

TIM: モバイルメッセージを活用した ビジネスシーン向けミーティング管理エージェントの提案

後藤 充裕¹ 中村 浩司¹ 木全 英明¹

概要: 企業における仕事の中でミーティング管理は自らの仕事を円滑に進める上で非常に重要である。しかしながら、ミーティング管理は非常に煩雑な作業であり、限られた仕事の時間を浪費する恐れがある。そこで、本研究ではモバイルメッセージを介してテキストコミュニケーションを取るだけで、ミーティング参加者のスケジュール参照やミーティング日時を設定をサポートするミーティング管理エージェント TIM (meeTing Management agent) を提案する。TIM の特徴は、ミーティング主催者からの送信されたテキストメッセージからミーティング設定に必要な条件を対話的に読み取りながら、ミーティング設定を行う点にある。また、企業で利用されている既存スケジュールの改造なしにミーティング管理機能を追加できる点も特徴である。本稿では、提案するミーティング管理エージェントのプロトタイプシステムを実装し、そのプロトタイプを用いて行なったユーザビリティ評価およびシステムパフォーマンスの検証について報告する。

TIM: A meeTing Management Agent over Mobile Messenger for Business Use

MITSUHIRO GOTO¹ KOJI NAKAMURA¹ HIDEAKI KIMATA¹

1. はじめに

企業での仕事の中でミーティングに参加する機会は多々ある。ミーティングの中では、関係する社内外のプロジェクトメンバに対して、自分の作業の進捗状況を共有しそのフィードバックを受けたり、プロジェクトのこれからの進め方について議論を行なうことが多い。このようなミーティングへの参加は自らの仕事を円滑に進める上で重要である。

一方で、自分が参加するミーティングの管理は非常に煩雑な作業である。現在多くの企業では、社内ネットワーク上から自由にアクセスできるスケジュールシステムを導入し、各メンバや会議室のスケジュールを管理・共有している。しかし、社外からスケジュールにアクセスしてミーティング参加者のスケジュールを確認するには、VPN などのセキュアな接続認証を伴うリモートアクセスアプリケーションを起動する必要がある、手間のかかる作業となる。また、ミーティング開催日時の設定は、参加者や会議室のスケジュール情報を読み取り、開催可能な日時を絞り込んでいく作業となるが、ミーティングの参加者の数が多い場合や多忙なメンバが参加者となった場合、この作業も非常に煩雑な作業となる。加えて、関連するプロジェクトが増えてくると参加すべき、もしくは開催日時を設定すべきミーティングの数も増えることになり、限られた仕事の時間大部分をミーティング管理に費やしてしまう恐れがある。

こういった状況から、多くのビジネスマンは自らのスケジュール管理をサポートする秘書を渴望しているが、人件費の観点から秘書によるサポートは管理職など一部のビジネスマンのみしか享受できないのが現状である。

そこで、本研究では、モバイルメッセージを介して、ユーザとテキストコミュニケーションをしながら、ミーティング参加者のスケジュールの参照やミーティング日時を設定をサポートするミーティング管理エージェント: TIM (meeTing Management agent と、ことわざの「Time Is Money(時は金なり)」から命名) を提案する。TIM の特徴は、ミーティング主催者からの送信されたテキストメッセージからミーティング設定に必要な条件を対話的に読み取りながら、必要なスケジュール情報の参照や、参加者・会議室の空き時間を考慮したミーティング日時を設定を、モバイル端末のみで簡易かつ即時に行なえる点にある(図 1)。

さらに、企業において、既存システムへの機能追加がシステム開発コストの関係から容易に実現できない点を鑑みて、スケジュールシステムの Web 画面の HTML をパースして、スケジュール情報を取得する方式を採用している。この方式により、システム連携用のインターフェース追加などの改造なしに、既存システムへ TIM を追加できる点も特徴である。本稿では、提案するミーティング管理エージェント: TIM のプロトタイプを実装するとともに、ミーティング主催者と TIM とのコミュニケーションに利用するモバイルメッセージインターフェースに関するユーザビリティ評価と既存スケジュールシステムからのスケジュール情報取得時のシステムパフォーマンスを検証した。

¹ NTT コムウェア株式会社 品質生産性技術本部 研究開発部
NTT COMARE CORPORATION Research and Development
Department

