

修士論文 2005年度 (平成17年度)

多層的な知人関係に基づく自己情報コントロールの実現

慶應義塾大学大学院 政策・メディア研究科

---

## 多層的な知人関係に基づく自己情報コントロールの実現

本研究では、知人関係などにより人間が持つ文脈ごとに、知人ネットワーク内でも文脈を切り替えられるモデルとして、複数人格モデルを提案する。

このモデルでは文脈ごとに別の行動主体となる人格を使い分けることで、文脈ごとに自分の見え方を制御し、秘密を守ることができる。この複数の文脈は個別独立に存在するものではなく、利用者からは多層的な一つの軸として扱えなくてはならない。このモデルにより、利用者は多層的な複数の文脈を自由に行き来したり、自分の見え方の制御に関して、自らの複数の人格が同じ自分であることを限られた人にだけ明かしたりといった自己情報コントロールが可能となる。

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キングサイト (SNS) を構築し、以上のような社会的要求を複数人格モデルが満たすことを確認した。また、規模性の面でこれまでの知人ネットワークに劣ることなく実現できることを明らかにした。

### 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

慶應義塾大学大学院 政策・メディア研究科

仲山 昌宏

<p>The Research on the Personal Information Control based on Multiple Social Network</p>
--

How to communicate about myself is very important problem. We have something to tell someone or secret something not to tell someone. Such personal information control we have been is also important in online communication.

There is the theory called 'Social Network' that archive the personal information control. In this theory, friendship of human is assumed a link of graph theory. A lot of applications to communication appears, but they can not reflect multiple social network we have. In result, our communication is inhibited.

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. The model is instantiated some case example supposed, then it can archives them. 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 Governamce

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	5
1.3	用語等の定義	5
1.4	本論文の構成	6
第2章	要求分析	7
2.1	ユースケース分析	7
2.1.1	単純なユースケース	7
2.1.2	弱い使い分け	8
2.1.3	強い使い分け	8
2.1.4	限定的な強い使い分け	8
2.2	要件定義	9
2.2.1	内容	9
2.2.2	公開範囲	9
2.3	既存の SNS についての考察	10
2.3.1	mixi	10
2.3.2	imeem	13
2.3.3	livedoor フレパ	15
2.3.4	wyco	17
2.3.5	複数の利用者の使い分け	20
2.3.6	考察	21
第3章	設計	22
3.1	複数人格モデルの概要	22
3.2	複数人格モデルの詳細	24
3.2.1	人格と文脈	24
3.2.2	自ら発信する情報	24
3.2.3	他者との関係に関する情報	25

3.2.4	活動の行為者	25
3.2.5	同一人物リンク	25
3.3	まとめ	26
<b>第4章</b>	<b>先行実験</b>	<b>27</b>
4.1	先行実験の概要	27
4.1.1	先行実験の目的	27
4.1.2	実験の手順	27
4.2	実験結果	28
4.2.1	利用状況	28
4.2.2	知人ネットワークの分析	28
4.2.3	実装要件の確認	30
4.2.4	アンケート調査	31
<b>第5章</b>	<b>アプリケーションの設計と実装</b>	<b>33</b>
5.1	実装の方針	33
5.2	実装環境	33
5.3	全体の概要	34
5.4	サイト構成	40
5.4.1	トップページ(ログインフォーム)	42
5.4.2	マイページ	43
5.4.3	人格ごとのページ	45
5.4.4	知人リンクの希望	46
5.4.5	プロフィールの編集	48
5.4.6	日記の登録	49
5.5	各モジュールの解説	49
5.5.1	利用者の管理	49
5.5.2	人格の管理	50
<b>第6章</b>	<b>評価</b>	<b>51</b>
6.1	必要性	51
6.2	要件	51
6.2.1	複数の文脈の使い分け	51
6.2.2	複数の文脈の使い分け	51
6.2.3	管理可能性	52
6.3	その他の要件の詳細	52
6.3.1	内容	52
6.3.2	公開範囲	52
6.4	ユースケースとの比較	53
6.5	モデルの考察	54

6.5.1	既存の SNS モデルとの親和性 . . . . .	54
6.6	他方式との比較 . . . . .	55
6.6.1	UNIX group . . . . .	55
6.6.2	アクセス制御リスト . . . . .	55
6.7	モデルに対する評価のまとめ . . . . .	56
6.8	実験 . . . . .	56
6.8.1	実験目的 . . . . .	56
6.8.2	結果 . . . . .	56
6.8.3	規模性 . . . . .	57
6.9	今後の課題 . . . . .	60
6.10	まとめ . . . . .	60
<b>第 7 章</b>	<b>結論</b>	<b>61</b>
7.1	本研究の成果 . . . . .	61
7.2	今後の課題 . . . . .	61
7.2.1	分散化 . . . . .	61
7.2.2	細かい制御 . . . . .	62
7.3	発展 . . . . .	62
	謝辞	<b>63</b>
	参考文献	<b>65</b>
	付 録 A アンケート集計結果	<b>66</b>

# 目次

1.1	知人ネットワークに基づく情報の配布	3
2.1	mixi ホームページ	11
2.2	知人関係の距離による制御	12
2.3	知人関係の可視性	14
2.4	知人関係の分類による制御	15
2.5	livedoor フレパ 公開範囲の設定	16
2.6	wyco プロフィール編集画面	18
2.7	wyco 名刺フォルダの管理画面	19
2.8	複数の「利用者」の使い分け	20
3.1	複数人格モデルの概要	22
3.2	同一人物リンク	23
3.3	利用者と複数の人格	24
5.1	システム全体のモジュール構成	35
5.2	各テーブル間の関係	39
5.3	Perso ウェブサイト構成 (非ログインユーザ)	40
5.4	Perso ウェブサイト構成 (ログインユーザ)	41
5.5	ログインフォーム	42
5.6	利用者のマイページ	44
5.7	人格ごとのプロフィール	45
5.8	知人リンクの要求	46
5.9	知人リンクの承認・拒否画面	47
5.10	プロフィールの編集	48
5.11	日記の登録画面	49

# 表 目 次

2.1	事例の比較 . . . . .	21
4.1	運用実験の利用状況 . . . . .	28
4.2	所有する人格の数による度数分布 . . . . .	28
4.3	同一人物関係の設定状況 . . . . .	29
5.1	実装環境 . . . . .	33
5.2	Catalyst Plugins . . . . .	36
5.3	テーブル一覧 . . . . .	38
6.1	ユースケースのまとめ . . . . .	53



# 第1章 序論

本章ではまず、人間のコミュニケーションが多層的であ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、多層的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、相手、内容、手段の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<sup>[1]</sup>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。つまり、秘匿性を要求するようなコミュニケーションの手段としては利用できない。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。

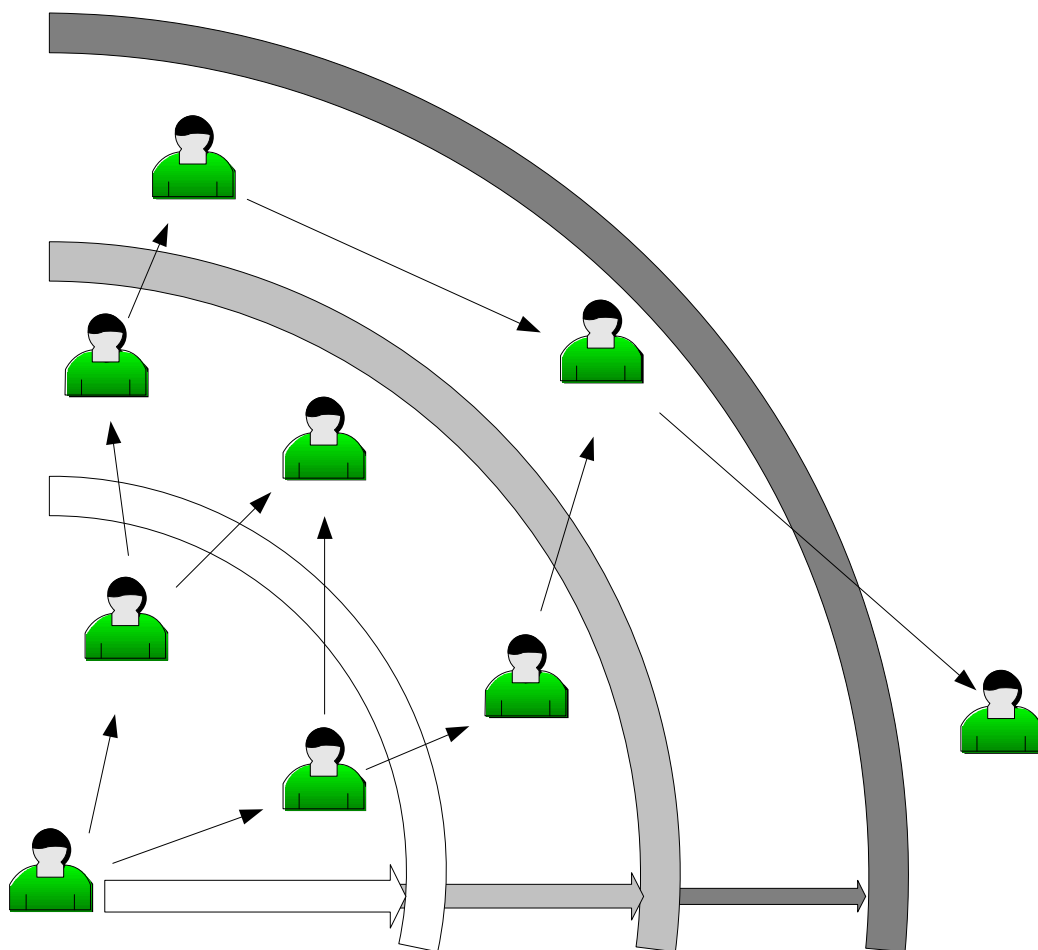


図 1.1: 知人ネットワークに基づく情報の配布

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。[2] 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もとに、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、SNS 内における一つの識別子によって表される存在として集約されてしまう。[3] その結果、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。

2005 年に実際の知人ネットワークのデータを用いた研究発表がされた社会情報学フェア 2005[4] においても、このように指摘されている。[5]

国立情報学研究所の大向一輝氏は「個人の“多重人格性”を保持しながら、サービスをどう進めるかが課題」と指摘する。ユーザーは現実社会で複数のコミュニティに所属しており、コミュニティごとにさまざまな“顔”を使い分けていることが多い。mixi は、すべてのマイミクに対して同じ顔しか見せられないため、マイミクが増えるにつれ息苦しくなってやめてしまうユーザーも少なくない。

これに対し、用途に応じた複数の SNS を利用するという方法が徐々に広まりつつある。特に、知人ネットワークによるウェブアプリケーションである SNS をオープンソースで実装している OpenPNE[6] の公開により、個人でもまかなえる範囲で SNS を運用することができるようになった。これを受けて、特定のグループや、趣味つながりの輪ごとの、たくさんの小さな SNS が設置されるようになった。[7][8] ま

