

修 士 論 文

題 目

作業発生の規則性を扱う
カレンダーシステムに関する研究

指導教員

報 告 者

三原 俊介

岡山大学大学院 自然科学研究科 電子情報システム工学専攻

平成 24 年 2 月 8 日 提出

要約

オフィス環境での作業の発生には、発生に関してある程度決まった周期性が存在する。また、1つの作業にともなって、関連した別の作業が発生し、一連の流れを形成している。このような各作業間の関連性や周期性を容易に確認できれば、将来の作業発生を予測する際や、一連の作業に関する記録を他者に引継ぐ際の有用な材料となる。

オフィス環境で作業を記録するツールとしてカレンダーシステムがある。しかし、現在多く利用されているカレンダーシステムでは作業発生の規則性を容易に確認できない。この理由として、作業のゆるい周期性を表現する方法がないことと、作業間の関連性を扱えないことがある。

そこで、本論文では「タスク」「リカーレンス」「ミッション」、および「ジョブ」といった作業発生の規則性を扱うためのモデルを提案した。このモデルは、作業のゆるい周期性と作業間の関連性を既存のカレンダーシステムが採用するモデルと親和性の高い形で表現できる。

また、このモデルを扱う「作業発生の規則性を扱うカレンダーシステム」を提案した。このシステムは、提案モデルを iCalendar フォーマットで表現することで、既存のカレンダーシステムと連携する仕組みを持つ。

さらに、作業発生の規則性を用いたユーザ支援の例として、AmReT Calendar を設計し、実装した。AmReT Calendar は、ユーザに負担の少ない形で作業発生の規則性を取得する機能、表示する機能、および作業発生の規則性を基に将来の作業発生を推測する機能を提供する。

そして、AmReT Calendar が既存のカレンダーシステムと比較して計画立案にどの程度有用か評価した。また、作業発生の規則性を扱う場合と扱わない場合を比較して計画立案にどの程度有用か評価した。これらの比較評価により、本システムの有用性を示した。

目次

1	はじめに	1
2	作業発生の規則性	3
2.1	規則的に発生する作業	3
2.2	既存のカレンダシステム	4
2.3	作業発生の規則性を表現するうえでの要求	5
2.4	作業発生の規則性を表現するうえでの問題点	5
2.4.1	作業の関連性の表現	5
2.4.2	作業の周期性の表現	6
3	作業発生の規則性を扱うためのモデル	7
3.1	方針	7
3.2	規則性の表現	8
3.2.1	タスク	8
3.2.2	リカーレンスによるタスク単位の周期性の表現	8
3.2.3	ミッションによる関連性の表現	8
3.2.4	ジョブによるミッション単位の周期性の表現	8
3.3	モデルの適用	8
4	既存のカレンダシステムへの適用	10
4.1	既存のカレンダシステムの構成	10
4.2	iCalender フォーマットへのモデルの適用	10
4.3	既存のカレンダシステムとの連携	12
4.3.1	全体構成	12
4.3.2	CalDAV プロキシ	13
4.3.3	AmReT Calendar	16

5	作業発生の規則性を用いたユーザ支援	17
5.1	AmReT Calendar の設計	17
5.2	周期性と関連性の継承	17
5.3	カレンダーシステムを用いた計画立案	18
5.4	ユーザ支援の実装	19
6	評価	22
6.1	評価の観点	22
6.2	実測評価	23
6.2.1	評価環境の準備	23
6.2.2	評価 1: 過去の予定を再利用した計画立案の速度	24
6.2.3	評価 2: 作業発生の規則性を扱う効果	26
7	関連研究	29
8	おわりに	30
	謝辞	31
	参考文献	32
	発表論文	34

目 次

2.1	規則的に発生する作業の例	4
3.1	本提案のモデルを適用した作業発生の規則性の例	9
4.1	現在主流のカレンダーシステム	11
4.2	iCalendar フォーマットで表現した作業発生の規則性の例	12
4.3	既存のカレンダーシステムとの連携した時の全体構成	13
4.4	既存カレンダー AP から作業発生の規則性を閲覧	14
4.5	既存カレンダー AP から作業発生の規則性を操作	15
5.1	周期性の継承に基づいたタスクの推測の例	18
5.2	関連性の継承に基づいたタスク推測の例	19
5.3	(特徴 1) 過去のタスクを参照しながら複製するタスク登録方式	20
5.4	(特徴 2) 周期性の継承によるタスクの予報機能	21
5.5	(特徴 3) 関連性の継承による関連タスクの一括登録機能	21
6.1	ユーザ A を対象とした AmReT Calendar と既存のカレンダーシステムの比較	25
6.2	ユーザ B を対象とした AmReT Calendar と既存のカレンダーシステムの比較	26
6.3	ユーザ A を対象とした作業発生の規則性を扱う場合と扱わない場合の比較	27
6.4	ユーザ B を対象とした作業発生の規則性を扱う場合と扱わない場合の比較	28

表 目 次

2.1 発生の規則性に関する作業の分類結果	4
6.1 ユーザごとのタスクの分類結果	24

第 1 章

はじめに

オフィス環境におけるさまざまな作業には、発生に関してある程度決まった規則性が存在する。また、1 つの作業にともなって、関連した別の作業が発生し、一連の流れを形成している。たとえば、進捗報告会議や忘年会は、それぞれ「約 2 週間に 1 回」や「毎年 12 月上旬」といった曖昧な周期を持つ。また、それぞれには、会場の予約や議事録の作成といった、事前、事後の作業をともなう。このような各作業間の関連性や周期性を容易に確認できれば、将来の作業発生を予測する際や、一連の作業に関する記録を他者に引継ぐ際の有用な材料となる。

作業を記録するシステムとして、カレンダーシステムの利用を考えた場合、現在利用されているカレンダーシステムの多くは、上記の「作業」に相当するものをうまく管理できない。カレンダーシステムは、「作業」をある時間を占める単発のイベントの集合として管理するため、ある作業に伴って発生する作業間の関係性をうまく表現できないからである。また、「約 2 週間」といった作業発生の曖昧な周期を扱うこともできない。そのため、通常、カレンダーシステムには、近い将来の予定を確認する程度の用途しか期待されておらず、入力済みのデータが有効活用されているとは言い難い。

これまでも、スケジュール情報を処理して今後の作業予測や仕事の引継ぎに利用するシステムの研究は、いくつか存在する [1, 2]。また、仕事引継ぎの観点から過去のスケジュール情報の利用方法を探る研究も現れてきている [3]。しかしながら、これらのシステムが扱う作業間の関連モデルは、作業間にある依存関係をグラフとして表現したり、独自のダイアグラムを用いて行動間の連続関係とつながりの強さを表現するなど、より厳密ではあるものの、既存のカレンダーシステムが扱っているモデルと比べてはるかに複雑であり、既存のカレンダーシステムとの相互運用性が低いという問題がある。そのため、これら研究の成果が広く一般に普及しているとは言い難い。

本稿では、現在広く利用されているカレンダーシステムと親和性の高いシンプルなモデルを採用しつつ、上記の問題を解決する「作業発生の規則性を扱うカレンダーシステム」について述べる。まず、「タスク」、「リカーレンス」、「ミッション」、および「ジョブ」といった作業発生の規則性を既存のカレンダーシステムと親和性の高い形で扱うためのモデルを述べる。次に、モデルに基づき、ユーザに負担の少ない形で関連性や周期性を取得する機能、ユーザにわかりやすい形で関連性や周期性を提示する機能、および過去の周期性や関連性を利用して将来の予定を提案する機能について述べる。さらに、これらの機能を実装したカレンダーシステムの例を示す。そして、カレンダーシステムを評価する観点について述べ、評価結果について述べる。

第 2 章

作業発生の規則性

2.1 規則的に発生する作業

我々が今後の予定について計画するとき、作業の発生はある程度予測することができる。なぜならば、多くの作業は、ある程度決まった規則性に基づいて発生しているからである。

