L-024

IT 全般統制のための職務分掌検証方式

A Verification Method for Segregation of Administrative Duties under IT General Controls

森田 陽一郎†，中江 政行†，小川 隆一†
Yoichiro MORITA, Masayuki NAKAE, Ryuichi OGAWA

1. はじめに

金融商品取引法 (日本版 SOX 法) では，金融庁の実施基準に基づき，各企業に内部統制と外部監査の実施を義務付けている．この実施基準では，内部統制の基本的要素のうち，特に IT に対応する統制活動 (IT 統制) として，IT 業務処理統制と IT 全般統制の 2 つが挙げられている．

IT 業務処理統制とは，業務を支援する IT システム (業務システム) について，作業の内容や流れを見直し，必要に応じて再構築することで，効率的で正確な処理を可能とし，結果として不正を防止できる体制作りを目指す活動である．IT 全般統制とは，IT 業務処理統制が有効に機能することを保証するため，業務システムの開発・変更や実行などの作業について，あらかじめ作業内容を規定し，規定外の不適切な作業の防止やチェックの体制に抜けがないよう IT 基盤 (ハードウェア，ソフトウェア，ネットワークなど) を管理する活動である．

このうち，IT 全般統制におけるアクセス権設定作業において，不正な管理業務を防止するために，管理者の担当する職務を分割して不適切な兼務を禁止する職務分掌制約の徹底が求められており，本論文ではその高速検証方式を提案する．

2. IT 全般統制における職務分掌制約の徹底

IT 全般統制において，職務分掌 (SoD; Segregation of Duties, Separation of Duties) 制約の徹底が重要とされている．職務分掌制約とは，重要な職務の兼務による不正作業を防止するための制約である．具体的には，管理者の職務を開発と運用に分割し，開発で扱うソースコードと，運用で扱う実行バイナリを，同一管理者が更新することを禁止するなどの制約を設ける．

IT 全般統制の実務では，こうした職務分掌制約を考慮して，管理作業の適正な担当者や手順を業務ポリシーとして明文化し，IT 基盤を構築する際に，その内容にしたがってアクセス権設定を施すことが求められる．IT 基盤の構築は人手に委ねられており，アクセス権設定が，厳密な職務分掌制約を満たすかを人手で検証することは困難なため，現状では，大雑把な職務分類に留めるなどの簡易な対応がなされている．しかし，今後はより厳密な職務分掌制約の徹底を図りたいとのニーズが高く，効率的な制約検証技術が求められている．

そこで筆者らは，業務システムを開発・運用する IT 基盤のアクセス権設定について，業務ポリシーに記載された職務分掌制約の充足性を自動的に検証する技術開発に取り組んでいる．この中では，制約を満たすか否かだけでなく，制約に反するアクセス権を提示する機能も必要となる．そのため，システム管理作業で発生するすべてのアクセス権設定について，制約との照合を行う必要があり，規模の大きい基幹業務システムへの適用を考慮すると，多数のアクセス権設定を実用時間内に検証できる高速性が特に重要な技術課題となる．

†日本電気（株）共通基盤ソフトウェア研究所，
Common Platform Software Res. Labs., NEC Corp.

3. 職務分掌検証方式

3.1. 従来のポリシー形式検証

筆者らが設計した職務分掌検証方式を述べる前に，ポリシー形式検証の従来技術を概観する．

まず，Harison-Ruzzo-Ullman Model (HRU)，Schematic Protection Model (SPM)，Typed Access Matrix Model (TAM) など，アクセス制御機構をオートマトンとしてモデル化し，あらゆるアクセス要求に対する認可判定をエミュレートしながら制約の充足性を検証する，モデル検証方式が提案されている．これらの利点は，OS やパケットフィルタなど特定のアクセス制御機構に依存せず，また任意の制約を取り扱える汎用性にある．しかし，プログラム停止性問題などと同様に決定不能 (いくつかの限定を加えても NP 困難) であることがわかっており，実用性に乏しいアプローチと考えられる．

