

## 人間関係ネットワークに基づく情報フィルタリングを用いた協調的タスクスケジューラ

大向 一輝<sup>†a)</sup> 武田 英明<sup>††b)</sup>

Collaborative Task Scheduler Using Information Filtering Method with Social Network

Ikki OHMUKAI<sup>†a)</sup> and Hideaki TAKEDA<sup>††b)</sup>

あらまし 本研究では複数の個人による協調的なタスク管理手法モデルを提案し、これを実現するためのアプリケーションとして携帯電話用スケジューラの実装を行った。一般に、複数人が関わる共同作業のようなタスクの管理コストを下げるためには、参加者がスケジュール情報を公開することが望ましい。しかしながら多くの個人は複数の組織に所属しており、全ての情報を公開することは難しい。しかも、それらの組織の一部はアドホックに形成されるためにグループウェアに代表されるトップダウンの管理モデルを適用することは難しい。そこで、本研究では個人同士がタスクの依頼および受理を行った履歴より人間関係ネットワークを生成し、完全グラフとなる部分グラフをグループと見なす手法および発見されたグループ単位での情報のアクセスコントロールを提供する手法によって、プライバシー侵害の問題を低減した上で情報公開を行うことを可能にする。また、本手法では明示的に記述されたプロファイルを必要としないために、ユーザの情報入力付加を軽減することができる。これらの手法を日常的なスケジューリングに適用するために携帯電話アプリケーションを開発し、実証実験を行った結果、半自動的なグループ発見およびアクセスコントロールは有効であるとの結論を得た。

キーワード 協調的問題解決, リソース管理, コミュニティ形成支援, パーソナルネットワーク, 携帯電話アプリケーション

### 1. ま え が き

情報化社会の進展やライフスタイルの多様化は、個人と組織の関係を大きく変えつつある。現在では企業あるいは学校といった既存の組織に加えて NPO・NGO などの民間組織が多数存在し、複数の組織に同時に所属する人々はもはや珍しくない。また、これらの公的な活動組織の他に私的な活動を行うグループが無数に存在する。人々はこういった多くの組織・グループを使い分けることで各々の生活を全うしている。

このような生活の質を高めるためには、個人の持つ時間や金銭、あるいは知識といった資源（リソース）を適切に管理することが重要である。これまでに多く

のリソース管理手法が提案されており、ツールやシステムとして具現化されている。これらは、組織を中心としたトップダウンモデルと、個人を中心としたボトムアップモデルに大別される。

トップダウンモデルの代表例としてグループウェアが挙げられる [1]。グループウェアは組織の生産性を最大限に高めるために、組織内のリソースの予約や情報の共有を可能にする。また、多くの組織には階層が存在し、この階層に従って上位の人間が下位の人間のリソースを管理することが望ましいとされている。グループウェアでは権限のコントロールによってこういった管理を可能にしている。

しかしながら、多くのグループウェアは組織の構成員が持つリソースを、全て組織のために利用できるという前提のもとで設計・運用されているために、先に述べた新しいライフスタイルには適合しない。複数のグループウェアの連携についても、データ構造の違いといった基本的な問題や、異なる組織間でのリソース

<sup>†</sup> 総合研究大学院大学  
The Graduate University for Advanced Studies  
<sup>††</sup> 国立情報学研究所  
National Institute of Informatics

a) E-mail: i2k@grad.nii.ac.jp

b) E-mail: takeda@nii.ac.jp

予約の競合といった高次の問題が発生するために現実的ではない [2] .

一方、ボトムアップモデルの例としては手帳や PIM・PDA を挙げるができる。これらの手法では個人がすべてのリソースを管理するために、複数の組織間のリソース配分は容易である。しかし、個人のための管理手法には恣意的なものが多いために効率的なリソース管理を行うことは難しい。筆者らはこの問題に対処すべく、意思決定論に基づくスケジューラを提案し有効性を検証したが [3] , ボトムアップモデルに共通する情報の入力コストの大きさを改善する方法については研究がなされていない。

そこで、本研究では、トップダウンモデルとボトムアップモデルの両者の問題点を解消し、主体的かつ効率的なリソース管理を可能にする協調モデルを提案する。また、このモデルを携帯電話用スケジューリングシステムとして実装し、実証実験によって有効性を検証する。

本論文の構成は以下の通りである。2 章では対象問題を概説し、リソース管理のための協調モデルを提案する。3 章では人間関係ネットワークおよびタスクの依頼関係ネットワークに基づく情報フィルタリング手法を提案する。4 章では携帯電話アプリケーションとしての実装について述べ、5 章で提案システムの実証実験とその考察を行い、まとめとする。

## 2. 協調的スケジューリング

タスクスケジューリング問題は、個人が抱えるタスクに対して時間リソースを適切に分配する問題である [4] . 基本的な制約条件としては 2 つ以上のタスクに同じ時間リソースを適用できない、すなわちダブルブッキングの禁止がある。また、工場での工程管理や航空・鉄道などのダイヤ編成といった従来の工学的なスケジューリング問題とは異なり、本研究で対象とする問題では参加者である個々の人間どうし、組織どうしはそれぞれ独立であり、異なった目的や制約を持つものであると考えられる。そのため、これまでに研究されてきた単一目的の最適化問題や制約解消問題として扱うことは難しい。同様に、参加者どうしの情報共有の効率化による問題解決手法がナレッジマネジメントやネットワークコミュニティの分野で提案されている [5] [6] が、これらは単一の組織内でのマネジメントや単一のコミュニティの形成をいかに支援するかに主眼が置かれているために、複数の人間が複数の組織を

