# 複数情報資源の統合や第三者への提示を取り扱うアクセス制御モデル MP-RBAC と医療健康情報への適用

# MP-RBAC Access Control Model to Integrate and Disclose Multiple Data Resources to Third Parties and its Application to Healthcare Infomation

塩田哲哉 †, 飯島正 †
Tetsuya Shioda†, and Tadashi Iijima†
† 慶應義塾大学　理工学部
†Faculty of Science and Technology, Keio Univ.

**要旨**
インターネットを介したサービス連携を促進するにあたり，複数の情報資源を統合することと，その継続的な同期が重要となっている．また，個人情報の所有権は本来，その個人に持たせるべきであり，情報の第三者への提示など，情報の取り扱いを，その個人が制御できるべきである，という考え方も浸透しつつある．そこで，資源の作成/提供者と，その資源の本来の所有者，その資源の閲覧者 (第三者) といった役割属性を設け，それに基づいたアクセス制御モデルを提案し，医療健康情報への適用事例を示す．

## 1. はじめに

アクセス制御は情報システムやネットワークサービスで重要な役割を果たしている．ユーザの持つ条件に応じて閲覧・編集権限を設定しておき，ユーザのアクセス状況に応じてアクセスの可否の決定を下し，意図しない動作や不適切な動作の実行を防止することが可能である．個人に関わる情報資源は様々な機関により管理・運用されていることが多い．近年ではクラウドサービスにより情報資源を分散管理する機関も増えてきている．個人で自らの情報資源を全て自身の管理下に置くことはセキュリティやデータ管理の観点から現実的ではないが，少なくとも個人が了承した範囲で情報資源は利用されるべきである．アクセス制御を行うための手法として，Sandlu らにより提案された役割に基づくアクセス制御 (RBAC : Role Baced Access Control)[1] がよく知られている．RBAC ではユーザはロールと呼ばれる役割属性の集合に割り当てられ，そのロールに基いてアクセス制御を実施する．そのためユーザ毎にアクセス制御設定を実施する場合と比較してポリシの記述量が減少し，容易なアクセス制御が実現される．他にも様々なアクセス制御手法 [2][3] が提案され，実際に運用されている．本論文では個人が自らの情報資源を自己管理の下に集約し，活用するための RBAC を拡張したアクセス制御モデル MP-RBAC(Multi-Party RBAC) を提案する．基本的なアイデアはユーザをパーティと呼ばれる情報作成者/管理者・情報の本来の所有者・第三者のいずれかに割当，情報取扱者の観点から情報資源へのアクセス権を決定する．提案モデルの有用性は，個人健康情報 (PHR : Personal Health Record)[7] への適応例を示しながら議論をすすめる．

## 2.Role-Based Access Control

RBAC では，ユーザが役割を果たす上で必要最低限の権限を役割に付与することにより，ユーザ毎にポリシ設定を行う場合に比べてシンプルなポリシ設定が可能になり，アクセス制御が容易になる．それと同時に，ポリシ設定ミスや管理コストの削減につながっている．NIST RBAC[4] は以下のように定義される．

- U, R, P, Se, Obs, Ops は User, Roles, Permissions, Sessions, Objects, Operations 集合を表す.
- $P \subseteq Obs \times Ops$ は対象と命令の組み合わせで表現される.
- $URA \subseteq U \times R$ はユーザとロールを多対多で割り当てる.
- $RPA \subseteq P \times R$ はパーミッションとロールを多対多で割り当てる.
- $RH \subseteq R \times R$ はロールの階層構造であり，順序関係を持つ.
- $user\_session : Se \rightarrow U$ はセッションをユーザへのマッピングする.
- $session\_role : Se \rightarrow 2^R$ はセッションをロール集合へマッピングする.

# 3.MP-RBAC モデル

本章では，情報資源に関わる複数の情報取扱者を中心としたアクセス制御を実現するにおいて，まず RBAC の問題点を説明する．サンプルシナリオとして個人医療情報 PHR(Personal Health Record)[7] を取り扱う．そして，問題点を克服するため新たに MP-RBAC(Multy-Party RBAC) モデルを提案し，詳細定義や実装環境について記述する

