

特 別 研 究 報 告 書

題 目

CalDAV プロキシの設計と実装

指導教員

報 告 者

村田 裕哉

岡山大学工学部 情報工学科

平成 25 年 2 月 7 日 提出

要約

オフィスや家庭において、スケジュール管理のツールとしてカレンダーシステムが広く使われている。カレンダーシステムは、個人でのスケジュール管理だけでなく、研究グループや家族といった複数人でのスケジュール管理も可能である。複数人でカレンダーシステムを用いてスケジュール管理することにより、予定を共有できたり、相互にカレンダーを閲覧できたりする。これにより、複数人に関わる日程の調整が容易になるといった利点がある。

しかし、複数人でのカレンダーシステムの利用には問題が3つある。1つ目の問題は、招待機能を用いる場合に起こる。招待機能は、メールを用いて予定を通知し、カレンダーへの登録を促す機能である。招待機能を用いて予定を共有する場合、複数人が同一のカレンダーシステムを利用する必要がある。この際、個人が利用するカレンダーシステムを自由に選択できない点が問題となる。2つ目の問題は、共有カレンダーを用いる場合に起こる。共有カレンダーを用いる方法では、共有カレンダーへのアクセス権を失った場合、過去の予定を喪失する。過去の予定を喪失するのが妥当な場合もあるが、組織や個人が過去の予定を喪失させるか否かを選択できない点が問題となる。3つ目の問題は、他者に自分のカレンダーの閲覧を許可する場合に起こる。他人にカレンダー情報の閲覧を許可する場合、見せる相手によって見せたい予定や隠蔽したい予定が異なる。しかし、既存のカレンダーシステムでは、目的に応じた粒度の細かい予定の公開設定ができない点が問題となる。

本論文では、これらの問題に対処する方法として「CalDAV プロキシ」を提案した。CalDAV プロキシは、CalDAV による通信を中継する。CalDAV は、カレンダー情報を通信するための通信プロトコルである。CalDAV プロキシは、複数人でのカレンダーシステムの利用における問題に対処するために、カレンダーサーバやカレンダー AP の違いを吸収する機能、個人のカレンダーでグループの予定を管理する機能、および相手に応じて予定をフィルタリングする機能を持つ。

また、これらの機能を実現するために CalDAV プロキシの設計について述べた。さらに、個人のカレンダーでグループの予定を管理する機能と、相手に応じて予定をフィルタリングする機能をプロトタイプとして実装した。プロトタイプの実装により、CalDAV の通信を中継することで通信するカレンダー情報を書き換えられる事を確認した。

目次

1 はじめに	1
2 カレンダシステムの現状	3
2.1 想定するカレンダシステムの構成	3
2.1.1 概要	3
2.1.2 iCalendar	4
2.1.3 CalDAV	5
2.2 複数人でのカレンダシステムの利用	6
2.2.1 分類	6
2.2.2 予定の共有	6
2.2.3 予定の公開	7
3 複数人のカレンダシステムの利用における問題点	9
3.1 概要	9
3.2 招待機能によるカレンダシステム間の予定の共有が不可能	10
3.3 共有カレンダによる過去の予定の喪失	11
3.4 公開設定による相手に応じた予定の見せ方の変更が不可能	12
4 対処	14
4.1 対処に必要な条件	14
4.2 CalDAV プロキシを用いた対処	15
4.2.1 CalDAV プロキシの概要	15
4.2.2 カレンダサーバやカレンダ AP の違いを吸収する機能	16
4.2.3 個人のカレンダでグループの予定を管理する機能	17
4.2.4 相手に応じて予定をフィルタリングする機能	18

5 設計	20
5.1 システム構成	20
5.2 動作概要	23
6 実装	25
7 おわりに	26
謝辞	27
参考文献	28

目 次

2.1	既存カレンダーシステムの利用形態	4
2.2	iCalendar の概要	5
2.3	招待機能の利用	7
3.1	Yahoo!カレンダーの招待機能	10
3.2	共有カレンダーへのアクセス権限の喪失	12
3.3	相手に応じた予定の見せ方の変更	13
4.1	CalDAV プロキシの概要	15
4.2	CalDAV プロキシを用いたグループの予定の管理	17
4.3	CalDAV プロキシを用いた予定のフィルタリング	18
5.1	CalDAV プロキシの構成	21
5.2	データベースの構成	22

表 目 次

6.1 プロトタイプの実装環境	25
---------------------------	----

第 1 章

はじめに

オフィスや家庭において、スケジュール管理のツールとしてカレンダーシステムが広く使われている。本研究で想定する既存カレンダーシステムでは、カレンダー情報をサーバで保持し、クライアントからサーバにアクセスすることでカレンダー情報の閲覧や操作を行う。カレンダーシステムは、個人でのスケジュール管理だけでなく、研究グループや家族といった複数人でのスケジュール管理も可能である。

複数人でカレンダーシステムを用いてスケジュール管理する場合の利点は2つある。1つ目の利点は、複数人のカレンダー間で予定を共有できることである。カレンダー間で予定を同期することで複数人に関わる予定を共有でき、複数人の間で日時や場所といった予定に関する情報の認識の食い違いを防げる。2つ目の利点は、他人のカレンダー情報を閲覧できることである。他人のカレンダー情報を閲覧し予定を確認することで、複数人に関わる日程の調整が容易になる。

しかし、複数人でのカレンダーシステムの利用には問題が3つある。1つ目の問題は、招待機能を用いる場合に起こる。招待機能は、メールを用いて予定を通知し、カレンダーへの登録を促す機能である。招待機能を用いて予定を共有する場合、現状では複数人が同一のカレンダーシステムを利用する必要がある。この際、個人が利用するカレンダーシステムを自由に選択できない点が問題となる。2つ目の問題は、共有カレンダーを用いる場合に起こる。共有カレンダーを用いる方法では、共有カレンダーへのアクセス権を失った場合、過去の予定を喪失する。過去の予定を喪失するのが妥当な場合もあるが、組織や個人が過去の予定を喪失させるか否かを選択できない点が問題となる。3つ目の問題は、他者に自分のカレンダーの閲覧を許可する場合に起こる。他人にカレンダー情報の閲覧を許可する場合、見せる相手によって見せたい予定や隠蔽したい予定が異なる。しかし、既存のカレンダーシステムでは、目的に応じた予定の公開設定ができない点が問題となる。

本論文では、これらの問題に対処する方法として「CalDAV プロキシ」を提案する。CalDAV プロキシは、CalDAV による通信を中継する。CalDAV は、カレンダー情報を通信するためのプロトコルである。まず、CalDAV プロキシを用いて予定を共有する方法について述べる。次に、CalDAV プロキシを用いて相手に応じて予定をフィルタリングする方法について述べる。さらに、これらの方法を実現するために CalDAV プロキシの設計について述べる。

