

# プログラム分散開発環境におけるデータベース定義・保守方式

7F-4

小林敦司 中村浩司 中村仁之輔

NTT情報通信網研究所

## 1. はじめに

ワークステーションの普及等により、データベース用アプリケーションプログラムの開発においても、埋込みSQLが記述されたプログラムのプリコンパイルが行われるマシン（開発環境）とプログラムが走行するマシン（走行環境）が異なるプログラム分散開発が進展しつつある。

分散開発環境の場合、DBの定義情報（スキーマ）は、開発環境と走行環境に存在し、DBの定義変更により、スキーマの内容が変更された場合、双方のスキーマを同期して保守する必要がある。

スキーマの同期は、通信により実現する方法があるが、銀行のオンラインシステムのように、サービス環境によっては、開発環境と走行環境の通信がセキュリティ等の面から許されない環境が存在する。

本稿では、DBの定義変更時に、通信が許容されない両環境のスキーマの同期を実現するデータベース定義・保守方式を提案する。

## 2. 前提

本稿で用いる環境のモデルを図-1に示す。

- (1) APのプリコンパイルは開発環境で、APの実行は走行環境で行われる。
- (2) スキーマは開発環境と走行環境のそれぞれで、スキーマファイルとして保有。
- (3) 開発環境と走行環境の通信は許容されない。

本稿では、上記の場合を前提として、スキーマの同期を可能とするデータベース定義・保守方式を提案する。

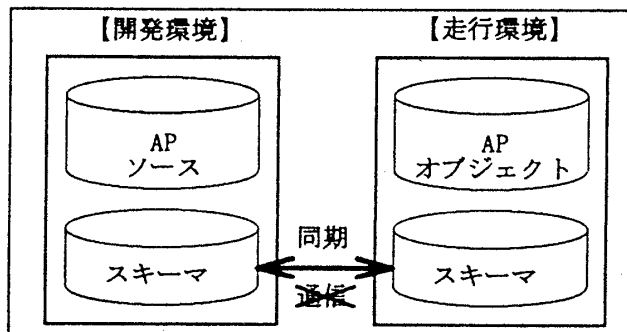


図-1 分散開発環境

## 3. データベース定義・保守方式

DBの定義変更により、スキーマの内容が変更された場合の問題点としては、開発環境と走行環境が異なるため、双方のスキーマの同期ずれが起こることが考えられる。そこで、スキーマの同期を実現するために、以下の方式を提案する。（図-2）

### (1) 開発環境でスキーマの定義変更

スキーマの定義変更は最初に必ず開発環境で行う。DDLによるスキーマの定義変更のインタフェースは、以下の2種類に限定する。

- ① 埋込みDDL（AP中に create table 等のDDLを直接記述する）
- ② ユーティリティプログラムへのファイル入力

開発環境のスキーマの定義を実際に変更するのはユーティリティプログラムとする。埋込みDDLインタフェースの場合は、プリコンパイラがDDL情報を出力し、ユーティリティプログラムがそのDDL情報を元にスキーマの定義変更を行う。

### (2) 開発環境でスキーマの世代情報の更新

世代情報として、埋込みDDLインタフェースの場合はプリコンパイル時のタイムスタンプ（例：YYMMDDHHMMSS）を、ユーティリティプログラムインタフェース

A Database Definition and Maintenance Method in Distributed Program Development Environment  
 Atsushi Kobayashi, Koji Nakamura and Jinnosuke Nakamura  
 NTT Network Information Systems Laboratories  
 1-2356 Take, Yokosuka, Kanagawa 238-03, Japan