たとえば、部署内での定例会議を考える。図 2.1 は、部署内での定例会議に関連して発生する作業をまとめたものである。定例会議の「会議通知」や「会場準備」といった作業は、他の作業に関連して発生する。また、それぞれの発生するタイミングは、ほぼ決まっている。図 2.1 の例では、「会議の通知は会議の約 1 週間前」や「会場の準備は会議の約 2 時間前」としている。また、図 2.1 の例では、会議は慣例的に約 1 か月周期で発生しており、関連する作業も同じ周期で発生している。

また、表 2.1 は、作業発生の規則性について、作業間の関連性と周期性に着目して、作業を分類したものである。表中右記の件数は、この分類に基づいて、1 人の学生のカレンダ上に記録されていた 1 年分の作業を分類した結果である。

表 2.1 の例では、カレンダに登録されている作業の約 6 割は、関連して発生する作業の時系列を想像できる。また、カレンダに登録されている作業のほぼすべては、過去に発生したことがある。そして、既存のカレンダシステムで扱えるような固定的な周期で発生している予定は約 3 割あり、「約 2 週間に 1 回」や「12 月末頃」といった曖昧な周期で発生している予定は約 6 割ある。

これら事例より、作業の発生にはある程度決まった関連性と周期性があると考えられる。この作業発生に関わる作業の関連性と周期性を組み合わせたものを作業発生の規則性と呼ぶ。作業発生の規則性に基づき作業を確認できれば、将来の作業予測や仕事引継ぎ時の情報伝達に有用だと考えられる。

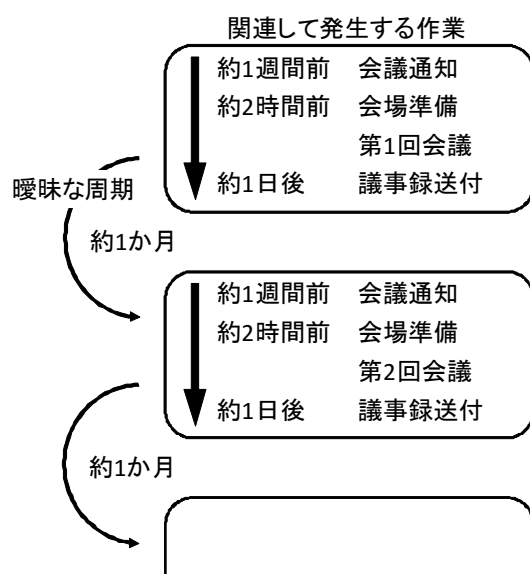


図 2.1 規則的に発生する作業の例

表 2.1 発生の規則性に関する作業の分類結果

分類	308 件中の件数
(1) 関連して発生する作業がある	269 件
(A) 関連して発生する作業を想像できる	254 件
(a) 関連して発生する作業の時系列を想像できる	183 件 (59.4 %)
(2) 過去に発生したことがある	300 件
(A) 固定的な周期性がある	90 件 (29.2 %)
(B) 曖昧な周期性がある	193 件 (62.6 %)

2.2 既存のカレンダーシステム

本提案では、作業を扱う最小の単位として、タスク (*Task*) を定義する。タスクは開始時刻と終了時刻を持ち、この間で連続的に行われる作業を表現する。また、個々のタスクは開始時刻による順序関係を持ち、タスクの全集合は時系列上での有界な集合である。

いわば、既存のカレンダーシステムとは、上記のタスクの集合を利用者に操作閲覧させるシステムだと考えられる。

2.3 作業発生の規則性を表現するうえでの要求

作業発生の規則性を表現するうえで、以下の 2 つの要求がある。

(1) 既存のカレンダシステムのモデルと親和性が高いこと

既存のカレンダシステムは、時間軸上に開始時刻と終了時刻の線を引き、その間をタスクとして管理するというシンプルなモデルを扱っている。このモデルとの差分を小さくすることで、既存の多くのカレンダシステムに適用しやすくなる。

(2) 規則性の曖昧さを許容できること

現実の作業の発生は、他の予定の影響を受けて「約 1 か月ごと」や「12 月末頃」といった曖昧な周期であることが多い。また、作業順序が前後する場合や、過去 1 回のタスクで済んでいた作業が複数回に分かれて行われる場合も多い。

2.4 作業発生の規則性を表現するうえでの問題点

2.4.1 作業の関連性の表現

作業の関連性を表現する方法として、アローダイアグラムやフローチャートを使ってタスクの関係性を表現する方法が考えられる。つまり、個々のタスク間について、依存関係や順序関係を定義する方法である。しかし、依存関係や順序関係をシステムで扱うには、以下の問題が存在する。

(1) 人間が扱うには複雑過ぎ、計算機が自動抽出するには曖昧過ぎる

モデルが複雑になると、ユーザがそれを直接操作することは現実的ではない。そのため、既存システムでは、計算機がそれらの関連性や依存関係を自動抽出することを試みるが、これには必ず誤りが存在する。この手の誤りは、わずかであってもシステムの利用価値を著しく下げる。また、関連する作業の中での個々のタスクの依存関係は、ユーザから見た場合に自明であることが多いため、その誤りは、人間にストレスを与える。

(2) 既存のカレンダシステムが扱うモデルと親和性が低い

2.2 節で述べたように、既存カレンダが扱うモデルは、タスクの集合という非常に単純なものであるため、依存関係を持ち込むと、これらとの連携が考慮しづらい。

よって、関連性を表現する場合には、2.3 節で述べたように既存のカレンダシステムと親和性が高いシンプルな表現を新たに用意する必要がある。

2.4.2 作業の周期性の表現

既存のカレンダシステムでは、繰り返し発生するタスクを表現するとき、1 つのタスクに固定的な周期を与えることで表現する。しかし、この表現方法では現実の作業を扱ううえで問題となる以下の 2 つを表現できない。

(1) 曖昧な周期

既存のカレンダシステムで扱えるのは「1 か月ごと」や「毎年 12 月 31 日」といった固定的な周期だけである。

(2) 作業内容の変化

同様の作業でも実際に繰り返す際には、毎回多少の差異が発生すると考えられる。しかし、この表現方法では、個々の繰り返しに応じて変化する作業間の差異を表現できない。

よって、周期性を表現する場合には、2.3 節で述べたように現実で発生する曖昧さを許容する表現を新たに用意することが必要である。

第 3 章

作業発生の規則性を扱うためのモデル

3.1 方針

以下の 2 つの方法により作業発生の規則性を表現する．

(1) 関連性はタスクの集合で表現する

カレンダーシステム上の個々のタスクは開始時刻と終了時刻という時間情報を持っている．よって，関連した作業の集合をカレンダーシステムで扱うだけで，関連するタスクの順序関係は自明となる．ただし，この方法ではタスクの依存関係はカレンダーシステムで扱われない．しかし，タスクの依存関係は利用者から見れば自明であることが多いため，本モデルではタスクの依存関係を扱わない．また，個々のタスクの依存関係や順序関係を直接扱うことに比べ，シンプルな構造で表現できる．

(2) 周期性は同様のものの集合で表現する

繰り返される作業を表現する際，1 つのタスクに固定的な周期を与えて表現するのではなく，繰り返しごとに個別のタスクとして扱い，集合を用いて繰り返しを表現する．そして，繰り返し発生する同様のものの集合に基づき周期を計算する．

次節では，上記の方針に基づき，作業発生の規則性を表現するための諸概念について定義する．

3.2 規則性の表現

3.2.1 タスク

2.2 節で述べたように，作業を扱う最小の単位はタスクである．タスクは開始時刻と終了時刻の間に連続的に行われる作業を表現する．そして，個々のタスクは，開始時刻により順序関係を持つ．

3.2.2 リカーレンスによるタスク単位の周期性の表現

タスク単位の周期性を扱うモデルとして，リカーレンス (*Recurrence*) を定義する．リカーレンスはタスクを要素とする集合である．リカーレンスは繰り返し発生している同様のタスクを 1 つの集合とする．