第 2 章

カレンダーシステムの現状

2.1 想定するカレンダーシステムの構成

2.1.1 概要

本研究で想定する既存カレンダーシステムの利用形態を図 2.1 に示す。カレンダーシステムには以下の 4 つの特徴がある。

- (1) カレンダー情報の閲覧と操作は、各種端末のカレンダーアプリケーション (以下カレンダー AP) あるいは、Web ブラウザから行う。
- (2) カレンダー情報の保持は、カレンダーサーバで行う。カレンダー AP によっては、サーバから取得した情報をキャッシュ、同期する機能を持つ。
- (3) カレンダー情報のデータフォーマットとして、iCalendar[1] を利用する。
- (4) カレンダーサーバとカレンダー AP とのカレンダー情報のやり取りは、CalDAV[2] を利用する。
- (5) Web インターフェース部を介したカレンダーサーバとブラウザとのカレンダー情報のやり取りは、HTTP を利用する。

iCalendar とは、カレンダー情報や TODO 情報といったスケジュール情報を記述するためのフォーマットであり、標準規格として RFC5545(旧 2445) で定義されている。また、CalDAV とは、iCalendar フォーマットのカレンダー情報を共有管理するための通信プロトコルであり、RFC4791 で定義されている。

これらの特徴によって、状況に応じて様々な端末のカレンダー AP から、同一のカレンダー情報を操作できる。

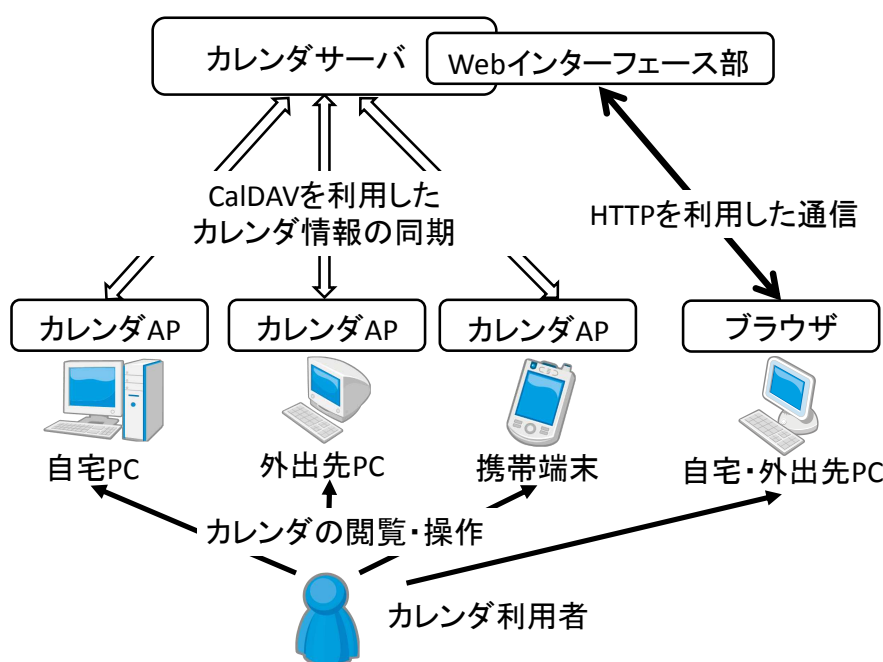


図 2.1 既存カレンダーシステムの利用形態

2.1.2 iCalendar

iCalendar とは、カレンダー情報や TODO 情報といったスケジュール情報を記述するためのフォーマットであり、可読テキストの形式で表現される。iCalendar の概要を図 2.2 に示し、以下で説明する。iCalendar では個々のカレンダー毎に 1 つの iCalendar オブジェクトが存在する。iCalendar オブジェクトは、1 つの VTIMEZONE コンポーネントと複数の VEVENT コンポーネントで構成される。VTIMEZONE コンポーネントには、時差情報や夏時間に関する情報が記述される。VEVENT コンポーネントには、カレンダー内の予定が記述される。1 つの VEVENT コンポーネントは 1 つの予定を表す。このため、iCalendar オブジェクトは、カレンダー内の予定の数だけ VEVENT コンポーネントを持つ。

VEVENT コンポーネントの中には、個々の予定に関する情報が記述される。VEVENT コンポーネントに記述される情報のうち、ユーザがカレンダー AP を利用して変更できる項目は主に次の 6 つである。

(1) DTSTART

予定の開始日時を指定する。

(2) DTEND

予定の終了日時を指定する。

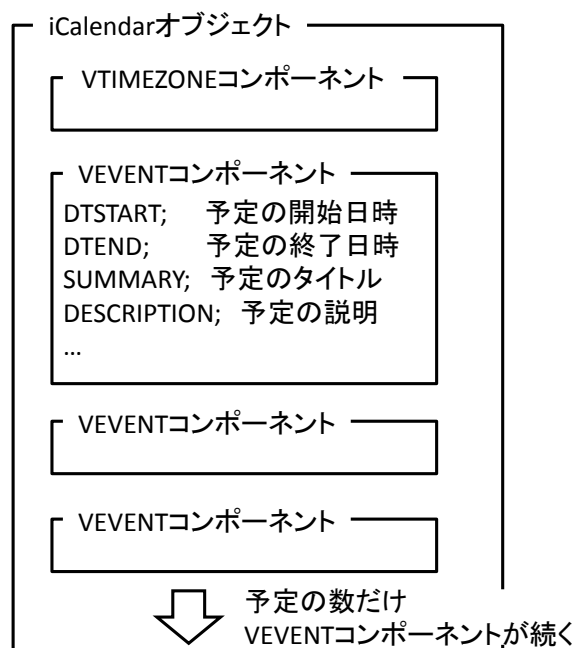


図 2.2 iCalendar の概要

(3) SUMMARY

予定のタイトルを記述する。

(4) LOCATION

予定に関係する場所を記述する。

(5) ATTENDEE

予定の参加者のメールアドレスを記述する。参加者は複数人追加できる。

(6) DESCRIPTION

上記以外の予定の説明を記述する。

また、VEVENT コンポーネントは、個々の予定を識別するためのユニークな ID として、UID を保持する。

2.1.3 CalDAV

CalDAV とは、iCalendar の形式でカレンダー情報をやり取りするために、HTTP をベースとした WebDAV[3] をさらに拡張した通信プロトコルである。CalDAV では、カレンダー情報

をカレンダーサーバ上で保持し、カレンダー情報における 1 つの VEVENT を 1 つの URI として扱う。この URI に対して、GET/PUT/DELETE などの操作することで予定の編集や取得を行う。

2.2 複数人でのカレンダーシステムの利用

2.2.1 分類

iCalendar や CalDAV は、複数人でカレンダーシステムを利用する方法を定義していない。このため、カレンダーシステムはカレンダーサーバやカレンダー AP に招待機能や共有カレンダーといった仕組みを独自に追加することで、複数人での利用を支援している。これらの仕組みを用いた複数人でのカレンダーシステムの利用は、以下の 2 つに分類できる。

(1) 予定の共有

複数人に関わる予定について、全員が同じ情報を見る。

(2) 予定の公開

他者に関係するスケジュールを考えると、他者の公開されたスケジュールを確認する。

次項から、カレンダーシステムにおける予定の共有と日程の調整について、概略を説明する。

2.2.2 予定の共有

複数人に関わる予定について、全員が同じ情報を見る必要がある。例えば、複数人に関わる予定に日時の変更があった場合、カレンダー内の予定を変更する。しかし、1 人がカレンダー内の予定を変更し忘れるなどした場合、問題となる。カレンダーシステムには、このような問題を防ぐための方法として以下の 2 つの方法がある。

