

ユーザの計画立案行動を監視する カレンダーシステムの検討

中園 武尊¹ 乃村 能成¹

概要：我々は将来の予定を計画する際に過去の予定を参考にすることがある．このとき，参考にすべき過去の予定を保持しているツールとして，カレンダーシステムがある．カレンダーシステムに登録された予定には，その予定の重要度や関連する予定との前後関係といった属性がある．

しかし，既存のカレンダーシステムでは，カレンダーを見ただけではこれらの情報が分からない．そこで，ユーザの計画立案行動といった，カレンダー上で行う操作の履歴を用いることでこれらの情報を抽出する手法を提案する．具体的には，ユーザのカレンダーの操作を監視し，記録することで，予定間の関係や重要度を推定する．

キーワード：カレンダー，計画立案，操作履歴

1. はじめに

我々は普段，将来の予定について計画する作業（計画立案）を行うとき，過去の予定を参考にすることが多い．たとえば，昨年行った予定の実施日時を参考にして今年のおおよその実施時期を検討したり，周期的に発生する予定において，前回の実施日時から次回の実施日時を検討したりする．

このとき，予定の管理によく用いられるシステムとして，カレンダーシステムがある．我々は普段，将来行う予定をカレンダーに登録することで管理している．このカレンダーシステムの例として，Google カレンダー [1] がある．Google カレンダーに登録されている予定は，予定の名前，開始日時，終了日時，場所，参加者，および予定の説明に関する情報をもつ．ユーザは計画立案の際，これらの情報を参照できる．

しかし，既存のカレンダーシステムで参照できる情報には，予定の重要度や予定間の相互の関係といった計画立案に有用だと思われる情報までは記録されていない．たとえば，既存のカレンダーシステムでは，予定を作成した日時の情報を参照できない．計画立案を行うユーザにとって，予定を作成した日時は重要な情報となる．いつ，誰が，どの予定を作成したのかという情報を得ることで，同様の予定について計画立案を行うとき，どの程度余裕をもって計画立案

を行うべきかが分かる．予定の作成の他にも，予定の変更や削除の操作履歴を用いることで，予定の優先度や関連性の推定が可能となる．

本稿では，ユーザの計画立案行動を監視するカレンダーシステムについて検討する．まず，ユーザの計画立案行動とカレンダーシステムにおける操作の関係について述べる．次に，計画立案行動を監視するシステムについて述べる．

2. ユーザの計画立案行動

2.1 ユーザの計画立案行動とカレンダーシステムにおける予定との関係

ユーザの計画立案行動の例として，予定の作成がある．予定の実施日だけでなく作成日も分かれば，計画立案をいつまでに行うべきかを把握できる．ここで，著者らが利用している共有カレンダーを対象に 2019 年 4 月 1 日から 2020 年 3 月 31 日までに登録されている予定の作成日と実施日の間隔について表 1 に示す．表 1 から，約 1 カ月以内の周期で発生する予定については，間隔日数が周期の日数と似通っていることが分かる．これは，予定終了時に次の予定を計画するためである．一方，周期が約 1 年の予定は，間隔が短い予定では 3 日前，間隔が長い予定では 83 日前に予定を作成している．

83 日前に予定を作成している合同研究会は，複数の大学の研究室が合同で行うイベントである．規模の大きな予定であるため，他の予定よりも前もって計画立案されていることが分かる．作成日と実施日の間隔が短い勉強会 2 や勉強

¹ 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology,
Okayama University

表 1 予定ごとの間隔の調査結果

通番	周期	予定名	平均間隔 (日)
1	約 2 週間	ミーティング 1	15.5
2		ミーティング 2	13.7
3		勉強会 1	16.6
4	約 1 カ月	ミーティング 3	27.7
5		ミーティング 4	26.0
6		談話会	21.9
7	約 1 年	勉強会 2	6
8		勉強会 3	6
9		勉強会 4	6
10		勉強会 5	9
11		お花見	18
12		BBQ	8
13		暑気払い	33
14		研修会	41
15		オープンキャンパス	38
16		合同研究会	83
17		B4 中間発表	18
18		M2 中間発表	65
19		M1 論文紹介	12
20		M2 論文紹介	28
21		忘年会 1	66
22		忘年会 2	50
23		卒論締切	56
24		卒論発表	63
25		修論締切	54
26		修論発表	61
27		書き初め	25
28		新 B4 ガイダンス	3
29		新 B4 歓迎会	3
30		入学式	6
31	年に 2, 3 回 (不定期)	レクリエーション 1	22.5
32		レクリエーション 2	27.5
33	なし	その他	22