3.2.3 ミッションによる関連性の表現

関連性を扱うモデルとして，ミッション (*Mission*) を定義する．ミッションはタスクまたはミッションを要素とする集合である．ミッションは関連する複数のタスクまたはミッションを 1 つの集合とする．ミッションは関連するタスクやミッションを集合としてまとめることで，複数の時間に分かれて実行される大きな粒度での作業を表現する．

3.2.4 ジョブによるミッション単位の周期性の表現

ミッション単位の周期性を扱うモデルとして，ジョブ (*Job*) を定義する．ジョブはミッションを要素とする集合である．ジョブは繰り返し発生している同様のミッションを 1 つの集合とする．

3.3 モデルの適用

図 3.1 は，前節のモデルを 2.1 節の例に適用したものである．まず，「会議通知」，「会場準備」，「第 1 回会議」，および「議事録送付」はタスクとする．次に「会議通知」，「会場準備」，「第 1 回会議」，および「議事録送付」の発生は関連していることを表現するために，これらのタスクを要素とするミッション「第 1 回会議関連の仕事」を定義する．同様に，「第 2 回会議関連の仕事」を定義する．さらに，各「会議通知」や各「会場準備」は曖昧な周期で発生

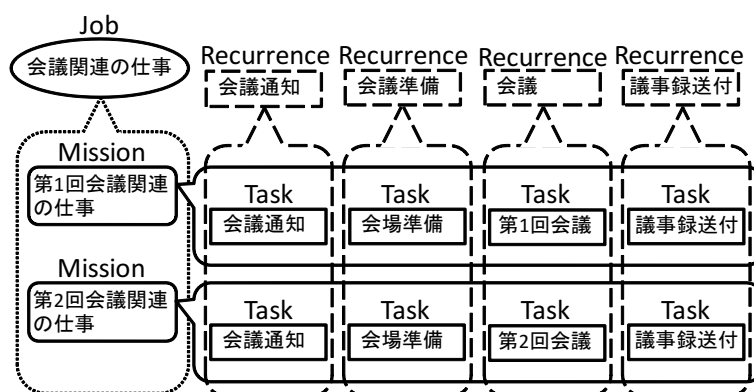


図 3.1 本提案のモデルを適用した作業発生の規則性の例

する同様のタスクであることを表現するために、それぞれリカーレンスを定義する。そして、「第1回会議関連の仕事」と「第2回会議関連の仕事」が曖昧な周期で発生する同様のミッションであることを表現するために、これらのミッションを要素とするジョブ「会議関連の仕事」を定義する。

第 4 章

既存のカレンダーシステムへの適用

4.1 既存のカレンダーシステムの構成

既存の主流となっているカレンダーシステムの構成について述べ、3 章のモデルを扱うためにシステムをどのように設計すべきか述べる。

図 4.1 は、現在主流のカレンダーシステムの構成を示したものである。図 4.1 で示すとおり、カレンダーアプリケーション（以降、カレンダー AP と略す）とカレンダーサーバは CalDAV[4] を使って iCalendar フォーマット [5] のカレンダー情報を共有している。これ以外の利用形態を持つカレンダーシステムに、WEB ブラウザを利用する Google カレンダー [6] や Yahoo! カレンダー（米）[7] がある。ただし、両者とも iCalendar フォーマットをデータ交換フォーマットとして利用でき、CalDAV にも対応している。

よって、CalDAV で扱う iCalendar フォーマット上で 3 章のモデルを表現できれば、本提案システムは既存の多くのカレンダーシステムと連携できる。

4.2 iCalender フォーマットへのモデルの適用

3 章のモデルを iCalendar フォーマットで表現する方法について述べる。

まず、タスクの表現方法について説明する。iCalendar フォーマットでは、カレンダー上の 1 つの予定が 1 つの VEVENT コンポーネントとして記述される。この VEVENT コンポーネント 1 つを 1 つのタスクとする。

次に、リカーレンス、ミッション、およびジョブの表現方法について述べる。リカーレンスは、各タスクの VEVENT コンポーネント同士を兄弟関係にして表現する。ミッションは、

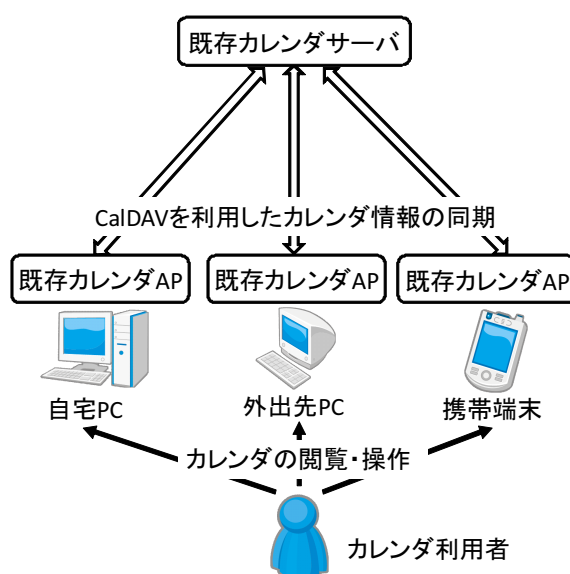


図 4.1 現在主流のカレンダーシステム

VEVENT コンポーネントとして定義し、ミッションの VEVENT コンポーネントとその集合の要素となる VEVENT コンポーネントを親子関係にして表現する。ジョブは、ミッションの VEVENT コンポーネント同士を兄弟関係にして表現する。

これらの親子、兄弟関係を表現するために、VEVENT コンポーネントの RELATED-TO プロパティを使用する。RELATED-TO プロパティは、PARENT（親）、CHILD（子）、および SIBLING（兄弟）といった属性と対象 VEVENT のユニークな ID を持つ。

ただし、RELATED-TO プロパティを使用する場合の問題として、次の 2 点がある。

(1) 他のシステムでカレンダー情報として保持されないことがある

RELATED-TO プロパティは、現在確認できる限り、どのカレンダーシステムでも利用されていない。また、システムによっては、RELATED-TO プロパティを破棄してしまう。たとえば、現時点の Google カレンダーは RELATED-TO プロパティを破棄してしまうことを確認している。

(2) 他のカレンダーシステムが別の解釈をする可能性がある

今後、RELATED-TO プロパティを利用するカレンダーシステムが開発された場合、RELATED-TO プロパティの解釈の違いで競合する可能性がある。

これらの問題に対応するために、iCalendar フォーマットで定義されている独自プロパティ用拡張子 X- を利用し、独自にプロパティを定義する方式が考えられる。しかし、一部のカレ

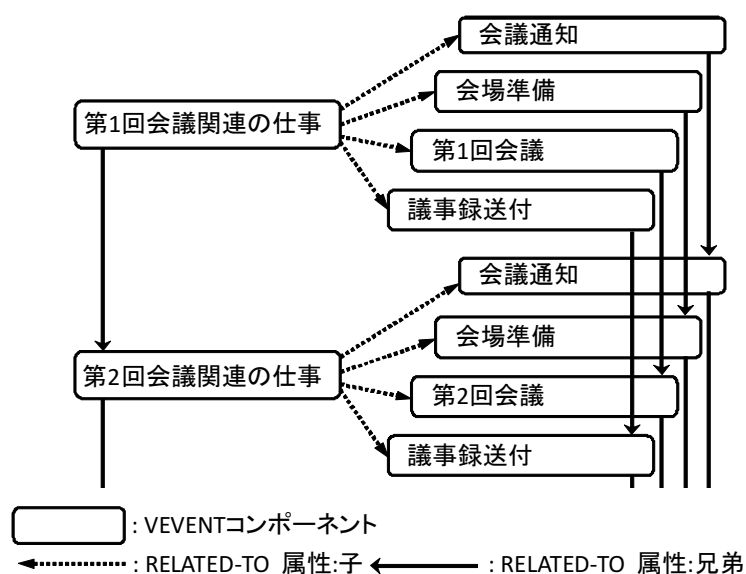


図 4.2 iCalendar フォーマットで表現した作業発生の規則性の例

ンダシステムは、この独自のプロパティすら保持しない。現時点の Google カレンダーは独自プロパティを保持しない。つまり、上記(1)の問題に対して RELATED-TO プロパティを利用する場合と変わらないので、ここでは、独自プロパティ拡張による方式を採用しない。