使いこなすといった状況には適用できない。

多くのタスクは個人が持つ人間関係の中からの依頼や共同作業という形で発生する。ここでの人間関係とは過去のタスクの協働によって発生する 2 者間以上の関係であると定義する。これらのタスクは一度受理すると期日等を変更するためには相手との交渉が必要になる。とくに、こういったタスクが異なる組織によって同じ時間に複数予約された場合には、1 つのタスクについて相手との交渉を行った後に、その結果をもとに別の相手との交渉をしなければならないなど、問題が複雑化し、競合解消のコストが著しく増大する可能性もある。

このような事態を回避するためには、タスクを依頼する時点で今後の交渉の余地の少ない時間帯を選択することが望ましい。

そのためには、参加者が自身の情報を公開することが重要となる。タスクの依頼者が相手のタスク状況を事前に確認し、ダブルブッキングを起こさないように配慮したリソースの予約が可能であれば妥結点への到達が早まる。すでにグループウェアをはじめとするトップダウンモデルでは、組織の構成員に関する全てのタスク情報を共有することで効率的なスケジューリングを実現している。

しかしながら、この手法によって複数の組織間にまたがるスケジューリングを行う際には、プライバシーの侵害の恐れがあるために参加者に対して全ての情報の公開・共有を要請することは現実的ではない。タスクの依頼に際して必要な情報のみを適切に公開することが求められる。

このように、情報の公開に際しては、誰に、どのような情報を公開すべきかというアクセスコントロールが必要である。コントロールの手段としてはプロファイルをユーザ自身に記述させるものが多いが、この方法では入力のコストが極めて大きくなる。そこで、本研究では日常的なリソース管理の行動をシステムが分析し、その結果をもとに情報のアクセスコントロールおよびフィルタリングを実現する手法を提案する。これにより、プライバシーの保護と情報入力コストの問題を同時に解決することが可能になると思われる。また、ボトムアップのアプローチをとることによって、固定的ではないアドホックなグループに対する支援が可能になる。

### 3. タスクの依頼関係に基づくアクセスコントロール

本研究では、前節で述べた問題を解決するために、過去のタスクの協働関係に関する情報を利用し、新たな協働関係を結ぶための支援手法を提案する。提案システムでは、複数人の中でのタスクの依頼関係からグループの同定を行い、この結果をもとに情報のアクセスコントロールを行うことを目指す。以下にユーザモデルおよびアルゴリズムの詳細を述べる。

#### 3.1 スケジューリング決定プロセス

本節ではユーザ側の視点から、提案システムを用いた複数のユーザに関わるタスクの決定プロセスについて述べる。システム側の動作およびアルゴリズムの詳細については次節以降で述べる。

##### ● 相互承認

提案システムによる協調的スケジューリングを行うにあたっては、事前にユーザ間の相互承認が必要となる。まず、ユーザ(A)は他のユーザ(B)に対してタスク情報の閲覧および登録を行ってよいかどうかを問うメッセージを送信する。その際には何らかのコミュニケーションを行ったうえで相手のIDを入手する必要がある。ユーザBはこの通知に対して問い合わせを許可するメッセージを返信すると、ユーザAとユーザBの間に相互承認が成立したとみなされ、相手のユーザ名が関連ユーザリストに登録される。相互承認は相手1人につき1回行えばよい。

##### ● タスク情報の閲覧

ユーザAは、自身が持つ関連ユーザリストに含まれるユーザのスケジュール情報を閲覧することが可能である。携帯端末上のソフトウェアにおいて相手ユーザのIDを指定すると、端末はインターネット上のサーバと通信し、相手ユーザの情報を取得する。複数人のIDを指定し、情報を一括取得することも可能である。取得された情報は、ユーザ自身のスケジュール情報とともに図1に示す時系列インターフェイス上に表示される。時系列インターフェイスは上段と下段に分かれており、上段は数日間の日付が縦に並べられている。横軸は0時から24時までの時間軸である。各行の内部はユーザが閲覧したい人数分に分割されており、各ユーザのタスク情報ないし空き時間<sup>(注1)</sup>の情報が時間

軸に沿ってガントチャート形式で表示されている。この例では最上段がユーザ自身、2段目および3段目が他のユーザ情報である。他のユーザのスケジュール情報は後述のフィルタリングアルゴリズムによってコントロールされた結果のみが表示される。なお、同時に閲覧したいユーザ数が多い場合にはそれに応じて各行の高さが変わる。

下段はユーザがカーソルで指定した日付に含まれるタスク情報のタイトルおよび時間が表示される。他ユーザの情報が公開されている場合にはそれについても同様に表示される。ユーザはここで表示される各タスクをカーソルで指定し、さらに詳細な情報を表示させることが可能である。