(1) 招待機能を使う

招待機能とは、招待した相手に予定に関する情報をメールで知らせ、カレンダーへの予定の登録を促す機能である。招待機能を用いる例を図 2.3 に示す。図 2.3 は、Yahoo! カレンダーにおいて、ユーザ A が招待機能を用いてユーザ B を招待する例である。ユーザ A は自身のカレンダーに予定を登録し、予定の参加者にユーザ B を追加する。予定を登録する際に招待機能を用いて、ユーザ B に招待メールを送信する。ユーザ B は招待メールの「Yahoo! カレンダーに予定を登録する」リンクを開き、1 クリックで予定を登

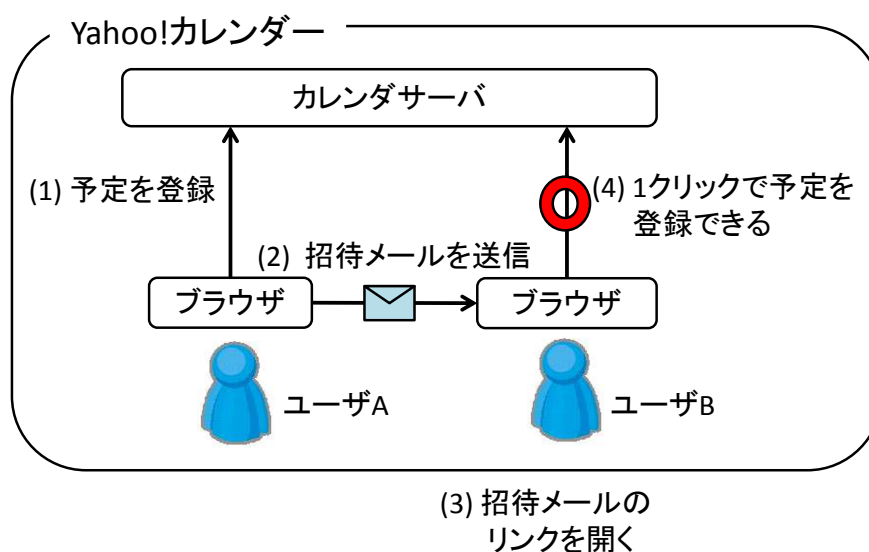


図 2.3 招待機能の利用

録する。

この機能を用いて予定を登録すると、招待したユーザと招待されたユーザの予定の UID が同一になる。予定の UID が同一なことにより、双方のユーザの予定が同一の予定であることが分かる。このため、一方のユーザが予定を編集し、招待機能を用いて予定の変更をもう一方のユーザに通知すると、予定の変更として処理される。このように、招待機能を用いて予定を登録することで、予定が共有できる。

(2) 共有用のカレンダーを使う

共有用のカレンダー (以下、共有カレンダー) とは、複数人が閲覧および編集することを許可されたカレンダーである。予定を共有したいメンバ全員が、1つの共有カレンダーを閲覧および編集することで、複数人で予定を共有できる。共有カレンダーを使うと、招待機能を使う方法のように、予定を作成する度に参加者を追加する必要がない。

2.2.3 予定の公開

他者に関するスケジュールを考えると、他者のスケジュールを確認できれば有用である。しかし、他者としては、自分の全てのスケジュールを閲覧させたくない場合がある。このように自分のスケジュールを隠蔽して他者に閲覧を許可するため、カレンダーサーバやカレンダー AP では、個々の予定に対して公開設定として公開／非公開を設定できる。また、Google

カレンダーや Yahoo!カレンダーでは公開／非公開だけでなく、非公開、予定の時間枠情報のみを公開、および予定の詳細を公開を選択できる。予定の時間枠情報のみを公開を選択すると、他者からはある時間帯における予定の存在のみを確認できる。また、予定の詳細を公開を選択すると、他者から予定の詳細を確認できる。

第 3 章

複数人のカレンダーシステムの利用における問題点

3.1 概要

2.2.2 項で、予定を共有する方法として招待機能を使う方法と、共有カレンダーを使う方法の 2 つを述べた。しかし、これら 2 つの方法にはどちらも問題がある。2 つの方法の持つ問題をそれぞれ、問題 1 と問題 2 に示す。

また、2.2.3 項で、他者に自分のスケジュールの閲覧を許可する場合ためにプライベートな予定を隠蔽する方法を述べた。しかし、この方法にも問題がある。この方法の持つ問題を問題 3 に示す。

問題 1 招待機能によるカレンダーシステム間の予定の共有が不可能

単一のカレンダーシステム内では、招待機能を用いて予定を共有できる。しかし、異なるカレンダーシステム間では、招待機能を用いて予定を共有できない。

問題 2 共有カレンダーによる過去の予定の喪失

カレンダーは、未来の予定表としての意味だけでなく、過去の行動履歴としての意味も持つ。共有カレンダーへのアクセス権を失うと、過去の行動履歴を喪失してしまう。

問題 3 公開設定による相手に応じた予定の見せ方の変更が不可能

予定を見せる相手によって、見せたい予定や隠蔽したい予定が異なる。しかし、既存カレンダーシステムにおける公開設定では、相手に応じて予定の見せ方を変更できない。

次節から、これら 3 つの問題点の詳細を述べる。

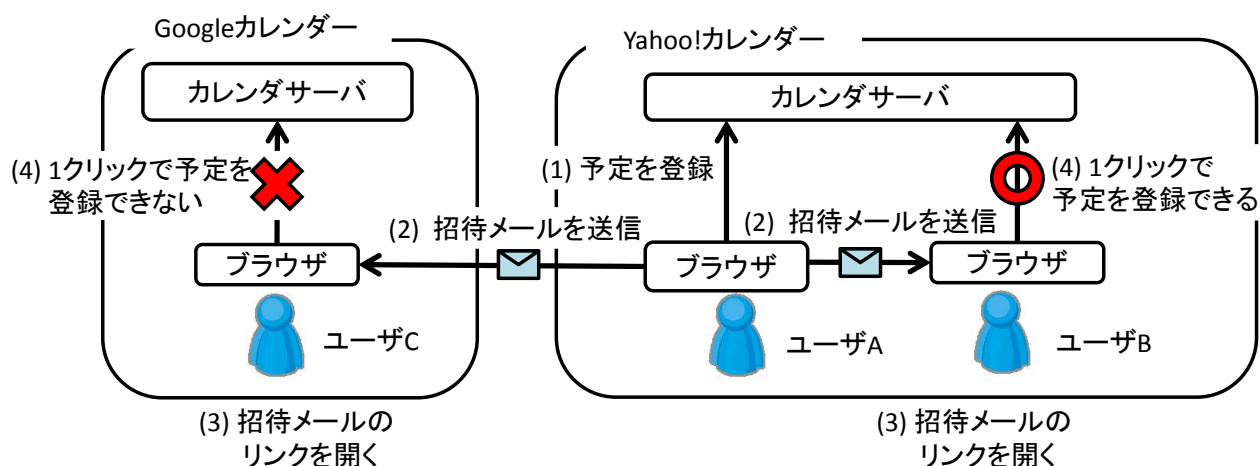


図 3.1 Yahoo!カレンダーの招待機能

3.2 招待機能によるカレンダーシステム間の予定の共有が不可能

単一のカレンダーシステム内では、2.2.2 項の (1) で述べたように、招待機能を用いて予定を共有できる。しかし異なるカレンダーシステム間では、招待機能を用いて予定を共有できない。これは、招待機能がカレンダーシステムに大きく依存した機能だからである。

