

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

大向 一輝[†] 武田 英明^{†,††} 三木 光範^{†††}

A Proposal and Implementation of a Management System for Various and Ambiguous Personal Tasks

Ikki OHMUKAI[†], Hideaki TAKEDA^{†,††}, and Mitsunori MIKI^{†††}

あらまし This paper proposes a person-centered approach for personal task management in which people can decide management of their tasks according to their environments including their subjective and multivalent judgment and human relationship. In our approach the task management is modeled as a decision-making process on their own resources. Human decision-making process consists of three types of activity, i.e., the intelligence activity, the design activity, and the choice activity. The proposal system assists each activity by three sub-systems, i.e., Visualizer, Optimizer, and Recommender respectively. At first, Visualizer indicates the attributes associated with each task such as the deadline, the subjective priority, and the workload, which are determined by the user. And Optimizer generates executable schedules from these tasks by active scheduler and multi-objective genetic algorithm. Finally, Recommender evaluates these alternatives by analytic hierarchy process. Also the system is able to analyze the human relationship of the user group with PageRank algorithm, and this result is utilized to improve the performance of the task scheduler. We implement client/server system which uses mobile devices. And we verify the function of the proposed system by a simulation with two scenarios and by an experiment.

キーワード scheduling, task management, mobile phone application, optimization algorithm, analytic hierarchy process, human network

1. はじめに

われわれ個人にとって、時間や金銭といった有限の資源をいかに管理していくかは重要な問題である。従来からわれわれは企業や学校などの組織に属し、組織の目的を達成するために多くの資源を割くとともに、それらの資源の管理を委託してきた。これらの組織は階層構造を形成していることが多く、ある階層に所属する人間の資源は、その上位の人間が効率的に管理するというシステムとなっている [1]。

近年の社会システムの変化は、個人のライフスタイルに大きな影響を与えている。非営利組織 (NPO) や非政府組織 (NGO) の活動が盛んになり、人々が複数の組織に所属することは珍しくない。このような状況では、複数の組織が連携して個人の資源を管理することは現実的でない。

また、個人が抱えるタスクの種類は多様であり、それらの定義は曖昧である。このようなタスクを評価するためには主観的な尺度や複数の尺度が必要となると考えられる。

そこで、本研究では個人を中心としたタスク管理を実現するための手法を検討し、実装を行う。

この手法の特徴は単にタスクスケジューリングを支援するだけではなく、ユーザに対して自身のタスク状況を明確に知らせることにある。本研究では、個人のタスク管理を意思決定プロセスとして捉え、そのプロセスに含まれる 3 種類の活動をそれぞれに支援する手法について検討する。

[†] 総合研究大学院大学, 東京都

Department of Informatics, The Graduate University for Advanced Studies, 2-1-2 Hitotsubashi Chiyoda-ku, Tokyo, 101-8430 Japan

^{††} 国立情報学研究所, 東京都

National Institute of Informatics, 2-1-2 Hitotsubashi Chiyoda-ku, Tokyo, 101-8430 Japan

^{†††} 同志社大学, 京都府

Department of Knowledge Engineering, Doshisha University, 1-3 Tatara Miyakodani, Kyotanabe-shi, 610-0321 Japan

まず、情報活動の支援では、Visualizer によるタスク情報の可視化を行い、ユーザに問題点を通知する。次に設計活動の支援としては、Optimizer が複数の実行可能なスケジュールを生成する。最後に、選択活動の支援としては、Recommender が階層分析法によって代替案を評価し、ユーザに提示する。また、Analyzer を用いてグループでの使用時に人間関係ネットワークを構築し、これを分析することで意思決定支援アルゴリズムの性能を高める。

本研究の有効性を確かめるために、2種類のシナリオに沿ってシステムの挙動を検証する。最初のシナリオでは、スケジューリングにおける典型的な問題であるダブルブッキングがどのように解決されるかを示す。2番目のシナリオでは、人間関係に基づくタスクの評価がいかに計算され、支援に生かされるかを示す。

