

7L-4

ISO IRDS1 の実装と評価

岩崎 一正、穂鷹 良介
筑波大学

1. はじめに

[1]にて同様の主旨の報告を行なったが、本報告はその後の経過報告である。IRDS(情報資源辞書システム)とは、企業など多種多様なメタデータベースを扱う組織において、それらを設計・開発・管理する統一的な手段を提供することを目的とするシステムである。現在、ISO/IECによって第一段階のサービスインタフェースについての規格化が検討されているが、そのドラフト[2]をもとにIRDS1のあるべき姿を模索し、試験用に実装した経過について報告する。

2. アーキテクチャ

[2]では、大まかに二つのことを規定している。それは、データ構造とそれを扱う際のサービスインタフェースである。

簡単に説明をすると、メタデータベースが(複数の)応用データベースを管理するのと同じように、(複数の)メタデータベースを管理するメタメタデータベースを考える。IRDS1では、これらのデータベース間の関係を階層構造をなすと見做して、AP_Lp(応用DBの属するレベル対)、IRD_Lp(メタDB、IRDSではIRDと呼ばれる、が属するレベル対)、IRDD_Lp(メタメタDBが属するレベル対)と呼んでいる。

IRDS1が直接対象とするのは、IRD_LpとIRDD_Lpである。このうちIRDD_Lpには、唯一つのシステムで提供されるデータベースだけが存在し、ユーザの定義する様々なIRDを設計・開発・管理するための機能を有する。ユーザによって作られたIRDは、今度はメタDBとして様々な応用DBの設計・開発・管理を目的に利用されることになる。

[2]では、IRDD_LpのDBの構造と二つのLpに属するDBを操作するための基本的なサービスのインタフェースを定義している。ここで規定されているIRDS1の機能には、DBMSの基本的な機能であるデータの追加・検索・更新・削除の他に、一つ下のLpに属するDBの設計データを蓄積し、それを基に新たなDBを作成し管理するための機能もある。このため、作業単位として、Working Setという概念が用意されており、これを単位として設計データのバージョン管理やライフサイクル管理が行なわれる様になっている。

しかし、[2]の仕様には実装面から考えると不備が残っているため、この面から詳細な検討を加えることにした。検討結果の概略を以下に示す。

IRDS1のすべてのオブジェクト(その記述がデータとして蓄えられるもの)は、

バージョンを持つものと持たないものに分類ことができ、バージョンを持つのはWorking Setのメンバーに限る。Working Setを管理するオブジェクトなどはバージョンを持たないものとする。

Working Setは、ひとつの既存のWorking Setに基づいて新規に作成することができ、このとき基になった方をBasis WS、新規作成されたほうをVersion WSと呼び、Version WSは、Basis WSのすべてのメンバーを継承するように作成される。また、Basis WSで行なわれた変更は基本的にそのすべてのVersion WSに伝播される。データとしては、各WSでのメンバーの更新データだけを記録する。つまり、あるVersion WSのメンバーに関するデータは、そのBasis WSのメンバーとの変更差分だけが記録保持されることになる。

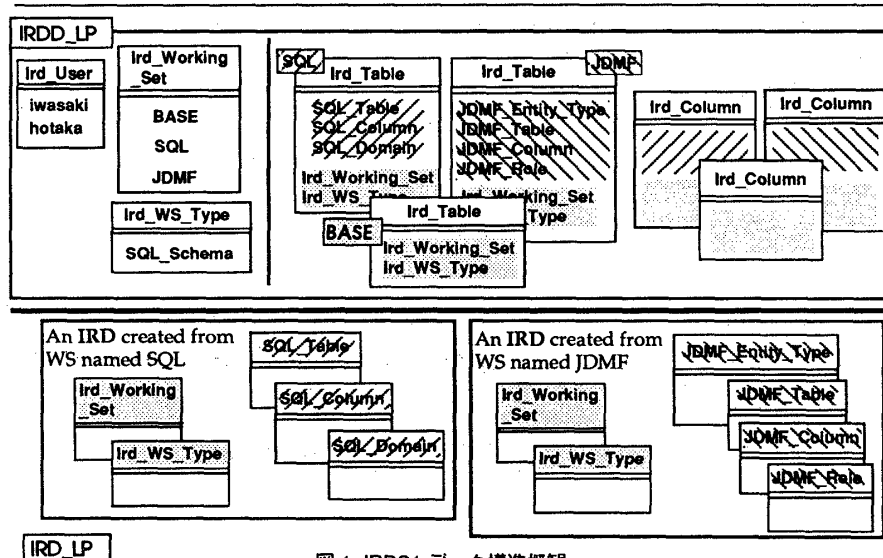


図 1. IRDS1 データ構造概観

これを可能にするため、各WSを対象にして何らかの作業を行なう場合には、メンバーデータを蓄積されている差分データから合成する、WS Sessionの開設という手順を踏むことにする（すなわち、バージョンを持つデータを扱うには特別な準備が必要とされる）。

