

電子メールのオブジェクト指向データベースによる 管理方法の提案

山川利治, 岩沢京子, 萩原洋一, 中森真理雄

東京農工大学工学部電子情報工学科情報工学大講座

電子メールをオブジェクト指向データベースを利用して管理する方法について提案する。電子メールシステムに対してオブジェクト指向データベースを使用することにより、今後さらに利用が進められるマルチメディア電子メールに対して有効なデータモデルと、管理機能を提供することができると考えられる。本稿では、メッセージを共有し重複したデータをできるだけ持たない電子メールシステムの構築を目標に、メッセージ本体を重複せずに管理する方法、Replyによる関係づけ、メッセージのモデル化、メッセージ管理のためのインスタンスフォルダによるインスタンス管理について述べる。

Management of Electronic Mail by an Object-Oriented Database

Toshiharu Yamakawa, Kyoko Iwasawa, Yoichi Hagiwara, Mario Nakamori

Department of Computer Science, Tokyo University of Agriculture and Technology

A method of managing electronic mail by an object-oriented database is proposed. Object-oriented databases offer an effective data model and a function of management for multimedia electronic mails. The aim of our research is to build an electronic mail system sharing messages without duplicated data. In the present paper a method of managing message body without duplication, relating messages by reply, modeling of messages, and managing instances by instance folder for message management are proposed.

1. まえがき

今日の電子メールシステムは管理検索機能が貧弱であることは多くの人々の指摘するところである。データベースシステムはデータの管理と検索にきわめて強力な機能を有しており、これを電子メールに応用するのは自然である。近年電子メールで、画像や音声などが扱われるようになってきたが、今後はさらにマルチメディアデータが扱われるようになると予想される。そのような大容量データを含む電子メールに対して有効なデータモデルと管理機能を提供できるシステムとしてオブジェクト指向データベースが適していると考えられる。

本稿では、オブジェクト指向データベースを利用した電子メールシステムを構築するための一方式を提案する。本稿の目的は、

- ①重複するデータをできるだけ持たない電子メールシステムの提案
- ②メッセージ管理向けのインスタンス管理方式の提案

である。

これらの目的を実現するために、本稿では次のことについて述べる。

- (1) メッセージ本体を重複せずに管理する方式
- (2) Reply による関係づけ
- (3) メッセージのモデル化
- (4) インスタンスフォルダによるメッセージの管理

2. 提案する電子メール管理システムの構想

電子メール管理システムを提案する上で現在の電子メールと WWW を参考にした。現在の電子メールは、ネットワークで接続されていれば遠くのユーザとのメッセージのやりとりが簡単に行える。しかし、同じメッセージを送信側と受信側で別々に管理するので、データが重複して管理されていると考えることも

できる。また WWW では、ページ間でリンクを張ることにより、通信に時間はかかるが、自分のところにデータがあるのと同じように扱うことができる。この二つのシステムを参考に、本論文の電子メール管理システムを考えた。

図1に、提案する電子メール管理システムの概要を示す。図1では、ユーザAが作成したメッセージをユーザBに送った場合のメッセージの扱い方を示してある。ユーザAが作成したメッセージは、ユーザAのデータベースに登録されている。ユーザBに対しては、メッセージにリンクする情報だけを送る。

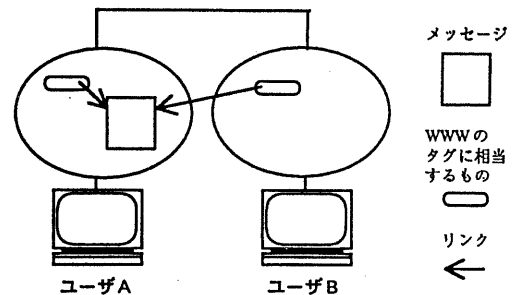


図1 電子メール管理システム

3. メッセージ本体を重複せずに管理する方式

本章では、電子メール環境の中でメッセージ本体（文章や図形、音声などからなる実際に相手に伝えたい事柄）を重複せずに管理する方法を述べる。

3. 1 現行のメッセージ管理方式の問題点

今日広く行われている文章中心のメッセージでもメッセージの数が多くなると記憶領域の大きさが問題になる。今後、画像や音声などを扱うようになると記憶領域の大きさがさらに大きな問題となる。メッセージの管理方法としては、MH などのメッセージを処理するプログラムを使用してハードディスクで管理

