

推薦論文

携帯端末用メッセージングシステムにおける分断時操作の同期機能の拡張

齋藤正史[†] 吉田玲子[†]

本論文ではメッセージングシステム、とりわけ電子メールシステムにおける据置型コンピュータ(PC)と携帯端末の両者からのメッセージ操作の並行性を向上するための同期機能の拡張手法について提案する。複数の端末よりアクセスされるメッセージストアの同期操作における不正な更新操作の実行を妨げるために、一意識別子と一意識別子正当性評価値を用いた手法が提案されている。この手法は、操作コンフリクトの発見には十分なものであるが、フォルダ単位での同期手法であるため利用者に必要以上の同期操作の実行を課すことになる。そこで、全域一意識別子とメッセージバージョン番号を利用した同期機能の拡張手法を提案する。本手法を用いることにより、モバイル環境の携帯端末とオンライン環境のPCでの並行アクセスによる遅すぎる同期操作をメッセージ単位で発見可能となり、結果的に利用者にとっての並行性が向上することができる。

Disconnected Operations' Synchronization Facility Extension for Mobile Terminal's Messaging System

MASASHI SAITO[†] and REIKO YOSHIDA[†]

In this paper, we propose novel data synchronization method that improves concurrency of messaging system operations from a desktop computer and a mobile terminal. To prevent inconsistent update on synchronization operations to a message store that is accessed from some terminals, a method that uses unique identifier (UID) and validity value of unique identifier (VUID) was proposed. This method is enough one to find out operation conflict but executes unnecessary synchronization operations for users because synchronization checking is done for each folder. We propose extended synchronization method, which uses universal unique identifier (UUID) and message version number (MVN). Using our method, late synchronization operations can be found for each message. As a result, messaging systems are able to have much more concurrency.

1. はじめに

本論文ではメッセージングシステム、とりわけ電子メールシステムにおける据置型コンピュータ(PC)と携帯端末の両者からのメッセージ操作の並行性を向上するための同期機能の拡張手法について提案する。PCからメッセージストアを保持しているコンピュータへのメッセージ操作に対しては、メッセージストアへのオンライン環境を仮定したシステム構築が行える。ところが、携帯端末からのメッセージ操作は、利用者の移動や通信環境の変化、費用の問題などにより、メッセージストアへの接続は一時的なものであるとしか仮

定できない。

一方、携帯端末における利用者の利便性を向上するためには、メッセージストアに保存されているメッセージの一部の複製を携帯端末に保持する必要があるだけでなく、複製データに対するメッセージ操作を可能とする必要がある。ところが、これらの操作がオンライン時のみ実施できるという仮定は現実的でなく、非接続時、すなわち分断時(disconnected)においてもメッセージの参照、他フォルダへの移動、メッセージ消去などのメッセージ操作が実行できる環境を提供する必要がある。

携帯端末上の複製メッセージへの分断時操作が可能

[†] 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

本論文の内容は1998年11月のDPSワークショップにて報告され、DPS研究会主査により情報処理学会論文誌への掲載が推薦された論文である。

な環境においては、オンライン状態となった際にデータの同期操作を実行しなければならない。携帯端末からの同期操作はある時点でのメッセージストア内のメッセージに対するものであり、同期操作時点でのメッセージストアの状態に対するものではなく、不正な更新操作の可能性を内在している。

本論文では、まずメッセージングシステムのアクセス形態の分類により分断時操作の定義を行う。次いで、分断時操作の必要性を明確とするために、我々が実装したクライアントソフトウェアとその評価結果を述べる。さらに、評価で問題となった分断時操作を実現するためのシステムモデルを定義し、同期操作において不正な更新操作の実行を妨げるための同期機能の拡張手法について提案し、提案アルゴリズムの正当性を示す。

2. メッセージアクセス形態

2.1 メッセージアクセスの3形態

メッセージアクセスを行うクライアントプログラムがメッセージストアにアクセスする手法は3種類に分類できる^{1),2)}。それらは、オフライン操作、オンライン操作、分断時操作である。

オフライン操作において、クライアントプログラムはメールユーザエージェント(MUA)としてメッセージストアに定期的にアクセスし、メッセージストアのメッセージを取得し、メッセージストアから取得済みのメッセージを消去する。したがって、すべてのメッセージに対する操作はクライアントコンピュータに保存されたメッセージに対して実行される。インターネットプロトコルとしてこのオフライン操作を提供するものとしては、Post Office Protocol³⁾(POP)が一般的に利用されている。

ところが、POPを使用した場合には、すべてのメッセージはあるクライアントコンピュータに保存されることになるため、複数のクライアントコンピュータから利用することができない。他のコンピュータからメッセージストアにアクセスするためには、MUAからのアクセス時にメッセージストアのメッセージを消去せずに保存しておくことで対処可能となる。しかし、この場合にはフォルダへの分類、既読フラグの付与などの操作結果を得ることはできない。さらに、すべてのメッセージをクライアントコンピュータに保存するために、小規模なクライアントコンピュータでは利用できないなどの欠点がある。