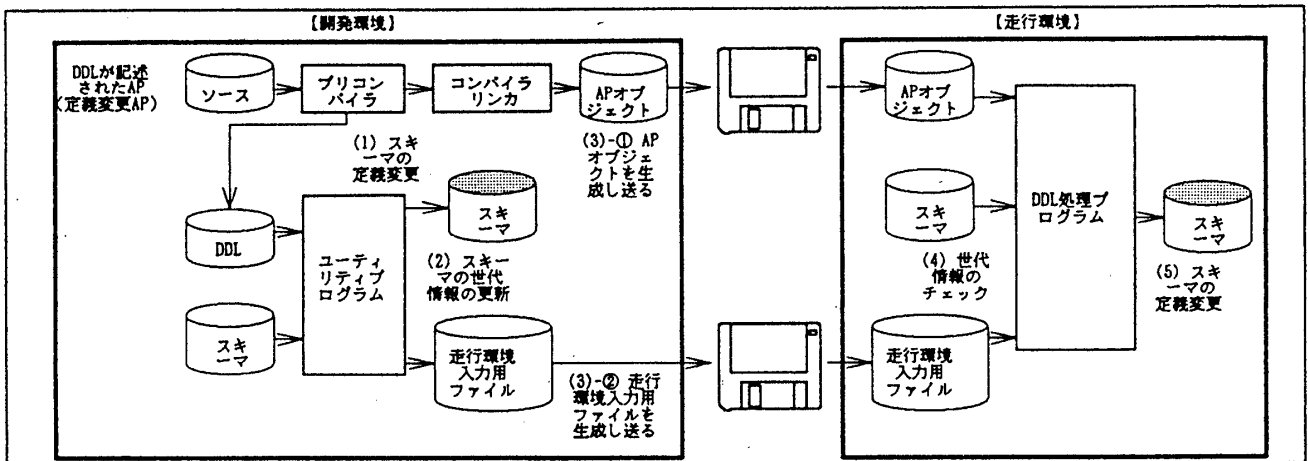


図-2 データベース定義・保守の流れ

の場合は開発環境のスキーマ変更時のタイムスタンプをテーブル毎にスキーマに格納する。

- (3)-① APオブジェクトを生成し実行環境にFD等で送る (埋込みDDLインタフェースの場合)

プリコンパイラはプリコンパイルした際に取得した世代情報を付加してコンパイラに引き渡す。APオブジェクトの中には世代情報が付加される。

- (3)-② 実行環境入力用ファイルを生成し実行環境にFD等で送る (ユーティリティプログラムインタフェースの場合)

実行環境入力用ファイルは開発環境のユーティリティプログラムがスキーマを変更した際に取得した世代情報を付加して実行環境に引き渡すファイルである。実行環境入力用ファイルの形式を図-3に示す。

...	DDL文	...	旧タイムスタンプ	新タイムスタンプ	...
...	DDL文	...	旧タイムスタンプ	新タイムスタンプ	...
⋮					

図-3 実行環境入力用ファイルの形式

- (4) 実行環境で世代情報のチェック

スキーマの格納形式を図-4に示す。実行環境でのスキーマの変更順序のチェックを行うために、開発環境のスキーマの世代情報 (タイムスタンプ) をテーブル毎に格納する。

スキーマヘッダ	スキーマ名称
	:
テーブル情報 (TABLES)	テーブル名称
	タイムスタンプ
	:
カラム情報 (COLUMNS)	カラム名
	:
インデックス情報 (INDEXES)	インデックス名
	:

図-4 スキーマ構成

埋込みDDLインタフェースの場合はAPオブジェクトからDDL処理プログラムが起動される。ユーティリティプログラムインタフェースの場合は実行環境入力用ファイルを入力として起動する。DDL処理プログラムはAPオブジェクト又は実行環境入力用ファイル中の旧タイムスタンプと該当するTABLESのタイムスタンプが一致しているかをチェックする。

- (5) 世代ずれがなければ、実行環境のスキーマの定義変更処理を行う

DDL処理プログラムは実行環境のスキーマの定義を変更し、タイムスタンプを新タイムスタンプに変更する。

4. おわりに

開発環境で行ったDBの定義変更時に、スキーマの変更情報およびスキーマの世代を出力し、その結果を実行環境の入力とすることにより、両環境のスキーマの同期を実現する方式を示した。