

作業発生の規則性を扱うカレンダーシステムの提案

三原 俊介^{†1} 乃村 能成^{†1} 谷口 秀夫^{†1}

オフィス環境での作業の発生には、ある程度決まった規則性が存在する。作業発生の規則性を確認できれば、将来の作業予測や仕事引継ぎ時の情報伝達に有益である。オフィス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、作業間の関連性を扱えないことと、作業のゆるい周期性を表現する方法がないことがある。そこで、本稿では、これらの問題を解決する「作業発生の規則性を扱うカレンダーシステム」を提案する。我々は、作業の関連性と作業のゆるい周期性をタスク、ミッション、およびジョブという概念を利用してモデル化する。また、これらの概念を扱うカレンダーシステムを既存のカレンダーシステムと連携可能な形で実現する。

A Calendaring System Conforming with Ambiguous Cycle of Recurring Tasks

SYUNSUKE MIHARA,^{†1} YOSHINARI NOMURA^{†1}
and HIDEO TANIGUCHI^{†1}

There are many recurring tasks in our office work. If we can grasp the pattern of the recurrences, 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. Calendaring system is one of the popular tools for enumeration of our tasks. However, current major implementations are not capable of handling the fluctuation of recurrences or relationships among the tasks. Therefore, they do not give us any analytic perspective view of the tasks. In this paper, we propose a new calendaring system, which is able to handle the fluctuations and the relationships of recurrence tasks. First of all, we introduced the three terms namely *Task*, *Mission*, and *Job* to describe relationships among the tasks. We also describe the implementation of our concept and how it works retaining interoperability with the existing systems.

1. はじめに

オフィスにおける様々な作業には、発生に関して周期性を持つものが少なくない。また、1つの作業に伴って、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とが多い。例えば、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、事後の作業を伴う。このような各作業間の関係や周期性を容易に確認できれば、将来の作業発生を予測したり、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、数ヶ月先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)}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。

そこで、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手法について述べる。まず、「タスク」、「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概念について述べる。次に、これらの概念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すかについて述べる。さらに、これらの概念を扱えるカレンダーシステムを既存のカレンダーシステムと連携可能な形で実現する方法を述べる。

^{†1} 岡山大学大学院自然科学研究科

Graduate School of Natural Science and Technology, Okayama University

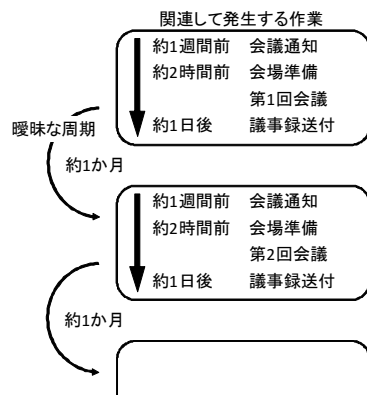


図 1 規則的に発生する作業の例

2. 作業発生の規則性

2.1 規則的に発生する作業

我々が行っている作業の多くは他の作業と関連し発生している。例として、部署内での定例会議の場合を考える。図 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。部署内での定例会議では、「会議の通知」、「会場の準備」、および「議事録の送付」といった作業が関連して発生する。それぞれの発生するタイミングは、ほぼ決まっている。図 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、我々の作業の繰り返しには、ある程度規則的な周期がある。図 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。

以上で述べたように、作業の関連性と周期性により、作業の発生には規則性が存在していると考えられる。本提案では、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。

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述べる。

2.3 作業発生の規則性を表現する上での要求

作業発生の規則性を表現する上で以下の 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。これらの曖昧さを許容できる表現であることが、現実の予定を扱う上では重要である。

2.4 作業発生の規則性を表現する上での問題点

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していく方法である。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
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。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、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ーシステムと親

和性が高いシンプルな表現を新たに用意することが必要である。

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、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上で問題となる以下の2つを表現できない。

(1) 曖昧な周期

既存のカレンダー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。

3. 作業発生の規則性を扱うための概念

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、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。

