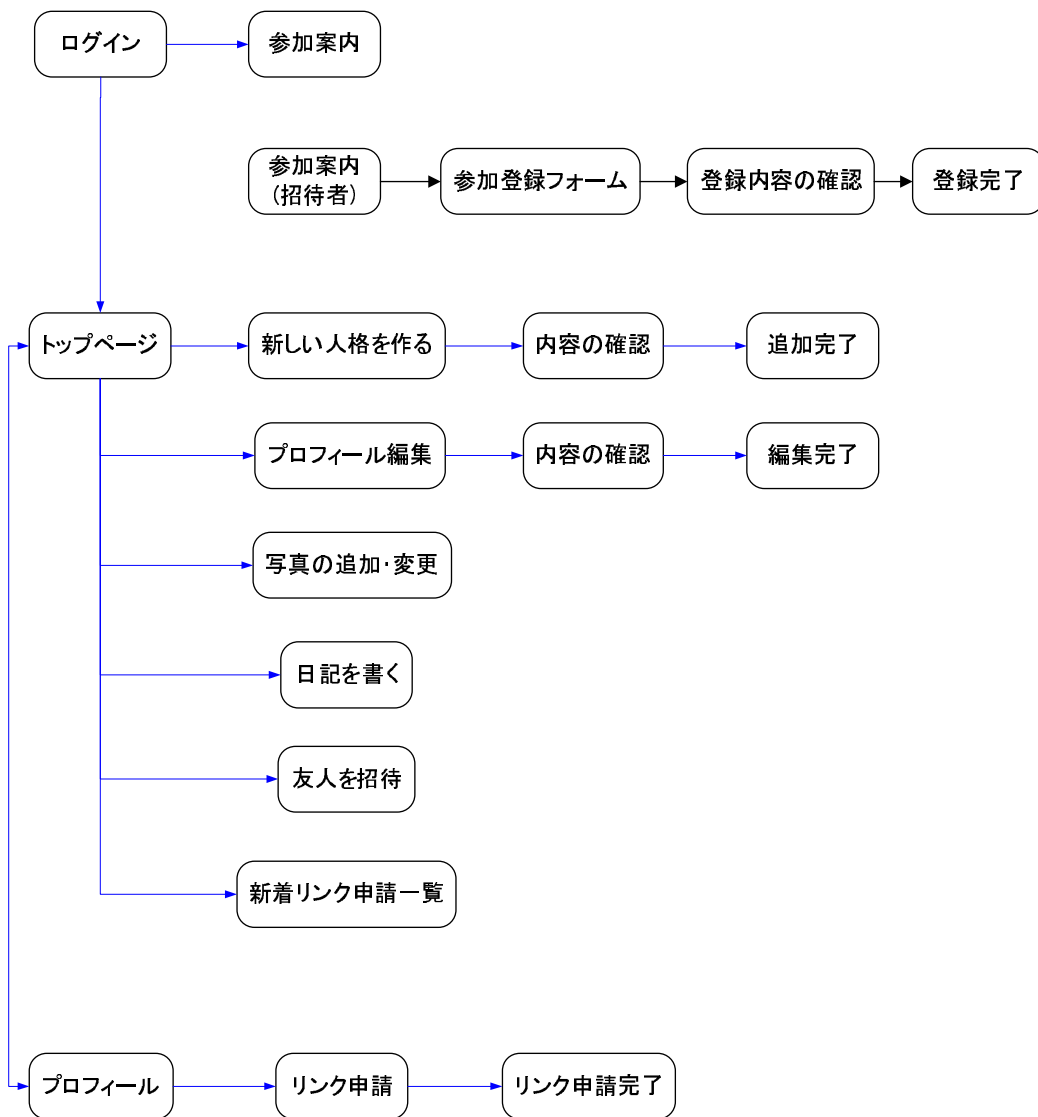


図 5.3: Perso ウェブサイト構成

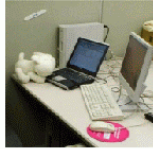
5.4.1 利用者自身のトップページ

図 5.4 は利用者自身のトップページである。自分が持つ人格それぞれに、プロフィール編集、知人関係が用意されている。

また、「友達の日記」欄では自分が持つ全ての人格における知人全員の新着日記へのリンクが用意されている。利用者はいくまで一人の人間であり、その視点からは周囲の知人は一様に扱われる。