異なるカレンダーシステム間で招待機能を用いる例を図 3.1 に示す。図 3.1 は、Yahoo!カレンダーと Google カレンダーの間で、ウェブインターフェース部を用いた招待機能の例である。ユーザ A とユーザ B は Yahoo!カレンダーの利用者、ユーザ C は Google カレンダーの利用者である。ユーザ A が Yahoo!カレンダーの招待機能を使い、ユーザ B とユーザ C を招待する流れを述べる。

(1) 招待者が予定を登録

ユーザ A がブラウザを用いて Yahoo!カレンダーのカレンダーサーバに予定を登録する。この時、ユーザ B とユーザ C のメールアドレスを予定の参加者として追加する。

(2) 招待メールを送信

予定の参加者に対し、招待機能を使って招待メールを送信するかどうかを選択できる。招待メールの送信を選択すると、ユーザ B とユーザ C に招待メールが送信される。

(3) 招待メールのリンクをクリック

ユーザ B とユーザ C に、予定の情報が書かれたメールが届く。メールには、『Yahoo!カレンダーに追加』リンクが記述されている。ユーザ B とユーザ C は、招待された予定を自分のカレンダーに登録する場合、『Yahoo!カレンダーに追加』リンクをクリックする。

(4) 被招待者が予定を登録

ユーザ B は Yahoo!カレンダーユーザのため、メールの『Yahoo!カレンダーに追加』リンクをクリックすることで、自身の Yahoo!カレンダーに予定を登録できる。しかし、ユーザ C は Yahoo!カレンダーユーザではないため、『Yahoo!カレンダーに追加』リンクをクリックしても、自身の Google カレンダーに予定を登録できない。このため、ユーザ C は招待された予定としてではなく、ブラウザを用いて手入力で新規の予定として登録しなければならない。

ユーザ C は、新規の予定として登録するため、この予定の UID は招待元の予定の UID とは異なる。このため、招待元の予定が変更されても、同期すべき予定が分からず、ユーザ C の予定は同期できない。Google カレンダーのユーザから Yahoo!カレンダーのユーザを招待した場合も、同様に予定を共有できない。

上記の例は、カレンダーサーバの Web インターフェース部を利用した場合である。CalDAV を用いる場合も同様に、異なるカレンダー AP を用いると招待機能を使うことが出来ず、予定の UID が異なってしまうため、予定を共有できない。このように、利用するカレンダーシステムが異なると、招待機能を用いて予定を共有できない事は問題である。

3.3 共有カレンダーによる過去の予定の喪失

カレンダーは、未来の予定表としての意味だけでなく、過去の行動履歴としての意味も持つ。2.2.2 項の (2) で述べた「共有カレンダーを使う方法」で予定を共有していた場合、自分の過去の予定を喪失する可能性がある。

グループからの脱退により、グループの共有カレンダーへのアクセス権限を失う様子を図 3.2 を用いて説明する。ユーザ A はあるグループに所属している。このグループには共有カレンダーがあり、グループの予定は共有カレンダーで管理されている。ユーザ A はグループに所属している間は、個人用のカレンダーとグループでの共有カレンダーを合わせて自分のカレンダーとして管理する。しかし、グループを脱退すると、グループでの共有カレンダーへのアクセス権を失う。これにより、グループに関わる予定がユーザ A のカレンダーから消えてしまい、過去に自分がグループで何をしたかを閲覧できなくなる。

グループの脱退により、過去の予定を喪失することは、妥当な場合もあり得る。例えば、企業で共有カレンダーを利用していた場合、退社と同時に共有カレンダーへのアクセス権を失い、過去の予定を喪失することは妥当である。しかし、過去の予定を喪失させるか、喪失させないかを組織や個人で選択できないことは問題である。

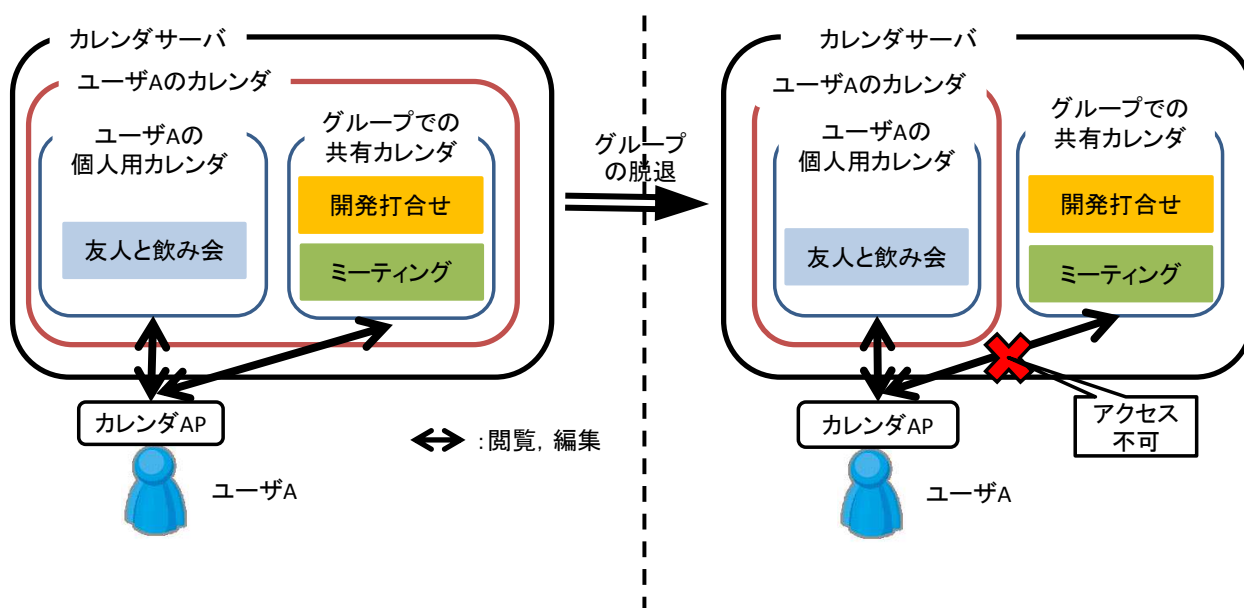


図 3.2 共有カレンダーへのアクセス権限の喪失

3.4 公開設定による相手に応じた予定の見せ方の変更が不可能

予定を見せる相手によって、見せたい予定や隠蔽したい予定が異なる。しかし、既存カレンダーシステムにおける公開設定では、相手に応じて予定の見せ方を変更できない。

既存カレンダーシステムで、この問題へ対処するには、見せ方の数だけ別々のカレンダーを用意する必要がある。複数のカレンダーを用いて、相手によって見せ方を変える様子を図3.3を用いて説明する。ユーザAには『開発打合せ』と『友人と飲み会』の2つの予定がある。2つの予定のうち、友人であるユーザBには『開発打合せ』の予定の詳細を伝えたくないが、仕事の予定があることは伝えたい。一方で、『友人との飲み会』の予定は詳細を伝えたい。同僚であるユーザCには、『友人との飲み会』の予定の詳細を伝えたくないが、プライベートな予定があることは伝えたい。一方で、『開発打合せ』の予定は詳細を伝えたい。このような場合、ユーザAは仕事やプライベートの予定のタイトルを相手の属性に相応しい表現に書き換えた、プライベート用のカレンダーと、仕事用のカレンダーを別々に用意する必要がある。

