# 大規模プロジェクトにおけるデータアクセス機能の効率的な開発方式*

堀野　智久　畑山　研　遠藤　浩　石川　貞裕 †
日立製作所　情報・通信グループ　プロジェクトマネジメント統括推進本部 ‡

## 1．はじめに

日立製作所では、多くのシステム開発に3階層アーキテクチャを採用している。3階層アーキテクチャでは、システムを画面、業務ロジック、データアクセスの3層に分けて開発を行う。この内、データアクセス層の開発では、他層の開発チームからSQL(Structured Query Language)の申請を受けて、データアクセス層開発専任者がシステムに実装する。

大規模プロジェクトにおけるデータアクセス層の開発では、申請の量や開発母体の大きさから申請内容の各種チェック(精査)の作業負荷が高く、開発効率が悪化する傾向にある。データアクセス層の開発遅延は上位層である画面/業務ロジック層の開発にも影響し、プロジェクト全体のスケジュール遅延につながる。また、データアクセス層のチェック(精査)作業が疎かになると、類似SQLが重複開発されたり、性能要件を満たせない品質の悪いSQLが登録されるため、チェックのスピードアップだけでなく、質の向上も求められている。

そこで本論文では、こうした問題に対する日立製作所での取り込みについて紹介する。

## 2．3階層アーキテクチャ

弊社におけるシステム開発でのアプリケーション設計にあたっては、図1のように3階層を明確に分離し、実装・テストまでこの分類に従う開発方式を推進している[1]。

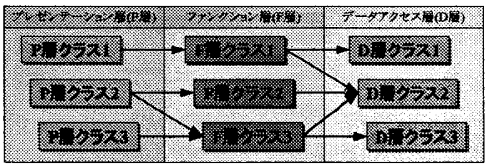


**図1：3階層アーキテクチャ**

プレゼンテーション層(P層)には画面表示の処理(P層クラス)、ファンクション層(F層)には業務ロジック(F層クラス)、データアクセス層(D層)には主にDB(データベース)アクセス処理(D層クラス)を実装する。D層を分離する狙いには、共通化したDBアクセス処理の上位層からの共有利用や、DBアクセスに関する性能問題発生時の影響範囲の局所化などが挙げられる。

開発の基本スタイルはP層、F層、D層ごとに1グループ以上の開発チームを割り当てるが、D層は単一のグループ（以下、「D層開発チーム」と呼ぶ）が責任を持って開発、運用を行う方針としている。D層開発を単一のグループが専任することで、次の効果を得ることができる。

① システム全体のDBアクセスパターンを集中的に管理し、D層クラスの冗長開発を防ぐと共に、共通利用の促進が図れる。

② D層開発チームにDataBase Management System(DBMS)の有識者を割り当てることで、DBアクセス処理の品質確保が早い段階で行える。例えば、性能要件を満たさないDBアクセス処理を申請段階で見直すことができる。

さらに弊社では、自社開発の専用のツール[2]を使用して、D層プログラムのソースコードをほぼ100%自動生成している（図2）。

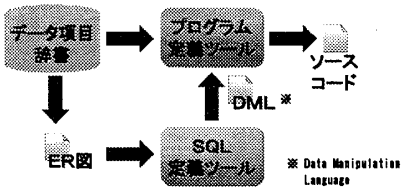


**図2：D層ソースコードの自動生成方式**

SQL定義ツールは専用のユーザインタフェースを備え、言語やDBMS製品に依存しないDBアクセスの設計情報をもとに、プラットフォームに依存したDML(Data Manipulation Language)の生成を行うことができる。

## 3．D層開発の運用

D層開発の運用の概要を図3に示す。図3におけるD層開発の流れは、次の通りである。

① 業務分析を行い、必要なテーブルとSQLを洗い出す。
② P層、F層開発チームが、SQLの申請を行う。
③ 重複申請の有無の確認を行う。
④ SQLについてアンチパターン等の精査を行う。
⑤ D層クラスの開発・テストを行う。

---

* Methods for Efficient Development of Data Access Programs in Large-Scale Projects

† Tomohisa Horino, Ken Hatayama, Hiroshi Endoh, Sadahiro Ishikawa
‡ Hitachi, Ltd. Information & Telecommunication Systems. Project & Process Management Division.

