

XML 文書のアクセス制御における ロール階層を考慮したルール簡略化 とルール変更操作の局所化

Simplifying Access Control Rules and Localizing Impact of Updates for XML Documents and Role Hierarchy

牛場 祐貴[▼]

岩井原 瑞穂[◆]

吉川 正俊[▲]

Yuki USHIBA

Mizuho IWAIHARA

Masatoshi YOSHIKAWA

XML のアクセス制御において、ロールベースドアクセス制御を行う際に、XML 文書の木構造だけでなくロール間の関係から構成されるロール階層の構造も利用したルールの削減を行う手法を提案する。提案手法では、XML 文書内の部分木の包含関係とロール階層内の到達可能集合の包含関係を利用し、求めるアクセス制御情報を実現するために必要なアクセス制限ルール数を削減する。また提案手法によって更新の範囲を局所的に押さえた結果、更新コストを削減できることがわかった。

We adopt Role-Based Access Control(RBAC) in order to use instead of user group and use role hierarchy as relation of roles. Role hierarchy as well as XML Document tree is useful for reducing redundant rules. We define and use DR-Region to reduce the space for rules and to localize impact of updates.

1. はじめに

顧客情報など秘匿性の高い文書を XML 文書形式で保存することが多くなり、XML 文書や XML 文書が存在するディレクトリや URI に対するアクセス権限を適切に管理するアクセス制御は情報保護において重要となりつつある。

大きな企業では数万を越すユーザが存在することがあり、またアクセス制御対象オブジェクトの文書も数千万を越え、管理コスト以外にも情報保護や内部統制の面からこのような大規模なアクセス制御の効率化を行う必要がある。また生成され、蓄積され

ているアクセス制御ルール集合は、ルール生成時のアクセス権限設定情報を保持し参照されるだけではなく、アクセス権限設定情報の変更に伴い適切に変更される必要がある。このようにルール集合を効率の良い形式で管理することは、管理コスト全体からみて重要な意味を持つてくる。そのためルールの冗長性を削減するとともに、ルール集合への更新操作に必要なコストを抑えることが可能な手法を提案する。

以下、本稿は 2 章において関連研究について述べ、3 章で DR 領域とアクセス制御ルールとしてのアクセス制御ラベルとその集合の簡略化について述べる。4 章において権限判定の方法と更新の方法について述べた後、コストを評価する。5 章で同一のアクセス権限設定情報から既存手法、提案手法によって作成されたルール集合に対し更新操作を行い定量的評価をする。6 章はまとめと今後の課題である。

2. 関連研究

2.1 XML 文書におけるアクセス制御

XML 文書は開始タグと終了タグで囲まれた入れ子構造を持ち、要素をノードとする木構造として文書表現できる(図 1)。さらに XML 文書が置かれているディレクトリ構造や URI を含めた木構造と考えることができるため、これ以降はそのように木構造をなしている XML 文書集合をアクセス制御対象としても考えることにする。

きめ細かいアクセス制御は XML 文書が存在するディレクトリや URI を含めた XML 文書の構造・内容に応じて直接的にアクセスの制約をかけるため、アクセス制御ルールは XML 文書の構造・内容に深くかかわってくる。文献 [1] では XML 文書におけるアクセス権限設定が親子間で同じアクセス権限設定を持つ傾向にあるという構造的な局所性に注目し、アクセス権限設定が XML 文書の階層構造によく一致するとき、より少ないラベル数でアクセス制御情報を表現できる特徴を持つ手法が示されている。また文献 [2] では文献 [1] で使用されるラベルを統合することにより、総ラベル数を削減を行う手法が示されている。

2.2 ロールベースドアクセス制御

アクセス制御の主体であるユーザ集合を特定の役割に応じてロールとしてまとめ、ロール間の継承関係を組織構造等に基づいてロール階層を構築することができる。アクセス制御対象のオブジェクトとユーザの間のマッピングにロール階層をささむことにより、オブジェクトとロールのマッピングおよびロールとユーザのマッピングに分割できる。これにより、アクセス制御ルールの管理コストを削減することができるロールベースドアクセス制御(RBAC)[4]は標準化が行われ普及が進んでいる。

