

## 作業発生の規則性を扱うカレンダーシステムの実現

三原 俊介<sup>†1</sup> 乃村 能成<sup>†1</sup> 谷口 秀夫<sup>†1</sup>

我々は、作業発生 of 規則性というモデルを提案した。このモデルを用いることで、作業間の関連性と作業のゆるい周期性を表現することができる。これにより、将来の作業予測や仕事引継ぎを支援できると考えられる。しかし、このモデルを用いた具体的なユーザ支援の方法は提案されていない。また、ユーザに負担の少ない方法で、関連性と周期性を取得する方法が必要となる。そこで、本稿では、「作業発生 of 規則性を扱うカレンダーシステム」の実現方法について提案する。まず、我々が提案した作業発生 of 規則性というモデルについて説明する。次に、このモデルの編集方法、読取方法、及び継承方法について提案する。そして、これらの機能としての実装例を示す。

### Implementation of Calendaring System Conforming with Ambiguous Recurring Tasks

SYUNSUKE MIHARA,<sup>†1</sup> YOSHINARI NOMURA<sup>†1</sup>  
and HIDEO TANIGUCHI<sup>†1</sup>

We proposed a model that is capable of handling the patterns of ambiguous recurring tasks. If we can grasp the pattern, it would be useful for prediction of the tasks which might happen in the future, and also be the hint on handing them over to co-workers. To utilize our model, we have developed a concrete method for manipulation or visualization of the modeled data. This method is useful to build a nifty user interface to the calendaring systems. In this paper, we propose an implementation of calendaring system conforming with ambiguous recurring tasks. First of all, we recall our model. Next, we propose some basic operations to the modeled data; EDIT, READ, and INHERIT. Finally, we describe the implementation of the method as an interface to our calendaring system.

<sup>†1</sup> 岡山大学大学院自然科学研究科

Graduate School of Natural Science and Technology, Okayama University

### 1. はじめに

オフィス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。

スケジュール情報を解析して将来の作業予測や仕事引継ぎに利用するシステムの研究は、これまでもいくつか存在する<sup>1),2)</sup>。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したりする等、普段我々が利用するカレンダーシステムや TODO 管理ツールが扱うモデルと比べて複雑である。よって、それらの研究成果は、作業予測や仕事引継ぎの支援として一般に普及しているとは言い難い。

このような現状を踏まえて、我々は、「タスク」、「ミッション」、及び「ジョブ」といった作業発生 of 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提案した<sup>3)</sup>。これらのモデルは、作業間の関連性と作業のゆるい周期性を表現する。しかし、これらのモデルを利用した具体的なユーザ支援の方法は提案されていない。また、作業予測や仕事引継ぎの支援として一般に普及するためには、ユーザに負担の少ない形で関連性や周期性を取得する方法と、ユーザにわかりやすい形で関連性や周期性を提示する方法が必要となる。

そこで、本稿では、上記のモデルを扱うカレンダーシステムの実現方法について述べる。まず、発生に関する作業の分類を述べる。次に、作業発生 of 規則性を扱うためのモデルについて述べ、このモデルを既存カレンダーシステムで利用されているデータ表現形式でどう表現するかについて説明する。さらに、このモデルに基づいたデータ構造をどのように操作するかについて述べる。そして、カレンダーシステムとしての実装例を示す。

### 2. 作業発生 of 規則性

#### 2.1 規則的に発生する作業の分類

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。

表 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 %)

たとえば、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、「会議への参加」や「掃除」といった作業は、ある程度決まった周期で発生する。

表 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ー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 1 の例では、カレンダー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、カレンダー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、既存のカレンダー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。

この事例より、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。

**2.2 既存のカレンダーシステムの扱うモデル**

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。

以降では、カレンダーシステム上で「作業発生の規則性」を扱う方法について説明する。

**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) 既存のカレンダーシステムのモデルと親和性が高いこと  
既存のカレンダーシステムは、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ー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**

関連性の表現方法として、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依存関係や順序関係を定義する方法がある。しかし、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  
扱う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。
- (2) 既存のカレンダー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、既存カレンダーが扱うモデルは、タスクの集合という非常に単純なものであるため、複雑な関係表現を持ち込むと、これらとの連携が考慮しづらい。よって、既存のカレンダーシステムと親和性が高いシンプルな表現が必要である。