強会 3 等は、研究室に新しく配属された学生のためのイベントである。年度のはじめに配属されてから計画立案を開始するため、間隔日数が短いことが分かる。

このように、その予定の規模や特徴によって、計画立案を開始する時期は異なる。予定の作成日を確認することで、予定の作成日と実施日の関係から、その予定の規模や重要度、および特徴を推定できる。

この他に計画立案行動と予定の関係として、以下が考えられる。

(1) 予定の変更理由の推定

ユーザの計画立案行動を用いる方法として、予定の変更理由の推定がある。ユーザの計画立案行動を読み取ることで、予定の変更理由を推定できる。たとえば、予定 A の実施日時に新規の予定 B が作成され、既存の予定 A の実施日時が変更されたとき、予定の重複によって変更されたという理由を推定できる。

また、ある予定が削除されたとき、予定を削除したという計画立案行動の履歴が残っていれば、作成されていた予定が中止となったため削除されたのか、もともと予定が存在しなかったのか判断できる。

(2) 予定の関連性の推定

ユーザの計画立案行動から、予定の関連性が推定できる。たとえば、予定 A を計画立案する際に予定 B を参照したことが分かれば、予定 A と予定 B が同様の内容であることが推定できる。また、予定 C の変更にともない予定 D に同じ変更を行っていた場合、予定 C と予定 D は関連性の強い予定だと推定できる。

(3) 変更前の予定の参照

ユーザの計画立案行動を読み取ることで、変更前の情報を参照できる。計画立案時に参照した予定について、過去に予定の情報が変更されていることがある。このような予定を参照するとき、変更前の情報はユーザにとって重要な情報となる。たとえば、参照した予定の実施日時が変更されていた場合、変更前の予定実施日時は、その時点においてもっとも適当日時であったと推定できる。

このとき、計画立案を行うユーザは、変更前の実施日時と変更後の実施日時を比較することでより適当日時を検討できる。

2.2 ユーザの操作履歴と計画立案行動との関係

ユーザは計画立案を行う際、カレンダーでさまざまな操作を行う。このときのユーザの操作から、ユーザの計画立案行動を推定できる。ユーザのカレンダー上で行う操作と計画立案行動との関係およびそれらから分かる各予定のもつ特徴について、以下に考察する。

(操作 1) 予定の閲覧

ユーザがどの予定をいつ閲覧したかという履歴から、次の計画立案時にどの予定を参照するべきかが分かる。たとえば、今年の忘年会の予定を計画する場合、カレンダーから去年の忘年会の予定を探し、その実施日時を確認する。そして、去年の忘年会の実施日時を参考に今年の忘年会の実施日時について検討する。このように、ユーザが計画立案を行うとき、過去に実施した同様の予定を閲覧することがある。

(操作 2) 予定の作成

ユーザがどの予定をいつ作成したかという履歴から、計画立案を行った時期が分かる。また、複数の予定の作成日時を比較することで、予定の優先順位を推定できる。

たとえば、過去に実施した予定が 8 月中旬に作成されていた場合、次の計画立案は 8 月中旬までに行えば良いことが分かる。また、2.1 節で述べたように、規

模の大きな予定や重要な予定は、比較的余裕をもって予定を作成する。このため、同時期に実施する予定でも、作成日時が古い予定の方が優先度が高いと推定できる。

(操作 3) 予定情報の変更

ユーザは、カレンダーに作成した予定に対して、情報の変更を行うことがある。ユーザが行う予定情報の変更内容として、以下の操作が考えられる。

(操作 3-1) 予定の名前の変更

予定の名前を変更した操作履歴から、変更理由の推定ができる。

たとえば、ある時間帯に実施する予定を予定 A から予定 B に変更することがある。このとき、ユーザは予定 A の名前を予定 B に変更することで、予定 B をカレンダーに作成する。この操作履歴を確認することで、もともとは予定 A が作成されていたことや実施される予定が予定 A から予定 B に変更されたことが推定できる。