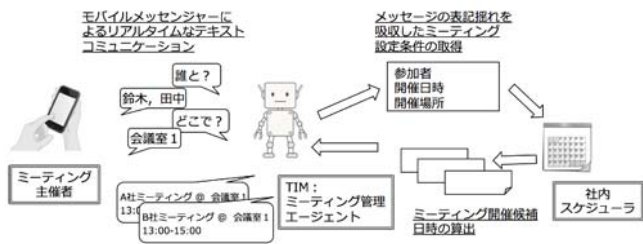


図1 ミーティング管理エージェント：TIMの概要

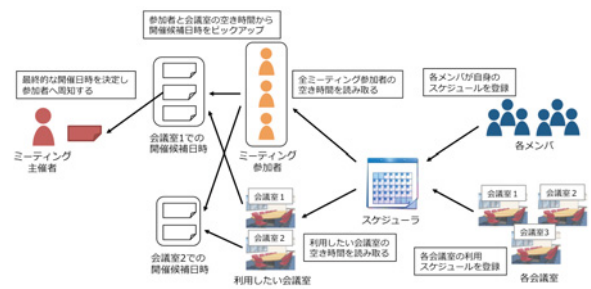


図2 対象とするミーティングの日時設定モデル

2. 関連研究

ミーティング管理に関する研究は、グループウェアカレンダーシステム(GCS)^[1]に関する研究の一環で行われており、その多くは、ミーティング開催の日時設定に関する研究である。これら研究の傾向は大きく分けて2つの方向性があり、1つはミーティング開催日時の設定アルゴリズムに関する研究で、もう一つは日時設定を簡易にするユーザインターフェースに関する研究となる。

2.1 ミーティング日時設定アルゴリズム

このタイプの研究には、ユーザ毎のスケジュールの重要度を把握したエージェントの協議により開催日時を決定する研究^[2]やミーティング設定時の人や場所、投入済みのスケジュールの重要度といった制約条件を制約充足問題として扱い、既に投入されているスケジュールの再スケジュールリングを加味して、新たなミーティングの開催日時を決定する研究^[3]、ユーザ毎の属性・志向に基づき、ミーティングの開催日時を決定する研究^{[4][5]}がある。これら研究で考案された日時設定のアルゴリズムは有用であり、我々が提案するエージェントシステムへ導入することで更なる有用性の向上が見込める。一方で、既にスケジュールへ投入されているデータ以外に各スケジュールの重要度やユーザ毎の属性・志向が必要となるため、既存スケジュールへ導入する場合には、各種データの追加投入を行う作業が発生する。そのため、本研究とは、既存のスケジュールに投入されたデータのみを利用して、ミーティング開催日時を設定する点が異なる。

2.2 日時設定を簡易にするユーザインターフェース

このタイプの研究には、既存スケジュールのユーザインターフェースに組み込む形で機能追加を行う研究やエージェントを操作する新たなユーザインターフェースを提供する研究^{[6][7]}がある。これらのユーザインターフェースはデスクトップ端末での利用が想定されており、モバイル端末で利用するには複雑な構成である。一方、本研究はモバイル端末での利用を想定し、テキスト入力だけでスケジュール設定を完了できる点が異なる。

また、これらの研究では、既存のスケジュールシステムの動作フローを大きく改造して、新たなユーザインターフェースを提供するアプローチを取っており、我々が狙ってい

る既存スケジュールへの改造をせずに、ミーティング管理エージェントを導入するという点も異なっている。

3. ミーティング管理

本章では、ミーティング管理におけるミーティング設定パターンについて整理した後に、本研究で対象にするミーティング設定パターンが抱える課題とその課題解決を図るためのエージェントを介したミーティング設定について述べる。

3.1 ミーティング日時設定パターン

ミーティング日時の設定パターンは、各メンバーのスケジュールの共有状態（共有・未共有）と最終的な開催日時の決定者（主催者・参加者）という2つの観点に基づき4つに大別できる。本節では、各パターンのそれぞれの特徴について述べる。

(1) スケジュール共有-主催者決定型

ミーティング主催者がスケジュールを用いて、ミーティング開催予定日の各参加者や会議室の空き時間をもとに、いくつかの開催候補日時をピックアップする。そして、ミーティング主催者が最終的なミーティング開催日時を決定して、その結果に参加者に周知するパターンである。参加者数が10名程度のプロジェクトミーティング設定で非常に多く用いられるパターンである。

(2) スケジュール共有-参加者決定型

パターン(1)と同じく、ミーティング主催者がスケジュールを用いて、ミーティング開催予定日の各参加者や会議室の空き時間をもとに、いくつかの開催候補日時をピックアップするが、最終的なミーティング開催日時を主催者ではなく、参加者によって決定するパターンである。ミーティング主催者が開催候補日時を参加者に周知し、各参加者が開催を希望する日時への投票による多数決を行ったり、参加者内のキーマン（意思決定者となるメンバーやそのミーティング開催に欠くことの出来ないメンバー）の判断により最終的な開催日時を決定し、その結果を主催者が参加者に周知するパターンである。全員の空き時間が一致する開催候補日時がなく、主催者がどの候補日時で開催するのが良い

か判断できない場合に用いられる。

(3) スケジュール未共有-主催者決定型

ミーティング主催者がミーティング開催を予定しているいくつかの日程の空き時間を各参加者から聞き出し、その空き時間から開催候補日時をピックアップする。そして、主催者が最終的なミーティング開催日時を決定して、参加者に周知するパターンである。スケジュール等でスケジュールが共有できない企業間や別スケジュールシステムを使っていて、お互いのスケジュールが確認できない部署間でのミーティング設定で用いられることが多い。

