空き時間とタスク間関係を利用したユーザのスケジューリング支援手法

堤 大輔[†,☆] 倉本 到[†]
渋谷 雄[†] 辻野嘉宏[†]

本論文では，既存のスケジューラシステムがかかえる問題を解消するため，ユーザが自由に使用できる時間である「空き時間」の概念を導入したタスク・スケジュール管理，タスクの階層構造と実行順序関係からなる「タスク間関係」に基づいたタスク管理の 2 つの手法を提案した．そして，2 つの提案手法の機能を付加したスケジューラシステム「タイムラインナビ」を設計・実装し，提案手法の有用性を評価するための実験を行った．その結果，空き時間表示がユーザのスケジューリングを効率的かつ容易にし，タスク間関係に対応したタスク管理が，かかえているタスクの現在の状態をより把握しやすくすることが分かった．

Scheduling Support Methods Using the Amount of Free Time and Relationship among Tasks

Daisuke Tsutsumi,[†,☆] Itaru Kuramoto,[†] Yu Shibuya[†] and Yoshihiro Tsujino[†]

In general scheduler systems, tasks and schedules are managed independently though they are actually related. So it is difficult for users to notice that their scheduling is overbooked. In addition, users cannot explicitly grasp a state of their tasks, such as a progress of each task because they have no chance to know such state with the scheduler systems. In order to solve these problems, we propose a task-and-schedule management method based on the amount of free time and a task management method based on the relationship among tasks. Furthermore a scheduler system with the proposed methods, named “Time Line Navi”, was implemented and evaluated experimentally. The results show that the free time is useful for the user’s scheduling and the relationship among tasks enable users to manage tasks clearly and effectively.

1. はじめに

個人にとって，時間をいかに管理していくかは重要な課題である．そのため，我々は手帳やカレンダーを用いて個人の予定を管理している．また近年，計算機利用環境の発展にともない，予定を管理するためのソフトウェアとしてスケジューラシステムの利用も増加している．

本論文では，まず，既存のスケジューラシステムがかかえている問題点について詳細に議論する．次に，それらの問題点を解消するために，ユーザが自由に使用できる時間である「空き時間」を利用したタスク・スケジュール管理手法と，タスクの階層構造と実行順序関係からなる「タスク間関係」に基づいたタスク管理手法を提案する．そして，提案手法の機能を付加したスケジューラシステムである「タイムラインナビ」を設計・実装し，評価実験によりユーザのスケジューリングにおける空き時間表示の有用性と，タスク間関係に基づいたタスク管理の有用性を評価する．

† 京都工芸繊維大学
Kyoto Institute of Technology
☆ 現在，京セラコミュニケーションシステム株式会社
Presently with KYOCERA Communication Systems Co., Ltd.

2. スケジューラシステム

2.1 タスクとスケジュール

タスクとスケジュールは，いずれもユーザが後に実行するためにあらかじめ定めた予定であるが，本論文ではこれらを開始時刻の有無，中断の可能・不可能から定義する．

開始時刻が決まっており，かつ，中断不可能であれ

ば，時間的な拘束が生まれる．そこで，このような予定をスケジュールと定める．スケジュールの例としては，会議や講義などがある．

一般にタスクとは，ユーザが任意の時刻に開始できる仕事・作業のことである．そこで本論文では，開始時刻が決まっているかどうかにかかわらず途中で中断可能な予定，および中断不可能ではあるが開始時刻が決まっていない予定をタスクと定義する．前者の例としては，レポートの作成やプログラミングなどがある．後者の例としては，一度始めると途中で止めることができない化学実験などがある．

なお，スケジュールの終了時刻やタスクの締切り時刻の多くは決定している．しかし，終了時刻の明確でないスケジュールや，締切りの定められていないタスクも存在する．本論文では，前者はおおよそ終了時刻を推定して通常のスケジュールとして扱う．後者はタスクの存在を記録しておくことしかできないので，管理対象としては扱わない．

ところで，一般的にユーザがスケジューラシステムを利用する環境において，これらのタスクやスケジュールが静的に決定されることはあまりない．これは，ユーザのスケジューリングに以下のような状況が高頻度で発生するためである．

1. タスクやスケジュールの詳細は，時間の経過にともない次第に明らかになってゆくことが多い．
2. ユーザの作業に対する見積りは強く束縛されたものではないので，ユーザはしばしば最適なタスク・スケジュール配置を得るためにそれらを割り当て直す．
3. 開始時点で明らかでなかった作業や，追加発生したタスクやスケジュールが優先的に割り込んでくることがある．

スケジューラシステムは，これらの状況にすばやく対応できるのが望ましい．ところが，既存のシステムの多くは，これらの状況に十分対応できていない．次節で，既存システムの問題をスケジューラシステムとプロジェクト管理システムの 2 つのシステムの側面から検討する．

2.2 既存システムの問題

2.2.1 スケジューラシステム

既存のスケジューラシステム[1),2)] の多くはユーザのスケジュールを管理する部分と，To-Do リストなどのタスクを管理する部分から構成されている．タスクには一般に締切りがあるが，それをスケジュール化するためには，いつそのタスクを実行するかを決定しなくてはならない．しかし，2.1 節の 2. および 3. で述べたように，タスクは日常的に頻繁に発生し，変更されるため，そのつどタスクをスケジュール化するだけの時間を確保することは難しい．したがって，タスクを忘れないために To-do リストに登録することになる．この結果，実際には多くのスケジュールが予定されており，タスクを行う十分な時間がないにもかかわらず，そのことに気づかず新たなタスクを追加する，また，多くのタスクをかかえているにもかかわらず，新たなスケジュールを追加するという実行しきれないタスクをかかえてしまうリスクが発生する．

また，タスクの中には，いくつかのサブタスクに分割できる場合や，メールの到着などタスク開始のために必要な情報や資源確保が必要になる場合がある．また，それらに基づき，タスクの実行順序や優先順位が発生する場合がある．多くのスケジューラシステムでこれらの情報を登録・管理し，実行可能性や実行優先順位の理解を支援するには，タスクの重要度を手動で登録する手法が代表的である．しかし，2.1 節の 1. で述べたように，タスクが発生した時点でこれらの構造の存在に気づくことは難しいため，後にそれに気づいた時点でそれらの情報を柔軟に修正する必要があるが，既存のスケジューラによる手動での修正や構造の管理には手間がかかり，すべてのタスクにわたって確認しなおすことは困難であるため，その実効性はきわめて低いと考えられる．