(操作 3-2) 予定の実施日時の変更

予定の実施日時を変更した操作履歴から、予定の関連性を推定できる。

2.1 節の (2) で述べたように、複数の予定で同じ変更の操作が行われたとき、これらの予定は関連性があると推定できる。たとえば、「打合せ会議」の翌日に「議事録送付」という予定があるときに、「打合せ会議」が 1 週間後に変更されて「議事録送付」も 1 週間後に変更された場合、この 2 つの予定は関連性のある予定だといえる。

(操作 3-3) 予定の場所の変更

予定の場所を変更した操作履歴から、その場所における予定の優先度がどちらが高いかを推定できる。

たとえば、会議室 A という場所で、グループ A とグループ B が、同じ日時にそれぞれミーティングを行いたいとする。このとき、どちらかのグループが、予定の実施時間、あるいは場所を変更することで予定を調整することが考えられる。ここでは、グループ B が、ミーティングの場所を会議室 A から会議室 B に変更することで予定を調整したとする。このとき、会議室 A の方が収容人数が多い、使用できる設備が多いといった条件があるとき、会議室 A においてはグループ A のミーティングの方が優先度が高いことが考えられる。

(操作 3-4) 予定の参加者の変更

予定の参加者を変更した操作履歴から、参加者の観点からその予定の特徴を推定できる。

例として、あるグループが参加者を募る形式で予定を計画する場合について考える。まず、グループのメンバーを予定の参加者として登録した予定を作成する。その後、予定の参加者を募り、応募してきた人を参加者として追加することが考えられる。また、参加を辞退した場合は参加者からその人を削除する。予定の参加者の情報について、変更回数が多いければ例のように参加者の変動が激しい予定だということが分かる。逆に参加者の変更回数が少ない場合、予定の参加者がある程度固定された予定だということが分かる。

(操作 3-5) 予定の繰返し規則の変更

予定には、「毎週水曜日」や「毎月 20 日」といった規則に基づいて繰返し発生する予定がある。繰返し規則を変更した履歴から、その予定の優先度や進捗度合いを推定できる。

たとえば、繰返し規則が「毎月 20 日」から「毎月 10 日と 20 日」や「毎週水曜日」といったように、繰返し規則の変更により予定の発生頻度が変化する場合がある。これは、変更前の繰返し規則では予定の実施回数が不十分であると判断されたため、発生頻度が上がるように変更したのだと推定できる。

(操作 3-6) 予定の繰返し期間の変更

予定の繰返し期間を変更した操作履歴から、予定の進捗度合いを推定ができる。

たとえば、半年で終わると想定した予定は、繰返し期間を半年とする。このとき、繰返し期間が延長されていた場合、進捗が想定より悪かったのだと推定できる。逆に繰返し期間が短縮されていた場合、進捗が想定より良かったのだと推定できる。

これらの操作について、変更内容を操作履歴として記録することで、いつ、誰が、どの予定をどう変更したか確認可能となる。

(操作 4) 予定の削除

どの予定をいつ削除したかといった操作履歴から、ユーザは、削除された予定も計画立案時に参照できる。また、カレンダー上には存在しない予定が、もともと計画立案を行わなかった予定なのか、計画立案を行ったうえで、中止になったためカレンダー上から削除された予定なのかが分かる。

既存のカレンダーシステムでは、ユーザに表示される情報だけではこれらの操作履歴について確認できない。図 1 のようにカレンダーに登録されている予定の情報が変更されても、ユーザはどのような変更が行われたか把握できない。たとえば、図 1 は、ある予定の実施日時が 3 週間前に変更

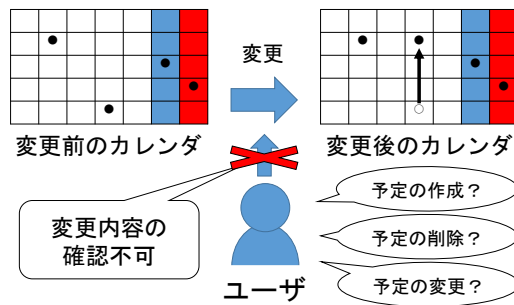


図 1 予定の変更内容が確認できない例

されたことを意味する。このとき、ユーザは操作履歴を参照できないため、このカレンダーにどのような変更が行われたのか確認することができない。

3. ユーザの計画立案行動の監視

3.1 提案システムの構成

本章では、カレンダー上で行うユーザの操作を監視し、更新が行われた予定を取得するシステムの設計と実装について述べる。提案システムの構成を図 2 に示す。提案システムは、以下の処理部から構成される。

