DEWS2007 B7-7

SuperSQLによるA-docファイルの生成

高橋　健太郎†　　遠山　元道††

† †† 慶應義塾大学理工学部情報工学科　〒 223–8522 神奈川県横浜市港北区日吉 3–14–1
E-mail: †kentaro@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　A-doc とは Web における利用者主導による情報資源の結合を実現する情報資源表現形式である。A-doc を閲覧中の Web 文書に結合することで閲覧中の文書の見出し語部分が自動的に対応する文書へのハイパーリンクに変換される。HTML にアタッチ可能な A-doc ファイルは複数のキーワードとそれに対応したドキュメントを表す HTML、もしくはその URL の組から成るエントリを持つ XML 形式のファイルである。A-doc のエントリ数は数個から数百万を超えるものもあると考えられる。そこで、出力媒体と出力構造を指定可能な SQL 拡張言語である SuperSQL の新しい出力媒体として A-doc を追加し、RDB 中のデータから A-doc の本体となる XML とドキュメントの HTML を自動で生成することを実現した。

キーワード　A-doc、Web、SuperSQL、DB 言語

Generation of A-doc files with SuperSQL

Kentaro TAKAHASHI† and Motomichi TOYAMA††

† ††Department of Information and Computer Science, Faculty of Science and Technology,
Keio University
Hiyoshi3–14–1, Kouhoku-ku, Yokohama-shi, Kanagawa, 223–8522 Japan
E-mail: †kentaro@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

Abstract　A-doc is an information resource expression form which join the information resources in Web. The keywords of Web document is converted into hyperlink to a corresponding document by attaching A-doc to it. The A-doc file that can be attached to HTML is a file of XML format with the entries consist of the pair of keyword and HTML or the URL expressing a document corresponding to it. There seems to be a A-doc with a large number of entries. So, in this paper, we added A-doc as a new output media of SuperSQL that SQL extension which can specify output structure and media such as HTML, XML and PDF, and it can generate both of XML, which is main body of A-doc, and HTML, which is document, from RDB at once.

Key words　A-doc, Web, SuperSQL, DB Language

1. はじめに

Web における利用者主導による情報資源の結合を実現するために、我々は A-doc と呼ぶ情報資源表現形式の提案、開発を行っている。A-doc は見出し語とそれに対応する文書のペアの集合で、辞書形式であり、それを閲覧中の Web 文書に結合すると、閲覧中の文書の見出し語部分が自動的に対応する文書へのハイパーリンクに変換される。

その利用目的が辞書的役割を果たすため、多数のエントリを持つ A-doc も存在すると考えられる。そこで本研究では A-doc のファイル形式を定義し、出力媒体と多様なレイアウト表現の指定を可能にする SQL 拡張言語、SuperSQL の新たな出力メディアとして A-doc を追加し、SuperSQL のクエリ一つで RDB から A-doc の本体である XML 文書と各ドキュメントの HTML を自動で生成することを実現した。

以下、本稿の構成を示す。まず 2. 章で SuperSQL の概要、3. 章で A-doc の概要について述べる。次に、4. 章で A-doc のファイル形式を示し、5. 章で SuperSQL による A-doc ファイルの生成について述べる。6. 章では本システムを用いた出力例を示し、7. 章で検討・今後の課題について述べる。最後に 8. 章で結論を述べる。

2. SuperSQL

この章ではファイル生成に用いる SuperSQL について簡単に述べる。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$< media >< TFE >$ の構文を持つ 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，PDF 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。

2.1 結 合 子

結合子はデータベースから得られたデータをどの方向 (次元) に結合するかを指定する演算子であり, 以下の 4 種類がある。括弧内はクエリ中の演算子を示している。

- 水平結合子 (,)

データを横に結合して出力。

例：Name, Tel

name	tel

- 垂直結合子 (!)

データを縦に結合して出力。

例：Name! Tel

name
tel

- 深度結合子 (%)

データを 3 次元方法へ結合。出力が HTML ならばリンクとなる。

例：Name % Tel

name	→	tel

2.2 反 復 子

反復子は指定する方向に、データベースの値があるだけ繰りかえして表示する。また反復子はただ構造を指定するだけでなく、そのネストの関係によって属性間の関連を指定できる。例えば

[科目名]! , [学籍番号]! , [評点]!

とした場合には各属性間に関連はなく、単に各々の一覧が表示されるだけである。一方，ネストを利用して

[科目名 ! [学籍番号 , 評点]!]!