複数のカレンダーを管理すると、管理に手間がかかるだけでなく、予定の書き間違いといったミスも起きるため、問題である。

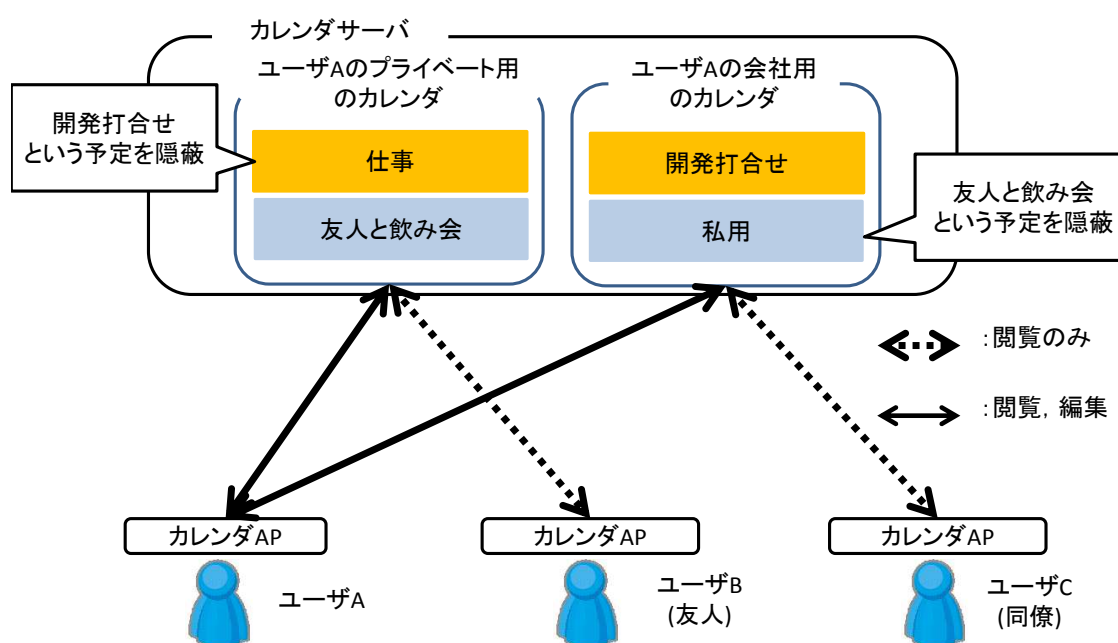


図 3.3 相手に応じた予定の見せ方の変更

第 4 章

対処

4.1 対処に必要な条件

前章で、複数人でカレンダーシステムを利用する際の問題点として以下の 3 つを述べた。

問題 1 招待機能によるカレンダーシステム間の予定の共有が不可能

問題 2 共有カレンダーによる過去の予定の喪失

問題 3 公開設定による相手に応じた予定の見せ方の変更が不可能

これらの問題を解決するために必要な条件について述べる。

(1) カレンダーシステム間の連携が可能

カレンダーサーバやカレンダー AP には変更を加えずに、カレンダー AP やカレンダーサーバが持つ固有の機能の連携を可能にする。

(2) 自身のカレンダーで複数人に関わる予定の管理が可能

共有カレンダーを用いずに、自身のカレンダーだけで全ての予定を管理する。自身のカレンダーの予定を、他者と共有する仕組みが必要となる。

(3) 相手に応じた予定の見せ方の変更が可能

個々の予定の見せ方を変更するための情報を付加し、相手に応じた予定の見せ方の変更を可能にする。

これらの条件を満たす仕組みとして、カレンダーサーバとカレンダー AP の中継を行う CalDAV プロキシを提案する。

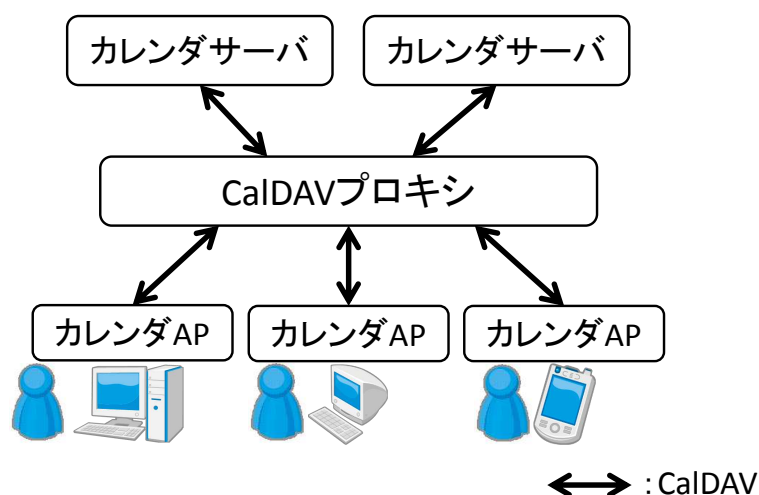


図 4.1 CalDAV プロキシの概要

4.2 CalDAV プロキシを用いた対処

4.2.1 CalDAV プロキシの概要

CalDAV プロキシは、図 4.1 に示すように、カレンダーサーバとカレンダー AP が CalDAV を用いて通信を中継する。通信を中継することにより、それぞれのカレンダーサーバやカレンダー AP に適した通信が可能になる。

CalDAV プロキシは、以下の 3 つの特徴をもつ。

- (1) カレンダー AP 側から見ると、CalDAV プロキシはカレンダーサーバとして通信する。
- (2) カレンダーサーバ側から見ると、CalDAV プロキシはカレンダー AP として通信する。
- (3) 通信するカレンダーサーバやカレンダー AP に応じて、適切な通信をする。

前節の 3 つの条件を満たすために、CalDAV プロキシに以下の 3 つの機能を持たせる。

- (1) カレンダーサーバやカレンダー AP の違いを吸収する機能

カレンダーシステム間の連携を可能にするために、カレンダーサーバやカレンダー AP の違いを吸収する機能を持たせる。これにより、CalDAV が利用できるカレンダーサーバやカレンダー AP であれば、機能や仕様の違いを意識せずに予定の共有や予定の閲覧が可能になる。

(2) 個人のカレンダーでグループの予定を管理する機能

個人のカレンダーで複数人に関わる予定の管理を可能にするために、個人のカレンダーでグループの予定を管理する機能を持たせる。この機能は、個人のカレンダーから予定を登録する際に、登録する予定がグループに関する予定か否かを判断し、グループに関する予定であればグループメンバー全員のカレンダーにも予定を登録する機能である。この機能により、招待機能や共有カレンダーといった問題のある機能を使わずに、予定を共有できる。

(3) 相手に応じて予定をフィルタリングする機能

予定をフィルタリングすることで、相手に応じて予定の見せ方を変更する機能を持たせる。これにより、閲覧させる相手によってカレンダーの見せ方を変更できる。

4.2.2 カレンダーサーバやカレンダー AP の違いを吸収する機能

カレンダーシステム間の連携を可能にするために、カレンダーサーバやカレンダー AP の違いを吸収する機能を考える。

カレンダーサーバやカレンダー AP の違いには、以下の 3 つがある。

(1) カレンダーサーバやカレンダー AP によって、カレンダー情報の扱いが異なる

例えば、iCalendar には、カレンダー AP が自由に作成できる項目 (以下、x-prop) がある。x-prop の扱いは、カレンダーサーバやカレンダー AP によって異なる。カレンダー AP やカレンダーサーバには、x-prop を破棄してしまうものがある。このため、独自の機能のためのプロパティとして x-prop を用いているカレンダーシステムがあった場合、他のカレンダーシステムによりプロパティを破棄されてしまうと、機能に支障が出る。

