

# 統合的データモデリング支援環境“Darwin”(4)

4E-7

## 一物理データモデル支援ツール“Darwin/PD”の紹介一

小山 宏志<sup>\*</sup> 内野 康彦<sup>\*</sup> 中尾 守<sup>\*</sup> 加藤 雅樹<sup>\*\*</sup>

<sup>\*</sup>株式会社 日本コンピュータ研究所

<sup>\*\*</sup>NTTデータ通信株式会社 技術開発本部

### 1 はじめに

データモデリングの普及と共に、方法論の適用を支援するCASEツールが数多く作られている。モデリングを行う際に論理的に正しいモデルを構築することの重要性は改めて述べるまでもないが、実世界のプロジェクトに応用するにはそれに加えてターゲットシステムの性質にあわせた無数の情報を付加しなければならない。

このことを考えると、データの分析からデータベースのスキーマ定義、さらには保守までを一貫して支援するCASEツールが望まれる。

“Darwin”はERモデルにもとづいてソフトウェアシステムのデータ構造の分析からリレーショナルデータベースのスキーマ実装までを支援するツールである。ツールは分析から設計までを概念・論理・物理の3段階に分け、モデルを徐々に詳細化することを前提としている。

本稿で報告する“Darwin/PD”はこの内物理段階の設計を支援するツールである。

### 2 コンセプト

先に述べたとおり“Darwin”はERモデルに基づいてデータ構造の開発を支援する。“Darwin/PD”の目指すのは、正しく構築された論理データモデルを前提とし、それを論理モデリングツールに近いルック・アンド・フィールの下で物理データモデルとして完全な形式に修正し、最終的にはそのまま使えるデータベース定義をDDLとして出力することである。(図1)

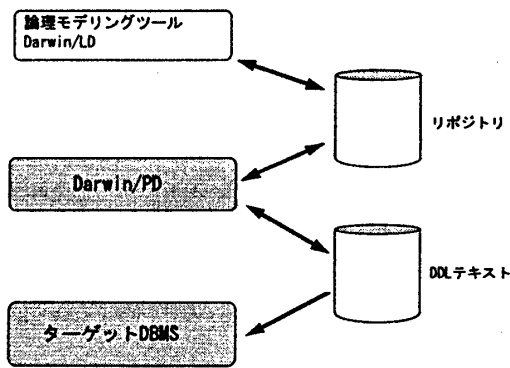


図1 ツールコンセプト

この目標を達成するために、“Darwin/PD”の扱う物理データモデルはERモデルそのものとは異なったものになっている。次章にその詳細を述べる。

### 3 物理データモデル

ERモデルとリレーショナルデータベースは全く等価ではないが、論理データモデルから実リレーショナルデータベースの定義を得るに当たってエンティティをテーブルに、エンティティのアトリビュートをカラムに、リレーションを外部キーに変換することが一般によく行われており、データベースの論理設計を目的としたとき概念的にも表記上も直観的に理解しやすい。“Darwin/PD”でもこの方式を採用している(図2)。

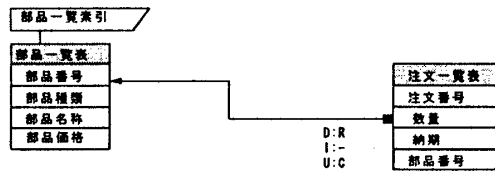


図2 物理モデル表記例

しかしながら以下に述べるように、物理段階の設計を支援するにはこれだけでは不十分である。

第1に、ターゲットDBMSのために実用的なデータベースを作るにはテーブルと外部キー定義だけでは不十分である。

使用者の観点からは、単に論理的に正しい物理スキーマを生成するだけでなく、実世界での性能上の要求や資源の割り当てまで含めた実用に堪えるスキーマ定義を得ることが望まれる。このような実用上の要求を満たすには、インデックス、ビュー、トリガ等の多様なデータベースオブジェクトをも要素として扱う必要がある。さらにこれらオブジェクトの各種物理的特性を定義するパラメータを与える必要がある。

そこで“Darwin/PD”はインデックス、ビュー、トリガ等を独立したオブジェクトとしてモデルの要素に加えている。それに加えてターゲットDBMSに固有のオブジェクトを何種類か定義できるようになっている。

第2に、物理設計の要素としてインデックス、ビュー等を扱う必要を認めたとして、モデルの表記としてそれらをどのように表すかという問題が残る。

ビューやシノニム等をERダイアグラムのエンティティ

## Integrated Data-Modeling Environment “Darwin”(4)

### ~Introduction of Physical Data-Modeling Tool “Darwin/PD”~

Hiroshi Koyama<sup>\*</sup>, Yasuhiko Uchino<sup>\*</sup>, Mamoru Nakao<sup>\*</sup>, Masaki Kato<sup>\*\*</sup>  
Computer Institute of Japan, Ltd.