3. 提案手法

XML 文書におけるアクセス制御ルールの簡略化に関する既存手法はロール階層に適用することは以下の理由から困難である。

- XML 文書のみ適用することを想定しているか [1]、または操作など固定された構造に対して適用することを想定して

[▼] 学生会員 京都大学大学院情報学研究科
y.ushiba@db.soc.i.kyoto-u.ac.jp

[◆] 正会員 京都大学大学院情報学研究科
iwaihara@i.kyoto-u.ac.jp

[▲] 正会員 京都大学大学院情報学研究科
yoshikawa@i.kyoto-u.ac.jp

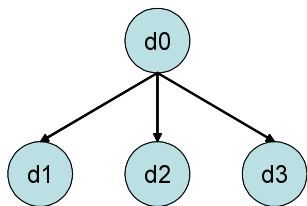


図1 XML文書木

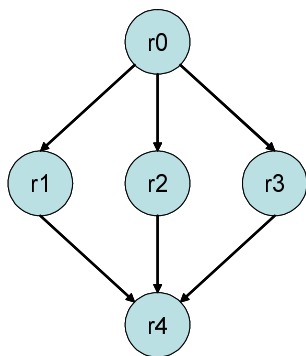


図2 ロール階層

おり [2], ロール数が数千に及ぶようなロール階層にはそのまま適用できず, またロール階層での継承等が考慮されていない。

- 制約条件があることにより, ロール階層にそのまま適用することはできない。

そこで XML 文書の木構造とロール階層構造を考慮した新たなラベル設定を提案する。

以下では, まずラベルを設定する DR 領域の定義を行った後に, DR 領域の包含関係により構成される DR 束について述べる。そしてアクセス権情報設定の手法であるラベル付けアルゴリズム, 簡略化の手法である冗長なラベルの除去について述べる。

3.1 DR 領域

提案手法は CAM のようにアクセス権設定が XML 文書の階層構造だけでなく, ロール階層の改造構造にもよく一致するときに, より少ないルール数でアクセス制御情報を表現できる特徴を持つ。まず階層構造における適用範囲を定める関数 DoS/AoS を定義したのち, アクセス制御ラベルを設定する DR 領域の定義を行う。

定義 1 ($DoS(descendant_or_self)/AoS(ancestor_or_self)$) 非巡回有向グラフ $G(V, E)$ が与えられたとき, それぞれ以下のように定義される関数である。 $DoS(n)$ は n からの有向枝の向きに沿った有向路が存在するノードの集合を返す。以下の式で定義される。

$$DoS(n) = \{n\} \cup \bigcup_{c \in V \wedge (n,c) \in E} DoS(c)$$

$AoS(n)$ は n への有向枝の向きに沿った有向路が存在するノードの集合を返す。以下の式で定義される。

$$AoS(n) = \{n\} \cup \bigcup_{p \in V \wedge (p,n) \in E} AoS(p)$$

補題 1 (有向路の存在と DoS の包含関係) n_1 から n_2 への有向路が存在することと $DoS(n_1)$ と $DoS(n_2)$ の間には包含関係が

存在することは同値。すなわち次の式が成り立つ。

$$n_2 \in DoS(n_1) \Leftrightarrow DoS(n_2) \subseteq DoS(n_1) \quad (1)$$

証明 1 n_1 から n_2 への有向路が存在するとき, n_2 からの有向路が存在する任意のノード n_3 に対して, n_1 から n_2 を通り n_3 へ至るような有向路が存在するため, $DoS(n_1)$ は $DoS(n_2)$ の要素全てを含む。逆に $DoS(n_1)$ と $DoS(n_2)$ が包含関係にあるとき, $DoS(n_2)$ の要素である n_2 は $DoS(n_1)$ の要素でもある。すなわち DoS の定義から n_1 から n_2 への有向路が存在することを示している。