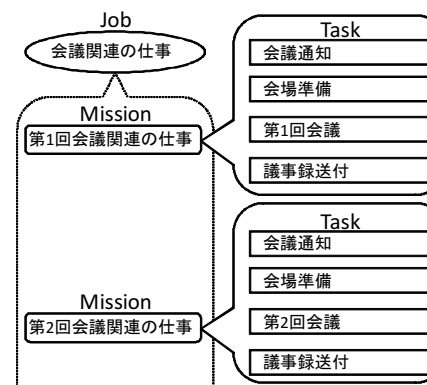


図2 本提案の概念を適用した作業発生の規則性の例

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

関連性を扱う概念として、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元とする集合である。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。ミッションに含まれるタスクやミッションは、ミッションの内容を細分化したものである。

3.2.2 ジョブによる周期性の表現

周期性を扱う概念として、ジョブ (Job) を定義する。ジョブはミッションを元とする集合である。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。ジョブは繰り返し発生する作業の種類を表現する。

3.3 概念の適用

図2は、前節の概念を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」、「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」、および「議事録送付」の発生は「第1回会議」の発生と関連していることを表現するために、これらのタスクを元とするミッション「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、「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

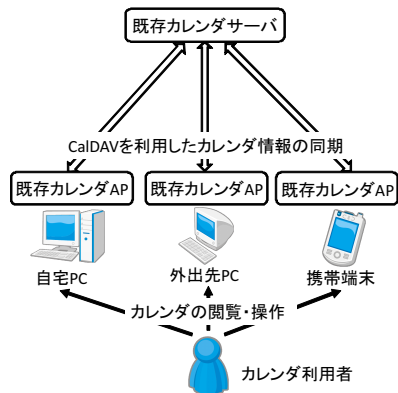


図 3 現在主流のカレンダーシステム

ションを元とするジョブ「会議関連の仕事」を定義する。

4. 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章の概念を扱うためにシステムをどのように実装すべきか述べる。

図3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3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CalDAV³⁾を使ってiCalendarフォーマット⁴⁾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⁵⁾やYahoo!カレンダー（米）⁶⁾がある。しかし、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章の概念を表現できれば、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 iCalendarフォーマットによる概念の表現

3章の概念をiCalendarフォーマットで表現する方式について述べる。

まず、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、ミッションとジョブの表現方法について述べる。ミッションは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。例えば、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、一部のカレンダーシステムは、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。

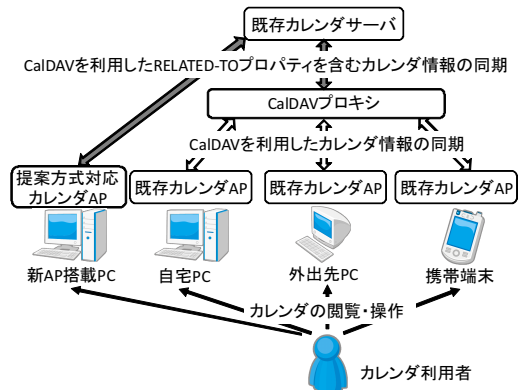
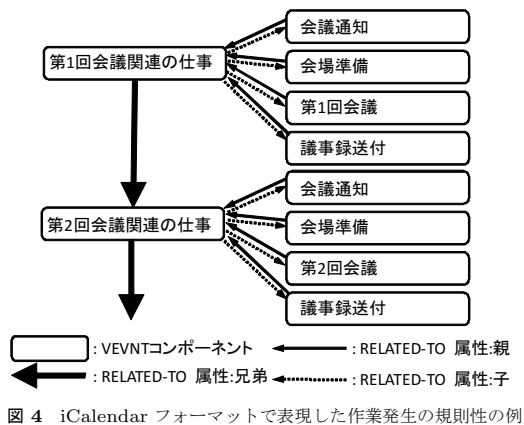
図4は、2.1節の例をiCalendarフォーマットで表現したものである。「第1回会議関連の仕事」のVEVENTは、「会議通知」と「会場準備」、「会議」、「議事録送付」のVEVENTとRELATED-TOプロパティで親子関係を持つ。また、「第1回会議関連の仕事」と「第2回会議関連の仕事」は、RELATED-TOプロパティ兄弟関係を持つ。