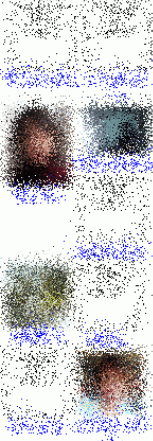
なお、知人リンクの申請を受け取ると上部の「お知らせ」欄に「リンク申請が～件、来ています」と表示される。

Perso

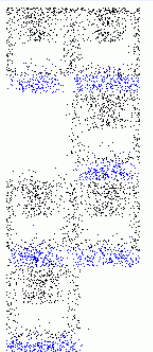


[写真の追加・変更]

まさの友達



ろゆの友達



お知らせ

- ようこそPersoへ！
- [アンケートにご協力ください!](#) (新しいウィンドウで開きます)
- [リンク申請が2件、来ています](#)

プロフィール

- [まさ \(仲山 昌宏\) \[編集\]](#)
- [\[編集\]](#)
- [新しい人格を作る](#)

友達の日記

- 2005/07/04: [\[編集\]](#)
- 2005/07/03: [\[編集\]](#)
- 2005/07/03: [\[編集\]](#)
- 2005/07/03: [\[編集\]](#)
- 2005/07/03: [\[編集\]](#)
- 2005/07/03: [\[編集\]](#)

2005/07/04 22:16:59 [まさ]Re: リアクションが来ねえ (友達まで公開)

ごめんなさい、実は懇親会欠席なんです……。

2005/07/03 22:55:17 [まさ]Re: 新しいP2P開発している (友達まで公開)

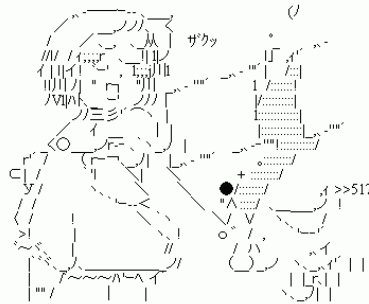
新規でP2Pアプリ作るよりも、Winyyのソースコードを改変して遊んでみる、という遊びを思いついてみました。

作者の許可が無いといけないんじゃないかという気もしますが、READMEとか見ても、リバースエンジニアリングや改変が禁止されているわけじゃないんですねえ。

しかも、元にするソースコードは本物じゃなくて、バイナリからの逆生成物。こういう場合はどこにどんな権利が絡んでくるのかな。

なんか「Winyyの技術」も出ないし、そういう遊びをするとどれくらい噛みついてくる人が出るかな。

2005/07/02 03:23:44 [Aki]もういーくつーねーるーとー (友達の友達まで公開)



2005/07/02 00:38:36 [まさ]というわけで瀕死です (誰にでも公開)

```
( ㄩ ) ゴルナサイゴルナサイ
( (
( ㄩ ) スマゼンスミマゼン
( )
( ㄩ ) コトオリエス
( )ZZ
```

図 5.4: 利用者のホームページ

5.4.2 人格ごとのページ

本研究の目的に基づき、自分以外の情報は人格ごとに別人として扱われる。したがって、利用者ごとのページは人格ごとに用意される。図 5.5 にその一例を示す。

Perso トップページ 日記を書く 友人を招待 ログアウト

まさ (仲山 昌宏)

[\[写真の追加・変更\]](#)

まさの友達

プロフィール 編集

名前	仲山 昌宏
誕生日	7/17
生まれた年	1980
自己紹介	Perso管理人の中の人です。 こんばんは。
性別	male

Mon Jul 4 22:16:59 2005 Re: リアクションが来ねぇ

ごめんなさい、実は懇親会欠席なんです……。

Sun Jul 3 22:55:17 2005 Re: 新しいP2P開発している

新規でP2Pアプリ作るよりも、Winyyのソースコードを改造して遊んでみる、という遊びを思いついてみました。

作者の許可が無いといけないんじゃないかという気もしますが、READMEとか見ても、リバースエンジニアリングや改造が禁止されているわけでもないんですね。

しかも、元にするソースコードは本物じゃなくて、バイナリからの逆生成物。こういう場合はどこにどんな権利が絡んでくるのかな。

なんか「Winyyの技術」も出ないし、そういうお遊びをするとれくらい眺みついてくる人が出るかな。

Sat Jul 2 00:38:36 2005 というわけで瀕死です

```
( Ⅵ ) ゴルナサイゴルナサイ
( (
( Ⅵ ) スミセンスミセシ
} }
( Ⅵ ) コトオラデス
} } }
```