**2.4.2 作業の周期性の表現**

既存のカレンダーシステムでは、1 つのタスクに固定的な周期を与えることで繰り返しを表現する。しかし、この表現方法では、現実で発生する以下の 2 つを表現できない。

- (1) 曖昧な周期  
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、個々の繰り返しに応じて変化する作業間の差異を扱えない。

よって、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。

### 3. 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。
- (2) 周期性は同様のものを集合にして表現する  
繰り返される作業は、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、繰り返すごとに個別のタスクとして扱う。そして、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各モデルについて説明する。

#### 3.2 規則性の表現

##### 3.2.1 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。

##### 3.2.2 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 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」、及び「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」、及び「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

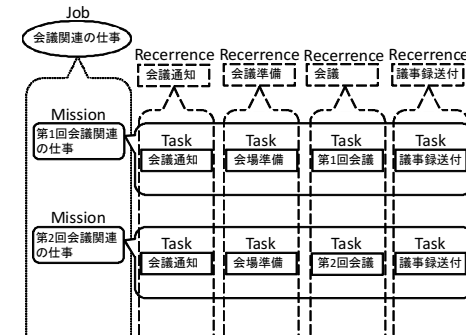


図1 本提案のモデルを適用した作業発生の規則性の例

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。

### 4. モデルに基づくデータ構造

3章のモデルを表現するデータ構造について説明する。データ構造は、既存のカレンダーシステムとの親和性を考慮し、iCalendar フォーマット<sup>4)</sup>で表現する。iCalendar フォーマットは、現在主流のカレンダーシステムがデータ交換に利用しているフォーマットである。

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、ミッションの表現方法について説明する。ミッションは VEVENT コンポーネントとして定義する。ミッションは集合であるので、その要素を表現するために VEVENT コンポーネントの RELATED-TO プロパティを利用する。このとき、ミッションの VEVENT コンポーネントと要素の VEVENT コンポーネントを親子関係にする。

そして、リカーレンスとジョブの表現方法について説明する。リカーレンスは、要素のタスクの VEVENT コンポーネント同士を兄弟関係にして表現する。また、ジョブは、要素

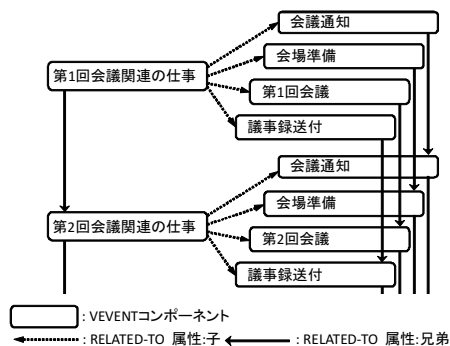


図2 iCalendar フォーマットで表現した作業発生の規則性の例

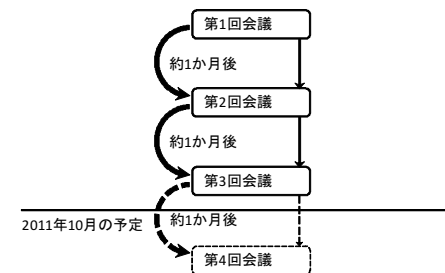


図3 リカーレンスに基づく周期性の継承

のミッションの VEVENT コンポーネント同士を兄弟関係にして表現する。

図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。

## 5. データ構造の操作

4章で述べたデータ構造を扱うために、データ構造の編集 (*Edit*)、読取 (*Read*)、及び継承 (*Inherit*) の操作を定義する。以降では、各操作について述べる。

### 5.1 データ構造の編集

タスクの編集方法は既存のカレンダーシステムにも存在する。よって、ミッション、リカーレンス、及びジョブの編集方法を述べる。

まず、ミッションの編集方法を述べる。ミッションは、要素となるタスクを選択して、名前を付けることで設定する。また、ミッションとタスクを選択することで、後からミッションの要素を追加できるようにする。