補題 2 (有向路の存在と AoS の包含関係) n_2 から n_1 への有向路が存在することと $AoS(n_1)$ と $AoS(n_2)$ の間には包含関係が存在することは同値。

証明 2 DoS と同様である。

定義 2 (cas(cascade)) 非巡回有向グラフ $G(V, E)$ とノード $n \in V$ に対して, $\{‘u’, ‘d’\}$ の 2 値で設定され, それぞれ ‘u’ は $AoS(n)$, ‘d’ は $DoS(n)$ を用いることを意味する。

定義 3 (DR 領域) XML 文書内の文書ノード d の $DoS(d)$ とロール階層内のロール r からの cas に基づく $DoS(r)/AoS(r)$ にそれぞれ所属する任意の文書ノードとロールの対の集合を DR 領域と呼ぶ。またこの基点となる d と r の対を代表対と呼び, 代表対と cas の組から DR 領域への関数を以下の式で定義する。

$$DR(d, r, cas) = \begin{cases} DoS(d) \times DoS(r), (cas = ‘d’) \\ DoS(d) \times AoS(r), (cas = ‘u’) \end{cases}$$

定理 1 (有向路の存在と DR 領域の包含関係) 文書ノード d_1, d_2 とロール r_1, r_2 に関して, d_1 と d_2 の間に有向路が存在し r_1 と r_2 の間に有向路が存在することと DR 領域 $DR(d_1, r_1, cas), DR(d_2, r_2, cas)$ の間に包含関係が成り立つことは同値である。すなわち以下の式が成り立つ。

$$d_1 \in DoS(d_2) \wedge r_1 \in DoS(r_2) \Leftrightarrow DR(d_1, r_1, ‘d’) \subseteq DR(d_2, r_2, ‘d’) \quad (2)$$

$$d_1 \in DoS(d_2) \wedge r_1 \in AoS(r_2) \Leftrightarrow DR(d_1, r_1, ‘u’) \subseteq DR(d_2, r_2, ‘u’) \quad (3)$$

証明 3 XML 文書木とロール階層それぞれに関する補題 1 から (2) が導出される。同様に XML 文書に関する補題 1 とロール階層に関する補題 2 から (3) が導出される。逆に例えば互いに包含関係を持たない 2 つの文書ノード d_1, d_2 と同一のあるロールを代表対とする DR 領域を考えると, 互いに含まれていない文書ノードとロールの対が存在するため包含関係は存在しない。ゆえに DR 領域において包含関係が存在するのは (2) または (3) を満たすときのみである。

定理 2 DR 領域の集合は半順序をなす．すなわち以下の式が成り立つ．

(反射律)

$$DR(d_1, r_1, cas) \subseteq DR(d_1, r_1, cas)$$

(推移律)

$$\begin{aligned} &(DR(d_1, r_1, cas) \subseteq DR(d_2, r_2, cas)) \\ &\wedge (DR(d_2, r_2, cas) \subseteq DR(d_3, r_3, cas)) \\ &\Rightarrow DR(d_1, r_1, cas) \subseteq DR(d_3, r_3, cas) \end{aligned}$$

(反対称律)

$$\begin{aligned} &(DR(d_1, r_1, cas) \subseteq DR(d_2, r_2, cas)) \\ &\wedge (DR(d_2, r_2, cas) \subseteq DR(d_1, r_1, cas)) \\ &\Rightarrow DR(d_1, r_1, cas) = DR(d_2, r_2, cas) \end{aligned}$$

証明 4 cas が 'd' の場合を示す．

(反射律)

同一の文書ノード間に有向路が存在すると考えられ、同一のロール間にも有向路が存在すると考えられる．よって定理 1 により式は成り立つ．

(推移律)

