

移動計算機における情報ベース検索スクリプトの合成方式

7P-6

大森 匡 Wisut Sae-Tung 星 守
電気通信大学 大学院 情報システム学研究所*

1 研究の目的

最近のコンピュータネットワークでは WWW など様々な情報資源を提供するサーバ（情報ベース）が自然発生的に点在しており、ユーザが移動計算機からこうした情報資源を自由に検索・加工したい、という要求が大きい。しかし、これら情報ベースはユーザが望む利用手段や情報構造をあらかじめ用意していないことが多く、結果としてほぼ全ての情報を移動計算機側からブラウズしがちである。この問題を解決するためには一般に仲介者と呼ばれる補助機構が必要である。我々はスクリプト言語による遠隔プログラムを使ってこの機構を提案してきた [1]。

図1が提案した仲介者機構による検索手順である。すなわち、ユーザは未知な情報ベースに対し、(1) 始めに、ユーザは情報ベースへそのクラス定義を問い合わせ、その仕様 (i.e. 情報構造と利用手段) を教えてもらう。(2) 次に、移動計算機側で自分がやりたい検索スクリプトを合成し、(3) 情報ベースに送って実行させる。この手法では、個々のユーザに依存した情報検索・加工手続き自体（「関心がある情報を探す」など）は移動計算機に載ってネットワーク内を移動し、必要な時に情報ベースへ送られる。従って情報ベース側ではそれ固有の検索手段以外はスクリプト言語の処理系だけを用意すれば良い。

我々は この機構を Perl に永続オブジェクト操作機能をいれたスクリプト言語 Persistent Perl を用いて試作している。しかし、移動計算機上にあらかじめ持っていた検索スクリプトと、情報ベース固有の情報との二つを合わせて使うようなスクリプトをユーザが直接 Persistent Perl を使って記述するのは難しい。本稿では オブジェクト指向 SQL を使ってこうした検索スクリプトを合成する方式を論じる。

2 検索スクリプト合成方式

本研究における検索スクリプト合成の方針は次のようなものである：

- ・「オブジェクト指向 SQL (OO-SQL) を使って図1の手順1で集めた情報ベースのクラス定義やメソッドとユーザが移動計算機に持っている検索手続きとを合成する。すなわち、図1でいうとユーザは OO-SQL を指

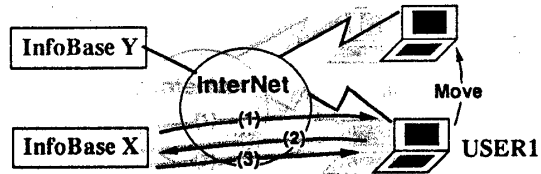


図1: 移動計算機と情報ベースとの検索手順モデル

定するだけであり、移動計算機上の OO-SQL コンパイラがこの SQL 文から必要な Persistent Perl スクリプトを合成し送り出す。」

この方式では、情報ベースが提供しているクラス定義のデータ構造、および移動計算機ユーザが持っている検索スクリプトはそれぞれ OO-SQL の選択対象と選択条件として与えられ、そこから情報ベース群の渡り歩きや適切な手続きの起動を行なうスクリプトが合成される。この最終的なスクリプトでは、ユーザ固有の手続きはサブルーチンとして実体を記述され一緒に転送されるが、情報ベース固有のメソッドはメソッド名だけが使われることになる。

図2に検索スクリプト合成機構の例を示す。ユーザは、情報ベースアドレス (Host List)、それらのクラス定義 (Class List/Definition)、移動計算機にあって使用したい検索手続き (User-Filtering Script) を選び、システムと会話しながら、OO-SQL 文を作る。Send ボタンを押すと、システムは自動的に検索スクリプトを合成し、情報ベースに送って実行させる。

提案した OO-SQL の構文は次のようになる。

```
select [Complex object の経路表現]
from [object variable] in [クラス名]
where [選択条件、検索手続き名の and/or 列]
```

ユーザがあらかじめ持っている検索スクリプトを Fx とすると、Fx はこの OO-SQL の where 節で使われる。もちろん、そのためには、Fx の引数のドメインとクラス定義の属性のドメインとが一致することが前提である。こうするとクラスの属性を検索スクリプトの引数にマップして簡単に OO-SQL 文を記述できる。

一般には、情報ベース (図1の X、Y) が提供しているデータ型やスキーマが検索スクリプト Fx の扱うものと整合しない、という場合も多く、この条件は満たされないこともある。この場合 (いわゆるスキーマ不整合) には、整合化ルールを情報ベースのメソッド仕様と共に送ってもらい、移動端末側で情報ベース X と Y の定義、および検索手続き Fx の三者を整合させるしかない。

*A Method of Generating Information-Retrieval Scripts on Mobile Computers. S.T.Wisut, T.Ohmori, M.Hoshi (U.Electro-Comm.)