図 4.2 は、3.3 節の例を iCalendar フォーマットで表現したものである。「第 1 回会議関連の仕事」の VEVENT コンポーネントは、「会議通知」と「会場準備」、「第 1 回会議」、「議事録送付」の VEVENT コンポーネントに対して RELATED-TO プロパティで親子関係を持つ。また、それぞれの VEVENT コンポーネントは、同様の内容の VEVENT コンポーネントに対して RELATED-TO プロパティで兄弟関係を持つ。

4.3 既存のカレンダシステムとの連携

4.3.1 全体構成

図 4.3 は、本システムが既存のカレンダシステムとの連携した時の全体構成である。システムは、以下の 4 つの要素により構成される。

(1) 既存カレンダー AP

既存カレンダー AP を利用する。ただし、CalDAV に対応しているものに限る。

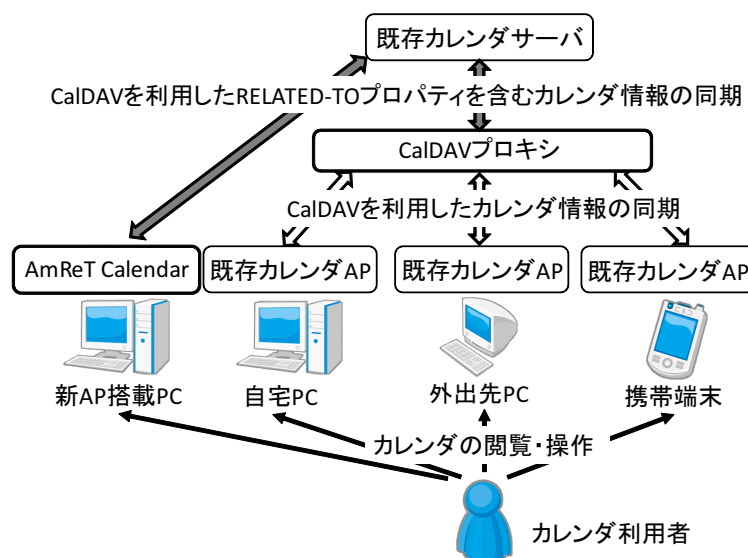


図 4.3 既存のカレンダーシステムとの連携した時の全体構成

(2) 既存カレンダーサーバ

既存カレンダーサーバを利用する。ただし、CalDAV に対応しており、RELATED-TO プロパティを破棄しないものに限る。

(3) CalDAV プロキシ

CalDAV での通信を中継し、RELATED-TO プロパティに基づきミッションやジョブ単位で CalDAV の「カレンダー」を提供する。

(4) AmReT Calendar

RELATED-TO プロパティを解析し、作業発生の規則性に基づきユーザを支援する機能する。また、RELATED-TO プロパティを直接記述できる機能を持ち、作業発生 of 規則性を直接操作するユーザインタフェースを持つ。

CalDAV プロキシ (3) と AmReT Calendar (4) に関しては、次項以降で詳しく説明する。

4.3.2 CalDAV プロキシ

RELATED-TO プロパティが扱えない既存カレンダー AP で作業発生 of 規則性の操作と閲覧を可能にするため、CalDAV プロキシを提案する。CalDAV プロキシは、既存カレンダーサー

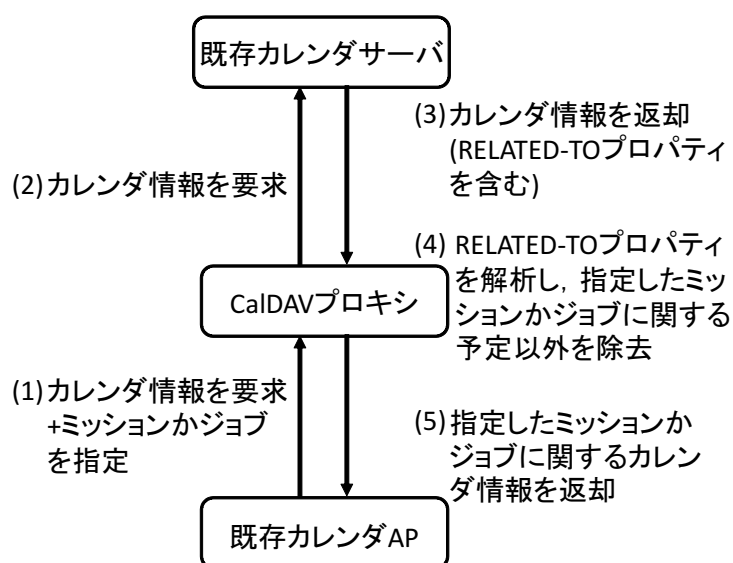


図 4.4 既存カレンダー AP から作業発生の規則性を閲覧

バからのカレンダー情報に含まれている RELATED-TO プロパティをモデルに従って解析し、既存カレンダー AP にミッションやジョブを 1 つのカレンダーとして集約した形で提供する。具体的には、以下の 2 つの機能を持つ。

(1) CalDAV を中継する機能

既存カレンダー AP に対して、リクエストを受信し、リクエストに応じた適切なカレンダー情報をレスポンスとして送信する機能を持つ。また、既存カレンダーサーバに対して、カレンダー情報を要求するリクエストを送信し、カレンダー情報が含まれたレスポンスを受信する機能を持つ。

(2) ミッションやジョブ単位のカレンダーを作成する機能

既存カレンダー AP からのカレンダー情報要求のリクエストを受信した場合、URL にミッションやジョブを指定するパラメータが含まれるとき、指定されたミッションやジョブに関するカレンダー情報のみをカレンダー AP に提供する。また、既存カレンダー AP がこのカレンダーに対して操作したとき、CalDAV プロキシは既存カレンダーサーバへの中継時に RELATED-TO プロパティを埋め込む。

図 4.4 と 図 4.5 は、CalDAV プロキシを利用して、既存カレンダー AP から作業発生の規則性を扱う手順の図である。

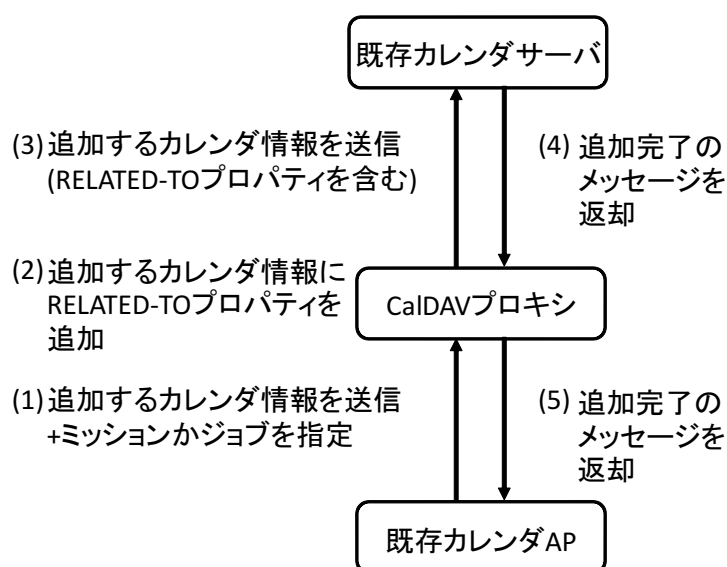


図 4.5 既存カレンダー AP から作業発生の規則性を操作

既存カレンダー AP からミッションやジョブを閲覧する場合（図 4.4）（1）カレンダー情報を要求するとき、既存カレンダー AP はミッションやジョブを指定する情報を含ませる。（2）CalDAV プロキシはミッションやジョブの情報を抜き出して、既存カレンダーサーバにカレンダー情報を要求する。（3）既存カレンダーサーバは CalDAV プロキシに要求されたカレンダー情報を返却する。（4）CalDAV プロキシはカレンダー情報内にある RELATED-TO プロパティを解析し、既存カレンダー AP から指定されたミッションやジョブに関するカレンダー情報以外を削除する。（5）CalDAV プロキシは既存カレンダー AP に指定したミッションやジョブに関するカレンダー情報を返却する。