オンライン操作では、メッセージストアにすべてのメッセージが保存されており、MUAはメッセージストアを遠隔から操作する。この際、MUAが動作する

コンピュータや端末はつねにメッセージストアにアクセスできる必要はないが、メッセージストアにアクセスするにはつねにオンライン状態でなければならない。この制約はオフィスなどからのみメッセージストアにアクセスする場合には問題とはならない。ところが、出先などからアクセスした場合には、公衆回線網などを利用してオンライン状態を維持しておく必要がある。したがって、実際的な利用においては費用や回線環境の問題により実現困難である。

分断時操作はメッセージストアにアクセスした際に、一部のメッセージを複製としてMUAが動作するコンピュータに保存する場合の操作である。MUAはメッセージの複製に対して操作を行うことが可能であり、次にメッセージストアにアクセスした際に同期が行われる。この分断時操作を提供するインターネットプロトコルとしてInternet Message Access Protocol Version4⁴⁾(IMAP4)が注目をあびている。

2.2 IMAP4プロトコル

IMAPはメッセージストアへの遠隔アクセスプロトコルとして1986年にスタンフォード大学で開発され、現行版の仕様はRFC2060⁴⁾で規定されている。

MUAはIMAP4プロトコルを用いてメッセージストアのメッセージとフォルダに対して以下の操作を行うことができる。

- (1) フォルダの選択, 作成, 削除, 名前の変更, 一覧表示
- (2) メッセージの検索, 取得, 移動, 削除
- (3) ログイン, ログアウト, ユーザ認証

メッセージの検索や取得時には、複雑なフィルタリングや検索が可能であり、必要なメッセージを選択するだけでなく、メッセージの必要な部分のみを選択して取得することが可能である。このメッセージ部位の指定には、MIME⁵⁾の各部位が指定可能であり、少量のテキストに添付された大量のイメージデータよりなるメッセージのテキスト部分のみを取得できる。つまり、IMAP4を用いることにより、利用者が必要となる情報を利用者自身が指定し、そのデータのみを取得することができる。

3. 分断時操作の必要性

3.1 メッセージキャッシュを提供するIMAP4クライアントの実装

我々は、モバイル環境での携帯端末利用によるタイムリーな情報交換を行うために、携帯端末上にメッセージキャッシュを提供するIMAP4クライアントとなるMUAを実装した⁶⁾。本実装は、企業内メッセージン

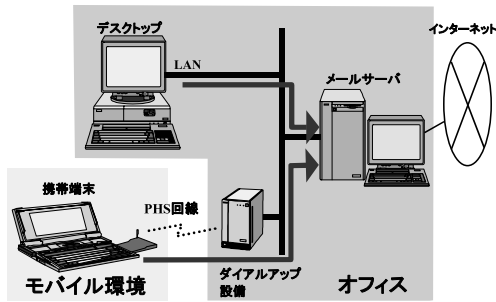


図1 IMAP4クライアント利用環境
Fig. 1 IMAP4 Client usage environment.

グシステムを対象とし、同一アカウントでオフィスのPCからだけでなく、企業内外の多くの端末や出張先のモバイル環境からもメッセージストアへのアクセスを可能とするものである。企業内メッセージングシステムとして、図1に示す利用環境を想定している。社内にIMAP4サーバを構築し、メッセージストアをIMAP4サーバによって管理する。オフィスのPCからは企業内LAN経由でメッセージストアにアクセスし、モバイル環境の携帯端末からはPHS回線による接続により、同一メッセージストアにアクセス可能とする。これにより、オフィスのPC、モバイル環境の携帯端末の両方から、メッセージストアを使用することができ、操作の二重化、データの重複を防ぐことが可能となる。ところが、接続は公衆回線経由で行うために費用がかかる。費用低減の手法として、オフライン送信機能、キャッシュ機能を実装している。本実装の特徴を以下に示す。

(1) オフライン送信機能

回線接続時間を節約するために、オフライン中のメッセージ作成、編集機能を実装した。ローカルの携帯端末上に送信用フォルダを設け、オフライン中に利用者が作成・編集した送信メッセージを送信フォルダに保存する。その後、MUAがオンライン状態を確認したならば、送信フォルダに蓄えられた送信メッセージは、自動的に一括送信される。

(2) キャッシュ機能

オフライン時にメッセージ操作を行うためには、メッセージの一部をMUAが動作するコンピュータに複製として保存しておく必要がある。メッセージストアのメッセージをつねにオリジナルと見なし、クライアントは読み出したメッセージを順にキャッシュとしてファイル保存していく。

図2にクライアントでの動作を示す。利用者がメッセージ読み取り操作を行うと、MUAは接続状態によってメッセージの読み取り先を自動的に変更する。オン

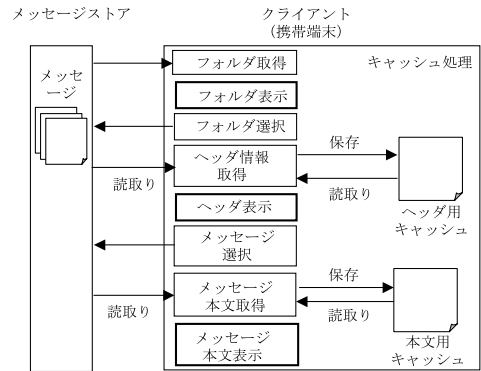


図2 キャッシュ処理の動作
Fig. 2 Cache data processing.