© 2005 Perso

図 5.5: 人格ごとのページ

利用者自身のホームページと比較すると、ある人格一つだけに関する情報のみが表示されていることが確認できる。日記や知人の一覧に注目すると、今表示されている人格のものではない記事は表示されていないことが明らかである。

5.4.3 リンク申請

知人リンクを構築する手順は第 3.6.2 節で述べた。

まず、図 5.6 にて知人リンクの要求機能の画面を示す。各人格ごとのページにおいて、自分自身もしくは直接の知人でなければ、この知人リンクの要求へのリンクが設置される。

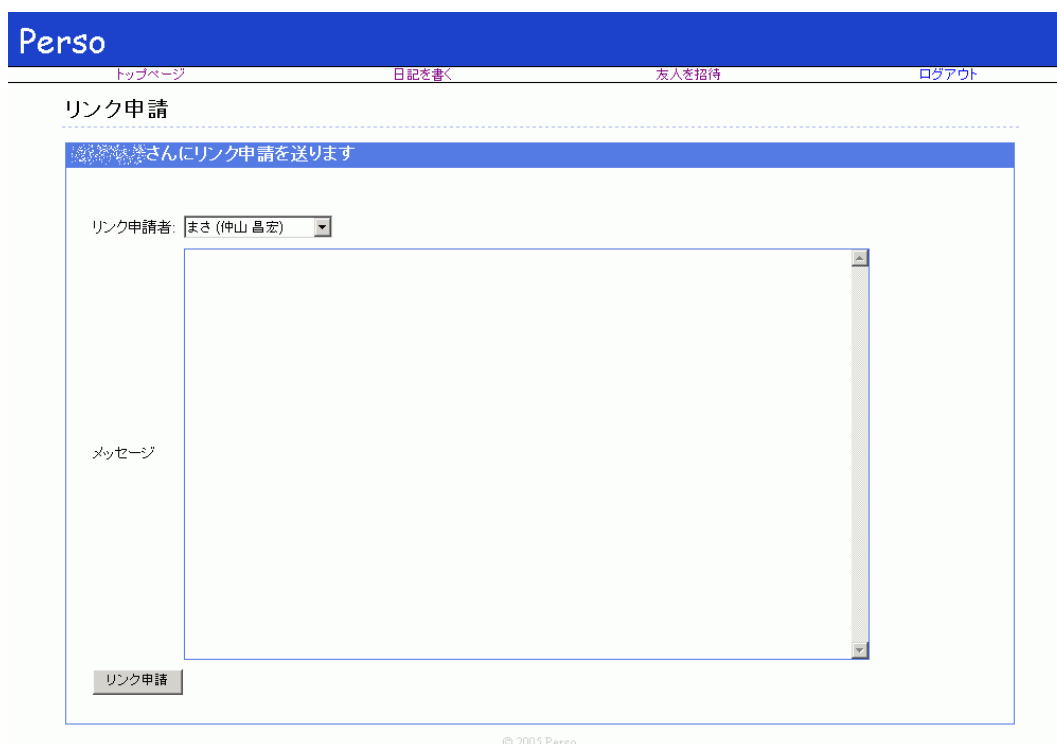


図 5.6: 知人リンクの要求

次に、図 5.7 に、知人リンクの承認・拒否画面を示す。未処理の知人リンク要求がある場合のみ、利用者自身のホームページにてこのページへのリンクが設置される。この画面での選択によって知人リンクが構築される。