た、そういった対象を限定するような SNS の中で 1 万人を超える規模のもの [9][10] も出てきている。

これは、SNS が基盤としている既存の知人ネットワークのモデルでは守りたい自分の秘密が守れない、それを運用で回避しているにすぎない。これは、知人ネットワークが広がっているべき世界が、いくつにも分断されてしまっているということである。これまでの SNS においてコミュニケーションを阻害するような問題が解決されることで、オンライン上に展開された知人ネットワークはその可能性を大きく高めるだろう。

## 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。特に、伝えるべき情報を相手に正しく伝えるというだけでなく、ある秘密の情報が伝わってほしくない相手に対しては届くことがないことが、個人情報コントロールの観点から重要とされている。また、情報が伝わってほしくない相手は悪意を持つ者だけでなく、知人だからこそ伝えたくないという場合も存在する。

すなわち、人間の知人関係には、その背景となる関係性により複数の文脈が存在する。これは例えば会社や大学の繋がりであったり、地元の幼なじみであったり、あるいはオンライン上で知り合った趣味を同じくする繋がりなどである。このような複数の文脈が複数存在するために、その知人には伝えたい、伝えたくないといった情報コントロールが必要とされる。

すでに述べたように、現在の知人ネットワークでは一人の利用者は一つの文脈しか持つことができず、複数の文脈であるはずの知人関係をまとめて扱っているために秘密にしたい情報は結局伝えることができていない。

複数の文脈を知人ネットワークの枠組みに組み込み、それらを全て利用者が管理可能であるようなモデルを提案することが本研究の目的である。

このようなモデルを実現するために、複数の文脈を分離することで秘密を守り、それらの文脈が十分に管理可能であるように知人ネットワークモデルを再定義し、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

**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** システムを利用する一人の人間を指す。

人格 本研究が提案する複数人格モデルにおける、行動主体の単位を指す。一人の利用者は、一つ以上の人格を持つ。

主人公 特に個別のユースケースにおいて、主役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もしくは人格を指す。

閲覧者 主人公から情報を得ようとする利用者を指す。

知人ネットワーク 知人関係によって結ばれた利用者もしくは人格により構成されるグラフ構造を指す。

公開範囲 知人ネットワーク上における情報の到達範囲を指す。

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を指し、意図しない第三者から、それらが同じ利用者のものであると知ることができない。

## 1.4 本論文の構成

本論文は7章から構成される。

第2章では要求分析を行う。さらに既存のユースケースについて考察を行い、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する。

これらの要件を元に、第3章において「複数人格モデル」について提案を行う。

第4章では、複数人格モデルの方向性を検証するために実施した先行実験について紹介する。第5章では、複数人格モデル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、第2章にて定義された要件を元に複数人格モデルが目的を実現していることを確認する。

最後に、第7章において本研究の総括と、複数人格モデルに関する今後の展望について述べる。

## 第2章 要求分析

本研究の目的は、社会的な要求を満たす知人関係モデルを実現することにより、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である。言い換えると、様々な人間が求めるコミュニケーションを実現することができる知人関係のあり方をモデル化することによって達成することができる。

そもそも、人間が求めるコミュニケーションのありかたとは、伝えたい相手に情報が伝わり、伝わらないべき相手には情報が伝わらないことである。繰り返しになるが、コミュニケーションの要素とは相手、内容、手段の三つであり、相手と内容によって手段が限定される。この相手と内容の組み合わせが多様であることから、コミュニケーションに様々な形態が発生する。

本研究では特に、「特定の相手だけに自分の情報を伝える」と同義となる「特定の相手以外には、その自分の情報が伝わらない」という点の実現を重視している。つまり本研究においては、どうやって「特定の相手」に限定するか、もしくは「どんな情報」の伝わりを限定したいかという二点が重要となる。

そこで本章では、いくつかの具体的なユースケースを取り上げることで要件を明らかにし、この要件に基づいて、既存のユースケースに対し考察を加える。

### 2.1 ユースケース分析

本節では、具体的なユースケースを検討することにより、実装すべき要件を洗い出す。

#### 2.1.1 単純なユースケース

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。

このユースケースでは、自分が発信する情報に対して、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。

### 2.1.2 弱い使い分け

次に、異なる背景を持つ複数の知人関係があり、相手との関係によってコミュニケーションの内容を変えたい場合について取り上げる。

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、自分の知人に対して、分類に基づく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。

### 2.1.3 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。自分から発信するプロフィール情報や日記等だけでなく、個人を特定できるような他者との関係である、知人関係やグループへの参加状況なども複数の主体として分離することになる。

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を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。

### 2.1.4 限定的な強い使い分け

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变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。

## 2.2 要件定義

次に、これらのユースケースからコミュニケーションの要素を抽出し、細かい要件を定義する。

### 2.2.1 内容

まずコミュニケーションの内容を定義する。

**自ら発信する情報** 利用者が独立して発信する情報で、利用者自身の名前、住所といったプロフィール情報や、日記・Weblog等の記事などが含まれる。

主に、利用者が他者に何らかの情報を示すために用いられる。

**他者との関係に関する情報** 利用者とは他者との関係を示す情報で、最も重要なものとして知人の一覧がある。また、システム内に提供されるような、テーマ毎のグループ等も含まれる。

利用者がどんな人間であるかを他者に示すだけでなく、知人関係やグループへの所属により利用者自身が情報を受ける立場となるという点で、自ら発信する情報とは影響力が異なる。

### 2.2.2 公開範囲

これらの情報について、誰に公開すべきかという制御を行う。本研究では、公開範囲の制御方法を示す指標を以下のように定義し、これら全てを実現することが必要とされる。この指標では「主人公」の情報を、「どのような閲覧者」が知ることができるかどうかという表記を行う。

**公開** あらゆる閲覧者が主人公の情報を知ることができる。

本研究では、閲覧者はシステムの利用者を前提とし、システムの利用者がたとえ他の利用者による招待に限定されている等については考慮しない。

知人関係の距離による制御 知人関係の有無を一般化したもので、知人ネットワークのグラフ構造において、閲覧者と主人公との距離が決められた範囲であれば情報を知ることができる。

特に、距離が1である場合には閲覧者が主人公の知人であるかどうか、すなわち「知人関係の有無」と言い換えることができる。

知人関係の分類による制御 閲覧者が知人であり、かつ主人公が定めた分類により許可されている場合にのみ、その情報を知ることができる。

知人関係の分類による強い制御 主人公が複数の識別子によって識別され、閲覧者が利用した識別子ごとに異なる情報を知ることができる。

知人関係の分類および距離による強い制御 同様に識別子毎に異なる情報を知ることができる。

さらに、閲覧者が主人公の知人であれば、別の識別子に結びつけられた情報も知ることができる。

## 2.3 既存の SNS についての考察

次に、知人ネットワークモデルに基づくコミュニケーションサイトである SNS(ソーシャルネットワーキングサイト) について紹介する。

特徴のある SNS として、もっとも標準的な基本機能のみを持つ mixi、知人関係そのものの可視性を設定できる imeem、知人のカテゴライズを可能とする Livedoor フレパおよび wyco の 4 種類を取り上げる。

### 2.3.1 mixi

日本国内においてもっとも利用者数の多い、一般的な SNS の例として mixi[11] を取り上げる。mixi のホームページ画面を図 2.1 に示す。

mixi では、以下のような自分が公開する情報に対して知人関係に基づいたアクセス制限を掛けることができる。

- 現住所
- 誕生日
- 生まれた年
- 出身地



図 2.1: mixi ホームページ

- 職業
- 所属
- 日記
- フォトアルバム

これらは既に述べた「自ら発信する情報」に含まれる。

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、入力が必須の誕生日、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、友人までに公開という「知人関係の距離による制御」だけでなく、「閲覧キー」と呼ばれるパスワードによるアクセス制御を選ぶことができる。これは、知人関係に限らない公開範囲の設定が求められる場合を想定し、それらを利用者の運用方法によって対応してもらう事が必要だと考えられたためであろう。

本研究における分類を利用すると、「自ら発信する情報」に対して「知人関係の距離による制御」を実現している。この概念図を図 2.2 に示す。

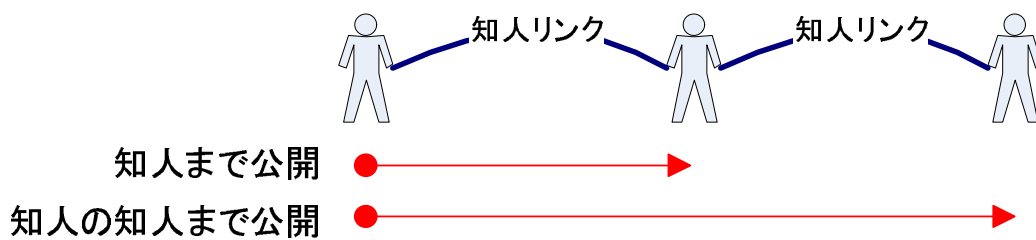


図 2.2: 知人関係の距離による制御

### 2.3.2 imeem

imeem[12] 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。利用者は、コンピュータ上で動くクライアントからアクセスする。

imeem の特徴として、プロフィール情報や日記だけでなく、知人関係やグループへの参加状況などについても、公開範囲を指定することができる点があげられる。知人関係の可視性設定画面を図 2.3 に示す。本研究における分類では、全ての情報に対して「知人関係の距離による制御」を実現している。

また、imeem クライアントだけでなく、imeem 提供者によるウェブサイトからも imeem 内の情報を閲覧することができ、「全ての imeem 利用者」に加え「The world (web publishable)」という公開範囲が存在する。これは本研究とは関係ないが、必要以上に SNS 内に情報を閉じこめないという点で評価できる。

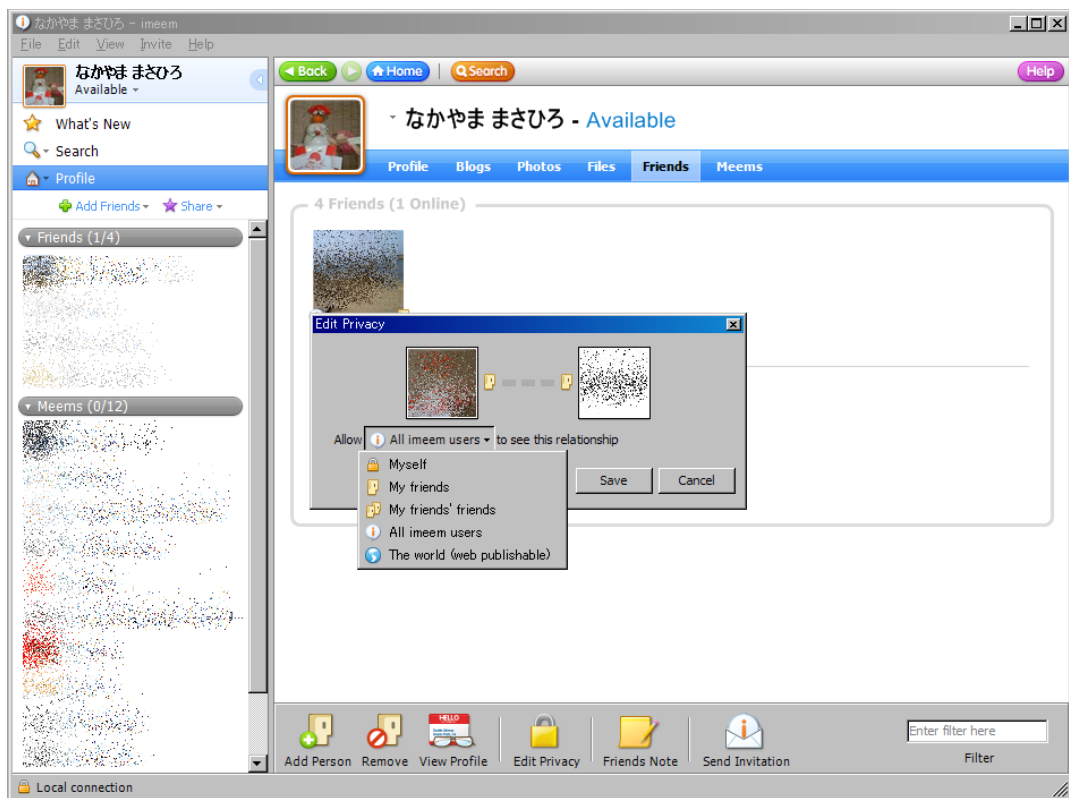


図 2.3: 知人関係の可視性

### 2.3.3 livedoor フレバ

livedoor フレバ [13] も mixi と同様に一般向けの SNS だが、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 フレバで特徴的な点は、図 2.5 で示すような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おり、実用上大きな問題となるだろう。

livedoor フレバ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。

本研究における分類を利用すると、「自ら発信する情報」に対して「知人関係の分類による制御」を実現している。この「知人関係の分類による制御」の概念を図 2.4 に示す。知人関係の分類を背景の円に表し、自ら発信する情報に対してそれぞれの情報の到達範囲を指定することができる。