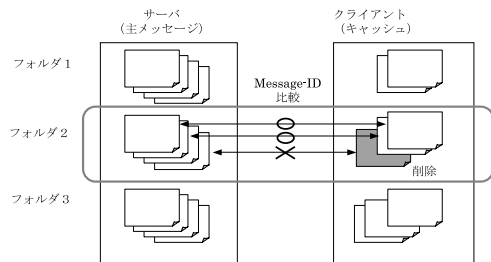


図3 メッセージ同期
Fig. 3 Message Synchronization.

ライン中のメッセージ読み取りはメッセージストアから、オフライン中にはキャッシュから読み取る。これにより、利用者は回線接続状態を意識することなくメッセージにアクセス可能である。

携帯端末は記憶容量が小さいため、全メッセージをダウンロードして保存するオフライン操作を仮定した方式は適さず、必要なメッセージのみを保存するキャッシュ方式によって利用資源の節約が可能となる。また、不要なメッセージのダウンロードを行わないために接続時間の削減も行える。

(3) メッセージ同期

キャッシュ機能を持つことにより、メッセージストアのメッセージの一部をクライアントコンピュータでも保持することになるため、メッセージストアとキャッシュメッセージの同期が必要となる。両者のメッセージが同一であるかどうかを検査するタイミングは、利用者がフォルダを選択することによって生じるフォルダ内メッセージのヘッダ情報取得処理の際に、フォルダ単位で行っている(図3)。

メッセージの同一性の判別方法として、メッセージ固有識別子であるRFC822⁷⁾の“Message-ID”フィールドを用いている。Message-IDは、一般に送信年月日とランダム番号とサーバ名から成り立っているため、

一意と見なすことができる。この仮定は、企業内システムにおいては満足のいくものである。

Message-ID の比較により、ローカルにキャッシュされているメッセージとメッセージストアのメッセージが同一でなければキャッシュを消去することにより、キャッシュ上のメッセージがメッセージストアのサブセットであることを保証する。非接続状態でメッセージ移動や削除などの分断時操作は、メッセージストアとの同期が問題となるために実装していない。

3.2 IMAP4 クライアントの評価

我々が開発した IMAP4 クライアントを実際に利用し、機能評価を行った。他の評価要素として、ユーザインタフェースや性能なども考えられるが、本論文では図 1 に示す想定環境で十分に利用可能であるかという点のみ評価結果を示す。

オフライン時のメッセージの参照については、既参照メッセージがキャッシュに保存されているため、モバイル環境においても、いつでも参照可能となる。また、返答の作成もオフラインで行うことが可能であり通常の利用には十分である。さらに、オンライン時に新着メッセージに対する他フォルダへの移動操作や削除操作などを、モバイル環境で実行することによりオフィスでの操作が不要となり、非常に使いやすいものとなっている。実際、オフィス、モバイル環境、自宅の 3 カ所からメッセージストアにアクセスを行う場合においても、円滑に最新状態を得ることができることを確認した。

ところが、現在の実装ではメッセージ移動や削除などの分断時操作を提供していない。これらの機能の利用はオンライン操作として実行すれば十分であると想定していたが、実際の利用においては問題があることが分かった。公衆網への接続時間を低減するような利用では、以下の順序による操作が頻繁に実行される。

- (1) メッセージストアに接続し、新着メッセージのヘッダを取得する。
- (2) ヘッダ一覧を取得し、読み取りたいメッセージを指定し、キャッシュに保存する。
- (3) メッセージストアと切断する。
- (4) キャッシュ上のメッセージを参照する。
- (5) 返事を作成する必要のあるメッセージについては、返事を作成し送信する。メッセージは送信フォルダに保存され、次回オンライン状態となった時点で送信される。
- (6) 既読のメッセージを適切なフォルダに移動する、または不要なメッセージを消去する。

ここで、上記の (1)~(5) については、現行の実装

で実施することが可能であるが、分断時操作である操作 (6) が実行できない。現状では、次回に接続した際に操作 (6) を実行することになるが、受信フォルダに多くのメッセージが残されてしまい、面倒な作業になってしまう。

したがって、モバイル環境でのメッセージ操作を円滑に実施するためには、分断時操作の提供が必要であることが分かった。ここで、分断時操作を提供するためには、クライアントとメッセージストアとの同期の問題を解決する必要がある。