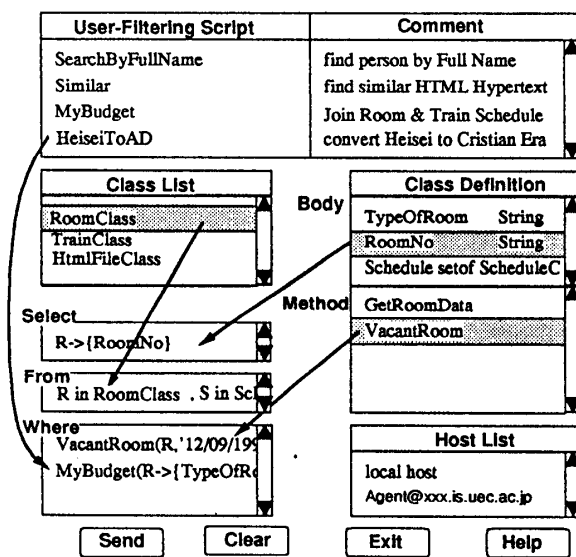


図 2: 検索スクリプトの合成インタフェース

つまり TSIMMIS の MedMaker [2] の機能を移動計算機の OO-SQL コンパイラにやらせるのである。すなわち、本研究では図 1 の情報資源ネットワークにおいて移動端末に置くデータベースとして (1) 情報ベースのクラス情報、(2) 整合化ルール、(3) ユーザ個人に対応した情報フィルタ手続き、を管理し、その上に (4) OO-SQL による検索スクリプト合成機能を提供していることになる。

3 適用例

3.1 情報検索スクリプトの合成

以下では具体的に例をあげて提案方式を説明する。まず、OO-SQL による検索スクリプトの合成例を示す。例として、WWW サーバ X に対し 移動計算機から「私が持っているハイパテキスト T と似ているテキストを探せ」という検索をしたいとしよう。ユーザは「ハイパテキスト T と似ている」という判定手続き `similar()` を移動計算機に持って来て、サーバ X に送り、実行させる。`similar()` は 個々のユーザが勝手に定義したスクリプトであり適用されるドメインのデータ型は String 形式の文書一般である。したがってユーザは以下の OO-SQL 文を記述すれば良い:

```
@ObjSet=&sql("select H->{FileName}, H->{Title}
from H in HtmlFileClass
where &similar(H->{FileName},
0.8,'perl','array')");
```

(`similar(X, th, K1, K2)` は 文書 X に対し 内部に持つテキスト T との間でキーワード K1, K2 についてベクトル空間モデルをつくって類似度 `th` 以上のものを true とする手続き。テキスト T は `similar` 内部のデータ構造として一緒に転送される。X のクラス構造は HTML 文書である。) 移動計算機側の仲介者機構はこの OO-SQL 文に対応した Persistent Perl スクリプトを生成し、テキ

スト T と一緒に転送する。このように String, TextFile など普及しているデータ型のドメインに対しては当然ながら OO-SQL による検索スクリプト合成は簡単にできる。

3.2 スキーマとデータタイプの不一致

次に、スキーマ不整合の場合を示す。下の例では宿泊情報ベースのクラス `RoomScheduleClass` が属性「客名」を姓と名に分けて維持しているが、ユーザの指定した検索スクリプト `SearchByFullName` は「姓名」を引数としている。したがって、両者の整合を行なう述語 `FNLNtoFullName()` を用いて OO-SQL 文を生成する:

```
@ObjSet = &sql("
select S->{Room}, S->{CustomerName},
S->{FromDate}, S->{ToDate}
from S in RoomScheduleClass
where &searchByFullName('Wisut Sae-tung',
&FNLNtoFullName(S->{FirstName},
S->{LastName}))");
```

次に 情報ベース二つを連動して使用する際に データ不整合が出る場合を示す。下の例では、各情報ベースは列車とホテルの空き情報検索メソッド `VacantTrain` (列車、到着日、..) (元号表示) と `VacantRoom` (部屋、到着日、出発日) (西暦表示) を提供しており、移動端末側では ある適切な組合せを選ぶ手続き `MyBudget` が作成されている (これは if-then-else の判定文である。) ここでは 空き情報の日程属性の値を元号から西暦へ変換する述語 `HeiseiToAD()` を使用している。

```
@ObjSet = &sql("
select R->{RoomNo}, R->{TypeOfRoom},
T->{TrainNo}, T->{TypeOfTrain}
from R in RoomClass, T in TrainClass
where &VacantRoom(R, &HeiseiToAD( TrainDate ),
&HeiseiToAD('H7.12.26'))
and &VacantTrain(T, TrainDate,
'Tokyo', 'Osaka')
and &myBudget(R->{TypeOfRoom}, 'VIP', 'single',
T->{TypeOfTrain}, 'Shinkansen', 'regular')");
```

いずれの例でも、スキーマやデータ型不整合を解消する述語を用いているが、こうした述語は、情報ベースから不整合解消ルールを送らせ それにより移動端末側で仲介者機構が対話的に合成するか、または 良く使うデータ型間の変換ルールを移動端末側で用意して使うしかない。特に、対話的なスクリプト合成を移動端末で行なう場合には 整合化ルールを情報ベースから移動端末に送ってもらった方が回線切断状態でもできて有利である。

4 おわりに

本稿では 情報ベース検索スクリプトを移動端末で合成するためにオブジェクト指向 SQL を用いる方式を提案した。この方法は 遠隔プログラムによる情報検索機構を考える時 極めて自然といえるが、スキーマ整合化をも移動端末側で行ないたい、という問題がある。今後 こうした課題も含め 試作を通して評価していきたい。

参考文献: [1] 大森, 他: 通信学会研究報告 DE95-54, 95 年 10 月. [2] Papakonstantinou, et al. *Med-maker: A mediation system based on declarative specifications*. Available by ftp at db.stanford.edu as file pub/papakonstantinou/1995/medmaker.ps