また、これらの手法を携帯電話を用いたタスク管理システムとして実装し、その有効性を実証実験によって検証する。

2. 個人のタスク管理およびスケジューリング

われわれは常に複数のタスクを抱えており、どのタスクを優先的に処理するかを決定する必要がある。タスクには締切日時の他に重要性、処理に必要な時間といった要素があり、これらを考慮しながら適切な評価を行うことは難しい。また、それぞれのタスクの目的は多様であり、また個人ごとに評価基準も異なるため、結果としてタスク管理が恣意的になることが多い。例えば、処理に時間のかからないタスクばかりを優先した結果、重要なタスクが処理できない事態が発生することがある。このように、個々のタスクを単一の尺度で評価することは危険である。

個人の生産性を高めるためには、タスクを多角的、総合的に評価し、その結果をもってどのタスクを優先的に処理するか、中・長期的にどのようなスケジューリングを行うかといった視点が重要であると考えられる。

本研究が対象とする個人のタスク管理およびスケジューリングには以下のような課題があると考えられる。

- 目的関数の扱い

一般的なスケジューリング問題では、与えられた全てのタスクを処理することは自明であるとされている。しかしながら、本研究で対象とする問題では、必ずしも全てのタスクを処理する必要はなく、コストや

タスクの重要性を考慮したバランスの良いスケジューリングが求められる。したがって、個人タスク管理システムでは、様々な状況に対応できるよう、目的関数には総作業時間の最小化だけでなく、処理すべきタスクの重要度パラメータによる報酬の最大化などの目的関数を追加した多目的最適化問題としてタスクスケジューリング問題を扱う必要がある。

- タスクの性質

タスクには移動可能タスクと固定タスクの2種類が存在する [2]。移動可能タスクとは、締切日時を過ぎない限りはどの時間帯にも配置できるタスクであり、固定タスクとは会議や学会のように開始時間と終了時間があらかじめ設定されており、個人の都合では変更のできないタスクである。これら2種類のタスクが混在した環境でのスケジューリングに関しては、あまり研究がなされていない。

- 環境および目的の変化

新たなタスクの発生や変更によって、ユーザのタスク状況や目的が大幅に変わる。このような問題では、目的関数の形状を利用した最適化手法では環境の変化に追従した再スケジューリングを行うのは難しい。また、データ自体が常に変化するため、データ構造を柔軟に変更することが可能なアルゴリズムを用いる必要がある。

このように、個人の資源管理は複雑さを増しており、特定の組織が特定の目的のために個人の資源を効率的に管理することは不可能になりつつある。また、ソフトウェアエージェントによって複数の人間が資源の調整を行う研究やグループウェア [3] などが存在するが、これらは基本的に単一の組織を支援するシステムであるため、これを適用することも難しい。

3. 提案手法

3.1 意思決定支援システム

前節で述べた問題点および課題を解決するために、本研究では意思決定論および意思決定支援システムの考え方を利用する。意思決定支援システムは、ユーザが対話的にデータおよびモデルを操作して情報を獲得し、質の高い意思決定を行うためのシステムであると定義されている [4]。

意思決定とは、一般に考えられているように決断や選択という1回性のイベントではない。Simon は、人間の意思決定は情報活動、設計活動、および選択活動の3つのプロセスから構成されると考察している。情

報活動では現状分析によって問題点を抽出し、設計活動によってその問題点を解決するための代替案を複数作成する。次に選択活動においてそれらの代替案を評価し、最終的に 1 つの代替案を選択する。

そして、意思決定支援システムは、この 3 つのプロセスのそれぞれを支援することが要求される。

これまで意思決定支援システムが対象としてきた問題は、目的が明確であるが意思決定の各プロセスを遂行することが何らかの理由で困難であるような問題であった [5]。このような問題においては、システムを設計する際に目的関数、すなわちどのような状態が最適であるかを定義し、その後目的関数と個々のプロセスとの関係を記述する。こうして構築されたシステムはユーザに代わって代替案の最適性を評価する。しかしながら、個人のタスク管理においては、ユーザにとっての最終的、総合的な目的が明確であることは少なく、最適な状態を定義することは難しい。そこで、本研究では個人タスク管理のプロセスを以下のように定義し、従来の意思決定支援手法を適用可能にする。