監視部

カレンダーを監視し、カレンダー情報に変更があった場合、取得部に変更を通知する。

取得部

更新された予定を取得し、予定 DB に保存する。

予定 DB

取得した予定の情報を保存する。

解析部

取得した予定の更新内容を解析し、更新内容を更新履歴 DB に保存する。

更新履歴 DB

予定情報の更新内容を保存する。

表示部

更新内容を表示する。

3.2 処理の流れ

提案システムがカレンダー情報の変更を監視し、変更内容を保存する流れを図 2 を用いて説明する。

ユーザは、計画立案にともないカレンダー A に予定の作成・編集・削除を行う (1)。監視部は、カレンダー A に対する変更を監視し、変更があれば取得部に更新が行われたことを通知する (2)。取得部は、カレンダーから更新された予定を取得し、予定 DB に追加する (4)(5)。更新された予定について、更新前と更新後の情報を予定 DB から取得する (6)。解析部は取得した情報から更新内容を解析し、更新履歴 DB に保存する (7)。表示部は更新履歴 DB から更新内容を取得し、更新内容を表示する (8)。

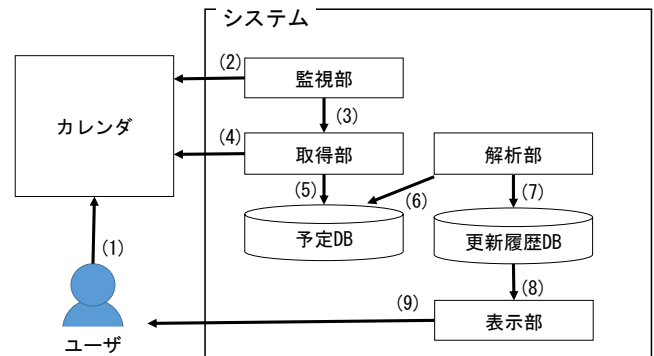


図 2 提案システムの構成

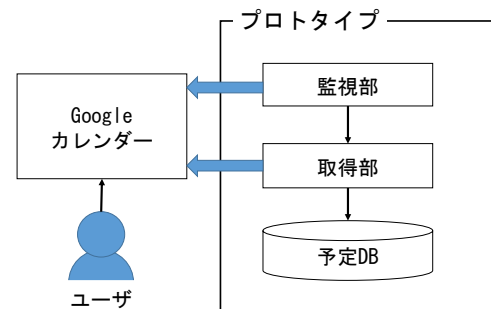


図 3 プロトタイプの構成

3.3 実装

提案システムのプロトタイプを実装した。プロトタイプは、Google カレンダーを対象に監視、および予定の取得を行う。実装したのは、図 2 のうち監視部、取得部、および予定 DB である。また、プロトタイプは著者らが所属する研究室で運用しているサーバに実装した。プロトタイプの構成を図 3 に示す。構成内容を以下で説明する。

監視部

監視部は Google Calendar API[2] を用いて実装した。Google Calendar API には Push Notifications 機能があり、指定した Google カレンダーに対して変更が行われたとき、Webhook を利用して変更を通知する。

取得部

取得部は、Google Calendar API を用いて、変更が行われた予定を取得する。このとき、syncToken というパラメータを用いることで、前回予定を取得した時の Google カレンダーと差分を取り、変更が行われた予定のみ取得する。取得する予定のデータ形式は JSON 形式となっている。