より実用的な検証方式として，グラフベース検証方式がいくつか提案されている．グラフベース検証方式とは，OS やパケットフィルタなど特定のアクセス権設定を，ツリーや有向グラフなどでモデル化し，グラフ構造に基づいて制約充足性を高速に検証するものである．このようなアプローチは，特定のアクセス制御機構に関するドメイン知識を用いてモデルを設計することで，高速な検証アルゴリズムを導出できる一方，検証対象が特定のアクセス機構に限定されるという問題を持つ．

実用的な検証方式の別のアプローチとして，マトリクス検証方式 [1] がある．マトリクス検証方式は，ルータやファイアウォールのパケットフィルタリングルールを，IP アドレスやポート番号などを軸とする多次元空間としてモデル化することで，制約の充足性を幾何学的な操作により高速に検証する．

3.2. マトリクスモデルに基づく職務分掌検証方式

筆者らは，マトリクスモデルに基づく職務分掌検証方式を提案する．従来のマトリクス検証方式を拡張し，一般的なアクセス権モデルの構成要素であるサブジェクト S・オブジェクト O・アクション A を軸とするマトリクスモデルを用いることで，特定のアクセス制御機構に限定されない汎用の高速検証アルゴリズムを導出できる．

本方式では，職務分掌検証を，アクセス権設定に含まれる各アクセス権ルールの中から，所定の職務分掌制約に違反するアクセス権ルールを抽出する問題であると捉える．また，職務分掌制約を，同一管理者による兼務を禁止する職務の指定と定義する．

以上の直観的定義を定式化する．まず，単一のアクセス権ルールは，ある管理者 (サブジェクト) s が あるファイル (オブジェクト) o に対して ある操作 (アクション) a を加えるというルールであり，それぞれアクセス権 (s, o,

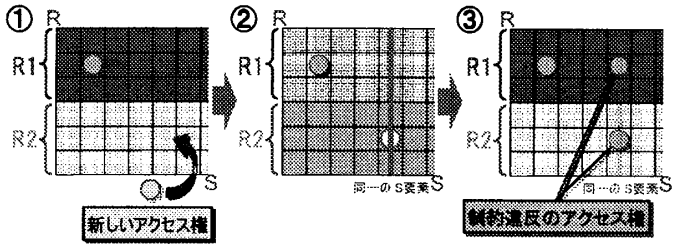

図 1: マトリクスモデルに基づく職務分掌検証方式

a) として定義され，アクセス権設定はアクセス権の系列 σ={(s0, o0, a0), (s1, o1, a1), …, (sn, on, an)} であると定義される．

次に，職務は上述の例のようにファイル o と，o に対する操作 a の組 (o, a) とみなすことができる．厳密には 1 つの職務に対して複数の (o, a) が指定され得ることから，(o, a) の集合であり，ロールベースアクセス制御モデルにおけるロールと同じものである．したがって，職務分掌制約とは 2 つのロールの組 (R1, R2) を指定したものと定義できる．ここで，あるロール R={(o0, a0), (o1, a1), …, (on, an)} に対して，任意の s が行使するアクセス権 (s, oi, ai) (0<=i<=n) を，R に合致するアクセス権と呼ぶ．

このとき，職務分掌検証とは，職務分掌制約 (R1, R2) と，アクセス権系列 σ とが与えられたとき，σ に含まれるアクセス権のうち，R1 に合致するアクセス権と R2 に合致するアクセス権について，双方に，同一の s によって行使されるアクセス権があれば，それらを抽出する問題であると定義できる．

以上の定式化を踏まえ，マトリクスモデルに基づく職務分掌検証方式のアルゴリズムを説明する．準備として，あるサブジェクトの集合 S と，職務分掌制約 (R1, R2) を与え，S と R (=R1∪R2) のマトリクスを生成する (図 1)．これは，R1 あるいは R2 に合致する各アクセス権を，それぞれ 1 つのセルとして持つテーブルである．このマトリクスを用いて，以下のような手順でアクセス権系列 σ の職務分掌検証を行う．

① アクセス権系列 σ から，各アクセス権を取り出して，そのアクセス権に該当するセルに格納する (図 1 の①)．このとき，R1 や R2 に合致しないアクセス権は該当するセルが無いため，格納することなく廃棄する．