##### ● タスクの依頼

ユーザが複数人との共同作業を行いたい場合には、まず前述の時系列インターフェイス上に相手のスケジュール情報を表示させる。次に、共通の空き時間検索機能呼び出すと、ソフトウェアは取得したデータから該当する時間帯を検索し、候補を提示する。ユーザがこの候補から適切と思われる時間帯を選択すると、図2右に示すタスク詳細設定インターフェイスが表示される。ユーザはこのインターフェイスを用いてタイトル等の情報を入力する。また、時間帯選択の時点で開始時間、終了時間およびタスクを実行するメンバー情報は自動的に入力されているが、これを手動で変更することが可能である。ただし、提示された空き時間以外の時間帯にタスクが登録された場合には警告が発せられる。全てのデータを入力し、登録すると、端末はサーバにアクセスし、他のユーザのデータに対して書き込み処理を行う。これとともに、端末内に保存されたデータを更新する。

##### ● タスクの受理と実行

共同作業タスクがサーバに登録されると、サーバは他のユーザに対してその旨をメールにて通知する。この通知を受け取ったユーザは端末側ソフトウェアからデータ同期機能呼び出し、サーバ上のデータを取得する。その際に、新しいタスクの詳細が表示され、受理または拒否を選択する。全てのユーザが受理した場合には、そのタスクは実行可能タスクとして各ユーザのデータに正式に登録される。また、1人以上のユーザが拒否した場合にはタスクはキャンセルされ、各ユーザにその旨が通知される。タスクの依頼者は同様の作

(注1): 空き時間とはタスクが存在しない時間帯を示し、提案システムによって自動的に計算される。なお、夜間や休日といった、ユーザがタスクの登録を拒否したい時間帯については、これらを空のタスクとして

一括登録するインターフェイスを用意し、入力の支援を行っている。

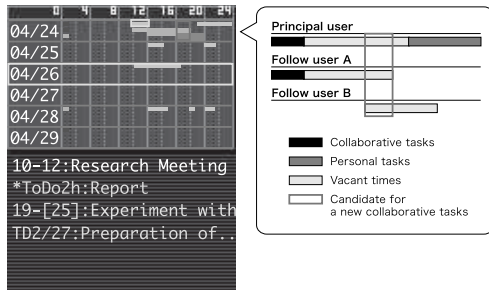


図 1 協調タスク表示  
Fig. 1 Collaborative task view.

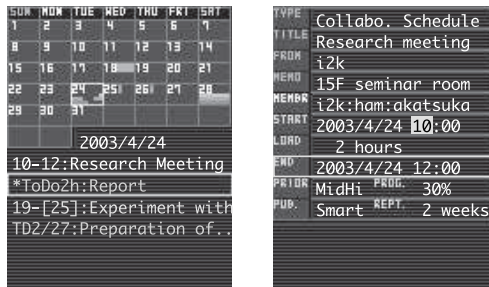


図 2 カレンダー表示 (左) とタスク詳細設定 (右)  
Fig. 2 Calendar view (left) and task data view (right).

業を繰り返す必要が生じるが、その際には 1 度拒否された時間帯は提示されないようになる。

以上のプロセスによって登録されたタスクの一覧は図 2 左に示すカレンダーインターフェイスで閲覧することができる。カレンダーインターフェイスではユーザ自身のタスクのみが表示される。カレンダーの各セルの下部は 3 段の簡易ガント表示になっており、タスクの登録状況を直感的に知ることが可能である。また、インターフェイスの下部は時系列インターフェイスと同様に、カーソルで指定した日付に含まれるタスクの詳細が表示される。

このように、提案システムはユーザに対して他のユーザの情報の取得、閲覧および共同作業タスクの登録作業をシームレスに支援する。ユーザは後述の人間関係ネットワークを用いた情報のアクセスコントロールといった機能の存在を意識することはない。また、事前の相互承認を除いて、こういった機能のための特別な設定は必要ない。ユーザは、日常的な行動を提案システム上で行うことにより、プライバシーを保護したうえで協調的なタスク登録を行うことが容易になる。

### 3.2 人間関係ネットワークを用いたアクセスコントロール

前節で述べたスケジュール決定プロセスを支援するため、提案システムは人間関係ネットワーク<sup>(注2)</sup>を用いた情報のアクセスコントロールを行う。この手法はユーザに意識されることなく実行される。

本研究で想定している個人が複数のグループに所属しているような状況では、互いの全ての情報を開示することは難しい。そこで、タスク情報の公開ポリシーを個別に設定することを考える。あるタスクの情報は、そのタスクの共同作業者の全員が属するグループのメンバーのみに公開するものとする。例えば、 $G = \{A, B, C, D\}$  となるグループ  $G$  があり、タスク  $T$  の共同作業者  $C_T$  が  $C_T = \{A, B, C\}$  であれば、このタスク  $T$  は  $G$  のメンバー全員に公開される。

この手法はユーザが所属するグループを自動的に発見するアルゴリズムおよび発見された複数のグループから適切にアクセスコントロールを行うアルゴリズムから構成される。詳細を以下に述べる。

#### 3.2.1 人間関係ネットワークからのグループ発見

本研究が対象とするような状況においては、ユーザが所属するグループは動的に変化する。すなわち、時間の進行にしたがって新たなグループが生起する。このような場合はユーザによる設定はユーザにとって負担であるだけでなく設定が不正確になる場合も考えられる。そこで、本研究ではグループを自動的に発見する手法を提案する。

この手法では、過去のタスクの依頼および受理関係に基づく人間関係ネットワークを形成し、その中で有意な部分を抽出して、これをグループとして用いる。ここでは有意な部分とは人間関係ネットワークをグラフとしてみた場合の極大完全部分グラフである。

