XMLデータベースのロール階層を考慮したアクセス制御におけるルールサマリを用いた高速化

牛場　祐貴† 　岩井原瑞穂† 　吉川　正俊†

† 京都大学情報学研究科　〒606-8501 京都市左京区吉田本町
E-mail: †y.ushiba@db.soc.i.kyoto-u.ac.jp, ††{iwaihara,yoshikawa}@i.kyoto-u.ac.jp

あらまし　アクセス制御を行う際には，アクセス制御ルール集合からルールを取得し，アクセスの可否の判定を行う必要がある．アクセス権限判定にかかる時間はルールの取得時間に大きく左右される．したがって効率的にルールの取得を行うために，ルール集合をその設定対象に基づいて構造化することで，権限判定の高速化が可能となる．また複数のルールを同時に取得することにより，取得にかかる総オーバーヘッドを減らすことが可能となる．これらの特徴を持つルール集合の構造としてルールサマリを提案する．ルールサマリはルール設定の局所性と XML 文書の文書構造およびロールの階層構造の相関性に基づいて集約により構造化されたルール集合である．ルールサマリを用いることで，ルールの探索と評価を効率化することができる．

キーワード　XML，セキュリティ，アクセス制御

Accelerating Hierarchical Role Based Access Control on XML Database using Rule Summary

Yuuki USHIBA†, Mizuho IWAIHARA†, and Masatoshi YOSHIKAWA†

† Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo–ku, Kyoto, 606–8591 Japan
E-mail: †y.ushiba@db.soc.i.kyoto-u.ac.jp, ††{iwaihara,yoshikawa}@i.kyoto-u.ac.jp

Abstract　To control the access, we need to get rules from access control rule set and to detect accessibilities. Detecting time depend heavily on collecting time. Accordingly constructing rule set leads to efficiently collecting rules, and it leads to accelerating to detection. We propose Rule Summary which has these merits. Rule Summary is structural rule set on the grand of correlation of the locality of rules and the structure of both of roles and XML documents. We use Rule Summary to lookup and evaluate rules efficiently.

Key words　XML, security, access control

1. ま え が き

XML が構造化文書を記述する形式として広く利用されるようになり，XML によって記述された文書は年々増加している．それとともに多様な情報をもつ XML 文書に対するアクセス権限を適切に管理することが求められている．きめ細かいアクセス制御におけるアクセス権限は XML 文書の構造や内容に応じて設定されるため，アクセス制御ルールは XML 文書の構造と深い関わりを持つ．本論文では，文書集合をディレクトリ構造や URI を含めて考えることにより，大きな木構造をなすとみなせる．

アクセス権限の管理者は，アクセスの主体であるユーザをその役割に基づいて分類し，ロールに割り当てることでアクセス権限管理を行う．ロールの導入の目的はアクセス対象であるオブジェクト-ユーザ間のアクセス権限割当をオブジェクト-ロール間のアクセス権限割当とロール-ユーザ間のロール割当に分割することである．このようにユーザから見たアクセス権限変更に対しても効率よくアクセス権限管理を行うロールベースアクセス制御 (RBAC) は NIST や ISO で標準化が行われ [1]，普及が進んでいる．RBAC ではロール間にアクセス権限の継承関係を持たせたロール階層を構成することにより，アクセス権限の付与や末梢を効率的に行うことができる．

企業などの大規模組織における様々な組織単位にアクセス権限管理を考えると，その組織単位ごとに多数のロールが存在しており，また管理対象となる文書も多数存在している．そのため，オブジェクト-ロールの組み合わせごとにアクセス権限割当

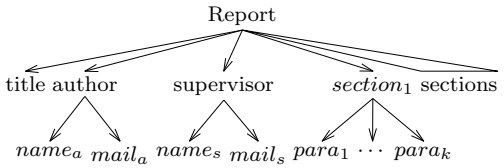

図 1　XML 文書の例

を行う方法では，適用されるアクセス制御ルール検索やアクセス権限更新において効率的ではない．そこで我々は検索を効率化するために適用されるアクセス制御ルールの集合を集約したルールサマリを導入し，また更新を効率化するために XML 文書の構造やロール階層構造に基づいたアクセス制御ルールの冗長性削減を行う．

