

分散開発環境における同時更新制御方式

4N-6

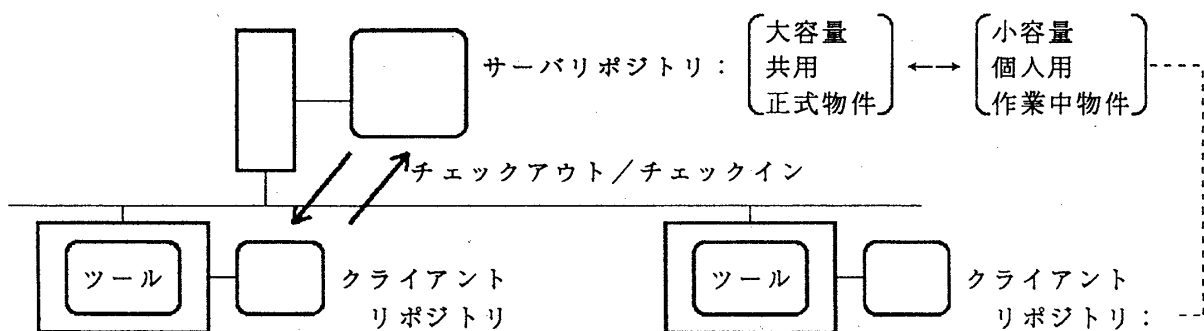
飯塚 まとひ 関 武夫 伊藤 久直
 (株) 東芝 青梅工場

1. はじめに

分散開発環境で、設計情報の更新を、チェックイン/チェックアウト方式により管理する場合、同じデータを複数人がチェックアウトして同時に更新すると、チェックイン時に同じデータに対する複数の版が出来るため、そのデータを使った設計作業を継続するためには、データの統合が必要になる。特に、複合データをひと固まりの設計情報として扱う場合、複合データの中の一部の共有情報を他人と同時に更新してしまうようなケースは、簡単に起り得るが、単一の共有情報を指定して更新する訳では無いので、利用者が意識的に制御するのが難しく、統合化作業も複雑になる。ここでは、このような同時更新を制御するメカニズムについて述べる。

2. チェックイン/チェックアウト方式による共同作業形態

複数人間が共同して一つのシステムを開発するための環境としてクライアント/サーバ環境を考える。まず、設計情報を共有するためにサーバ上に共有リポジトリ（設計情報の入ったデータベース）を置く。クライアント上のリポジトリには利用者が必要な設計情報だけをサーバから持ってきて（チェックアウト）、設計ツールを使って編集等が終った後、サーバに書き戻す（チェックイン）。設計ツールで扱う情報が単一の設計情報から成るような単純な情報では、チェックアウトからチェックインまでの期間、サーバ上の設計情報をロックしておくことにより、同時に複数人が同じ設計情報を更新する問題を回避する事ができる。例えば、各利用者がテキストファイルを排他的に編集するだけで良い場合は、このような単純な制御で十分である。



Concurrent Update Control in Distributed Development Environment

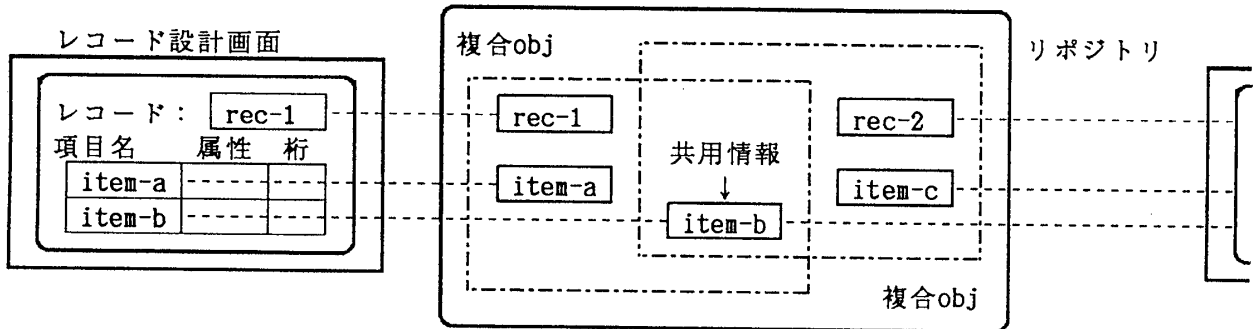
Matoi Iizuka, Takeo Seki, Hisanao Ito

Toshiba Corporation Ome Works

2-9 Suehiro-cho, Ome, Tokyo 198, Japan

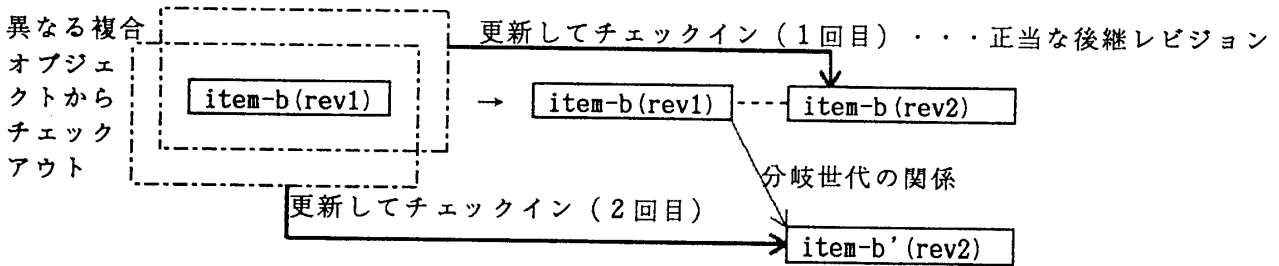
3. 複合オブジェクトを扱う場合の問題点

CASEにより木目細かい設計情報を管理する場合、設計ツールで操作を行う単位は、単一の設計情報からなる単純な情報よりも、複数の設計情報を一つとして扱う複合オブジェクトである場合が多い。例えば、レコード内のデータ項目の構成を定義する場合、レコード自体の定義と其中的構成要素であるデータ項目をまとめて一つの複合オブジェクトとして扱うのが自然である。この場合、複合オブジェクト間で共用される情報があるので、すべての情報にロックをかける方式ではロック待ちが多発するし、ロック無しで上書きするのでは更新内容が無条件に破壊される問題が起きる。



4. 同時更新制御方式

複合オブジェクト間で共用情報を持つ場合の共用制御メカニズムとして、(1)複合オブジェクト単位ではロックを掛けて排他書込みを行い、(2)複合オブジェクトの要素にはロックを掛けずに同時更新は更新世代を管理することにより検出し同時更新の結果は分岐世代として保存、と言う方式で実現する。このような同時更新の制御を複合オブジェクトのチェックイン操作に組込むことにより、利用者からは複合オブジェクト単位の操作だけを行えば、個々の要素データの同時更新制御を一括して行うことができる。同時更新の結果は分岐世代の関係によって表現されるので、この分岐世代の統合もこの関係を検索することにより分岐世代の統合機構を駆動することができる。



5. おわりに

テキストファイルなどの単純なデータの同時更新を分岐世代として表し、後で統合することは、ごく普通の世代管理機構で実現されているが、CASEでもっと粒度の細かいデータを扱う場合、複合オブジェクトの同時更新の問題は避けて通れない問題となる。複合オブジェクトの中の個々の要素データについて人手で世代管理を行うのは現実的ではないので、ここに述べたようなメカニズムが複合オブジェクトを扱う場合の基本的な機能として組込まれる事は、分散開発を支援するためのCASEリポジトリにとっては殆ど必須の機能と思われる。