## 3.1.PHR によるサンプルシナリオと問題点

PHR(Personal Health Record) は個人に関わる医療情報資源は自らの管理下に置き，自ら参照したり，第三者に公開することで自身の健康状態を把握・向上することを目的としている．PHR に関わるユーザは，医者や薬剤師のように医療情報を提供するユーザ，医療情報の記述対象である患者，患者の医療情報の利用を求める第三者からなる．PHR に対する操作は，閲覧・編集はもちろん，病院から患者への情報提供や患者による第三者への情報公開などがあげられる．ここでは簡単のために以下のようなシナリオを考える．

1. 病院 A の医師は患者 P に診察を行い，カルテを作成する．
2. 患者 P はセカンドオピニオンを求めるため，病院 A で作成されたカルテを病院 B へ提供する．
3. 病院 B の医師は病院 A で作成された患者 P のカルテを参照し，診察結果を追記する．

以上で示したシナリオでは，RBAC をそのまま適用するのは望ましくない．例えば，ある患者が病院関係者へ，自分を診察した医師や看護師以外にはカルテの診察部分を公開したくないと要求したとする．RBAC で定められるロール階層は機関内の役割に基いてに一意に定められるものであり，患者と医者との間に生じる診察関係までは記述しきれないため，RBAC のみでポリシを設定することは適当ではない．また，病院 A,B のように複数機関により資源が譲渡されたり管理運用される場合，ロールの重複が生じる可能性もある．さらに，病院で扱うカルテは膨大な量に渡るため，病院側が患者個人の要望に合わせてアクセス制御ポリシを設定することは作業量の点から現実的ではない．上記の問題を解決するため，RBAC の拡張アクセス制御モデル MP-RBAC を提案する．

## 3.2.Core MP-RBAC

MP-RBAC(Multy-Party RBAC) は，NIST RBAC を拡張したアクセス制御モデルである．概念図を図 1 に示す．複数の情報取扱者の異なる属性を扱うために，新たな要素であるパーティ属性 Pa を導入する．Pa は情報資源の作成者/管理者である DataProvider, 情報資源の所有者である DataOwner, 第三者である ThirdParty から構成される．まず，Core-RBAC の定義を加えてから，そのメカニズムについて説明を加える．

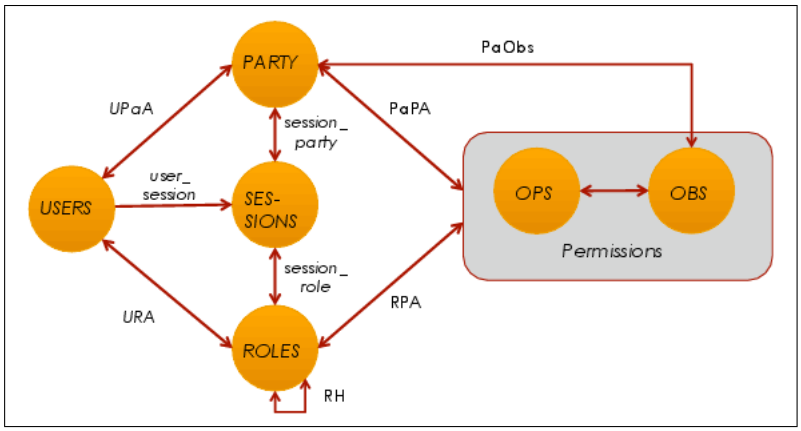


図 1: MP-RBAC

## 3.3.Core MP-RBAC の定義

本節では MP-RBAC が新たに持つ Core-RBAC の拡張要素を記述する．

- $Pa \leftarrow Obs \times U$：Party 集合を表す．Obs と U によって決定される．
  - Data Provider：情報の作成者/管理者．管理者は追加・削除が可能である．
  - Data Owner：情報の所有者．情報に記述されているユーザを示す．
  - Third Party：第三者．DataProvider と DataOwner 以外を示す．
- $UPaA \subseteq Pa \times U$ はパーティとユーザを多対多で割り当てる．
- $PaObs \subseteq Pa \times Obs$ はパーティと資源を多対多で割り当てる．
- $PaPA \subseteq Pa \times P$ はパーティとパーミッションを多対多で割り当てる．
- $session\_party : Se \rightarrow 2^{Pa}$ はセッションをパーティ集合へマッピングする．

### 3.4.MP-RBAC のメカニズム

MP-RBAC のアクセス制御は次のようにして行われる．ユーザ user はアクセスを求める資源 obs，及び動作 ops を入力する．user, obs 情報に基づき PartyDesignator がパーティが決定し，更にロールが RoleDesignator により決定される．user が DataProvider もしくは DataOwner ならば DataProvider が作成したポリシ，ThirdParty ならば DataOwner が編集したポリシが選択される．選択されたポリシに対し，role, obs, ops に合致する記述がある場合，アクセスは成功し true が出力される．そうでなければアクセスは失敗となる．