ユーザ間での相互承認およびタスクの依頼といったインタラクションが発生すると、システムはそれらのユーザ同士を人間関係ネットワークに登録する。人間関係ネットワークはユーザごとに構築されるものとし、それぞれは図 3 に示すようにユーザ本人を中心としたスター型のトポロジーとなる。また、各リンクには「Authorized (相互承認済み)」もしくは「Collaboration (タスク依頼関係あり)」のラベルが付加される。

(注2): 本研究における人間関係とは、タスクの依頼および協働関係に基づくユーザ間のネットワークを意味する。

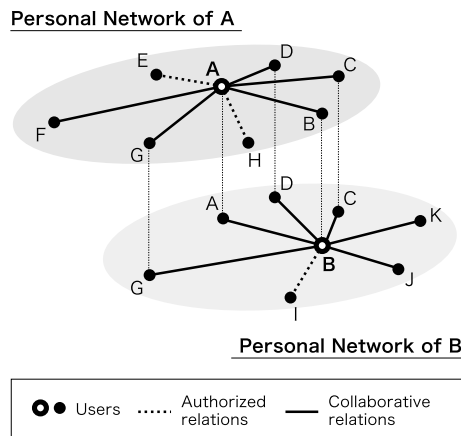


図 3 ユーザごとの人間関係ネットワーク  
Fig. 3 Personal networks of each user.

あるユーザ ( $self$ ) が他のユーザ ( $target$ ) のスケジュール情報を閲覧する際に、システムはこれらの人間関係ネットワークから両者がともに所属しているグループを発見する。グループ発見アルゴリズムの手順を図 4 に示す。まず、 $self$  と  $target$  の 2 名を最小のグループとして登録する。次に、 $self$  がパスを持つ任意の 1 ユーザ ( $candidate$ ) を選択し、グループ内の各ユーザに対して  $candidate$  へのパスが存在するかどうかを問い合わせる。全てのユーザがパスを持っていた場合には、 $candidate$  をグループに追加し、このグループをリストに登録する。登録の際には、すでにリストに存在するグループと構成メンバーを比較し、前者が後者を包含する場合には後者をリストから削除する。これらの操作を再帰的に繰り返し、最終結果をグループリストとして出力する。なお、 $self$  が複数人のデータを取得する際には、それぞれのユーザごとに複数回繰り返す。

本手法の計算量は、 $self$  と  $target$  の両者がともにパスを張っているユーザ数  $N$  の増加にしたがって  $2^N$  のオーダーで増加する。 $N$  の最大値は、 $self$  および  $target$  が持つパスの数をそれぞれ  $n_{self}$ ,  $n_{target}$  とすると、 $\min[n_{self}, n_{target}]$  となるが、一般的に、1 人が持つパスの数は、システム全体のユーザ数には比例せず、一定の数で飽和する<sup>(注3)</sup>。よって、計算量が全体のユーザ規模にしたがって無限に増加することはない。また、同定されるグラフの規模が比較的小さいと予想

(注3): 社会学の分野での調査 [7] によれば、1 人が持つ知人の数は平均 40 人程度であるとの報告がある。

```

procedure FindMaximalPartialGraph( $self, target$ )
begin
   $g \leftarrow \{self, target\}$ 
   $FoundGroups \leftarrow \{g\}$ 
   $cl \leftarrow \langle self \text{ が持つ Collaboration リンクのユーザリスト (target を除く) } \rangle$ 
  for each  $x \in cl$  do
    FindPartialGraph( $g, cl-x, x$ )
   $\langle FoundGroups \text{ を出力} \rangle$ 
end

procedure FindPartialGraph( $group, clist, candidate$ )
begin
  if  $\langle group \text{ の全ての要素が } candidate \text{ への Collaboration リンクを持つ} \rangle$  then
    begin
       $\langle candidate \text{ を } group \text{ に追加} \rangle$ 
      AddGroup( $group$ )
      for each  $x \in clist$  do
        FindPartialGraph( $group, clist-x, x$ )
    end
  end

procedure AddGroup( $group$ )
begin
  for each  $g \in FoundGroups$  do
    begin
      if  $group \supseteq g$  then  $\langle g \text{ を } FoundGroups \text{ から削除} \rangle$ 
      if  $group \subseteq g$  then exit
    end
   $\langle group \text{ を } FoundGroups \text{ に追加} \rangle$ 
end

```

図 4 グループ発見アルゴリズム  
Fig. 4 Group detection algorithm.

されることから、深さ優先の探索手法を取り入れ、早い段階での枝刈りを可能にしている。しかしながら、大規模な実運用に際しては、極大完全部分グラフの探索の高速化手法 [8] などを取り入れる必要があると思われる。

### 3.2.2 グループによるアクセスコントロール

次に、上で求めたグループを用いて、ユーザ  $self$  によるユーザ  $target$  のスケジュール閲覧の可否を決定する。この場合、スケジュール閲覧の可否とはユーザの持つ個々のタスクの詳細を表示するか否かを指す。

前節の手法により、 $self$  および  $target$  を含むグループ集合  $FoundGroups$  を得ることができる。ユーザの環境によっては複数のグループが発見される可能性がある。次に、システムは  $target$  の持つ全てのタスクより  $self$  に公開可能なものを選択する。選択アルゴリズムを図 5 に示す。