する方法などがあるが、現在どのような管理方法を行っていたとしても、ユーザとしてはできることならメッセージを MH やデータベースなどで管理・保存して、利用できればよいと考えている。このときに問題となるのは、ハードディスクなどの記憶装置が無限ではないので、できるだけ記憶しておくものを少なくしなければならないことである。特に画像や音声データなどを扱うとそれだけで大量の記憶領域を消費してしまう。そこで、記憶しておくものを減らす一つの考えとして、メッセージ本体を重複せずに管理する方法を提案する。メッセージ本体を一つにしても十分であると考える理由は次のとおりである。

何人かのユーザ間でやりとりしたメッセージでも、保存しておくメッセージはみんな同じものであるということである。保存した後の利用の方法はいろいろあるが、一度出したメッセージの内容を変えるということはない。内容を訂正したいときは基本的には新しいメッセージを改訂版とするであろう。

次にメッセージ本体を重複しない方式の利点と欠点を示す。

利点

- ①メッセージを複数持つよりも記憶領域を削減できる。
- ②メッセージの一貫性を保つことができる。
- ③他人のメッセージに改ざんを加えたニセ情報を流しにくくなる（コピーすればできるが、コピーであることがわかってしまう）。

欠点

- ①通信回数と通信量が増加し、通信時間がかかる。

現在のネットワーク環境では、メッセージを一つにしたときの一番の問題点が通信時間で、通信時間を無視してシステムを考えることはできないが、本稿では通信がネックとなっていない前提でのモデルを考えた。このモデルによる通信に関する評価については、本論文の最後に述べる。

3. 2 メッセージを重複しない方法

まず、本論文で提案する電子メール管理システムで、扱われるメッセージについて定義する。

メッセージの定義

メッセージは、相手に伝えたい情報の部分であるメッセージ本体と、メッセージ本体に関する情報であるメッセージヘッダから構成される。メッセージヘッダはさらに、メッセージ本体に関する文字情報である基本情報と、メッセージ本体へのリンクになる鍵から構成される。すなわち

$$\begin{aligned} \text{メッセージ} &= \\ &\text{メッセージヘッダ} + \text{メッセージ本体} \end{aligned}$$

$$\text{メッセージヘッダ} = \text{基本情報} + \text{鍵}$$

となる。

図2に示すようにユーザがメッセージを作成したときには、メッセージヘッダとメッセージ本体からなるメッセージが作成されるが、これらは作成したユーザのデータベースに登録される。通信相手には、メッセージヘッダだけを送る。

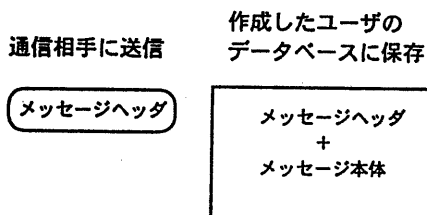


図2 メッセージの関係

3. 3 「鍵」

鍵は、メッセージヘッダを受け取ったユーザがメッセージ本体にアクセスするために必

要な情報である。鍵は、メッセージを作成するときシステムが、メッセージ本体のオブジェクト識別子とメッセージ本体のあるシステムのアドレスを自動的に作成する。鍵の属性には、作成したユーザも、メッセージヘッダを受け取ったユーザも直接アクセスすることはできない。この鍵を持つことによりメッセージヘッダとメッセージ本体の間でリンクが張られたことになる(図3)。

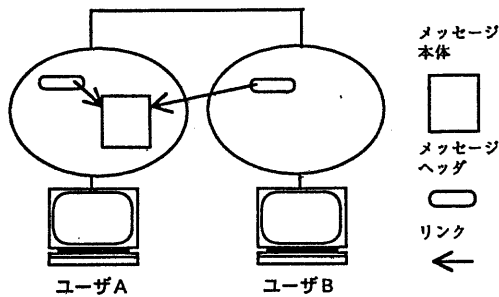


図3 メッセージのリンクの様子

3. 4 ルール

メッセージ本体を重複しないので、受信側は受け取ったメッセージの内容を変更して利用することができない。その場合は受信側はコピーして利用することになる。コピーのルールは次のとおりである。

ルール1 (コピー)

コピーは、送信側がメッセージ本体に設定したレベルによって、可能か否かが決定する。コピーしたメッセージは、受信側のデータベースに新しいデータとして登録されて、送信元のメッセージとの関係はなくなる。Replyの関係づけがあるものでも、その関係はすべて解消した独立なものとして扱われる。

メッセージは、送信側で消去すると受信側ではそれ以降は参照できなくなる。送信側が予告なしに消さないように、次のルールを設ける。

ルール2 (消去)