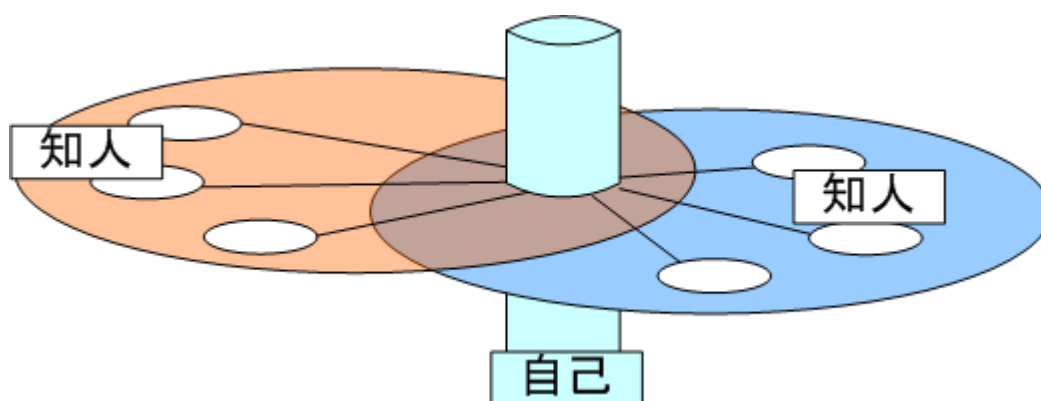


図 2.4: 知人関係の分類による制御

マイページ 新着日記 メンバー検索 コミュニティ 友達を誘う

トップ メッセージ 日記 レビュー 足あと プロフィール 設定変更

● プロフィールの修正

あなたのプロフィールを変更したい場合は、こちらのフォームで変更してください。  
公開設定を行うと、自分のフレ友や特定のグループだけに情報を表示することができます。

livedoor ID:

名前: 姓:  名:  [\* 全体に公開] 公開設定

ニックネーム:  \* 全角16文字まで

Skype名:  [\* 全体に公開] 公開設定

住所:   \* 都道府県のみ登録したい場合は、都道府県のまにしてください。

性別:  男性 /  女性

生年月日: --年--月--日

血液型: --

職業:  [\* 全体に公開] 公開設定

趣味:  映画鑑賞  DVD鑑賞  スポーツ  音楽鑑賞  ライブ  カラオケ・バンド  グルメ  お酒  宴会  ショッピング  ファッション  アウトドア  ドライブ  旅行  バックパッカー  アート・芸術  語学  読書  マンガ  創作  テレビ  ゲーム  インターネット  キャンブル  ペット  美容・ダイエット

自己紹介:  [\* 全体に公開] 公開設定 \* 全角400文字まで \* タグは無効です。URLは自動的にリンクされます

好きな:

● 公開設定

「名前: なかやま まさひろ」についての公開設定です。

公開設定を行うと、自分のフレ友や、特定のグループだけに情報を表示することができます。

全体に公開

フレ友全員に公開

グループで指定

設定するグループを選択してください。

リアル友人(2)

大学・研究室絡み(0)

P2P界限(0)

P2P界限(4)

設定してください(4)(0)

設定してください(5)(0)

設定してください(6)(0)

設定してください(7)(0)

設定してください(8)(0)

設定してください(9)(0)

公開設定変更

[このウィンドウを閉じる](#)

ページが表示されました インターネット

図 2.5: livedoor フレパ 公開範囲の設定



### 2.3.4 wyco

wyco[14] は株式会社毎日コミュニケーションズが運営する SNS で、知人関係を「名刺の交換」、複数的人格の使い分けを三種の「名刺」として表現している。

知人関係を「友達」「同じ会社」「仕事関係」の三種に分類し、この分類に基づいて細かい使い分けを実現としている。「相手に見せる名刺」をこの三種から選択することで、情報の可視範囲を制御する。図 2.6 に示すプロフィール編集画面では、各項目に公開範囲指定として、分類されたどの知人に見せるかというチェックボックスがある。

また、「知人関係」に対して「知人関係の分類による制御」を実現しているのが、図 2.7 に示す名刺フォルダの管理である。「相手にどの名刺を見せたか」によって分類をしている。

以上を本研究に基づいて言い換えると、全ての情報に対して「知人関係の分類による制御」を実現していると言える。

wyco  
wycocom

マイページ | ダイアリー | コミュニティ | フックマーク | Wycoメンバー検索 | Wycoに招待 | ヘルプ | ログアウト

wyco  
ソーシャルネットワークワーキングサービスWycoでは、登録したプロフィールを公開して、あなたのお知り合いを探し、wycoにお誘いください。ご協力をお願いします。

config

あなたの設定

プロフィール | Wycoプロフィール | アドレス設定 | 名前 | 設定

名前	姓 山田 (20文字)*	
	名 山田 (20文字)*	
ニックネーム	山田 (30文字)*	
生まれた年	1999 年 (*)	<input type="radio"/> 全公開 <input type="radio"/> 非公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
誕生日	7 月 17 日 (*)	<input type="radio"/> 全公開 <input type="radio"/> 非公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
性別	<input checked="" type="radio"/> 男性 <input type="radio"/> 女性 <input type="radio"/> 秘密 (*)	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
血液型	O型	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input checked="" type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
会社名	慶徳義塾大学 (50文字)	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
業種	研究者・ネットワークエンジニア (30文字)*	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
職階		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
職歴		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
具体的な仕事内容		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
こんな人と仕事がしたい		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
ワークスタイル	<input type="radio"/> 正社員 <input type="radio"/> 派遣社員 <input type="radio"/> 契約社員 <input type="radio"/> フリーランス/自営 <input type="radio"/> アルバイト <input type="radio"/> 会社経営/役員 <input type="radio"/> 学生 <input type="radio"/> 雇用なし	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
入社年		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
現住所(経度経緯)	東京都 (*) ※ 海外の場合は「その他」を選択し、国名を記入してください。	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
現住所(市町村)	千代田区 (20文字) 例) 中央区	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
出身地(経度経緯)	東京都 (*) ※ 海外の場合は「その他」を選択し、国名を記入してください。	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
出身地(市町村)	千代田区 (20文字) 例) 千代田区	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
大学名	慶徳義塾大学 (30文字)	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
学部+学科名	経営・メディア研究科 (50文字)	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
サークル		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
ゼミ	千代田研究室 (30文字)	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
大学入学年	1999 年	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
就職者の有無		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
就活事情		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
持っている資格		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
休日の過ごし方		<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係
自己紹介	研究室ではSNSの研究やっています。 大学のネットワーク専攻の運用とかもやっています。 様々な知人関係がごちひになる環境は、人によって まよってと誤解な部分もありますよね。複数の「名 刺」を持つことができるwycoに期待をしながら進ん でいきます。 そんな感じでよろしくです。	<input type="radio"/> 全公開 <input type="radio"/> 公開範囲指定 <input type="checkbox"/> 友達 <input type="checkbox"/> 同じ会社 <input type="checkbox"/> 仕事関係

SSLで送信する

wycoは | 株式会社wyco | 個人情報取扱いについて | 利用規約 | ヘルプ | お問い合わせ  
copyright©2005 Mainichi Communications Inc. All rights reserved.

図 2.6: wyco プロフィール編集画面

all すべての名刺	friends 友達の名刺	company 同じ会社の名刺	business 仕事関係の名刺
---------------	------------------	--------------------	---------------------

<友達の名刺> フォルダの公開範囲設定

wyco全体に公開  
 名刺フォルダのメンバーにのみ公開

[→ 設定する](#)

 あなたからの紹介文	まだ記入されていません
	まだ記入されていません
	friends

[この名刺を編集する](#)

[← 一番最初へ](#)
[← 前の20件](#)
[次の20件 →](#)
[一番最後へ →](#)

[名刺フォルダを見る](#)

設定メニュー

- [写真の設定](#)
- [プロフィールの設定](#)
- [ダイアリーの設定](#)
- [パスワードの設定](#)
- [メールアドレスの設定](#)
- [名刺フォルダの設定](#)
- [ステータスの詳細設定](#)
- [コミュニティの設定](#)
- [退会](#)

毎日フレッシュアップ 新着コンテンツ

- ・色々情報追加！卒業旅行特集
- ・今から始めて、入社後ロードパワーポイント入門



図 2.7: wyco 名刺フォルダの管理画面

### 2.3.5 複数の利用者の使い分け

既存の各 SNS において複数の「利用者」として振る舞うことにより、全ての情報に対して「知人関係の分類による強い制御」を実現することができる。この場合、それぞれの「利用者」はシステム内で全く別の存在として表現されるため、自分の情報や、知人の分類に応じて、知人関係を設定する「利用者」を使い分けることができる。この複数の「利用者」による「知人関係の分類による強い制御」の概念図を、図 2.8 として示す。