次に、リカーレンスとジョブの編集方法を述べる。これらは、過去に発生したミッションかタスクを選択して、次の発生を登録することで設定する。たとえば、約1年周期で発生するタスクが、去年のカレンダーに登録されているとする。ユーザは、去年のカレンダーから今年のカレンダーにタスクを複製することで、リカーレンスを設定する。

## 5.2 データ構造の読取

任意のタスクを選択時に、同一のミッションに含まれる他のタスクを一覧表示する。また、タスクかミッションを選択時に、リカーレンスかジョブに基づいて過去の一覧を表示する。

以上のデータ構造の読取方法を用意することで、過去の作業内容について確認する際に有用である。たとえば、例年発生する「会議」について確認したいとする。この場合に、「会議」を選択時に、ミッションに基づいて「会議」の前後に必要なタスクを確認できる。また、リカーレンスに基づいて過去に発生した「会議」を簡単に参照できる。

## 5.3 データ構造の継承

### 5.3.1 リカーレンスに基づく周期性の継承

1つのリカーレンスの要素となっているタスクは、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、次に発生するタスクにも継承されると考えられる。よって、この周期性を用いて、次に発生するタスクを予測する。

これにより、ユーザがある月の予定を立てる際に、その月に発生すると考えられるタスクをシステムから提案できる。そして、提案されたタスクをカレンダーに登録するとき、このタスクは自動的にリカーレンスの要素として設定される。

図3は、リカーレンスに基づくタスク提案の図である。図3では、2011年10月の予定を立てるとき、システムは同一リカーレンスを持つタスクを解析して、この月に発生すると考えられる「第4回会議」のタスクを提案する。

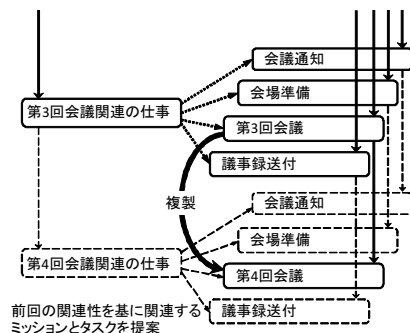


図 4 ミッションに基づく関連性の継承

### 5.3.2 ミッションに基づく関連性の継承

リカーレンスの要素となっているタスクは、発生時に関連して発生する他の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、次に発生するタスクにも継承されると考えられる。よって、この関連性を用いて関連して発生するタスクを予測する。

これにより、ユーザが繰り返し発生するタスクをカレンダーに登録したとき、関連して発生すると考えられるタスクをシステムから提案できる。そして、それらのタスクを要素とするミッションを自動的に作成する。また、提案された個々のタスクをカレンダーに登録する際、個々のタスクは自動的にそれぞれのリカーレンスの要素として設定される。さらに、新たに作成されたミッションも、自動的にジョブの要素として設定される。

図 4 は、ミッションに基づいてタスクを提案する様子である。「第 4 回会議」をカレンダーシステムに登録したとき、関連して発生する「会議通知」、「会場準備」、及び「議事録送付」というタスクを提案する。また、それらのタスクを要素とする「第 4 回会議関連の仕事」というミッションも提案する。

## 6. プロトタイプの実装

5 章で述べた機能を持つカレンダーシステムのプロトタイプを実装した。実装には、Ruby on Rails を利用した。本プロトタイプでは、基本的なカレンダーシステムに、いくつかの作業発生 の規則性を扱う機能を実装した。

以下の (a) から (f) の 6 機能がこれらにあたり、それぞれを (1) から (3) の特徴に分類した。

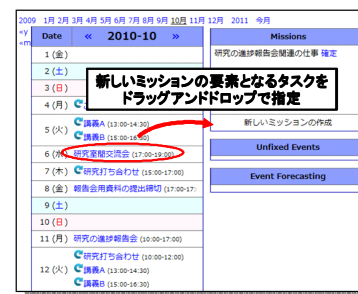


図 5 ミッションのドラッグアンドドロップによる設定



図 6 リカーレンスのドラッグアンドドロップによる設定

- (1) データ構造を編集する機能
  - (a) ミッションのドラッグアンドドロップによる設定
  - (b) リカーレンスのドラッグアンドドロップによる設定
- (2) データ構造を読み取る機能
  - (c) ミッションに基づくタスクの表示
  - (d) リカーレンスに基づくタスクの表示
- (3) データ構造を継承する機能
  - (e) リカーレンスに基づくタスクの予報機能
  - (f) ミッションに基づくタスクの仮登録機能