2.2.2 プロジェクト管理システム

ユーザのスケジュールを管理するものとして，Microsoft Project[3)] に代表されるプロジェクト管理システムソフトウェアが実用に供されている．それらの主な機能は，

- プロジェクト作成（入力）支援：PERT 図，ガントチャート作成など
- 進捗管理支援：どのタスクが予定より遅れているかをすぐに分かるようにする
- メンバのスケジュール管理
- その他メンバに対するグループウェア的支援

であり，同時に複数のプロジェクトがあることを想定して，複数プロジェクトをまとめて管理できるものもある．この意味で，プロジェクト管理ソフトはスケジュールとタスクを同時に管理するためのものであるといえる．

しかし，プロジェクト管理のアイデアは，「プロジェクト作成時にすべてのタスクとその見込時間を同定し」それらのタスクを「いつ，だれが実行するかを決定し」そのタスクの「進捗状況をつねに監視する」というものである．本論文の用語を用いていい換えると，

タスクをスケジュール化して管理運用するということである．

プロジェクト管理でこのような管理手法が有効となるのは，一度プロジェクトが始まると，長期間それらのタスク・スケジュールを頻繁に変更することはない，という性質を持つためである．ところが，2.1 節で述べたように，個人ユーザがかかえるタスクを事前にすべて同定し，静的に配置することはその性質上困難である．

以上で述べたプロジェクト管理の性質に鑑みても，プロジェクト管理ソフトウェアでは，頻繁かつ動的な変更を簡便に行うという点は考慮されていない．そのため，プロジェクト管理ソフトウェアを単純にユーザの個人環境に適用することは難しい．

2.3 関 連 研 究

タスクやスケジュールの管理に関して様々な研究が進められている[4)~6)]．

Bellotti らは人々のタスクの管理方法やタスク管理のための資源について調査し，タスクリストマネージャの設計のための指針を示している．そして，タスクリストマネージャのプロトタイプを作成している[4)]．また，E メールクライアントでタスク管理を行うためのシステムも構築している[5)]．しかし，これらの研究ではタスク管理に関して詳しく調査しているが，スケジュールとの関係については言及していない．

また，大向ら[6)] は複数のタスクやスケジュールによってダブルブッキングが起こり，すべてを実行しきれなくなった場合に，タスクやスケジュールの重要度に応じて複数のスケジューリング案をその評価値とともに提示し，ユーザの意思決定を支援している．この研究では，すべてのタスクやスケジュールを実行しきれないことが分かってからスケジューリング案を提示しているのに対し，本論文では，新たなタスクやスケジュールの追加によりすべてのタスクやスケジュールが実行しきれなくなることをユーザに知らせることによって，ユーザ自身がスケジューリングを行うことを支援する．

3. 提 案 手 法

本論文では，前章で述べた既存システムの問題点を解消し，頻繁に変更が発生するユーザ個人環境でのスケジューリングに適した手法として，

- 空き時間を利用したタスク・スケジュール管理
- タスク間関係に基づいたタスク管理

の 2 手法を提案する．

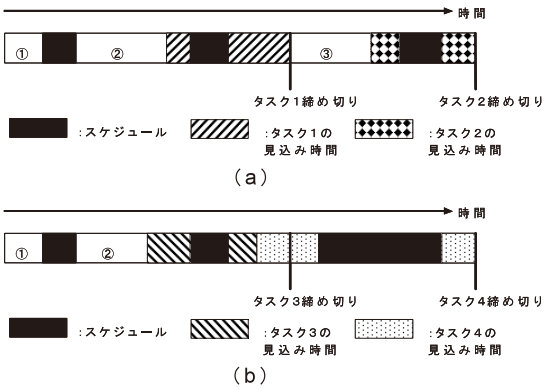


図 1 空き時間の例
Fig. 1 Examples of free time.

3.1 空き時間を利用したタスク・スケジュール管理

実行しきれないタスク・スケジュールの登録を防ぐことを目的とし，既存のスケジューラでは独立に扱っているタスクとスケジュールを統一的に扱うために，新たに「空き時間」の概念を導入する．

本論文では，空き時間を，ユーザがかかえているあるタスクの締切りまでに自由に使える時間と定義する．空き時間は各タスクごとに定まり，その値は他のタスクやスケジュールの影響を受ける．すなわち，空き時間は，現在時刻からあるタスクの締切りまでの時間から，その間に予定されている全スケジュールによる拘束時間と全タスクの見込み時間の合計時間を除くことにより求められる．ここでタスクの見込み時間とは，タスクを完了させるために必要であるとユーザが見積もる主観的な作業時間のことである．

図 1 に例を示す．図 1 (a) では，タスク 1 の空き時間は締切り日時までの時間からスケジュールの実行にかかる時間とタスク 1 の見込み時間を除いた時間であり，図の ① と ② の合計時間である．同様に，タスク 2 の空き時間は ①，②，③ の合計時間である．また，図 1 (b) のように，タスク 4 を締切りまでに完了させるためには，タスク 3 を締切りより前に完了させなければならないこともある．その場合のタスク 3，タスク 4 の空き時間はともに ① と ② の合計時間となる．

直近のタスクに対する空き時間を示すことで，ユーザは現在どの程度の時間的余裕を持っているのかを直感的に知ることができ，頻繁な変更やタスクの発生による再スケジューリングの必要性の判断をすばやく行うことができる．また，タスクの空き時間が 0 になる前にユーザにタスクの実行を促すことでそのタスクが実行できなくなることを防ぐこともできる．

さらに，新たなタスクやスケジュールの追加時に，追加後のすべてのタスクについての空き時間をシステムに計算させることにより，追加するタスクやスケジュールが実行可能であるかどうかや，追加によってすでに予定しているタスクやスケジュールが完了できなくならないかを知ることができる．これにより，ユーザはタスクやスケジュールを追加するかどうか，あるいは既存のタスクやスケジュールを変更・削除するかどうかを容易に判断できる．すなわち，ユーザは実行不可能なタスクやスケジュールの追加を防ぐことができ，効率的で無理のないスケジューリングを行うことができると考えられる．

なお，タスクの見込み時間はユーザが主観的に予測する作業時間であり，空き時間を算出するためにはユーザからタスクの見込み時間を取得する必要がある．また，見込み時間はユーザがタスクを実行するに従い減少していくものであるが，見込み時間が減少するたびにユーザに見込み時間を再入力させることは，ユーザにとって大きな負担となる．したがって，ユーザがタスクを実行していることを検知し，実行しているタスクの見込み時間を自動的に減少させる必要がある．一方，見込み時間はユーザがある時点で推測した値であるので，タスクの実行に従い，その時点の見込み時間から実際に作業を行った時間分を引いた時間と，そのときにユーザが考えている見込み時間の間にずれが生じることがある．そのため，自動減少と同時に，ユーザによる見込み時間の修正を可能にする必要がある．