既存カレンダー AP からミッションやジョブを追加する場合（図 4.5）（1）カレンダー情報を追加するとき、既存カレンダー AP はミッションやジョブを指定する情報を含ませる。（2）CalDAV プロキシはミッションやジョブの情報に対応した RELATED-TO プロパティをカレンダー情報に追加する。（3）CalDAV プロキシは追加するカレンダー情報を既存カレンダーサーバに送信する。（4）（5）既存カレンダーサーバは追加完了のメッセージを CalDAV プロキシを通して既存カレンダー AP に返却する。

4.3.3 AmReT Calendar

CalDAV プロキシを用いても既存カレンダー AP 上では、作業発生の規則性をユーザにわかりやすく見せるのは難しい。よって、RELATED-TO プロパティを解釈する AmReT Calendar を用意し、ユーザが直感的に作業発生の規則性の操作と閲覧ができるようにする。

このカレンダー AP に実装する具体的な機能については、次章以降で述べていく。

第 5 章

作業発生の規則性を用いたユーザ支援

5.1 AmReT Calendar の設計

作業発生の規則性を用いた具体的なユーザ支援の手法について述べる．以降で述べる具体的なユーザ支援の手法は，4.3.3 項で述べた AmReT Calendar 上に実装する．

5.2 周期性と関連性の継承

作業発生の規則性を用いて将来の作業予測を行うための継承という操作について述べる．リカーレンスを構成するタスクは，時系列に並べた時にある程度決まった周期性を持っていると考えられる．この周期性は，リカーレンス内の最後に発生したタスクと将来発生するタスクの間にも適用できると考えられる．これを周期性の継承と呼ぶ．

図 5.1 は，周期性の継承に基づいたタスクの推測の例である．リカーレンス「会議」では，「第 1 回会議」と「第 2 回会議」の発生から次に発生する「第 3 回会議」を推測している．同様に，他の 3 つのリカーレンスにおいても，次のタスク発生を推測している．これらのタスクの次回発生日時は，リカーレンス内の周期性の継承によって推測される．

また，繰り返し発生するタスクは，関連して発生するタスクもある程度決まっていると考えられる．この関連性は，将来発生するタスクにも適用できると考えられる．これを関連性の継承と呼ぶ．

図 5.2 は，関連性の継承に基づいたタスク推測の例である．図 5.2 では，先行する 2 つのミッションから関連性の継承により，次に発生するミッションとタスクを推測する．

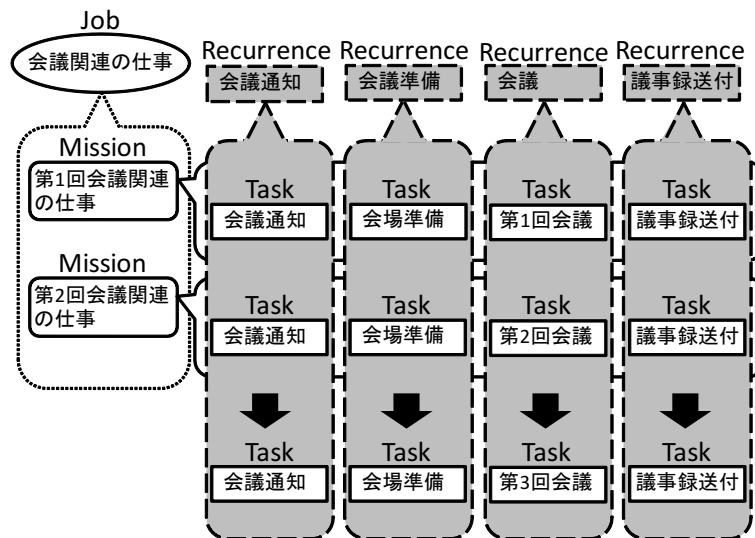


図 5.1 周期性の継承に基づいたタスクの推測の例

5.3 カレンダシステムを用いた計画立案

我々は、多くの場合に過去の履歴を基にして将来の計画を立てる。この際、過去の履歴が登録されているカレンダーシステムを参照しながら計画を立てることがしばしばある。たとえば、来月の計画を立てる際には、昨年の同じ月のカレンダーを確認しながら、今年も発生するタスクを書き写す。あるいは、1つのタスクに関連して発生するであろう一連のタスクを時系列で確認する。

しかし、既存のカレンダーシステムは、これらの作業を意識したデータ構造やユーザインタフェースを備えていない。従って、直近に発生する作業の単なる忘備録としては有用であるものの、過去の作業を振り返り、将来の計画立案を支援するには不十分であるといえる。

対して、AmReT Calendar では、以下の3つの特徴を持つ。

(特徴1) 過去のタスクを参照しながら複製するタスク登録方式

具体的には、1年前の同月のタスクを参照しながら、簡便な複製操作で当月のタスクを作成する登録方式を指す。この操作により登録されたタスクは、複製元のタスクと同じリカーレンスとして登録される。このため、本登録方式は、ユーザのタスク登録操作そのものを容易にするだけでなく、今後のタスク予測に有用なリカーレンス情報を同時に収集可能である点においても意義がある。

(特徴2) 周期性の継承によるタスクの予報機能

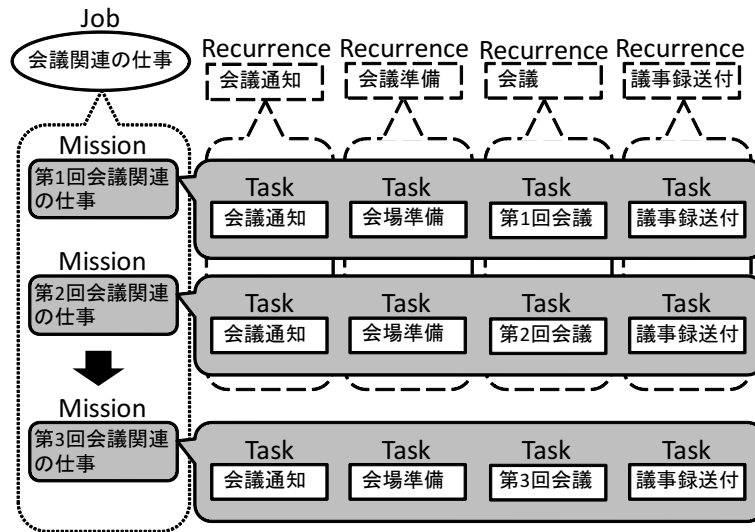


図 5.2 関連性の継承に基づいたタスク推測の例

本機能は、5.2 節で示した周期性の継承に基づいて、近い将来発生しうるタスクを予測して提示する機能である。リカーレンスに基づく周期性継承は、(特徴 1) で示した複製の励行によって暗黙的に蓄積されるため、予報に関する情報を陽に与える必要がないことも特徴である。

(特徴 3) 関連性の継承による関連タスクの一括登録機能

ミッションの情報を利用して、ミッションに属する全タスクを将来の予定として展開して登録する機能である。5.2 節で示した関連性の継承を利用して、ミッションそのものを明示的に複製する。これをトップダウンな複製方式と称する。あるいは、あるミッション A 内の 1 つのタスクを (特徴 1) に示した複製によって登録した場合、ミッション A そのものの複製を促す。これをボトムアップな方式と称する。これらの方式により、一度の複製操作によって多くの関連タスクを同じタイムグリッドを維持したまま登録可能であり、同時に関連タスクの登録漏れを防止する。また、この操作の結果からリカーレンスとミッションの継承関係を暗黙的に収集可能である。

5.4 ユーザ支援の実装

5.3 節で述べた特徴を AmReT Calendar にどのように実装するかについて述べる。



図 5.3 (特徴1) 過去のタスクを参照しながら複製するタスク登録方式

図 5.3 は、(特徴1) で述べた複製によるタスク登録方式を実装した画面である。主として操作するカレンダーの横に過去のカレンダーを表示し、ドラッグアンドドロップすることでタスクを複製できる。そして、タスクは、複製元のタスクと同じリカーレンスとして登録される。

