データベースシステム 118－7
情 報 学 基 礎 54－7
（1999．5．17）

集約演算の構成子による構造化文書ビューに対する問合せの最適化手法

加藤弘之† 吉川正俊§

†学術情報センター
§奈良先端科学技術大学院大学 情報科学研究科

データベースビューとして構築された構造化文書に対する問合せ処理の最適化手法を提案する．この手法の特徴は，全文索引を用いることで効率的に評価できる構造化文書固有の選択条件式に着目し，データベースの分野における伝統的な最適化手法である pushing selection をこの選択条件式に適用した点にある．特に，構造化文書データに固有の集約演算の特殊性に着目し，この集約によって定義された構造化文書ビューに対する問合せ処理において，この選択条件がどのように push されるかを示す．

A Query Optimization for Structured Document Views Constructed by Aggregations

Hiroyuki KATO† and Masatoshi YOSHIKAWA§

†National Center for Science Information Systems(NACSIS)
§Graduate School of Information Science, Nara Institute of Science and Technology (NAIST)

We propose a optimization method for queries to structured documents defined as database views. This method is characterized by applying *pushing selection*, well known as a traditional optimization method in database area, into a selection peculiar to retrieving structured documents. We focus on, especially, a aggregation over structured documents. We show how pushing the selection over queries to structured document views constructed by the aggregation.

1. はじめに

近年，DBMS は管理対象を広げており構造化文書もその管理対象の一つとなっている．構造化文書を DBMS で管理する利点は，文書の作成，編集，更新など文書処理の様々な側面において，同時実行制御，アクセス制御，障害回復などのデータベース機能を適用できる点にある．このような理由により，構造化文書データベースに関する研究は重要である．

伝統的なデータベースビュー同様，構造化文書をデータベースビューとして構築できることの利点は，同じデータを様々な視点で利用者に提供できる点 [AHV95] 及び，概念的データ独立性が維持できる点 [Dat95] にある．このようなデータベースビューとしての構造化文書（以下，本稿では「構造化文書ビュー」と記述する．）の構築とは，データベースで管理されている同じソース文書から異なる内容と構造を有する構造化文書を構築することである．例えば，図1は，新聞記事が格納されているデータベース中の表 **article** から作成された内容と構造の異る二つのビュー **view1** と **view2** を示している．表 **articles** は属性 **i_d,r,h,b** から構成されており，それぞれ発行日，記者，見出し，本文を表しているものとする．**view1** は，見出しと本文を順番に連結したものに対して新たな論理構造**<article>**を割り当てたものを，属性 **a** とし，発行日と併せて提供している．また，**view2** は，発行日毎の見出しに対して，新たな論理構造**<h-by-d>**を割り当てたものを発行日と併せて提供している．

ビューとは問合せのことである [AHV95]．一般に，データベース問合せには，抽出と再構成の二つ機能が要求される．構造化文書の操作という文脈のもとで，再構成とは，所望の論理構造を有する結果文書の構築のことである．構造化文書を記述するための新たな言語に，XML(eXtensible Markup Language)[WWWC98] がある．この XML 文書を操作する問合せ言語の制定に関する活動が存在する [Wor98]．したがって，近い将来 XML 文書を操作するための宣言的な問合せ言語の標準が決定される [Abi99]．この活動において，XML 文書を操作する問合せに要求される再構成の機能の重要性を，Maier は “restructuring”,“combination”[Mai98]，Aberer らは “restructuring”，“combination”,“aggregation”[AFN98]，Ceri らは “reshape”[CCD+98] という用語で述べている．

構造化文書固有の選択条件に，論理構造内の単語の出現に関するものがある．例えば先ほどの図1に示した新聞記事に対する選択条件である，「文字列（単語）’Clinton’ を含む**<article>**」は，その典型的な例である．予てより情報検索の分野では文書検索を効率的に行うための全文索引構造が提案されている [FBY92]．この全文索引を用いることで単語の出現に関する選択条件を安価なコストで評価することができる．更に，構造化文書の論理構造の出現も索引で管理することにより，論理構造内の単語の出現に関する選択条件を効率的に評価することが可能である [Bur92, CCB95, DSDT97]．