4. 対象システムモデル

4.1 システム構成

分断時操作を提供するための機器構成は、図 1 に示したものと同一とする。メッセージストアは、1 人の利用者に対して 1 つ提供され、複数のフォルダよりなる。フォルダには一意な名前がつけられ、互いを区別することができる。フォルダは複数のメッセージを保持し、メッセージ識別子によりフォルダ内で区別できる。

オフィス環境の PC からの更新はオンライン操作であるため、メッセージストアに対して直接実行される。モバイル環境の端末からの操作は以下の順序で実行される。

- (1) メッセージストアへの接続
- (2) メッセージストアに対する同期操作
- (3) オンライン操作 (新着メッセージの取得など)
- (4) メッセージストアとの切断
- (5) 分断時操作

分断時操作を提供するために操作列を保持する実行予定リストを用意し、分断時操作の実行ごとに次回接続後の同期のための操作を追加する。したがって、オンライン操作は分断時操作に優先して実行されることになる。

4.2 メッセージ操作実行順序による不正な更新

メッセージストアへのアクセス順序はオフィス環境の PC とモバイル環境の端末間で任意である。図 4 の a) は、操作実行の順序の観点から全順序付けでき、その順序に従って逐次実行されている場合である。この場合には、メッセージストアに対して誤った操作を行うことはない。ところが、b) はモバイル環境の端末の同期操作がオフィス環境の PC でのオンライン操作に対して遅れた場合を示している。この場合には、モバイル環境の端末の同期操作は、メッセージストアの過去の状態に対する更新を行っているために、時間的には後に実行されたオンライン操作と分断時操作の実行順序が入れ替わっている。この状態で、実行予定リストの操作

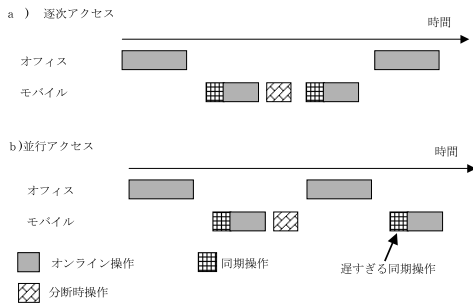


図4 メッセージストアへのアクセス順序
Fig. 4 Access order to a message store.

の実行したならば、不正な更新となる可能性がある。この不正な更新を妨げるための方策が必要となる。

4.3 メッセージ操作の分類

メッセージ操作は、フォルダに対する操作とメッセージ自身に対する操作、メッセージ作成操作とに分類できる。本論文では、各操作は特定のメールプログラムやプロトコルに依存しないものとしている。

フォルダ操作は、フォルダ内メッセージ一覧表示 (scan)、フォルダ名変更 (rename-folder)、フォルダ作成 (new-folder)、フォルダ削除 (delete-folder)、フォルダの選択 (select-folder) よりなる。

select-folder は、MUA が以降のメッセージアクセスの対象を指定するためのものであるとともに、フォルダの更新が原始性を満足した更新であることを保証するために利用される。

メッセージ操作は、選択されたフォルダのメッセージに対して実行され、他フォルダへの移動 (refile)、メッセージ削除 (delete)、メッセージ表示 (show)、メッセージの追加 (append)、読み取り済みや返答済みなどのメッセージフラグ変更 (store-flag) よりなる。

メッセージ作成操作は、特殊なフォルダ「Send」に対するメッセージの追加であるが、最終的にはメッセージストアへの直接的な更新は行わず、他の送信手順によりメッセージ送信が行われるため、本論文では対象としない。

4.4 メッセージアクセスに対する仮定

以降では、分断時操作としてメッセージ操作に注目する。実際、3章で説明した IMAP4 クライアントにおいても、フォルダ名変更、フォルダ作成、フォルダ削除はモバイル環境ではほとんど利用されていない。また、メッセージ削除 (delete) は、特殊なフォルダ「Trash」に対するメッセージの移動として扱うこととする。

したがって、分断時のメッセージ操作として、scan、show、append、refile、store-flag の 5 つの操作を対象とする。

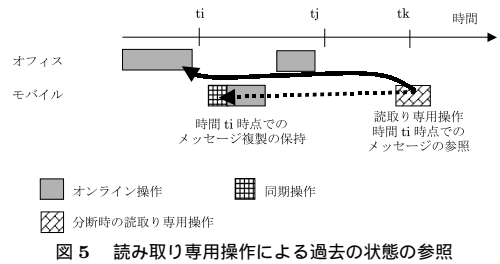


図5 読み取り専用操作による過去の状態の参照
Fig. 5 Referring the past status by a read only operation.