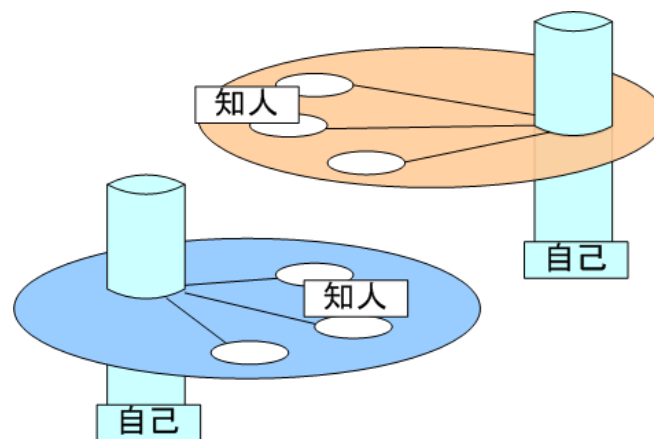


図 2.8: 複数の「利用者」の使い分け

この問題点は、本来の利用者が複数の別れた「利用者」を使わなければならないため、利用者からみると操作が冗長なものとなる。具体的には、「知人が新しく書いた日記」が二つの異なるリストとして存在し、システムに別の「利用者」としてログインし直さなければ両者を見るできないなどの例を挙げることができる。日常的なコミュニケーションツールとして、この複数の「利用者」としてログイン・ログアウトを繰り返さなければならないというコストはきわめて大きく、複数の利用者を使い分けることによるメリットを十分に阻害している。つまり、本研究ではシステム内のモデルとして「知人関係の分類による強い制御」を実現するが、この点はシングルサインオンの利便性の面において効果大きい。

また、本研究の本題ではないが、既存の利用者による招待に新規登録を限定している SNS においては、一人の利用者による複数の利用者登録を認めていない場合もあることも述べておく。

### 2.3.6 考察

これまでに紹介した5つの事例を表にまとめたのが表 2.1 である。参考までに、「自ら発信する情報」には従来の WWW による情報公開について加えた。

表 2.1: 事例の比較

公開範囲	自ら発信する情報	他者との関係に関する情報
公開	WWW	mixi livedoor フレバ
知人関係の距離による制御	mixi, imeem	imeem
知人関係の分類による制御	wyco livedoor フレバ	wyco
知人関係の分類による強い制御	複数利用者	複数利用者
知人関係の分類および距離による強い制御	本研究	本研究

このように、本研究が必要とする「知人関係の分類および距離による強い制御」を実現できているものは無く、より緩い「知人関係の分類による強い制御」についても実現できているとは言い難い。

期待されている公開範囲の制限ができないということは、人間が期待した相手だけに限定して秘密の情報を伝えることができないということである。そのため、1.1.3 節で述べたような「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」という事例に代表されるようなコミュニケーションの阻害が起きてしまっている。

本研究により、モデルの不備によるコミュニケーションの抑制が解決されることが、既存の事例との比較から明らかになった。

# 第3章 設計

## 3.1 複数人格モデルの概要

前章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。

この複数人格モデルのおおまかな概要を図3.1に示す。中央にある柱が自分そのものである自己を示し、柱から伸びている線と、線につながれた小さな円が知人関係および知人を示す。

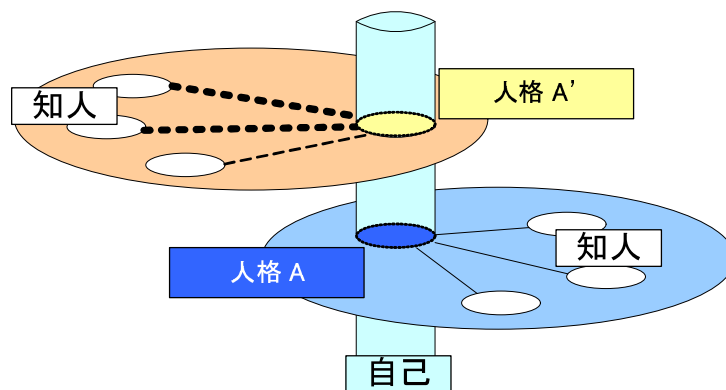


図 3.1: 複数人格モデルの概要

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。概念図では、点線と実線の区別で文脈の違いを示している。

単純なユースケースでは、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。

柱を輪切りにしている大きな円が自分が持つ多層的な知人関係を指し、輪切りにされている二つの自己の面「A」および「A'」が、「人格」を示している。「人格

A」を含む層で結ばれている知人には、別の層に存在する「人格 A'」とこの「人格 A」が同じ人間である事は分からない。

さらに、「知人関係の分類および距離による強い制御」を実現するために、自分が持つ複数の「人格」を結ぶ「同一人物リンク」を定義する。同一人物リンクを含めた概念図を 3.2 に示す。自分が持つ二つの「人格 A」および「人格 A'」に対し、「人格 A」から「人格 A'」に同一人物リンクを設定することで、「人格 A」の存在が「人格 A'」の層から見えるようになる。言い換えれば、「人格 A'」の知人は、「人格 A'」と「人格 A」が同一人物ということを知ることができる。この同一人物リンクは方向をもつため、逆に「人格 A」の知人からも、依然「人格 A'」が同一人物であることは分からず、全く関係のない他人として見える。

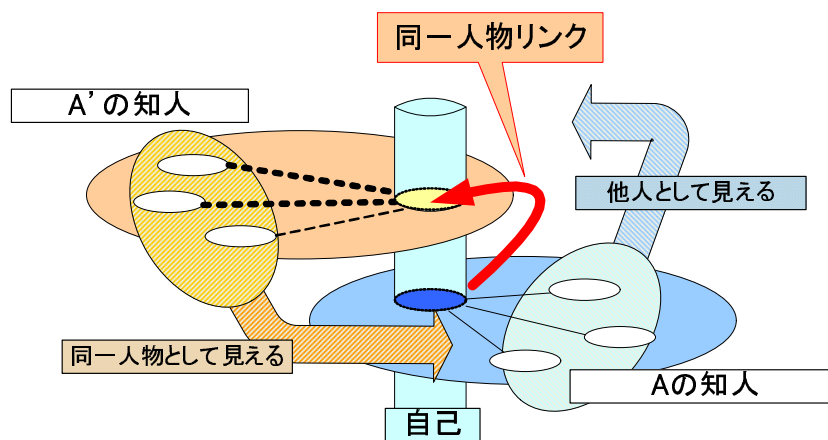


図 3.2: 同一人物リンク

## 3.2 複数人格モデルの詳細

### 3.2.1 人格と文脈

このモデルでは、それぞれの利用者は一つ以上の「人格」を持ち、それぞれの人格ごとに一つ以上の「文脈」を持つ。これらを制御したい粒度に合わせて使い分ける。

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、利用者は図 3.3 で示すように、システムの中に持つ 1 つ以上の人格をその行動主体として操作する。

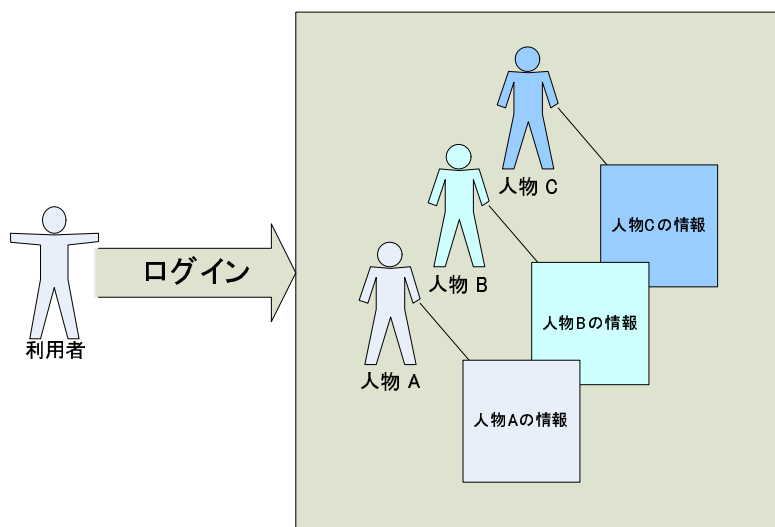


図 3.3: 利用者と複数の人格

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。

この二つは、人間がどれだけ自分の行動を秘密にしておきたいかに依存して変化する自己情報コントロールの粒度である。理想的には様々な粒度でのコントロールがあり得るが、それを二つの段階として定義したのが本モデルの特徴である。

### 3.2.2 自ら発信する情報

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、本名などの唯一性の高いものを使う場合には検索によって別の人格の存在が明かされてしまうことになる。これは知人ネッ



トワークに関する問題ではなく、複数のウェブサイトを別個に運営している場合に、唯一性の高い名前を共有して使っている場合は検索エンジンによって知られてしまう等のような一般的な問題である。そのため、これは利用者が運用によって注意すべき範囲である。

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。誕生日や職業等を公開している場合に、その同一性から推測の危険性が高まる点は名前と同様である。

### 3.2.3 他者との関係に関する情報

他者との知人関係のような情報は、自分だけでは制御できないという点から、自ら発信する情報以上に、自分がどんな人であることを示す重要な情報となる。知人関係をオンライン上で可視化することによる活動の支援がこの知人ネットワークを利用した SNS において重視されることも多く、たとえば閲覧者に対して、自分の知人を紹介するような紹介文機能の普及は、この点をよく表している。

よって、特定の知人に対して、自分のある側面は見せたくないような場合には、その知人関係なども別の人格として分離しなければならない。

### 3.2.4 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ことを、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。

また、閲覧者の情報を記録する「アクセスログ」機能が必要な場合には、どの人格で相手を閲覧しているのかを、閲覧者が選択できなければならない。選択方法としては、常に画面内にどの人格で活動しているかを表示する、知人ネットワーク上での距離が最も近い人格を選択するなどが考えられる。

### 3.2.5 同一人物リンク

第 3.1 節で述べたように、同一人物リンクによって、複数に分かれた人格の間に関係を作る。この関係は、情報の流れる方向を示しており、同一人物リンクの始点となる人格を同一視人格、終点となる人格を主人格と呼ぶ。

主人格の知人に対し、同一視人格はその同一人物として知ることができ、同一視人格において「知人にのみ公開」として設定された情報も、主人格の知人が閲覧できる。言い換えれば、知人ネットワークというグラフ構造において、同一人物リンクは長さ 0 の有向リンクとみなすことができる。

### 3.3 まとめ

ここまで述べてきたように、システム上で同一視されないモデル化が実現したとしても、それを使うのは人間であり、その運用次第で守られるべき秘密の強度が如何様にも変化する。

しかし、この複数人格モデルは人間が実際に行っているような「立場の使い分け」という自己情報コントロールの手法に近いものと考えられ、その親和性の高さから習得の困難なく受け入れられるものである。

これまでオンライン上で新しいコミュニケーションツールが設計されてきたが、実際に多くの人間が利用するためには、そのツールによって得られるメリット以上に、ツールを使うための習得コストというデメリットが極めて低くなければならない。本研究の複数人格モデルは、何よりこの点において優れていると考えている。

## 第4章 先行実験

本研究を進めるにあたり、簡易な実装を用いた先行実験を行った。本章ではその結果を示す。

### 4.1 先行実験の概要

#### 4.1.1 先行実験の目的

先行実験の目的は、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のプロトタイプを実装することで、実装における機能要件を明らかにすることである。

二点目は、SNSとして実装した本モデルを実際に利用してもらい、複数人格モデルに対する意見を集めることである。

#### 4.1.2 実験の手順

今回の実験では、機能要件を明らかにし意見を集めるためのアンケートを行うため、著者を起点とする招待制度を採用し、著者の知人には、著者と直接の知人関係を持たない利用者を招待するよう依頼した。また、この招待制度には、知人ネットワークが著しく分断することを防ぐ目的もある。