構造化文書ビューは，基底関係とは異なる論理構造を有する場合がある．これはビュー定義において，問合せ言語に要求される再構成の機能である新たな論理構造を定義する演算が適用されている場合である．このような場合，基底関係に対して構築されている論理構造を管理する索引にこの新たな論理構造が登録されていない．したがって，この新たな論理構造内の単語の出現に関する選択条件の評価は，索引を用いることができないので非効率である．例えば，先ほどの構造化文書ビュー **view2** に対する選択条件「文字列（単語）’Clinton’ を含む**<h-by-d>**」を評価するのに，論理構造を管理する索引に**<h-by-d>**が登録されていないので，索引を用いた評価ができないという問題点が生じる．

この問題を解決する方法には二つの候補が考えられる．1) 論理構造を管理している索引を再構成する．2) 問合せを書換える．このうち，全文索引の再構成は非常に高価な処理であることが知られている [山本 98] ので，ビュー定義によって新たに定義された構造を索引に反映させることは困難である．特に，ビューが仮想ビューとして定義されている場合は，問合せの評価と同時にビュー定義が評価されるので，問合せ処理中に索引を再構成することは現実的ではない．

本研究では，この問題がデータベースの分野で従来より用いられてきた pushing selection[Ram97] を適用することで解決されることを示す．本研究で提案する手法による利点には，次の二つがある．

- 基底関係上に構築されている既存の全文索引を用いて選択条件を効率的に評価できる
- 先に選択条件を評価することで問合せ処理における中間結果のデータ数を減少でき，これが問合せ全体の最適化につながる場合がある．

但し，一般的な pushing selection 同様，本手法もヒューリスティックなものである点に注意されたい．実際は，各問合せ処理候補のコストを見積もり，最も低いコストの候補が実行されるものである．本手法は，あくまでもこの候補の一つを提案するものである．

以下，2 節では本稿において展開する議論の準備として，XML の概要について述べた後で，XML 文書をデータベースで管理する手法に関して，文書データと全文索引の粒度についてそれぞれ述べる．また，この全文索引を用いることで，構造化文書検索固有のある選択条件が安価なコストで評価できることを示す．3 節では構造化文書ビューに対する問合せ処理機構を分類し，本稿で採用する問合せ処理機構の有効範囲を明確にする．4 節では，構造化文書データに対する集約演算の必要性とその特徴について

