効率的な更新が可能な構造化文書索引手法*

1Y－1

金本 博隆　加藤 弘之　絹谷 弘子　吉川 正俊†
奈良先端科学技術大学院大学　情報科学研究科‡

1. はじめに

電子化文書の共有や再利用の観点からSGML[2)3)]に代表される構造化文書が利用されつつある．また，SGML文書をWeb環境で利用することを考慮した文書記述言語XML[6)]の仕様の策定作業が最終段階に入った．XMLが普及すると，構造化文書がさらに広く利用されるものと考える．

これまで，文字列の更新を考慮した構造化文書の索引手法の研究はあまりされていない．そこで本研究は，効率的な更新が可能な構造化文書の索引手法の提案を目的とする．本稿ではStructure Index[7)]の提案をもとに，XML文書インスタンス[1]を対象とした索引のディスクへの格納手法について議論する．

2. 提案する索引手法

2.1 Structure Index

Structure Indexは，XML文書インスタンスの階層構造に関する問合せに適した木構造の索引である（図1）．

Structure Indexは次の条件を満たす．

1. 根は，XML文書インスタンスの最初に出現するelementに対応する．
2. 節はelementに対応する．節の子は，節に対応するelementの子にあたるelementまたは文字列である．子はXML文書インスタンス中の出現順に並ぶ．
3. 節や葉は，elementまたは文字列ごとにelement名，elementに含まれる文字数，elementまたは文字列の出現位置（XML文書インスタンスが格納されているディスクのページ番号とスロット番号）を管理する．

2.2 Content Index

Structure Indexの検索能力を補うためにCotent Indexを用いる．Content Indexはelementと文字列に関する問合せに適した転置索引である．検索対象（XML文書インスタンス中の部分文字列，element名，属性名，属性値）に対応した四つの辞書と，それぞれの辞書に対応したPosting Listからなる．四つの辞書は出現回数を管理し，Posting ListはXML文書インスタンス中の出現位置を管理する．

3. Structure IndexとXML文書インスタンスのディスクへの格納

3.1 Structure Index

Structure Indexの効率的な更新を目指すため，索引階層の最も深い葉までのディスクアクセスが少なくなるよう節や葉をまとめてページに格納する．

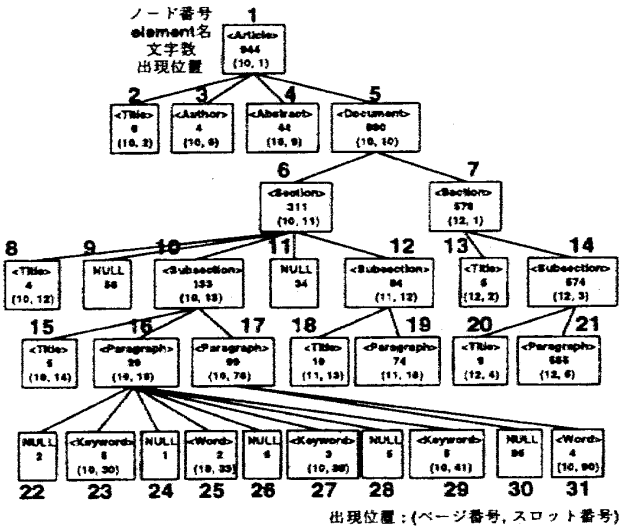

図1 Structure Index

Structure Indexのディスクページへの分割アルゴリズムは次の通りである（1ページにm個の葉または節が格納可能で，節nの子孫の数を$d(n)$，nの親をn_pとする）．

[Step 1] 索引階層の高さhを求める．h - 1階層の葉または節の集合をN_{h-1}とする．

[Step 2] N_{h-1}の中で，最大の$d(n)$を持つ節nを見つける．nが複数存在する場合は，XML文書インスタンス中に最初に出現する節を見つける．

[Step 3] n, $d(n)$, $d(n_p)$について，

1. $m \leq d(n)$ならば，nの子孫を複数のページに分割して格納する．[Step 4]に行く．
2. $d(n) \leq m \leq d(n_p)$ならば，nの子孫をディスクに格納する．[Step 4]に行く．
3. $d(n) \leq d(n_p) \leq m$ならば，n_pをnとして[Step 3]に行く．

[Step 4] ページに格納した葉または節を削除する．

[Step 5] 索引に葉または節が存在するならば[Step 1]に行く．節や葉が存在しないならば終了．

このアルゴリズムを使うと，図1について$m = 10$とした場合，ノード番号{1-5}，{6, 7, 13, 14, 20, 21}{8-12, 18, 19}，{15-17, 30, 31}，{22-29}の5グループに分割できる．索引の根から最大4回のディスクアクセスによりすべての節や葉に到達可能である．

出現頻度の高いelementをStructure Indexで管理すると，索引自身のサイズが大きくなる可能性がある．その場合は出現頻度の高いelementを索引で管理しないで，XML文書インスタンスが格納されているペー

*An Efficiently Updatable Index Scheme for Structured Documents.