scan 操作と show 操作は読み取り専用操作である。読み取り専用操作は、同期されたメッセージストアのある時点での状態、ならびに分断時操作により更新されたクライアントでの一時的な状態の参照のみを行い、メッセージストアに対する更新をとまわらない。したがって、メッセージ同期に関しては何の影響も及ぼさず、実行予定リストへの追加も行われぬ。図5において、これらの読み取り操作の時間 t_k での結果は、時間 t_j での最新のメッセージストアの状態への参照ではなく、時間 t_i 時点での状態への参照、すなわち過去の状態の参照であるけれども分断時操作の結果として正当なものであると定義する。

append 操作はフォルダにメッセージを追加するものである。実際には異なる順序でフォルダにメッセージが追加され、すべてがオンライン操作として実行された場合と比較して、メッセージ識別子が異なるなどの軽微な相違が生じるだけである。したがって、append 操作も同期操作に対して影響を及ぼさない。append 操作は、メッセージストアの状態を変更するけれども、分断時操作の結果として正当なものであると定義する。モバイル環境の端末とオフィスの PC の時刻が完全に同期しているならば、すべての操作を全順序付けることが可能となる。ところが、複数端末の時刻を完全に同期させることは困難であるだけでなく、メッセージストアに接続していない環境を想定した場合には、意味がない。つまり、複数端末における時刻は同期していないと仮定し、物理時間を利用した順序付けは行えないとする。

メッセージストアへの更新については、図4の a) ならびに b) の両者を想定する。ただし、a) の場合には、結果的には逐次実行されつねに一貫した状態を保持できるため、以降では説明の対象としない。

4.5 操作コンフリクト

分断時操作の同期に際し、考慮すべき操作は refile 操作と store-flag 操作である。これらの操作がどのように正当性を損なうかについて図6を用いて説明する。メッセージ識別子がフォルダ内の番番として指定さ

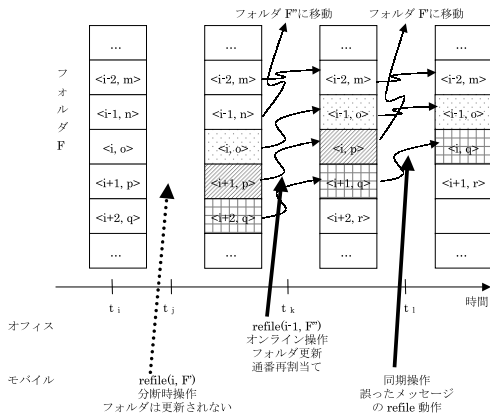


図 6 操作コンフリクトによる不正な更新

Fig. 6 Illegal update by operation conflict.

れており、メッセージが移動された場合には新たな通番が割り当てられるとする。フォルダ F のメッセージ $\langle i-2, m \rangle, \langle i-1, n \rangle, \langle i, o \rangle, \langle i+1, p \rangle, \langle i+2, q \rangle$ に注目する。各組の第 1 要素は通番であり、第 2 要素はメッセージ本体である。時間 t_i においてフォルダ F は同期されており、以下の操作がそれぞれ t_j, t_k, t_l に実行されたとする。

- t_i : モバイル環境の端末のフォルダ F とメッセージストアのフォルダ F が同期されている初期状態。
- t_j : モバイル環境の端末により、分断時操作として $\langle i, o \rangle$ をフォルダ F' に移動する。
- t_k : オフィス環境の PC により、オンライン操作として $\langle i-1, n \rangle$ をフォルダ F' に移動する。ここで、通番が再割当てされ、上記のフォルダ F のメッセージは $\langle i-2, m \rangle, \langle i-1, o \rangle, \langle i, p \rangle, \langle i+1, q \rangle$ となる。
- t_l : モバイル環境の端末からの再接続により、分断時操作の同期を実行。
ここで、同期のための操作として通番が用いられている場合には、メッセージ $\langle i, p \rangle$ をフォルダ F' に移動してしまう。

この状況は、store-flag 操作でも発生する可能性があり、異なるメッセージのフラグ変更を行ってしまう。この状態を操作コンフリクトと呼ぶ。操作コンフリクトを、以下のように定義する。

操作コンフリクト： 同期操作において、オンライン操作と分断時操作の実行順序により、異なる結果を導く操作の指定

したがって、定義により操作コンフリクトが起こらなければ、不正な更新操作は行われぬ。ところが、システムモデルにおける仮定により、オンライン操作は分断時操作に優先して実行されるため、操作コンフ

リクトが起こらない実行列を規定することはできない。操作コンフリクトは起きてしまうが、発見することが可能であるならば、利用者もしくはデフォルト動作の自動実行によりコンフリクトの解決を行うことができる。