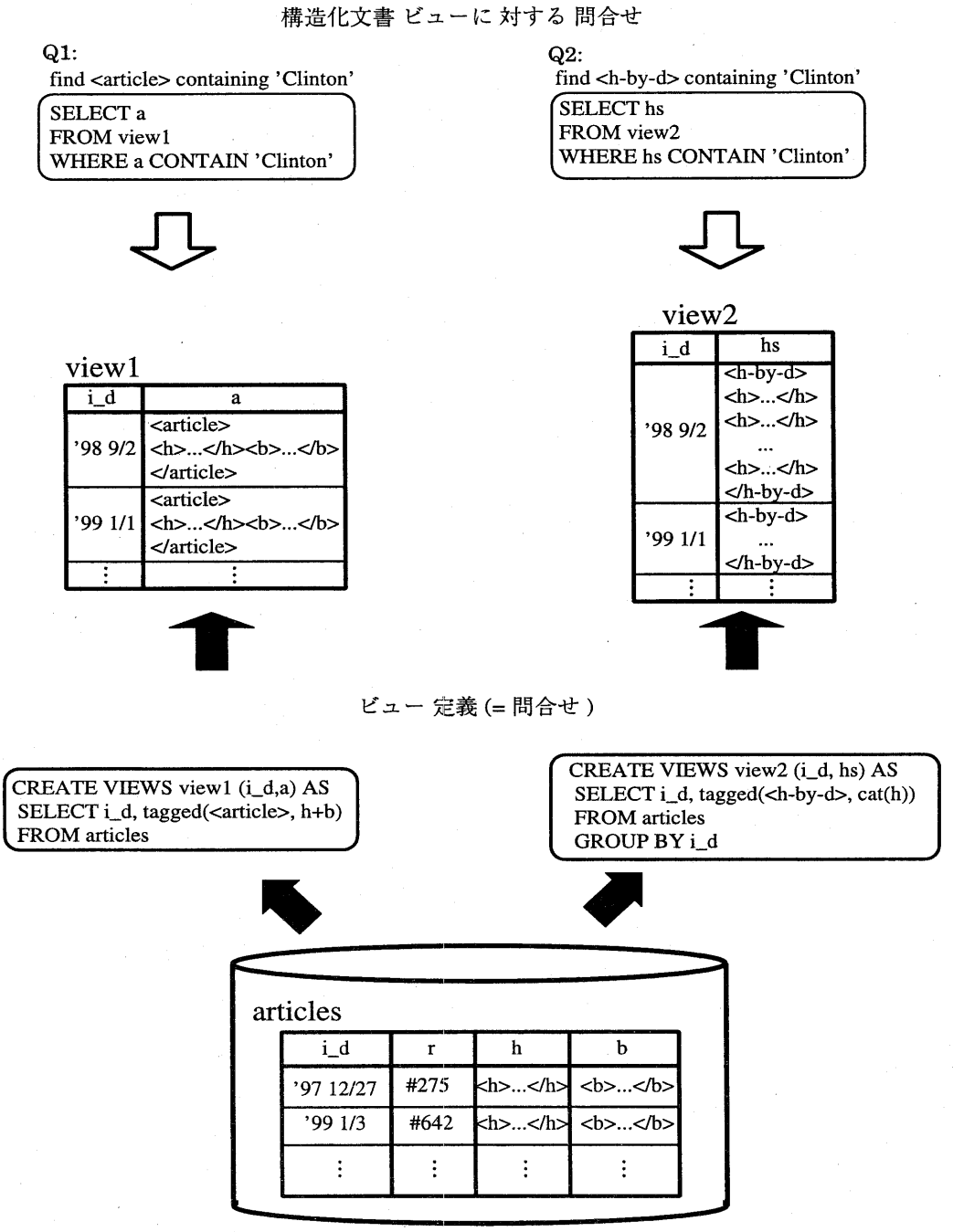

図 1 新聞記事に関する構造化文書ビューの例

述べる．5節では，集約によって定義されている構造化文書ビューに対する問合せのうち構造化文書検索固有の選択条件をpushすることによる問合せ処理の最適化手法を提案する．6節はまとめと今後の課題である．

尚，本稿の目的はXML文書を操作するための問合せ言語の構文を提案することではない．本稿で用いている問合せの構文を，いずれ登場してくる問合せ言語の標準へ変換することは容易なものであると信じる．

2. 準備

2.1 XMLの概要

XMLは，W3C(World Wide Web Consortium) のもとで開発された構造化データを記述するための言語である．XMLは**SGML**[ISO86, JIS92, JIS98a] のきわめて簡単な方言である．SGMLが文書を対象として開発された経緯により，XMLで記述されたオブジェクトは，XML文書と呼ばれるが，文書だけではなく，内部に論理構造を有するデータ全てがXMLの記述対象である．XMLの目標は，現在のHTMLと同様に，汎用的なSGMLがWeb上で配布，受信，処理できるようにすることである [JIS98b]．XML文書はエレメント[1]と呼ばれるデータの論理構造を，開始タグと終了タグを明示的に文書中に埋め込むことで表現したデータオブジェクトである．

文書型定義 (Document Type Definition, DTD) は，文書中のエレメントの出現の順序を規定したものであり，本質的な役割は文書の論理構造に関する文法とみなすことができる．エレメントはDTD中の**エレメント宣言**によって定義される．XML文書中のあるエレメント A がエレメント Bに含まれており，かつ Bに含まれる他のエレメントには含まれていないとき，A を Bの**子エレメント**と呼ぶ．一つのエレメント宣言は，エレメント名とその**内容モデル (content model)** から構成されている．内容モデルは，子エレメント名と5種類の演算子 (“,”, “*”, “+”, “|”, “?”) を用いた正規表現であり，各演算子の意味は以下の通りである．但し，r, r_1, r_2はエレメント名を表し，ϵは空列を表すものとする．

- “r_1,r_2” は，r_1と r_2の列，すなわち結合を表す．
- “r_1 | r_2” は，r_1または r_2を表す．
- “r*” は，rの0回以上の繰り返しを表す．
- “r+” は，“r,r*’を表す．
- “r?” は，“$r|\epsilon$” を表す．

