チーム作業支援環境

7J－1

中野 智加良　市村 哲　稲村 浩　岡田 謙一　松下 温
(慶応義塾大学)

1 はじめに

我々の組織における実際の作業は、複数人による共同作業という形態をとることが多い。それにもかかわらず、実際にそのような共同作業を支援する環境はあまり無く、個人の生産性向上にのみ重点がおかれているのが現状である。[1]

そこで、本研究では、チームという作業形態を考え、共同作業集団の中の個人用データ・共用データの融合を中心に、そのチームの作業を支援する環境提供を目的とする。本稿では、パーミッションに基づくレイヤーという構造を用いることで、この個人用データと共用データの融合を実現し、チーム作業を支援するデータベース管理システムを構築したことを報告する。

2 チームによる作業環境

チームという作業形態には、意志決定や会議という形態は含めないことにする。その形態の中でのデータの組織化という面から見ると、この意志決定や会議という作業では、すべてのデータがいつでも皆に見え、途中での変更、生成、削除も見えている。すなわち、途中経過をその集団の中に逐次取り込んでいくのである。例えば、会議でのデータの組織化はFig.1のように示される。出された意見は、出されたそのときに皆に伝わり、すでに出されていた意見に密接に関係しながら会議という集団の中に取り込まれていく。ところが、チームとは、意志決定はすでになされており、その決定により開発という目的がはっきりしているエキスパートの集まりであると考えられる。そこでの各自の独立性は高く、途中経過は各自のみが必要で、その各自の作業の結果のみが共有される。これは、Fig.2に示されるような共有部分を持ったレイヤー構造で示される。また、共有された結果に対して他人による変更が成されれば、それがフィードバックして戻ってきて、作業の完成度をさらに高めると考えられる。

このような、例えばプログラム開発、マニュアル、カタログなどのドキュメントの共同作成、そして、CADといった、開発という目的を中心に集まった、すでに意志決定がなされた集団を支援するシステムをチームウェアと呼ぶことにする。

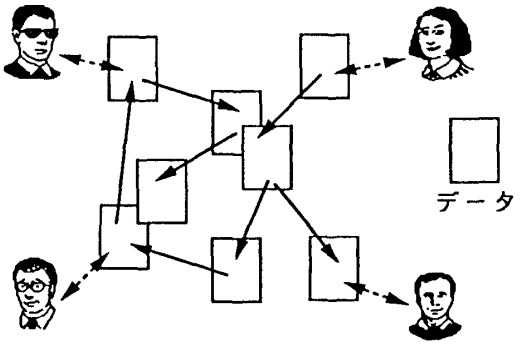

Fig. 1 会議の中のデータの組織化

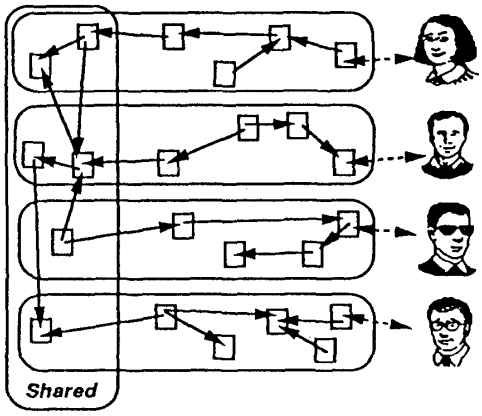

Fig. 2 チームの中でのデータの組織化

3 システムの構成要素とその特徴

我々は、このチームウェアを実現するために次に示すような特徴を持つシステムを構築した。

3．1 レイヤー構造

このシステムは、少数精鋭のチームでの共同作業支援を目標の一つとしている。従来のデータベースは共有して利用するデータ（共用データ）に注目したものが多かった[2]。しかし、チームでデータベースを利用する場合、全メンバーで共有するデータと、個人で持つべきデータという区分が生じるので、同一のシステ

Team Support Environment
Chikara Nakano, Satoshi Ichimura, Hiroshi Inamura, Ken-ichi Okada, Yutaka Matsushita
Keio University