とした場合には、その科目毎に学籍番号と評点の一覧が表示されるといったように、属性間の関連が指定される。以下、その種類について述べる。

- 水平反復子 ([ ],)

データインスタンスがある限り、その属性のデータを横に繰り返し表示する。

例：[Name],

name1	name2	…	name10

- 垂直反復子 ([ ]!)

データインスタンスがある限り、その属性のデータを縦に繰り返し表示する。

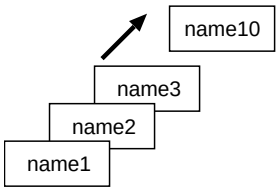
例：[Name]!

name1
name2
…
name10

- 深度反復子 ([ ]%)

データインスタンスがある限り、その属性のデータを奥行き方向 (HTML ではリンク、PDF ではページ変換) に繰り返し表示する。

例：[Name]%



3. A-doc

ここでは、本研究の出力結果を用いる A-doc の概要について述べる。A-doc は Web における利用者主導による情報資源の結合を実現するための情報資源表現形式である。

3.1 背 景

近年、検索エンジンの普及によって人々は日常的に Web を利用し情報検索を行うようになった。その上で、ユーザは検索エンジンによって必要な情報資源を検索し結果の Web 文書中から必要な情報を得る。Web 上では関連する情報はハイパーリンクで結合される。リンク元の文書とリンク先の文書間の関連はホームページ作成者の意図した関係のみで提供されている。このため、ユーザが Web 文書中で見つけた単語や名称を元に新たな情報を得たいという要求が生じた場合、ユーザはさらにその単語を検索エンジンにかけて情報検索を行わなければならない。

3.2 アタッチ

基本的な A-doc は見出し語とそれに対応する文書のペアの集合で、言わば辞書形式といえる。見出し語は関係データモデルの主キーに相当し、文書は主キーに代表されるテキスト、HTML もしくは XML 文書である。例として山の地理情報に関する mountain.adoc の概念表を表 1 に示す（実際には A-doc は XML で記述する)。

Web 利用者は閲覧中の文書にある山についての詳細な情報を得たい場合、従来では山の名前を一つずつ検索エンジンで検索せねばならない。これに対し A-doc では現在の文書に “mountain.adoc” を結合することで、閲覧中のページにある山の名前から、それぞれの山の情報へのハイパーリンクが一斉に生成される。これは、関係データベースにおける外部キーと主キーに基く結合と本質的に等しく、この結合を “アタッチ” と呼ぶことにする。A-doc の名称における A はその形態が連想的であることを表す Associative と、その利用においてアタッチを行うことから Attachable の両義を代表している。

同じ山の名前をキーとした A-doc でも例えば、ある組織が上

表 1　mountain.adoc

見出し語	ドキュメント
富士山	山梨県・静岡県 標高 3778m
浅間山	群馬県・長野県 標高 2568m
阿蘇山	熊本県 標高 1592m
…	

記の地理情報の A-doc を提供し、一方、別の組織が山の気象情報を記載した A-doc を提供すれば、利用者は必要に応じて自由に A-doc を選択し、自分の閲覧している文書にアタッチすることで山々の詳細情報を簡単に切り替えることができる。

A-doc はアタッチを行う文書に独立に存在するものであるため、誰でも “辞書” を作成・公開することができ、Web ページ作成者の意思に関係なく、ユーザ主体で動的に自由な結合を行うことができ汎用的であると言える。

アタッチには以下の三種類がある。

フルアタッチ

キーワードに合致する文書中の全ての単語に対しリンクを生成する

アンカータグによるアタッチ

Web 文書作成者がアタッチを行いたい単語に対しアンカータグを付与し、指定された単語にのみリンクを生成する

ユーザ指定アタッチ

Web 文書を閲覧するユーザがアタッチを行いたい単語を指定し、指定された単語が A-doc のキーワードと合致すればリンクを生成する

4. A-doc のファイル形式

論理的には A-doc ファイルはキーワード・文書のペアの集合であり、これを XML によって記述する。

固定的なコンテンツの場合、文書は文字列で直接記述すればよい。しかし、例えば地名をキーワードとし、その地域の気象情報を提供するコンテンツでは、文書は時々刻々変化する性質のものであるためコンテンツを直接記述するのではなくデータベースから情報を得てそれをレイアウトし、HTML に変換して表示するほうが妥当である。そこで、A-doc のファイルタイプを大きく分けて

（1） Static A-doc(静的 A-doc)

（2） Dynamic A-doc(動的 A-doc)