5.4.4 日記の登録

日記は記事単位で、どの人格で書くのか、どの範囲まで公開するのかが選択される。図 5.8 に登録画面を示す。

このように、まだ実装されていないものも含めて、全ての情報発信についてそれぞれの人格が行動主体となるのかが選択しなければならない。



図 5.7: 知人リンクの承認・拒否画面

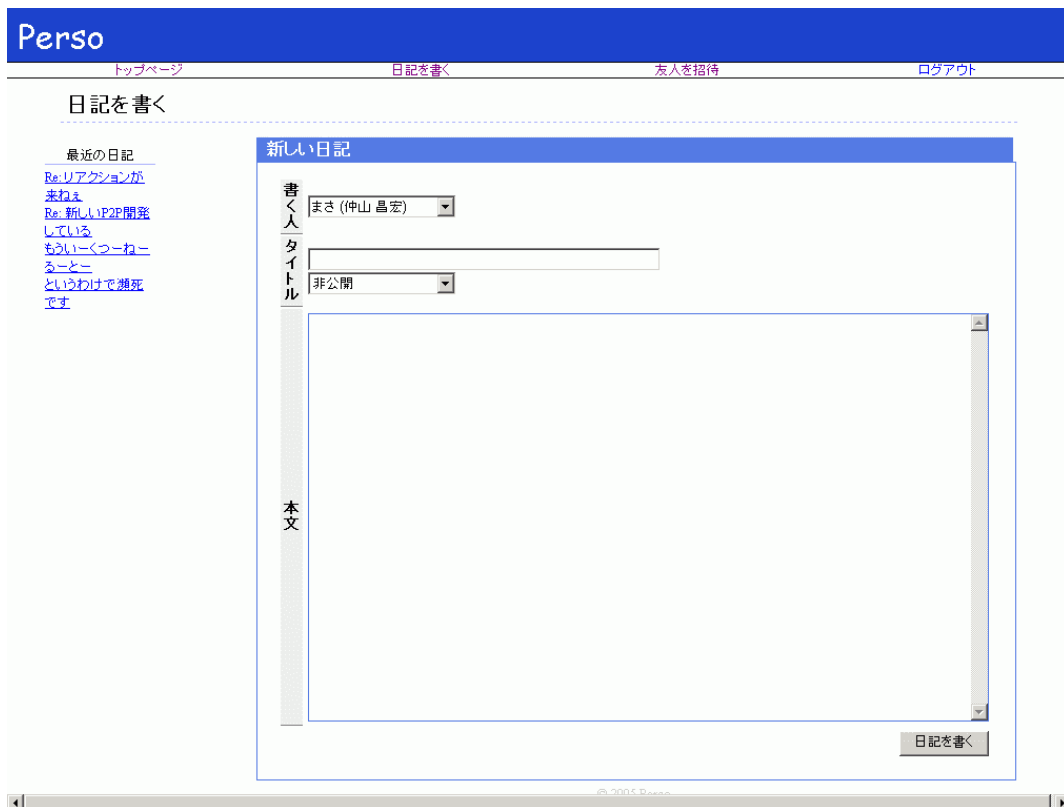


図 5.8: 日記の登録画面

5.4.5 プロフィールの編集

各利用者は、それぞれの人格に付随するプロフィールを管理する。図 5.9 がその編集画面である。各項目ごとに、知人ネットワーク上の距離に基づく公開範囲を設定することができる。

The screenshot shows the 'My Profile' editing interface. At the top, there's a blue header with 'Perso' and navigation links: 'トップページ', '日記を書く', '友人を招待', and 'ログアウト'. Below the header, the page title is 'プロフィールの編集'. The main content area is titled 'My Profile' and contains several form fields:

- ニックネーム**: Text input with 'まさ' entered.
- 名前**: Text input with '仲山 昌宏' entered.
- 性別**: Radio buttons for '男性' (selected), '女性', and '無回答'. To the right is a dropdown menu set to '誰にでも公開'.
- 生まれた年**: Text input with '1980' entered. To the right is a dropdown menu set to '友人まで公開'.
- 誕生日**: Date picker showing '7 / 17'. To the right is a dropdown menu set to '友人の友人まで公開'.
- 自己紹介**: A large text area containing the text: 'Perso管理人の中の人です。こんばんは。'
- 同一人物関係**: A checkbox labeled 'チェック' and text: 'まささんの友達には、チェックを入れた別人格が同一人物だと分かるようになります。'

At the bottom of the form is a '変更' (Change) button. The footer of the page contains '© 2005 Perso'.

