

4S-12

統合プログラミング環境 (5)

ソフトウェア情報用データベースPDDの階層化の試み

濱崎 勝久 山田 晃智 小林 茂
(株) 東芝

1 はじめに

ソフトウェア設計情報用データベースPDD (Program Data Dictionary) [1][2][3]は、ソフトウェア開発過程で作成されるソフトウェア・オブジェクト(以下、単にオブジェクトと呼ぶ)やコマンド等の情報を一元管理するためのソフトウェア開発用データベース(DB)である。

本稿では、DBの開発環境への適用度を更に高めるためにPDDに新たに追加したサブDB機能について報告する。

2 サブDB開発の目的

ソフトウェア開発支援用DBとして求められる条件として以下のものが挙げられる。

- ・プログラムからのアクセスの柔軟性
- ・DBアクセスのシステムに与える負荷が小さいこと
- ・データの高速な登録・検索

PDDでは、これらの条件をDBアクセスインタフェースの形式やデータテーブル形式の工夫により実現してきた[2]。しかし、ソフトウェア開発量の増加や、開発環境の分散化に伴い、次の項目の強化を行う必要が出てきた。

- ・検索・登録速度の高速化
- ・検索範囲の指定による不要情報の隠ぺい
- ・不正なアクセスからのデータ保護
- ・開発環境の分散に伴うDBの分散化

これらの要求に対し、PDDではDB内部をサブDBと呼ぶ領域に分割することにより、検索時間の短縮、DBへの不正アクセスの防止および分散DBの実現を行った。

3 サブDBの機能

上記のことをサブDBはどのような機能として実現しているかを次に説明する。

3.1 DBの階層化

複数のメンバによるソフトウェア開発においては、図1に示すように開発全体が階層構造を持っているのが普通である。それぞれの開発担当は、より上位の開発担当に対して成果物(モジュールやデータ等)を提供することにより、最終的には開発リーダーの元に全ての必要な情報や機能が揃うことになる。この場合、各開発担当にとっては、自分が保有している情報と、より上位の担当が保有している共通情報のそれぞれが必要である。

例えば、図1のような開発プロジェクトにおいては、担当Fにとって必要な情報はサブリーダーBと開発リーダーAの情報とを合わせた部分である。

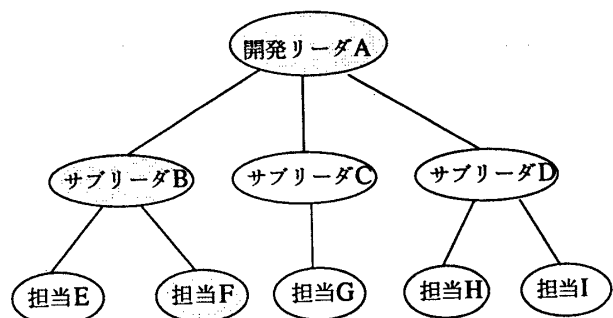


図1 プロジェクトのメンバ構成

PDDは、DBに格納される情報を階層化された領域(これらの個々の領域をサブDBという)の中にグループ化して管理する。例えば、図1の開発プロジェクトの各メンバの所有する情報を一つのサブDBに割り当てた場合、DBは図2のような構造になる。DB内の情報を取り出す時にはPDD内のサブDB情報(インデックス部)より該当するレコードを検索し取り出す。

図2のようなDBでは担当Fが検索できる情報はサブDB-B、B、Fの範囲に含まれる。(レコードの部分)

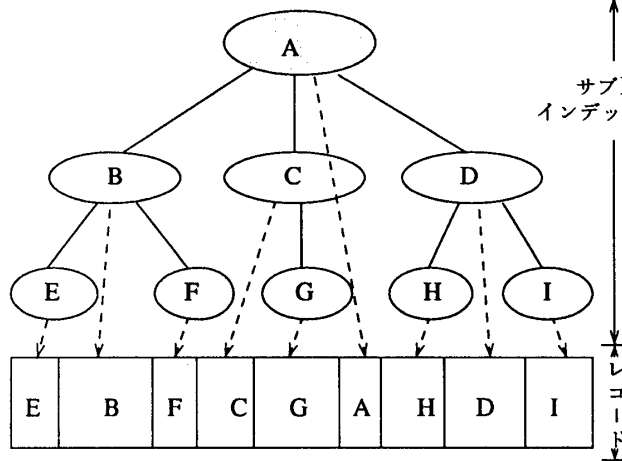


図2 PDDの内部構造
(サブDBインデックス部とレコード部)

3.2 サブDBのアクセス権

他の開発者による情報の破壊や不当な情報参照からサブDBを保護するために、PDDではサブDB単位にアクセス権の設定が行える。このアクセス権には、サブDB毎に所有者、同一グループの開発者またはその他の開発者に対して情報の参照権および更新権がある。

3.3 サブDBの払い出し/戻し

大規模なソフトウェアの開発では、マシン負荷の集中を避ける等の理由から、いくつかのマシン上で分散して開発を行う。この場合、開発DBの情報も、それぞれのマシン上で利用できなければならない。このような場合のために、PDDは、DB間の情報の一部をサブDB単位でコピーし、払い出す機能がある。また、情報の追加・変更の行われたDBのコピーを、元のDBに払い戻すことも可能である。

DBの払い出し/払い戻しの時には、DBの他の部分から参照される情報（オブジェクト、名前）の管理に注意しなければならない。PDDは、元のDBとコピーされたDBとの間で、この参照に矛盾が生じることのないようDBが払い出されている間、これらの情報の削除や移動を保護する。

4 性能

サブDBを利用して情報を管理した場合の情報検索および登録時間を図3に示す。

測定条件は次のように設定した。

- ・総レコード件数：50000件
- 検索（1件、インデックスなし）
- サブDBインデックス部・レコード形式：4フィールド（整数型2、文字型2）
- ・サブDBの分割数：1～10

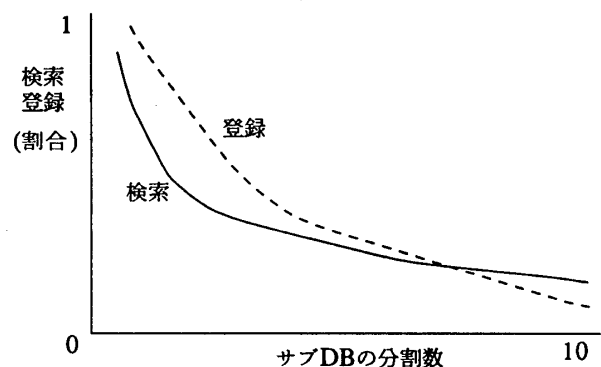


図3 サブDB分割数に対する検索時間

5 おわりに

ソフトウェア設計情報用データベースPDDのサブDBについて報告した。

サブDB機能を追加したことにより、検索範囲の絞り込みやデータの保護など、複数のメンバによるソフトウェア開発の支援に適した情報の管理が可能になった。今後、このPDDを有効に利用していくためにユーザインタフェースであるPDDシェルの機能強化などと併せて、改良を継続していく予定である。

参考文献

- [1] 小林他：「統合プログラミング環境(2) - 開発支援データベース -」 情報処理学会第39回全国大会
- [2] 湯原他：「統合プログラミング環境(3) - ソフトウェア設計情報用データベースPDDの開発」 情報処理学会第41回全国大会
- [3] 濱崎他：「統合プログラミング環境(4) - ソフトウェア設計情報用データベースPDDに基づく開発環境」 情報処理学会第41回全国大会