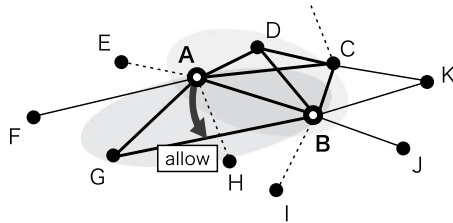
いま  $self$  をユーザ  $A$ 、 $target$  をユーザ  $B$  とした場合に発見されたグループを  $G_1 = \{A, B, C, D\}$ 、 $G_2 = \{A, B, G\}$  とする。

あるタスク  $T_1$  に含まれる共同作業者のリストを  $C_1 = \{B, G\}$  とする。このタスクについて  $C_1 \subseteq G_2$  が成立するため、タスク  $T_1$  はユーザ  $A$  および  $B$  がともに所属するグループ内で行われるタスクであるとして  $A$  に公開される (図 6(a)) 一方、 $C_2 = \{B, K\}$  で

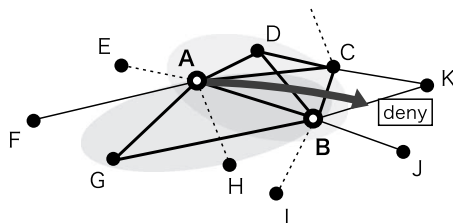
```

procedure FilterTask (self, target, FoundGroups)
begin
  tasklist ← <target が持つタスクのリスト>
  for each task ∈ tasklist do
    begin
      for each group ∈ FoundGroups do
        begin
          if <group のメンバーリストが task の共同作業リストを包含する> then
            <task を self に公開>
          end
        end
      end
    end
  end
end
    
```

図 5 アクセスコントロールのアルゴリズム  
Fig. 5 Task filtering algorithm.



(a) Acceptable access in the working group



(b) Unacceptable access

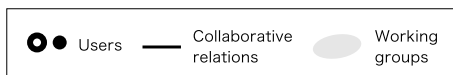


図 6 グループ発見を用いたタスク情報のアクセスコントロール  
Fig. 6 Information control method with working groups.

あるようなタスク  $T_2$  においては  $C_2 \not\subseteq G_1, G_2$  であるために公開されない (図 6(b))

以上で述べた手法により、過去の協働関係からのユーザ間のネットワークの構築およびグループの発見、グループ内のアクセスコントロールを自動的に行うことが可能になる。ユーザはグループの構成員をあらかじめ設定するなどのプロファイリングの必要がない。他人へのタスクの依頼を行う操作のみで複数のグループ間にまたがる情報のアクセスコントロールが実現される。

なお、ユーザは個々のタスクに対し、この手法を利

用したアクセスコントロールを適用する、全てのユーザに公開する、全てのユーザに公開しない、という 3 段階の情報公開レベルを設定することができる。

#### 4. 携帯電話への実装

個人の日常的なスケジューリングを支援するためには、利用シーンに限られる PC よりも、常に持ち歩くデバイスが適していると考えられる。また、複数人での協調的なタスク登録作業を支援するためには通信機能が必須である。そこで、本研究では提案手法を携帯電話上で動作するソフトウェアを用いたサーバ・クライアントシステムとして実装した。これにより、全てのユーザが場所および時間を問わず互いのスケジュール情報を閲覧し、タスクの依頼を行うことができるようになる。

一方で、クライアントに携帯電話を利用することによる制約も大きい。現状の携帯電話環境では端末同士の直接通信は不可能であるため、インターネット上のサーバに各ユーザのタスク情報を格納している。同様の情報は各端末内にも保存されており、データの同期はユーザが明示的に指定した時点で行われる。相互承認ネットワークやタスクの依頼関係ネットワークはサーバ上でのみ保存される。また、端末へのプッシュ型通信が行えないことから、他のユーザが登録した共同作業タスクの存在を知るための時間差が生じる。そこで、提案システムでは携帯電話のメール機能と連携し、サーバがタスクを登録した時点で共同作業員に対してその旨を伝えるメールを送信する。メールにはタスク情報の概要が記載されているほか、埋め込まれたリンクからクライアントソフトウェアを起動することができる。

インターフェイスの制約としては、表示画面の小ささや解像度の低さが挙げられる。情報の一覧性の低さはこれまでに述べた協調的なタスク登録プロセスを妨げる原因となるため、既存の携帯電話用スケジューラと比較して 1 画面中の情報量を高められるよう、カレンダーインターフェイスおよび時系列インターフェイスを設計している。また、携帯電話の機種によって解像度が異なる問題に対しては、機種ごとのプロフィールを作成し、ソフトウェアのダウンロード時に機種を判別して適切なプロフィールを送信する。

各クライアントはインターネット接続が可能な携帯電話である。これらの端末は Java アプリケーションのダウンロードおよび実行が可能である。本プロジェク



トでは株式会社NTTドコモが提供する端末(ソニー株式会社製 SO504i, 日本電気株式会社製 N504iS, および松下電器産業製 P504iS)を利用した。ソフトウェア開発環境としてはサン・マイクロシステムズ株式会社が提供する Java 2 Platform Micro Edition Connected Limited Device Configuration (J2ME CLDC)[9] および J2ME Wireless SDK for DoJa [10] を利用した。

