

## WWWによるワークフロー処理系の試作と評価

3 J-2

— 永続オブジェクト管理機構の実装について —

元田 敏浩、 瀧野 修、 川崎 隆二

NTT ソフトウェア研究所

## 1. はじめに

我々はこれまでにワークフロー処理系の問題点について様々な角度から検討を進めてきた[1][2]。現在のワークフロー処理系の主たる問題点として、ベンダ依存性と閉域性が挙げられる。これに対する一つの解決手法として WWW[3] を用いたワークフロー処理系の構築を試みた。処理系の実装に当たってはワークフロー処理モデルとの相性から全てをオブジェクトとして実装する事にした。また、ワークフローが実行されるタイムスケールや執行履歴の保管の必要性を考慮すると、全てをメモリオブジェクトとして実装する事は現実的でなく、何らかの永続オブジェクト機構を必要とする。オブジェクト指向データベース[5]は永続オブジェクトの一実装形態ではあるが、データベース毎の差異が大きく移植性に問題がある他、ホスト言語が C++ や Small Talk 等に限定され、開発効率や WWW とのインタフェース性の面で問題がある。我々はワークフロー処理系の構築用にリレーショナルデータベースと perl5 [4] を用いた永続オブジェクト管理機構を実現した。本稿ではこの実装について述べる。ワークフロー処理系の実装については別稿で言及する。

## 2. 永続オブジェクト管理機構の実装

リレーショナルデータベース(以下 RDB と略す)に永続オブジェクトを格納するに当たり、(1) メソッドは perl のモジュールファイルとして保管し、(2) 属性情報は RDB のテーブルに格納するレコードとした。メソッドを RDB に格納する事も検討したが、複数の異なるバージョンの永続オブジェクトが同時に存在する状況は現実には少ないと想定され、実装の複雑化や性能劣化を避けるため最新バージョンのメソッドのみを保持する事とした。また属性は RDB のテーブルにレコードとして格納するため、単一のスキーマのみ許されるという RDB の制約から、属性情報の構成も最新版のみを保持する。クラスが永続オブジェクトとして保持すべき属性はクラス定義ファイル中に宣言する。RDB スキーマとの同期は手動で行う。(図 1)

1つのクラス定義は原則的に RDB の 1 テーブルに対応させた。perl では数値・文字列およびそれらの配列程度の単純なデータ型を利用する機会が多いため、1属性値は 1 カラムデータとして保管する。1 オブジェクトは 1 レコードとして保管する。ただし、配列データは通常の RDB では単純にカラムデータとして格納できないため、1 配列属性に対して 1 サブテーブルを用意し OID による主テーブルと関連させた。主テーブルには全てのスーパークラスの属性を含めたカラムを配置した。クラス毎にテーブルを用意する方法も考えられるが、検索が複数パスになるため採用しなかった。

perl5 の言語仕様の的には永続化機能を持つクラス「Pobj」を用意し、永続化するクラスを Pobj のサブクラスとして定義し永続性を持たせる。Pobj が提供するメソッドは、クラスの登録・削除、メモリオブジェクトの生成・永続化・格納・検索・削除・更新、OID 等の Pobj 属性の取得・設定である。サブクラスでは、これらのメソッドを利用して意識的にオブジェクトの永続化を行う。

本実装において特に工夫を要した点は、コヒーレンスキャッシュである。永続オブジェクトは RDB 中のレコードデータとして保持するが、操作時には利便性や性能面から通常の perl5 のメモリオブジェクトとして属性データを保持する。このため、一時的にデータが二重管理される状態となり、一貫性制御が必要とな

A workflow processing system with WWW, its implementation and evaluation.

- An implementaion of persistent objects manager -

Toshihiro Motoda, Osamu Takino, Ryuji Kawasaki

NTT Software Laboratories, 3-9-11 Midoricyo Musashino-shi Tokyo 180, Japan

る。複数プログラム間での一貫性制御はRDBのトランザクション機能をそのまま利用した。同一プログラム内での一貫性を制御するためにコヒーレンスキャッシュを用意した。コヒーレンスキャッシュは、RDB検索時に検索結果のキャッシュとして働くと同時に、オブジェクトIDに対するメモリオブジェクトを管理する事により、同一プログラム内でのメモリオブジェクトの単一性を保証している。そのため、検索時のみならず、新規メモリオブジェクトの永続オブジェクト化に際してOIDを付与する際もコヒーレンスキャッシュへの登録を行う。また、コヒーレンスキャッシュ以外にも処理性能の観点からオブジェクトテーブル管理用テーブルに対するキャッシュも用意した。