電子メールなどのメッセージングシステムにおいては、コンフリクトは頻繁に起こらないと考えられるため、解決操作を利用者に委ねることとする。

5. メッセージ同期機能の拡張手法

5.1 一意識別子の利用

IMAP4 プロトコルでは、分断時操作を提供するために、メッセージ識別子としてフォルダ内での一意識別子 (UID) と UID による操作、UID 正当性評価値 (UID Validity Value: UIDVV) が定義されている。UID はセッションの間には変更されず、変更が起きた場合には MUA に通知される。フォルダの内容が変更された場合には、UIDVV が偽となり、MUA に再同期の必要性を通知する。ここで、UIDVV が偽の場合には操作コンフリクトが起こる可能性があることを示している。この場合には、分断時操作はすべて破棄され、利用者によるコンフリクトの解決がなされなければならない。この手法は、コンフリクトの発見のためには十分であるが、フォルダ単位での正当性の確認であるために、利用者に対して必要以上のコンフリクトの解決を課してしまう。

n 個のメッセージ $\langle 1, UID_1, m_1 \rangle, \dots, \langle n, UID_n, m_n \rangle$ よりなるフォルダ F において、各メッセージが 3 つの要素、最初の要素が通番、2 番目の要素が UID、3 番目の要素がメッセージとする。ここで、オンライン操作により、 $\langle i, UID_i, m_i \rangle$ (ただし $1 \leq i \leq n$) が delete 操作により消去され、分断時操作として、 $\langle UID_l, m_l \rangle$ (ただし $i \neq l$) がフォルダ F に移動されたとする。この refile 操作は、実際には操作コンフリクトを起こさないけれども、UIDVV は偽となってしまう。

つまり、IMAP4 の UID の付与と UIDVV を用いたコンフリクトの発見は、図 4 a) の逐次アクセスを保証するためのものであり、フォルダ単位での再同期が必要となるため、利用者が必要以上の同期操作の実行を課することになる。

5.2 全域一意識別子 (UUID) の利用

IMAP4 の UID は、UIDVV により操作コンフリクト状態を検出している。ここで、UID としてメッセージストア全域において一意な識別子を割り当てると仮定する。ここでは、区別のためにこの識別子を UUID と呼ぶ。UUID を UID として用いた場合には、refile 操作により移動されたメッセージの追跡が可能となる。

したがって、操作コンフリクトの解決処理において、UUID が指定されているならば、実際に操作コンフリクトを起こしているかを自動判別し、操作コンフリクトでない場合には、分断時操作を実行できる可能性がある。残念ながら、UUID の適用のみでは操作コンフリクトを発見できない。分断時操作として、メッセージ M がフォルダ F' に移動され、その後のオンライン操作において、フォルダ F のメッセージ M が一時的にフォルダ F' に移動され、しばらく後にフォルダ F に戻されたとする。ここで、操作コンフリクト解決処理は、メッセージ M に対して実際のコンフリクトが起きているかを判別することができない。つまり、利用者が最終的に、メッセージ M をフォルダ F もしくはフォルダ F' のどちらに移動することを意図しているかは判断することができない。

5.3 メッセージバージョン番号 (MVN) の利用

上記 UUID の使用に加えて、メッセージストアの各メッセージにバージョン番号を付与することとする。したがって、メッセージ M は $\langle Seq, UUID, Ver, m \rangle$ よりなるものとする。ここで、 Seq はフォルダ内の通番、 Ver はメッセージバージョン番号とし、メッセージ M の更新は原始性を満足したものとする。

メッセージバージョン番号は、メッセージがメッセージストアに新規保存されたときに 1 を割り当て、それ以降、該当メッセージに対する `refile` 操作、`store-flag` 操作が実行された場合に 1 ずつ増加し、メッセージ取得などの読み取り操作では変更しない。分断時操作を行うために取得したメッセージは、メッセージストアのある時点でのバージョン番号を保持している。

分断時操作を実行した場合には、実行予定リストには取得時のバージョン番号を追加する。

分断時操作として、同一メッセージに対して複数の更新操作が実行された場合には、メッセージストアでの動作同様に 1 ずつ増加させる。実行予定リストの構成を図 7 に示す。

なお、モバイル環境の端末に保存されているメッセージをオンライン参照した際に、メッセージストア内の各メッセージのバージョン番号が増加していた場合には、クライアント MUA は分断時操作作用に保持していたメッセージの複製を破棄し、新バージョンのメッセージを複製として保持する。

5.4 UUID と MVN による同期アルゴリズム

クライアント MUA は、オンラインとなった際に実行予定リストのアイテムを検査する。アイテムが存在する場合には同期操作として、実行予定リストの各アイテムに対して、以下のアルゴリズムを実行する。

UUID	Ver. No.	操作	操作引数
		<code>select-folder</code>	"INBOX"
73457	7	<code>refile</code>	"最近の話題"
83862	10	<code>store-flag</code>	"Seen"
545	136	<code>delete</code>	
83862	11	<code>store-flag</code>	"Replied"
			取得時のバージョン番号
			増加されたバージョン番号

図 7 実行予定リスト

Fig. 7 Intention list.