(2) カレンダーサーバによって、所有できるカレンダーの数が異なる

Yahoo! カレンダーでは 1 人のユーザは 1 つのカレンダーを持てる。一方、Google カレンダーでは 1 人のユーザは複数のカレンダーを持てる。

(3) カレンダー AP によって、カレンダーサーバとの接続の方法が異なる

CalDAV という通信プロトコルを用いているにも関わらず、カレンダー AP によってカレンダーサーバとの接続方法が異なる。これは、カレンダーサーバ上のディレクトリ構造を CalDAV が定義していないからである。iCalendar や CalDAV に定義されていない部分は、カレンダーサーバやカレンダー AP が自由に実装する。例えば、Apple の提供する iCal (現在はカレンダーという名称に変更された) [6] は、ユーザが持っている複数のカレ

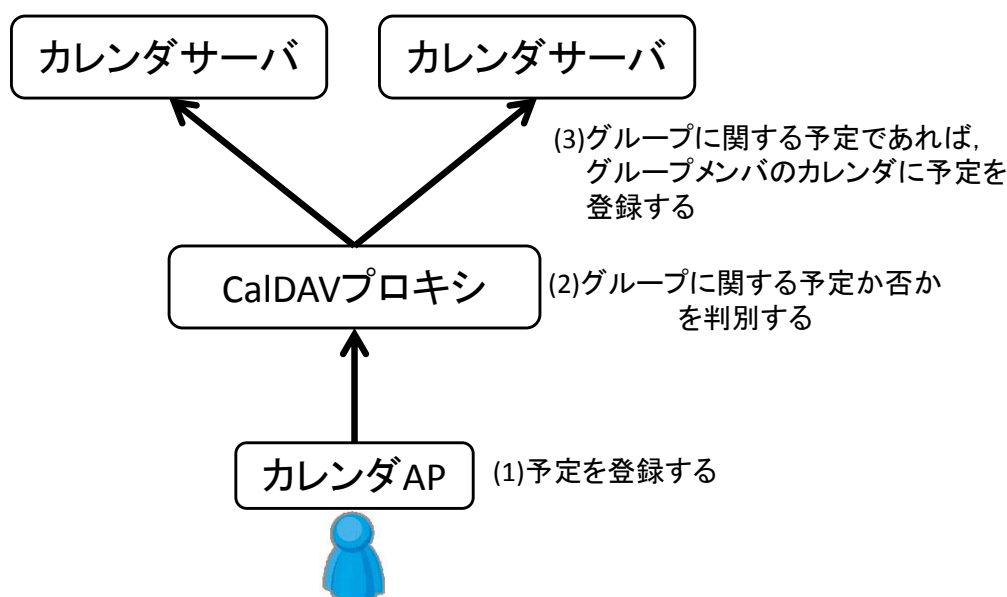


図 4.2 CalDAV プロキシを用いたグループの予定の管理

ンダを一度に登録できる。一方、Mozilla Foundation が提供する Lightning[7] は、ユーザが複数のカレンダーを持っていたとしても、カレンダーを 1 つずつしか登録できない。Google カレンダーではこの違いに対処するために、Lightning と同期するための URL と、Apple の提供するカレンダーと同期するための URL を別々に用意している。このように、カレンダー AP によってカレンダーサーバとの接続の方法が異なる。

CalDAV プロキシは、このような違いに個別に対処することで、カレンダーシステム間の違いを吸収する。これにより、様々なカレンダーシステム間での連携を可能にする。

4.2.3 個人のカレンダーでグループの予定を管理する機能

個人のカレンダーで複数人に関わる予定の管理を可能にするために、個人のカレンダーでグループの予定を管理する機能を実現する。この機能を図 4.2 を用いて説明する。まず、ユーザはカレンダー AP を用いて予定を登録する。次に、CalDAV プロキシが、この予定がグループに関する予定か否かを判別する。判別した結果、グループに関する予定であれば、予定登録者を含むグループメンバーのカレンダーに予定を登録する。グループに関する予定でなければ、予定登録者のカレンダーのみに予定を登録する。

この機能を実現するためには、個人の予定とグループの予定を識別する必要がある。個人の予定とグループの予定を識別する方法として、以下の 2 つの方法が考えられる。

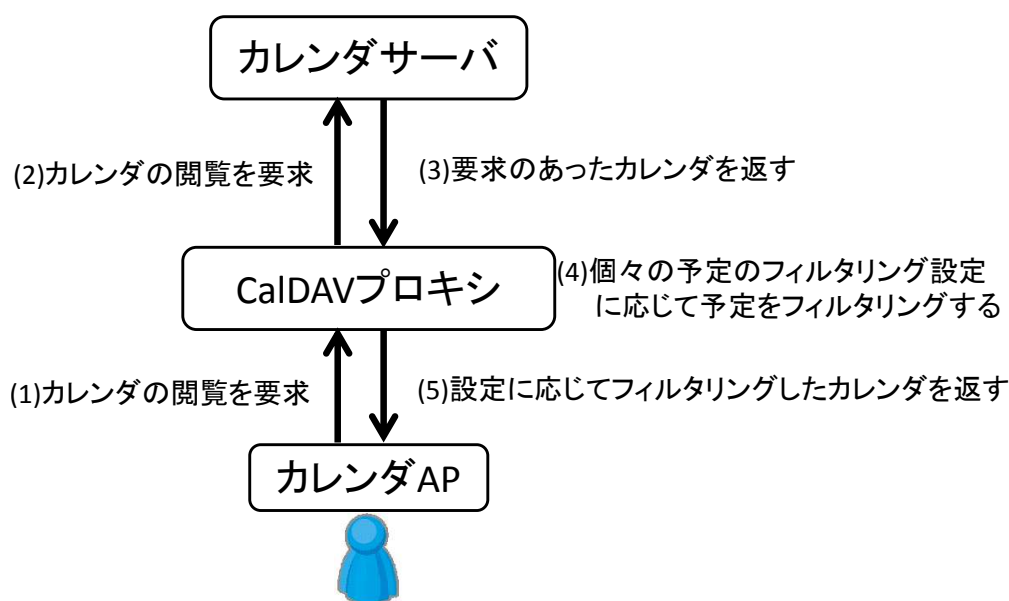


図 4.3 CalDAV プロキシを用いた予定のフィルタリング

(1) 個々の予定にグループ設定を持たせる

グループ設定を予定の DESCRIPTION に記述する。

(2) 予定のタイトルで判別する

CalDAV プロキシに、グループワードを持たせておく。予定のタイトルにグループワードが含まれているか否かで判別する。

個々の予定にグループ設定を持たせる方法では、CalDAV プロキシにグループ設定を用意する必要がないという利点がある。しかし、個々の予定について、どのグループに関連する予定なのかを予定を入力する度に DESCRIPTION に記述するのは手間である。一方、予定のタイトルで判別する方法では、事前に CalDAV プロキシにグループワードを設定しておく必要がある。しかし、予定を入力する度にグループ設定を入力する必要が無い。このため、CalDAV プロキシの存在を意識せずにカレンダー AP を利用できる。