以下，本論文では 2 節で問題として扱う範囲と既存手法を紹介している．3 節では，新たなアクセス制御ルールの集約方法と集約によるアクセス制御ルールの集合の構築を導入し，ルールサマリを提案する．4 節はまとめと今後の課題である．

2. 問題設定 - XML と RBAC

アクセス制御の対象である XML 文書について述べた後，アクセス制御の主体であるロール階層について述べる．その後に既存手法における問題点を挙げ，次節でその問題点を解決する手法であるルールサマリを提案する．

2.1 XML 文書に対するアクセス制御

XML 文書は各要素が開始タグと終了タグで囲まれた入れ子構造になっているため，親要素，子要素が定義できる．親子間に有向枝を張ることにより，図 1 のように XML 文書を木構造として図示できる．

図 1 の XML 文書に対してアクセス制御を行うことを考えると，タイトルや著者情報は見ることができるが，特定の節の情報は見ることができないなどといった XML 文書の構造に応じてアクセス制御ルールが設定される．

このようなアクセス制御ルールと文書構造の相関性を利用した方法が文献 [5] で提案されている Compressed Accessibility Map(CAM) である．Accessibility Map という XML 文書のノードごとのアクセス権限設定を表現するビットマップに対して，CAM は Accessibility Map のアクセス権限をアクセス制御ラベルというノード自身と子孫に関する情報で表現し，アクセス制御ルールが XML 文書の階層構造に沿って定義されることを利用して冗長なラベルを削除することでコンパクトにアクセス権限設定を管理するデータ構造である．文献 [5] ではこの CAM を用いて効率的にアクセス制御管理を行うことを提案している．さらに文献 [2] では，異なる操作に対して設定された複数の CAM を操作階層に基づき統合することで Integrated CAM(ICAM) を構成している，

文献 [3] では，アクセス制御ルールの検索効率をあげるために XML 文書に対して範囲ラベル (図 2) を設定し，アクセス制御対象をそのラベルで示したアクセス制御ルールを 2 次元平衡木を用いて管理している．さらに文献 [3] では，アクセス制御ルールを先読みする仕組み，および Dynamic Predicate(DP)

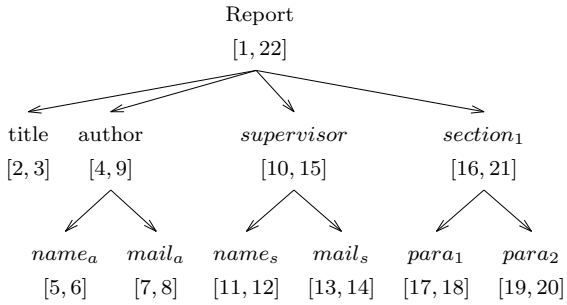

図 2　範囲ラベルをつけた XML 文書の例

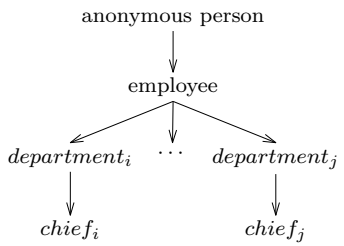

図 3　ロール木の例

という評価しながら権限判定対象とする文書範囲を書き換える仕組みにより，高速にアクセス権限判定を行っている．

2.2 ロールベースアクセス制御 (RBAC)

National Institute of Standards and Technology(NIST) で標準化されている RBAC [1] の目的は，ロールの導入によりオブジェクト-ユーザ間のアクセス権限設定を分割することで，アクセス制御管理を容易にするためである．またロール間に権限の継承関係を定義したロール階層を取り入れることで権限管理をより容易にすることができる．図 3 のように，単一の継承元を持つロール木を考え，さらにロール木の集合によってロール階層を表現するものと仮定する．

図 3 の例では，企業の部署ごとに所属者及び責任者というロールが存在している．ある部署の責任者であるためには，その部署に所属しており，かつ社員である必要がある．これはロール木内の継承関係に課す制限であるが，実世界でも同様のことが言える．

2.3 既存手法の問題点

アクセス制御ルールによって文書の部分木全体といった文書構造を単位としてアクセス権限が指定されることが多い．文献 [3], [5] ではこのようなアクセス制御ルールと文書構造の相関性を利用して冗長なアクセス制御ルールを除去することで，アクセス権限判定を効率化している．