- 1) タスク群から実行可能なスケジュールを複数生成する。
- 2) 複数のスケジュールから 1 つを選択し、実行する。
- 3) 新たなタスクの発生や予測しない事態が生じた場合には 1 に戻る。

このプロセスは意思決定のサイクル（繰り返し）であるといえる。

4. 提案システム

本研究で提案するシステムの構成を図 1 に示す。システムのアーキテクチャはクライアント/サーバ型である。ネットワーク上に設置されたサーバは、タスク情報が格納されたデータベースを中心に後述の 4 つのアルゴリズムを備える。データの入力や表示、システムの操作のためのクライアントには、インターネット接続が可能な携帯電話上で動作する Java アプレット (iappli)[6] を用いる。

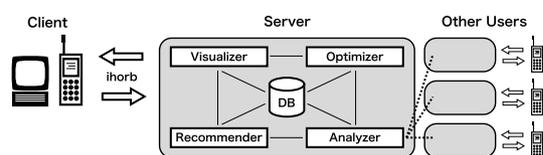


図 1 System configuration

4.1 データ構造

ユーザはそれぞれのタスクに関して以下のパラメータを設定し、システムに入力する。

- 名称 (Name)
- 時間パラメータ
 - タスクの開始時間 (Start)
 - タスクの終了時間 (Deadline)
 - 拘束時間の見積もり (Work)

移動可能タスクと固定タスクは、これらのパラメータのうちのどれが定義されているかによって区別することができる。すなわち、開始時間と終了時間が定まっているのが固定タスク。終了時間と拘束時間が定まっているのが移動可能タスクである。

- 重要度 (Priority)

タスクの重要度を 1~5 の 5 段階で評価する。ここでは意思決定者の主観によって設定する。

- 依頼者 (From)

このパラメータを用いて、依頼者ごとのタスクの依頼頻度を集計する。

4.2 Visualizer: 情報活動の支援

個人のタスク管理を適切に行うためには、まずユーザの行動に問題点があることを認識させる必要がある。したがって、入力データから問題点を引き出すことが情報活動の支援として重要になる。

携帯端末は表示画面が小さく、高解像度のグラフィックスを表示することができない。そのため、携帯端末ではクロノグラム形式でタスク状況を表示する。この形式では、詳細なデータを読みとりやすいことや、複数のタスクの重複 (ダブルブッキング) などの問題点が容易に把握できるという利点がある。

動作中の画面を図 2 に示す。図 2 上のクロノグラムは横軸を時間とし、個々のタスクを棒状に表示する。縦軸は Deadline を表し、締切までの時間が短いタスクほど上部に表示される。また、各タスクを選択するとダイアログが表示され、タスク情報の変更、追加などを行うことができる。クロノグラムにおいては時間軸の表示スケールを変更することも可能である。

情報を表示する際に、締切が過ぎるなどにより実行不可能なタスクが発生した場合や、ダブルブッキングが検知された場合には、これをユーザに警告する。また、再スケジュールリングが必要であると判断された場合には、次に述べる Optimizer を呼び出すためのボタンを表示する。

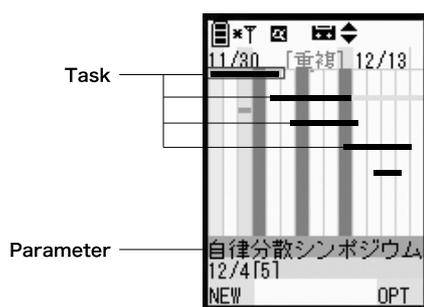


図 2 2D interface on prototype system

4.3 Optimizer: 設計活動の支援

Optimizer は、ユーザが与えたタスク情報をもとに、実行可能なスケジュールを生成する。まず、アクティブスケジューラアルゴリズムにより、ダブルブッキングが生じているタスクのパラメータを変更し、タスク集合の直列化を行う。