CalDAV プロキシを利用しない場合と同様にカレンダー AP を利用できる事は重要であるため、CalDAV プロキシでは予定のタイトルで判別する方法を用いる。

4.2.4 相手に応じて予定をフィルタリングする機能

相手に応じて予定をフィルタリングする機能を図 4.3 を用いて説明する。ユーザは、カレ

ンダ AP から CalDAV プロキシに対してカレンダーの閲覧を要求する。CalDAV プロキシは、要求のあったカレンダーを保持しているカレンダーサーバに対して、カレンダーの閲覧を要求する。カレンダーサーバは、要求のあったカレンダーを返す。次に、CalDAV プロキシはカレンダーの個々の予定にフィルタリングが必要か否かを判断し、必要に応じて予定をフィルタリングする。そして、予定をフィルタリングしたカレンダーをユーザに返す。

予定をフィルタリングするかどうかを判断する方法は、4.2.3 項の「個人のカレンダーでグループの予定を管理する機能」と同様に 2 つある。

(1) 個々の予定に公開設定を持たせる

公開設定を予定の DESCRIPTION に記述する。

(2) 予定のタイトルで判別する CalDAV プロキシに、公開設定、公開ワード、および非公開ワードを持たせておく。予定のタイトルに公開ワードや非公開ワードが含まれているかに応じて、予定の見せ方を変更する。

4.2.3 項と同様に、CalDAV プロキシを利用しない場合と同様にカレンダー AP を利用できる事は重要であるという理由から、予定のタイトルで判別する方法を用いる。

第 5 章

設計

5.1 システム構成

CalDAV プロキシのシステム構成を図 5.1 に示す。以下に、システムの各要素について詳細を述べる。

(1) CalDAV サービス部

CalDAV サービス部は、カレンダー AP からリクエストを受信する。受信したリクエストとアクセスのあった URI をパラメータとして、レスポンス処理部に渡す。また、リクエストに対するレスポンスをレスポンス処理部から受け取り、カレンダー AP に送信する。

(2) カレンダー情報処理部

(A) リクエスト処理部

CalDAV サービス部から受け取ったリクエストを処理する。リクエスト処理部に、個人のカレンダーでグループの予定を管理する機能を実装する。

(B) レスポンス処理部

CalDAV クライアント部から受け取ったレスポンスを処理する。レスポンス処理部に、予定をフィルタリングする機能を実装する。

(3) CalDAV クライアント部

CalDAV クライアント部は、リクエスト処理部からリクエストとカレンダーサーバの URI を受け取る。受け取ったリクエストを URI に応じたカレンダーサーバに送信する。また、

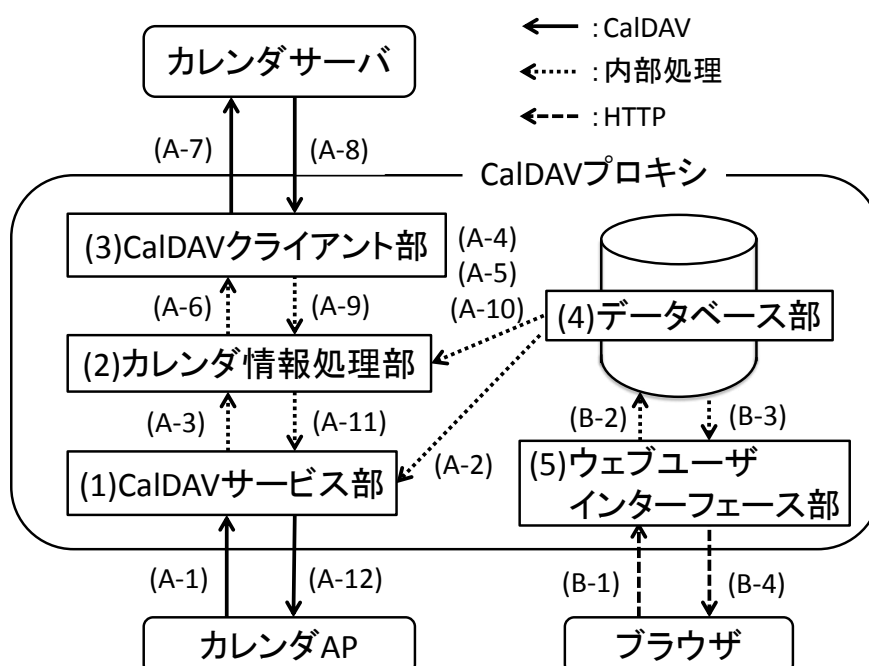


図 5.1 CalDAV プロキシの構成

カレンダーサーバからのレスポンスを受信し、レスポンス処理部に渡す。カレンダーに対して行う処理をリクエストとしてカレンダーサーバに送信する。また、リクエストに対するカレンダーサーバからのレスポンスを受信する。

(4) データベース部

通信の中継や予定のフィルタリングに必要な情報を保持する。データベース部で管理する情報を図 5.2 を用いて説明する。

(A) CalDAV プロキシユーザ

CalDAV プロキシユーザは、ユーザ名とパスワードを持つ。また、1 個以上のカレンダーを持ち、複数のグループに所属できる。

(B) グループ

グループは、グループ名を持つ。カレンダーは、グループに対するカレンダー情報の見せ方をカレンダー設定として持つ。このため、グループはカレンダー設定と関連する。

(C) カレンダー

カレンダーは、カレンダー名、URI、カレンダーユーザ名、およびカレンダーパスワードを持つ。URI、カレンダーユーザ名、およびカレンダーパスワードは、カレンダーサーバに

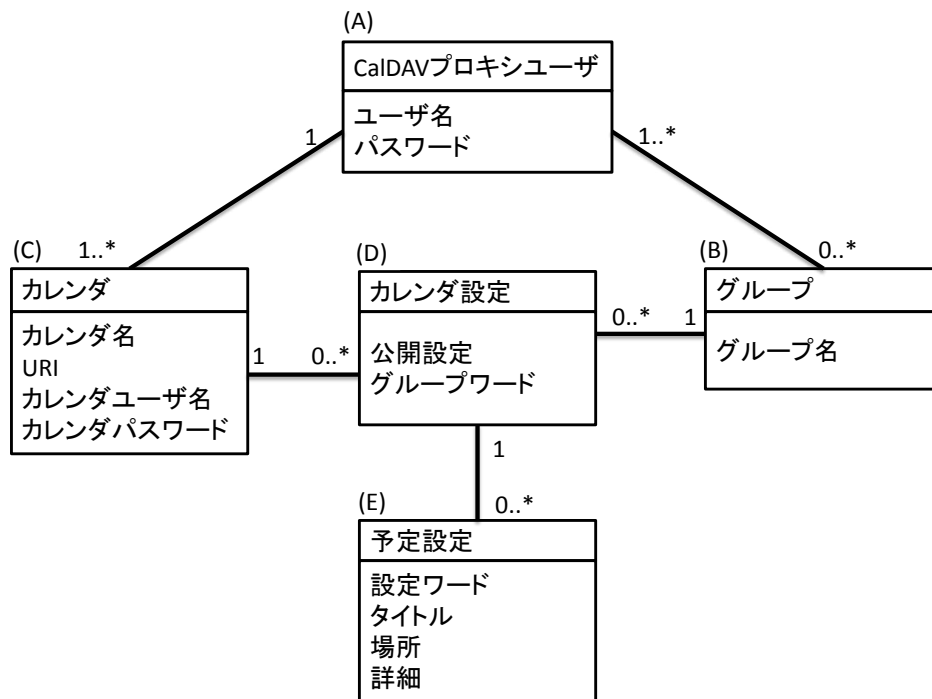


図 5.2 データベースの構成

接続するために必要な情報である。CalDAV プロキシでは、カレンダーを閲覧するユーザの所属するグループに応じてカレンダーの見せ方を変える。このため、ユーザが所属するグループの数だけカレンダー設定を持つ。