以降では、各機能の詳細について説明する。

### 6.1 データ構造を編集する機能

図 5 と図 6 を用いて、(a) ミッションのドラッグアンドドロップによる設定と (b) リカーレンスのドラッグアンドドロップによる設定の実装例を示す。

図 5 のように、カレンダーに登録されているタスクをドラッグアンドドロップすることで、ミッションを作成できる。そして、カレンダーに登録されているタスクをリスト表示されているミッションにドラッグアンドドロップすることで、ミッションの要素として登録される。

また、図 6 のように、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。このとき、複製元のタスクと複製先のタスクはリカーレンスとなる。

これらの機能により、ユーザは作業発生 の規則性を直感的に操作することができる。



図 7 ミッションに基づくタスクの表示



図 8 リカーレンスに基づくタスクの表示

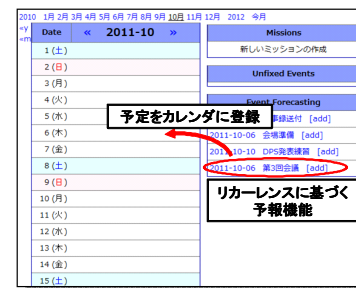


図 9 リカーレンスに基づくタスクの予約機能



図 10 ミッションに基づくタスクの仮登録機能

## 6.2 データ構造を読み取る機能

図 7 と図 8 を用いて, (c) ミッションに基づくタスクの表示と (d) リカーレンスに基づくタスクの表示の実装例を示す.

図 7 のように, 表示月に関係するミッションは, カレンダーの右側にリスト表示される. そして, タスクにマウスオーバーしたとき, 関連するタスクは別の色で表示される.

また, 図 8 のように, 任意のタスクを選択時, リカーレンスに基づいて過去の同様のタスクをリスト表示される. そして, このリストから過去のタスクを参照することも可能である.

## 6.3 データ構造を継承する機能

図 9 と図 10 を用いて, (e) リカーレンスに基づくタスクの予約機能と (f) ミッションに基づくタスクの仮登録機能の実装例を示す.

予約機能とは, リカーレンスに基づいて解析した結果, その月に発生すると予想されるタスクを提示する機能である. 図 9 のように, 予約機能はカレンダーの右側にリストとして表示される. このリストからカレンダーにドラッグアンドドロップでタスクを登録できる.

また, タスクの仮登録機能とは, リカーレンスを持ったタスクがカレンダーに登録されたとき, ミッションに基づいて関連したタスクをカレンダーに仮登録する機能である. 図 10 のように, 仮登録のタスクは通常のタスクと別の色で表示される. さらに, 必要に応じてドラッグアンドドロップで適切な日時にタスクを移動できる.

## 7. まとめと残された課題

本提案では, 将来の作業予測や仕事引継ぎを支援するため, まず, 作業発生の規則性を扱

うモデルを説明した. 次に, このモデルを既存カレンダーシステムに適用しやすい形で表現する方式について説明した. さらに, このモデルに基づくユーザ支援の方法について述べた. そして, 作業発生の規則性を扱うカレンダーシステムの実装例について述べた.

残された課題として, 1 つ目にリカーレンスに基づく作業の提案手法の改善がある. 実用のためには, 長期休暇, 例外的な発生, 及び曜日といった情報を考慮した提案が必要となる. 2 つ目に, 作業情報を収集する手段の考案がある. 「行事の参加」や「打合せ」等の一部の作業以外にも, 「通知」や「書類作成」等の日常的な作業を収集する方法が必要となる.

謝辞 本研究の一部は, それぞれ科学研究費補助金・若手研究 (B)(課題番号: 21700139) による研究費, 及び日本電信電話株式会社 NTT サイバースソリューション研究所と国立情報学研究所の提供する研究設備, 回線を活用した. ここに記して謝意を示す.

## 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) 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.
- 4) B.Desruisseaux, “Internet Calendaring and Scheduling Core Object Specification (iCalendar),” RFC 5545, 2009.