図 5.4 は、(特徴2) で述べたタスクの予報機能の画面である。リカーレンスに基づいて周期性を解析した結果、その月に発生すると推測されるタスクを右側にリスト表示する。ユーザは、このリストからドラッグアンドドロップで適切な日時に予定を登録することができる。

図 5.5 は、(特徴3) で述べた関連タスクの一括登録機能の画面である。リカーレンスを持ったタスクをカレンダーに登録したとき、ミッションに基づいて関連したタスクをカレンダーに一括登録する。



図 5.4 (特徴 2) 周期性の継承によるタスクの予報機能



図 5.5 (特徴 3) 関連性の継承による関連タスクの一括登録機能

第 6 章

評価

6.1 評価の観点

将来発生すると推測されるタスクをその発生時期を考慮しながらカレンダーに仮に登録していく作業をここでは計画立案と呼ぶ。計画立案の作業は、おおよそ以下の手順で行われると想定する。

- (1) 計画立案したい月のカレンダーと前年の同月のカレンダーを並べて開く
- (2) 前年のカレンダーの各タスクについて
 - (A) そのタスクが、再び発生するかどうか判断する
 - (B) 発生すると判断したならば、立案したい月のカレンダーに複写登録する
- (3) 計画について納得できるように調整し確認する

これを必要な月について繰り返す。

ここで注意すべきは、この計画立案の作業においてのゴールは、立案時点でユーザが納得できる計画を作成することであり、結果、登録されたタスクが、実際のタスク発生と一致するかどうかは、問題としないことである。なぜなら、実際のタスク発生とその日時は直近にならないと決まらないため、計画立案の段階では、正確な日時はわからないからである。つまり、計算機やユーザによる推測の精度を問題としていない。

ここで問題とすべきは、ユーザが (2-A) について十分な時間をかけて納得できる計画を立てた場合の、(2-A) にかかる時間と (2-B) の複写登録にかかる時間である。

AmReT Calendar の特徴がこれらの観点とどう対応するかを考える。(特徴 1) の複製機能は、(2-B) の時間短縮に主に効果が現れると考えられる。また、(特徴 2) の予報機能と(特徴 3) の一括登録機能は、(2-A) に影響し、立案初期の段階の登録速度(初速)に主に効果が現れると考えられる。なぜならば、作業発生の規則性を扱うことで(2-A) の判断を支援できると考えられるからである。

これらを踏まえて、まず、AmReT Calendar の 3 つの特徴によって、計画立案作業が、どの程度短縮されるかを評価する。具体的には、(評価 1) として以下を実施する。

(評価 1) AmReT Calendar と既存のカレンダーシステムで経過時間ごとの登録件数の変化を比較する

また、実際の計画立案作業は、限られたごく短い時間で実施する(打ち切られる)ことを考慮すると、立案開始から短時間でも納得できる計画に近い立案が可能か否かも問題となる。これは、タスク登録件数が立案開始時から早い段階で理想の件数に漸近する(初速が大きい)ことと関係する。この初速は、(特徴 2) と(特徴 3) の有効性の度合を示しているといえる。また、(評価 1) では、一般的な既存のカレンダーシステムと AmReT Calendar 双方のユーザインタフェースの差(主に(特徴 1) で生じる)が登録速度の差にとって支配的である可能性を排除できない。そこで、(評価 2) として以下を実施する。

(評価 2) AmReT Calendar において(特徴 2) と(特徴 3) を有効にした場合と無効にした場合で計画立案の初速を比較する

これらによって、AmReT Calendar の有効性を評価する。

6.2 実測評価

6.2.1 評価環境の準備

まず、実験に用いた初期データについて述べる。実験協力者は、ユーザ A とユーザ B の 2 名である。実験協力者には、あらかじめ過去 2 年間の予定を AmReT Calendar と Google カレンダーの双方に登録してもらった。加えて、本カレンダーシステムでは、昨年分の予定についてリカーレンス、ミッション、およびジョブの関係を手動で与えてもらった。

表 6.1 は、各実験協力者のタスクを分類したものである。ユーザごとの一昨年のタスク数、昨年のタスク数、および計画立案で登録されたタスク数をあげている。これは Google カレ

表 6.1 ユーザごとのタスクの分類結果

ユーザ	一昨年		昨年		計画立案	
	総数	ミッション内	総数	ミッション内	総数	ミッション内
ユーザ A	164	-	291	27	152	26
ユーザ B	487	-	454	140	342	140

ンダも同様である．また，昨年のタスクのうち AmReT Calendar において何らかのミッションに属していたタスクの数をあげている．この数は，(特徴 3) の一括登録機能が有効に働く可能性のあるタスク数であるといえる．

表 6.1 より，過去 2 年間のタスク数を比較すると，ユーザ B はユーザ A の約 2 倍ある．また，ユーザ B は，ミッション内に含まれるタスクが全体の約 3 分の 1 を占めている．このことから，ユーザ B は，複数のタスクが関連しながら発生している度合いが大きいユーザである．

6.2.2 評価 1: 過去の予定を再利用した計画立案の速度

実験では，Google カレンダーと AmReT Calendar において，実験協力者に時間無制限で次の 1 年間の計画立案をしてもらった．計画立案は，3 章で述べた計画立案の方法に基づいてカレンダーにタスクを登録することで行う．Google カレンダーでのタスク登録では，別のモニタに昨年のカレンダーのウィンドウを開き，それを参照しながら登録してもらった．また，定期的な周期で発生する講義のようなタスクは，Google カレンダーの繰り返し登録機能を利用して登録してもらった．計画立案の様子をビデオに録画し，システムのデータベースと照合しつつ，時間あたりのタスクの登録件数を測定した．

図 6.1 と図 6.2 は，ユーザ A とユーザ B の各カレンダーを利用した開始後 30 分間における計画立案の比較である．横軸は経過時間，縦軸は各時点でのタスクの登録件数である．各時点でのタスクの登録件数は，常に AmReT Calendar を利用した場合が Google カレンダーを利用した場合と比べて多い．

また，図 6.1 と図 6.2 から以下の 3 つのことが分かる．

- (1) ユーザ A に関して，1 分あたりの登録件数は，AmReT Calendar で平均 5.1 件，Google カレンダーで平均 2.9 件であり，AmReT Calendar の登録速度は Google カレンダーの登録速度の約 2 倍である．