3.2 タスク間関係に基づいたタスク管理

既存のスケジューラが有する，タスクの実行順序や実行可能性を柔軟に登録・管理できないという問題を解決するために，タスク間の関係に基づいたタスク管理手法を提案する．

3.2.1 タスク間関係

本論文では，タスク間関係として，階層構造と実行順序関係を導入する．

多くの場合，タスクは複数のサブタスクから構成され，それにより階層構造が生じる．また，サブタスク間には逐次実行と並行実行の 2 種類の実行順序関係がある．逐次実行とは，あるタスクの完了後初めて次のタスクが実行できるという関係である．たとえば，「データ分析」というタスクの完了後にしか「図表作成」のタスクを実行することができないなどの関係である．これに対し，並行実行とは，同時に実行可能なタスクが複数あり，それらを任意の順で実行することができるという関係である．たとえば，「論文作成」というタスクにおいて，「文書作成」と「図表作成」の 2 つのサブタスクが同時に実行可能な状態であり，どちらを先に実行してもよいなどの関係を意味する．

さらに，2.2 節で述べたように，タスク実行において，データの到着などのイベントが発生するまでタスクが実行可能とならないこともある．たとえば知人にメールで質問し，その返信があるまで次の作業が行えないなどが考えられる．

また，ユーザはタスクを進めるに従い，タスクに関するより正確な内容が分かってきたり，タスク内容に変更が生じたりする．それに従い，タスク間関係も変化していく．

3.2.2 タスク間関係に基づいた管理の効果

非定形タスクの場合，ユーザはタスクの実行を進めるに従い，具体的に何をするのか，どれくらいの時間がかかるのかがより正確に分かってくる．タスク管理においてこのことを考慮することは重要である．3.2.1 項で分析したようなタスク管理を動的にかつ簡便に行うことにより，タスクを分割できることや新たなサブタスクを追加する必要があることが分かったときや，サブタスク間に実行順序関係があることが分かったときに，それらを随時管理することが可能となる．これらの情報を用いると，現在ユーザが具体的にどのタスクを実行する必要があるか，現在どのタスクが実行可能であるかを自動的に決定することができ，それらを理解するためにそのつど多くのタスクを確認したり修正したりする必要はなくなる．

また，ユーザは必ずしも詳細なタスク管理を行う必要はなく，タスクの規模やユーザの必要性に応じて，ユーザの希望する粒度でタスク管理の細かさを決定すればよい．

4. タイムラインナビ

3 章で述べた 2 つの提案手法を導入したスケジューラシステムを実装した．以下，このスケジューラシステムをタイムラインナビと呼ぶ．

4.1 タイムラインナビのインタフェース

タイムラインナビのインタフェースは，メインウィンドウ（**図 2** 左），タスクウィンドウ（図 2 右），入力ダイアログで構成されている．また，メインウィンドウは，月表示，週表示，日表示の 3 種類の表示（図 2 は週表示）を切り替えることができる．メインウィンドウの「NEW」ボタンをクリックすることによって入力ダイアログが表示され，タスクあるいはスケジュールを入力することができる．

入力ダイアログではタイトル，スケジュール・タスク・中断不可能タスクのどれであるか，開始日時，終

図 2 タイムラインナビ（週表示）
Fig. 2 Time Line Navi (week view).

了（締切り）日時，見込み時間，種類，作業フォルダ，場所，カレンダー表示（月表示）への有無，メモ，繰返しの設定を入力することができる．なお，入力必須項目はタイトル，スケジュール・タスクのどちらであるかのみで，それ以外の決まっていない項目や必要ない項目は入力しなくてもよい．

作業フォルダとは，3.1 節で述べた見込み時間の減少を自動で行うために指定するフォルダである．ユーザはタスクを実行するときに使用するファイルやデータなどを入れておくためのフォルダをタスクごとに指定することができる．ユーザが作業フォルダをデスクトップ上で開いているとき，そのタスクを行っているものと見なし，そのタスクの見込み時間を減少させていく．なお，見込み時間はユーザの主観的な値であるので，作業をするにつれてシステムが計算した値とユーザが考えている値にずれが生じることが考えられる．また，計算機上で行われない資料の閲覧やペンによる作業などのタスクも多く存在する．これらに対応するために，見込み時間の手動での修正も可能とした．

メインウィンドウの週表示，日表示では，画面は上下に分割されており，上側にはタスク，下側にはスケジュールが表示される．さらに日表示では，表示されている各タスクごとに，現時点での空き時間，締切りまでの時間，見込み時間，締切り日時を確認することができる．

図 2 で示されているように，週表示，日表示では，スケジュールにダブルブッキングが生じていれば，そのスケジュールの表示が重なり，そのことが分かるようになっている．また，タスクの空き時間が 0 以下になった後は，どれくらい時間が足りなくなっているかを負の空き時間表示で知らせる．

表 1 タスクウィンドウの記号
Table 1 Signs of the task situation.

記号	意味
▶	現在実行可能なタスク
■	現在実行不可能なタスク
▮▮	イベント発生後に実行可能となるタスク
!	空き時間が 0 以下となっているタスク

図 3 サブタスク入力ダイアログ
Fig. 3 An input dialog window for a subtask.

タスクウィンドウ（図 2 右）は，3.2.2 項で述べたように，現在実行可能な状態のタスクとイベントの発生待ちとなっているタスクを，タスクの階層構造とともに表示し，下部に最も締切りの近いタスクの空き時間を表示するウィンドウである．表示されるタスクはメインタスクの締切りの近い順に並んでいる．タスク名の前に表示されているマークの意味を**表 1** に示す．また，ウィンドウに表示されているタスク名をクリックするとそのタスクに対する現時点での情報を示すダイアログが表示される．このダイアログでも日表示と同様に現時点での空き時間や締切りまでの時間，見込み時間，締切り日時を確認することができる．

タスクウィンドウは，ユーザの作業や時間の経過，イベントの発生に応じてつねに現在のタスク状況を表示する．タスクウィンドウはつねにデスクトップ上に表示される．したがって，ユーザはこのウィンドウによってメインウィンドウを開かなくてもかかえているタスクの状況を概観することができる．