初めにメッセージをいつ消去するかを設定するか、鍵を所有しているユーザに通知を出してから消去する。希望によっては、コピーしてもらおう。回答を待つ場合も、期間(期限)を設定する。

4. Reply による関係づけ

本章では、Reply したメッセージの内容を、新しいメッセージにコピーするのではなく関係づけを行い、情報の重複を避けてお互いが利用できる仕組みを考える。

4. 1 Reply の現状と改良点

メッセージのやりとりをしていて、Reply することはよくある。Reply したときに、もとのメッセージの内容を引用して自分の意見を書くことは一般的によく行われている。このような引用は現在のメッセージが文字中心であるから行われている面があると考えられる。Reply するメッセージの内容を分かっている人に Reply する場合ならば、文章ならば一部を引用すれば分かることもある。しかし、そのような場合でも、前のメッセージを見ることができるようになればより便利である。さらに、画像や音声メッセージが扱うようになると引用の仕方も難しくなってくる。画像や音声の場合はやはり前のメッセージを参照できたほうがよい。またメッセージを使用して質問と答を繰り返して Reply しているとそれまでの答の流れや最初のメッセージが分かるような仕掛けがあれば便利である。

このような事情を考えて、Reply したメッセージを関係づける方法を電子メール管理システムでは提案する。電子メール管理システムでは、個人レベルのメッセージを関係づけて利用できるような方法を提案する。

4. 2 Reply の方法

(1) 関係づける方法

メッセージの関係づけは、Reply をしたときに行われる。Reply したメッセージはもと

のメッセージと関係づけられる(図4)。メッセージの Reply に関する情報は、Reply されたときにシステムによって設定される。

Reply は、一つのメッセージに対してだけ行うことができる。逆に、一つのメッセージに対していくつものメッセージを Reply することができる。

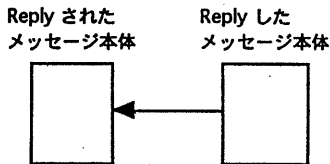


図4 Reply による関係づけ

(2) アクセスの方法

一連のメッセージへのアクセスを行うには、最低どれか一つのメッセージへアクセスするための鍵を持っていなければならない。一つのメッセージにアクセスすることによって、それ以降の、一連のすべてのメッセージにアクセスすることが可能になる。

5. メッセージのモデル化

メッセージをオブジェクト指向データベースで管理するために3で定義したメッセージをオブジェクト指向モデルで表現する。

5.1 メッセージ本体のクラス表現

メッセージ本体をクラスとして表現するのに、次の二つのクラスが考えられる。

- (1) Subject に注目したクラス
(例 連絡クラス)
- (2) 媒体に注目したクラス
(例 文章クラス)

本論文の方式では、上記のうち方法(2)でクラス表現をすることにした。理由は、Subject は検索のときに利用できるように属

性の一つとして持っていれば十分であることと、方法(1)より方法(2)の方がクラス定義の数が少ないことによる。

5.2 メッセージの基本情報

メッセージをやりとりするのに、メッセージ本体の他に付随するいくつかのデータが必要である。少なくとも次のものは必要である。メッセージ本体に関する Subject は、ここに含まれる。

ユーザが書くもの

- (1) Subject (メッセージの名前)
- (2) address (送信先のアドレス)
- (3) keywords (内容に関するキーワード)

システムが自動的に書くもの

- (4) 送られてきた日時
- (5) 作成した日時
- (6) 作成ユーザ

5.3 メッセージのクラス表現

図5にメッセージを構成するクラスを示した。メッセージヘッダクラスは、基本情報と鍵を持つクラスと定義する。鍵はユーザには公開しないクラスであり、ユーザは鍵の属性にアクセスすることはできない。本来オブジェクト識別子は、オブジェクト指向データベースがつけるもので、ユーザがその値を使うことはできない。しかし、本論文の方式でのクラス表現の場合、鍵はメッセージヘッダとメッセージ本体の間のリンクを表現するためのもので、その鍵にはユーザが直接アクセスできないようにすることで、ユーザには存在していないクラスと考えることもできる。鍵の属性値はメッセージを作成したときに電子メール管理システムが自動的に設定する。