一方，RBAC においてもロール木における子ロールの権限は親ロールにも与えられるといった継承関係を用いるアクセス制御ルールが多く存在している．しかし文献 [3], [5] を含む既存研究では，このようなロール間の継承関係の考慮は行われていなかった．文献 [2] では，複数の操作に関して操作階層を考慮してアクセス制御ルールの統合を行っているが，操作集合からなる階層構造の極大要素のみを権限付与対象とする制限を与えているため，操作階層をロール階層ととらえて ICAM を構成することはできない．

ここで問い合わせを考えると，ユーザが同時に複数の部分文

no	制御の主体	制御対象	権限
1	employee	Report//*	deny
2	employee	title//*	grant
3	employee	author//*	grant
4	$department_A$	$section_1$//*	grant
5	$chief_A$	supervisor//*	grant
6	$chief_B$	supervisor//*	grant

表 1　設定されるアクセス制御ルールの例

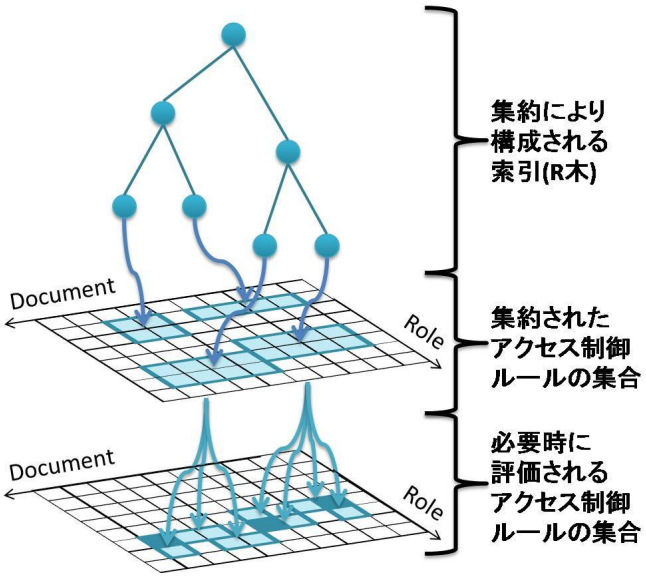

図 4　ルールサマリの構造

書を要求することがあるため，文献 [3] では Dynamic Predicate(DP) により複数のアクセス制御対象を同時に判定することを可能にし，そのような複数の対象に対する問い合わせに対処している．同様に，ユーザに同時に複数のロールに割り当てられることもあり，そのように割り当てられたロール集合をまとめて判定を行うことができれば，効率的にアクセス権限判定を行うことができる．

この様に，文書構造とロール階層の双方の継承関係を考慮したアクセス制御ルールの冗長性除去や複数のロールについてまとめて権限判定を行うといった効率化がこれまでの研究では考慮されていなかった．

次節では，文書構造を利用して効率化する方法とロール階層を単純に組み合わせて効率化する方法を説明した後に，ルールサマリを導入した方法を説明する．

3.　提案手法 - ルールサマリ

表 1 のように，複数のルールが適用されている場合に，高速にアクセス権限判定を行うことができるルールサマリを提案する．

ルールサマリは，与えられたアクセス制御ルールの集合をロール階層および文書階層の双方に従って集約することで構成されるものであり (図 4)，アクセス権限判定のために必要なアクセス制御ルールの数を削減する目的で利用される．また継承関係による権限の導出をアクセス制御ルールの評価時に行うことにより，文書の更新やロール割り当ての更新によるオーバヘッドも抑えることを目的としている．

まずアクセス制御ルールを集約するために，アクセス制御の対象と主体の構造を利用する理由を述べた後，構造を利用するために用いる木構造に対するラベリング手法について述べる．次に複数のアクセス制御ルールを集約することで，同時にアクセス権限判定を行うことができるため，アクセス制御の対象範囲であるロール階層および文書階層のいずれか，あるいは双方において包含関係にある，あるいは近接関係にあるアクセス制御ルールの集合を集約する方法を述べる．