の二つに分類する。

4.1 Static A-doc

Static A-doc は単純に一つ以上のキーワードとそれに対応する文書のペアを一つのエントリとして記述する。キーワードを <kw> タグ、ドキュメントの HTML を <doc> タグで囲いそれらを一つとして <entry> タグで囲う。ドキュメントはタグ不整合の HTML も許容するためにコメント（<!--　-->）の形、もしくは HTML エンコーディングを施した形で持つ。各エントリの集合を <body> 要素の子要素とする。後で述べるように A-doc のエントリ数は数百万を超えるものもあると考えられる。エントリの HTML に共通部分が存在する場合、各エントリにそれぞれ記載するとファイルサイズも大きくなり冗長である。そこで、HTML の始めと終わりの共通部分をそれぞれ <html_preamble> と <html_trailer> に保持する。

エントリ数の多い A-doc のキーワードとドキュメントのペア全てを一つの XML 文書で保持するとファイルサイズが非常に大きくなってしまう。また、ドキュメントの内容を別のものに置き換えたい場合、該当エントリのドキュメント部分の HTML を全て書き換えなければならない。そこで、上記の Static A-doc を Static-standard タイプとし、Static A-doc のサブタイプとしてドキュメントの HTML を A-doc 本体の XML 文書中には記述せず、それぞれ個別の HTML 文書として保持するタイプを Static-separate タイプとする。Static-separate タイプでは <doc> タグにはそのキーワードに対応する HTML の URL が入る。Static-separate タイプの A-doc はある文書から他の文書へのインデクスであると見なすこともできる。

4.2 Dynamic A-doc

Static A-doc はキーワードと文書のペアを一つのエントリとしていたのに対し、Dynamic A-doc はキーワードに対する文書をを持たず、キーワードの集合のみを持つ。Dynamic A-doc をアタッチした場合、文書中に A-doc のキーワードと合致する単語を見つけると、静的なページへのハイパーリンクを生成するのではなく、そのキーワードを元にデータベースから必要な情報を得て、レイアウトする動的なページへのリンクを生成する。また、キーワード集合さえも保持せず指定された単語をデータベースから検索し、その情報を動的に生成するタイプの Dynamic A-doc も考えられる。この場合、アタッチはアンカータグによるアタッチと、ユーザ指定アタッチのみが可能である。

Dynamic A-doc は静的に保持することが困難な情報、例えば気象情報や株式情報などの時系列データを提供するのに適している。

5. SuperSQL による A-doc ファイルの生成

Static A-doc は単純な XML 文書であるためその作成は容易である。しかし、利用目的によっては A-doc のエントリ数は数個のものから数百万を超えるものも存在すると考えられる。その際、RDB のコンテンツからエントリ数の多い A-doc ファイルを生成したいという要求が生まれると考えられる。エントリ数の多い A-doc を手作業で作成するには多くの時間と労力を要する。RDB から XML を生成する試みは既に数多くのものが存在する [5] [6]。しかしこれらの技術を用いて生成した XML を A-doc として用いるには、さらにスタイルシートなどの技術を用いて XML を A-doc 形式に変換する必要がある。そこで、本論文では SuperSQL を利用して一つのクエリでデータベース

```
1  /* sample.ssql */
2  GENERATE ADOC
3          [
4                  {m.name, [a.alias],}
5                  %
6                  {m.name! m.hignt}
7          ]!
8  FROM mountain m, alias a
9  WHERE m.id = a.id
```

図 1　サンプルクエリ

中の情報を用いて A-doc の本体となる XML と各エントリのドキュメントを一括に生成する。

以下、A-doc 生成のための SuperSQL クエリについて述べ、その内部処理について説明する。

5.1 クエリ

サンプルクエリを図 1 に示す。

A-doc 生成のクエリは従来の SuperSQL の演算子のみで記述できる。A-doc 生成のクエリにおける二つの制約がある。まず、TFE 部分は垂直反復子 ([ ]!) で囲う。これは A-doc がキーワードとそれに対応する文書のペアを一つとするエントリの集合から構成されることによる。次に、垂直反復子の中でキーワードを指定する部分とドキュメントを指定する部分を深度結合子 (%) で分離する。ドキュメントは各キーワードに対する情報を表した HTML を示しており、アタッチを行うと、キーワードからその HTML へのハイパーリンクを生成することから深度結合子を用いるように実装を行った。