**Algorithm 1** Access Control Algorithm

**Input:** user, obs, ops (user ∈ U, obj ∈ Obj, ops ∈ Ops)

```
initialize candidate policy set PS = ∅
party = PartyDesignator(user, obs)
role = RoleDesignator(user, obs)
if party = (DataProvider ∨ DataOwner) then
    PS = PolicySet made by DataProvider
else
    PS = PolicySet made by DataOwner
end if
for all pol_i ∈ PS do
    if pol_i.role = role ∧ pol_i.ops = ops ∧ pol_i.obs = obs then
        return true
    end if
end for
return false
```

### 3.5.MP-RBAC における Party 属性の割り当て

本モデルでは DataProvider，DataOwner が編集可能である．

- DataProvider の追加・編集
  DataProvider は資源の作成者である．そのため資源が複数ユーザにより作成された場合は DataProvider は複数登録される．資源に対する変更が DataProvider 以外により成された場合，編集したユーザは編集された部分にのみ DataProvider として登録されることになる．また，DataProvider としての役割を放棄した場合には，DataProvider 属性は削除され，ThirdParty へと変更される．
- DataOwner の追加・編集
  DataOwner は資源に記述されているユーザを指し示す．資源に新たなユーザの情報が書き込まれた場合，そのユーザは書き込まれた部分に対してのみ DataOwner 属性を所有することになる．

### 3.6.MP-RBAC における 2 次公開ポリシ作成とその編集

DataProvider が情報を作成し，DataOwner へと公開するプロセスは 1 次公開である．2 次公開は「DataProvider,DataOwner が ThirdParty へ情報を公開すること」と定義する．

DataProvider が情報を ThirdParty である外部機関へ公開する場合，機関毎にロール名が異なる，もしくは同じであってもロールが果たせる役割が異なる場合が存在する．そのためロールへのパーミッションの割り当ては必要最低限度の．を持たせたポリシセットを予め作成し，特定の機関に強い権限を持つアクセスを許可する場合には，ロールに機関名を付加したものとしてアクセス制御を行う．例えば，ロール Doctor の EHR に対する操作は通常 read のみであるが，ある機関 A にのみ write の操作を許可する場合，ロールを Doctor から A.Doctor というように，機関名をつけて記述する．

DataOwner は自身の情報を ThirdParty へ公開する場合，DataProvider が定めたポリシを順守して情報公開を行う必要がある．そのため DataProvider がパーミッションを P(obs, read) と定めている場合，DataOwner は read, もしくは read より弱い権限を持つ操作を実行するようにポリシを編集することが出来る．read 以上の権限を要求する場合は，必ず DataProvider へ承認を求めなければならない．

### 3.7.MP-RBAC における 2 次公開メカニズム

3.6. の手続きに従って作成された 2 次公開ポリシの MP-RBAC での適用方法について記述する．

1. ユーザが DataProvider の場合
   - 自身の機関内へデータを公開する場合，承認は不要．
   - 2 次公開する場合，DataOwner の承認を受けてから DataOwner が編集したポリシに従い情報公開を行う．承認後は DataOwner の定めた Context に基づき公開が継続される．
   - DataOwner が予め DataProvider の作成した 2 次公開用ポリシを承認している場合，公開に際して予め承認が得られていることになり，新たな承認は不要である．
2. ユーザが DataOwner の場合
   DataProvider が作成した 2 次公開ポリシを編集したものに従い，情報公開を行う．
3. ユーザが ThirdParty の場合
   2 次公開は必ず DataOwner の承認が必要となり，承認を得た場合のみ 2 次公開が可能である．

### 3.8.MP-RBAC の利点

Pa 属性を導入することで，従来の RBAC が抱えるロールの一意的割当問題を解決することが出来る．あるユーザがある情報資源へアクセスを求める場合，従来の RBAC ではユーザが割り当てられているロールに従いアクセス制御が実施されていた．その他，RBAC の拡張モデルは多数存在し，[5] では現在のタスクに従いロールが決定し，[6] では時間によりロールが切り替わることでアクセス制御を行なっている．これらのモデルはユーザが置かれている状況によりロールが定められた．一方，本モデルはユーザとアクセス対象情報に基いてロールの変更，及び Pa 属性の決定が行われ，適用されるポリシが選択される．すなわち，情報取扱者と情報資源との関係でアクセス制御が実施されており，情報所有者の意思を尊重したアクセス制御が実施されることになる．