- (1) UUID で指定されたメッセージを検索。
- (2) 検索が失敗したならば、不正な操作として利用者に通知。
- (3) 検索されたメッセージのバージョン番号と実行予定リストのバージョン番号を比較。
- (4) バージョン番号が等しい場合には、指定された操作を実行する。これにより、メッセージストアのバージョン番号は、1 増加。
- (5) メッセージストアのバージョン番号が大きい場合には、すでに更新が実行されたと見なし、メッセージ更新のコンフリクトを利用者に通知し、実行予定リストの未処理アイテムの同一 UUID を持つアイテムに、コンフリクト発見フラグを設定。
- (6) メッセージストアのバージョン番号が小さい場合には、メッセージストアの更新もしくはクライアント MUA の実行予定リストの更新が破棄されたことを示しているため、メッセージ更新のコンフリクトを利用者に通知し、実行予定リストの未処理アイテムの同一 UUID を持つアイテムに、コンフリクト発見フラグを設定。

このアルゴリズムでは、メッセージを取得した時点でのバージョン番号を保持しており、メッセージストアにおいて更新がなされた場合にはバージョン番号が増加していることにより、遅すぎず同期による操作コンフリクトをメッセージ単位で発見することができる。操作コンフリクトの発見例を図 8 に示す。

本同期アルゴリズムの正当性を以下のように定義する。

正当性： 4.5 節で定義した不正な更新の原因となる操作コンフリクトをメッセージ単位で必ず発見することができる。

証明概要： UID と MVN による同期アルゴリズムを

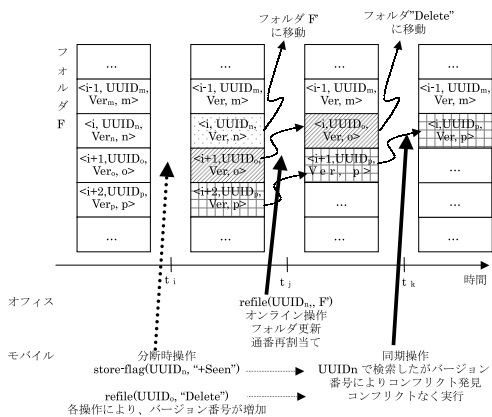


図8 UIDとMVNによる操作コンフリクトの発見

Fig.8 Finding operation conflict by using UID and MVN.

用いたとしても、操作コンフリクトをメッセージ単位で発見できないとする。

定義により、同一UIDを持つメッセージは同一である。また、メッセージストア内のメッセージを $M_s : \langle Seq_s, UID, Ver_s, m_s \rangle$ とし、実行予定リストメッセージを $M_i : \langle Seq_i, UID, Ver_i, m_i \rangle$ とする。

Case1: UIDで指定されたメッセージがメッセージストアに存在しない場合。アルゴリズムのステップ(2)により操作コンフリクトとして利用者に通知される。

Case2: UIDで指定されたメッセージがメッセージストアに存在する場合。まず、オンラインでの更新操作ならびに同期操作による更新操作の時間を t_j と表現する。

分断時操作を実行したということは、ある時間 t_0 において、クライアントMUAはオンライン状態となり、メッセージ M_s の時間 t_0 での値 $M_s[t_0]$ を保持している。したがって、時間 t_0 での $Ver_s[t_0]$ と $Ver_i[t_0]$ ならびに $m_s[t_0]$ と $m_i[t_0]$ の値は等しい。

$$Ver_s[t_0] = Ver_i[t_0]$$

$$m_s[t_0] = m_i[t_0]$$

ここで、何度かのオンライン操作ならびに分断時操作が実行され、時間 t_k ($t_k > t_0$) まで経過したとする。仮定により、この状態はある時間 t_j ($t_k > t_j > t_0$) において操作コンフリクトが発見できずに更新が実行されたことを意味する。ところが、アルゴリズムのステップ(4)・(5)・(6)により時間 t_k における実行予定リストのメッセージ M_i のバージョン番

号 $Ver_i[t_k]$ とメッセージストア内のメッセージ M_s のバージョン番号 $Ver_s[t_k]$ は等しくなければ、操作コンフリクトを発見している。メッセージの更新は原始性を満足しているとともに、更新操作ごとに Ver_s は1ずつ増加しているため、 $Ver_i[t_k - 1]$ と $Ver_s[t_k - 1]$ も等しい。同様に $Ver_i[t_i]$ と $Ver_s[t_i]$ も等しいとともに、 $m_s[t_i]$ と $m_i[t_i]$ も等しい。これは、最初の仮定に反駁している。

本論文で提案した、UIDの利用ならびにバージョン番号利用アルゴリズムは、上記のようにIMAP4プロトコルを変更せずに、MUAとメッセージストアにおける処理として実現することも可能である。しかし、多種の端末をサポートするために、オプションとして現行のIMAP4プロトコル要素のパラメータ拡張として実現することも可能である。

このようにメッセージ単位での操作コンフリクトの発見を行うことにより、利用者に対する操作の並行性が向上したこととなる。

6. 関連研究

メッセージストアなどのデータストアに対する複数クライアントからのアクセスに関するデータの一貫性に対する研究は種々の場所で行われてきた^{(8), (9)}。これらの多くは、読み込み操作と書き込み操作に対する並行性制御としてデータベースやファイルシステムの設計のために考案されたものである。ロックやバージョン番号に基づく逐次透過性の確保、タイムスタンプによる楽観的並行性制御が代表例である^{(10)~(12)}。これらのいくつかは分断時操作に対して適用可能であるが、多くがオンライン操作を仮定している。