キーワードを複数指定する場合は水平結合子 (,) を用い、条件の合致する限り繰り返してデータベースからの値を指定したければ水平反復子 ([ ],) を用いる。%の前半ではキーワードとなる値を指定するだけなので、水平結合子・水平反復子以外の演算子は使用しない。

%の後ろ、ドキュメントの指定部分では通常の HTML 生成におけるクエリの TFE 同様に指定する。

本システムでは同じクエリで Static-standard タイプと Static-separate タイプの双方を生成することが可能である。現段階では通常の実行で Static-standard タイプ、起動オプションとして <-sep> を付けて SuperSQL を実行することで Static-separate タイプの A-doc が生成されるように実装した。

5.2 内部処理

A-doc 生成の処理は大きく%の前半と後半に分けて処理が行われる。%の前半部分ではデータベースから返された値にそのまま <kw> タグを付けて出力する。%の後半では通常の HTML 生成の処理を行う。その際、生成される HTML は全て同じレイアウトであるため、共通のスタイル情報を持っている。そこで、Static-standard タイプの A-doc を生成する場合は <HTML> の開始タグから HTML のヘッダ情報と <BODY> の開始タグまでが各 HTML で共通であるため <html_preamble> の要素とし、</BODY>、</HTML> の各終了タグを <html_trailer> 要素とする。HTML ソースの <BODY> タグ内の記述をそれぞれコメント（<!--　-->）にし <doc> タグを付けて出力する。Static-separate の場合は生成した HTML のソースを別名の HTML ファイルで出力し、XML 中の <doc> タグの中にはその URL を記述し出力する。

6. 実行例

図 1 のクエリにおける実行結果を図 2〜図 4 に示す。図 2、図 3 はそれぞれ同じクエリを用いて Static-standard、Static-separate タイプの A-doc を生成したものである。Statisc-separate の場合は実際には、図 3 の XML 文書と共に図 4 の様な各エントリのドキュメントである sample1.html、sample2.html ･･･ といった HTML 文書がエントリの数分生成される。

7. 検討と今後の課題

7.1 検討

従来の技術を用いて RDB から A-doc ファイルを生成する場合、まず RDB から XML 文書を生成し、次にスタイルシート等で A-doc のファイル形式に変換するという二段階の処理が必要となる。それに対し本システムではクエリ一つで RBD 中のデータから A-doc 形式の XML 文書を生成することが可能である。さらに、同時に多様なレイアウトを持ったドキュメントの HTML も生成することが可能である。

7.2 ドキュメントの深度方向への結合

A-doc を生成する SuperSQL クエリにおいてドキュメントの指定部分では従来の SuperSQL による HTML 生成と同様の TFE を指定することが可能である。ドキュメント指定部分で深度結合子 (%) を用いた場合 Static-separate タイプの A-doc を生成するのであれば、A-doc の <doc> タグで指定される HTML 文書の他に、そこからリンクの張られた HTML 文書が別に生成される。Static-separate タイプの A-doc は、ドキュメントを XML 文書中に保持するものであるため、SuperSQL の出力はいつも XML 文書のみであることが好ましい。しかし、現段階の実装ではキーワードに対応する深度 0 のドキュメントは XML 文書中に保持されるが、深度結合を使い、そこからリンク方向に結合した HTML を出力する場合は、二枚目以降の HTML は XML 文書の外部ファイルとして個別に生成されてしまう。この解決方法として以下の二つが考えられる。

まず、Static-separate タイプの <entry> の子要素として <doc> とは別に新たにドキュメントのリンク先用の要素を加え、そこにリンク先の HTML を持たせる方法である。