サーバは常時インターネットに接続されている。ハードウェアには一般的な PC サーバを用いた。オペレーティングシステムには Debian/GNU Linux 3.0 を使用し, Java によるアプリケーションサーバ実行環境としてサン・マイクロシステムズ株式会社が提供する Java 2 Platform Enterprise Edition (J2EE) 1.2, Java 2 Standard Edition Runtime Environment (J2SE JRE) 1.4, および Apache Project が提供する Tomcat 4.0 を利用した。

## 5. 実証実験

提案システムの有効性を検証するために実証実験を行った。期間は 2003 年 1 月 26 日から 2 月 17 日までの約 3 週間である。被験者は 9 名(総クライアント数は 16)である。実験内容は、期間中にクライアントソフトウェアを実際に利用し、被験者自身のタスクを入力してもらうというものである。なお、被験者のグループ構成は、A 研究グループに属するのが 3 名, B 研究グループに属するのが 2 名, C 研究グループに属するのが 5 名, C 研究グループの OB (D グループとする) が 3 名, それら以外に個人的な友人関係が含まれる。

実験期間中に登録された個人タスクの平均は 73.3 件であった。また共同作業タスクは計 62 件であった。この協調タスク・依頼タスクは平均 3.0 名に関係しており、個人タスクの 28.3% を占める。また、共同作業タスクの入力支援機能により、1 名あたり 13.9 件のタスク入力の手間を軽減している。

### 5.1 グループ発見アルゴリズムの評価

実証実験で得られたタスクの依頼関係ネットワークを図 7 に示す。このようにタスクの依頼関係は極めて複雑なものになる。このネットワークに対し前述のグループ発見アルゴリズムを適用すると図 7 (a) の楕円に示すような複数のグループが得られる。また、特定のユーザを含むグループの一覧を図 7 (b) に示す。このように、提案手法によってあるユーザが複数のグループに所属していることがわかる。ここで、提案手

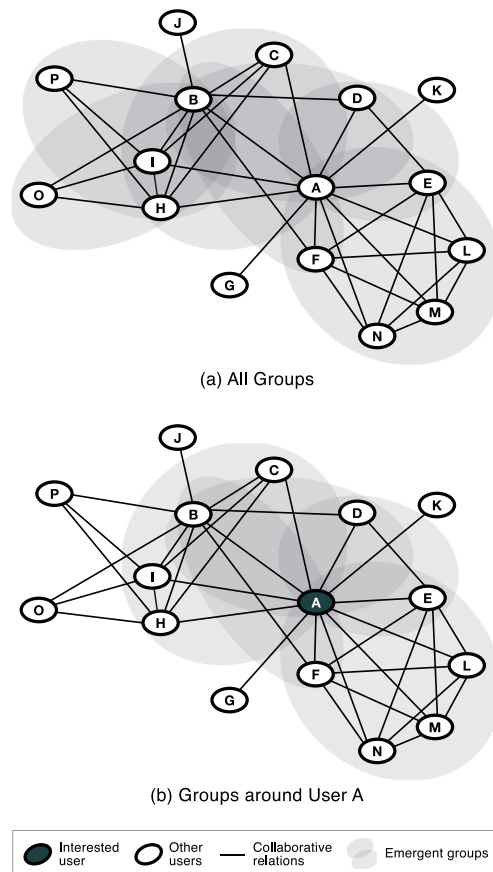


図 7 人間関係ネットワーク  
Fig. 7 Social network.

法の有効性を検証するため、被験者のそれぞれについて本人を含むグループのリストを提示し、それらのグループが適切であるかどうかを評価してもらった。この実験では、個々のグループに名前をつけてもらい、正しくつけられたものを適切であるとし、リストに含まれる名前が多すぎる、もしくは少なすぎると指摘されたものについては発見されたグループが適切でないとした。

被験者に提示したグループはのべ 24 個(計 9 グループ)である。うち有効回答は 21 個(8 名分)である。結果は 20 個(95.2%)のグループについて適切であるとの回答を得た。適切でないと言われたグループについては構成員の数 1 名足りないと指摘された。また、被験者の 1 人は、提示された 5 名を含むグループのうち 3 名で構成されるもうひとつのグループが存在すると回答した。このような包含関係にあるグループの

発見は現状のアルゴリズムでは解決できない．タスク依頼の頻度によるクラスタリングといった新しい手法が必要になると考えられる．

提案システムによって発見された 9 グループの中には、被験者のメンバー選定の段階では考慮していなかったグループ (3 名) が含まれる．このグループに所属する全員が適切であると回答していることから、本システムのグループ発見アルゴリズムは有効に機能しているといえる．

## 5.2 情報フィルタリングの評価

次に、情報フィルタリングの効果を検討するため、全ての被験者に対してそれぞれが入力したタスクの一覧およびそれらのタスクがどのユーザに閲覧可能な状態であったかをリスト化したものを提示した．その際には結果に対して同意または非同意のチェックを入れてもらい、備考として「1. 誰にも見せなくなかった」「2. 一部のユーザだけに見せたかった」「3. 全員が見てもよかった」「4. もっと見せたいユーザがいた」「5. 全員に見せたかった」という 5 種類の選択肢を用意した．各タスクに対するチェックは必須とし、備考は前述の質問で「非同意」であったものは必須「同意」であるものは任意で付加してもらうものとした．