図 5.9: プロフィールの編集

下部にある「同一人物関係」が、本研究で提案している複数の人格間の関係を設定するものである。編集を行っている人格が持つ知人に対して、ここで設定した別の人格が別の側面として存在することを見せることができる。

ここで設定された同一人物関係は、知人ネットワーク上での距離演算に影響を与え、利用者のホームページにおける「友達の日記」欄などにも知人の別人格が書いている日記が見えるようになる。

5.5 各モジュールの解説

5.5.1 利用者の管理

本システムの利用者の情報は、`Perso::M::Network::User` クラスによって管理される。主キーとして利用者 ID(`user_id`) のほか、メールアドレスとパスワードのみを持つ。

5.5.2 人格の管理

知人ネットワークのノードとなるそれぞれの人格は、`Perso::M::Network::Person` クラスにて実装した。人格を単位となる知人ネットワーク上の演算もこのクラスに含まれる。

繰り返し述べたように、各人格には人格 ID(`person_id`) が主キーとして割り当てられている。本システム上での行動はこの人格 ID に結びつけられる。また、所属する利用者 ID を保持し、複数の人格が利用者に結びつけられる

知人ネットワークの距離演算 `Perso::M::Network::Person` のインスタンスメソッド `hops` により、知人ネットワーク上での目的ノードまでの距離を演算する。ただし、現在は演算コストと必要性を吟味した上で、2hop 以上は最大値 (無限遠) としている。

ソーシャルネットワークによるアクセス制御に関する機能 第 5.5.2 節で述べた知人ネットワーク上の距離に基づいて、入手可能な情報を取得する機能を、いくつかのインスタンスメソッドとして実装した。

`readable_attrs` メソッドは、指定した利用者が、どれかの人格から読むことができる属性情報を取得する。この属性情報は、利用者自身が自分の人格毎に設定した情報である。

`diary` メソッドは、同様に取得可能な日記を取得する。

5.5.3 知人ネットワークの管理

知人ネットワークを管理する機能は `Perso::M::Network` クラスにて実装した。

知人リンクの構築に関する機能 知人リンクを構築する機能は、`Perso::M::Network` のクラスメソッド `make_link` として実装した。リンクを構築する二つの人格の ID を引数とする。

また、知人リンクを最終的に構築する以前のプロセスについては、`link_request` テーブル (`Perso::M::CDBI::LinkRequest` クラス) とコントローラによって実現され

ている。双方の承認プロセスが完了した後、このメソッドによって知人ネットワークに登録される。

このメソッドによって登録される知人リンクは、既に述べたとおり両方向である。

同一人物リンクの構築に関する機能 同様に、同一人物リンクを構築する機能は、`Perso::M::Network` のクラスメソッド `make_link_sameuser` にて実装した。このメソッドによって登録される知人リンクは片方向である。

表 5.2: RDBMS 上のテーブルの抽象化

オブジェクト名	内容	対応するテーブル
Perso::M::CDBI::User	利用者	user
Perso::M::CDBI::Person	人格	person
Perso::M::CDBI::PersonAttr	人格の属性情報	person_attr
Perso::M::CDBI::Link	知人リンク	link
Perso::M::CDBI::LinkRequest	知人リンク要求	link_request
Perso::M::CDBI::Diary	日記	diary
Perso::M::CDBI::DiaryComment	日記コメント	diary_comment
Perso::M::CDBI::Message	メッセージ	message
Perso::M::CDBI::Introduction	紹介文	introduction
Perso::M::CDBI::Invitation	招待者	invitation
Perso::M::CDBI::Community	コミュニティ	community
Perso::M::CDBI::Topic	コミュニティ内トピック	topic
Perso::M::CDBI::TopicArticle	トピック書き込み	topic_article
Perso::M::CDBI::AccessRecord	アクセス履歴	access_record