本論文は、IMAP4の環境ではメッセージに対して限定的な操作のみが可能である点に着目している。メッセージ操作の分析に基づき、並行動作可能となる分断時操作を簡便に経済性が高く実現するための同期機能の拡張を提案している。これにより、利用者の利便性の向上を保ったまま通信回線費用の削減が可能となる。

また、IMAP4プロトコルを利用したメッセージングソフトウェアは数多く製品化されている。たとえば、マイクロソフト社のInternet Explorer(IE5.5)においては分断時操作をサポートしているとの記述がある。しかし、IE5.5においては、UIDとUIDVVを用いた実装であるために、あるフォルダのメッセージが追加された場合においても、すべてのヘッダ情報を再度読み込む必要があるだけでなく、キャッシュされていたメッセージが無効となり再ダウンロードする必要があ

り、必要以上の同期操作の実行を利用者に課している。さらに、分断時にフォルダ移動を行った場合には、移動先には消去された状態のファイルが移動されてしまう。これは、操作の順序を維持することができていないために起こったものである。

これらの課題は、本論文で提案している全域一意識別子ならびにメッセージバージョン番号を用いた同期アルゴリズムを用いることによる解決できる。

7. まとめと今後の課題

複数の端末よりアクセスされるメッセージングシステムのメッセージストアの同期操作における不正な更新操作の実行を妨げるために、全域一意識別子とメッセージバージョン番号を利用した同期機能の拡張手法を提案した。本同期手法を用いることにより、モバイル環境の端末とオンライン環境のPCでの並行アクセスにおける遅すぎる同期操作による操作コンフリクトをメッセージ単位で発見可能となり、結果的に利用者にとって並行性が向上する。

今後は、提案手法の実装と実使用による評価を通して、提案手法の実際的な効果を確認する予定である。また、本論文ではメッセージストアが各個人の電子メールを対象としたが、複数利用者で共有されるフォルダやメッセージの取扱い、ならびに実体を持たないメッセージの参照の取扱いについても取り組んでいく予定である。

参考文献

- 1) Gray, T.: Message Access Paradigms and Protocols. <http://www.imap.org/imap.vs.pop.html>.
- 2) Crispin, M.: Distributed Electronic Mail Models in IMAP4, Request for Comments: 1733 (1994).
- 3) Mayer, J. and Rose, M.: Post Office Protocol—Version 3, Request for Comments: 1939 (1996).
- 4) Crispin, M.: Internet Message Access Protocol—Version 4 rev1, Request for Comments: 2060 (1996).
- 5) Freed, N. and Borenstein, N.: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, Request for Comments: 2045 (1996).
- 6) 吉田ほか: 携帯端末向けメッセージングシステムの開発, 情報処理学会 DPS 研究会資料 (1998).
- 7) Crocker, D.: Standard for the Format of

ARPA Internet Text Messages, Request for Comments: 822 (1982).

- 8) Coulouris, G., et al.: *Distributed Systems Concepts and Design*, 2nd edition, Addison-Wesley (1994).
- 9) Gray, J., et al.: *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann Publishers (1993).
- 10) Kung, H.T. and Robinson, J.T.: Optimistic Methods for Concurrency Control, *ACM Trans. Database Syst.*, Vol.6, No.2, pp.213–226 (1981).
- 11) Gray, J.: Notes on Database Operating Systems, Lecture Notes in Computer Science, Vol.60, pp.394–481, Springer-Verlag (1978).
- 12) Bernstein, N.: *Concurrency Control and Recovery in Database Systems*, Addison-Wesley (1987).

(平成 12 年 12 月 1 日受付)

(平成 13 年 9 月 12 日採録)

推薦文

本論文では、モバイル端末からも、オフィスのオンライン端末からも、メッセージストアにアクセスできる環境において、モバイル端末からメッセージストアと分断されたときのキャッシュのメッセージフォルダへの操作とオンライン端末からの操作との整合性を保つための同期手法を提案し、IMAP4をベースとした実装を行っている。モバイル端末の利用者が著しく増える現在、有用で、実用性の高い研究である。

(DPS 研究会主査 宮部博史)



齋藤 正史 (正会員)

1983 年東京工業大学工学部情報工学科卒業。同年三菱電機(株)入社。1992 年コーネル大学コンピュータサイエンス学科修士課程修了。オペレーティングシステム、インターネット、分散システム、情報家電ネットワークの研究開発に従事。共訳書「分散システム—コンセプトとデザイン」。



吉田 玲子

1991 年早稲田大学理工学部工業経営学科卒業。同年三菱電機(株)入社。インターネット、情報家電ネットワークの研究開発に従事。