また、アクティブスケジューラでは問題が解決しない場合には、多目的最適化アルゴリズムを用いて代替案を生成する。

提案システムでは遺伝的アルゴリズム (Genetic Algorithm: GA)[7] を多目的化し、パレート保存戦略を用いた多目的遺伝的アルゴリズム (Multi-Objective GA)[8] によってスケジュールを生成する。

提案システムで用いる多目的遺伝的アルゴリズムでは、目的関数を以下のように設定した。

- タスク処理数の最大化

スケジュールに含まれるタスクの数を N_{task} と表し、これを最大化する。タスクをより多く処理できるようなスケジュールに高い適合度が与えられる。

- 時間の有効利用率の最大化

現在時刻を T_0 とし、スケジュールリングの対象となる最後のタスクの締切時間を T_{last} とすると、タスク処理が可能である時間 T_{all} は $T_0 - T_{last}$ である。そして、スケジュールに含まれるタスクの総拘束時間を T_{task} とすると、時間の有効利用率 R_{util} は $R_{util} = \frac{T_{task}}{T_{all}}$ で求められる。ここでの目的は R_{util} の最大化である。ただし、 T_{task} が T_{all} を超える場合、すなわち $R_{util} < 1.0$ である場合には、ペナルティとして $R_{util} = 0.5$ とする。

- 重要度によって与えられる報酬の最大化

スケジュールに含まれるタスクの重要度 (Priority) の合計 $R_{priority}$ を最大化する。これは、主観的な重要性を目的関数としたものである。

- 依頼者によって与えられる報酬の最大化

スケジュールに含まれるタスクの依頼者の依頼頻度 (From) の合計 R_{from} を最大化する。これは、タスクの依頼頻度が大きい人物ほど意思決定者にとって重要な人物であるという仮定に基づいたタスクの評価方法である。 $R_{priority}$ が個人的な評価であるのとは対照的に、 R_{from} は人間関係を用いた外部的な評価であるといえる [9]。

多目的遺伝的アルゴリズムでは個々の解候補を評価し、被支配的な候補を消去する。最終的に残った候補が代替案として次の Recommender に送信される。

4.4 Recommender: 選択活動の支援

個人のタスク管理問題では、選択した解が最適であるかどうかを検証することは難しい。しかしながら、数値データを評価し、ユーザに提示することで、従来の主観的・恣意的な決定を回避し、客観的で説明可能な決定をシステムが支援することは可能である。

Recommender では Optimizer によって生成された代替案を評価し、ユーザに評価値を提示する。アルゴリズムには定量的なデータおよび定性的な情報を同時に扱うことのできる階層分析法 (Analytic Hierarchy Process: AHP)[10] を用いる。

通常の AHP では代替案の数が増えるに従い、一対比較の回数が増大し、意思決定者の負荷が非常に大きくなる。また代替案が追加された場合には、再び全ての一対比較を行う必要がある。そのような事態を避けるために、ここでは絶対評価法 (Absolute Measurement)[10] による AHP を用いる。

4.5 Analyzer: 依頼者による重要度の推定

Visualizer, Optimizer, Recommender による意思決定支援の他に、個々のタスクに与えられたパラメータから新たな情報を抽出、活用することで、支援の質を向上させる。

タスクの重要度には、主観的に設定できる属性とともに、タスクの依頼者によって変化する属性が存在すると考えられる。ここでは、データベースに格納された依頼者情報から PageRank アルゴリズム [11] を用いて、後者の重要性を算出する手法を提案する。この結果を意思決定支援アルゴリズムにフィードバックすることで、より正確な支援が可能になると考えられる。

5. 提案システムの有効性

本章では、提案システムを 2 種類のシナリオに沿って動作させた結果を述べ、考察を行う。

5.1 シナリオ 1: スケジュールの提示およびダブルブッキングの解決

本シナリオでは、システムの構成要素である Visualizer, Optimizer, および Recommender の連携によるスケジューリングの手順を示す。