具体的には、以下の手順で参加者を応募した。

なお、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。

最終的に、システムの利用者に対してアンケート募集の告知を行うことで、アンケートを実施した。

## 4.2 実験結果

先行実験の結果は以下の通りである。

### 4.2.1 利用状況

実験を行った平成 17 年 7 月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

### 4.2.2 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 4.3 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。

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

### 4.2.3 実装要件の確認

先行実験において明らかになった実装要件について述べる。

**規模性** SNSでは、様々な情報に対してその可視性を制限することができる。これは情報を表示するたびに閲覧可能かどうかを演算しているということである。

本研究で提案する複数人格モデルでは、これまで利用者同士を結んでいた知人関係のかわりに、利用者が複数持つ人格を単位として結ぶ。つまり、相手が自分の知人かどうかを知るためには、自分が持つ人格それぞれに対して相手との関係を調べなければならない。

リレーショナルデータベース上では、JOIN 結合を行うテーブルが一つ増えてしまうため、インデックス等を正しく設計し、計算量を最小限に保つことが必要となる。

**ユーザインターフェース** 複数人格モデルでは、一つしか人格を持たない利用者を許容している。人格の使い分けを必要としない利用者は行動主体を使い分ける必要がない。そのような利用者に対しては、操作コストを軽減し混乱を防ぐため、可能な限り複数人格モデルを採用しない既存の SNS と変わらない操作体系を提供しなければならない。

#### 4.2.4 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。アンケート結果の詳細については付録 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。

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 [15] 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 [15] 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、複数の人格という、完全に分離したモデルだけでなく、文脈に基づいた制御も実現できることが望ましい。

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ふ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。



# 第5章 アプリケーションの設計と実装

本章では、複数人格モデルをウェブアプリケーションとして実装した Perso の設計と実装について述べる。

## 5.1 実装の方針

本研究で提案するモデルは、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。mixi や GREE[16] のような既存のサービスとの比較に基づいた評価が可能となる。

## 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 566MHz
OS	CentOS release 4.2
開発言語	Perl 5.8.5
フレームワーク	Catalyst 5.61 mod_perl 2.0.0 RC5 / Apache 2.0.52
OR マッパー	DBIx::Class 0.04001 / DBD::mysql
テンプレートエンジン	Template-Toolkit 2.14
RDBMS	MySQL 5.0.17

今回の実装には、全てオープンソースソフトウェアによるコンポーネントを利用した。

ウェブアプリケーションのフレームワークとして Catalyst[17] を採用した。Catalyst は Apache[18] mod\_perl[19] を含む複数のウェブサーバ上において、MVC アーキテクチャの枠組みを提供する Perl[20] ライブラリ群である。実装に当たり、既存のプラグイン等の豊富さから選択した。

さらに本実装では、Template-Toolkit[21] をテンプレートビューパターンによるビュー部に、DBIx::Class[22] による O/R マッピングライブラリをアクティブレコードパターンによるモデル部に採用した。知人ネットワークの操作などは、DBIx::Class によるアクティブレコードに対して実装を行った。そのため、知人ネットワークに関する情報はモデル部だけで完結している。

これらのフレームワーク、モデル部およびビュー部の上にウェブアプリケーションとしてユーザインターフェースを構築した。

RDBMS には MySQL[23] を採用した。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 DBIx::Class による O/R マッピング層を経由しており、他の RDBMS に移行することも容易である。

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。今回の実装においても、分散化をふまえて設計を行った。

## 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。既に述べたように、Catalyst 等の既存のオープンソースモジュールを利用することにより、複数人格モデルによる SNS という本質の部分のみを今回実装した。

具体的には、知人ネットワークに関する操作を実装したアクティブレコードオブジェクトである Perso::Model::DB と、ウェブアプリケーションの動作を実装した Perso::Controller および対応する HTML テンプレートが該当する。

全体は、大きく五つのモジュールから構成されている。それぞれ、フレームワーク、コントローラ、モデル、ビュー、データベースである。うち三つは MVC アーキテクチャのそれぞれに対応する。

フレームワーク ウェブサーバ Apache を Perl プログラムのアプリケーションサーバにする mod\_perl モジュール上で、アプリケーションフレームワークである Catalyst が動作する。Catalyst はクライアントからの接続を受けると、あらかじめ登録されたルールに基づいて、適切なコントローラを呼び出す。また、セッション管理

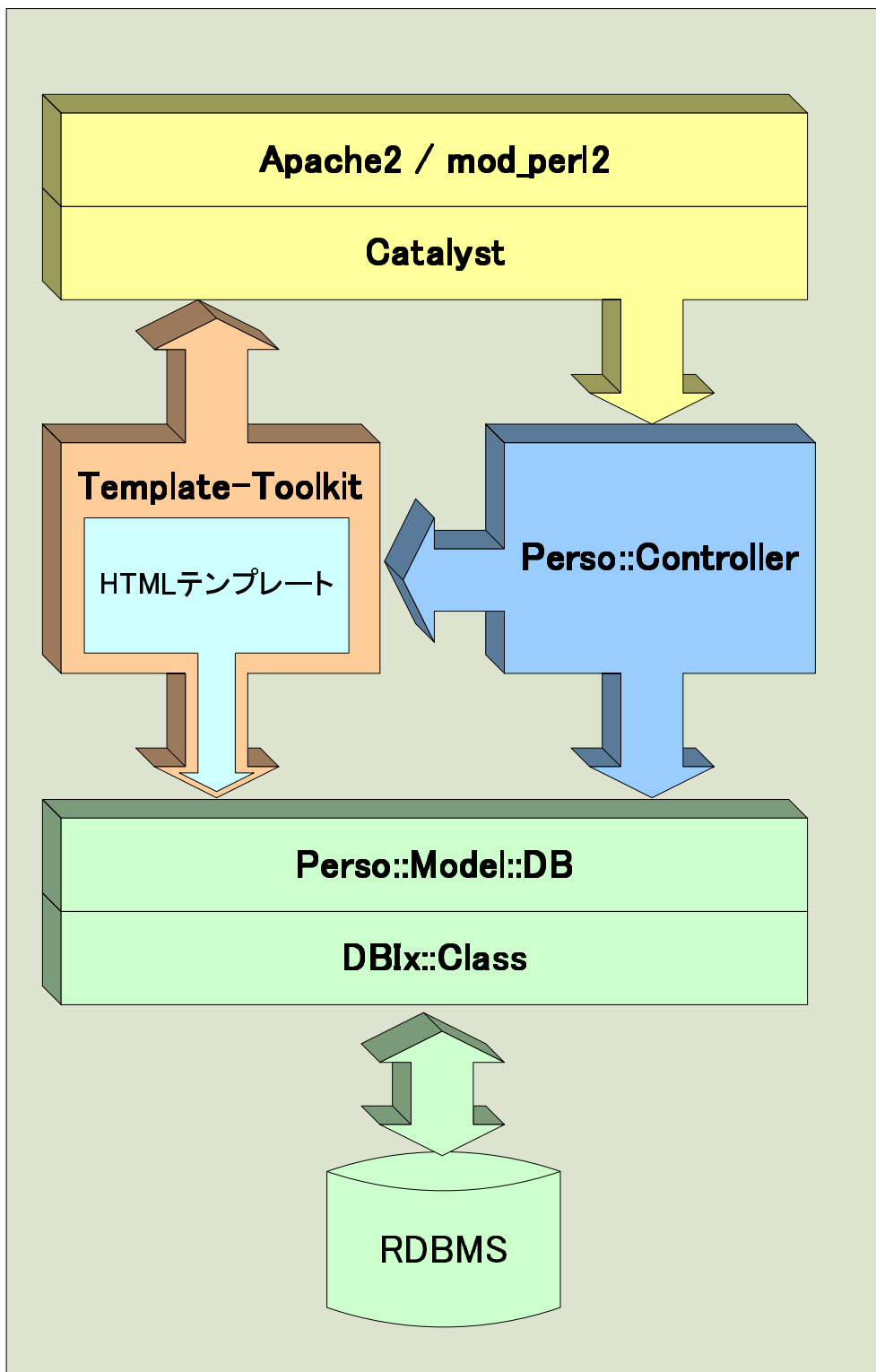


図 5.1: システム全体のモジュール構成

や認証もフレームワークの機能を利用している。今回利用した Catalyst のプラグインの一覧を表 5.2 に示す。

表 5.2: Catalyst Plugins

プラグイン名	機能
Dumper	デバッグ用: オブジェクト詳細の表示
StackTrace	デバッグ用: スタックトレース
DateTime	日付管理
Config::YAML	全体の設定ファイルの処理
DefaultEnds	テンプレートへの遷移処理
FormValidator::Simple	入力値の検証、オブジェクト化
RequestToken	CSRF 攻撃の回避
FillInForm	フォーム内容の展開
Email::Japanese	日本語処理を含むメール送信
Session Session::Store::File Session::State::Cookie	セッション管理
Authentication Authentication::Credential::Password Authentication::Store::DBIC Authorization::Roles	認証処理
Static::Simple	静的ファイルの提供
Redirect	リダイレクト処理

**コントローラ** Catalyst からアクセスごとに呼び出され、様々なウェブによるユーザインターフェースを実装する。

必要に応じてモデルから情報を取得もしくは更新した後、ビューに遷移する。

**ビュー** コントローラから呼び出され、必要に応じてモデルから情報を取得して指示されたテンプレートを元に HTML 画面を出力する。必要な値をテンプレート内に適用するテンプレートエンジンとして、Template-Toolkit を利用している。なお、ビューからはモデルは更新されない。

**モデル** DBIx::Class 上に構築されたアクティブレコードオブジェクトで、複数人格モデルそのものを実現している知人ネットワークに関する情報や、SNS としての機能において必要とされる情報を管理する。

コントローラ・ビューからの情報取得やコントローラからの更新指示を元に、データベースに対して SQL 言語によるクエリを発行する。

データベース Perso 内の情報を蓄積するリレーショナルデータベースで、今回の実装では MySQL 5.0.17 を利用している。

表 5.3 にデータベース上のテーブルとモデル上のオブジェクトの一覧を示す。

さらに、各テーブル間の関係を図 5.2 に示す。これは MySQL の提供する外部キーにより関連づけられ、利用者の退会や、人格の削除などが起きても情報の矛盾が発生しないようになっている。

表 5.3: テーブル一覧

テーブル	オブジェクト	内容
users user_roles roles people	Users UserRoles Roles People	利用者 利用者の権限 利用者の権限内容 人格
link_requests links sameperson faces fof	LinkRequests Links Sameperson Faces Fof	知人リンクの希望 知人関係 同一人物リンク 文脈 (知人カテゴリ) Friend of Friend
invitations	Invitations	Perso への招待状
person_attribute_types person_attributes	PersonAttributeTypes PersonAttributes	人格のプロフィールの項目 人格のプロフィール
diary_articles diary_comments diary_permission diary_trackbacks	DiaryArticles DiaryComments DiaryPermission DiaryTrackbacks	日記の記事 日記の記事へのコメント 日記の記事の公開範囲 日記の記事へのトラックバック
communities person_has_communities community_comments community_invitations community_topics	Communities PersonHasCommunities CommunityComments CommunityInvitations CommunityTopics	コミュニティ コミュニティへの所属関係 コミュニティのコメント コミュニティへの招待 コミュニティのトピック
incoming_comments incoming_diaries incoming_events	IncomingComments IncomingDiaries IncomingEvents	新着コミュニティコメント 新着知人日記 新着イベント
system_message	SystemMessage	システムメッセージ



