B-023

RBAC モデルの形式検証
Formal Verification of RBAC Models

鈴木 大輝† 小林 秀幸† 岡本 圭史† 高橋 薫†
Daiki Suzuki Hideyuki Kobayashi Keishi Okamoto Kaoru Takahashi

1. はじめに

アクセス制御を用いて不正な利用から情報システムを守る手法の 1 つとして RBAC(Role-Based Access Control) [1]がある．RBAC では，ロールごとにパーミッションの割り当てを行い，ユーザがどのロールを所有しているかによりアクセス可能なオブジェクトとその操作が決まる．パーミッションをロールごとに割り当てることで，複雑化したアクセス制御を効率よく行うことができる．しかしその反面，ユーザに直接パーミッションを割り当てないことやロール継承の概念により，具体的にどのユーザがどのパーミッションを持っているか分かりにくい場合がある．また，RBAC モデルの記述は人手で行われるため，矛盾や誤りが含まれる可能性がある．そのため，具体的に記述された RBAC モデルが意図通りに表現できているか検証を行う必要がある．

本稿では，記述された RBAC モデルを検証する手法を提案する．検証には，ソフトウェアや組込みシステムの検証において有効なモデル検査の手法を応用する．モデル検査ツールとしては NuSMV[2]を用いる．

2. RBAC と状態遷移システム

本節では，RBAC モデルを NuSMV で扱うための検証モデル（状態遷移システム）について述べる．

NuSMV で扱うモデルは状態遷移システムであるため，RBAC モデルを状態遷移システムで表現する必要がある．そこで本研究では，RBAC モデルを状態遷移システムに変換する（対応づける）規則を導入する．RBAC モデル *RM* と状態遷移システム *Sys* は図 1 のように定義される．

RBAC モデル *RM* において，*U* はユーザの集合，*R* はロールの集合，*P* はパーミッションの集合，*S* はセッションの集合，*UA* はユーザ割当て，*PA* パーミッション割り当て，*RH* はロール階層である．一方，状態遷移システム *Sys* において，*Q* は状態の集合，→は状態遷移関係である．

状態遷移システムでは，RBAC モデルにおける(1)ロール，(2)そのパーミッションの集合，(3)そのロールに割り当てられるユーザの集合の三つ組を”状態”とし，ロール階層を”遷移”とする．また，RBAC モデル *RM* が与えられた時，対応する状態遷移システム *Sys* を得るための規則として，*GetAllState* ()，*GetAllTrans* ()を定義する．

GetAllStates ()は，すべてのロールについて，そのパーミッションの集合，ユーザの集合の三つ組を返し，状態全体 *Q* を得るための関数である．*GetState*(*r*)では，ロール *r* について，そのパーミッションの集合，ユーザの集合を返す．

また，すべてのロール階層の組を返し，状態遷移関係全体を得るための関数として，*GetAllTrans* () を定義する．*GetTrans*(*rh*)は，ロール階層 *rh* について，*Sys* の状態遷移要素を構成する．具体的には r_u を上位ロール，r_u を下位ロールとし，$rh = (r_u, r_u)$としたとき，*GetState*(r_u)と *GetState*(r_u)の間に関係を与えることで達成する．

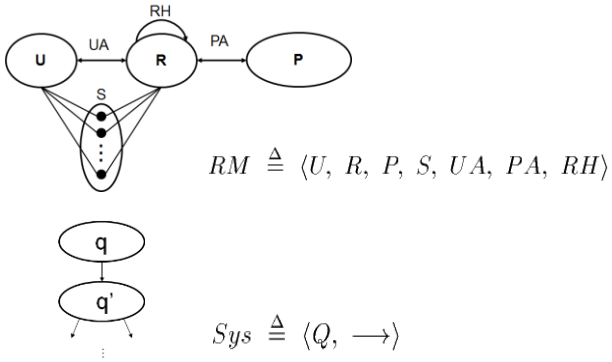

図 1 RBAC モデルと状態遷移システム

```
GetAllStates() : set Q
{
    var X : set Q;
    X := {};
    for all r ∈ R
        X := X ∪ GetState(r);
    return X;
}
GetAllTrans() : set ⟶
{
    var X : set ⟶;
    X := {};
    for all rh ∈ RH
        X := X ∪ GetTrans(rh);
    return X;
}
```

図 2 RBAC モデルから状態遷移システムを得る関数

3. 検証例

本節では，RBAC モデルの検証項目として考えられるパターンを例に用いて検証を行う．前節に述べた変換を行い，RBAC モデルから状態遷移システムへの変換を行う．状態遷移システムを NuSMV に入力し，検証内容を時相論理式で記述する．NuSMV は時相論理式が満たされているか否かの判定を行い，満たされていない場合，反例を出力する．その反例を用いて解析を行うことで，RBAC モデルの修正を図る．

† 仙台高等専門学校
Sendai National College of Technology

3.1　階層関係にループを含む RBAC モデルの検証

図 3 に示す RBAC モデルは，「ロール階層にループを含む RBAC モデル」である．RBAC モデルにおけるロール階層は半順序関係であり，上位ロールが下位ロールの権限を継承する．しかしこのモデルでは，***r2, r5, r6*** のロール階層にループが生じているため，下位ロールが上位ロールの権限を持つ，といった問題が生じている．このようなモデルに対して検証を行う必要がある．図 3 を変換した状態遷移システムを図 4 に示す．