(D) カレンダー設定

カレンダー設定は、公開設定とグループワードを持つ。ユーザは、個々のカレンダーごとに、ユーザが所属するグループと等しい数のカレンダー設定を持ち、グループに対するカレンダーの見せ方を設定する。公開設定では、カレンダーを基本的に非公開にするか、公開にするかを設定する。グループワードは、個人の予定とグループの予定を識別するためのワードである。また、カレンダー設定は、予定の公開設定として予定設定を持つ。

(E) 予定設定

予定設定は、設定ワード、タイトル、場所、および詳細を持つ。予定のタイトルに設定ワードが含まれる場合に、予定をフィルタリングする。予定をフィルタリングする場合に予定をどのように見せるかについて、タイトル、場所、および詳細で指定する。

(5) ウェブユーザインターフェース部

ウェブユーザインターフェース部からデータベース部で管理する情報を編集する。編集する情報はデータベース部で管理する情報全てである。データベース部の構造に合わせた、階層構造を用いたユーザインターフェースを提供する。

5.2 動作概要

CalDAV プロキシの動作概要について説明する。図 5.1 の矢印に記されている英数字は、以下で述べる動作概要の説明と対応している。

CalDAV プロキシを用いた、カレンダー情報の通信の流れを以下に示す。

- (A-1) ユーザはカレンダー AP を操作し、カレンダー AP が CalDAV プロキシにリクエストを送信する。
- (A-2) CalDAV サービス部は、CalDAV プロキシユーザ名と CalDAV プロキシパスワードをデータベース部から受け取り、認証を行う。
- (A-3) CalDAV サービス部は、アクセスされた URI とリクエストをカレンダー情報処理部に転送する。
- (A-4) データベース部は、アクセスされたカレンダーの公開設定に関する情報をカレンダー情報処理部に転送する。
- (A-5) データベース部は、リクエストを送信する先の URI と、カレンダーサーバでの認証に必要な情報をカレンダー情報処理部に転送する。
- (A-6) カレンダー情報処理部は、カレンダー情報処理部が処理したリクエストと、カレンダーサーバへの通信に必要な情報を CalDAV クライアント部に転送する。
- (A-7) CalDAV クライアント部は、リクエストをカレンダーサーバに送信する。
- (A-8) CalDAV クライアント部は、カレンダーサーバからレスポンスを受信する。
- (A-9) CalDAV クライアント部は、レスポンスをカレンダー情報処理部に転送する。
- (A-10) データベース部は、予定の公開設定をカレンダー情報処理部に転送する。

(A-11) カレンダ情報処理部は、予定の公開設定をもとにフィルタリングしたレスポンスを CalDAV サービス部に転送する。

(A-12) CalDAV サービス部はレスポンスをカレンダ AP に送信する。

複数人に関する予定だった場合は、関係する人数の分、(A-5) から (A-9) まで繰り返す。

ウェブユーザインターフェースを用いた、データベース部の管理の流れを以下に示す。

(B-1) ユーザがブラウザを用いてウェブユーザインターフェース部にアクセスし、データベースの変更を依頼する。

(B-2) ウェブユーザインターフェース部はデータベース部にデータベースの変更を指示する。

(B-3) データベース部は、変更の指示に対する応答をウェブユーザインターフェース部に通知する。

(B-4) ウェブユーザインターフェース部は、変更の依頼に対する結果として、レスポンスコードを含むレスポンスをブラウザに送信する。

第 6 章

実装

CalDAV プロキシのプロトタイプを作成した。プロトタイプの実装環境を表 6.1 に示す。

表 6.1 プロトタイプの実装環境

OS	OSX 10.8.2
使用言語	Ruby 1.9.3
フレームワーク	Ruby on Rails 3.2.6
ライブラリ	net/http

プロトタイプでは、システム構成のうち、ウェブユーザインターフェースを除く、CalDAV サービス部、カレンダー情報処理部、CalDAV クライアント部、データベース部を作成した。CalDAV サービス部としての機能は、Web アプリケーションフレームワークである Ruby on Rails を利用した。カレンダー情報処理部の機能は、4.2 節で述べた 3 つの機能うち、「個人のカレンダーでグループの予定を管理する機能」と「相手に応じた予定をフィルタリングする機能」の実装を行った。CalDAV クライアント部としての機能は、汎用データ転送プロトコル HTTP を扱うライブラリである net/http を利用した。

実装によって、CalDAV による通信を中継し、設計通りにカレンダー情報を書き換えられる事を確認した。

第 7 章

おわりに

本論文では、まず、カレンダーシステムを複数人で利用する場合の問題点として以下の 3 つを挙げた。

- (1) 招待機能によるカレンダーシステム間の予定の共有が不可能
- (2) 共有カレンダーの利用による過去の予定の喪失
- (3) 公開設定による相手に応じた予定の見せ方の変更が不可能

次に、これらの問題点に対処するための仕組みとして、CalDAV プロキシを提案した。CalDAV プロキシは、カレンダーサーバとカレンダー AP の通信を中継する。3 つの問題点に対処するため、CalDAV プロキシに以下の 3 つの機能を持たせる。

- (1) カレンダーサーバやカレンダー AP の違いを吸収する機能
- (2) 個人のカレンダーでグループの予定を管理する機能
- (3) 相手に応じて予定をフィルタリングする機能

また、CalDAV プロキシの設計を行い、プロトタイプを作成した。プロトタイプでは、3 つの機能のうち、(2) 個人のカレンダーでグループの予定を管理する機能と、(3) 相手に応じて予定をフィルタリングする機能を実装した。これによって、CalDAV プロキシが CalDAV の通信を中継し、カレンダー情報を書き換えられる事を確認した。

残された課題として、(1) カレンダーサーバやカレンダー AP の違いを吸収する機能の実装と、CalDAV プロキシの評価がある。

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，山内利宏准教授ならびに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] Desruisseaux,B.: Internet Calendaring and Scheduling Core Object Specification(iCalendar), RFC5545, 2009.
- [2] Daboo,C., Desruisseaux,B. and Dusseault,L.: Calendaring Extensions to WebDAV (CalDAV), RFC4791, 2007.
- [3] Goland,Y., whitehead,E., Faizi,A., Carter,S. and Jensen,D.: HTTP Extensions for Distributed Authoring – WEBDAV, RFC2518, 1999.
- [4] Google Inc. : Google カレンダー, 入手先 [〈http://www.google.com/calendar/〉](http://www.google.com/calendar/) (参照 2013-02-07) .
- [5] Yahoo! Inc. : Yahoo!カレンダー, 入手先 [〈http://calendar.yahoo.co.jp/〉](http://calendar.yahoo.co.jp/) (参照 2013-02-07) .
- [6] Apple Inc. : iCal (カレンダー), 入手先 [〈http://www.apple.com/jp/osx/apps/〉](http://www.apple.com/jp/osx/apps/) (参照 2013-02-07) .
- [7] Mozilla Foundation: Lightning - Project Home, available from [〈http://www.mozilla.org/projects/calendar/lightning/index.html〉](http://www.mozilla.org/projects/calendar/lightning/index.html) (accessed 2013-02-07).