

SuperSQL 構文解析部の再構築によるエラー検出の品質向上

田嶋 将大[†]

慶應義塾大学理工学部
情報工学科[†]

五嶋 研人[‡]

慶應義塾大学大学院
情報工学専修[‡]

遠山 元道[¶]

慶應義塾大学理工学部
情報工学科[¶]

1 はじめに

SuperSQL は、独自のクエリを記述することによって関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とする SQL の拡張言語である。通常の SQL では、シンプルでフラットな表しか再現できないが、SuperSQL を用いることで様々な表を作成することができる。また、SuperSQL を用いると、HTML と PHP などを用いたサーバサイトプログラミングなどによる一般的な方法に比べて、はるかに少ない行数で Web ページを生成することができる。

従来の SuperSQL の研究開発において、その機能の拡張などの実装が行われ、必要に応じて構文解析部の構築や修正が行われてきた。このことから派生する問題点として、SuperSQL 特有のクエリの複雑な構文解析ルールが不明瞭になってきてしまっていること、エラー処理が適切に行われず、ユーザーへのエラー表示が適切に提示されないことがあることなどが考えられる。そこで、本論文では SuperSQL クエリの構文解析ルールの定義を行うことで構文解析部を再構築することで、エラー処理の品質を向上させ、ユーザーに対するエラーメッセージをより正確にすることを目指した。

2 SuperSQL とは

SuperSQLは関係データベースの出力結果を構造化し、多様なレイアウト表現を可能とするSQLの拡張言語であり、慶應義塾大学遠山研究室で開発されている^{[1][2]}。そのクエリはSQLのSELECT句をGENERATE <media> <TFE> の構造を持つGENERATE句で置き換えたものである。ここで<media>は出力媒体を示し、HTML, XML, Mobile_HTML5^[3]などの指定ができる。また<TFE>はターゲットリストの拡張であるTarget Form Expressionを表し、結合子, 反復子などのレイアウト指定演算子を持つ一種の式である。

Quality improvement of the error detection by the rebuilding of the SuperSQL parsing department

[†] Masahiro Tajima : Keio Uni. Computer Science department

[‡] Kento Goto : Keio Uni. Computer Science department

[¶] Motomichi Toyama : Keio Uni. Computer Science department

SuperSQLのアーキテクチャを図1に示す。

SuperSQL処理系は、構文解析部(Parser)、リスト構造生成部(List Constructor)、メディア生成部(Code Generators)から成る。SuperSQL クエリが発行されると、まず構文解析部で通常のSQL文とレイアウト式に分け、SQL文をDBMSに渡して検索結果を受け取る。リスト構造生成部では受け取ったフラットな結果に対し、レイアウト式に従って階層的な構造を持たせる。最後にメディア生成部がクエリで指定したメディアファイルを結果として出力する。

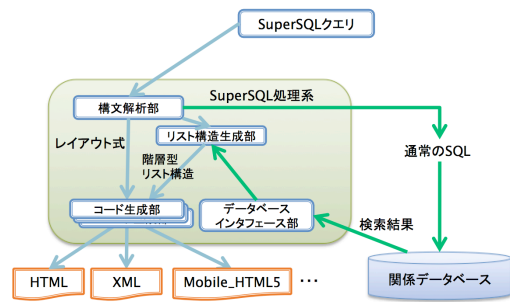


図1 SuperSQLアーキテクチャ

3 SuperSQL 構文解析部の再構築

この章では、以下のクエリを例にして本研

```
GENERATE Mobile_HTML5 {
  header("映画レビュー一覧")!
  [image(m.image, ),
  {
    {m.title}@{bgcolor=BlanchedAlmond}!
    :
  }
}
FROM movie m, company c, genre g
WHERE m.company = c.id and m.genre = g.id
```

究で行った内容について述べる。

```
*** mismatch paren at '!' ***
Error[TFEparser]: Syntax Error in TFE
{header("映画レビュー一覧")!{m.title}@{bgcolor=BlanchedAlmond}!{image(m.image,
{公開年:}|m.year|line("")!制作会社:|"c.name"|line("")!映画ジャンル:|g.name
@{width=150} >>>>} <<<< !レビュー! ...
```

実際のエラー要因

エラー指摘箇所

図2 従来の構文解析によるエラーメッセージ例

図2は従来の SuperSQL の構文解析部によるシンタックスエラーメッセージの例であり、クエリエラーの内容によって実際のエラー要因と指摘

されるエラーメッセージとがずれてしまうことが問題として挙げられる。図2の例では、image関数内の記述が誤ってしまっているが、指摘されるエラー要因はその後の”}”を指している。