テスト問題として、表 1 に示す 5 つのタスクをシステムに登録する。なお、From のあとの数字は依頼頻度を表すものとする。

表 1 Scenario 1: Test data

Name	Start time	Deadline	Workload	Pri.	From
A	—	1/10 15:00	10:00	3	W (3)
B	1/11 11:00	1/11 15:00	4:00	5	X (5)
C	—	1/11 20:00	12:00	4	Y (2)
D	—	1/13 14:00	11:00	2	Z (4)
E	—	1/13 18:00	13:00	5	W (3)

これらのタスクを Visualizer で表示すると図 3 のようになり、2カ所でダブルブッキングが発生していることが認識できる。なお、黒で表示されるタスクが移動可能タスク、白で表示されるタスクが固定タスクである。

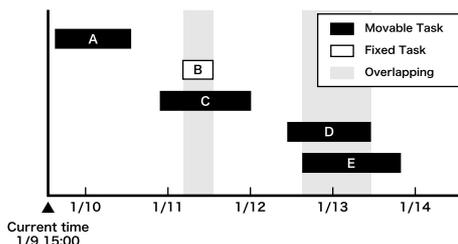


図 3 Scenario 1: Displaying initial condition by Visualizer

システムは、1 つめのダブルブッキングを解決するために Optimizer を呼び出す。Optimizer はアクティブスケジューラにより、移動可能タスクであるタスク C の開始時間を、タスク B の拘束時間である 4 時間分だけ前に設定する。この結果、最初のダブルブッキングは解決される。

次に、2 つめのダブルブッキングを解決を図る。Optimizer はタスク D とタスク E の締切時間を比較し、早い方のタスク D をタスク E の開始時間よりも前に移動させる。しかしながら、この移動によって連鎖的にタスク C とタスク D との間に新たなダブルブッキングが発生する。同様に、タスク C とタスク D のダブルブッキングを解決すると、タスク A とタスク C

との間に問題が発生し、これを解決しようとする、図 4 のタスク A の開始時間が現在時刻よりも前に設定されてしまう。すなわち、アクティブスケジューラによる一連のスケジューリングは失敗したといえる。

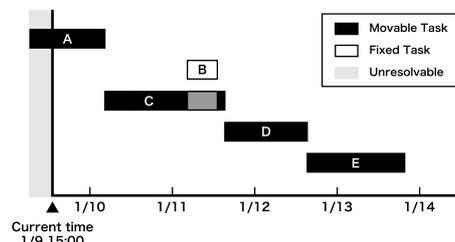


図 4 Scenario 1: Fail to schedule

そこで、Optimizer は多目的最適化アルゴリズムを用いて実行可能なスケジュールを生成するためのタスクの組み合わせを求める。これによって求められた候補に対してアクティブスケジューラによる時間の調整を行った結果、図 5 に示す代替案が得られる。ここでは処理タスク数 N_{task} が多いものから順に一部を示した。

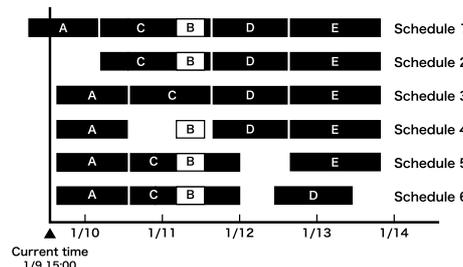


図 5 Scenario 1: Alternatives designed by Optimizer

これらの代替案の評価は表 2 となる。括弧で示した数字は各目的関数でのランキングである。これによれば、全ての評価項目に対して Schedule 6 は Schedule 2 および 4 と同等もしくは下回っている。Schedule 6 は有効な代替案とはいえないため、Optimizer によって候補から除外される。

表 2 Scenario 1: Evaluation value of alternatives

Alternative	N_{task}	R_{util}	$R_{priority}$	R_{from}
Schedule 1	5 (1)	0.500 (6)	19 (1)	17 (1)
Schedule 2	4 (2)	0.851 (2)	16 (3)	14 (3)
Schedule 3	4 (2)	0.979 (1)	14 (5)	12 (6)
Schedule 4	4 (2)	0.809 (4)	15 (4)	15 (2)
Schedule 5	4 (2)	0.830 (3)	17 (2)	13 (5)
Schedule 6	4 (2)	0.787 (5)	14 (5)	14 (3)