第6章 評価

本研究の目的は、多層的な知人関係をより要求に即した形で抽象化することにより、人間の求める多様なコミュニケーションを実現することである。本章では、第??節にて定義したシステム要件を満たし、目的を実現できていることを確認する。また、第5章にて紹介した実装による運用実験の結果について、その知人ネットワークの状況とアンケート調査から考察を行う。

6.1 事例との比較

本研究で提案するモデルが、第??節にて定義した要件を満たしていることを確認する。

6.1.1 単純な事例

この事例では、単一の人格・文脈のみを利用することで実現できる。全ての公開された情報は知人関係による距離によってのみ公開範囲を設定できる。

6.1.2 弱い使い分け

この事例では、様々な背景による文脈が存在するが、それらの背景があることを隠す必要性はない。そこで、単一の人格に、どんな関係かに基づいて複数の文脈を使い分けることにより、情報の公開相手などを制御することができる。

6.1.3 強い使い分け

この事例では、様々な背景による文脈が存在し、それぞれの文脈ごとに、別個の存在として活動を行っている。また、別個の存在であるそれぞれが、同一人物によるものであることは知られてはならない。これを満たすためには、それぞれの文脈ごとに複数の異なる人格を使い分ける。

6.1.4 強い使い分け +

この事例では、前節の事例に加え、近しい、つまり知人に対しては複数の人格が同一人物であることが伝わらなければならない。本研究で提案するモデルでは、複数の人格を使い分けた上で、人格間リンクを利用することで、知人にのみ複数の人格が関連付いて示される。

6.1.5 まとめ

以上の事例を表??にまとめた。

TODO: ここで表を出す。

このように、要求分析で提示した事例は本モデルにより全て満たすことができることが確認された。

6.2 モデルの考察

本節では、事例から抽出した本モデルの要素ごとに考察を加える。

6.2.1 複数の人格の分離

本モデルでは、各利用者ではなく個別の人格がシステム内における主キーとなる識別子を持ち、知人ネットワークに参加する構成要素となる。知人関係は、この人格ごとの識別子を端点として結ばれる。これにより、既存の SNS が持つ知人関係の表示・操作や、知人関係に基づくアクセス制御の単位を個別の人格毎に分離することが実現された。後述するように、明示的に開示しない限りはシステム上で全くの別個の存在となるため、当人の意図しない情報の漏洩は起きない。

6.2.2 一方向的な同一人物関係の開示

特殊な知人関係により実現される同一人物関係の設定は、一方向的・包含的に別の側面を見せることを実現した。複数の人格を使い分けては居ても、親しいつきあいをしている相手には様々な側面を見せるという需要を満たしている。

6.2.3 既存の SNS モデルとの親和性

これまでの SNS モデルとの差異は、知人ネットワークの構成要素を利用者から人格に置き換えたことによる知人ネットワークによるアクセス制御の変更と、インターフェース的な変更の二点である。それぞれ、実装コストと利用コストに相当する。

知人ネットワークによるアクセス制御の変更 人格を単位とする知人ネットワークでは、利用者はネットワーク内に分散した複数の人格を所有する。知人関係に基づくアクセス制御において目的の相手の人格との知人ネットワーク上の距離を算出する場合、自らの持つ複数の人格のうち、最も近いものを選択しなければならない。

本モデルでは、各人格が利用者 ID を属性値として保持し、相手との関係を演算する全ての処理において自らの利用者 ID から複数の人格にマッピングし、もっとも距離の近いものを選択する機能を付加した。この処理は、RDBMS 上にて適切な索引を用意することで $O(n)$ の計算量で実現でき、このモデルの採用を著しく困難なものとする範囲ではない。

インターフェースの変更 利用者の視点における最大の変化は、複数の人格を管理しなければならないことである。そのため、自らの人格に設定するプロフィールや知人関係が所有する人格の数だけ多重化するだけでなく、さらには日記や掲示板への書き込み毎にどの人格による行動なのかを選択しなければならない。