(4) スケジュール未共有-参加者決定型

パターン(3)と同じく、ミーティング主催者がミーティングの開催を予定している、いくつかの日程の空き時間を各参加者から聞き出して、その時間から開催候補日時をピックアップする。そして、最終的なミーティング開催日時を主催者ではなく参加者の多数決により決定するパターンである。企業内で非常に多くの参加者（例えば、1つの部署丸ごと）を呼ぶ必要のあるミーティング設定時に用いられ、会議調整 Web サービス¹⁴⁾などを利用して、参加者数が最も多くなる日時を開催日とすることが多い。

以上述べてきた様々な設定パターンの内、本研究で主に対象とするのは、企業内での非常に多く用いられるパターン(1)の「スケジュール共有-主催者決定型」である。以降はパターン(1)に関して検討結果について詳述する。

3.2 ミーティング設定時の課題

パターン(1)によるミーティング設定では、図2に示す通りミーティング主催者が次の手順でミーティング開催日時を決定していく。

- 手順 1 ミーティング開催予定日における全参加者と利用したい会議室の空き時間をスケジュールから読み取る
- 手順 2 全参加者の空き時間と各会議室の空き時間からミーティング開催が可能な候補日時をいくつかピックアップする。
- 手順 3 ミーティング開催候補日時から、最終的なミーティング開催日時を決定し、全参加者のスケジュールにその内容を書き込む。
- 手順 4 全ミーティング参加者に対して、Eメールなどを利用してミーティング開催日時や開催場所、参加予定者を周知する。

しかしながら、現状このように行われているミーティング設定では、以下で述べるように、ミーティング主催者の作業負担が高くなるという課題が存在する。

課題 1：詳細な検索キーワードの入力

主催者がスケジュールへ事前に設定している参加者や会議室以外のスケジュールは、参加者名や会議室名などの検索キーワードを用いて、スケジュール内を検索しながら収集していく必要がある。しかし、この検索キーワード検索において、同姓のメンバが複数いる場合や会議室によって人数収容のキャパシティや付属設備が異なる場合には、検索キーワードを詳細に入力して、参加者名や会議室名を正確に指定する手間が増え、ミーティング主催者への負担が大きくなる。

課題 2：開催候補日時の突合

参加者数が多い場合や多忙なメンバが参加者となる場合には、それぞれの空き時間がなかなか合致せず、候補日時を見つけ出す作業負担は高くなる。また、参加者の増減や利用する会議室、開催日時の変更により、ミーティングの再スケジュールリングが必要になった場合には、空き時間の読み取りからミーティング設定をやり直す必要があり、主催者にとって大きな負担となる。

課題 3：アプリケーションスイッチ

ミーティング主催者は様々なアプリケーションを利用してミーティング設定を行う必要がある。例えば、スケジュールの読み出しには Web ブラウザやスケジュールクライアント、空き時間の突合には表計算ソフト、参加者の周知には、Eメールクライアントというように、複数のアプリケーションを切り替えながらミーティング設定を行うことが多い。しかし、アプリケーション間でシームレスな連携が実現されていないため、主催者が全てを手動で行う必要があり作業負担が高くなる。

3.3 エージェントを介したミーティング設定

そこで、本研究ではスケジュールと連携し、ミーティング設定をサポートするミーティング管理エージェントを提案する。本エージェントは、ミーティング設定におけるミーティング主催者の作業負担を軽減するために、モバイルメッセージャーを用いてミーティング主催者とエージェントがコミュニケーションを取るだけで必要なスケジュール情報を収集し、ミーティング開催日時の決定を可能にする仕組みを提供する。

モバイルメッセージャーに着目した理由として、リアルタイムかつ対話的に情報入力が行なえ、システムからの情報表示にも適したユーザインターフェースであることや、近年パーソナルユースに加えてビジネスユースでの利用が増加し、誰もが使い慣れたユーザインターフェースとなったことが挙げられる。また、いくつかのメッセージャーサービスでは、セキュリティに考慮したサービスも登場しており、企業の社内ネットワーク上にあるスケジュールシステムへのリモートアクセスにも適したアプローチであると言える。そして、このようなモバイルメッセージャーを用いることで、ミーティング主催者に入力された情報をもと