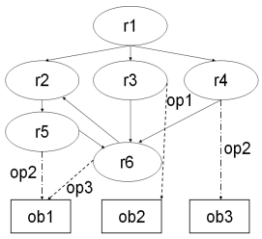

図 3　ロール階層にループを含む RBAC モデル

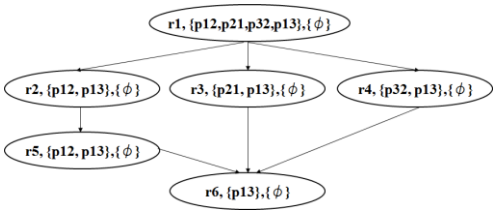

図 4　図 3 の状態遷移システム

この RBAC モデルに対し，検証を行う時相論理式は以下の式である．

[検証式]

$$AG\,((r = r_i) \rightarrow !\,EX\,(EF\,(r = r_i)\,))\quad (r_i \in R)$$

この検証式では，現在のロールから始まるロール階層に再び自分が存在するか否かを検証している．ここで r_i は検証対象となる具体的なロールである．たとえば $r = r1$ の場合には，次の状態から始まるパスで，再び $r = r1$ となるパスが存在しない，という意味になる．この検証式と状態遷移システムを NuSMV に入力し，検証を行う．この検証式が真となれば，ループは存在していないことを示す．偽であれば，反例を用いて確認することができる．

図 5 は，この検証式によって得られた検証結果である．$r = r2$，$r = r5$，$r = r6$ のときに，偽が出力されているため，これらのロールにはループが生じている，と解析することができる．

Properties database

Context	Index	Select	Value	Trace	Type	Property
Main						
	0	☑	True		CTL	AG (r = r1 -> !EX EF r = r1)
	1	☑	False	1	CTL	AG (r = r2 -> !EX EF r = r2)
	2	☑	True		CTL	AG (r = r3 -> !EX EF r = r3)
	3	☑	True		CTL	AG (r = r4 -> !EX EF r = r4)
	4	☑	False	2	CTL	AG (r = r5 -> !EX EF r = r5)
	5	☑	False	3	CTL	AG (r = r6 -> !EX EF r = r6)

図 5　ループの検証結果

3.2 SoD の検証

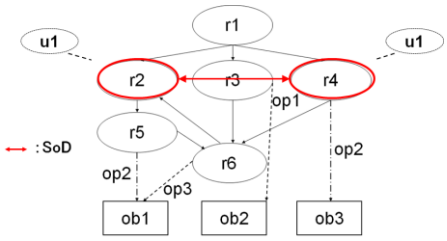

図 6　SoD 違反を含む RBAC モデル

RBAC は Separation of Duty(SoD) [3]の概念を持つ．SoD は，同じユーザが同時に持つことのできないロールの制約であり，ここでは，Static Separation of Duty (SSoD) について扱う．SSoD では，1 人のユーザに複数のロールを割り当てた場合，それらのロールが SoD 関係で結ばれていないかを検証する必要がある．

図 6 の例では，*r2* と *r4* の間に SoD の制約があり，ユーザ *u1* はこの 2 つのロールに割り当てられている，といった矛盾が含まれている．この RBAC モデルに対し，検証を行う時相論理式は以下の式である．

[検証式]

$$!\,(EF(r = r2\ \&\ u = u_i)\ \&\ EF\,(r = r4\ \&\ u = u_i)\,)\quad (u_i \in U)$$

ここで u_i は検証対象となる具体的なユーザである．たとえば，*r=r2* かつ *u=u1* である状態と，*r=r4* かつ *u=u1* である状態が同時に存在するかどうかを検証でき，この検証式の真偽判定が偽となれば，矛盾が生じていることになる．

図 7 は，この検証式によって得られた検証結果である．偽が出力されているため，同一ユーザ(*u1*)が SoD 関係で結ばれているロール(*r2*，*r4*)に同時に割り当てられている，と解析することができる．

Properties database

Context	Index	Select	Value	Trace	Type	Property
Main						
	0	☑	False	1	CTL	!(EF (r = r2 & u = u1) & EF (r = r4 & u = u1))

図 7　SoD の検証結果

4．おわりに

本稿では，モデル検査ツールを用いて RBAC モデルを状態遷移システムに変換するための規則を定義し，RBAC モデルを状態遷移システムで表現することで NuSMV による検証を行い，矛盾や誤りの発見，修正の手掛かりとすることができた．

今後は，現実的な RBAC モデルへの適用実験を行うことによる性能評価や，検証項目の充実化を行っていく．

【参考文献】

[1] R. S. Sandhu, E. J. Coyne, H. L. Feinstein and C. E. Youman, ``Role-based access control models," IEEE Computer, Vol. 29, No. 2, pp. 38-47, 1996.

[2] NuSMV, http://nusmv.fbk.eu/.

[3] V. Gligor, S. Gavrila, and D. Ferraiolo, “On the formal definition of separation-of-duty policies and their composition,” in I. C. S. Press, editor, IEEE Symposium on Security and Privacy, pp.172–185, 1998.