⑥ D 層クラスを P 層、F 層開発チームへ提供する。

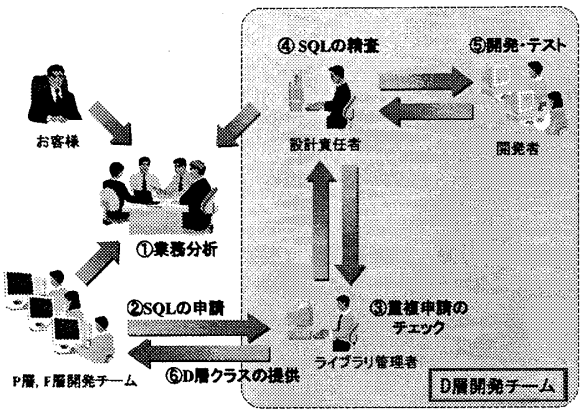


**図 3：データアクセス層の開発の流れ**

SQL の精査は、各層の開発チームで、それぞれ表 1に示した観点で行う。プロジェクト規定のルールや重複チェックなどの単純な精査は申請側の P 層、F 層開発チームでも行い、作業負荷の分散を図っている。

**表 1：精査方針の観点一覧**

| No | 精査方針の観点 | 担当 |
|---|---|---|
| 1 | プロジェクト規定のルールに違反した SQL | P 層, F 層, D 層 |
| 2 | 重複や類似(既存で代替可能)の SQL | |
| 3 | 文法的に誤りのある SQL | D 層 |
| 4 | 性能上、問題がある SQL | |

## 4．D 層開発を効率化する取り組み

D 層開発の効率を上げるための施策として、主に次の 2 点について取り組んでいる。

(1) 重複・類似 SQL の排除

SQL 定義ツールに登録済みの D 層クラスの中から、重複・類似した SQL を機械的に検索する仕組みを用意する。大規模プロジェクトでは SQL の本数が数百から千のオーダとなるため、重複・類似の申請有無のチェックを目視で行うのは非現実的である。そこで、重複・類似の SQL を検索するツールを用意する（以降「SQL 定義検索ツール」と呼ぶ）。

SQL 定義検索ツールは、SQL 定義ツールに登録した SQL の中から、SQL の構造（検索条件、ソート条件など）をキーに SQL を検索する。SQL の一部分を指定して検索することもできるため、類似した SQL の検索も可能である。

(2) SQL コーディングのチェック

インデックスの利用有無やテーブルの結合順序など、SQL のコーディングで留意すべき点は多い。弊社では、SQL コーディングでのアンチパターンを纏め、各プロジェクトに展開を行っている。SQL の精査作業の段階でアンチパターンを用いたチェックを取り入れることで、後工程のチューニング過程で発生する SQL 文見直しなどの手戻り作業を減らすことができる。

現在は、各プロジェクトの D 層開発専任者が手作業でアンチパターンのチェックを行っているが、さらなる効率化を図るため、今後は我々の部門で自動化のための仕組みを作る予定である。

## 5．施策の評価

これまでに紹介した施策の効果について、表 2にまとめる。表 2は、実際のプロジェクトへ適用したデータを元に試算した。

**表 2：施策の評価**

| No | 取り組み | 試算効果 |
|---|---|---|
| 1 | 重複・類似 SQL の排除 | 開発効率が約 1.2 倍に向上 (削減 SQL 本数は約 14%) |
| 2 | SQL コーディングチェックの自動化 | 開発効率が約 1.1 倍に向上 |

実際のプロジェクトのデータから、削減可能な SQL の本数は約 14%と見込まれる(表 2 No.1)。この場合、D 層の開発効率は約 1.2 倍向上する。

コーディングチェックの自動化では、見直しが不要な SQL の抽出にかかる工数で差が出ると想定すると、開発効率は約 1.1 倍向上する見込みである(表 2 No.2)。

## 6．まとめと今後の課題

今回、大規模プロジェクトでのデータアクセス層の開発において、効率的で効果的なシステム開発を実現する施策とノウハウを、システム開発者へ展開・提案できたと評価している。

我々の部門では、これまでここに記載した開発プロセスを各プロジェクトに適用、支援する活動を行ってきた。今後も各プロジェクトでの実績を評価し、手法の改善、ツールの機能強化を図っていく。

参考文献：
[1] 斉藤岳、他『DOA を取り入れたコンポーネント指向開発手法』, FIT2003
[2] 堀野智久、他『データベースセントリックに行うモデルベースの開発手法』, 情報処理学会第 67 回全国大会