DTDにおいてEMPTYキーワードを用いて宣言されたエレメントは，そのエレメントが文書中に現れたとき空でなければならない．空エレメントは開始タグがそのエレメントの全体を構成する．以下，データベースの世界における空値と区別するため，空エレメントの内容をEMPTYと記述する．

内容モデルには，**要素内容 (element content)** と**混在内容 (mixed content)** の2種類がある．要素内容は，文字データは含まず，子エレメントだけを必ず含み，子エレメント間は空白だけしか現れないような内容モデルである．これに対して，混在内容は，子エレメントに混在して文字データが含まれる可能性のある内容モデルである．

与えられたDTDに対して，そのDTDに従っているXML文書を**妥当な (valid)** XML文書と呼ぶ．また，XMLではDTDを持たない，もしくはDTDに従わないXML文書の存在を許している．しかし，このXML文書のタグの出現がXMLの文法（例えば開始タグと終了タグの対が必ず存在するなど）に従っている必要がある．このようなXML文書を**整形式の (well-formed)** XML文書と呼ぶ．各エレメントには，属性を定義することができる．これはDTD中の属性宣言で定義する．

あるエレメントのXML属性をそのエレメントの子エレメントとみなし，混在内容に関しては疑似エレメント [DD94] を導入することで，全てのXML文書を要素内容モデルから構成されているものとすることができる．そこで，本研究では要素内容モデルから構成されているXML文書を研究の対象とし，XML属性及び混在内容については取り扱わない．

2.2 XML文書のデータベースによる管理

この節では，XML文書のデータベース内部表現について記述する．

2.2.1 XML文書の分割の粒度

文書をデータベースに格納する際に分解する粒度に関しては，i) 各エレメント毎に分解するか，ii) 全く分解しないか，iii) 部分的に分解するかの三つの候補がある．本研究では，[BAK95] における実験結果を踏まえて，応用からの要請と再利用の単位に考慮して，部分的に分解する手法を採用する．

2.2.2 全文索引の粒度

データベースの分野における索引と情報検索の分野における索引とでは，その粒度が異る．データベースの分野では，データベースで管理されているスキーマ構成要素 (表，またはクラス) の一つ以上の属性に対して一つの索引が構築される．索引が構築されている属性に関する問合わせは，その索引を用いることによって効率的に処理される．これに対して情報検索の分野では，通常，管理されている文書集合全体に対して一つの全文索引を構築する [FBY92]．全文索引には転置索引を B^+木で構築したり，PAT木，PAT配列，そしてシグニチャファイルなど様々なデータ構造が既に提案されている．構造化文書に対するこれら全文索引の特徴の一つは，単語の出現を位置情報 (先頭から何単

[1] JIS規格 [JIS98b] では，「要素」と訳されているが，集合の要素などと区別するために，本論文では「エレメント」と記述することにする．

語目かという整数値) で管理し，エレメントの出現をリージョン (始まり位置と終り位置の有理数の組) で管理することで，いくつかの文書固有の選択条件を索引だけを用いることで，実データへのアクセスなしに評価することが可能である [Bur92, CCB95, DSDT97]．ここでいう文書固有の選択条件とは，単語の近接出現の特定，エレメント内の単語の出現の特定や，エレメント間の順序関係や包含関係の特定ことである．

本研究では，情報検索の分野における全文索引の粒度を採用する．すなわち，データベースで管理されている構造化文書データ集合全体に対して，一つの索引を構築するものとする．これにより，従来のデータベース索引の構築粒度に比べて，索引の格納コストのみならず検索コストに関しても優位となる．

3. 問合わせ処理機構の分類

この節では，構造化文書ビューに対する問合せ処理機構を分類し，本研究で採用する問合せ処理機構の有効範囲を明確にする．この分類のために以下に示す三つの分類軸を導入する．

まず，問合せの対象となる構造化文書ビューの構築方法に関して次のような分類軸を導入することができる．

- 構造化文書ビューが仮想ビューとして定義されているか，具現ビューとして構築されているか．
  - (1) 仮想ビューとして定義されている．すなわち，ビューに対する問合せを処理するときに同時にビュー定義を評価する．
  - (2) 具現ビューとして定義されている．すなわち，ビューを定義したときにビューの評価を行う．ビューに対する問合せは，索引を用いないのであれば，この具現ビュー上で評価される．

