

個人/協同作業フェーズの連続性を重視したデータ管理*

5 Q-2

紙田 剛 市村 哲 八並 純子 岡田 謙一 松下 温†
慶應義塾大学 理工学部‡

1 はじめに

グループ作業を支援するデータベースには、共有データを利用できるだけでなく、各自が独立して作業するための環境が整備されている必要がある。著者らは以前の研究において、データベース管理にレイヤ構造を取り入れることにより、個人・協同の作業空間の統合を実現した。

今回、この空間的透過性に加えて「時間的透過性」を実現するデータベースシステムを試作した。作業者が何か書類なり、企画書なりを作成する場合、一度で仕上がることはまれであり、多くの場合過去の作業内容を再考、あるいは再利用することによって、何度も修正を重ね仕上げていく。従って、過去の作業の一連の流れが十分に把握でき、個人作業の結果を協同作業空間に移出した後も、個人作業の時と変わらない方法でアクセスできれば、作業能率の向上につながる。これをもって「時間的透過性」とする。以下でこれを実現する仕組みについて述べる。

2 個人/協同作業空間の共存

個人用データベースと共用データベースを統合したオブジェクト指向データベースは、数は少ないものの、ORION、VERSANTなどが存在する。これらのシステムは共通に、「チェックイン・チェックアウトモデル」を採用している。しかし、今まで人が行ってきた実際の協同作業の進めかたに着目すると、チェックイン・チェックアウトモデルとはまったく違った方法で資源が利用されてきたことがわかる。

1. 各作業者は、協同作業間で共有する資源(仕様書、作業計画書、マニュアル類等)の他に、常に愛用している資源(参考資料、自筆メモ、文書ファイル等)を独自に多数所有しており、作業を進める際には頻繁にこれらにアクセスする。よって、個人の作業空間は、このような私的資源を長期間保持する能力を持たなくてはならない。しかしながらチェックイン・チェックアウトモデルでは、個人用データベースはあくまでも一時的なワークスペースにすぎず、チェッ

クインを済ませると同時に個人作業空間上の情報は消滅してしまう。

2. 各作業者が新規に作成したデータを共有の場に公開しようとするのは、ある程度完成度が高まった段階であり、それまでの作業途中データは公開したがない。たとえチェックイン・チェックアウト機構を利用する際にでも、個人で管理しているファイルを用いて(データベースを用いずに)データ更新作業を行い、ある程度まとまった時点ではじめて、個人データベース上にチェックアウトしてきたデータに書き写すという手順がとられることが多い。このために、作業データ(作業ファイル)と公開データ(データベースに蓄積されるデータ)の間の関連性が失われてしまい、また、個人作業時の作業履歴が実質上データベース中に残らないという問題が生じる。

3 空間的透過性とレイヤ構造

グループで作業を行うことの長所の一つに、情報を共有化できるということがあげられる。グループ作業時には、個人で所有される情報と、グループで共有される情報を区別し、さらに、各作業者はグループ資源をあたかも自分のもののように扱えることが望ましい。これを実現するのがレイヤ構造である。このレイヤ構造では、各個人の「個人レイヤ」と、複数の人に共有されるデータを管理する「共有レイヤ」が存在する。各レイヤは物理的に別々のコンピュータサイトに分散しているが、システム全体で論理的に層状に重なり合う構成をとっている。

レイヤ構造の重要な特徴として、レイヤの透過性がある。これは、グループ全員で共有するような大域的なデータを共有データに置き、共用レイヤ上にあるクラスが有する資源を個人用レイヤ上のクラスが継承するのである。これにより、各ユーザは共用レイヤ上のデータを読むだけでなく、クラス属性やメソッドを個人用レイヤ上のクラスに継承することができる。

4 時間的透過性とバージョン管理

バージョン管理の視点から従来の協同作業を捉えたと、個人作業空間と協同作業空間では、バージョンを管理することの意義が異なることに気付く。また、個人作業フェー

*Data management supporting seamlessness between personal and group work phase

†Takeshi Kamita, Satoshi Ichimura, Jyunko Yatsunami, Kenichi Okada, Yutaka Matsushita