前件より d_2 から d_1 への有向路が存在し、かつ d_3 から d_2 への有向路が存在する．すなわち d_3 から d_1 への有向路が存在する．また r_2 から r_1 への有向路が存在し、かつ r_3 から r_2 への有向路が存在する．すなわち r_3 から r_1 への有向路が存在する．よって定理 1 により式は成り立つ．

(反対称律)

前件より d_2 から d_1 への有向路が存在し、かつ d_1 から d_2 への有向路が存在する．また r_2 から r_1 への有向路が存在し、かつ r_1 から r_2 への有向路が存在する．よって $DoS(d_1) = DoS(d_2)$ かつ $DoS(r_1) = DoS(r_2)$ が成り立つ．ゆえに DR 領域の定義より 2 つの DR 領域は等しい．

cas が 'u' の場合も同様である．

定義 4 (DR 束) 反射的包含関係、推移的包含関係を除いた包含関係をもつ 2 つの DR 領域間に有向枝を張ることで構成する．このとき有向枝の始点を親 DR 領域、終点を子 DR 領域という．

3.2 DR 束の構成と冗長性除去

次に DR 領域に対して以下の順でアクセス制御情報であるアクセス制御ラベル (以下、単にラベルという) の設定と冗長なラベルの除去を行う．

1. DR 領域に (s, d) という形式のラベルを設定する
2. 文書構造における冗長なラベルに印を付ける
3. ラベルに r 部を追加設定する．
4. ロール階層における冗長なラベルを除去する

アクセス制御ラベルの設定 ラベルは (s, d) という形式で与え、設定方法は自身のアクセス制御情報がアクセス許可 (またはアクセス拒否) を与えられていれば s 部に $+(-)$ を設定し、自身

が $positive/negative/neutral$ であるとき d 部にそれぞれ $+/-/$ 親のラベルと同じものを設定する．

定義 5 (positive/negative/neutral) DR 領域 が $positive(negative)$ とはその DR 領域がアクセス許可 (アクセス拒否) の葉であるか、 $positive(negative)$ の子の数が、 $negative(positive)$ の子の数より多いこととする． $neutral$ とは $positive$ の子の数と $negative$ の子の数が等しいこととする．

文書構造における冗長なラベル 文書構造における冗長なラベルとは同じロールを代表対に持つ DR 束内の他の DR 領域から包摂されるラベルであり、以下で定義される． DR 束を親から順に冗長なラベルであるか判定し、もし冗長なラベルであれば自身に印をつける．

定義 6 (文書構造における subsumed label) DR 束内において自身のラベルが自身と同じロールを代表対に持ち、かつラベルを持つ他の DR 領域集合から導出可能であるときに *subsumed label* であるとし、導出は自身のラベルが存在しない場合に以下の導出アルゴリズム 1 により行われる．

アルゴリズム 1 (導出アルゴリズム 1) 自身が代表対である DR 領域にラベル付けされていれば、そのラベルの s 部に示されているアクセス権限判定を返せばよい．そのようなラベルが存在しないとき、自身と同じロールを代表対に持ち、かつラベルを持つ最も近い先祖 DR 領域のラベルによって以下のように導出する．

(+, +) のとき (+, +) が導出される．

(-, -) のとき (-, -) が導出される．

(+, -) のとき 自身と同じロールを代表対に持ち、かつラベルを持つ最も近い子孫 DR 領域集合のラベルに *marker* でない $(+, *)$ が存在すれば、(+, -) が導出される．そのようなラベルが存在しなければ、(-, -) が導出される．

ラベルの追加設定 ある DR 領域に関して自身と同じ文書ノードを代表対に持ち、かつ文書構造において冗長なラベルを持つ子 DR 領域を持つならば、自身のラベルの r 部に $-$ を追加する．そうではないときに自身のラベルの r 部に $+$ を追加する．