被験者に提示したタスクの総数は 660 件である．うち 604 件 (8 名分) について回答を得た．結果を表 1 に示す．必須項目の「結果に対する同意・非同意」については「同意」が 484 件 (80.1%)、「非同意」が 120 件 (19.9%) であった．「非同意」のうち、その理由として備考の「1. 誰にも見せなくなかった」を挙げたものは 96 件である．これは有効回答全体の 15.9%、「非同意」の 80.0%を占める．本システムでは、4.1 節に述べたように各タスクに対して情報の公開レベルを「全てのユーザに公開しない」に設定することができる．そのため、これらのタスクについてはシステムとして問題の解決が容易である．同様に、非同意の理由が「3. 全員が見てもよかった」であった 13 件 (全体の 2.2%、「非同意」の 10.8%) についても情報公開レベルを「全てのユーザに公開する」とすることで解決できる．しかしながら、非同意の理由が「2. 一部のユーザだけに見せたかった」すなわち閲覧可能なユーザが多すぎるとされる 11 件 (全体の 1.8%、「非同意」の 9.2%) については、前節のグループ発見アルゴリズムに関する問題点と同様に別の手法によるフィルタリングや手動設定が必要になると思われる．これらの解消は今後の課題である．

表 1 フィルタリング手法の評価

Table 1 Evaluation of information filtering.

	Agree	Disagree
Evaluation	484 (80.1%)	120 (19.9%)
Remark 1	0	96 (15.9%)
Remark 2	0	11 (1.8%)
Remark 3	159	13 (2.2%)
Remark 4	30	0
Remark 5	48	0
None	247	0

一方、極大完全部分グラフの抽出によるグループ発見手法においては、所属するユーザの多いグループを正確に抽出することが難しい．そのため、システム全体のユーザ数が増加するにつれ、「非同意」であった理由が「4. もっと見せたいユーザがいた」となるようなタスクが増加すると予想される．このような問題を解決するためには、グループ発見の条件を緩和する必要がある．ただし、条件の緩和はユーザの予期しない情報公開につながる恐れがあるため、これらのトレードオフを考慮した手法を導入する必要がある．これについては今後の課題とする．

全体としては、「同意」もしくは解決が容易な「非同意」タスクは 593 件、有効回答全体の 98.2%となっており、提案手法によるスケジューリング支援は有効に機能しているといえる．

## 6. 関連研究

本研究に関連のある研究分野としては、CSCW やグループウェアといった知識共有に関するもの、インターフェイス、P2P ネットワーク、セキュリティ等が挙げられる．

IKNOW は個人の持つ知識を用いて個人同士の社会的ネットワークを構築するためのシステムである [11]．IKNOW では個人を Web ページ、知識を Web ページ内のリンクとして扱い、同じリンクを持つ Web ページ同士に人間関係リンクを張る．ユーザは、結果として得られた知識ネットワークを利用して知識を検索することが可能になる．IKNOW では人間関係ネットワークが誰にでも参照可能であるとしている．一方で本研究ではパーソナルネットワークには複数のコミュニティが存在し、それぞれは競合関係にあるとの想定をしている．そのため、同様の社会ネットワークが構築されたとしても、本研究と IKNOW では情報の提示方法が異なる．

Tyler らの電子メールを用いた研究では大規模な組



織内での複数のコミュニティの発見手法を提案している [12]。この手法はボトムアップのコミュニティ発見アルゴリズムを複数回試行し、同定された確率から各個人のコミュニティへの寄与度を測定するものである。しかしながら、この手法の目的はコミュニティのクラスタリングであり、本研究が目指すような 1 人が複数のコミュニティに所属しているといった状況を想定してない。また、Tyler らの手法ではあらかじめ大規模な組織が存在している場合でのみ適用することができる。

乃村らの MHC は電子メールからイベント情報を抽出し、その時間データをカレンダーに自動登録するシステムである [13]。電子メールによるタスク情報の流通は本研究で用いている携帯電話アプリケーションよりも簡便であり、潜在的なユーザ数も大きい。しかし、MHC では複数人の中での情報公開や協調的なタスク登録といった問題を取り扱っていない。

## 7. ま と め

本研究では個人のリソース管理の手法として協調モデルを提案し、携帯電話用タスクスケジューラとしてこのモデルを実装した。

本研究が対象としているのは会社組織のようにあらかじめ上下関係が定められておらず、その範囲が明確でないようなグループに複数所属しているような人々である。

提案システムでは、既存のグループウェア等とは異なり、メンバーリストの作成や管理者の決定といったグループの定義をあらかじめ行う必要はない。ユーザは、他のユーザに対して協調・依頼タスクを発行するという操作のみを行い、そこからの関係の発見やグループの同定はすべてサーバ側が自動的に行う。これにより、ボトムアップに構築されるグループや、タスクの発生とともに生まれるアドホックなグループに対しても適切な支援が可能になる。

提案システムでは、各ユーザがあらかじめ自身のリソース情報を公開することを前提としている。このような環境では、他人の時間リソースの状況を考慮した上でリソースの予約をすることが可能になるために、予期しないリソースの競合を引き起こすことが少なくなる。また、競合が実際に発生したとしても、交渉のサイクルが短くなると思われる。

過剰な情報公開によるプライバシーの侵害に対しては、タスク情報のアクセスコントロール機能によって

望まない情報公開や全ての認証ユーザに全ての情報を公開してしまうといった状況を回避することが可能である。実際には、タスクの依頼関係からグループを同定し、グループを横断するような情報アクセスを禁ずる。これにより、プライバシーの侵害を最小限に抑えた上で、情報共有によるメリットを享受することが可能になる。