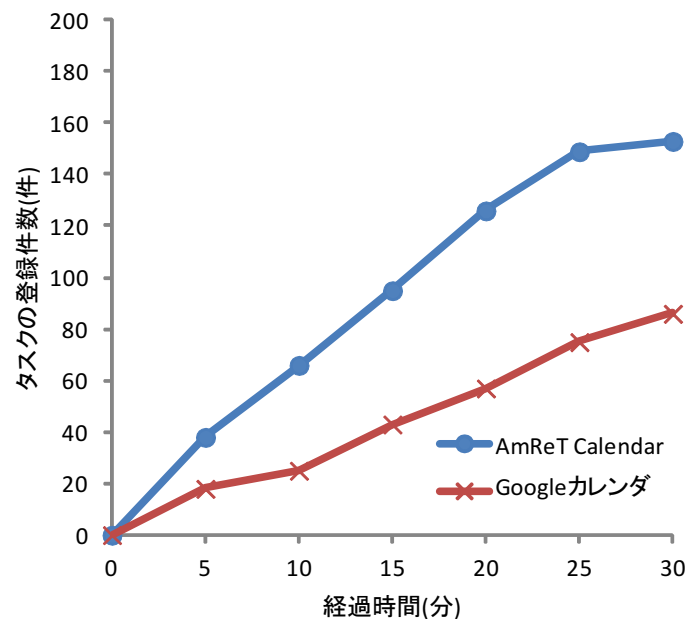


図 6.1 ユーザ A を対象とした AmReT Calendar と既存のカレンダーシステムの比較

- (2) ユーザ B に関して、開始直後 5 分間で、AmReT Calendar では計画立案で登録したタスクの約 25 パーセント、Google カレンダーでは計画立案で登録したタスクの約 15 パーセントが登録されている。
- (3) ユーザ B に関して、開始直後 5 分間を除いた 1 分あたりの登録件数は、AmReT Calendar で平均 3.5 件、Google カレンダーで平均 3.1 件であり、AmReT Calendar の登録速度と Google カレンダーの登録速度はほぼ同じである。
- (1) に関して、AmReT Calendar は Google カレンダーと比べて計画立案を効果的に支援できていると考えられる。ただし、この効果は (特徴 1) の複製機能の影響が大きく、(特徴 2) と (特徴 3) はあまり影響していない。なぜならば、表 6.1 より、ユーザ A はタスクの総数や過去のミッション内のタスクが多くないため、(特徴 2) と (特徴 3) で過去のタスクが再び発生するかどうかの判断を支援する効果が少ないと考えられる。(2) の原因として、ユーザ B は、AmReT Calendar における (特徴 3) の一括登録機能と Google カレンダーにおける繰り返し登録機能を利用し、多くのタスクを登録したことがある。(3) に関して、ユーザ B はユーザ A と比較して予定数が多かったため、実装上の問題で (特徴 2) の予報機能の処理に時間がかかり、AmReT Calendar のレスポンスが遅くなっていることが観測された。

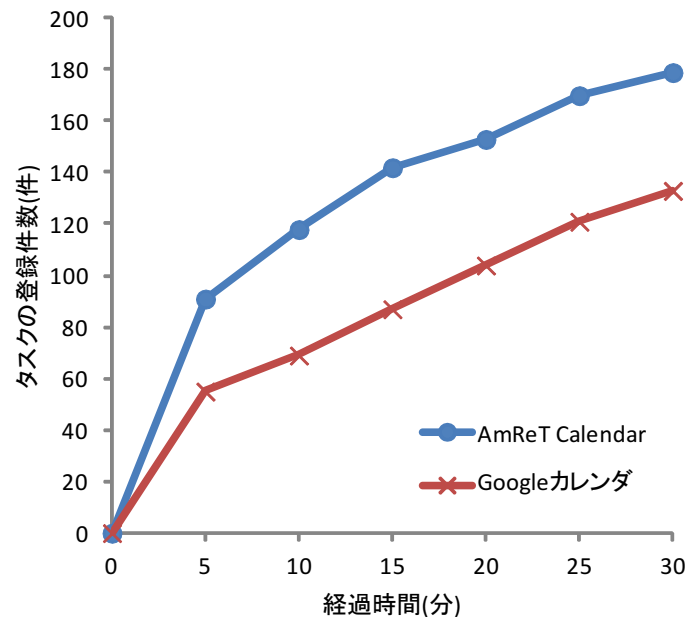


図 6.2 ユーザ B を対象とした AmReT Calendar と既存のカレンダーシステムの比較

以上のことから，AmReT Calendar は，ユーザによって効果の程度は違うものの，計画立案において既存のカレンダーシステムと比べてユーザ支援ができていると考えられる．

6.2.3 評価 2: 作業発生の規則性を扱う効果

作業発生の規則性を扱うことによる効果を調べるため，計画立案の初期段階における初速を比較する．開始直後 10 分間について，AmReT Calendar において (特徴 2)(特徴 3) を有効にした場合と無効にした場合の差を詳細に比較評価する．

図 6.3 と 図 6.4 は，作業発生の規則性を扱う場合と扱わない場合で，開始後 10 分間における計画立案の比較である．横軸は経過時刻，縦軸は各時点でのタスクの登録件数である．各時点での登録件数は，作業発生の規則性を扱う場合が扱わない場合と比べて同じかそれ以上である．

また，図 6.3 と 図 6.4 から以下の 3 つの事が分かる．

- (1) ユーザ A に関して，作業発生の規則性を扱わない場合に 1 分以上入力が止まった時間帯が開始後 4 分から 5 分の間にある．一方，作業発生 of 規則性を扱う場合にはこのよ

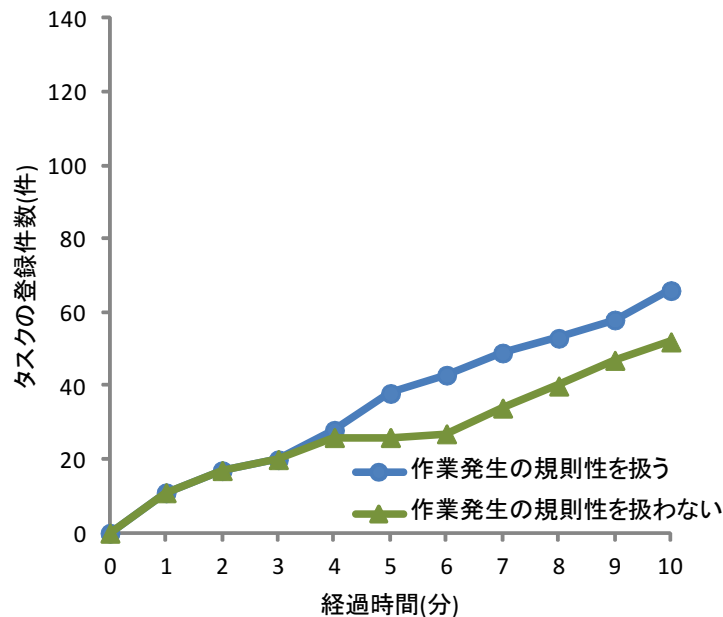


図 6.3 ユーザ A を対象とした作業発生規則性を扱う場合と扱わない場合の比較

うな時間帯が発生していない。つまり，ユーザ A は，長時間の考察を必要とせずにタスクを登録できている。

- (2) ユーザ B に関して，開始直後の 2 分間において，1 分あたりの登録件数は作業発生規則性を扱う場合に平均 38.5 件，扱わない場合に平均 4.0 件であり，作業発生規則性を扱う場合は扱わない場合と比較して登録速度は約 10 倍である。
- (3) ユーザ A に関して，開始直後の作業発生規則性を扱う場合と扱わない場合の登録速度の差は発生していない。

(1) の原因として，(特徴 2) の予報機能により，再び発生するかどうかの手間が発生しなかったことが考えられる。(2) と (3) の違いの原因として，(特徴 3) の一括登録機能の有効性に違いが出たことが考えられる。これは，(特徴 3) の一括登録機能で登録したタスク数がユーザ B の場合に 140 件であったのに対して，ユーザ A の場合に 26 件であった (表 6.1 参照) ことに起因する。

以上のことから，作業発生規則性を扱うことは計画立案において有用だと考えられる。しかし，ユーザ A のように過去の作業履歴が作業発生規則性に当てはまらない場合，有用性は低くなる。つまり，利用する過去 2 年間の履歴が既存のカレンダーシステム由来であるの