ロール階層における冗長なラベル ロール階層における冗長なラベルとは同じ文書ノードを代表対に持つ DR 束内の他の DR 領域から包摂されるラベルであり、以下で定義される． DR 束を親から順に冗長なラベルであるか判定し、もし冗長なラベルであれば除去する．

定義 7 (ロール階層における subsumed label) DR 束内において自身のラベルがラベルを持つ親 DR 領域集合から導出可能であるときに *subsumed label* であるとし、導出は自身のラベルが存在しないとして以下の導出アルゴリズム 2 により行われる．

アルゴリズム 2 (導出アルゴリズム 2) 自身が代表対である DR 領域にラベル付けされていれば、そのラベルの s 部に示されてい

	d0	d1	d2	d3
r0	x			x
r1			x	
r2	x	x		x
r3	x	x	x	
r4	x	x	x	x

表 1 Accessibility Map

DT	XML 文書木
RH	ルール階層
Lattice	DR 束
	要素数
h	データ構造の高さ

表 2 コスト計算に使用する変数・関数

るアクセス権限判定を返せばよい。そのようなラベルが存在しないとき、自身と同じ文書ノードを代表対に持ち、かつラベルを持つ親 DR 領域集合のうちラベルの r 部に $+$ を持ち、かつ d 部に $-$ を持つようなラベルが 1 つでも存在するとき、 $(-, -, +)$ を導出し、そうでないとき $(+, +, +)$ を導出する。

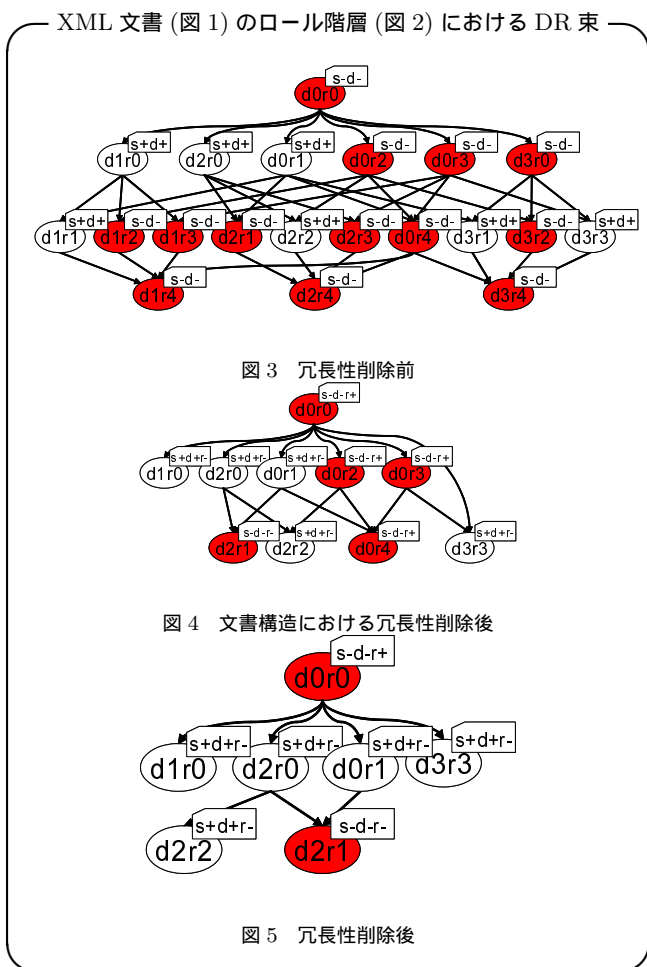
例 1 (DR 束) XML 文書木 (図 1), ルール階層 (図 2) に対して DR 束を構成する。アクセス権限設定が表 1 で与えられているときに、得られる DR 束が図 3 である。ここから文書構造において冗長なラベルを削除しラベルに r 部を追加したものが図 4 である。そこからさらにルール階層における冗長なラベルを削除したものが図 5 である。ここではラベルを持つ DR 領域のみを描いている。