### 3.9. サンプルシナリオの解決

3.1. で述べた問題点に対して，MP-RBAC を適用する．まず，患者 P と病院 A の間に存在する診察関係を，診察時に記録する．DataProvider はカルテ作成者の病院 A，DataOwner はカルテが対象とする患者 P となる．次に，セカンドオピニオンを求める際は患者 P は病院 B へカルテを提供する．この時点では病院 B は ThirdParty であるため，病院 B は患者 P による公開ポリシによって部分的に閲覧可能となる．最後に，病院 B はカルテに診断結果を追記する．結果として，病院 B はカルテの DataProvider として登録されることになる．ただし，病院 A,B が DataProvider となるのはカルテの作成した部分のみであり，病院 A が病院 B の作成したカルテを見る場合は病院 B の公開ポリシによりアクセス制御が加えられることになる．以上のようにして，資源の譲渡・開示を行うことで DataProvider が追加されることになる．

また，患者個人のアクセス制御への要望に対しても答えることが出来る．DataProvider,DataOwner に対しては DataProvider が作成したポリシを，ThirdParty に対しては DataOwner が作成したポリシを適用することで，DataOwner となったユーザは自分の望みどおりに資源を第三者へ公開することができるようになる．さらに，DataProvider が ThirdParty に情報公開を行う場合を考える．DataProvider

は情報資源作成者であるが，作成した情報が DataOwner のものであるので，やはり第三者へ公開する場合は DataOwner が作成したポリシが適用される．また公開に際しては DataOwner の事前認証が必要となる．

# 4. 実装

## 4.1. アクセスポリシ記述言語

本アクセス制御モデルでは，アクセス制御ルールは対象情報の属性やアクセス環境因子，また利用者のロールや属性によって記述される．アクセス制御ポリシは情報資源との独立性を保つため，また外部アプリケーションとの連携の容易性や可読性の高さを考え，XML で記述する．我々はロールとオブジェクト間のパーミッションは XACML(eXtended Access Control Markup Language)[8] を用いて定義，ユーザとデータ，ロールの紐付けは我々が独自に定義した DODL(Data Object Description Language) を用いて定義しアクセス制御を行う．XACML は OASIS により NIST 標準化された XML ベースのアクセス制御ポリシ記述言語仕様である．アクセス制御ポリシは Subject, Action, Resource で記述され，特定状況下でのアクセスの可否を決定する．DODL は独自に定めた仕様で，情報資源, DataProvider, DataSubject, ロールで記述され，ユーザ，ロール，資源，パーティ属性を紐付ける役割を担っている．