提案システムでは、1 人のクライアントソフトウェアから他の複数のユーザの持つタスクデータに直接タスク情報を書き込むことが可能である。他ユーザはこのタスク情報を自分で入力する必要がないため、トータルで見るとデータ入力のコストは大幅に低減する。携帯電話のアプリケーションを導入する上で阻害要因となるのはデータ入力の手間であることが多い。このような協調モデルには携帯電話用アプリケーションのデメリットを打ち消す効果があると考えられる。

現状のシステムでは、ユーザ間に張られたパスを消去しないために、発見されるグループの数が単調増加する。今後の課題として、タスクの依頼関係パスに時間の要素を導入し、一定期間タスクのやりとりのないユーザ間の関係を消去することや、パスの時間的要素を考慮してアクティブなグループのみを発見する手法 [14] を導入することを考える。

謝辞 本研究は情報処理推進機構 (IPA) による「平成 14 年度末踏ソフトウェア創造事業」の支援を受けております。

## 文 献

- [1] C. A. Ellis, S. J. Gibbs and G. Rein: "Groupware: Some Issues and Experiences", *Communications of the ACM*, **34**, 1, pp. 39-58 (1991).
- [2] B. Wellman: "Computer Networks As Social Networks", *Science*, **293**, pp. 2031-2034 (2001).
- [3] I. Ohmukai, H. Takeda and M. Miki: "A Proposal of the Person-centered Approach for Personal Task Management", *Proceedings of 2003 Symposium on Applications and the Internet (SAINT2003)*, pp. 234-240 (2003).
- [4] S. J. Noronha and V. V. S. Sarma: "Knowledge-Based Approaches for Scheduling Problems: A Survey", *IEEE Transactions on Knowledge and Data Engineering*, **3**, 2, pp. 160-171 (1991).
- [5] M. Klusch: "Information Agent Technology for the Internet: A survey", *Data and Knowledge Engineering*, **36**, (2001).
- [6] 亀井, 藤田, E. Jettmar, 吉田, 桑原: "ネットワークコミュニティの形成を支援するシステム Community Organizer における情報提示手法の検討", *電子情報通信学会論文誌, J84-D-I*, 9, pp. 1440-1449 (2001).

- [7] 工藤, 大谷: “数量として測る友人数”, <http://www-soc.kwansei.ac.jp/otani/research/02sotsuron/3-3.htm> (2000).
- [8] 富田, 平賀, 若月: “最大クリーク抽出アルゴリズムの効率化とその評価”, 情報処理学会研究報告, 99-MPS-24-1, pp. 1-4 (1999).
- [9] Sun Microsystems, <http://jcp.org/aboutJava/communityprocess/final/jsr139/>: “Java 2 Platform Micro Edition Connected Limited Device Configuration (J2ME CLDC) Specification” (2003).
- [10] NTT DoCoMo, [http://www.nttdocomo.co.jp/p\\_s/imode/java/pdf/jguide504\\_021216.pdf](http://www.nttdocomo.co.jp/p_s/imode/java/pdf/jguide504_021216.pdf): “i アプリコンテツ開発ガイド for 504i” (2002).
- [11] N. S. Contractor, D. Zink and M. Chan: “A Tool to Assist and Study the Creation, Maintenance, and Dissolution of Knowledge Networks”, Community Computing and Support Systems, pp. 201-217 (1998).
- [12] J. R. Tyler, D. M. Wilkinson and B. A. Huberman: “Email as Spectroscopy: Automated Discovery of Community Structure within Organizations”, Condensed Matter (2003).
- [13] 乃村, 花田, 牛島: “MHC - Message Harmonized Calendaring System の設計と実装”, 情報処理学会論文誌, 42, 10, pp. 2518-2525 (2001).
- [14] R.Kumar, J.Novak, P.Raghavan and A.Tomkins: “On the Bursty Evolution of Blogspace”, Proceedings of the Thirteenth International World Wide Web Conference (WWW2003) (2003).

(平成 xx 年 xx 月 xx 日受付)

#### 大向 一輝



2000 年 3 月同志社大学工学部知識工学科卒業。2002 年 3 月同大学院工学研究科博士前期課程修了。同年 4 月総合研究大学院大学複合科学研究科博士後期課程入学。現在に至る。セマンティック Web 技術を用いた知識共有の研究に従事。情報処理学会, 人工知能学会各会員。

#### 武田 英明 (正員)



1991 年 3 月東京大学大学院工学系研究科博士課程修了。1993 年 4 月奈良先端科学技術大学院大学助手。1995 年 4 月同助教授。2000 年 4 月国立情報学研究所助教授。2004 年 5 月同教授。現在に至る。総合研究大学院大学教授(複合科学研究科情報学専攻)を併任。特に知識共有, ネットワークコミュニティ, 実世界エージェントなどの研究に従事。AAAI, 人工知能学会, 情報処理学会など会員。工学博士。

**Abstract** This paper proposes a collaborative personal task management with social network. Social network is obtained from analysis of requested task among people then our system discovers emergent groups in the network. These groups are used to control information disclosure, that is, only group members can access detailed information belonging to the group. This approach maximizes usability with keeping users' privacy. We implement client/server system on cellphones environment called Social Scheduler. We remark the advantages of our approach with an experiment.

**Key words** collaborative problem solving, resource management, support for creating communities, social network, cellphone application