次に、各ページからのリンク先の HTML をドキュメントとして持つ新たな A-doc 出力する方法である。この場合、深度 0 のドキュメントの HTML 中では、そのリンク部分には適当なキーワードを持たせ、リンク用 A-doc へのアンカータグを付与しておく。このキーワードとリンク先のドキュメントの HTML をペアとしたエントリを持つ A-doc をリンク用 A-doc とする。深度 0 の HTML に自動的にリンク用の A-doc をアタッチすることでユーザにはあたかもハイパーリンクが埋め込まれた

```
<?xml version="1.0" encoding="Shift_JIS"
                                  standalone="yes" ?>
<ADOC type="Static-standard">
 <html_preamble><HTML><HEAD><STYLE
  TYPE=text/css><!-- .nest
  { height:100\%;} .att { padding:
  5px;} .linkbutton {text-align:center; margin-
  top: 5px; padding:5px;}
  --></STYLE></HEAD>
  <BODY></html_preamble>
 <html_trailer></BODY></HTML></html_trailer>
 <body>
   <entry>
     <kw>富士山</kw>
     <kw>Fujiyama</kw>
     <kw>Mt.Fuji</kw>
     <doc>
       <!-- <TABLE cellSpacing=0 cellPadding=0
       border=1 class="TFE10006"><TR><TD
       class="TFE10004 nest"><table class=
       "att TFE10004"><tr><td>富士山</td></tr>
       </table></TD></TR><TR><TD class=
       "TFE10005 nest"><table class="att
       TFE10005"><tr><td>3776</td></tr></table>
       </TD></TR></TABLE>
       -->
     </doc>
   </entry>
   <entry>
     <kw>浅間山</kw>
     <kw>Mt.Asama</kw>
     <doc>
       <!-- <TABLE cellSpacing=0 cellPadding=0
       border=1 class="TFE10006"><TR><TD
       class="TFE10004 nest"><table class=
       "att TFE10004"><tr><td>浅間山</td></tr>
       </table></TD></TR><TR><TD class=
       "TFE10005 nest"><table class="att
       TFE10005"><tr><td>2568</td></tr></table>
       </TD></TR></TABLE>
       -->
     </doc>
   </entry>
    ...
 </body>
<ADOC>
```

図 2　実行結果:Static-standard

```
<?xml version="1.0" encoding="Shift_JIS"
                                  standalone="yes" ?>
<ADOC type="Static-separate">
 <body>
   <entry>
     <kw>富士山</kw>
     <kw>Fujiyama</kw>
     <kw>Mt.Fuji</kw>
     <doc>sample1.html</doc>
   </entry>
   <entry>
     <kw>浅間山</kw>
     <kw>Mt.Asama</kw>
     <doc>sample2.html</doc>
   </entry>
   <entry>
     <kw>阿蘇山</kw>
     <kw>Mt.Aso</kw>
     <doc>sample3.html</doc>
   </entry>
    ...
 </body>
<ADOC>
```

図 3　実行結果:Static-separate

```
<HTML>
<HEAD><STYLE TYPE=text/css><!--
.nest { height:100%;}
.att { padding: 5px;}
.linkbutton {text-align:center; margin-top:
5px; padding:5px;}
--></STYLE></HEAD>
<BODY>

<TABLE cellSpacing=0 cellPadding=0 border=1
class="TFE10006"><TR><TD class="TFE10004 nest">
<table class="att TFE10004"><tr><td>
阿蘇山
</td></tr></table>
</TD></TR>
<TR><TD class="TFE10005 nest">
<table class="att TFE10005"><tr><td>
1592
</td></tr></table>
</TD></TR>
</TABLE>

</BODY></HTML>
```

図 4　実行結果:sample1.html

```
GENERATE ADOC
    {
        [ 日本人選手名 % {レイアウト 1} ]!
    }!
    {
        [ メジャーリーガー % {レイアウト 2} ]!
    }
FROM 野球選手
```

図 5　複数レイアウトの指定

HTML 文書のように見えるだろう。

7.3　複数のドキュメントレイアウト

現段階では、ドキュメントのレイアウトは全てのエントリで同じであり、キーワードによって異なるレイアウトを指定することはできない。しかし、実際には条件によって、異なるドキュメントのレイアウトを指定できることが好ましい。その場合、クエリの TFE 部分は条件ごとに垂直反復子（！）で連結を行う（図 5)。また、条件によるレイアウトの数が増えるとクエリ自体が大きくなり、見にくくなるため、レイアウト指定部分を外部ファイル化し読み込むことも考えている。

8. おわりに

本研究では、A-doc のファイル形式を定義し、SuperSQL の出力メディアに A-doc を追加した。これによってデータベースから SuperSQL のクエリ一つで Static-standad、Static-separate 両タイプの A-doc 本体の XML とドキュメントの HTML を一括で生成することが可能となった。

文　献

[1] SuperSQL: http://ssql.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] 遠山 元道, “ターゲットリストの拡張によるデータベース出版と概視の実現”, 信学技報, *Vol.93,No.152,P79-88,* 電子情報通信学会, 1993

[4] 中谷圭吾, 鈴木優, 川越恭二, “文書間類似度とキーワードを用いた Web リンク自動生成手法”, 日本データベース学会 *Letters Vola.4,No.1,pp.89-92*, 2005

[5] Oracle Corporation. XML SQL Utility (XSU): http://www.oracle.com

[6] W3C Web Site. XML representation of a relational database: http://www.w3.org/XML/RDB.html