4. アクセス権限判定および更新における操作

アクセス権限判定および更新操作とその時間コストについて述べる。また考察する更新は以下に挙げた操作である。

- (アクセス権限変更) 文書ノードとロールの対に設定されているアクセス権限情報を変更する。
- (文書ノード/ロールの追加) 新しく追加された文書ノード(ロール)におけるアクセス権限設定は追加先の祖先ノード(ロール)に設定されたアクセス権限判定ルールに従う。
- (文書ノード/ロールの削除) 削除対象の文書ノード(ロール)が XML 文書木(ルール階層)において子を持つ場合、子を文書ノード(ロール)の親の子として引き上げるような削除を行う。

また操作のコストを評価するために表 2 に用意した変数, 関数を用いる。また提案手法における要素数はラベル数とする。



4.1 提案手法における各操作と時間コスト

アクセス権限判定は以下の権限判定アルゴリズムを用いて行う。

アルゴリズム 3 (権限判定アルゴリズム) (a) 自身が代表対である Dr 領域にラベルが設定されていれば、ラベルの s 部に設定されているアクセス制御情報を用いて判定する。(a)ではなく(b)同じ文書ノードを代表対に持つ親 DR 領域集合から導出アルゴリズム 2 を用いてラベルを導出できれば、導出して判定する。(a)(b)のいずれでもなければ、同じロールを代表対に持つ親 DR 領域についてラベルが導出できるまで同様のことを繰り返す。導出されたラベルから導出アルゴリズム 1 を用いて自身のラベルを導出する。

定理 3 (提案手法における時間コスト) 提案における時間コストは表 3 で示される。

証明 5 提案手法に関して格納されているラベルの数は $|Lattice|$ であるため、1 回の問い合わせにつき $\log(|Lattice|)$ の時間コストが生じる。

操作	時間コスト
権限判定	$O((h(DT) \times RH) \times \log(Lattice))$
アクセス権限変更	$O(\log(Lattice))$
文書ノード追加	$O(1)$
文書ノード削除	$O(RH \times \log(Lattice))$
ルール追加	$O(1)$
ルール削除	$O(DT \times \log(Lattice))$

表3 提案手法における時間コスト

(権限判定) 権限判定アルゴリズムの適用に必要な親 DR 領域集合を得るために、最悪の場合権限判定対象の文書ノードの XML 文書木における先祖集合とロールのロール階層における可達集合の直積集合の回数の問い合わせが生じるが、得られた段階で探索は終了する。

(アクセス権限変更) ラベルが存在すれば書き換え、存在しなければ追加するため問い合わせは 1 回生じる。

(文書ノード追加) ラベルに対する処理はない。

(文書ノード削除) 各ロールを順にラベルが存在するか参照していくためロールの数だけ問い合わせと削除が生じる。

(ルール追加) ラベルに対する処理はない。

(ルール削除) 各文書ノードを順にラベルが存在するか参照していくため文書ノードの数だけ問い合わせと削除が生じる。

5. 更新操作の定量的評価

アクセス権限設定である Accessibility Map(AM)[1] とそれを圧縮した Compressed Accessibility Map(CAM)[1] と提案手法に関して XML 文書とロール階層を処理する計算機実験により、各手法に基づいたルール簡略化後に権限判定および更新操作をそれぞれ 30 回ずつ行い、それぞれの処理時間の平均値を得た。実験で用いた XML 文書の文書ノード数は 500 個、ロール数は 100 個である。

5.1 実験環境

Pentium 4 CPU 3.20GHz, 3.21GHz, 2.00GB RAM の PC, OS は Microsoft Windows XP, 実行プログラムは Java, DBMS は MySQL 5.0 を使用した。