## 5.4 サイト構成

図 5.3 および図 5.4 に、複数人格モデルに関する部分を抽出した大まかなサイト構成を示す。

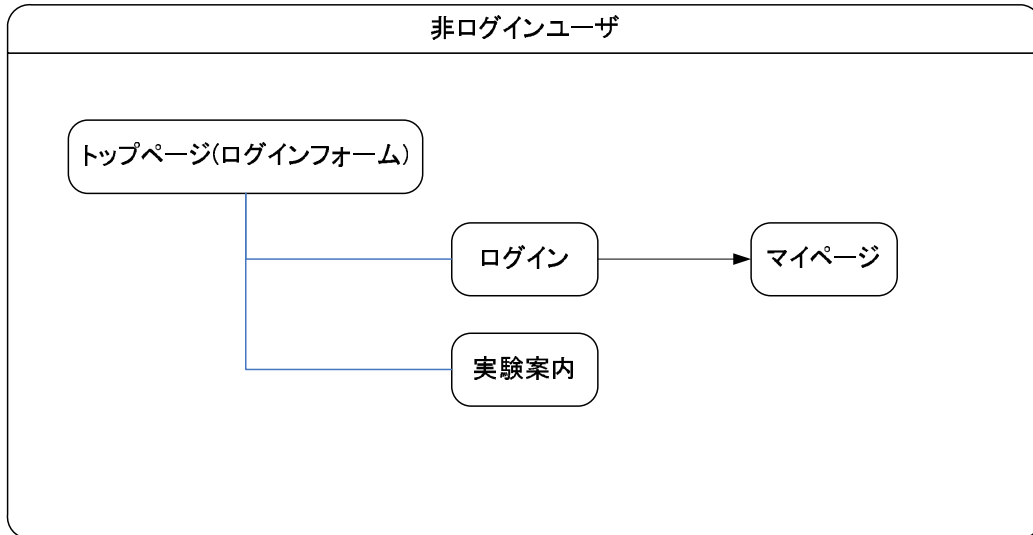


図 5.3: Perso ウェブサイト構成 (非ログインユーザ)



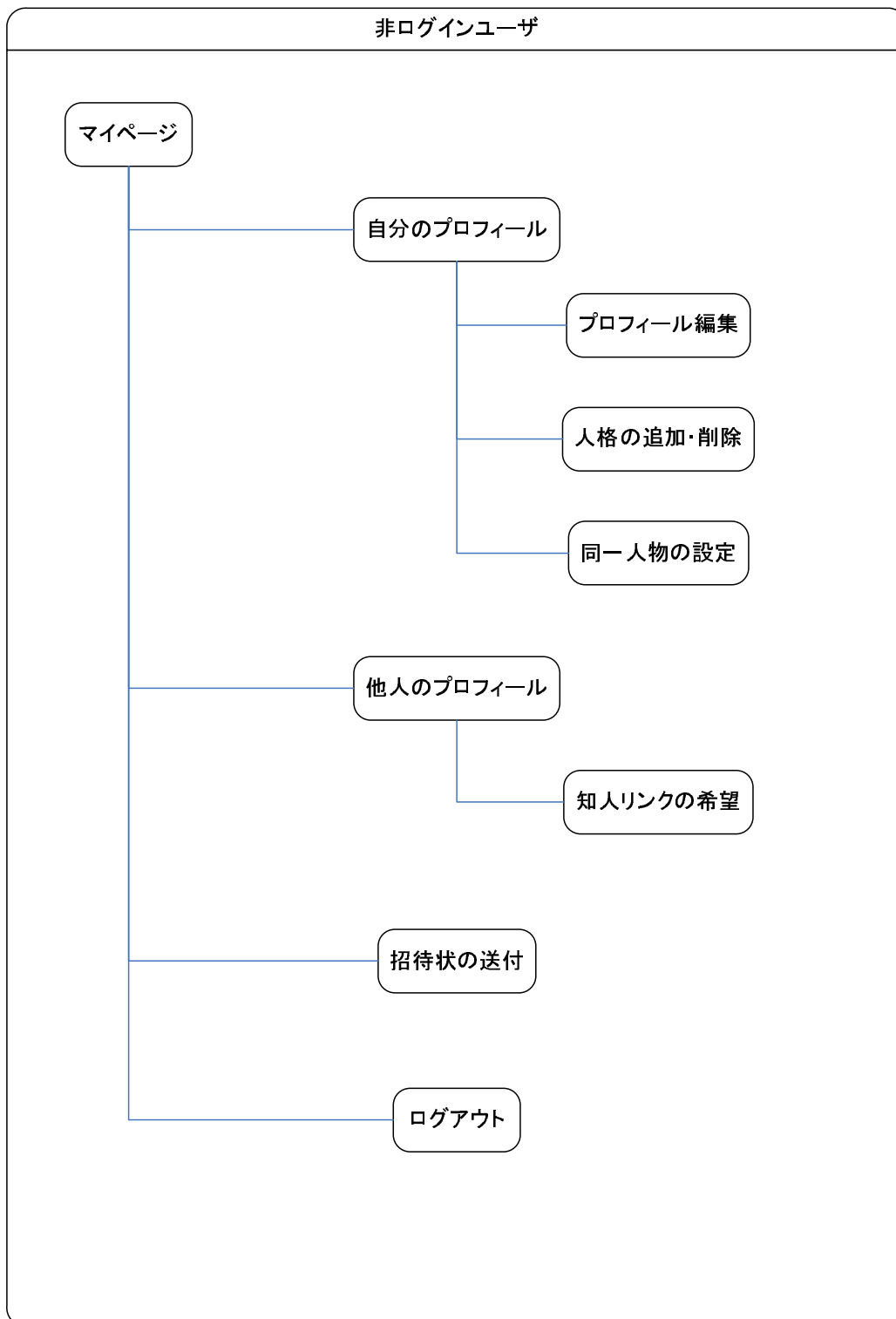


図 5.4: Perso ウェブサイト構成 (ログインユーザ)

### 5.4.1 トップページ (ログインフォーム)

まず利用者は、登録されたメールアドレスとパスワードを用いてシステムにログインする。これは、いくつ人格を使い分けていても認証は一つであり、システムから見れば一人の利用者にすぎないことをはっきりと表している。このことは、複数の利用者アカウントを併用する場合に比べ、システム提供者から利用者の追跡がしやすいという点からも長所と考えられる。

Perso

[トップページ](#)

#### Perso Login

Persoについて  
Persoでは、今までのソーシャルネットワーキングサイト(SNS)とちょっとだけ違う、新しいSNSのあり方を試しています。  
詳しくは、[Persoの紹介](#)をご覧ください。

ログイン

メールアドレス

パスワード

LOGIN

© 2005 Perso

図 5.5: ログインフォーム

## 5.4.2 マイページ

図 5.6 は利用者自身のトップページとなる、マイページである。自分が持つ人格ごとに分かれて、知人関係が表示されている。

「友達の日記」欄には、知人関係を持っている人格にかかわらず、自分の知人全員の新着日記へのリンクが用意されている。利用者はあくまで一人の人間であり、利用者からは全て一様に知人として集約される。

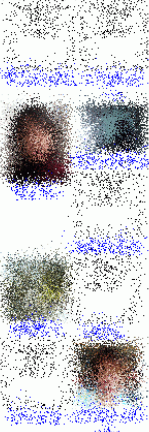
知人リンクの希望を受け取ると上部の「お知らせ」欄に「リンク申請が～件、来ています」というリンクが表示される。このリンクから、知人リンクの希望を管理する画面に移動することができる。これについては後述する。

**Perso**

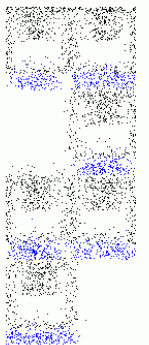


[写真の追加・変更](#)

まさの友達



るるの友達



**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、Winny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では無いんですね。

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「Winnyの技術」も出ないし、そういうお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)

図 5.6: 利用者のマイページ

### 5.4.3 人格ごとのページ

複数人格モデルに基づき、全ての情報は人格を単位として扱われる。この人格ごとに用意されるユーザページを図5.7に示す。このページには、プロフィールや日記などの自ら発信した情報や、知人の一覧などが含まれる

利用者自身のホームページと比較することで、ある人格一つだけに関する情報のみが表示されている点が大きく異なる。

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え**

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、Winnie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「Winnieの技術」も出ないし、そういうお遊びをするとどれくらい噛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**

...

© 2005 Perso

図 5.7: 人格ごとのプロフィール

#### 5.4.4 知人リンクの希望

知人関係を設定するには、双方の合意が必要である。よって、二段階の手順が発生する。

自分や知人以外のユーザページに存在するリンク希望のリンクから、知人リンクの希望フォームに移動する。この図 5.8 に示すフォームから、知人リンクの希望を相手に送信する。長い間疎遠になっていた相手など、名前や公開されているプロフィールだけでは特定できないような場合を考え、知人リンクの希望にはメッセージを付けることができる。

Perseo

トップページ      日記を書く      友人を招待      ログアウト

リンク申請

あなたさんにリンク申請を送ります

リンク申請者: まさ (中山 昌宏)

メッセージ

リンク申請

© 2005 Perseo

図 5.8: 知人リンクの要求

次に、図 5.9 に、知人リンク希望の承認・拒否画面を示す。既に述べたように、未処理の知人リンク要求がある場合のみ、利用者自身のマイページ内にこのページへのリンクが設置される。リンクではなく直接画面を開いた場合に、知人リンクの希望を受け取っていなければその旨表示される。

この画面にて承認を選ぶことで、相手との間に知人リンクが設定され、知人にもみ公開される情報などがみられるようになる。また、この新しい知人リンクにより、相手以外にも知人ネットワーク上での距離が短くなり、新しく閲覧できるようになった情報が他に存在する場合もある。

拒否を選んだ場合には、知人リンクを希望した人には一切通知されない。長い期間知人リンクが設定されなければ拒否したことを推測できるが、相手がまだ知人リンクの希望をまだ見ていないのか、拒否を選んだのかは知ることができない。



図 5.9: 知人リンクの承認・拒否画面

## 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 5.10 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。

My Profile

ニックネーム

名前

性別  男性  女性  無回答

生まれた年

誕生日  /

誰にでも公開

友人まで公開

友人の友人まで公開

自己紹介

Perso管理人の中の人です。  
こんばんは。

同一人物関係  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。

変更

© 2005 Perso

図 5.10: 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。



## 5.4.6 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する。図 5.11 に登録画面を示す。

The screenshot shows the '日記を書く' (Write Diary) interface. At the top, there's a blue navigation bar with 'Perso' and links for 'トップページ', '日記を書く', '友人を招待', and 'ログアウト'. Below this, the page title is '日記を書く'. On the left, there's a sidebar with '最近の日記' (Recent Diaries) and a list of links. The main area is titled '新しい日記' (New Diary) and contains a form with the following elements: a dropdown menu for '書く人' (Author) with 'まさ (仲山 昌宏)' selected; a text input field for 'タイトル' (Title); a dropdown menu for '非公開' (Privacy) with '非公開' selected; a large text area labeled '本文' (Body) for writing the diary content; and a '日記を書く' (Write Diary) button at the bottom right.