Working Set単位で設計データのライフサイクル管理を行なうが、そのフェーズとしては、Uncontrolled, Validated, Controlled, Archivedという四つを用意し、ユーザはそれぞれをさらに細分化することもできるものとする。それぞれのフェーズの意味は、現在開発途中にある（U）、Working Setの種類ごとに定められた規則に照らしてその妥当性が保証されている（V）、現在何らかの利用状態にある（C）、現在はお蔵入りの状態である（A）となっている。

Working Set単位のバージョン管理やライフサイクル管理はIRDD_Lp, IRD_Lp共に適用可能な機能であり、そうあるべきであるため、各IRDは、Working Setを管理するためにIRDD_Lpのデータベースで採用されているデータ構造を有することが期待される。また、二つのLpで共通のデータ構造を採用することで同じサービスがどちらのLpでも利用可能となる。これを半ば強制するために、BASEという名前の特別なWorking Setを用意し、Working Setを管理するためのデータ構造の定義データをこのWSのメンバーとし、他のすべてのIRD設計用WSをBASEのVersionとして作成するという規約を設ける（図1）。

3. 実装

前回の発表時には、Sun ワークステーション上で動くORACLE DBMSを利用して実装したが、今回はSun ワークステーション上で動くSQL インタフェースを持つSYBASE DBMSを利用して実装した。具体的には、DBの部分にSYBASE上のDBとし、このDBにアクセスする応用プログラムとしてIRDS1のサービスを提供するプログラムをC言語で作成した。

さらに、今回はSun-OSのRPC機能を利用してサーバとして機能するようにした。

[2]では、データ構造の定義をSQL2のDDLで記述しているが、ORACLEもSYBASEもSQL2を完全にはサポートしていないため参照キーの定義など幾つかの制約を実現する機能を持っていない。今回の実装では、[2]で規定された内の一部の制約をSYBASE独自の機能であるTriggerとRuleを利用して実現した。

実装の目的が、IRDS1の仕様が実際の仕様に耐えうるものか実験するために、実現されたプログラムインタフェースを利用する応用プログラムも作成することにした。そのために、原始的ながらも対話的なコマンドインタフェースをユーザに提供する応用プログラムCLIPを作成した。これは、Working Setを作成したり、Working Set Sessionを開設したり、データの検索・更新など、IRDS1の提供するサービスを使って実現できる簡単な操作を行なうことができる（図2）。

4. まとめ

IRDS1の仕様検討とそれに基づく実装を行なった。今後の課題としては、実装されたシステムを実際に使ってみた経験をIRDS1の仕様にフィードバックすること、IRDやそれによって管理される応用DBの設計支援を行なうプログラムの開発などが残されている。

参考文献

- [1] 岩崎他：SQLによるIRDSの実現、第40回全国大会講演論文集(II)
- [2] ISO/IEC JTC1/SC21 WG3 N4895: IRDS Services Interface, Working Draft, June 1990
- [3] ISO/IEC 10027: IRDS Framework, June 1990
- [4] 穂鷹：ISO IRDSのレベルバラレリズムと自己増殖性、本論文集

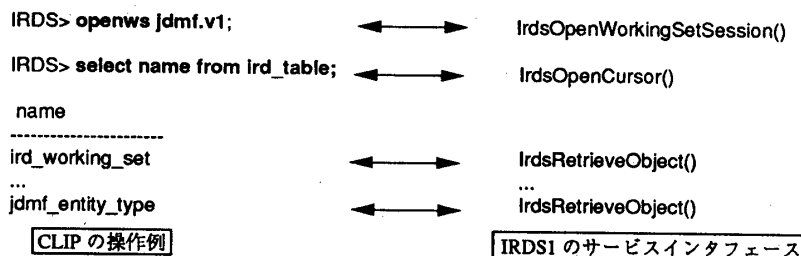


図2. CLIP コマンドと IRDS1 サービスインタフェースの対応関係 (抜粋)