鍵の属性であるオブジェクト識別子は、オブジェクト指向データベースに登録されているメッセージ本体になり得るクラスのインスタンスのものならばどのクラスでも良い。

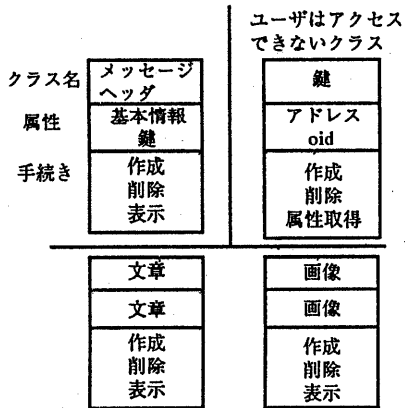


図5 メッセージのクラス定義

6. インスタンス管理方法

本章では、電子メール管理システムで使用するためのオブジェクト指向データベースのインスタンス管理方法について考える。初めに、現在のインスタンスの管理方法を説明して、次に筆者の提案するインスタンスフォルダを使用した管理方法を述べる。インスタンスフォルダを電子メール管理システムではどのように利用しているかを最後に述べる。

6.1 通常のインスタンス管理方法

通常のオブジェクト指向データベースでは、インスタンスはクラスと関係づけられて管理している。この管理方法では、インスタンスの検索は、クラスを指定しそのクラスのインスタンスすべてを検索するという手順で行う。

図6に示したのは、メッセージヘッダクラスのインスタンスの例であるが、五つあるインスタンスのうちユーザAが所有するメッセージが1と4でユーザBが所有するメッセージが2と3と5のとき、この管理方法ではお互いのメッセージにアクセスできてしまう。保護するためにはインスタンスごとにセキュリティを管理しなければいけない。

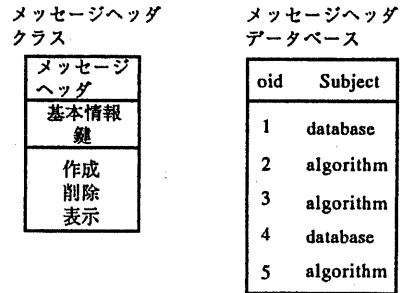


図6 通常のインスタンス管理方法

6.2 インスタンスフォルダを使用したインスタンス管理方法

本論文で提案するインスタンス管理方法は、所有者ごとのセキュリティが確保されしかもクラスに関係なくインスタンスを管理する方法である。本論文では、そのようなことを行えるものを提案しインスタンスフォルダと呼ぶことにする。検索は、検索したいインスタンスフォルダを指定することによって行う。

インスタンスフォルダの定義

インスタンスフォルダは、所属するクラスに関係なくインスタンスを管理するためのデータベースである。インスタンスフォルダには、アクセスできるユーザとインスタンスフォルダに対する手続きを定義することができる。

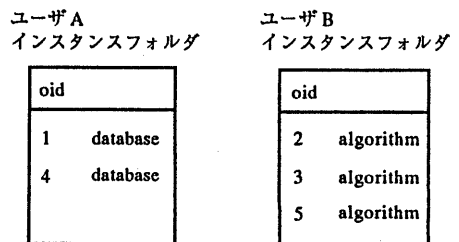


図7 インスタンスフォルダを使用したインスタンス管理

6. 3 電子メール管理システムでのインスタンスフォルダ

6. 2で提案したインスタンスフォルダを電子メール管理システムでは、メールボックスとメールフォルダと定義して使用しているが、どのように定義して使用しているかを述べる。

6. 3. 1 メールボックス

提案する電子メール管理システムでは、メッセージを送信すると、受信側ではメッセージヘッダを受け取る。メールボックスは、受信したメッセージヘッダの管理を行うインスタンスフォルダである。図9に示すように、電子メール管理システムの通信機能を使用してメッセージヘッダが送られて来たら、受け取った電子メール管理システムは、それをアドレスのユーザのメールボックスに分配する。メールボックスは、ユーザ登録が行われたときにオブジェクト指向データベース上に作成されている。メールボックスにアクセスできるユーザは、決まったユーザだけである。

メールボックスに入っているメッセージヘッダにユーザがアクセスすると、システムは自動的にメッセージ本体にアクセスして表示する。一度表示されたメッセージのメッセージヘッダは、メールフォルダに移して保存するか、消去するかどちらかしなければならない。メールボックスの手続きとして、表示されたものはメールフォルダに移動する手続きが定義してあるので、消去する場合だけユーザは指定する。

6. 3. 2 メールフォルダ

ユーザが作成したメッセージ（メッセージヘッダとメッセージ本体）と、残しておきたいメッセージヘッダを保存するために、メールフォルダがある。メールフォルダもインスタンスフォルダのひとつで、ユーザが登録されたときにメールボックスと同時に作成される。アクセスできるユーザはメールボックス