4.2 タスク管理機能

3.2.1 項で述べたタスクの詳細管理のためのインタフェースとして，タスクの情報を示すダイアログ中の「タスク詳細管理」ボタンを押すとサブタスク入力ダイアログ（**図 3**）が表示される．このダイアログの上半分では選択したタスクの変更・確認が，下半分の表部分では選択したタスクのサブタスクの追加・変更・確認が行える．表部分に直接，タイトルや見込み時間，

締切り，イベントのうち必要な項目を直接入力することで，サブタスクの追加・変更が可能である．また，表の下にある「逐次実行」，「並行実行」のラジオボタンを選択することで，サブタスク間の関係を指定できる．さらに，サブタスクの状態や明確に指定しなかったサブタスクの締切りは，実行順序関係，開始イベントの有無，各サブタスクの見込み時間などを基に，登録時に自動的に判定・計算され，その後確認可能となる．

また，タスクの空き時間が 0 になるということは締切りまでにタスクを完了させるための時間に余裕がなくなったということを表している．そこで，ユーザのかかえているタスクの空き時間が 0 になったとき，ユーザにタスクの実行を促すためのダイアログを表示する．

5. 評価実験 I：空き時間

5.1 実験目的

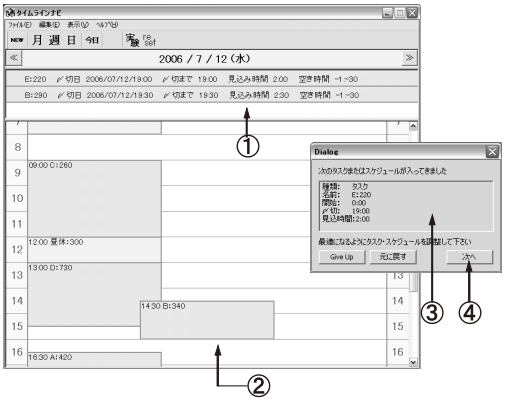
評価実験 I の目的は，空き時間表示によるユーザのスケジューリング速度と精度の変化を分析し，提案手法によりより素早く，確実に，快適にスケジューリングを行えるかを調べることである．

5.2 実験方法

被験者は 20 代の学生 16 人である．被験者には，空き時間表示のある通常のタイムラインナビ（以下，提案システム）と，空き時間表示を除いたタイムラインナビ（以下，対照システム）の両方を使用して実験タスクを行わせた．

実験画面を**図 4** に示す．具体的な実験の流れは以下のとおりである．

1. 被験者が ④ の「実験開始」ボタン（図 4 上では「次へ」となっているが，実験開始時にはこのボタンには「実験開始」と書かれている）を押すと実験が開始される．
2. ③ の新規タスク・スケジュール情報表示欄に新たなタスクまたはスケジュールの情報が表示される．その情報は，種類（タスク，スケジュール），名前（アルファベット：得点），開始時間，終了（締切り）時間，見込み時間（タスクのみ），である．
3. 被験者はこれらの情報を基に，タイムラインナビにタスクおよびスケジュールの登録を行う．なお，提案システムのみ，① のタスク表示部に表示されている各タスクに対して空き時間が表示される．
4. スケジューリングに無理が生じていないかを確認し，無理が生じている場合にはタスク・スケジュールの調整を行う．
5. ④ の「次へ」ボタンを押す．



番号	名称
①	タスク表示部
②	スケジュール表示部
③	新規タスク・スケジュール情報表示欄
④	「実験開始」／「次へ」ボタン

図 4 評価実験 I：実験画面
Fig. 4 A screenshot of experiment I.

被験者には手順 2 から 5 を繰り返し行わせた．以下では，手順 2 で表示されるタスクまたはスケジュールの情報を問題と呼び，手順 2 から 5 までの時間を問題 1 問にかかる操作時間とする．

手順 4 で生じる無理なスケジューリングとは以下の 2 つの状況である．

- 複数のスケジュールの一部または全部の時間が重なってしまう（ダブルブッキング）．
- スケジュールや他のタスクに時間がかかりすぎて，あるタスクが締切りまでに間に合わなくなる．

もしスケジューリングに無理が発生していることに気づいた場合には，無理が生じないように，タスク・スケジュールの削除によって調整を行う．このとき，各タスク・スケジュールにはあらかじめ名前とともに得点（数字 3 桁）が設定されており（例：E:290），調整は登録されているタスク・スケジュールの合計得点を最大にするように行わなければならない．なお，合計得点が最大となるパターンは 1 通りである．

被験者が正しく調整できなかった場合には，不正解であることが通知され，被験者は再びタスク・スケジュールの調整をやり直す必要がある．

実験タスクは 2 種類（Task-A，Task-B）用意した．各実験タスク全 41 問中，調整が必要なものは 12 問で，調整のためにタスク・スケジュールを 1 つ削除しなければならないものが 8 問，2 つ削除しなければならないものが 4 問である．Task-A，Task-B の調整パターンは同じにしたが，問題間の難易度の差や，慣れによる影響をなくすため，被験者 4 人ずつに，以下の 4 通りの順序で実験を行わせた．

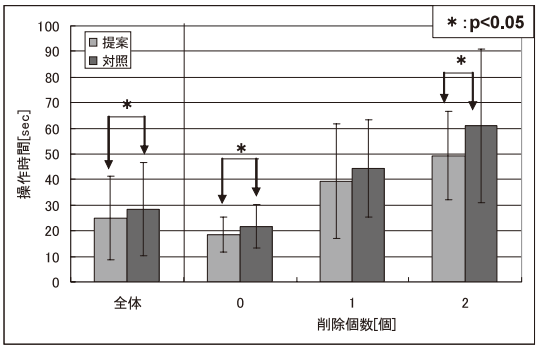


図 5 評価実験 I：操作時間
Fig. 5 Total operation time (Experiment I).

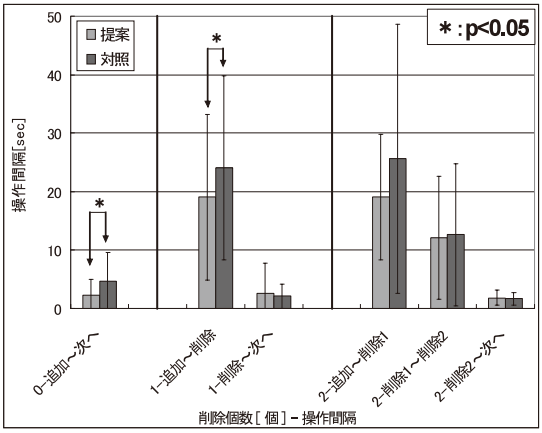


図 6 評価実験 I：各操作間隔
Fig. 6 Operation intervals (Experiment I).