5.2 実験用 XML 文書, ロール階層の生成

文書ノードとロール階層の対に対してアクセス権限情報を設定するために、文書ノードおよびロールを ID で管理できるようにデータベースに格納する。XML 文書における木構造は、求めるノード数に達するまで木に乱数で得られた ID を持つ文書ノードの葉として次々と追加することで生成した。またロール階層における非巡回有向グラフは求めるノード数で連結になるまで無向枝を追加し、乱数により source を定め、そこから枝をたどる際に有向枝に変えることで生成した。このようにして生成した文書木の子の数は平均 12 個、ロール階層の子の数は平均 3 個となった。XML 文書, ロール階層をともにグラフの隣接リストモデルを用いてデータベースに格納した。

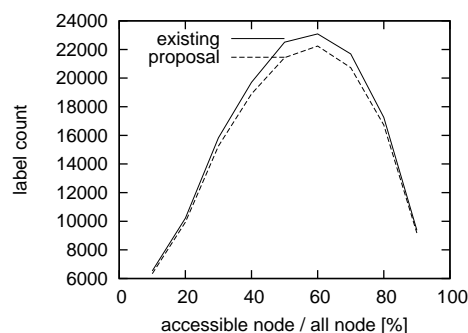


図6 ランダムに設定した場合

5.3 アクセス権限情報の付与

文書ノードとロールの全ての対に対してアクセス権限情報を設定を以下のように行う。まず局所性を持ちうる領域の指定として XML 文書内の文書ノード d とロール階層内のロール r を乱数により選び、 $DoS(d) \times DoS(r)$ に含まれる文書ノードとロールの対各々に対して次のように行う。確率 p_s (選択確率とする) によりアクセス制御情報の設定を文書ノードとロールの対ごとに独立に設定するか、または親のアクセス制御情報を考慮して設定するか決定する。文書ノードとロールの対ごとに独立に設定する場合、確率 p_r (許可確率とする) に従ってアクセス許可/アクセス拒否を設定する。また親のアクセス制御情報を考慮して設定する場合、確率 p_l (局所性確率とする) に従って親のアクセス権限情報を考慮してアクセス権限情報を設定する。これを全ての文書ノードとロールの対に対して設定されるまで繰り返す。

今回選択確率を文書ノードとロールの対ごとに独立に設定することを選択する確率とし、許可確率をランダムに設定する際にアクセス許可である確率とし、局所性確率を親のアクセス制御情報と等しい確率として、それぞれの確率について 10%~90% の 9 段階となるように設定した。それゆえそれぞれの確率の大小により次のようになる。選択確率が小さいほど文書ノードとロールの対ごとに独立に設定し、大きいほど親のアクセス制御情報を考慮して設定する。許可確率が小さいほどアクセス拒否を設定し、大きいほどアクセス許可を設定する。局所性確率が小さいほど親のアクセス制御情報と異なる設定をし、大きいほど親のアクセス制御情報と等しい設定をする。

5.4 構成とルール削減数の評価

まず文書ノードとロールの対ごとに独立にアクセス制御情報を設定した場合 ($p_s = 90\%, p_l = 50\%$) について行った結果が図 6 である。大きな差は見られないが、提案手法のほうが CAM からさらに圧縮を行う手法のため常にラベル数を下回っている。

一方、局所性を持たせてアクセス制御情報を設定した場合 ($p_s = 10\%, p_r = 50\%$) について行った結果が図 7 である。

5.5 各操作の実行時間

各手法で構成されたものそれぞれに対して乱数により選んだ対象に更新操作を行い、その平均所要時間で比較する。また比較の

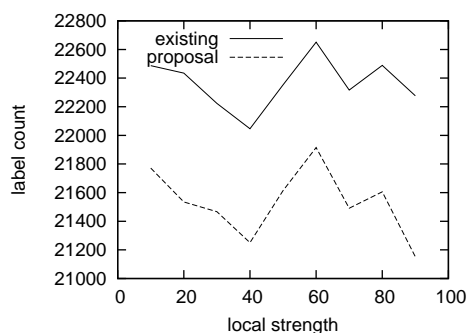


図7 局所性を持たせた場合

操作 \ 手法	AM	CAM	提案手法
アクセス権限判定	0.25	3.66	10.91
アクセス権限変更	0.02	0.58	0.66
文書ノードの追加	3.66	処理が不要	処理が不要
文書ノードの削除	29.83	12.58	11.91
ロールの追加	547.16	452.83	処理が不要
ロールの削除	135.00	51.58	51.16