3.1　ロール階層を考慮したアクセス制御ルール

アクセス権限判定を行う際に，アクセス権限の継承を予め評価し，その結果を保持すると高速に実行できる．しかし，継承を予め評価するのはアクセス権限を分配することであり．更新の対象が多く存在することになる．そのためアクセス権限の変更が多い場合は管理の効率が低下するというトレードオフがある．

一方，継承関係を予め評価しない場合では，アクセス制御ルールの数を減らすだけでなく管理を効率化する目的から，冗長なアクセス制御ルールを削減することが行われている．冗長なアクセス制御ルールを判定するために，文献 [3], [5] を含む既存のアクセス制御手法ではアクセス制御対象の構造，また RBAC ではアクセス制御の主体の構造をそれぞれ独立に利用しているため，ルールサマリでは双方の構造を同時に使用した場合を考える．

例えばロール木 (図 5) と XML 文書 (図 6) に対して，表 1 のようにルールが設定されたとする．もし文書構造あるいはロール構造を考慮していなければ，より多くのルールが必要になるとともに，管理の面で非効率となる．すなわちここで文書の $author$//* 部分のアクセス権限を現在の $employee$ およびその継承先から $department_A$ およびその継承先に変更することを考える．ロール階層を考慮した場合では 1 つのルールを削除して新たに 1 つのルールを追加するだけで済むが，考慮しない場合では 5 つのルールを削除する必要がある．

このようにアクセス制御ルールをアクセス制御の対象と主体双方の構造に関して集約していない場合は，管理の点で非効率である．そのためルールサマリでは，アクセス制御の対象と主体である文書構造とロール木に関して集約を行う．

3.2　構造に対するラベリング手法

継承関係を予め評価しない場合でも比較的高速にアクセス権限判定を行うことができ，またアクセス権限を評価した形式で保持されていた場合でも更新によって影響を受ける範囲の特定を容易に行うことができるように，我々は木構造のノードラベリング手法を利用する．

ルールサマリに設定するラベルに要求する特徴は，アクセス制御ルールの集合を効率的に管理する目的で，ラベル間の比較だけでラベルが指す対象間の包含関係の判定を行うことができるもの，かつラベルが指す対象間の近接関係を判定を行うことができるものである．さらに効率的に管理を行うために，更新に強いラベリング手法を採用する．この様なラベルの特徴によ

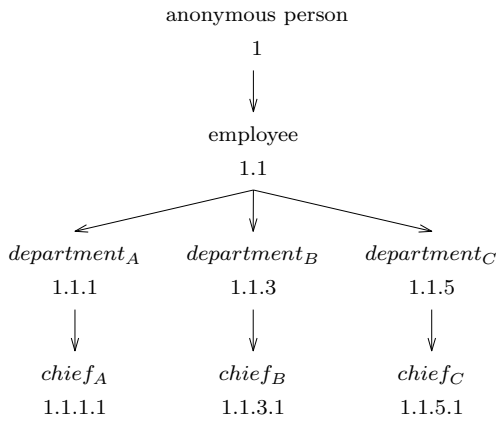

図 5　ORDPATH をつけたロール木の例

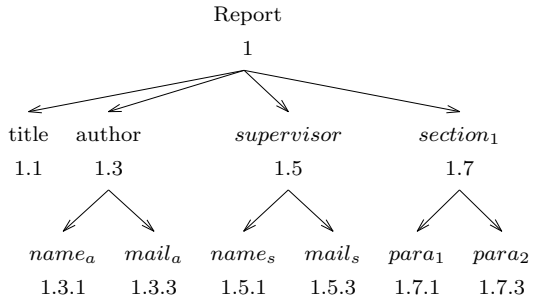

図 6　ORDPATH をつけた XML 文書の例

り，アクセス制御ルールの適用範囲の特定だけでなく，アクセス権限の継承の特定及び更新対象となる範囲の特定をも高速に行うことができる．

このようにルールサマリでは，実際のアクセス制御ルールが持つ構造を参照すること無く利用する目的で，構造をラベルにより表現している．またこの目的で設定されるラベルを構造に対するラベルと呼ぶ．