Recommender ではこれら 5 つの代替案について評価を行う．AHP での初期設定である各評価基準の対比較行列は表 3 のように定めた．この行列から得られた固有ベクトル，すなわち評価の重みは表 4 である．Optimizer による代替案の評価値は正規化する．

表 3 Scenario 1: Matrix for pairwise comparison

	N_{task}	R_{util}	$R_{priority}$	R_{from}
N_{task}	1	1/7	1/5	1/3
R_{util}	7	1	1	3
$R_{priority}$	5	1	1	5
R_{from}	3	1/3	1/5	1

表 4 Scenario 1: Weighting vector of criteria

Criterion	Weight
N_{task}	0.059
R_{util}	0.391
$R_{priority}$	0.424
R_{from}	0.126

AHP による最終結果を表 5 に示す．括弧で示した数字はランキングである．これによれば，重みの大きい重要度と時間の有効利用率の評価が比較的高い Schedule 5 に最大の評価値が与えられている．一方，重要度の評価がもっとも高い Schedule 1 では，時間の有効利用率の評価においてペナルティが課されているために最低の評価値となった．

表 5 Scenario 1: Final result

Alternative	Evaluation value
Schedule 1	0.193 (5)
Schedule 2	0.204 (2)
Schedule 3	0.202 (3)
Schedule 4	0.196 (4)
Schedule 5	0.206 (1)

これらの結果をもとに，ユーザは代替案を選択する．そのスケジュールに含まれていないタスクが実行不可能タスクとして登録され，Visualizer での表示形式が変更される．

5.2 シナリオ 2：タスクの依頼関係に基づく重要度の推定

シナリオ 1 では，依頼者によるタスクの重要性評価の基準は，依頼頻度のみに基づいていた．しかしながら，タスクの依頼関係は双方向的なものであり，複数人が関係しあうために複雑なものとなる．

また，階層型の組織とは異なり，ネットワーク型の組織ではタスクの依頼者が上下関係にあるとは限らず，依頼者による重要性を把握することが難しくなる．そ

こで，提案システムを組織に導入し，Analyzer を活用することによって，組織内のタスクの流れが明確になり，タスクの評価に反映させることが可能になると考えられる．

ここでは，図 6 のような小規模の組織を想定した．この組織は 5 人から構成されており，過去のタスクの依頼頻度は矢印に添えられた数字で示されている．これらの依頼頻度のデータを Analyzer が集計し，重み付きの隣接行列として表現したものが表 6 である．

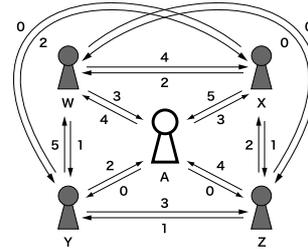


図 6 Scenario 2: Dependency relation of task (group)

表 6 Scenario 2: Task Network

Name	A	W	X	Y	Z
A	0	4	3	0	0
W	3	0	4	1	0
X	5	2	0	0	1
Y	2	5	2	0	3
Z	4	0	2	1	0

この隣接行列に対して PageRank アルゴリズムを適用し，得られた固有ベクトルがこの組織のタスクの依頼関係から推定される重要度である．この重要度を表 7 に示す．

表 7 Scenario 2: Weighting vector of person by PageRank

Name	Weight
A	0.044 (5)
W	0.093 (4)
X	0.170 (3)
Y	0.370 (1)
Z	0.322 (2)

Analyzer によって得られた重要度の値を個々のタスク情報に上書きし，新たに再スケジュールリングを行う．Optimizer による代替案の各目的関数に対する評価値を表 8 に示す．これをシナリオ 1 と同様の評価基準を用いて Recommender が処理した結果が表 9 である．表 5 と比較すると，Analyzer によって高い評価を得

た Y と Z の人物から依頼されたタスクが含まれない Schedule 4 および 5 の総合評価値が相対的に減少しており、これらのタスクが両方含まれている Schedule 2 および 3 の順位が上昇している。