- 提案システム（Task-A）⇒ 対照システム（Task-B）
- 対象システム（Task-A）⇒ 提案システム（Task-B）
- 提案システム（Task-B）⇒ 対照システム（Task-A）
- 対照システム（Task-B）⇒ 提案システム（Task-A）

また，各システムの利用を終えるたびに，アンケートに回答させた．

5.3 実験結果および考察

評価実験の結果および考察を以下に示す．評価項目は，操作時間，正答率，アンケートによる主観評価の 3 項目である．

5.3.1 操 作 時 間

図 5 は全体と削除個数ごとの平均操作時間，**図 6** は削除個数ごとの各操作の平均操作間隔を表している．横軸項目名の頭の数字は削除個数を示す．被験者は 5.2 節の手順 3. でまずタスクかスケジュールを追加する．この直後を図 6 では「追加」と表現する．それから手順 4. に移り，必要があればタスクやスケジュールを削除する．削除した直後を「削除」（削除個数 2 の場合は，第 1，第 2 のそれぞれのタスクを削除した直後を「削除 1」「削除 2」）と表現する．そして，手順 5. の「次へ」ボタンを押した直後を「次へ」と表現する．平均操作間隔とは，上記の 2 時点で定義される（図では〜記号で表現される）時間の平均である．また，同じく図 6 の削除個数とは，調整時にタスク・スケジュールを削除しなければならない個数である．なお，一度でも不正解となったデータは省いた．

図 5 より，全体として空き時間表示のある提案システムにおける操作時間は空き時間表示のない対照システムの場合よりも約 3.4 秒速く，その差は有意であった（$p < 0.05$）．

また，図 5 の削除個数 0 個のグラフは，削除するタスク・スケジュールがないことから，新たなタスク・スケジュールを追加し，スケジューリングに無理が生じていないかを確認するまでの時間（確認時間）を示しているといえる．すなわち，空き時間の表示の有無で，追加にかかる時間は変わらないと考えられるので，空き時間表示によって確認時間が短縮されていることが分かる．

この原因としては，空き時間表示のある方では表示がマイナスになっているかどうかですぐにスケジューリングに無理が生じているかどうかを判断できたのに対し，表示のない方では無理が生じていないかを確認する作業が必要であったことが考えられる．

また，図 5 より，削除個数が増加するに従って，提案システムと従来手法の操作時間差も増加していることが分かる．図 6 から「次へ」ボタンを押す直前の操作間隔は両システムで非常に短くなっていることが分かるので，この差は主に，追加して削除を行うまでの時間によるものであることが分かる．したがって，削除個数ごとにスケジューリングの確認作業の難易度に差はないと考えると，この時間差はスケジューリングの調整方法を考える時間によるものといえる．また，全体的には削除個数が増加するほど調整の難易度も上がると考えられることから，調整の難易度が上がるほど空き時間表示による効果が高くなっているといえる．

5.3.2 正 答 率

一度でも不正解のダイアログが表示された場合にその問題を不正解とし，正答率を求めた．**図 7** に正答率のグラフを示す．また，不正解がどのようなミスによるものかを以下の 2 種類に分類した．

- 確認・調整ミス：スケジューリングの確認・調整時に生じるミス
- 入力ミス：入力間違いによるミス

図 7 における全体のグラフから空き時間表示がある方が高い正答率となっていることが分かる．さらに，

分類の結果，入力ミスの数に大きな差はなく，確認・調整ミスが減少していることが分かった．したがって，正答率の観点から見ても，空き時間表示は，スケジューリングに無理が生じていないかの確認や，タスク・スケジュールの調整に有用に働いていることが分かる．ただし，削除個数が 2 の場合，提案手法の正答率が高くなっているが，その差は有意ではなく，操作時間で見られた難易度との対応関係が見られない．全体として正答率は向上しているので，空き時間表示の効果は見られたといえるが，調整の難易度と正答率との関係はさらに詳しい調査が必要である．

5.3.3 アンケートによる主観評価

図 8 にアンケート結果を示す．それぞれの項目について 7 段階で評価を行った．評価値は 4 を「どちらでもない」とし，数値が高いほど好評価である．

Q1 から空き時間表示が，タスク・スケジュールの状態の把握に有用であることが分かる．この状態の把握のしやすさが，操作時間の短縮に役立っていると考えられる．また，Q2 でも好評価を得たことからも空き時間表示の有用性が分かる．また，Q3 でも空き時間表示の有用性から提案システムの方が高評価を得たと考えられる．

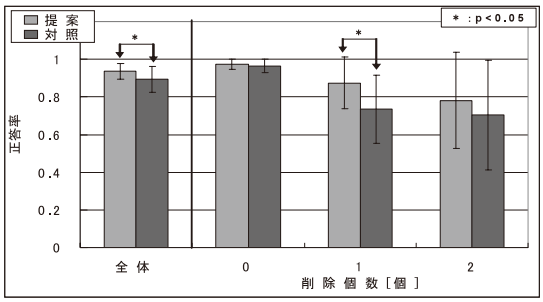
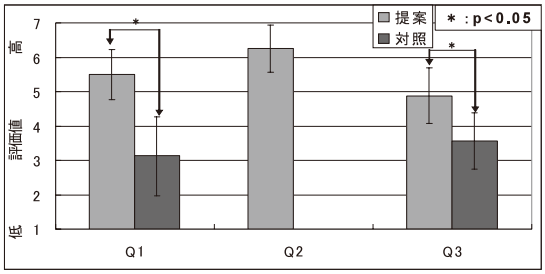


図 7 評価実験 I：正答率
Fig. 7 Correct answer rate (Experiment I).



Q1：タスク・スケジュールの状態は把握しやすかったですか？
Q2：空き時間の表示は役に立ちましたか？（提案システムのみ）
Q3：このシステムをあなたのスケジュール管理で使用したいですか？

図 8 評価実験 I：アンケート結果
Fig. 8 Subjective evaluation (Experiment I).