次に，サーバ・クライアント環境を仮定したとき次のような分類軸を導入することができる．

- 問合せをサーバ側で処理するか，クライアント側で処理するか．
  - (i) サーバ側だけで問合せ処理を行う．
  - (ii) クライアント側だけで問合せ処理を行う．
  - (iii) サーバ側とクライアント側の双方で協調して問合せ処理を行う．

最後に，構造化文書ビューにおいては，一般に基底関係における論理構造とは異なる論理構造を有している場合があるので，ビューの論理構造に即して索引を再構成するか，伝統的な仮想ビューに対する問合せ処理同様，問合せ中のビュー名をビュー定義で展開するかの軸を導入することができる．

- 索引を再構成するか，問合せを書換えるか．
  - (a) 索引を再構成する．
  - (b) 問合せを書換える．

上記の分類の次元の組合せにおいて，現実的でない組合せとして仮想ビューのもとで，全文索引の再構成を行うという手法が挙げられる．なぜならば，一般に全文索引の再構成は非常に高価な処理コストであることが知られている [山本 98] が，仮想ビューにおいてビュー定義の評価は問合せ処理時に同時になされるので，問合せ処理の最中に索引を再構成することは，現実的なパフォーマンスを得ることができないと思われるからである．

本稿では次節以降，次のような組合せを仮定して，構造化文書ビューに対する問合せ処理の最適化手法を提案する．

(1) 構造化文書ビューは仮想ビューとして定義されており，

(i) サーバ側だけで問合せ処理を行い，

(a) 問合せ中のビュー名をビュー定義に展開する．

尚，本稿で採用する問合せ処理機構は仮想ビューを仮定しているが，具現ビューを対象とした場合でも，問合せにおける論理構造内の単語の出現に関する選択条件の評価には，本手法が適用できる．

また，ネットワークを介した構造化文書ビューの配信を考えると，具現ビューの場合を考える必要がある．しかしながら，利用者は配信された文書を全て保存しないかもしれない．このような状況においては，本研究で採用する手法が有効である．

4. 構造化文書データを対象とする集約演算

この節では，構造化文書を対象とする集約について，その必要性と特徴及び，本稿で用いる集約演算の構文について記述する．但し，本稿の目的は，問合せ言語の提案ではない．[Wor98] における多くの position paper が述べているように，XML 文書を操作する問合せ言語には再構成の機能が必要であると我々も考える．再構成機能の中には集約機能もある．したがって，いずれ標準化されて現れてくる XML 文書を操作する問合せ言語にも，集約機能が採用されているものと，我々は信じる．我々の構文をこの標準化された言語の構文に変換することは，容易なことであると考える．

4.1 集約機能の必要性と特徴

データベース問合せ言語に要求される機能には，抽出と再構成がある．構造化文書を対象とする場合，再構成とは所望の論理構造を有する結果文書の構築にある．既に述べたように，

XML 文書の論理構造は DTD で定義されている．この DTD で宣言される論理構造の型は，正規表現の演算子を用いて定義されている．したがって，XML 文書を扱う問合せ言語に要求される再構成の機能として，DTD で用いられる正規表現の演算子で定義できる任意の論理構造の型を定義できることが挙げられる．本研究ではこれらの演算子のうち，特に繰り返し演算子 (“+”,“*”) に着目する．この繰り返し演算子によって定義される新たな論理構造の構築は，集約演算子によってなされるものである．例えば，図 1に示した新聞記事のビュー **view1** は，発行日毎の見出しの（順序付）集合に対して，新たなエレメント **<h-by-d>**を割り当てている．したがって，このエレメント**<h-by-d>**は，エレメント**<h>**の繰返しから構成されている．このように繰り返しを定義するためには集約演算が必要となる．本稿ではこの繰り返し演算子に相当する構成子である集約演算に着目する．次節以降，この集約演算によって定義されている構造化文書ビューに対する問合せの処理に焦点を当てる．

本研究で扱う構造化文書を対象とする集約演算と従来のデータベース問合せで用いられてきた集約演算との大きな違いは，集約の要素の保存性にある．例えば，数値型の値の集合の平均を計算する集約関数 Avg() の結果には，元の要素は保存されていない．したがって，定義中に従来型の集約が施されているビューに対して，利用者は元の要素に関する問合せはできない．これに対して，構造化文書を対象とする集約の結果には，元の要素が保存されている．したがって，このような集約が適用されて定義されている構造化文書ビューに 対して，利用者は元の要素の関する問合せを記述することができる．尚，このような集約の特徴は構造化文書だけでなく，内部に構造を有する型固有の集約に関する特徴とみなすことができる．