しかし、相手との関係を元に見せ方を変えるという本研究の目的を必要とする利用者にとっては、何か行動を行う際にその影響範囲を考えることは自然なことである。例えば掲示板やその掲示板が属するグループ単位でそれに対応する人格をあらかじめ指定するなどの支援を行うことで、大きな操作コストの増加を避けることが可能である。また、複数の人格を持たない利用者は人格の選択機能を不可視にすることで、これまで通りの操作体系を変える必要がない。よって、このモデルを採用することによるデメリットは無いと言える。

6.3 他方式との比較

本節では、他の方式との比較を行う。

6.3.1 他の SNS

6.3.2 UNIX group

UNIX において利用されている、ユーザおよびグループによるアクセス制御と比較する。

- 複数 ID

6.3.3 アクセス制御リスト

本来、人間は知人一人ごとに自分とどんな関係かを考え、それに基づいてコミュニケーションを行っている。これはもっともきめ細かい制御を必要とするモデル

で、情報それぞれに見せる知人、見せない知人を一つずつ指定するアクセス制御リストとみなすことができる。

- 設定コスト

6.4 運用実験

第5章にて述べた実装を用いて実験運用を行った。本節ではその結果を示し、考察を行う。

6.4.1 実験概要

実験目的 本実験の目的は、大きく二点に分けることができる。

一点目は、本研究で提案するモデルが実装可能なものであるとともに、その使い勝手の変化について確認することである。

二点目は、人格を使い分けるモデルを採用する SNS を運用し実際の利用者に触れてもらうことにより、社会的に本研究が要求され意義が有るものであることを示すことである。

実験の参加者 今回の実験では、モデルの有意性を明らかにするという目的から多層的な知人関係を再現することが望ましく、単純に参加者を増やすだけでは実験の意義が薄れてしまう。そこで、以下の二点を前提に実験の参加者を募った。

- 知人ネットワークが著しく分断することを防ぐため、著者を起点とする招待制を採用した。
- 知人ネットワークの広がりを重視するため、著者の知人を媒介に、著者と直接の知人を持たない利用者を増やした。

具体的には、以下の手順で参加者を応募した。

システムにはメールアドレスを指定して実験への参加依頼を送信する機能を実装した。この機能を利用して実験に参加してもらうことを本実験において招待と呼ぶことにする。

1. システムに初期から存在する管理者ユーザから、著者自身を招待した。
2. 著者は、二つの人格を用意し、既存の知人関係に基づいてそれぞれ5人程度を招待した。
3. 実験の参加者となった被験者に対し、さらにその知人関係をもとに別の被験者を募り、招待を行った。

6.4.2 実験結果

実験状況 本実験はその性質から継続的に運用を続けており、本章にて示す数値はあくまで執筆時点での数値である。

6.4.3 アンケート調査

利用者を対象として、アンケート調査を行った。アンケート調査の結果については付録 A に載せる。

第7章 結論

7.1 本研究の成果

本研究では、人間がコミュニケーションにおいて必要とする知人関係の抽象化において、様々な要求を満たすモデルを提案し、そのモデルに基づいた実装による運用実験を行った。

さらに第6章にて、本研究が提示するモデルは人間が要求する様々なケースにおいてそれを実現することができることを示した。また、これを実現するコストについても十分に実現可能かつ規模性があることを示した。さらに、運用実験により、実環境において有用であることを示した。

7.2 今後の課題

7.2.1 分散化

本モデルではRDBMSのように知人ネットワーク全体を一つのデータソースに収めることを前提としている。だが、言うまでもなく知人関係は特定のサービスに所属するものではなく、個々人がそれぞれ持つものであり、集中管理を必要としない分散型のモデルを提案していく必要があるだろう。

分散化を実現する方向性としては、既に Affelio??をはじめとした分散的に知人ネットワークを形成する枠組みは存在するが、一つの利用者が複数の人格を持つ、と言う部分をそのままこの枠組みに組み込むことは困難である。本研究では、サービス運営者が利用者と複数の人格との関係を隠匿しているが、複数の人格感の関係を分散的に表現する部分が最大の問題点となるだろう。