② 新しく格納したセルと同じ S 要素のセル列のうち，新しく格納したセルのロールとは逆のロールに対応するセルを走査し，格納済みのセルの有無を確認する．この結果，格納済みのセルが無い場合 (図 1 の②) は，制約違反ではない．

③ 以降①②を繰り返し，同じ S 要素のセル列の中で，双方のロールについて，格納済みのセルを検出した場合 (図 1 の ③) は，同じ管理者 s が，職務分掌制約で与えられた R1 と R2 の双方のアクセス権を行使するため，制約違反であると判定し，それらのセルのアクセス権を提示する．

以上の手順で すべての σ を検証すると，職務分掌制約に違反する すべてのアクセス権を列挙できる．特に，与えられた職務分掌制約に応じてマトリクスを生成することで，膨大なアクセス権設定の空間を，制約の検証に必要十分な記憶領域に圧縮し，高速な検証が可能となる．

4. 評価実験

4.1.実験方法

本方式のパフォーマンス評価のため，Perl を用いて前節のアルゴリズムを実装し，シミュレーション実験を行った．実験の入力データとして，S, O, A の各集合からランダムに要素を抽出して，職務分掌制約 (R1, R2) とアクセス権系列 σ を生成した．R1, R2 のサイズは，o, a の すべての組合わせのサイズの 1% ずつとした．なお，R1 と R2 は互いに素 (R1∩R2=φ) とした．

実験環境には，Pentium4 3GHz，メモリ 1GB (PC2700), Red Hat Linux 7.3 上の Perl 5.6.1 を用いた．

検証のパフォーマンスは，アクセス権設定のサイズ σ と，ファイル数などシステム規模の指標となる O に依存すると考えられるため，σ や O のサイズを変化させた場合の実行時間とメモリ使用量の変化を計測した．なお，S のサイズは 100,000，A のサイズは 100 とした．

4.2. 実験結果

図 2 に実験結果を示す．上段は O を 100,000 で固定して σ を増減させた場合，下段は σ を 100,000 で固定して O を増減させた場合の，実行時間とメモリ使用量の変化を示すグラフである．

実行時間は，σ のサイズに比例して増減するが，O のサイズ変化に対しては一定であった．提案方式の高速性はシステム規模に対してロバストであると言える．設定のサイズについても，1,000,000 の σ の検証が 4 秒弱で可能であり，実用上 十分であると予想される．

メモリ使用量は，σ 100,000 件あたり約 500kB，O 100,000 件あたり約 20MB 増加した．提案方式のメモリ効率は，設定のサイズに対してロバストであると言える．システム規模については，より多数のファイルを持つ基幹システムなどを対象とする場合，メモリ効率の悪化が見込まれるため，さらに効率化を要すると考えられる．

5. おわりに

マトリクスモデルに基づく職務分掌検証方式について述べた．職務分掌制約の検証と違反部分の抽出が可能なアルゴリズムを設計・実装し，シミュレーション実験を通じて，高速な制約検証が可能であることを確認した．

参考文献

[1]松田 勝志, “マトリックス分解によるパケットフィルタリングルールの分析 —不要ルールと冗長条件ルールの検出—,” 情報処理学会研究報告, v. 2005, n. 122, 2005, pp. 1-6.

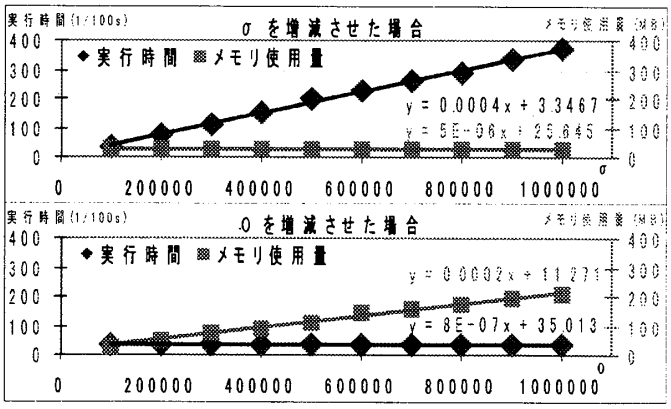

図 2: 実験結果