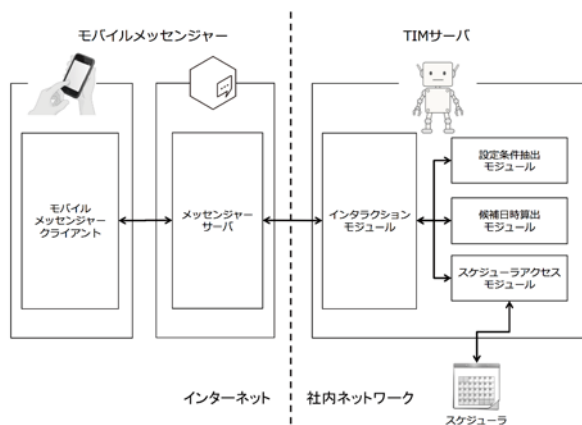


図3 プロトタイプシステムの構成

にエージェントがスケジュール収集したい対象を一意に絞り込めない場合、ミーティング主催者に対して情報の絞り込みを促すようなコミュニケーションを行ったり、情報が誤って入力された場合にも、ミーティング主催者に対して対話的に情報の再入力を求めることができる。これにより、スケジュール収集に必要な参加者名や会議室名といった検索キーワードの入力負担を軽減する(課題1の解決)。ミーティング開催候補日時の突合時にも、本エージェントがスケジュール上の空き時間をマッチングすることで短時間かつ正確に処理できる(課題2の解決)。また、モバイルメッセンジャー上での情報入力と項目選択のみで、ミーティングの開催日時を決定し、そのままシームレスに参加者へのメッセンジャーへ開催通知を行うことで、アプリケーションスイッチに関する作業負担の軽減も期待できる(課題3の解決)。

4. TIM：ミーティング管理エージェント

本研究で提案するミーティング管理エージェント：TIMの概要を図1に示す。TIMは、ミーティングを設定したいミーティング主催者がモバイルメッセンジャー経由でメッセージ送信を行うことで、ミーティング設定をサポートする。まず、TIMはテキストベースのコミュニケーションを行ないながら、ミーティング主催者からミーティングの設定条件(開催予定日と開催場所、参加者)を取得する。次に、TIMは取得した設定条件をもとにして、社内スケジュールから開催予定日の参加者や開催場所となる会議室の空き時間を収集し、ミーティング開催候補日時を算出する。最後に、これらミーティング開催候補日時をミーティング主催者にメッセンジャー上で提示して、最終的なミーティング開催日時の選択・決定を促す。

4.1 テキストコミュニケーション

ミーティング主催者とTIMの間のコミュニケーション

には自然言語で書かれたテキストメッセージが用いられる。ミーティング主催者がミーティングを設定する際には、モバイルメッセンジャーを介してTIMにメッセージを送信する。すると、TIMは「いつミーティングを開催しますか?」、「どこでミーティングを行ないますか?」、「誰とミーティングをしたいですか?」と対話的にミーティングの日時設定に必要な情報の入力を促すメッセージを返信してくる。そして、TIMは、ミーティング主催者が開催予定日と開催場所、そのミーティング参加者名を漏れなく入力するまで対話的な質問メッセージを送信する。

4.2 表記揺れを吸収したミーティング設定条件の取得

TIMは、ミーティング主催者から送信されたメッセージを事前に作成したキーワード辞書を用いて、メッセージに含まれる開催予定日と開催場所、参加者名をキーワードとして取得する。もし、ミーティング主催者が設定条件に関するキーワードを含まないメッセージを送信してきた場合には、TIMは「入力された設定条件を理解できません。もう一度正しく設定条件を入力してください」といったメッセージを返信し、正しい設定条件の入力を促す。

4.3 ミーティング開催候補時間の算出

設定条件が全て入力されると、TIMは開催予定日における全参加者と開催場所に該当する会議室の空き時間をスケジュールから収集する。そして、収集したそれぞれの空き時間をもとに、開催場所単位でのミーティング開催候補日時を算出して、その結果をミーティング主催者へメッセージの形で返信する。ミーティング主催者は返信されたメッセージ内の開催候補日時から、実際にミーティングを開催する日時を1つ選択することで、ミーティング設定を行う。最後にTIMは、選択されたミーティング日時をミーティング参加者毎のスケジュールに書き込むと共に、各参加者のメッセンジャーに対してミーティング開催が予定されたという旨のメッセージを送信して、ミーティング設定のサポートを完了する。

5. プロトタイプシステム

図3に4章で述べたTIMのプロトタイプシステムの構成を示す。プロトタイプシステムは、モバイルメッセンジャーとTIMサーバ、スケジュールから構成される。今回モバイルメッセンジャーには、NTTコムウェア製品であるシャナイン®TALK^[8]を利用した。TIMサーバは、ソフトウェアパッケージであり、社内ネットワーク上のノートPC上で動作し、インターネット上のシャナイン®TALKのメッセンジャーサーバを介して、ミーティング主催者とコミュニケーションを行う。



図4 ミーティング設定時の画面表示例

5.1 インタラクシオンモジュール

本モジュールはインターネット上のメッセージサーバと介して、ミーティング主催者のメッセージクライアントとTIMサーバとの間のメッセージの送受信を管理する。全てのメッセージはセキュアなHTTPSプロトコルを用いてやり取りされるため、スケジュール情報から算出したミーティング開催候補日時をミーティング主催者へ安全に届けられることができる。ただし、本システムを実際に企業で運用するには、企業毎に定められている社内ネットワークのセキュリティポリシーに準じたセキュリティ対策を行う必要がある。