本研究では、構文解析部の構築に、LL(*)方式の解析手法をとる構文解析部を生成する ANTLR^[4] というパーサージェネレーターを使用した。一般に用いられる構文解析手法には、LL方式とLR方式があるが、LR方式ではエラーの検出は早いことが言えるが、LL法に比べてコンパイルエラー時のエラーメッセージを適切に出しにくいという欠点があり、本研究の実現の為にLL方式の構文解析手法を採用した。図3のように SuperSQL クエリの記述ルールの定義を行い、SuperSQL クエリの構文解析ファイルや字句解析ファイルを生成させている。

```
media : (K_GENERATE | error) IDENTIFIER ;
operand :
(
  (function
  | OPEN_BRACE{braceflag++;} exp CLOSE_BRACE{braceflag--;}
  | (sorting)?((table_alias '.')? column_name)
  | grouper
  | composite_iterator
  | if_then_else
  | STRING_LITERAL
  | NUMERIC_LITERAL
  | keyword
  )
  ('||' operand)*
)
(DECOLATOR)?
;

grouper :
OPEN_BRACKET{bracketflag++;}
exp
CLOSE_BRACKET{bracketflag--;}
C1
|
OPEN_BRACKET{bracketflag++;}
exp
CLOSE_BRACKET{bracketflag--;}
C2
|
OPEN_BRACKET{bracketflag++;}
exp
CLOSE_BRACKET{bracketflag--;}
C3
;
;
```

図3 SuperSQL 構文解析ルール定義(一部抜粋)

ここで、この生成されるプログラムによるクエリの構文解析中で定義したルールに反する記述が発見された時、その例外が発見された時のクエリ中のトークンを保持し、そのトークンに基づいてエラー処理を行う処理系が ANTLR の内部で定義されている。本研究では、SuperSQL クエリ特有のエラーに合わせてエラー処理を行うように、またユーザーにエラーメッセージを適切に提示できるように、この内部処理系と別で定義を行っている。この定義された処理系では、保持されるエラー要因となっているトークンやその前後のトークンによってエラー要因を推定し、そのエラー箇所の提示とエラーメッセージの生成を行っている。

この処理によって、図2と同じクエリの構文解析を行った結果のエラーメッセージが図4である。これによって、エラー要因のクエリ中での位置の特定が行えるとともに、実際のエラーの原因と考えられる箇所を従来のエラーメッセージと比べてより正確に指摘することができるようになった。

```
NoViableAlt
line 3:16 parse error detected.
The OffendingToken is : ')'
GENERATE Mobile_HTML5 {
  header( "映画レビュー一覧" )!
  [image(m.image >>>>,<<<<< ) ,
  {
    {m.title}@{bgcolor=BlanchedAlmond}!
```

図4 新構文解析部によるエラーメッセージ例

4 評価

本研究の評価のため、慶應義塾大学理工学部情報工学科の3年生を対象とした授業であるデータモデリングの SuperSQL を用いた最終課題において、実際に3年生の記述したクエリのログを用いて、それぞれの構文解析を行う。

そのクエリログには全 9862 クエリが採取されており、それぞれのクエリに対して SuperSQL の実行が成功したか失敗したかが情報として保持されている。

この情報を用いて、これらのクエリの中から、実行の失敗となったクエリ(全 2373 クエリ)のみを収集し、それぞれのクエリのエラー原因の特定とその分類を行っている。このログを用いて、各エラーパターンに対応できているのかを確認していく予定である。

また、従来のエラーメッセージと新たに生成されるエラーメッセージとを比較し、その内容の正確さが向上したことかを確認していく予定である。

参考文献

[1] SuperSQL: <http://supersql.db.ics.keio.ac.jp/>
 [2] M. Toyama: “SuperSQL: An Extended SQL for Database Publishing and Presentation”, Proceedings of ACM SIGMOD '98 International Conference on Management of Data, pp. 584-586, 1998
 [3] 五嶋 研人, 遠山 元道. “SuperSQLによるモバイルWebアプリケーション生成機構の実装”, 慶應義塾大学 修士論文, 2013.
 [4] Terence Parr “Adaptive LL(*) Parsing: The Power of Dynamic Analysis”, Proceedings of ACM International Conference on Object Oriented Programming Systems Languages & Applications(OOPSLA), pp. 579-598, 2014