<sup>\*\*</sup>Research and Development Headquarters,

NTT DATA COMMUNICATIONS SYSTEMS Corp.

に準じる形式で表現することは可能であるが、その様にすると論理モデルと物理モデルとの間で表記上のギャップが大きくなるし、図表上でオブジェクトが増えすぎてツールの操作が煩雑になる可能性が高い。

さらに生成のターゲットとなる実データベースは、使用するDBMSによって異なった種類のオブジェクトが存在するため、ターゲットDBMSの種類によってモデルの表記方法を変えなければならない。このことは現在の様に異機種混在の計算期環境が一般的になってくると都合が悪い。

第2の点に対して“Darwin/PD”では、ほぼ全てのRDBMSに存在すると思われるテーブル、外部キー、インデックスだけをERダイアグラム風の図表で表示し、その他のオブジェクトに関しては別の図表または一覧表形式による表記を採用している。

## 4 機能概要

これまでに述べたとおり“Darwin”ではデータモデルの分析から実データベースの生成までを支援する。“Darwin/PD”の持つ物理モデル定義支援の機能としては以下のようなものがある。

### 4.1 論理モデルから物理モデルへの変換機能

エンティティをテーブルに、リレーションを外部キーにそれぞれ変換する。論理モデルのサブジェクトエリアはスキーマに変換する(図3)。エンティティ内の基本キー定義やアトリビュートのNOT NULL情報、ユニーク情報、値域情報から、テーブルおよびカラムに制約またはインデックスを付加する。また論理段階でのアトリビュートの型と長さから、実データベースの為のカラムの型と長さを設定する。さらにデータベースオブジェクトの物理パラメータにデフォルトの値を設定する。

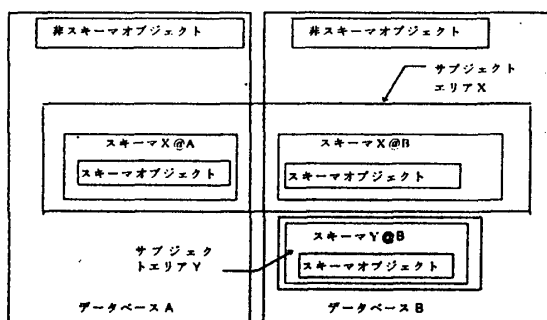


図3 論理モデルと物理モデル

### 4.2 物理モデルの操作機能

論理モデルから引き継いだ情報に加えて、物理段階でビュー、トリガ、シノニム等を作成し、テーブルに関連付けることができる。テーブルや外部キーに対する変更も可能である。変更の際には物理モデルの整合性を自動または半自動的に保つことができる。

例えばカラム定義機能を用いてあるカラムがプライマリであると入力すれば、そのカラムは自動的にプライマ

リインデックスの構成要素に含まれるようになる(または、必要ならモデル上で新たにインデックスを作る)。データベースオブジェクトの物理パラメータの変更が可能である。パラメータの値に関してチェック機能があり、異常な入力にはエラーまたは警告を発する。

ターゲットDBMSが分散データベースをサポートする場合には物理モデルを複数のデータベース領域に分割することができる。分割した領域間を可視にするために必要なオブジェクトがあれば自動生成する。

また物理パラメータの設定を支援するため、与えられたパラメータに基づいてテーブル及びインデックスが消費する記憶容量を計算する機能がある。

### 4.3 DDL生成機能

作成した物理データモデルを実現するために、DBMSに直ちに入力できる形式のDDL文のテキストファイルを生成する。

### 4.4 DDLリバースエンジニアリング機能

DDLを記述した既存のテキストファイルを読み込んで、物理データモデルを逆生成する機能がある。当然逆生成したモデルを操作して変更可能である。

## 5 おわりに

本稿では、統合的データモデリング支援環境“Darwin”のうち、物理モデル設計を支援するツールである“Darwin/PD”のコンセプトと機能概要を紹介した。本ツールで将来改良していくべき機能として以下のようなものが挙げられるであろう。

- 最終的出力をDDLテキストで得るのではなく、直接DBMSのリポジトリを操作する。
- 多種類のRDBMSに対応する。
- 分散データベースでのトラフィック量の予測を支援する。
- 物理データモデルの表記法を改善する。
- OODBMSに対応する。