図 5.11: 日記の登録画面

このように、全ての自ら発信する情報に対して、どの人格が行動主体であるのかが選択しなければならない。

## 5.5 各モジュールの解説

本節では、重要ないくつかのコンポーネントについて詳しく解説する。

### 5.5.1 利用者の管理

本システムの利用者の情報は、users テーブルおよび Perso::Model::DB::Users クラスによって管理される。主キーとして唯一性のある利用者 ID を持ち、認証および連絡に用いるメールアドレスとパスワードをフィールドに持つ。

また、知人ネットワーク上において特定の人格に対して自分からの距離を演算する `distance` メソッドを持つ。この利用者の持つ複数の人格に対して、もっとも近い人格から相手までの距離を返す。ただし、実際に知人ネットワーク上の距離としてアクセス制御に用いるのは「知人の知人」である 2 までであることから、2 よりも遠い人格は最大値である 255 とみなすことで切り捨てている。

この `distance` メソッドは、さらに自分自身かどうかを調べる `is_myself` メソッド、知人かどうかを調べる `is_friend` メソッド、知人の知人かどうかを調べる `is_fof` メソッドの組み合わせにより実現されている。

これらは、データベース上ではインデックスを設定された項目同士の外部キーによる JOIN 演算として実現されている。

### 5.5.2 人格の管理

複数人格モデルにおける知人ネットワークの単位となる人格については、`people` テーブルおよび `Perso::Model::DB::People` クラスで管理する。主キーには人格ごとに割り当てられた識別子を用い、所属する利用者 ID をフィールドに持つ。

人格ごとのプロフィール情報を保持する `person_attributes` テーブルから、距離に応じた閲覧可能な情報のみを検索する `readable_attributes` メソッドなど、人格に関する情報の集中的なインターフェースの役割を果たしている。

知人リンクの構築についてもこのクラスで実装され、知人リンクを設定する `link_friend` メソッドおよび知人リンクを削除する `unlink_friend` メソッドが存在する。これらは、自分を起点とし、指示された相手を終点とするような知人リンクを扱い、後述する `links` テーブルおよび `fof` テーブルに対して操作を加える。なお、「知人の知人」という関係を管理する `fof` テーブルの管理については、この 2 つのメソッドが責任を負う。

## 第6章 評価

本研究の目的は、知人関係の文脈に基づいて異なる情報コントロールをしたいという要求に対し、一人の人間が複数の人格を管理し使い分けることで実現するモデルを提案することである。本章では、このモデルの必要性および要件、有効性について議論する。

### 6.1 必要性

先行実験におけるアンケート結果を元に本研究の必要性について議論した。

その結果、情報コントロールの粒度を問わなければ、何らかの形で文脈を使い分けたいという要求は多数存在することが明らかになった。また、Q204の自由記入欄においても、複数の人格の管理可能性についての意見が出たように、本研究のモデルが必要とされていることが推測できる。

### 6.2 要件

先行実験により必要性が確認された本モデルに対し、第2章にて定義した要件を満たしていることを確認する。

#### 6.2.1 複数の文脈の使い分け

#### 6.2.2 複数の文脈の使い分け

人間が持つ複数の文脈を知人ネットワーク上で表現するために、本モデルでは活動主体を人格として形で抽象化した。人格それぞれにプロフィールや知人関係が存在することから、文脈を使い分けるのに必要な分離を実現できている。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。

### 6.2.3 管理可能性

システム内では、利用者は利用者として、複数の人格を持つ立場として表現されている。また、複数の人格の間に同一人物リンクを設定することで、限定的にそれらの人格が一人のものであることを知人に見せることもできる。

このように本モデルにおいては、利用者と人格という抽象化によって管理可能性を実現している。

## 6.3 その他の要件の詳細

### 6.3.1 内容

情報の内容に対する要件として、「自ら発信する情報」および「他者との関係に関する情報」の二つを定義した。

「自ら発信する情報」は人格それぞれに結びつけられる。「他者との関係に関する情報」は、複数の人格に対して結びつけられる。

### 6.3.2 公開範囲

情報の公開範囲に対する要件としていくつかの指標を定義した。本モデルをその指標に基づいて評価する。

公開 全く制限を掛けないことにより、「公開」を実現している。

知人関係の距離による制御 人格どうしを結ぶ知人関係に基づいて、知人ネットワーク上での距離計算を行うことで実現している。

知人関係の分類による制御 知人関係を、異なる文脈に属させることにより、可視範囲の分離を実現している。

知人関係の分類による強い制御 知人関係を、異なる人格に属させることにより、可視範囲の分離を実現している。

知人関係の分類および距離による強い制御 知人関係を、異なる人格に属させたうえで、自分の人格同士を同一人物リンクによって結ぶことにより、可視範囲の分離に対して単方向的な「穴」を作ることができる。

## 6.4 ユースケースとの比較

提案したモデルが要件を満たすことを確認した。そこで、要件定義に用いた個別のユースケースとの比較により、多様なコミュニケーションの形態をどのように本モデルで実現しているかを示す。

**単純なユースケース** このユースケースでは、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。

**弱い使い分け** このユースケースでは、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。

**強い使い分け** このユースケースでは、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。

**限定的な強い使い分け** このユースケースでは、前節のユースケースのようにただ別人として見えるだけでなく、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、知人にのみ別の人格が同一人物として示される。

まとめ 以上のユースケースをまとめたものを表 6.1 に示す。

表 6.1: ユースケースのまとめ

事例	実現方法
単純な事例	単一の人格、単一の文脈
弱い使い分け	単一の人格、複数の文脈
強い使い分け	複数の人格
限定的な強い使い分け	複数の人格、同一人物リンク

このように、要求分析で提示したユースケースを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。

## 6.5 モデルの考察

本節では、本モデルに対して考察を加える。

### 6.5.1 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最低限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。

**現実との比較** 最終的に複数人格モデルを利用するのは人間であり、人格や文脈をどのように使い分けるかは人間が考えなければならない。本研究で提案した複数人格モデルは、現実で行われているコミュニケーションにおいて想定される様々なケースを実現することができ、人間が元来脳で行っているモデルに近いものだと考えられる。

また、先行実験にて実施したアンケートの Q103、Q104 への回答から、オフラインにおける実名と、オンライン上でのハンドルという異なる名前で活動をしているという意見が見られた。Q201 においては、「相手によって自分の立場を使い分けたい」「ハンドルと実名では交流関係も違う」という意見があった。このように交流関係の違いに対して、実名とハンドルという名前の違いを関連づける意見が複数見られたことは、それぞれが異なる名前を持つ行動主体となる複数人格モデルと近い考え方をしているということが推測できる。

## 6.6 他方式との比較

本節では、他の方式との比較を行う。

### 6.6.1 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。

この方式では、個別の利用者であるユーザおよび、複数の利用者の集合であるグループを単位とする。

利用者には一つのユーザと、複数のグループが関連づけられている。情報にはそれぞれユーザおよびグループが一つずつ関連づけられている。情報に対し、関連づけられたユーザ、関連づけられたグループに所属するユーザ、それ以外の三種の区別によってアクセスを制御する。

この方式では、「他者との関係に関する情報」はグループに相当するが、誰がどのグループに属しているかは公開されている。すなわち情報の公開範囲の制御としては「自ら発信する情報」のみが対象となる。

ユーザが複数所属するグループから、情報に関連づけるグループを選択し、そのグループに対してアクセスを許可することでこれは、「知人関係の分類による制御」を実現できる。ただし、情報の存在そのものの秘匿性は考慮されていない。これは SNS においてたとえるならば、許可されていない人にも日記を書いていることは分かるが、その内容を知ることはできないというユースケースに相当する。

このアクセス制御方式では、まず「他者との関係に関する情報」を制御することができないという点で本モデルと大きく異なっている。さらに、「自ら発信する情報」についても、「知人関係の分類による制御」までしか実現していない。

### 6.6.2 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。この方式では、「他者との関係に関する情報」および「自ら発信する情報」の一つずつに対し、誰に見せるかを細かく設定することができる。

しかし多層的な人間関係が存在していても、実際に必要とされるのは、「ある特定の背景での知人」に限定したい、という場合が多い。つまり、アクセス制御リストが必要とされるのは、相手との関係が持つ背景ではなく、モデル化が難しいような細かい制御を運用によって実現したい場合に、その記述方法として求められている場合のみである。

また、アクセス制御リストでは全ての情報に対して個別に公開範囲を設定できるが、言い換えれば自分が管理する情報の数に、自分の知人等の数を乗じただけ設定項目が発生する可能性があるということである。

いわば本研究が提案する複数人格モデルは、人格と文脈という枠により柔軟性を犠牲にして設定コストを軽減させたものである。既に述べたように、本研究の目的に基づく要件は既に満たしていることから、利用者に対し膨大な設定コストを強いる必要は無い。

## 6.7 モデルに対する評価のまとめ

以上から、モデルが目的を満たすために必要十分なものであることが確認された。

## 6.8 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。

### 6.8.1 実験目的

本実験の目的は、先行実験の結果に基づいて設計した複数人格モデルの実装により、規模性が本モデルの採用の妨げとならないことを明らかにすることである。

### 6.8.2 結果

複数人格モデルにおいて規模性を失わない実装は十分に可能であり、規模の大きい既存のSNSにおいても本モデルを利用することができる。次節において、具体的な処理内容と比較して検証を行う。



### 6.8.3 規模性

機能毎に計算量の評価を行い、利用者や知人リンクの増加に対する規模性が、既存の SNS と比較して悪くなっていないことを確認する。なお、アプリケーションの性質上圧倒的に更新より参照の回数が多いことから、更新に必要な計算量については省略する。

**セッションの確認** ページを表示する際に、アクセスしているのが正しく認証された利用者かどうかを毎回確認しなければならない。HTTP cookie によるセッションを利用しており、セッションに関連づけられた利用者が存在するかを確認している。ページ表示ごとに発行されている SQL 文は以下の通りである。

```
SELECT me.passwd, me.email, me.registered_time, me.id FROM users me
WHERE ( email = ? )
```

この SQL 文が示すように、本モデルにおいて、利用者が単一の ID のみ所有する点で既存の SNS との差異は無い。よって、複数人格モデルの採用がセッションによる認証情報の確認に与える影響は無い。

**知人の一覧** マイページやユーザページなど、多くの場面で知人の一覧が必要とされているが、一利用者が複数の人格を持つことから、それぞれの人格ごとに知人を調べなければならない。該当する SQL 文は以下の通りである。

```
SELECT me.name, me.user_id, me.id FROM people me
WHERE ( user_id = ? )
SELECT me.name, me.user_id, me.id FROM people me
LEFT JOIN links link_from_me ON ( link_from_me.link_from = me.id )
WHERE ( link_from_me.link_to = ? )
SELECT me.name, me.user_id, me.id FROM people me
LEFT JOIN links link_from_me ON ( link_from_me.link_from = me.id )
WHERE ( link_from_me.link_to = ? )
```