6. 評価実験 II：タスク間関係

6.1 実験目的

評価実験 II の目的は，タスク間関係に基づいたタスク管理手法によりユーザが現在のタスク実行可能性や優先順序を素早く，確実に理解できるかどうかを調べることである．

6.2 実験方法

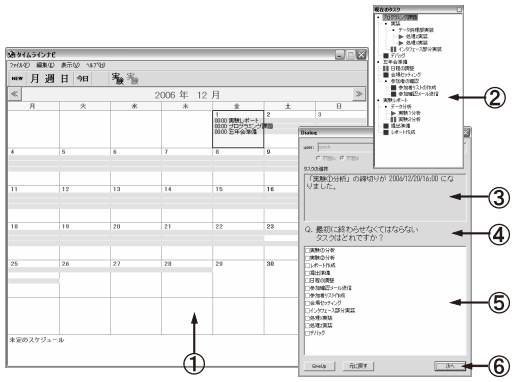
被験者は 20 代の学生 12 人である．被験者には，タスク間関係に基づいたタスク管理機能がある通常のタイムラインナビ（以下，提案システム）と，その機能がないタイムラインナビ（以下，対照システム）の両方を用いて実験タスクを行わせた．対照システムでは，タイムラインナビのタスク間関係に基づいたタスク管理を行うための機能である以下の機能が省かれている．

- サブタスク入力ダイアログ（図 3）の使用
- タスクウィンドウ（図 2 右）のタスクの階層構造表示
- タスクウィンドウ（図 2 右）のタスク状態を示すマーク（表 1）の表示

対照手法では，タスクウィンドウは登録したタスクが締切りの早い順に並ぶ To-Do リストとなっている．

実験画面を**図 9** に示す．実験の流れは以下のとおりである．なお，被験者には，あらかじめ実験者が用意した複数のタスクを各システムで入力させ，入力後の状態を初期状態として使用した．

1. 被験者が ⑥ の「実験開始」ボタンを押すと回答



番号	名称
①	タイムラインナビ：メインウィンドウ
②	タイムラインナビ：タスクウィンドウ
③	タスクの進捗表示欄
④	質問
⑤	回答チェックボックス
⑥	「実験開始」／「次へ」ボタン

図 9 評価実験 II：実験画面
Fig. 9 A screenshot of experiment II.

表 2 評価実験 II：タスク進捗パターンと質問の種類
Table 2 Situations and questions (Experiment II).

タスクの進捗パターン	質問の種類*	問題数（実験タスク当たり）
初期状態	実行可能	1
初期状態	優先	1
サブタスクの完了	実行可能	18
メインタスクの完了	優先	2
イベントの発生	実行可能	6
締め切りの変更	優先	3
サブタスクの分割	実行可能	6
サブタスクの追加	実行可能	3

＊：実行可能：現在実行可能なタスクはどれですか？
優先：最初に終わらせなくてはならないタスクはどれですか？

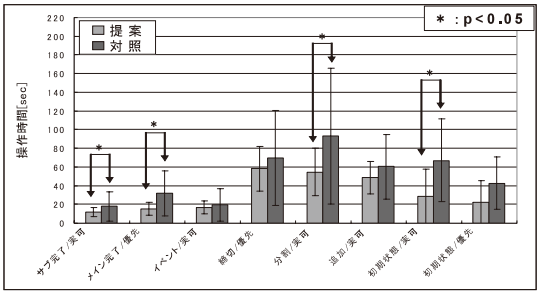


図 10 評価実験 II：操作時間
Fig. 10 Operation time (Experiment II).

実験が開始される．

2. ③ のタスクの進捗表示欄に「○○ タスクが完了しました.」などのように新たなタスクの進捗が表示される．
3. 被験者は ③ のタスクの進捗表示欄に表示されたタスクの進捗をタイムラインナビに反映させる．
4. 反映させた後の状態で ④ に表示される質問に答える．質問は 2 種類あり，
   - 現在実行可能なタスク（以下，実行可能タスク）はどれですか？
   - 最初に終わらせなくてはならないタスク（以下，優先タスク）はどれですか？

   のいずれかである．回答は ⑤ の回答チェックボックスの選択肢からあてはまるものすべてにチェックを入れることで行う．
5. ⑥ の「次へ」ボタンを押す．

被験者には手順 2 から 5 を繰り返し行わせた．ただし，最初の 2 問はタスクの進捗表示欄に「最初の状態で答えてください」と表示され，初期状態で手順 4 の各質問に答えてもらった．また，手順 5 で正しく回答できなかった場合には，不正解であることが通知され，被験者には手順 4 に戻り回答をやり直させた．以下では，手順 2 から 5 までの時間を操作時間とする．

手順 2 で表示されるタスクの進捗パターンと手順 4 での質問の種類との関係を**表 2** に示す．なお，進捗パターンは，タスクの進捗として実環境に存在するものを模したものを用いた．また，質問の種類は，初期状態の場合を除き，その進捗の発生時にタスク状況が大きく変化する方を選択した．たとえば，締切りの変更が発生した場合では，現在実行可能なタスクに変化はないが，優先的にやらなければならないタスクが変化すると考えられるので，優先タスクを尋ねる質問を選択した．

問題間の難易度の差や，慣れによる影響をなくすため，実験タスクを 2 種類（Task-A，Task-B）用意し，被験者 3 人ずつに，以下の 4 通りの順序で実験を行わせた．

- 提案システム（Task-A）⇒ 対照システム（Task-B）
- 対照システム（Task-A）⇒ 提案システム（Task-B）
- 提案システム（Task-B）⇒ 対照システム（Task-A）
- 対照システム（Task-B）⇒ 提案システム（Task-A）

また，各システムの利用を終えるたびに，アンケートに回答させた．

なお，操作や問題形式に慣れるため実験前に簡単な練習問題を実施した．

6.3 実験結果および考察

評価実験の結果および考察を以下に示す．評価項目は，操作時間と正答率，アンケートによる主観評価の 3 項目である．

6.3.1 操 作 時 間

図 10 は，表 2 で示した各項目ごとの平均操作時間を表している．なお，一度でも不正解となったデータは省いた．

メインタスクの完了，サブタスクの完了の項目で提案システムが有意に短い結果となった（$p < 0.05$）．タスクを完了させる操作において，提案システムと対照システムの間に大きな時間の差はない．したがって，タスク状況を把握する時間に差が現れていると考えられる．よって，タスク状況の把握には提案システムが有用であるといえる．これは提案システムではタスクウィンドウを見れば，各タスクの状態がマークで示されており，すぐに状態を把握できるためと考えられる．

イベントの発生の項目では有意な差は見られなかった．この原因としては，イベントが発生するとそのイベントが発生したタスクが実行可能な状態に変化するだけで他のタスク状況は変化しないため，両システムとも状況の把握が容易であったことが考えられる．

締切りの変更，サブタスクの分割・追加の項目では，提案システムの方が時間が短くなる傾向にあるが，有意差が見られたのはサブタスクの分割のみであった

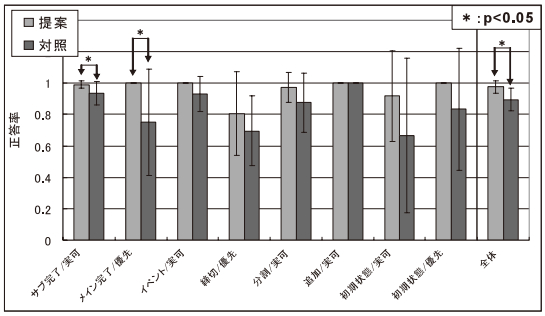


図 11 評価実験 II：正答率
Fig. 11 Correct answer rate (Experiment II).