‡Faculty of Science and Technology, Keio University

ズと協同作業フェーズを貫いてバージョンを長期的・連続的に管理する必要がある。これらについて、以下の例を挙げて議論する。

1. 個人作業空間においては、他人には見せる必要のない自分自信のためだけのバージョンを保管したい場合がある。きれいに清書しなおした原稿よりも、自筆の走り書きが書き込まれた草稿を参照した方が、作業内容や作業過程を思いだしやすく作業の点検もしやすいことが多い。よって、個人作業空間には、自分専用のバージョンを保持する機能が提供されているべきである。
2. 協同作業空間に公開されているデータは、複数の協同作業によってバージョンがアップされてゆくが、個人的な覚え書きやアイデアの走り書きなどが付けられることはない。しかし、他人の原稿をバージョンアップするときでも、覚え書きをメモした自分独自のバージョンを保存しておきたい場合があり、この個人専用のバージョンを個人作業空間上に残せるための機構が必要となる。
3. 協同作業空間にデータを公開した後でも、そのデータに対して責任を負うのは最終的にそのデータの作成者本人である場合がほとんどである。この時、作成者は、個人作業空間内に残されていた自分専用のバージョンを参照しながら、協同作業空間内のデータをバージョンアップする。したがって、同一データに対してなされた、個人作業空間のバージョンとデータ作成者本人の個人専用バージョンを、容易に関連付け・比較できる機構が求められる。

以上の考察に基づけば、個人と協同の作業フェーズを貫いて、常に変わらない作業環境の中でデータにアクセスできるためには、個人用レイヤと共用レイヤを貫いたバージョン管理を行う必要がある。実装したデータベースは以下のような機能を持つ(図1参照)。

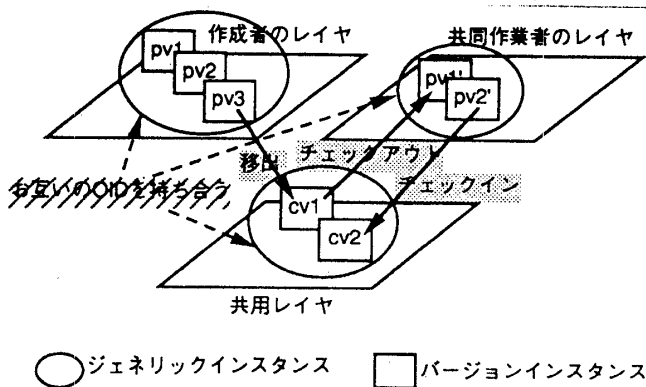


図1: レイヤ構造に基づくバージョン管理

- 個人用レイヤ上で行ったバージョンアップは共用レイヤ上に公開されず、データの作成者以外のグループのメンバは共用レイヤ上にあるバージョンのみ見ることができる。
- データの作成者は、両レイヤ上のバージョンアップを統合した履歴を見ることができる。
- データの作成者が自分のレイヤ上のデータを共用レイヤに移出した場合でも、個人レイヤ上の作業途中データと共用レイヤ上のデータは同一オブジェクトとして扱われ、バージョンのみが違っていると認識される。
- データの作成者が共用レイヤ上のデータを自分のレイヤに戻した場合、もとの作業途中データと同一オブジェクトとして扱われ、バージョンのみが違っていると認識される。

以上の機能は、個人用レイヤ上のデータが共用レイヤ上に移った時でも、別のオブジェクトとして扱うのではなく、バージョンのみが更新されたとみなす機構によって実現されている。以上のようなバージョン管理を行なうことで、時間的透過性を得ることができると考える。

5 協同執筆作業の支援

試作したオブジェクト指向データベース上に、ドキュメントの協同生成を目的としたアプリケーションを実装した。マニュアルの作成においては、一つのドキュメントを分担して執筆することが多く、協同執筆者との調整や、関連部門からの情報を適切に収集する必要があり、一つのグループとして作業方法や支援環境が必要と考えられる。(図2参照)

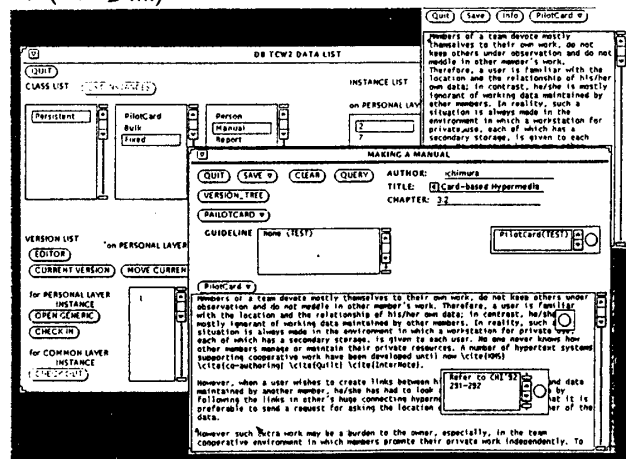


図2: マニュアル作成画面

参考文献

- [1] Won Kim, Elisa Bertino, Jorge F. Garaza, 'Composite Objects Revisited' ACM SIGMOD '89, Vol.18, No.2, June 1989, pp.337-347