と同じになる。

メールフォルダのなかのメッセージヘッダの並び方は、最近使用したもののほど先頭に近いところにくるようになっている。

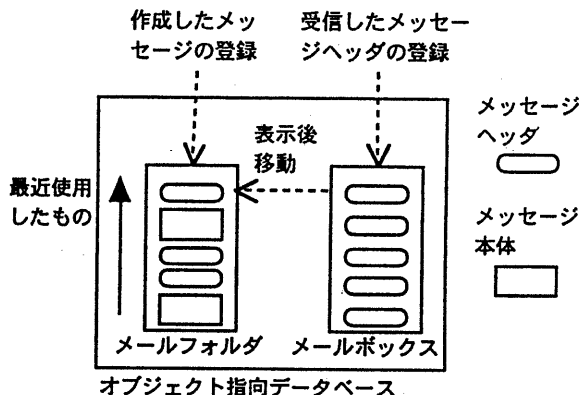


図9 メールボックスとメールフォルダ

7. 評価

(1) メッセージ本体を重複しないで管理する方法の提案を行った。現在あるシステムと提案したシステムでメッセージを作成して送信した場合の比較を表1に示す。

表1 現在ある電子メールとの比較

	消費容量	通信回数
現在	NX	N
提案	X	$N + 2A$

N : メッセージを送信したユーザの数

X : 送信したメッセージの消費容量

A : メッセージ本体がアクセスされた回数

表1のように提案したシステムでは、大容量でたくさんのユーザにメッセージを送って

も消費する記憶容量は変わらないで管理することができる。ただし、表1に示したように提案したシステムでは、ユーザがアクセスするごとに通信が起こるので、頻繁にアクセスされるものは、通信回数が多くなりネックとなる可能性がある。

(2) Reply したときのデータの流れを知ることができるようになった。しかし、十分な安全性の実現はできなかった。

安全性を実現していくために今後は、メーリングリストを使用したユーザ管理機構を導入することを考えている。

(3) インスタンスフォルダを提案することでメッセージ向けの管理方法ができるようになった。

(4) 通信量に関する評価

表1に示したように、ユーザがメッセージにアクセスする度ごとに二回の通信が起こる。一回は、メッセージ本体の要求に関するものなので、通信量は少ない。もう一回は、メッセージ本体を通信するので、メッセージ本体の大きさによる。例として、画像(白黒の1152×900のビットマップ)メッセージで考えると、大きさは約665Kbyteであった。これをイーサネットを使用して通信したとする。イーサネットは、10 Mbps であるが、実行時にネックとならない最大転送容量は経験的に4割程度である。この例ではメッセージ本体に一度に6回以上の通信が起こると効率が悪くなってしまふ。これでは、頻繁にアクセスされるメッセージは本体は、使用するのに耐えられなくなる。

改善案としては、頻繁にアクセスされるものを一時的に記憶しておくためのキャッシュによる方法などが必要になる。

8. まとめ

本稿では、目標である重複したデータを持たないシステムの提案を行った。今回提案した方法では、メッセージ本体を一つのデータ

ベースで管理することにより、重複しないようにした。

Reply によりメッセージの流れを見ることができるようにした。

メッセージをオブジェクト指向モデルで表現し、オブジェクト指向データベースで管理できるようにした。

メッセージ管理向けのインスタンス管理方法としてインスタンスフォルダによる管理方法の提案を行った。

9. 今後の課題

今後の課題は、電子メール管理システムの完成と、その後の通信時間と通信回数の評価である。また今回提案した方法をさらに検討して機能を加えていく必要がある。

謝辞

本研究に貴重な助言を頂いた奈良先端科学技術大学院大学の植村俊亮教授と同大学の植村研究室の学生のみなさんに感謝します。

本研究は文部省科学研究費補助金一般研究(C) 05680267 の援助を受けた。

参考文献

- [1] 山川利治：電子メールシステムのためのオブジェクト指向データベースに関する研究。東京農工大学大学院工学研究科電子情報工学専攻、1994年度修士論文、1995
- [2] Jerry Peek：MH & xmh E-mail for Users & Programmers, O'reilly and Associates Inc., 1992
(訳書、砂原秀樹/鈴木麗 監訳、倉骨彰 訳：MH & xmh, アスキー出版局、1994)
- [3] 特集 オブジェクト指向データベースシステム、情報処理、vol. 32, No. 5, 1991