このモジュールを通して、ミーティング主催者は、実在の人物と対話するように、自然言語でテキストメッセージをやり取りする。そして、このモジュールは開催予定日や開催場所、参加者といったミーティング設定条件が全て入力されるまで主催者と対話的にやり取りする。図4にモバイルメッセージクライアント上での画面表示例を示す。図4左部には、ミーティング主催者がミーティング設定条件を全て入力した時の画面を示す。また、図4右部には、このモジュールがミーティング開催候補日時をミーティング主催者に送信し、主催者がその中の1つを選択した画面表示例を示す。

5.2 設定条件抽出モジュール

本モジュールは、ミーティング主催者から送信されたメッセージを分析し、メッセージに含まれるミーティング設定条件を抽出する。設定条件の抽出は以下の手順で行われる。

- 手順1 主催者が送信したメッセージを形態素解析し、名詞と複合名詞のみを検出する。形態素解析エンジンにはMeCab⁹⁾を利用している。
- 手順2 検出した名詞が日時キーワード辞書に含まれる相対的な時間を表す表現(「明日」、「2日後」など)の場合には、絶対的な時間を表す表現(「明日→2016.05.12」、「2日後→2016.05.13」)に変更する。

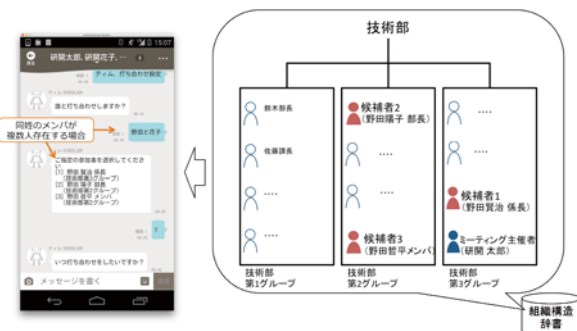


図5 組織構造辞書による人名選択画面の表示例

手順3 検出した名詞が日時キーワード辞書に含まれない場合には、人名・設備名キーワード辞書と突合する。もし、人名・設備名キーワード辞書内に検出した名詞が存在しない場合には、インタラクシオンモジュール経由で、設定条件の再入力を促す。

手順4 手順(3)で人名・設備名キーワード辞書と突合した名詞が複数の人名・設備名で利用されている場合には、組織構造辞書をもとにミーティング主催者に関わっている可能性の高い人物を優先して表示するように画面表示順を決定し、インタラクシオンモジュール経由でミーティング主催者に返信し、人物の選択を促す。

ここで、日時キーワード辞書は、一般的に利用される時を表す相対表現を手動で登録した辞書である。人名・設備名キーワード辞書はWikipediaやWordNetなどの日本語リソースをもとに作成した人名や設備名の候補を学習させた辞書である。さらに、その企業内でよく利用される人名や設備名の手動登録が可能で、TIMを運用する企業ごとにカスタマイズできる。組織構造辞書は、ミーティング主催者と各メンバの関連性を所属部署・グループや役職をもとに登録した辞書である、この辞書により、ミーティング主催者と部署・グループが近い人物を役職の高い順に探すことができる。図5に組織構造辞書を用いて、同姓のメンバが複数存在した場合の人名選択画面の表示例を示す。この例ではミーティング主催者の「研開太郎」が「野田」という人物を指定した際の3名の候補者の表示例を表している。まず、ミーティング主催者と同じグループに所属している「野田賢治 係長」を第1候補者にしており、次に別グループに所属している2名の「野田」を役職順に並べ「野田陽子 部長」を第2候補者に、「野田哲平 メンバ」を第3候補者にして、ミーティング主催者にどの候補者をミーティング参加者とするか選択を促している。