4.2 採用する集約の構文

既に，我々は XML 文書を操作する問合せ言語を提案している [KY98, 加藤 99]．この節では，以降の節で説明を容易にするために，特に再構成の機能のうちの集約演算の構文を説明する．

我々は，抽象データ型 (ADT) アプローチにより XML 文書を操作する問合せ言語を提案している [KY98, 加藤 99]．これにより，SQL 中にこの ADT（**XML** 型）に定義されている関数や述語を記述することで，データベース中の XML 文書を操作することが可能となる．以下，この **XML** 型に定義されている関数のうち，本稿で着目している再構成に関する機能を有する関数について説明をする．

尚，型に関して次の点を明記しておく．エレメント名に関する問合せができるように，文字列型の下位型[2]でエレメント名を値とする **ELEMENT** 型を導入する．**ELEMENT** 型の値のリテラルはエレメント名を ‘<’ と ‘>’ で囲むことで表されるものとする．また，**XML** 型の上位型として **ST_TEXT** 型を導入する．**ST_TEXT** 型には，**XML** 型の値から根エレメントを取り除いた値も所属できるものとする．最後に，nil と EMPTY の違いについて記述する．nil は空値を表すのに対して，EMPTY は ST_TEXT 型の値である．既に述べたように XML では内容が EMPTY であるエレメントを許している．したがって，EMPTY に根エレメントを付加した値は XML 型の値である．

- **XML tagged(ELEMENT** *arg1*, **ST_TEXT** *arg2*)
  ELEMENT 型の *arg1* を ST_TEXT 型の *arg2* の根エレメントに割り当てる．*arg2* が EMPTY の場合，EMPTY を要素とするエレメント *arg1* を根とする ST_TEXT 型の値が返る．*arg2* が nil の場合，この関数は nil を返す．
- **ST_TEXT cat*(set of ST_TEXT** *arg1*)(
  内容モデルの定義に用いられる繰り返し演算子 “*” に相当)
  入力された ST_TEXT 型の順序付き集合に対して，その集合の要素全てが指定された順序に応じて連結された一つの文書を出力する．但し，この関数は入力が nil の場合，結果は EMPTY とする．
- **ST_TEXT cat(set of ST_TEXT** *arg1*)(
  内容モデルの定義に用いられる繰り返し演算子 “+に相当)
  入力された ST_TEXT 型の順序付き集合に対して，その集合の要素全てが指定された順序に応じて連結された一つの ST_TEXT 型の値を出力する．但し，この関数は入力が nil の場合，結果も nil とする．
- *arg1* **CONTAIN** STRING *arg2*
  はテキスト *arg1* 中に文字列 *arg2* が出現する時かつその時に限り真を返す述語である．

5. 構造化文書ビューに対する問合せ処理

この節では，前節で述べた構造化文書固有の集約演算によって定義されている構造化文書ビューに対する，問合せ処理の最適化手法について提案する．この最適化手法の特徴は，構造化文書検索固有の選択条件に着目し，この選択条件にたいして伝統的な最適化手法である pushing selections[Ram97] を適用する点にある．2 節で述べたように，伝統的な全文索引技術を用いることでエレメント内の単語の出現に関する選択条件を効率的に評価することが可能である．本稿ではこの選択条件に着目する．すなわち，前節で述べた述語**CONTAINS** に着目し，この述語を pushing する際の問合せの書換えを示す．