†Hirotaka KANEMOTO, Hiroyuki KATO, Hiroko KINUTANI and Masatoshi YOSHIKAWA

‡Graduate School of Information Science, Nara Institute of Science and Technology (NAIST)

[1]SGML文書インスタンスでも同様の議論が可能である．

表 1 索引手法の比較

	既存の索引手法		提案する索引手法
	Position-Based Index	Path-Based Index	Structure Index & Content Index
問合せ能力			
・文字列	○	△	○
・構造	○[3]	○	○
・属性	−	−	○
更新コスト	$O(r)$	$O(h)$	$O(h)$
問合せコスト	$O(1)$	$O(h)$	$O(h)$

r: 索引づけされた部分文字列の数.
h: 索引の階層構造の高さ.

ジを直接探索する方法が考えられる. さらに, XML 文書インスタンスごとに Structure Index の管理する element の種類を動的に変化させることも考えられる.

3.2 XML 文書インスタンス

構文解析した XML 文書インスタンスは, タグつきの文字列が格納されているデータ部と, 文字列を指している Slot Directry 部から構成されるページを単位として格納される. Slot Directory はページの先頭からの文字数を管理しており, XML 文書インスタンスは索引から Slot Directory を通して参照される.

4. 既存の索引手法と提案した索引手法との比較

第 2 章で提案した索引手法と, 既存の索引手法 (Position-Based Index と Path-Based Index)[2] を比較した (表 1).

ただしこの比較は次の条件のもとで行った.

1. 一つの部分文字列に関する索引の更新コストと問合せコストについて比較した.
2. Content Index と Position-Based Index はハッシュ法を用いることとする.

既存の索引手法について, Path-Based Index は索引の複数の葉をまたぐ近接問合せが不可能である. Position-Based Index は索引の更新コストが高い. 今回提案する索引手法の更新コストや問合せコストは $O(h)$(h: Structure Index の階層構造の高さ) であり, これらのコストは低いといえる.

5. 実験システムのアーキテクチャ

実験システムのアーキテクチャは図 2の通りである.

Index Generator は XML 文書インスタンスの構文解析機能, Structure Index と Content Index の生成機能を持つ. 構文解析された XML 文書インスタンスや Structure Index, Content Index は Shore[5] リポジトリに格納する. Query Processor & Index Controller は, XML 文書インスタンスや索引の更新, XML 文書インスタンスへの問合せ機能を持つ.

6. まとめと今後の課題

Structure Index の性質を述べ, 索引と XML 文書インスタンスのディスクへの格納手法について議論した. 今後の課題として, element の出現頻度を用いる

[2]属性に関する問合せを考慮しない分類法である[4].

[3]SC-lists モデル[1]等を利用することにより間接的に求めることが可能.

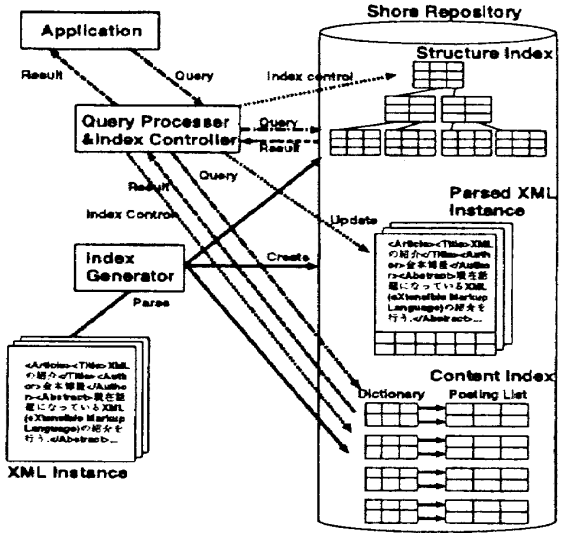

図 2 実験システムのアーキテクチャ

Structure Index の動的な生成手法の考察と, 実験システムの実装と評価が挙げられる.

謝辞 貴重で有益な意見や助言を与えて頂いた植村研究室の皆様に厚くお礼を申し上げます.

参 考 文 献

1) 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.

2) ISO 8879: 1986. *Information Processing - Text and Office System - Standard Generalized Markup Language (SGML)*, Oct. 15 1986.

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

4) Ron Sacks-Davis, Tuong Dao, James A. Thom, and Justin Zobel. Indexing documents for queries on structure, content and attributes. In *International Symposium on Digital Media Information Base(DMIB'97)*, Nov. 1997.

5) University of Wisconsin. Shore project home page. http://www.cs.wisc.edu/shore/, Mar 1996.

6) World Wide Web Consortium. Extensible Markup Language 1.0 (XML 1.0). Proposed Recommendation, http://www.w3.org/TR/PR-xml-971208, December 1997.

7) 金本博隆, 加藤弘之, 絹谷弘子, 吉川正俊. 効率的な更新が可能な構造化文書の索引. 情報処理学会第 114 回データベースシステム研究会研究報告, January 1998.