7.2.2 細かい制御

本モデルでは、本来は相手ごとに異なる様々な背景や情報制御の粒度を文脈という概念を用いて抽象化した。しかしながら、実際には知人一人ごとの制御が必要になる場合も考えられる。たとえば、知人の誕生日のサプライズイベントについての情報は、その知人を除いた相手に届けたいというような事例が考えられる。しかしながら、知人ごとに制御を行うためには、自分が知識として持っている知

人関係を細かくシステムに設定しなければならず、コストが大きくなることから本研究ではそのような事例を対象外とし、あくまで一定の粒度としてまとめる手法を採用している。

知人ごとの細かい制御を単にシステムに実装するだけでなく、何らかの演算などにより相手を柔軟にグルーピングすることができれば、前述のような事例においても容易に利用できるだろう。

7.3 発展

本研究では、関係性を持つ複数の人格を使い分けるモデルを提示し、その有意性を示した。有意性が明らかになったことにより、今後、既存の SNS が本モデルを採用することが考えられる。

これまで多層的な知人関係を一つの平坦な関係に合わされてしまうことで、自分の振るまいが極端に制限されるという状況が起きていたが、本研究のモデルを採用する SNS を利用することで、そのような状況から避けることができる。その結果、SNS 上でのコミュニケーションがさらに活発なものとなり、最終的には文化の発展に寄与することができる。

謝辭

参考文献

- [1] 児童の権利に関する条約. <http://www.mofa.go.jp/mofaj/gaiko/jido/zenbun.html>.
- [2] 須子喜彦. 知人関係を用いたプライバシー保護型マッチングシステムの研究.
- [3] ソーシャル・ネットワーキングサイト [mixi(ミクシィ)]. <http://mixi.jp/>.
- [4] Japan.internet.com デイリーリサーチ - sns で人気があるのは「mixi」「livedoor アミーゴ」. <http://japan.internet.com/research/20050629/1.html>.
- [5] ソーシャルネットワーキングの利用に関する調査.
<http://www8.plala.or.jp/revir/works/sns/>.
- [6] グリー (gree) : ソーシャルネットワーキングサービス. <http://gree.jp/>.

付録A アンケート集計結果

あなた自身について質問します。

Q101 あなたはソーシャルネットワーキングサイト (SNS) と呼ばれるサービスをこれまで利用したことがありますか。

ある	9人
ない	0人
無回答	0人

Q102 「SNS を利用したことがある」とお答えした方に質問します。SNS をどれくらいの頻度で利用していますか。「利用」には閲覧のみも含まれます。

毎日	9人
3、4日おきに1回程度	0人
1週間に1回程度	0人
それよりも少ない	0人
無回答	0人

Q103 「SNS を利用したことがある」とお答えした方に質問します。主に利用している SNS には実名で参加していますか。

実名で参加している。	5人
よく知られているハンドルネームで参加している。	3人
上記のどちらでもない。	1人
無回答	0人

Q104 「ハンドルネームで参加」もしくは「どちらでもない」とお答えした方に質問します。実名を使わない理由は何故ですか。

実名をインターネット上で名乗るのは危険だと考えている。	1人
実名によって検索されたくない。	1人
SNS内の知人に、実世界での所属・活動等を知られたくない。	0人
ハンドルネームの方がよく知られているから。	1人
無回答	6人

Persoの特徴である「複数の人格」という形について

Q201 状況や相手によって、異なる自分を使い分けたいと思いますか。

どうしても使い分けたい。	2人
使い分けられるならば使い分ける場合もある。	7人
自分には使い分ける必要はない。	0人
無回答	0人

Q202 Perso上で複数の人格を設定しましたか。

2つ以上の人格を設定した	6人
1つの人格だけを利用した	3人
質問の意味が分からない	0人
無回答	0人

Q203 「2つ以上の人格を設定した」と答えた方に質問します。日記を書く度にどの人格で書くのかを指定するのは手間と感じますか。

手間と感じる。	1人
手間とは感じない。	4人
無回答	4人

Q204 全く別の自分を独立した人格として扱う Perso の形についてどう思いますか。

(自由記入項目のため割愛)

Q205 今後、この複数の人格を使い分けることができる Perso のような枠組みが、他の SNS においても提供された方が良いと思いますか。

はい	6人
いいえ	3人
無回答	0人