($p < 0.05$)．対照システムにおける被験者のタスクの登録方法は，メインタスクのメモ欄にすべてのサブタスクの情報を書き込む，サブタスクごとに分けて別々にタスクを登録する，など様々であった．この対照システムでの被験者ごとの登録方法の違いによって，システムに対する操作量が多いこれら 3 項目では分散が大きくなり，操作時間に関して差が現れにくかったと考えられる．

初期状態の項目において，実行可能タスクを尋ねる質問では有意差が見られ（$p < 0.05$），平均で 2 倍以上の差があった．これは，この問題が回答実験の最初の問題であり，全タスク状況を把握する必要があったため，提案システムと対照システムの差が顕著に現れていると考えられる．これに対し，優先タスクを尋ねる質問では有意な差は見られなかった．この原因としては，最初の問題である程度タスクの状態を把握できてしまっていたことが考えられる．したがって，初期状態で最初に優先タスクを尋ねる場合についても，今後調査する必要がある．

6.3.2 正 答 率

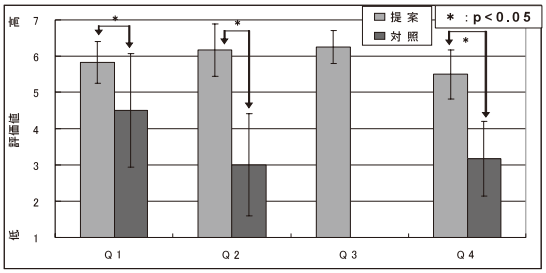
一度でも不正解のダイアログが表示された場合にその問題を不正解とし，正答率を求めた．**図 11** に正答率のグラフを示す．

グラフより提案システムの方が正答率が高くなっていることから，提案手法によって被験者はより正確にタスク状況を把握できていたことが分かる．これは，前項でも述べたように，提案システムではタスクウィンドウを見れば，各タスクの状態がマークで示されており，すぐに状態を把握できるためと考えられる．

6.3.3 アンケートによる主観評価

図 12 にアンケート結果を示す．それぞれの項目について 7 段階で評価を行った．評価値は 4 を「どちらでもない」とし，数値が高いほど好評価となっている．

Q1，2，3 の各項目で提案システムが好評価となっていることが分かる．したがって，操作時間，正答率



Q1：入力は思い通りにできましたか？
Q2：タスクの状態は把握しやすかったですか？
Q3：タスクの階層構造に沿ったタスク表示は役に立ちましたか？
（提案システムのみ）
Q4：このシステムをあなたのタスク管理で使用したいですか？

図 12 評価実験 II：アンケート結果
Fig. 12 Subjective evaluation (Experiment II).

の結果と同様に，アンケート結果からも提案システムが有用であることが確認できる．また，Q4 でも提案システムは好評価を得たことから，被験者はタスク間関係に基づいたタスク管理を，自身のタスク管理でも使用したいと考えていることが分かる．

7. 実使用評価

実際にタイムラインナビを実践的に用いた場合の評価として，著者の 1 人（大学教員）がタイムラインナビの試作システムを 6 カ月，現行システムを 5 カ月使用した経験を述べる（現行システムは現在も使用中である）．タイムラインナビが利用可能になる前は，手帳および Palm OS を利用した PDA のスケジュール/タスク管理ソフトを用いていた．

タイムラインナビ使用開始当初，まず，私的な用件を除いた仕事上のスケジュール管理をタイムラインナビで行うこととした（現在では，プライベートな時間も入力している．それは，土日に仕事をすることがあるためである）．この理由は，タイムラインナビはユーザのすべてのスケジュールの管理をすることができるが，評価データとして公表したくない個人情報を除くことと，現状のタイムラインナビが PC 上で稼働しているため，オフィスから持ち出して使用するのが不便なためである（現在では，タイムラインナビのデータベースファイルをインターネット上のディスクスペースにおいて共有しているため，自宅でも利用可能である）．これを実現するため，

- 毎日，0 時から 9 時までの睡眠
- 毎日，21 時以降の夕食その他
- 毎土曜日 9 時から 21 時までの休息
- 毎日曜日 9 時から 21 時までの休息

をスケジュールとして登録した．タイムラインナビに

は毎日，毎週の設定が可能なため，これらの設定には4個のスケジュールを登録するだけでよかった．もちろん，土曜日や21時以降に仕事上のスケジュールが入る場合もあるが，その場合は，その日のスケジュールだけを取り消せばよい．

教員には，定例のスケジュールがある．たとえば，講義や定例の会議（教授会など），学生との輪講や研究ミーティングなどである．また，定例のタスクも存在する．講義の準備や，主催する定例会議の準備などである．タイムラインナビでは，スケジュールだけでなく，タスクも毎日，毎週，毎月の設定ができるため，これらの登録も，それほど手間をかけることなく終了した．以上が，タイムラインナビを使い始めるときにした作業であった．

実際に使用し始めると，タイムラインナビのある環境は快適であった．つねに直近の締切りのタスクに対して空き時間が表示されているので，突然学生が部屋のドアをノックして割り込んできても，どの程度の時間をその学生のために使っても問題ないかがその場で分かった．また，その残り時間を見ることで，今日はもう帰宅してもよいのか，もう少し作業して帰らないと明日に負担がかかりすぎるのかも明確に判断できた．

不定期のスケジュールやタスクの登録は，それまで用いていたスケジューラに登録するのと手間はほぼ同じであった．また，タスクをステップワイズに詳細化できる機能は有用であった．たとえば，学科のカリキュラムに関する資料を作成するというタスクが生じたとき，最初はそのように登録していたが，資料の様式を調べてみると，各科目の担当教員に問い合わせなければならない項目があることが分かり，その時点で，「問合せメールを送る」（サブタスク，これはその場でメールを送信し，実行済みになった），「問合せの回答が帰ってくるのを待つ」（イベント），「回答結果をまとめる」（サブタスク）のように，このタスクの残りの部分が分割された．このように，タスクが一度にできない場合，どこまで実行したかがタイムラインナビ上で管理され明確である．問合せの回答が戻ってこない場合にもタイムラインナビが警告してくれる（次の回答結果をまとめるタスクの空き時間が0になるため）ので，回答を督促することができる．