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<DataObject DataId="Mary's EHR" DataLocation="http://www.iijima.ac.keio.ae.jp/sec/phr/ehr/mary_ehr.xml">
    <Providers>
        <Provider Id="hiyoshi@ae.keio.ac.jp" Role="Patient"/>
        <Provider Id="yagami@ae.keio.ac.jp" Role="Scholar"/>
    </Providers>
    <Owners>
        <Owner Id="shonan@ae.keio.ac.jp" Role="Nurse"/>
        <Owner Id="mita@ae.keio.ac.jp" Role="Manager"/>
    </Owners>
</DataObject>
```

図 2: Sample DODL

## 4.2. デモ

図 3 にアルゴリズム 1 を適用したユーザと資源によるアクセス制御機構を示す．ユーザは名前およびアクセスを求める資源を選択する．DODL によって Party 及び Role が割り当てられ，ユーザに適用されるポリシが選択される．選択されたポリシ及びロールにより XACML による RBAC が行われた結果，各資源に対するパーミッションが決定され，右下のパーミッションテーブルへ表示される．

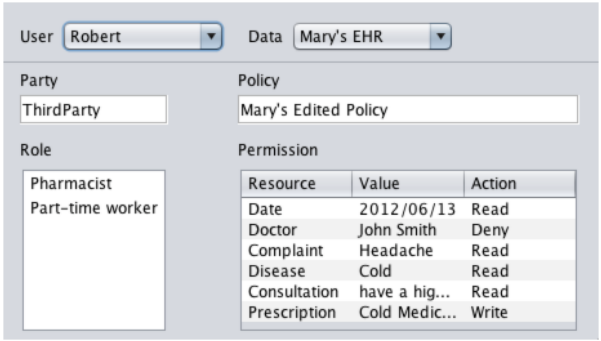


図 3: Sample of display on permission assignment function

# 5. 関連研究との比較

提案 MP-RBAC モデルと既存の RBAC モデルとの比較を [9] に基づき行った結果を表 1 に示す．Dynamic/Static はパーミッションが動的変更性の基準であり，本モデルはパーティ属性を用いることでポリシが動的に切り替わることで動的にパーミッションが変更される．RBAC はロールに割り当てられたパーミッションは変更されない．Fine Grained Control はきめ細かなアクセス制御の基準であり，本モ

デルはポリシを情報所有者が任意に変更することで，RBAC ではパーミッションが一意に割り当てられていた問題を解決していることできめ細かなアクセス制御が行われる．Easy to Use はエンドユーザのモデルの使いやすさを表す基準であり，本モデルはパーティ属性を導入することで RBAC に比べ運用に一段回理解が必要な時点で，RBAC より低くなっている．

表 1: 既存アクセス制御モデルとの比較

| Criteria | RBAC | MP-RBAC |
|---|---|---|
| Dynamic/Static | Static | Dynamic |
| Fine Grained Control | Low | High |
| Easy to Use | High | Medium |

次に，提案した MP-RBAC モデルは Kambiz らによる LPAAC[10] と類似している．本モデルでは RBAC の拡張としてロールを階層構造で表現しているのに対し，Kambiz らは目的，可視性，保持期間および粒度を束構造により表現している点で大きく異なる．さらに本モデルでは資源とアクセス主体の関係を用いることで，LPAAC よりアクセス対象情報に応じたきめ細かくアクセス制御が実施できる．

## 6. まとめ

本論文では，情報資源とそれに関わるユーザに対するアクセス制御モデル MP-RBAC の提案及び，アクセス制御エンジンおよび PHR システムの構築を行った．MP-RBAC モデルでは NIST RBAC の拡張モデルであるが，それは属性ベースでポリシが表現できる多くの利点を取り入れるためである．新たにパーティ属性を設けることで複数の情報取扱者とアクセス対象情報の関係に基づくアクセス制御の実施が可能になった．

## 参考文献

[1] R. S. Sandhu, E. J. Coyne, et al, ”Role-based access control models”, IEEE Computer, vol.29, no.2, pp.38-47, 1996.
[2] Sylvia Osborn, Ravi Sandhu, and Qamar Munawer, “Configuring role-based access control to enforce mandatory and discretionary access control policies”. ACM Transactions on Information and System Security (TISSEC). pp. 85-106, 2000.
[3] Ravi Sandhu, Qamar Munawer, “How to do discretionary access control using roles”. 3rd ACM Workshop on Role-Based Access Control. pp. 47-54, October, 1998.
[4] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli, ”Proposed NIST standard for role-based access control”, ACM Transactions on Information and System Security (TISSEC), vol.4, no.3, pp.225-274, August 2001.
[5] Arun Kumar, Neeran Karnik, Girish Chafle, ”Context sensitivity in role-based access control”, ACM SIGOPS Operating Systems Review, vol.36, no.3, pp.53-66, July, 2002
[6] Elisa Bertino, Piero A Bonatti, Elena Ferrari, ”TRBAC: A temporal role-based access control model”, ACM Transactions on Information and System Security (TISSEC) , vol.4, no.3, pp.191-233, August, 2001
[7] Paul C Tang, Joan S Ash, et al, ”Personal Health Records: Definitions, Benefits and Strategies for Overcoming Barriers to Adoption”, Journal of the American Medical Informatics Association, vol.13, no.2, pp.121-126, March/April 2006.
[8] Simon Godik, Tim Moses, et al, “eXtensible Access Control Markup Language (XACML) Version 1.0”, OASIS Standard, February 18th, 2003
[9] William Tolone, Gail-Joon Ahn, Tanusree Pai, Seng-Phil Hong, ”Access control in collaborative systems”, ACM Computing Serveys(CSUR), vol.37, no.1, pp.29-41, March 2005.
[10] Kambiz Ghazinour, Maryam Majedi, Ken Barker, ”A Lattice-based Privacy Aware Access Control Model”, International COnference on Computational Science and Engineering, pp.154-159, vol.3, 29-31 August, 2009.