予定 DB

JSON 形式で取得した予定のデータを保存する。

実装したプロトタイプによって、変更が行われた予定を Google カレンダーから取得し、予定 DB に保存できることを確認した。

3.4 取得した予定の情報

プロトタイプによって取得した予定の情報のなかで、計

表 2 提案システムを利用して取得した予定の情報のなかで
計画立案行動にともなう変更の可能性がある項目

通番	項目名	概要	関係する操作
1	created	予定の作成日時	(操作 2)
2	updated	予定の更新日時	(操作 3)
3	status	予定の状態	(操作 4)
4	summary	予定の名前	(操作 3-1)
5	start	予定の開始日時	(操作 3-2)
6	end	予定の終了日時	(操作 3-2)
7	location	予定に関する場所	(操作 3-3)
8	attendees	予定の参加者	(操作 3-4)
9	recurrence	予定の繰り返し規則	(操作 3-5) (操作 3-6)
10	recurringEventId	属する繰り返し予定の ID	(操作 3)

画立案行動によって変更される可能性のある項目を表 2 に示す。

表 2 の通番 1 から通番 6 の要素は、すべての予定に存在する要素である。通番 7 と通番 8 の要素は、予定ごとに任意に設定する項目であり、情報が入力されている場合のみ存在する。また、通番 9 と通番 10 は、繰り返し発生する予定にのみ存在する要素である。

以下に各項目についての説明および 2.2 節で述べた操作のうちどの操作により変更されるかを述べる。

(1)(2) created と updated

予定の作成日時と更新日時を示す。(操作 2) により予定を作成したとき、1 件分の予定を表すデータが生成される。(操作 3) により予定の更新を行ったとき、updated の値が書き換わる。

(3) status

予定の状態を示す。デフォルトでは「confirmed」が設定されている。(操作 4) により削除された予定の場合、status は「cancelled」を示す。Google カレンダーにおいて、カレンダー上で削除された予定は、データベース上では status が「cancelled」に変更された状態で存在する。このとき、このデータがもつ要素は、status の他にその予定のエンティティタグを示す etag、識別子を示す id、およびそのデータのタイプを示す kind のみである。summary 等の情報は消失しているため、削除する前の予定の情報を得るためには、同じ id をもつ過去の予定のデータを保存しておいて、id を照合して削除する前のデータを参照する必要がある。

(4) summary

予定の名前を示す。(操作 3-1) により変更される。

(5)(6) start と end

予定の開始時刻と終了時刻を示す。繰り返し発生する予定の場合、最初の予定の開始時刻と終了時刻を示す。(操作 3-2) により変更される。

(7) location

予定に関する場所を示す。オプションで設定する項目で、情報をフリーフォームテキストで指定する。(操作

表 3 recurrence がもつ要素

通番	要素名	概要
1	RRULE	繰り返しの規則
2	EXRULE	例外の規則
3	RDATE	規則化できない繰り返しの日時
4	EXDATE	例外の日時

3-3) により変更される。

(8) attendees

予定の参加者を示す。(操作 3-4) により変更される。

(9) recurrence

iCalendar フォーマット [3] で規定されている、繰り返し発生する予定の要素を示す。recurrence がもつ要素を表 3 に示す。たとえば、繰り返しの規則を毎週水曜日から毎月 20 日等に変更すると RRULE が変更される。また、繰り返し発生する予定の 1 つを削除すると削除した予定の日時が EXDATE として追加される。recurrence は、(操作 3-5) と (操作 3-6) により変更される。

(10) recurringEventId

その予定が属する繰り返し予定の ID を示す。繰り返し発生する予定の 1 つに (操作 3) を行った場合、recurringEventId をもつデータが新たに生成される。

表 2 に示す通り、予定の閲覧を示す (操作 1) によって変更される項目はない。このため、現在のプロトタイプで取得できる情報からは、ユーザがいつ、どの予定を閲覧したかを確認できない。

3.5 計画立案行動と予定情報との関係

ユーザの計画立案行動とそれによって変更される予定情報の例について述べる。

(1) 実施時間が重複した予定の調整

計画立案において、異なる予定の実施時間が重複することがある。既存の予定 A が実施される日時に新規の予定 B が作成されるとき、予定 A の実施日時を変更して調整するとする。このとき行う操作の組み合わせとして、次の 2 つが考えられる。

(a) 予定の作成と実施日時の変更

新規の予定 B をカレンダーに作成し、予定 A の開始日時と終了日時を変更する場合がある。この操作は、予定 A の start と end の変更履歴、および予定 B の created を参照することで確認できる。