これは、まず自分の利用者 ID(`user_id`) から自分の人格の一覧を求め、それぞれの人格が持っている知人関係を集めるという動作を示している。つまり、この処理は最初に人格の一覧を取得した後、その人格の数だけ繰り返すということになる。O 記法では一人が持っている人格の数  $n$  に対して  $O(n)$  となる。

しかしながら、現実には人間一人が使い分けることができる人格の数には限度があり、極めて少ない数に留まると予想される。また、一人あたりの人格の数は利用者数の増加には依存しない。よって、利用者の増加に対する規模性に何ら影響は無い。

また、自分に対するリンク希望者に対しても、リンクとほぼ同様の計算を行っているため同様のことが言える。

自分自身かどうかの確認 様々な処理において、閲覧者と自分との知人ネットワーク上における距離を求めなければならない。今回の実装では「知人の知人」という距離が2までしか必要としないため、自分自身かどうか、知人かどうか、知人の知人かどうか、を順次調べることで距離を求める処理とした。

自分自身かどうかを調べる処理については、既存の SNS では、認証されているかどうかを確認した時点で自分の利用者 ID が判明しているため、データベースへの問い合わせ無しに、閲覧先の ID と単純に比較することで完了する。

複数人格モデルでは、利用者をシステムが識別し、さらにその利用者 ID に対して複数の人格が存在する。これをデータベースから取り出して比較しなければならない。該当する SQL 文は以下の通りである。

```
SELECT me.name, me.user_id, me.id FROM people me
      WHERE ( ( ( user_id = ? ) AND ( id = ? ) ) )
```

今回は純粋に people テーブルに対して検索を行った。この処理では人格 ID が people テーブルの主キーであり、索引が利用される。そのため、常に最低限の計算量となる。よって、規模性に影響することはない。

知人かどうかの確認 知人かどうかの確認には、links テーブルを利用するが、その相手から同一人物リンクが存在する場合には、どの同一人物リンクの終点が知人かどうかによっても判断しなければならない。この処理が複数人格モデルにおいて、既存の SNS と比較して複雑になっている部分である。以下に SQL 文を示す。

```
SELECT COUNT(*) FROM links me
      JOIN people link_to ON ( link_to.id = me.link_to )
      LEFT JOIN sameperson rel_sameperson
            ON ( rel_sameperson.same_to = me.link_from )
      WHERE ( ( link_from = ? AND link_to.user_id = ? )
            OR ( link_to.user_id = ? AND rel_sameperson.same_from = ? ) )
```

この処理では、people、links、sameperson の三つのテーブルに対して JOIN 結合を行っている。より詳細な検討を行うために、EXPLAIN 文によりクエリの内部処理について調査した。

```

mysql> EXPLAIN SELECT COUNT(*) FROM links me
-> JOIN people link_to ON ( link_to.id = me.link_to )
-> LEFT JOIN sameperson rel_sameperson
-> ON ( rel_sameperson.same_to = me.link_from )
-> WHERE ( ( link_from = 8 AND link_to.user_id = 1 )
-> OR ( link_to.user_id = 1 AND rel_sameperson.same_from = 8 ) ) \G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: link_to
         type: ref
possible_keys: PRIMARY,people_FKIndex1
         key: people_FKIndex1
        key_len: 4
         ref: const
         rows: 1
      Extra: Using index
***** 2. row *****
      id: 1
  select_type: SIMPLE
        table: me
         type: ref
possible_keys: PRIMARY,links_pairunique,links_FKIndex1,links_FKIndex2
         key: links_FKIndex2
        key_len: 4
         ref: persotest.link_to.id
         rows: 1
      Extra: Using index
***** 3. row *****
      id: 1
  select_type: SIMPLE
        table: rel_sameperson
         type: ref
possible_keys: sameperson_FKIndex2
         key: sameperson_FKIndex2
        key_len: 4
         ref: persotest.me.link_from
         rows: 1
      Extra: Using where; Using index
3 rows in set (0.00 sec)

```

この結果から、MySQL データベースは最初に閲覧者の利用者 ID を people テーブルに問い合わせた人格の一覧を得ている。people テーブルでは外部キー user\_id に対して索引が利用できるため、索引サーチとなっている。次に、people テーブルと JOIN 結合した links テーブルに対してリンクの一覧を得ている。これも同様に索引サーチとなっている。同一人物リンクではなく直接の知人であれば、この段階で処理が完了する。最後に、sameperson テーブルを LEFT JOIN 結合し、同一人物による知人関係があるかどうかを判断している。この時点でもやはり索引サーチが行われ、WHERE 節による絞り込みが行われている。

よって、既存の SNS と比較して処理そのものは複雑にはなっているが、すべて適切なデータベース設計により索引サーチのみで実現可能である。なお、今回実装に利用した MySQL では索引に BTree 法を採用していることから、平均計算量はシステムに登録された人格およびリンク、同一人物の数に対して  $O(\log N)$  となる。

知人の知人かどうかの確認 これは、fof テーブルに対して同様の操作を行うものだが、知人の知人に対しては同一人物による影響がないため、sameperson テーブルを JOIN 結合する必要がない。よって、同じようにすべて索引サーチにより完了する。

## 6.9 今後の課題

本論文では実施しなかった評価手法について最後に述べる。

コミュニケーション量の比較 既存の SNS と同様に複数人格モデルによる SNS を広く提供し、同じような規模の SNS としてコミュニケーションの質や量を測定するという方法が考えられる。

評価のためには通常運用として長期間運用し、利用者を募る必要があることから、本論文の評価としては用いなかった。

## 6.10 まとめ

以上から、複数人格モデルという本研究で提案する新しい知人ネットワークのモデルは十分に有効であり、また実際の環境においても規模性があることを示した。

また、先行実験におけるアンケート結果から、自分の情報の全てを見せずに済む「自己情報コントロールが可能なコミュニケーション手段」が求められていることは明らかになっており、このような要求を十分満たすことができる。SNS というサービスの運営にはコストがかかるが、規模性におけるデメリットが無いことから既存の SNS 事業者が本モデルを採用するということも可能である。

# 第7章 結論

## 7.1 本研究の成果

本研究では、知人関係などにより人間が持つ文脈ごとに、知人ネットワーク内でも文脈を切り替えられるモデルとして、複数人格モデルを提案した。

このモデルでは文脈ごとに別の行動主体となる人格を使い分けることで、文脈ごとに自分の見え方を制御し、秘密を守ることができる。この複数の文脈は個別独立に存在するものではなく、利用者からは多層的な一つの軸として扱えなくてはならない。このモデルにより、利用者は多層的な複数の文脈を自由に行き来したり、自分の見え方の制御に関して、自らの複数の人格が同じ自分であることを限られた人にだけ明かしたりといった自己情報コントロールを実現した。

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キングサイト (SNS) を構築し、以上のような社会的要求を複数人格モデルが満たすことを確認した。また、規模性の面でこれまでの知人ネットワークに劣ることなく実現できることを明らかにした。

## 7.2 今後の課題

### 7.2.1 分散化

本モデルではリレーショナルデータベース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[24] 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ケースが考えられる。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ユースケースを対象外とし、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、前述のようなユースケース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。

# 謝辞

本研究を進めるにあたり、ご指導を頂いた主査の村井純教授、副査の中村修助教授、徳田英幸教授に感謝いたします。また、neco 及び近縁 pub という場を通して手厚く指導を頂いた斉藤賢爾氏、須子善彦氏にも厚く感謝を申し上げます。研究者としての資質に欠ける私を見捨てず最後まで助言を頂けたことは、本当に頭が上がりません。

湘南藤沢キャンパスで過ごした7年間は、ふとした縁で環境情報学部1年の初登校日に 31 教室の扉を叩いた日から、この修士論文の提出に至るまで、ずっと村井教授の研究室と共にありました。この研究室では様々な経験をさせて頂きました。ある時は地方自治体の情報化、ある時は諸外国を NetMeeting で繋いだ学会の実施、ある時は研究会イベントの運営、そして何よりインターネット技術に関する研究に携わることができました。ですが、私にはまだこの経験を社会にフィードバックするという、重要なミッションが残されています。今後数十年間、そのミッションを終えることができはじめて、ようやくこの村井研究室を引退することができるのだと考えています。

この A4 の紙には書ききれないほど沢山の方のおかげで、この修士論文は提出することができそうです。特に、インターネットを通じて常に私を励まし続けてくれた IRCnet の某チャンネルの住人の皆様、卒業論文でのスターバックスにかわり集中して思考できる空間を提供して頂いた秋葉原の某喫茶店にはお世話になりました。こうやって並べてみると、私の「実名ではない別の名前」で繋がっている関係ばかりですね。私を含めてこのような多層的な知人関係を持つ人がもっと幸せになれるよう、学生という籍を離れた後も頑張っていこうと思います。

最後になりますが、湘南藤沢キャンパスの村井研究室で研究者として過ごした7年間はどう見ても私にとって最良の期間です。本当にありがとうございました。

## 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] さまざまなコミュニティサービスの枠組みと可能性 :小林 scrap book.  
<http://blog.heartlogic.jp/archives/000592.html>.
- [3] ネット社会と実社会のギャップを埋めるには.  
<http://easy.mri.co.jp/20051115.html>.
- [4] 社会情報学フェア 2005 web が生み出す関係構造と社会ネットワーク分析ワークショップ. <http://ymatsuo.com/si2005/>.
- [5] Itmedia ニュース:研究対象としての「mixi」.  
<http://www.itmedia.co.jp/news/articles/0509/14/news040.html>.
- [6] 手嶋屋 ニュース openpne とは. [http://blog.livedoor.jp/tejimaya\\_news/archives/2004-01.html](http://blog.livedoor.jp/tejimaya_news/archives/2004-01.html).
- [7] 留学生 sns. <http://japanic.jp/>.
- [8] アキバ系 sns filn. <http://www.filn-blog.jp/>.
- [9] 世界初オタク系 sns otaba. <http://otaba.jp/>.
- [10] オタバ公式ブログ: 会員数 10000 人突破.  
<http://otaba.seesaa.net/article/11527359.html>.
- [11] ソーシャル・ネットワーキングサイト [mixi(ミクシイ)]. <http://mixi.jp/>.
- [12] imeem! <http://www.imeem.com/>.
- [13] livedoor フレパ. <http://frepa.livedoor.com/>.
- [14] wyco 20代のネットワーク最適化 sns. <http://wyco.jp/>.
- [15] 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/>.
- [16] グリー (gree) : ソーシャルネットワーキングサービス. <http://gree.jp/>.



- [17] Catalyst - web framework. <http://catalyst.perl.org/>.
- [18] The apache http server project. <http://httpd.apache.org/>.
- [19] mod\_perl: Welcome to the mod\_perl world. <http://perl.apache.org/>.
- [20] Perl.com. <http://www.perl.com>.
- [21] Template toolkit home page. <http://www.template-toolkit.org/>.
- [22] Dbix-class - search.cpan.org. <http://search.cpan.org/dist/DBIx-Class/>.
- [23] Mysql ab :: The world's most popular open source database.  
<http://www.mysql.com/>.
- [24] Affelio - -the open social network-. <http://www.affelio.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