図 1に示した構造化文書ビュー **view1** に対する問合せ例 **Q2** を用いてこの問合せの最適化手法を説明する．まず，この問合せをナイーブ

[2]ここでいう下位型関連は定義域包含関係に基づくものとする．

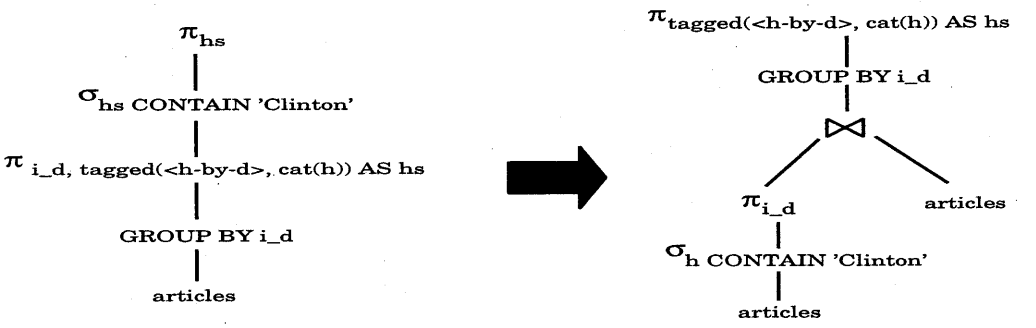

図 2 問合せ例 **Q2** の pushing-selection による書換え

に処理した場合と，本稿で提案する手法で処理した場合とを比較する．一般に，仮想ビューに対する問合せは問合わせ中のビュー名の出現をビュー定義に展開することで処理される．この問合せ **Q2** において `view1` のビュー定義を展開した問合せの木表現を図2の左側に示す．この問合せをこのままナイーブに処理する場合，次のような点で効率が良くない．

- 表 `article` 中の全てのデータに対して，ビュー `view1` の定義を評価した後で，選択条件を評価する．
- 選択条件式 `hs CONTAIN 'Clinton'` は，基底関係には存在しないエレメントに関する単語の出現に関するものあるので，この選択条件の評価にサーバ側に既に存在する全文索引を用いることができない．

したがって，この選択条件式を push することによる問合せの書換えを考える．この書換えによる利点は，次の点にある．

- 先に選択条件式を評価することで，問合せ処理中の中間結果のデータ量を減すことができる．
- 選択条件式を基底関係に対するものに書換えることで，その評価に全文索引を用いることができ，安価なコストで評価できる．

図2の右側の問合せ木は，問合せ例 **Q2** を本稿で提案する手法を用いて書換えた結果を示している．

尚，この問合せの書換えの一般的な場合を図3に示す．

6. 議論

本研究では，抽象データ型アプローチによる問合せ構文を採用した．このアプローチの最大の利点は，モジュール性と拡張可能性にある [Sto96, CD96]．すなわち，各抽象データ型の追加や削除を，データベースの他の部分に影響を与えることなく逐次的に達成できる．

しかしながら，本稿で述べた最適化の手法は関係 ADT と XML ADT との相互作用により達成されるものである．この相互作用は関係 ADT と XML ADT 間の独立性を維持するためには，相互作用は障害となる．しかし，より一層の最適化を考えると相互作用が必要となる．

7. まとめと今後の課題

本稿では，集約演算が適用された構造化文書ビューに対する問合せの処理において，pushing-selection を適用した問合せの書換えによる最適化手法を提案した．既存の全文索引を用いることができるだけでなく，

今後は索引の再構成による問合せ処理手法について研究を進める予定である．

参考文献

[Abi99] Serge Abiteboul. On views and xml. In *Proc. ACM Symp. on Principles of Database Systems*, May 1999.

[AFN98] Karl Aberer, Peter Fankhauser, and Erich Neuhold. XML is the data model, what is the function of the query language? In *Position papers for W3C Query Language Workshop*. 1998.

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[BAK95] Klemens Boehm, Karl Aberer, and Wolfgang Klas. Building a configurable database application for structured documents. *Multimedia - Tools and Applications*, 1995.

[Bur92] Forbes J. Burkowski. An algebra for hierarchically organized text-dominated databases. *Information Processing & Management*, Vol. 28, No. 3, pp. 333–348, 1992.

[CCB95] Charlie L. A. Clarke, Gordon V. Cormack, and Forbes J. Burkowski. An Algebra for Structured Text Search and A Framework for its Implementation. *The Computer Journal*, Vol. 38, No. 1, pp. 43–56, 1995.