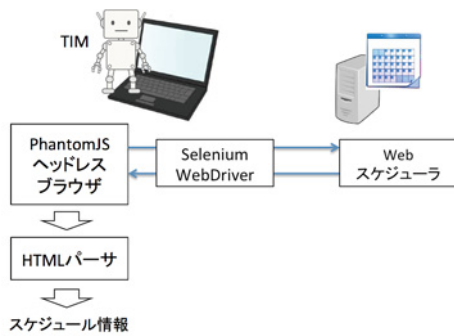


図 6 Web スクレイピングの実装

5.3 スケジューラアクセスモジュール

本モジュールは、社内ネットワーク上のスケジューラからミーティングの全参加者と会議室のスケジュール情報を収集する。このモジュールは Web スクレイピング技術^[10]を利用して既存スケジューラの改造を行なわなくとも、スケジュール情報を収集できるのが特徴である。本プロトタイプでは、Web サービスのテストフレームワークである Selenium WebDriver^[11]を活用して、スケジューラから様々な情報を収集可能にしている。図 6 に Web スクレイピングの実装について示す。WebDriver を用いて、ヘッドレスブラウザである PhantomJS^[12]を制御し、TIM 用のスケジューラアカウントでログイン処理を行い、HTML で書かれたスケジューラの Web 画面を取得する。そして、その Web 画面から HTML パーサを用いて必要なスケジュール情報の抽出を行う。この収集時に TIM はスケジューラシステムの 1 ユーザとして振る舞いながらスケジュール収集行なうため、TIM 用に用意するスケジューラアカウントは企業内メンバが利用する通常アカウントで良い。また、PhantomJS は GUI 表示をしない Web ブラウザで、通常の Web ブラウザを利用する場合に比べて、動作オーバーヘッドが少なく TIM サーバへの負荷を軽減できる。本モジュールは、スケジュール情報を収集するスケジューラに API (Application Program Interface) が用意されている場合には、そちらを利用してスケジュール情報を収集することも可能である。

5.4 候補日時算出モジュール

本モジュールは、スケジューラアクセスモジュールで収集したスケジュール情報をもとにミーティングの開催候補日時を算出する。このモジュールは、収集したミーティング開催予定日の全参加者と各会議室の空き時間を始業時刻（例えば、午前 9 時）から 1 時間単位で利用可能かどうかをチェックする。このチェックを経て、全参加者と会議室が利用可能な時間帯がある場合には、その時間帯をミーティング開催候補日時としてキープする。全ての時間帯についてチェックを行い、1 つでもミーティング開催候補日時がある場合には、インタラクションモジュールを介して、この開催候補日時をメッセージとしてミーティング主催者

に送信する。ミーティング主催者は、TIM から送信された開催候補日時から実際にミーティングを開催したい日時を 1 つ選択する。すると、TIM はその開催日時を参加者全員に周知し、各参加者のスケジューラに反映する。

6. 評価実験

プロトタイプシステムの有用性を確かめるために、ユーザビリティ評価とシステムパフォーマンス検証のそれぞれについて実験結果について報告する。

6.1 実験 1: ユーザビリティ評価の概要

TIM とコミュニケーションするユーザインターフェースとして、モバイルメッセージャーが適切であるかを確認するため、実際の会社員 7 名 (22-50 歳の男性 6 名と女性 1 名) を被験者とした被験者実験を実施した。本実験では、被験者がモバイルメッセージャーと E メールクライアントの 2 種類のユーザインターフェースを利用して TIM とのコミュニケーションを行い、ミーティング設定条件の入力に要した時間や各ユーザインターフェースの使い勝手に関するアンケート結果を比較した。

6.1.1 実験 1: 手順・条件

実験 1 の手順は以下の通りである。

- (1) 被験者に実験手順書に書かれたミーティング設定条件（開催日時、開催場所、参加者）を確認してもらう。
- (2) 被験者にモバイルメッセージャーもしくは E メールクライアントを用いて、TIM とコミュニケーションしてもらい、(1)で確認した設定条件を全て入力してもらう。全ての設定条件を入力すると、TIM が複数の開催候補日時を提示するので、どれか 1 つを選択し、開催日時として決定してもらう。
- (3) 被験者に(1)-(2)の手順に従って、異なる 5 つの設定条件について入力してもらう。

各被験者には、まずモバイルメッセージャーで(1)-(3)の手順に従って、TIM とコミュニケーションを実施し、次に E メールクライアントでコミュニケーションを実施するように指示した。そして、手順(1)のミーティング設定条件の読み込みから手順(2)の開催日時を選択するまでの時間をミーティング設定条件の入力所要時間として、TIM が配置されたサーバで計測し、モバイルメッセージャーと E メールクライアントとの差を比較した。また、上記手順を全て終了した後各ユーザインターフェースを利用した際の TIM の使い勝手に関する 4 種類の項目について 5 段階のアンケート(5 が最も良く、1 が最も悪い)を被験者に記入してもらい、主観的にもユーザインターフェース間の比較を行った。アンケート項目は以下の通りである。

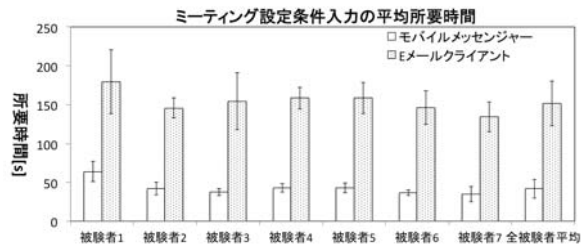


図 7 ミーティング設定条件入力平均所要時間

- Q1. ミーティング日時の設定条件入力に適切なインターフェースだったか？
- Q2. TIM の返答テキストの表示に適切なインターフェースだったか？
- Q3. 入力したテキストと TIM の返答テキストを一覧して見やすかったか？
- Q4. 全体的に TIM とのコミュニケーションに使いやすいインターフェースだったか？

