

分散開発環境におけるデータベース定義運用方法

1 P - 6

上溝徹 中村浩司 芳西崇

NTT情報通信研究所

1. はじめに

アプリケーションプログラム（AP）開発の生産性向上のため、分散開発環境が整えられてきている。

分散開発環境で埋め込みSQLが記述されたプログラムの開発を行う場合、データベースの定義情報（スキーマ）は開発環境と走行環境の両方に必要となる。更に、スキーマの内容を変更する場合は、双方のスキーマを同期して保守する必要がある。

また、スキーマの内容変更を重ね、世代が進んでいる場合に、最新世代のスキーマを持つ走行環境を新たに構築する方法も考慮する必要がある。

本稿では、分散開発環境において世代が進んだスキーマの管理を容易にするデータベース定義運用方法について提案する。

2. 現状の問題点

2.1 分散開発環境

本稿で対象とする分散開発環境に対する前提条件を以下に示す（図1）。

- ①同一のAPを複数の走行環境で走行可能とするため、走行環境間でのスキーマ世代の同期が必要である。

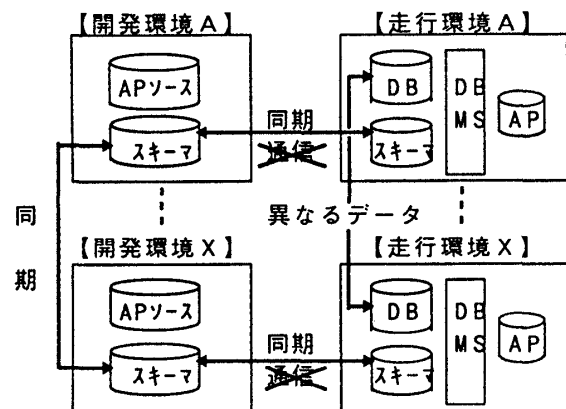


図1. 分散開発環境

A Database Definition and Operation Method in Distributed Development Environment.

Tohru Uemizo, Koji Nakamura and Takashi Honishi
NTT Information and Communication Systems Labs.

1-2356 Take, Yokosuka, Kanagawa 238-03, Japan

ある。

- ②走行環境上のDBには、スキーマ世代が同一であっても異なるデータが格納されている。
- ③開発環境と走行環境間の通信を介さず、スキーマ同期を実現する。

2.2 現状のスキーマ更新方法

現状のスキーマ更新の流れを図2に示す。まず開発環境のスキーマを更新し、それを世代情報とともに走行環境に反映する手順でスキーマの同期を実現している。^[1]

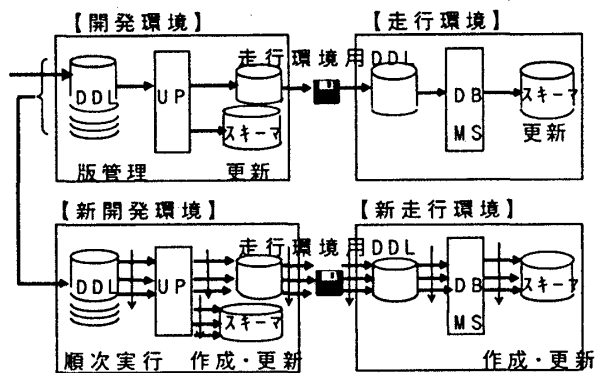


図2. 現状のスキーマ変更の流れ

- (1) 走行環境のスキーマを更新する場合

開発環境において、スキーマ更新用DDL文（データベース定義変更言語:Data Definition Language）をユーティリティプログラム（UP）に入力し、スキーマ更新と、走行環境スキーマ更新用DDL文生成を行う。

走行環境において、このDDL文を用いてスキーマを更新する。

- (2) 走行環境を新規に作成する場合

開発環境でスキーマ初期生成に用いたDDL文を(1)と同様の手順により、新しい開発環境と走行環境のスキーマ初期生成を行う。

開発環境にて用いられたスキーマ更新用DDL文を、実行された順序通りに同様の手順で実行し、走行環境用スキーマを順次更新する。

2.3 問題点

現状のスキーマ更新方法の問題点を以下に示す。

(1) DDL文の版数管理

スキーマの内容変更が重ねられた場合、同一世代のスキーマを得るには、全てのスキーマ変更用DDL文を同一実行順序で実行する必要がある。従って各開発環境では、DDL文の版数管理が必要になり、その数は増大する一方となる。

(2) 新走行環境作成時

新規に同一世代のスキーマを持つ走行環境が必要となった場合、保管していたDDL文を順番に実行する必要がある。

3. データベース定義運用方式

前述の問題点の解決を図るために、開発環境上の最新スキーマから走行環境上のスキーマを生成する方法を提案する。

3.1 DBMSの内部処理

データアクセス方法は、DBMSの内部処理方式に左右される。本稿では、以下のようなDBMSを想定する。

- ①埋め込みSQL方式の場合、実行時にカラム位置を決定する方法とプリコンパイル時に決定する方法があるが、高速化を図るために、後者の方法を採用する。
- ②①で採用した方法では、カラム追加やカラム属性変更を行ってカラム位置が変化すると、通常はAPの再コンパイルが必要になるが、極力APを再コンパイルしないようにすることを考慮し、変更のあったカラムのみ位置を変更するDBMS内部処理を採用する。

3.2 実現方法

上記の場合、全ての走行環境で同一のAPを再コンパイルなしに走行可能とするには、テーブルの物理的構造も一致させる必要がある。

開発環境上の最新スキーマから走行環境上のスキーマを生成するには、単純にユーザが意識するカラム以外の情報も反映する必要があり、DDLだけでは記述できない。そこで、走行環境への入力情報と

して、物理的構造を一致させるために必要な情報をDDL文に補完した独自の入力ファイルを生成する機能を開発環境に持たせる方式で解決した(図3)。

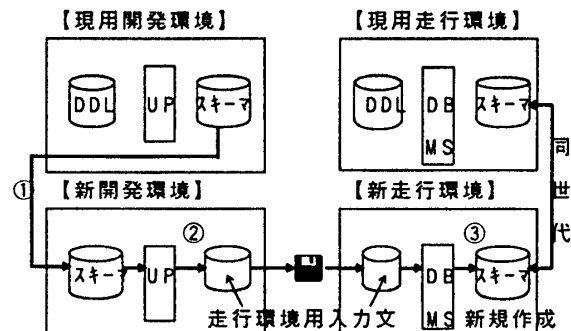


図3. 新しい運用方式

- ①現用の開発環境上の最新スキーマを新開発環境へコピーする。
- ②コピーしたスキーマを入力ファイルとして、UPにより、最新のスキーマを生成する走行環境用入力文を出力する。
- ③②で得られた入力文を走行環境で実行することにより、新走行環境にスキーマを生成する。

3.3 効果

本方式により以下の効果が期待される。

- ①スキーマ変更のDDL文の版数管理が不要となる。
- ②新規に同じスキーマを持つ走行環境を構築することが容易になる。
- ③現用走行環境で動作している埋め込みSQLが記述されたAPの再コンパイルを必要としない。

4. おわりに

分散開発環境におけるスキーマ定義の運用方法として、開発環境上の最新スキーマから走行環境スキーマを生成する方法を提案した。この方法により、現在稼働している走行環境と同じスキーマを持つ新走行環境構築の運用処理を少ない工数で実現可能とし、開発環境におけるスキーマの世代管理も容易となる。

参考文献

- [1]小林敦司, 中村浩司, 中村仁之輔: 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, 情処第48回全国大会, 7F-04, 1993