本論文では，ORDPATH [4] を構造に対するラベルとして利用する． ORDPATH は，例えばロール木 (図 5) と文書構造 (図 6) に付与したように，親に設定されたラベルを接頭辞とし兄弟間での順序に応じて奇数をつけることで構造に対するラベルを設定する．また ORDPATH において，偶数はノードを構造に追加した時に利用するために空けられている．さらに構造に対するラベル間の距離を定義するために，ORDPATH 間に辞書順による全順序を導入することで，親 < 先行する兄弟 < 自身 < 子孫 < 後続する兄弟とすることができる．

3.3　包含型ルールサマリの構成

まず冗長なアクセス制御ルールを削減するため，包含関係に基づく集約を行うことで，包含型ルールサマリを構成する．

複数のアクセス制御ルールを集約する方法として，アクセス制御ルールの対象範囲の包含関係を用いて，他のアクセス制御ルールに包含されるアクセス制御ルールを削除する方法が文献 [5] で提案されている．この集約方法はアクセス制御ルールは同一ユーザ/ロールを対象にしたルールで文書構造において包含される対象範囲に設定されたアクセス制御ルールを削減し冗長性を除去するものである．しかし，文献 [5] においてはこ

	制御の主体	制御対象	権限
集約対象	$department_A$	$section_1$//*	grant
	$chief_A$	$section_1$//*	grant
	↓		
対象書換	1.1.1	1.7	grant
	1.1.1.1	1.7	grant
	↓		
集約結果	1.1.1	1.7	grant

表 2　包含型ルールサマリの構成

	制御の主体	制御対象	権限
集約対象	employee	title//*	grant
	employee	author//*	grant
	↓		
対象書換	[1.1, 1.2)	[1.1, 1.2)	grant
(範囲明示)	[1.1,1.2)	[1.3,1.4)	grant
	↓		
集約結果	[1.1,1.2)	[1.1,1.4)	grant

表 3　近接型ルールサマリの構成

のように文書構造のみに適用しているが，この集約方法は文書構造とロール階層の双方を考慮した場合にも同様に適用することが可能である．そのためルールサマリでは，ロール階層および文書構造の双方における包含関係を考慮して，他のアクセス制御ルールに包含されるアクセス制御ルールを削減する．

すなわちアクセス制御ルールの主体と対象からなるアクセス制御ルールの対象範囲を考えることにより，この集約を行うことが可能となる．ここで設定対象範囲の包含関係の判定には，構造に対するラベルを利用する．これを包含関係に基づく集約と呼び，包含関係に基づく集約によって構成されるルールサマリを包含型ルールサマリと呼ぶ．

例えば，表 2 において，集約対象の二つのアクセス制御ルールから包含型ルールサマリを構成することを考える．まずアクセス制御の主体と対象をそれぞれロール木 (図 5) と文書構造 (図 6) を参照して，それぞれの構造に対するラベルに書き換える．次にそれらのラベルから包含関係を判定し，包含されるアクセス制御ルールを削除することで包含型ルールサマリの構成を行う．

このように包含型ルールサマリは，冗長なアクセス制御ルールを削減する目的で，包含関係に基づく集約により構成される．

3.4　近接型ルールサマリの構成

包含関係に基づく集約のみでは，包含関係にない複数のアクセス制御ルールを集約することができないため，ルールサマリを構成することはできない．

例えば，表 3 において，集約対象の二つのアクセス制御ルールは包含関係にないためにルールサマリを構成することができない．

そこで包含関係にないが，与えるアクセス権限が一致する場合に，複数のアクセス制御ルールを集約することで，ルールサマリを構成する方法を提案する．

まずこれらのアクセス制御ルールを構造に対するラベルで書き換え，アクセス制御ルールの主体および対象を [start, end) という左閉右開区間で表記することで，そのアクセス制御ルールによる対象範囲を明示する．次にアクセス制御ルールの集合をそれらの対象範囲の最小被覆長方形 (MBR) を対象範囲とするアクセス制御ルールに置き換えることで，アクセス制御ルールを削減する．これを近接関係に基づく集約と呼び，近接関係に基づく集約によって構成されるルールサマリを近接型ルールサマリと呼ぶ．

すなわち集約対象のアクセス制御ルールの対象範囲の MBR を対象範囲とするアクセス制御ルールに集約することで，近接型ルールサマリを構成し，これらの対象範囲に対する問い合わせを効率化することができる．