実験に利用したモバイル端末は Nexus 5 (Android 4.4.4) であり、モバイルメッセージャーはシャナイン®TALK (バージョン 2.1)、E メールクライアントは Android の標準メールクライアント (バージョン 6.3) である。この E メールクライアントを採用した理由は、シンプルな E メールクライアントであり、メッセージャーとの情報表示方法の違いを比較するのに適切であると考えたためである。

6.1.2 実験 1: 結果・考察

図 7 にユーザインターフェース毎のミーティング設定条件入力の平均所要時間を示す。全ての被験者において、E メールクライアントを用いるよりも、モバイルメッセージャーを用いた方が短時間でミーティング設定条件の入力が行なえたことが分かる。このような結果となった理由として、モバイルメッセージャーを用いた場合、メールクライアントに比べて TIM とコミュニケーションを行ないながら簡単にミーティング設定条件の入力が行なえたためと考えられる。

図 8 にインターフェース毎のアンケートによる主観評価結果について示す。Q4 の結果を見ると、モバイルメッセージャーの方が、E メールクライアントに比べて約 2 倍のスコアを得ている。この結果より、全体的に見てモバイルメッセージャーを利用することが TIM とのコミュニケーションに有用であるということが示された。以降では、どのような観点でモバイルメッセージャーが有用であるかを考察する。

Q1 の結果から、多くの被験者が設定条件入力のユーザインターフェースとしてモバイルメッセージャーが有用であったと感じていることが分かる。この理由として、4 章で述べた通り、TIM では質問に答える形で、対話的にミーティング設定条件入力していく方式を採っており、人と会

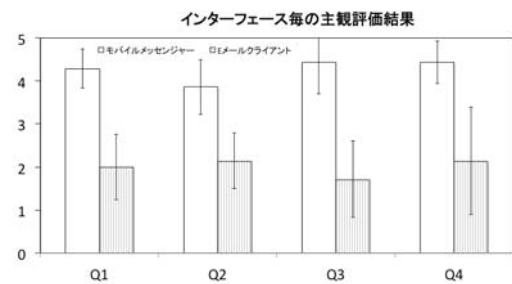


図 8 インターフェース毎の主観評価結果

話するように入力できるモバイルメッセージャーのほうが E メールクライアントに比べて、自然に設定条件を入力できたためであると考えられる。

Q2 の結果は、他の設問に比べて差が少なかった。この理由として、今回 TIM の返信するメッセージは平易なテキスト文であり、ユーザインターフェースの違いが情報の読み取りに大きく影響しなかったためと考えられる。

Q3 の結果は、全ての設問の中で最も差が大きくなった (平均して、モバイルメッセージャーが 4.4 に対して、E メールクライアントは 1.7 となった)、これは、モバイルメッセージャーが画面遷移なしで過去のコミュニケーション履歴を一覧できるためであると考えられる。一方で、E メールクライアントはスレッドを未表示に設定して実験をしており、被験者の送ったメッセージは送信メールボックスに、TIM の送ったメッセージは受信メールボックスに蓄積されるようになっている。そのため、過去のコミュニケーション履歴を振り返るには、毎回、画面遷移を伴い、それぞれのボックスを行き来しながら確認する必要があり、被験者が煩雑に感じたためだと考えられる。

6.2 実験 2: システムパフォーマンス検証の概要

TIM の特徴の 1 つとして、既存スケジューラを改造せずに、ミーティング管理エージェント機能が追加できるが、その際に 1 つのエージェントがどれだけのミーティング主催者をカバーできるかというスケーラビリティが重要となると考えている。そこで、システムパフォーマンスを処理時間の観点から評価した。プロトタイプシステムは複数のモジュールで構成されているが、その中でスケジューラアクセスモジュールの処理が最も支配的であることが分かった。そこで、スケジューラアクセスモジュールの Web スクレイピングに要する処理時間を計測し、スケーラビリティについて考察した。

6.2.1 実験 2: 手順・条件

実験 2 の手順は以下の通りである。

- (1) 指定した参加者人数の 1 日分のスケジュール情報をスケジューラから収集する。なお、スケジューラの負荷を考えて、スケジュール情報収集後は 30 秒のスリープ処理を入れている。

- (2) スケジュール情報を収集する人数を 2 名→4 名→8 名→16 名→32 名→64 名→128 名と増やしながら、各指定人数でのスケジュール収集の処理時間を計測する。
- (3) 各指定人数でのスケジュール情報の収集を 5 回ずつ実行し、その平均時間を算出する。

実験に利用した PC は MacBookPro (Core i5 2.9 GHz, RAM 16GB, SSD 256GB) であり、スケジュール情報の収集はシングルスレッドで動作させている。