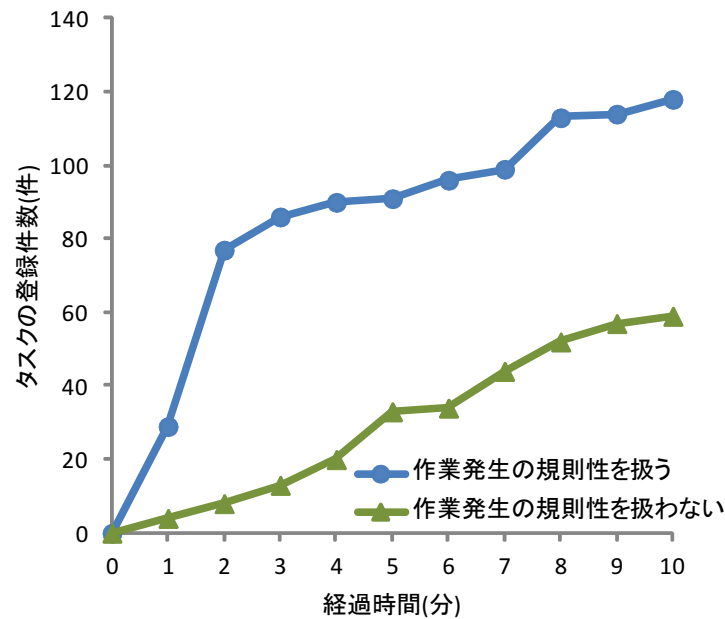


図 6.4 ユーザ B を対象とした作業発生の規則性を扱う場合と扱わない場合の比較

で、ユーザによってはメリットが少ない。カレンダーシステム利用者の多くは、中心となるイベントの日時のみを登録するのが普通であり、付随して起こるタスクの詳細な日時までは記録していない。しかし、今後、過去の仕事を振り返るツールとして本システムが位置付けられるようになれば、多くの付随するタスクを登録すると期待される。たとえば、書類作成や議事録の送付といった他の予定と関連して発生する作業がカレンダーに登録されるようになれば、(特徴 3) の一括登録機能がさらに有効に使えるようになると思われる。

第 7 章

関連研究

本カレンダーシステム以外にも、作業間の関係をモデル化して、今後の作業予測や仕事の引継ぎに利用するシステムの研究は存在する。たとえば、スケジュール情報の共有・再利用に着目した協調作業支援システム [1] では、業務知識としてスケジュールに着目し、スケジュールをリポジトリで管理することにより、業務知識を共有し活用する方式を提案している。AcTrec[2] は、個人の行動を記録し、統計処理によりシステム側から模範的な予定を提案する。知識蓄積・継承のためのスケジュールデータ構成法 [3] では、組織活動の中での基本となる 5 つのイベントを提案し、このモデルの適応事例を述べている。しかし、これらの研究が扱うモデルは、作業間にある依存関係をグラフとして表現したり、独自のダイアグラムを用いて行動間の連続関係とつながりの強さを表現するなど厳密である。このため、2.4.1 項で示したシステムや人が扱うには複雑すぎるという問題と既存のカレンダーシステムとの相互運用性が低いという問題が存在する。

また、作業を記録するツールを利用し、カレンダーシステム上に多くのタスクが記録されるようになれば、本システムはさらに有効になる。作業を記録するツールはいくつか存在する。たとえば、timeEdition[8] は、作業開始時に作業を選択してタイマーをセットすることで作業履歴をカレンダーシステムに保存する機能を提供する。デスクトップブックマーク [9] は、計算機上の作業履歴を仕事状態という単位で管理し、保存と復元をする機能を提供する。

さらに、タスクを抽出する対象としてメールに着目する。電子メールコミュニケーションにおけるスケジュール情報抽出 [10] や MHC[11] では、パターンマッチングにより、イベントの開催日時や返信依頼といったスケジュール情報を抽出する方法を提案している。そして、Web サービスとのマッシュアップを支援するメーリングリスト機構の提案 [12] では、グループウェアとメーリングリストを連携させる仕組みを提案している。この機構を利用することで、ユーザのロールに基づいてメールからタスクを収集することが可能になる。

第 8 章

おわりに

作業発生の規則性を扱うカレンダーシステムを提案した。まず，作業発生の規則性の特性について説明し，規則的を扱う際の問題点を述べ，問題点への対処法を示した。提案システムは，作業間の関連性と曖昧な周期性を表現するモデルを扱う。次に，このモデルを扱うカレンダーシステムと既存のカレンダーシステムを連携する方法について示した。さらに，作業発生の規則性を扱うことによる具体的なユーザ支援の方法について説明し，実装例を示した。そして，本システムの有用性を実測評価によって示した。

残された課題として，作業を記録するツールであるデスクトップブックマークや TODO 管理機能との連携がある。

謝辞

本研究を進めるにあたり，懇切丁寧なご指導をしていただきました乃村能成准教授に心より感謝の意を表します．また，研究活動において，数々のご指導やご助言を与えていただいた谷口秀夫教授，山内利宏准教授，後藤佑介助教に心から感謝申し上げます．

また，日頃の研究活動において，お世話になりました研究室の皆様に感謝いたします．

最後に，本研究を行うにあたり，経済的，精神的な支えとなった家族に感謝いたします．

参考文献

- [1] 安部田 章, 松並 勝, 碓崎 賢一, “スケジュール情報の共有・再利用に着目した協調作業支援システム,” 情報処理学会研究報告. [グループウェア], Vol. 95, No. 67, pp. 7-12, 1995.
- [2] 山根 隼人, 長尾 確, “ AcTrec : 行動履歴を用いた個人行動支援,” 情報処理学会第 66 回全国大会講演論文集, Vol. 66, No. 3, pp. 115-116, 2004.
- [3] 斉藤 典明, 金井 敦, 赤埴 淳一, “ 知識蓄積・継承のためのスケジュールデータ構成法,” 情報処理学会研究報告. [グループウェアとネットワークサービス (GN)], Vol. 2012-GN-82, No. 19, pp. 1-7, 2012.
- [4] C.Daboo, B.Desruisseaux, L. Dusseault, “Calendaring Extensions to WebDAV (Cal-DAV),” RFC 4791, 2007.
- [5] B.Desruisseaux, “Internet Calendaring and Scheduling Core Object Specification (iCalendar),” RFC 5545, 2009.
- [6] Google Inc., “Google Calendar,” <http://www.google.com/calendar/>.
- [7] Yahoo! Inc., “Yahoo! Calendar,” <http://calendar.yahoo.co.jp/>.
- [8] Living-e AG, “ timeEdition,” <http://www.timeedition.com/>.
- [9] 小笠原 良, 乃村 能成, 谷口 秀夫, “ デスクトップブックマーク : 計算機上の仕事状態の保存と復元機能の評価,” マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム論文集, Vol.2008, pp. 1418-1423, 2008.
- [10] 長谷川 隆明, 高木 伸一郎, “ 電子メールコミュニケーションにおけるスケジュール情報抽出,” 情報処理学会研究報告. 自然言語処理研究会報, Vol.98, pp. 73-80, 1998.
- [11] 乃村 能成, 花田 泰紀, 牛島 和夫, “ MHC-Message Harmonized Calendaring System の設計と実装,” 情報処理学会論文誌, Vol.42, pp. 2518-2525, 2001.

- [12] 藤原 啓輔, 乃村 能成, 谷口 秀夫, “ Web サービスとのマッシュアップを支援する メーリングリスト機構の提案 ,” 情報処理学会研究報告 第 141 回マルチメディア通信と分散処理研究会 (DPS), 電子媒体 ,vol.2009-DPS-141, 2009.

発表論文

- [1] 三原 俊介, 乃村 能成, 谷口 秀夫, “作業発生の規則性を扱うカレンダーシステムの提案,” マルチメディア通信と分散処理ワークショップ論文集, Vol.2010, No.11, pp. 215-220, 2010.
- [2] 三原 俊介, 乃村 能成, 谷口 秀夫, “作業発生の規則性を扱うカレンダーシステムの実現,” 研究報告マルチメディア通信と分散処理 (DPS), Vol.2011-DPS-149, No.10, pp. 1-6, 2011.
- [3] Yoshinari Nomura, Syusuke Mihara, Hideo Taniguchi, “A Practical Calendaring System Conforming with Ambiguous Pattern of Recurring Tasks,” Proceedings of the 14th International Conference on Network-Based Information Systems (NBIS2011), pp.553-558, 2011.
- [4] 三原 俊介, 乃村 能成, 谷口 秀夫, 南 裕也, “作業発生の規則性を扱うカレンダーシステムの評価,” 研究報告マルチメディア通信と分散処理 (DPS), Vol.2011-DPS-150, (掲載予定).
- [5] Yoshinari Nomura, Syusuke Mihara, Hideo Taniguchi, “Implementation of a Practical Calendaring System Conforming with Ambiguous Pattern of Recurring Tasks,” Proceedings of The 4th International Workshop on Information Technology for Innovative Services ITIS-2012-03, (to appear).