このように近接型ルールサマリを構成することにより，例における [1.2,1.3) のように本来入るべきでない範囲が対象範囲に入ってしまい false positive/false negative が生じることになる．このために元の二つのアクセス制御ルールを削減するのではなく，検証のためのアクセス制御ルールとして保持しておき必要に応じて評価することで，アクセス権限判定を行う．しかし，この対象範囲に入らない場合は元のアクセス制御ルールを評価する前に判定が終了するため，アクセス権限判定を効率化することができる．

さらに，近接型ルールサマリを構成する際には，false positive/false negative が起きる割合が小さい近接関係にあるようなアクセス制御ルールの集合に関して集約を行うことで，効率的にアクセス権限判定を行うことができる．

すなわち，遠く離れた関係にあるアクセス制御ルールの組み合わせから近接型ルールサマリを構成した場合，MBR が大きくなるため，評価すべき対象範囲が大きくなる．これにより頻繁に別テーブルへ集約前のアクセス制御ルールの集合を取得する必要があるため，効率が悪くなる．この点から近接型ルールサマリは近接関係に基づく集約を行った結果生じる MBR がより小さく false negative/false positive が起きにくい近接関係にあるアクセス制御ルールの集合に対して構成される．

さらに，構造に対するラベルによる範囲を用いて対象範囲を指定することにより，階層構造のパス式で指定する場合に比べて小さな対象範囲に集約することができ，positive false/negative false が生じる割合を小さくできる．

このように近接型ルールサマリは，複数のアクセス制御ルールを同時に判定する目的で，近接関係に基づく集約により構成される．

3.5 集約したアクセス制御ルールに対する索引

近接関係に基づく集約は，与えるアクセス権限が一致するアクセス制御ルールの集合に関しての集約であり，与えるアクセス権限が一致しないアクセス制御ルールの集合に関して適用することができない．

そこで与えるアクセス権限が一致しないアクセス制御ルールの集合に対して集約を行う．

前節までの集約されたアクセス制御ルールの集合に対して，その対象範囲の MBR を対象範囲とするようなアクセス制御ルールを追加することで，R 木を構成しアクセス制御ルールの索引として用いる．集約は R 木構成における MBR の面積総和最小戦略と同様に，各集約ごとに作られる MBR が最小となるように集約を行うことで，索引の探索効率を上昇させる．

それと同時にアクセス制御ルールを評価する際に，一つずつ取り出していたのでは I/O コストが高く非効率である．そこである程度の大きさのアクセス制御ルールの集合を一度に取り出すことで効率を上げる．階層化されたアクセス制御ルールの集合において，あるアクセス制御ルールを取得する際に，同時に評価される下の階層のアクセス制御ルールの集合ごと取得することでこれを実現する．

4. まとめと今後の課題

アクセス制御ルール集合に対して，ロール階層および文書構造の双方を考慮しながら集約を行うことで構成され，権限判定を高速化するルールサマリについて述べた．

今後の課題として提案手法の定量的評価を行い，ルールサマリの効果を実証することが挙げられる．またそれとともに，ルールサマリが有利にはたらく状況，あるいは不利にはたらく状況を特定することで，さらに効率的な手法を開発する．

文　献

[1] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

[2] Mingfei Jiang. Integration and efficient lookup of compressed xml accessibility maps. *IEEE Trans. on Knowl. and Data Eng.*, 17(7):939–953, 2005. Member-Ada Wai-Chee Fu.

[3] Jae-Gil Lee, Kyu-Young Whang, Wook-Shin Han, and Il-Yeol Song. The dynamic predicate: integrating access control with query processing in xml databases. *The VLDB Journal*, 16(3):371–387, 2007.

[4] Patrick O'Neil, Elizabeth O'Neil, Shankar Pal, Istvan Cseri, Gideon Schaller, and Nigel Westbury. Ordpaths: insert-friendly xml node labels. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 903–908, New York, NY, USA, 2004. ACM.

[5] Ting Yu, Divesh Srivastava, Laks V. S. Lakshmanan, and H. V. Jagadish. A compressed accessibility map for xml. *ACM Trans. Database Syst.*, 29(2):363–402, 2004.