次章では、このようなiCalendarフォーマットで表現した作業発生の規則性を扱えるシステムについて述べる。

5. 提案方式の実装

5.1 全体構成

図5は、本提案方式によるカレンダーシステムの全体構成である。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。
- (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 プロパティを元に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) 提案方式に対応したカレンダー AP
RELATED-TO プロパティを解析し、作業発生の規則性をユーザが直感的にわかる形に表示する。また、RELATED-TO プロパティを直接記述できる機能を持ち、作業発生の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) と提案方式に対応したカレンダー AP (4) に関しては、次節以降で詳しく説明する。

5.2 CalDAV プロキシ

5.2.1 機能

RELATED-TO プロパティが扱えない既存カレンダー AP で作業発生の規則性の操作と閲覧ができるようにするため、CalDAV プロキシを提案する。CalDAV プロキシは、カレンダー AP からのカレンダー情報に含まれている RELATED-TO プロパティを本提案で定義している概念に従って解析し、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に絞ったカレンダー情報をカレンダー 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。

5.2.2 動作

図 6 と図 7 に、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を示す。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 6), (1) カ

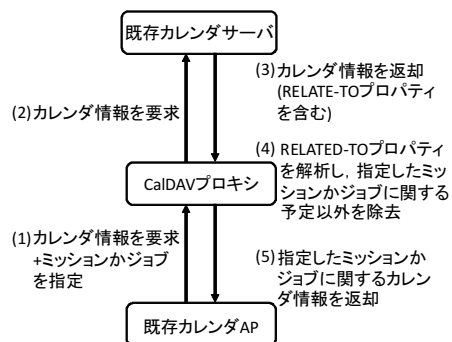


図 6 既存カレンダー AP から作業発生の規則性を閲覧

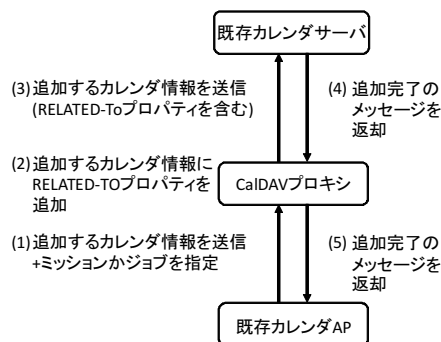


図 7 既存カレンダー AP から作業発生の規則性を操作

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 7), (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。

5.3 提案方式に対応したカレンダー AP

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提案方式に対応したカレンダー AP 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 は、オープンソースのカレンダー AP である Lightning⁷⁾ を拡張することで作成する。また、このカレンダー AP は、以下の 6 つの機能を持つ。

(1) RELATED-TO プロパティを直接編集する機能

- (2) 任意のミッションを選択時、自身の元を時系列に一覧表示する機能
- (3) 任意のミッションまたはタスクを選択時、自身が属するミッションの元を時系列に一覧表示する機能
- (4) 任意のジョブを選択時、自身の元を時系列に一覧表示する機能
- (5) 任意のミッションを選択時、自身が属するジョブの元を時系列に一覧表示する機能
- (6) (2) (3) (4) (5) の一覧から任意の項目を選択することで、項目のミッションまたはタスクを直接参照する機能

6. おわりに

本提案では、将来の作業予測や仕事の引継ぎを支援するため、まず、作業発生の規則性を扱う諸概念を定義した。次に、これらの概念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述べた。そして、提案方式の実装方法について述べた。今後は、提案システムの実装と評価を行う。

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 サービスインテグレーション基盤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) C.Daboo, B.Desruisseaux, L. Dusseault, “Calendar Extensions to WebDAV (CalDAV),” RFC 4791, 2007.
- 4) B.Desruisseaux, “Internet Calendaring and Scheduling Core Object Specification (iCalendar),” RFC 5545, 2009.
- 5) Google Inc., “Google Calendar,” <http://www.google.com/calendar/>
- 6) Yahoo! Inc., “Yahoo! Calendar,” <http://calendar.yahoo.co.jp/>
- 7) Mozilla Foundation, “Lightning,” <http://www.mozilla-japan.org/projects/calendar/lightning/>