表4 更新操作の所要時間 [ms]

ために AM に対しても同様の更新操作を行った。表4がその結果である。

(アクセス権限判定) 提案手法は探索すべき範囲が CAM に比べて大きいので、差が生じている。

(アクセス権限判定) アクセス権限判定した後に変更を行うため、先ほど述べた判定に必要な時間の影響が強くなっている。

(文書ノードの追加) 圧縮を行う手法では共通して何も行わないので、所要時間は発生しない。

(文書ノードの削除) AM および CAM では参照すべきノード数が多いので、処理にかかる時間が提案手法より必要となっている。

(ロールの追加) AM または CAM では新しくアクセス制御情報を構成する必要があるのに対し、提案手法では処理を行わないため差が生じている。

(ロールの削除) AM および CAM では参照すべきノード数が多いので、処理にかかる時間が提案手法より必要となっている。

6. おわりに

XML 文書のアクセス制御において、ロールベースドアクセス制御を取り入れ、ロール階層を導入することで、DR 領域という文書ノードとロールの対の集合を定義し、DR 領域に対してアク

セス制御ルールであるアクセス制御ラベルを設定し冗長性の除去を行った結果、ルール数削減効率と更新操作の効率に関して提案手法の優位性を実験により確認した。しかし、アクセス権限判定にかかる時間が増大したために、それに伴ってアクセス権限変更の効率は悪くなった点は改良の余地を残しているといえる。

今後の方針として提案手法を大規模な XML 文書木、大規模なロール階層に適用した場合のスケラビリティに関する調査を予定している。提案手法の優位性を顕著にし、また権限判定を高速化するため、新たな冗長性削除ルールを開発し、それとともに新たな探索アルゴリズムや格納方式の開発の予定である。

[謝辞]

本研究の一部は、平成19年度科研費基盤研究(B)(課題番号18300031)、および科学技術振興機構 戦略的国際科学技術協力推進事業「アイデンティティ連携におけるリスクを考慮した個人情報共有方式」による。

[文献]

- [1] T.Yu, D.Srivastava, L.V.S.Lakshmanan, and H.V.Jagadish, "A Compressed Accessibility Map for XML", ACM Transactions on Database Systems, Vol. 29, No. 2, June 2004, Pages 1-38.
- [2] M.Jiang and A.Wai-Chee Fu, "Integration and Efficient Lookup of Compressed XML Accessibility Maps", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 7, July 2005.
- [3] M.Iwaihara, B.Wang, and S.Chatvichienchai, "Extracting Global Policies for Efficient Access Control of XML Documents", WISE 2005, LNCS 3806, pp. 161-174, 2005
- [4] D.F.Ferraiolo, R.Sandhu, S.Gavrila, D.Richard Kuhn, and R.Chandramouli, "Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224-274.

牛場 祐貴 Yuki USHIBA

京都大学大学院情報学研究科修士課程在学中。京都大学工学部情報学科卒業。日本データベース学会学生会員

岩井原 瑞穂 Mizuho IWAIHARA

京都大学大学院情報学研究科准教授。1993年九大工学研究科情報学専攻博士後期課程修了。工学博士。日本データベース学会、情報処理学会、電子情報通信学会、ACM、IEEE 各会員。

吉川 正俊 Masatoshi YOSHIKAWA

京都大学大学院情報学研究科教授。1985年京都大学大学院工学研究科博士後期課程修了。工学博士。電子情報通信学会、ACM、IEEE Computer Society 各会員。日本データベース学会理事。