ムで個人のデータも管理できることが望ましいと考えられる。そこで、全員で共有する一般的なクラスやデータは共用データレイヤーに、また、個人がそれぞれプライベートに定義したクラスやデータは個人用データレイヤーにと、区別して保存する（Fig.3）。共用データは修正不能で読みだしのみとする。これに対し、個人用データは自由に読み書き可能で、共用データからのクラス属性・メソッドの継承も可能である。このようなレイヤー構造を用いることにより、同一のシステムでの個人データの管理が容易となる。チームウェアにこのようなレイヤーの概念が有効であることは、チームというものがFig.2のような構造を持っていると言うことから明かであると言える。

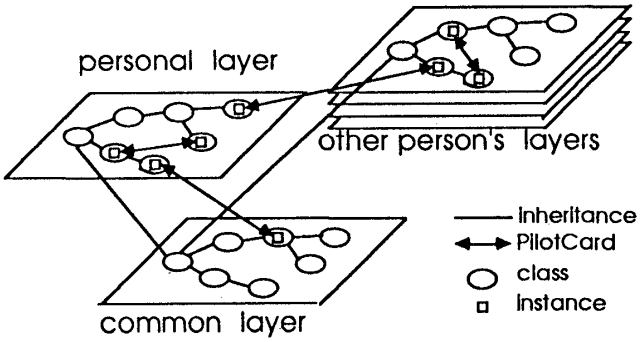

Fig.3 データベース中のレイヤー構造

3．2 個人用レイヤー間でのデータ共有

データの共同利用というよりも、その共同作成、例えば、ドキュメントの共同作成というような場合には、個人用データレイヤーに他のユーザの個人用データを持つことが必要である。ここで、ユーザ毎のレイヤーを超えてポインタを渡す機能が要求されるが、これはユーザ間のコミュニケーション機能（PilotCard[3]）により可能となる。

PilotCardとは、双方向のポインタというリンクを持ったメモである。まず、共有されるデータを指すポインタを持ったPilotCardを共有先のユーザに送り、それを中継地点として共有するのである。これにより、各個人用レイヤー間を透過したデータ参照が可能になる。

また、本システムは、チームという共同作業を支援するものではあるが、Fig.2からも分かるように、そのためにはまず個人環境が充実している必要がある。そこで、このPilotCardを用いて、データに対してユーザにカスタマイザビリティを与える。PilotCardを用いてクラス定義に無い属性を新たに表現するのである。

3．3 インスタンス属性

システムは、データベース中の一貫性や正当性を検出するために、インスタンス属性として作成日時、更新日時、バージョンを管理する。さらに、そのインスタンスを見ることが可能であるのは誰か、また、共有の度合はどの程度であるかを表現するために、<可視制御>というインスタンス属性を管理する。これは、インスタンスのパーミッションといえる。

また、共有の度合は、次の四つの状態に分けられる。

a)他のユーザからは見えず、個人のみの所有。

そのユーザ自身のレイヤーにあるプライベートなデータである。

b)他のユーザは参照のみ可能。

インスタンス変数の変更を起こすようなメソッドの起動は禁止される。

c)他のユーザは、共有したいインスタンスはそのままに、それをバージョンアップしたものを自分自身のレイヤーに生成させ、それに対して読み書き可能。

この時、元のインスタンスからバージョンアップしたインスタンスへ、バージョンポインタがリンクされる。

d)他のユーザも読み書き可能。

そのインスタンスのポインタを渡されたユーザは、そのインスタンスへの自由な変更を許される。

4 おわりに

本稿では、チーム作業支援のためのレイヤー構造を持ったデータベース管理システムの概要を述べた。このシステムにより、チーム作業における個人用データと共用データの融合が実現できた。なお、本システムは、現在、Sun Workstation上で構築途上中である。

参考文献

[1]石井、大久保、”コンピュータを用いた人間の共同作業支援技術について”、情報処理学会「マルチメディア情報と分散協調」シンポジウム論文集、pp.27-36、平成元年11月.

[2]Ullman, J.D., Principles of Database Systems (2nd edition),Computer Science Press,1988.

[3]三上、松浦、稲村、岡田、松下、「連想アクセスとデータ共有のためのPilotCard」、本大会.