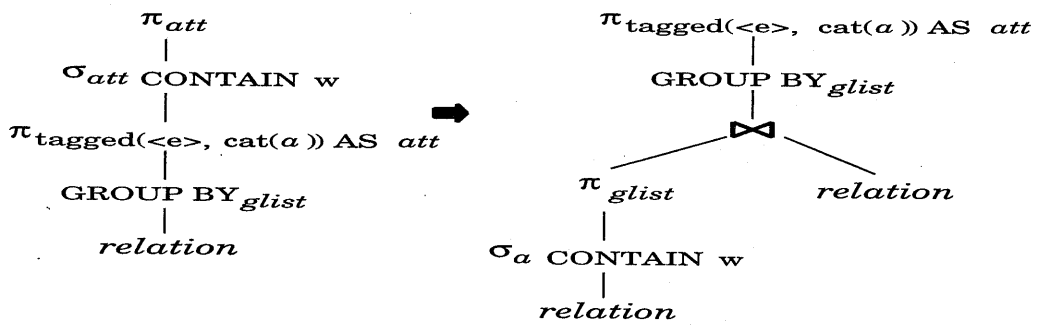

図 3 集約による構造化文書ビューに対する問合せの書換え規則

[CCD+98] Stefano Ceri, Sara Comai, Ernesto Damiani, Piero Fraternali, Stefano Paraboschi, and Letizia Tanca. XML-GL: A Graphical Language for Querying and Reshaping XML Documents. In *Position papers for W3C Query Language Workshop*. 1998.

[CD96] Michael J. Carey and David J. DeWitt. Of objects and databases: A decade of turmoil. In *Proc. of the 22nd International Conference on Very Large Data Bases (VLDB)*, pp. 3–14, Bombay, September 1996.

[Dat95] C. J. Date. *An Introduction To Database Systems*. Addison-Wesley publishing company, 6th edition, 1995.

[DD94] Steven J. DeRose and David G. Durand. *MAKING HYPERMEDIA WORK -A User's Guide to HyTime-*. KLUWER ACADEMIC PUBLISHERS, 1994.

[DSDT97] Tuong Dao, Ron Sacks-Davis, and James A. Thom. An indexing scheme for structured documents and its implementation. In *Proc. of the 5th International Conference on Database Systems for Advanced Applications (DASFAA'97)*, April 1997.

[FBY92] William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval – Data Structures & Algorithms –*. Prentice-Hall, 1992.

[ISO86] ISO 8879: 1986. *Information Processing – Text and Office System – Standard Generalized Markup Language (SGML)*, Oct. 15 1986.

[JIS92] JIS X 4151:1992 文書記述言語 SGML (Standard Generalized Markup Language), 日本規格協会, 1992.

[JIS98a] JIS X 4151:1998 文書記述言語 SGML (Standard Generalized Markup Language)(追補 1), 日本規格協会, 1998.

[JIS98b] TR X 0008:1998 拡張可能なマーク付け言語 XML (eXtensible Markup Language), 日本規格協会, 1998.

[KY98] Hiroyuki Kato and Masatoshi Yoshikawa. Constructing structured document views. In *International Workshop on New Database Technologies for Collaborative Work Support and Spatio-Temporal Data Management (NewDB'98) (to appear In Lecture Notes in Computer Science(LNCS), Springer-Verlag.)*, pp. 440–447, November 1998.

[Mai98] David Maier. Database Desiderata for an XML Query Language. In *Position papers for W3C Query Language Workshop*. 1998.

[Ram97] Raghu Ramakrishnan. *Database Management Systems*. McGraw-Hill, 1997.

[Sto96] Michael Stonebraker. *OBJECT-RELATIONAL DBMSs-THE NEXT GREAT WAVE*. Morgan Kaufmann, 1996.

[Wor98] World Wide Web Consortium. QL'98 - The Query Languages Workshop. http://www.w3.org/TandS/QL/QL98/, December 1998.

[WWWC98] World Wide Web Consortium. Extensible Markup Language (XML) 1.0. http://www.w3.org/TR/1998/REC-xml-19980210, February 1998. W3C Recommendation 10-February-1998.

[加藤 99] 加藤弘之, 吉川正俊. オブジェクトリンクを有する構造化文書を管理するデータベースのモデルと問合せ. 電子情報通信学会論文誌 (D-I), Vol. J82-D-I, No. 1, January 1999.

[山本 98] 山本毅雄, 橋爪宏達, 神門典子, 清水美都子. 全検索 技術と応用 学術情報センター編. 丸善株式会社, 1998.