表 8 Scenario 2: Evaluation value of alternatives

Alternative	N_{task}	R_{util}	$R_{priority}$	R_{from}
Schedule 1	5 (1)	0.500 (6)	19 (1)	1.049 (1)
Schedule 2	4 (2)	0.851 (2)	16 (3)	0.956 (2)
Schedule 3	4 (2)	0.979 (1)	14 (5)	0.879 (3)
Schedule 4	4 (2)	0.809 (4)	15 (4)	0.678 (5)
Schedule 5	4 (2)	0.830 (3)	17 (2)	0.727 (4)

表 9 Scenario 2: Final result

Alternative	Evaluation value
Schedule 1	0.194 (4)
Schedule 2	0.207 (1)
Schedule 3	0.206 (2)
Schedule 4	0.189 (5)
Schedule 5	0.203 (3)

このように、1 人のみのデータだけではなく、複数人のデータを統合することで、意思決定者がおかれている状況がより明確になり、それに応じてシステムの提案も変化することがわかる。

6. 提案システムの実証実験

6.1 システムの構築

本研究では提案システムの実装に際して以下の環境を用いた。

クライアントとしての携帯端末には、NTT DoCoMo の SO503i を用いた。画面の有効解像度は縦 120 ピクセル、横 120 ピクセルである。この端末上で動作するプログラムは Java 言語の携帯端末専用のサブセットである Java 2 Micro Edition Wireless SDK for DoJa [6] で記述した。

サーバの OS には Debian/GNU Linux 2.2 を用いた。また、Java の実行環境としては Java 2 Runtime Environment 1.4 を用いた。

クライアントとサーバの通信には、Java の分散オブジェクト技術である HORB [12] および携帯端末用 Java での実装である iHORB [13] を用いた。

6.2 実証実験

構築したシステムを用いて、実証実験を行った。被験者のサンプル数は 10 人である。被験者には Web ページ上のフォームより実際に自身のタスク情報を入力してもらい、その後提案システムでの可視化、最適化等の結果を携帯端末にて提示した。実験の終了後に

は後述のアンケートを行い、回答してもらった。この実験での 1 人あたりの平均入力タスク数は 17.1 であった。総タスク数に占める移動可能タスクの割合は 47.4 %、固定タスクは 52.6 % である。また、重要度の分布は 1 が 2.1 %、2 が 6.3 %、3 が 29.9 %、4 が 33.3 %、5 が 28.4 % であった。

6.2.1 提案システムによるスケジュール生成に関する評価実験

この実験では、個人のスケジュールを計算機で生成すること自体の可否について検討する。手順としては、まず被験者が入力したタスク情報を Visualizer がそのまま表示する。被験者にはその状態で一定時間携帯端末を操作してもらい、自身の状況を理解してもらう。次に、アクティブスケジューラを実行し、変更されたスケジュールが表示されている端末を被験者に操作してもらった。その後「直列化後のスケジュールに納めできるか」という質問を行った。評価は 1~7 の 7 段階とした。7 が最も評価が高く、4 は最適化の前後で評価の変化なし、1 が最も低い評価であるとした。

結果を表 10 の Step 1 に示す。この評価では被験者の平均値は 4 (評価尺度の中央値) を上回った。よって個人のスケジュールを計算機が生成することは有効な手段であるといえる。また、タスクの入力数が多く、複雑なスケジューリングが要求される被験者ほど評価値が高まる傾向があった。

アクティブスケジューラの適用前と適用後における実行不可能なタスク数を表 11 に示す。Step 0 すなわちアクティブスケジューラ適用前の値はすべてダブルブッキングに起因する。Step 1 の値はアクティブスケジューラでは解決不可能なダブルブッキングの数、およびスケジューリングにより現在時刻よりも前に移動されたタスクの数を含む。表 11 よれば、アクティブスケジューラによって実行不可能なタスクの数は半分以下に減少している。この結果は Visualizer によってユーザに明確に示される。このことが被験者の高い評価につながっているのではないかと考えられる。