(b) 予定の作成と予定名の変更

変更後の実施日時に予定 A を新たに作成し、予定 A の予定名を予定 B に変更することが考えられる。この操作は、新たに作成した予定 A の created と予定 B の summary の変更履歴を参

照することで確認できる。

(2) 変更前の予定の削除忘れ

ある予定の実施日時を変更するとき、変更後の日時に新たに予定を作成することがある。このとき、変更前の予定の削除を忘れた場合、カレンダー上に同じ予定が2つ存在することになる。2つの予定の created を参照することで、どちらの予定が新しく作成された予定か判断できるようになる。

4. 関連研究

関連研究として、我々は、作業発生の規則性に基づく作業予測手法を提案している [4]。文献 [4] の手法は、過去の作業履歴から、多くの作業が1年を周期として発生することに着目し将来の作業発生を予測する手法である。この手法を用いて、将来の予定を表示する年間カレンダーの開発を行っている。このとき、年間カレンダーが提案する予測結果すべてがユーザに受け入れられるとは考えにくい。実際は、予測結果を参考に日程を修正しながら予定を登録すると思われる。このとき、ユーザが予測結果をどのように修正したかフィードバックを得ることで、次の予測に利用することができる。たとえば、予測結果が複数の予定を同じ時間帯に発生すると予測したとき、ユーザは実施時間が重ならないように実施日時を修正する。このときのユーザの計画立案行動を監視することで、次に同様の予測結果が発生したとき、自動で調整することが可能となる。

また、操作履歴を用いた研究として、操作履歴を基にデータ間関連度を求め、ユーザの操作を契機に過去に参照したデータをユーザに提供するシステムの研究 [5] や、参照や編集といったユーザのファイル操作履歴から関連性を求める研究 [6] 等がある。カレンダーの予定においても、他の予定と関連して発生する予定がある。たとえば、「打合せ」の翌日に「打合せ議事録送付」という予定がある場合、「打合せ」の日程が変更されると「打合せ議事録送付」の日程も関連して変更される。また、予定間の優先度の関係や、参照の関係といった関連性も存在する。カレンダーにおいても、予定の参照や編集の操作履歴から、予定間の関連性を求めることが可能になるとと思われる。

5. おわりに

本稿では、ユーザの計画立案行動を監視するカレンダーシステムについて検討した。まず、我々は普段、過去の予定を参考に計画立案を行うことが多いことを述べた。しかし、既存のカレンダーでは、計画立案行動について記録している情報は不十分であることを述べた。

次に、ユーザの計画立案行動として、ユーザがカレンダーに対し行うであろう操作について述べた。また、この操作を組み合わせることにより、ユーザの計画立案行動が行わ

れることを述べた。

最後に、ユーザの計画立案行動を監視するシステムの構成について述べた。また、プロトタイプを実装した。プロトタイプでは、監視部、取得部、および予定 DB を実装した。これにより、実際に変更が行われた予定を取得できることを確認した。また、実際に取得した予定が保持している情報について述べた。この情報のうち、ユーザの計画立案行動により変更される項目について述べた。

残された課題として、解析部と表示部の実装と予定の参照を監視する手法の検討がある。

参考文献

- [1] Google: Google カレンダー, Google(オンライン), 入手先 (<https://calendar.google.com/calendar>) (参照 2020-8-19)
- [2] Google: Calendar API — Google Developers, Google (online), available from (<https://developers.google.com/calendar>) (accessed 2020-8-19).
- [3] Desruisseaux, B.: Internet Calendaring and Scheduling Core Object Specification (iCalendar), RFC 5545 (2009).
- [4] 吉井英人, 北垣千祐, 乃村能成, 谷口秀夫: 作業発生の規則性に基づく作業予測手法と評価, 情報処理学会論文誌, Vol. 56, No. 2, pp. 543–553 (2015).
- [5] 亮 大澤, 一紀高汐, 英幸徳田: 俺デスク: ユーザ操作履歴に基づく情報想起支援ツール, 第 47 回プログラミングシンポジウム予稿集, Vol. 2006, pp. 15–21 (2006).
- [6] 呉 怡, 渡辺陽介, 横田治夫: RMC 操作に基づくタスクとタスク間関連度を考慮したファイル検索, 電子情報通信学会論文誌, Vol. J96-D, No. 5, pp. 1166–1177 (2013).