6.2.2 実験 2: 結果・考察

図 9 に収集する参加者数を変化させた場合の Web スクレイピングによる平均収集時間を示す、この図より、参加者 8 人分までのスケジュール収集では、処理時間が 10 秒以内に完了できることが分かる。一方で、8 名以上の参加者のスケジュール収集を行う場合には、参加人数に比例して、処理時間が増加することが分かった。この理由として、現在の実装では、参加者のスケジュール収集を行う場合に、「TIM アカウントでスケジューラにログイン→必要な Web 画面を取得→HTML パーサでスケジューラ情報の抽出」という流れで処理する。そして、複数人のスケジュール収集には、この処理を繰り返している。そのため、何度もスケジューラへのログイン処理が発生している。また、このスケジュール収集処理はシングルスレッドで実行しているため、参加者数が増えるほど繰り返し数が多くなっていき、処理時間を要する。TIM をビジネスシーンで実際に利用するには、10 秒程度の処理時間でスケジュール設定を行なえないと、ミーティング主催者が許容できないと考えており、今後は Web 画面取得と HTML パーサによるデータ抽出を複数の参加者数分まとめて実行したり、複数の TIM で同時にスケジュール取得を行って、高速化していくことが求められる。

7. まとめ

本研究では、参加すべきもしくは設定すべきミーティングを多く抱えるビジネスマンを対象にミーティング管理をサポートするミーティング管理エージェント: TIM の提案を行なった。TIM の特徴は、既存のスケジューラへの改造なしに、ミーティング管理機能を追加できると共に、広く普及しているモバイルメッセージングのユーザインターフェースを用いて、実際の人と対話するようにミーティングの設定が行なえる点にある。これらの特徴は、実際のビジネスシーンに導入する際に非常に重要である。また、提案するエージェントシステムのプロトタイプを実装し、そのユーザビリティの評価およびシステムパフォーマンスの検証を行なった。この実験結果より、既存の E メールクライアントを利用するよりもモバイルメッセージングを利用した方が有用であるということが主観的にも客観的にも示された。システムパフォーマンスの観点では、実用に向けて

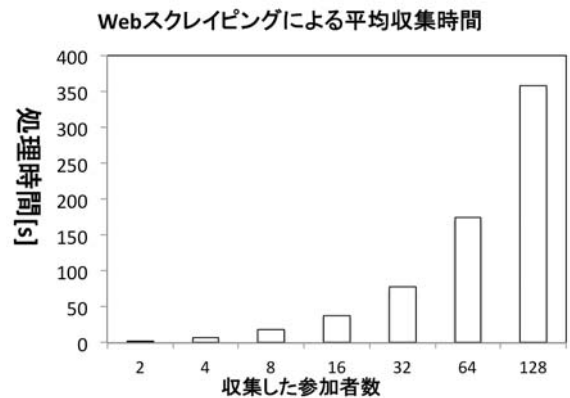


図 9 Web スクレイピングによる平均収集時間

Web スクレイピングによるスケジュール収集処理に更なる高速化が必要であることが判明した。今後は、分散環境でのスケジュール収集処理の高速化や、複数のエージェントを利用して効果的なミーティング管理の方法について検討していく。

参考文献

- [1] Leysia, P.: *Social, individual and technological issues for groupware calendar systems*, In Proc.of CHI '99. ACM, New York, NY, USA, 17-24 (1999).
- [2] Jacques, W., et al.: *Scheduling meetings through multi-agent negotiations*, Decision Support Systems. vol. 44, no. 1, pp. 285-297 (2007).
- [3] Tsuruta, T., et al.: *Scheduling meetings using distributed valued constraint satisfaction algorithm*, In Proc. EC AI (2000).
- [4] Megasari, R., et al.: *Towards host to host meeting scheduling negotiation*, Int. J. Adv. Intell. Informatics, vol. 1. no. 1. 23-29, (2015).
- [5] Pauline, M., et al.: *PTIME: Personalized assistance for calendaring*, ACM Trans. Intell. Syst. Technol. 2, 4, Article 40 (July 2011), 22 pages (2011).
- [6] Pragnesh, J, M., et al.: *CMRadar: a personal assistant agent for calendar management*. AOIS'04, Springer-Verlag, Berlin, Heidelberg, 169-181 (2004).
- [7] Doodle. 入手先(<http://doodle.com/>)(2016.05.09)
- [8] シャナイン®TALK. 入手先(<https://www.nttcom.co.jp/shanain/>)(2016.05.09)
- [9] MeCab: Yet another part-of-speech and morphological analyzer. 入手先(<http://mecab.sourceforge.net/>)(2016.05.09).
- [10] R, B, Penman., et al.: *Web scraping made simple with sitescraper* (2009).
- [11] Selenium WebDriver. 入手先(<http://www.seleniumhq.org/>)(2016.05.09)
- [12] Phantom JS. 入手先(<http://phantomjs.org/>)(2016.05.09)