表 10 The relevancy of the proposed system

	Average	SD
Step 1	4.5	1.0
Step 2	5.5	1.25

6.2.2 Optimizer および Recommender に関する評価実験

この実験では、Optimizer の多目的最適化アルゴ

表 11 The number of unexecutable tasks

	Average
Step 0	7.3
Step 1	3.1

リズムおよび Recommender によってよりよいスケジュールが提示されているかどうかを検証する。手順は、前節の実験後に AHP の評価行列を作成してもらう。この作業には Web ページ上のフォームを用いた。その後携帯端末上から最適化機能呼び出し、計算結果を提示する。計算結果は複数提示される可能性があるが、ここでは Recommender によるランク 1 のスケジュールのみについて評価を行う。被験者には「先ほどのスケジュールと比較してよいといえるか」という質問を行った。評価の方法は前節の実験と同様である。

結果を表 10 の Step 2 に示す。この実験の評価値は 4 を大きく上回った。また、Step 1 の結果よりも良好である。したがって、Optimizer および Recommender の連携によるスケジュールの提示は有効であるといえる。しかしながら、標準偏差が大きいことから、最適化機能が適切に動作する場合とそうでない場合の差が大きいことがわかる。また、AHP の評価行列がどのように候補解の提示に影響するかがわかりづらいために最適化の精度に違和感を覚える被験者がいた。

7. 結 論

本研究では、個人の資源管理のための意思決定支援システムについて検討を行い、タスクスケジューリングを対象としたシステムの構築とそれに対する考察を行った。

人間の意思決定プロセスは主に情報活動、設計活動および選択活動に分割できることから、それぞれの活動に対する支援を行うことが重要である。また、目的が明確でない問題に対しては、長期間にわたってよりよい意思決定を補助するシステムが必要となる。

次に、これらの提案をもとに、タスクスケジューリングを支援するサーバ/クライアント型システムを構築した。情報活動の支援には可視化技術を、設計活動の支援には多目的最適化アルゴリズムを、選択活動の支援には階層型意思決定手法を用いて個人の意思決定を補助する。また、フラットな組織でのグループワークを支援するためにネットワーク分析アルゴリズムを用いた。

このシステムの有効性をいくつかのシナリオに沿って検証した。第一に、タスクスケジューリングにおけ

る最大の問題であるダブルブッキングがシステムによってどのように解決するかを示した。次に、フラットな組織において構成員の重要度がタスク情報にどのように加算されるかを示した。さらに、意思決定の履歴をもとにした学習によって意思決定者の意向を反映させるプロセスについて示した。また、実証実験を行い、システムの有効性を確かめるとともに、多くの意見をを得ることができた。

以上の結果から、本研究での提案は今後の個人の支援の方向性として有効であることが明らかになった。今後はより大きな規模のグループでの実証実験を考えている。

文 献

- [1] Herbert A. Simon. 経営行動. ダイアモンド社, 1989.
- [2] 野口悠紀雄. 続「超」整理法・時間編 タイムマネジメントの新技术. 中央公論社, 1995.
- [3] 塚田晃司, 岡田謙一, 松下温. 情報へのアクセスの形態に着目した情報共有モデルの一提案. 情報処理学会論文誌, Vol. 39, No. 10, 1998.
- [4] 日本オペレーションズ・リサーチ学会(編). OR 用語辞典. 日科技連, 2000.
- [5] 飯島淳一. 意思決定支援システムとエキスパートシステム. 日科技連, 1993.
- [6] NTT ドコモ株式会社. i モード対応 java コンテンツ開発ガイド. 2001.
- [7] D.E.Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, 1989.
- [8] C. A. Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. *Proc. of Congress on Evolutionary Computation*, 1999.
- [9] 安田雪. ネットワーク分析 何が行為を決定するか. 新曜社, 1997.
- [10] 木下栄蔵(編). AHP の理論と実際. 日科技連, 2000.
- [11] Larry Page, Sergey Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
- [12] 産業技術総合研究所. Horb. <http://www.horb.org/>, 2001.
- [13] 岡山理科大学 HORB 研究会. ihorb, 2001.