このように，タイムラインナビではすべての仕事を一度に実行できるタスク単位で管理できる．これにより，次の問題も緩和できた．タイムラインナビでは，タスクの仕事量（見込み時間），すでにとりかかっているタスクでは残りの仕事量の管理が必須である．タスクがコンピュータ上で行われているときには，そのタスクの作業フォルダを登録しておけば，そのフォルダを開いている時間は作業していると判断して自動的に残り仕事量を更新してくれる．しかし，当初登録した仕事量に誤差があったり，コンピュータ上で作業を行わなかったりしたときには，ユーザがそのつど残り仕事量を修正しなくてはならない．実際，1月から2月にかけての期間では，卒業論文や修士論文の紙の原稿のチェックを頻繁に行っていたため，コンピュータを使わずに作業を行っている状態であった．しかし，前述のように，管理単位を一度に行える細かなタスク単位にできるので，実際には仕事量の修正回数はそれほど多くはなかった．たいていの場合はそのタスク（サブタスク）の終了ボタンを押していた．このように，最初にタスクを登録するときには大まかに登録し，タスクの詳細が分かるにつれ，それを細分化して，最終的には一度に実行可能なサブタスク単位で管理することが，スケジュール/タスク管理のオーバヘッドを本来のタスクの切替えのときに行うことにつながり，本来のタスクの邪魔にならず，タスクを円滑に実行できることが分かった．

8. おわりに

本論文では，既存のスケジューラシステムがかかえている問題を解消するために，空き時間を利用したタスク・スケジュール管理，タスク間関係に基づいたタスク管理という2つの手法を提案した．そして，これらの提案手法の機能を付加したスケジューラシステムである「タイムラインナビ」を設計，実装し，評価実験を行った．その結果，空き時間を表示するインタフェースにより，スケジューリングに無理が生じていないかの確認時間を短縮することができ，その後のタスク・スケジュールの調整にも有用であることが分かった．また，タスク間関係に基づいた管理を行えるようにしたことによって，分かりやすく効率的なタスク管理ができ，現在のタスクの状態をより把握しやすくなることが分かった．また，11カ月にわたる利用経験により，タイムラインナビは十分実用的価値を有するシステムであることも分かった．

今後の課題としては，タイムラインナビを実環境でさらに長期的に使用し，評価・改善を行っていくことなどがあげられる．

また現在，非対面コミュニケーションにおいて適切なタイミングで他者に割り込むための研究が行われている[7]．我々の研究グループでも適切なタイミングで割り込むために人の「忙しさ」を判定する研究を行っている．その中でユーザのかかえているタスクの締

切りや空き時間がユーザの「忙しさ」に影響を与えることが分かっている[8),9)]．そこで，本研究で実装したタイムラインナビを用いることによって，より正確な「忙しさ」判定法への応用が期待できる．

謝辞 本研究を進めるにあたり，協力をいただいた京都工芸繊維大学人間情報技術研究室の皆様に深く感謝する．

本研究の一部は，平成 17〜19 年度科学研究費補助金（基盤研究 (C) 17500068）の支援を受けている．

参 考 文 献

1) Microsoft Office Outlook. http://office.microsoft.com/ja-jp/outlook/default.aspx
2) iCal. http://www.apple.com/jp/ical/
3) Microsoft Office Project. http://office.microsoft.com/ja-jp/project/default.aspx
4) Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D.G. and Ducheneaut, N.: What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager, *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp.735–742 (2004).
5) Bellotti, V., Duchenaut, N., Howard, M. and Smith, I.: Taking email to task: The design and evaluation of a task management centered email tool, *Proc. CHI 2003 Conference on Human Factors in Computer Systems*, pp.345–352 (2003).
6) 大向一輝，武田英明，三木光範：多様かつ曖昧な個人タスクのための管理システムの提案と実装，エージェント合同シンポジウム（JAWS2002）講演論文集，pp.502–509 (2002).
7) Fogarty, J., Hudson, S.E., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S., Lee, J.C. and Yang, J.: Predicting human interruptibility with sensors, *ACM Trans. Computer-Human Interaction*, Vol.12, No.1, pp.119–146 (2005).
8) 松田康弘，倉本 到，渋谷 雄，辻野嘉宏：オフィス環境におけるタスクの時間制約による切迫感を考慮した「忙しさ」判定，ヒューマンインタフェース学会論文誌，Vol.7, No.3, pp.99–106 (2005).
9) 宮柱知愛，堤 大輔，倉本 到，渋谷 雄，辻野嘉宏：スケジュール情報に基づく忙閑度の推定，情報処理学会研究報告，2006-HI-119, Vol.2006, No.7, pp.39–46 (2006).

(平成 19 年 3 月 23 日受付)
(平成 19 年 9 月 3 日採録)

堤 大輔

2007 年京都工芸繊維大学大学院工芸科学研究科博士前期課程修了．同年より京セラコミュニケーションシステム株式会社に入社．在学中はアウェアネス情報の抽出と応用に関する研究に従事．

倉本 到（正会員）

2001 年大阪大学大学院基礎工学研究科情報数理系専攻博士後期課程修了．博士（工学）．同年より京都工芸繊維大学工芸学部電子情報工学科助手．2007 年同大学大学院工芸科学研究科助教．CSCW，ヒューマンインタフェースおよびエンタテインメントコンピューティングに関する研究に従事．ヒューマンインタフェース学会会員．

渋谷 雄（正会員）

1990 年大阪大学大学院工学研究科通信工学専攻博士後期課程修了．工学博士．同年より京都工芸繊維大学工芸学部電子情報工学科助手，同講師，同助教授を経て，2007 年京都工芸繊維大学情報科学センター教授．1997〜1998 年ドイツ，カッセル大学客員研究員．ヒューマンインタフェース，人間情報技術の研究に従事．電子情報通信学会，ヒューマンインタフェース学会，システム制御情報学会，日本人間工学会，ACM 各会員．

辻野 嘉宏（正会員）

1979 年大阪大学基礎工学部情報工学科卒業．1984 年大阪大学大学院博士課程修了．工学博士．同年大阪大学基礎工学部助手．同大学講師，助教授を経て，1999 年より京都工芸繊維大学大学院工芸科学研究科教授．計算機言語，並列処理技術，HCI，ソフトウェア工学に関する研究に従事．IEEE-CS，ACM，電子情報通信学会，日本ソフトウェア科学会，ヒューマンインタフェース学会各会員．