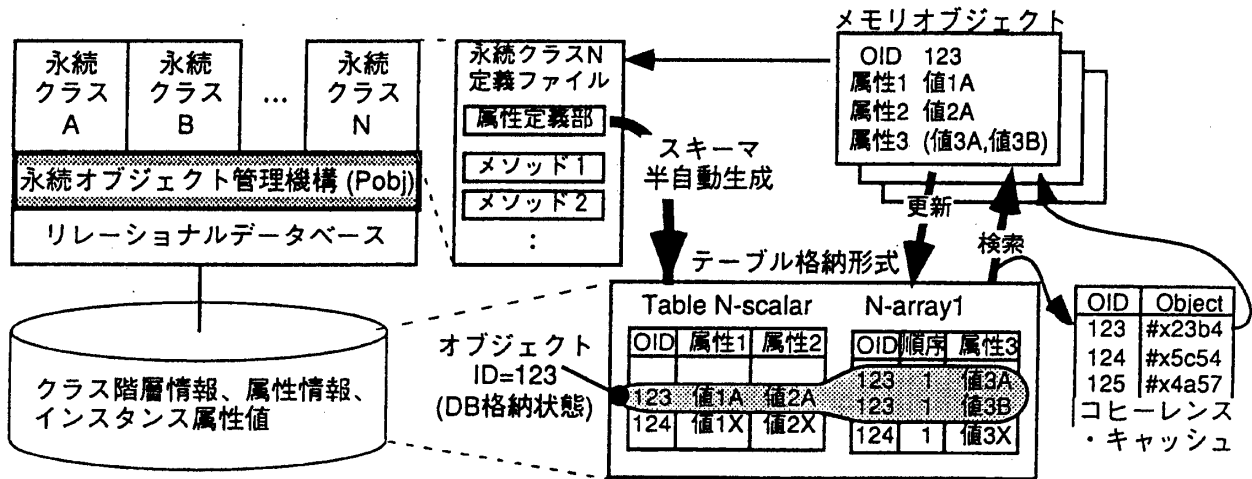


図1 永続オブジェクト管理機構 Pobj の構成

### 3. 評価

処理系をインプリメントした結果、RDBへのアクセス部：1,196 step、Pobj：2,063 stepとなった。7個の手続きオブジェクトを持つ1個のワークフロー定義オブジェクト(合計8オブジェクト)を複製し、ワークフロー実行体を生成する手続きで測定した。最初の複製ではSQL発行回数合計 N=36、うちデータ入出力 Nd=22、オブジェクト管理テーブルアクセス Nc=14であった。Ndのうち14件は複製されたオブジェクトの格納のための insert 命令であった。2回目以降は N=24 に減少する。これは Nd=20、Nc=4 と減少するためである。Ndの減少はコヒーレンスキャッシュ、Ncの減少は管理テーブルキャッシュの効果である。データ入出力に比較して管理テーブルへのアクセスが多いが、キャッシュ化によりアクセス回数が33%減少した。しかし、WWWではプロセスが断続的に起動され、必ずしもキャッシュの効果が期待できるとは限らない。

### 4. おわりに

perl5上に永続オブジェクト管理機構を実現し一応の動作が確認されたが、性能的には実用的なレベルにはない。今後はWWWに適した構成による高性能化を目指したい。また、本機構は既存のリレーショナルデータベースへのオブジェクトラッパーとしての応用も考えられ多方面への活用も検討したい。

### 謝辞

本研究を進めるに当たり、日頃御討論頂く山本修一郎グループリーダーに感謝します。

### 参考文献

- [1] 元田, 瀧野, 長岡: HCI設計手法とその環境に関する一検討, 情報処理学会第47回全国大会
- [2] 元田, 瀧野, 川崎: 市販ワークフロー製品の業務システムへの適用評価, 情報処理学会第49回全国大会
- [3] The World Wide Web consortium: "The World Wide Web", <http://www.w3.org/hypertext/www/>
- [4] Bill Middleton: perl5 WWW page, <http://www.metronet.com/perlinfo/perl5.html>
- [5] Setrag Khoshafian著, 野口喜洋・小川東訳: オブジェクト指向データベース, 共立出版