デュアルラベルを利用したアクセス制御モデル

双紙正和† 加藤丈治† 前川守†

近年，コンピュータの価格性能比の飛躍的な向上により，コンピュータで処理可能な応用分野は急激に広がりつつある．それにともない，アクセス制御モデルに対する要請も複雑化・高度化してきている．しかしながら，従来のアクセス制御モデルはこのような状況に十分に対応できていない．このような状況を打開するためには，サブジェクトやオブジェクトのさまざまな関係やその状態などに基づいた柔軟なアクセス制御を可能にするアクセス制御モデルが必要になる．そこで，本論文では，以下のようなアクセス制御モデルを提案する．まず，権限や保護レベルの階層を表す静的ラベルと，現在の状態を表す動的ラベルとから構成される，デュアルラベルを定義する．そして，サブジェクトやオブジェクトにこのデュアルラベルを割り当て，それに基づいたアクセス制御を行う．さらに，そのデュアルラベルに基づいた認可の導出ルールを定義し，サブジェクトやオブジェクトの階層的な関係やその状態に基づいた認可の設定を容易に行えるようにする．

An Access Control Model Based on Dual Labels

MASAKAZU SOSHI,† TAKEHARU KATO† and MAMORU MAEKAWA†

In recent years, due to the drastic improvement of cost-performance ratio of computers, the area where computers play an important role becomes broader and broader. As a result, requirements to access control models tend to become more diversified and advanced ones than before. Unfortunately, traditional access control models cannot be fully applicable to such a situation. Therefore, in this paper we propose the access control model as follows. First, we define a *dual label* which consists of two labels: a *static label* representing the privilege of a subject or the security level of an object and a *dynamic label* representing the state of a subject or object. Next we assign a dual label to each subject and object, and we control an access of a subject to an object, based on their dual labels. Furthermore, we introduce derivation rules according to which authorization is derived from another in order to make it easier to specify authorizations based on the hierarchical relationships and states of subjects and objects.

1. はじめに

近年，コンピュータの性能の飛躍的な向上や分散環境の一般化などにより，コンピュータで処理可能な応用分野は急激に広がりつつある．それにともない，利用されるアプリケーションはますます高度化の一途をたどっており，同時に，アクセス制御モデルや機構に対する要請も複雑化・高度化してきている．しかしながら，従来のアクセス制御モデルはこのような状況に十分に対応できていない．そこで，本論文では，デュアルラベルに基づいたアクセス制御モデルを提案する．そのモデルによって，ユーザや保護されるべきコンピュータ資源のさまざまな関係やその状態などに基づいた，柔軟なアクセス制御を行うことが可能になる．

† 電気通信大学大学院情報システム学研究科
Graduate School of Information Systems, University of Electro-Communications

本論文の残りの部分は以下のように構成される．まず，2章において本研究の背景について述べ，現在アクセス制御モデルに対して求められる特徴のうち，重要なものについて議論する．そして，3章で，それらの特徴を持つ，デュアルラベルに基づいたアクセス制御モデルを提案する．その後，4章において，本論文で提案したモデルの適用事例を示す．さらに，5章で関連研究との比較を行い，最後に6章で結論を述べる．

2. 背景

この章では，本研究の背景について議論する．

最初に，以降の議論に用いられる基本的な用語について定義する．まず，ファイルやI/Oデバイスといった，セキュリティ制御機構によって保護される対象を**オブジェクト**（object）と呼び，ユーザやプロセスといった，オブジェクトをアクセスする能動的な主体を**サブジェクト**（subject）と呼ぶ．次に，サブジェク

トがオブジェクトに対して行うアクセスの種類を，アクセスモードと呼ぶ．アクセスモードの例としては，ファイルに対する read，write，append や，プログラムに対する execute などを考えることができる．さらに，サブジェクト s が，あるオブジェクト o に対してアクセスモード a によるアクセスを許可されているとき，3 つ組 (s, o, a) を認可（authorization）と呼ぶ☆．このような認可は，サブジェクトに対応する行と，オブジェクトに対応する列とからなるアクセス制御行列によって表現できる．そして，認可が存在するときに限り，該当するアクセスモードによるアクセスを実行することができる．現在多くのシステムが，このようなアクセス制御モデルに基づいてアクセスの制御を行っている．

しかしながら，従来のアクセス制御モデルでは，複雑化・高度化の一途をたどる現在のコンピューティング環境に十分に対応することができない．現在アクセス制御モデルに求められている特徴のうち，重要なものについて以下の節で述べる．

2.1 サブジェクト間の関係の表現

セキュリティポリシーの観点からは，サブジェクトが持つ権限やそれが行うオペレーションの間には，階層構造を仮定できることが多い．たとえば，ユーザやグループの階層[1] や，ロール（role）の階層[2),3)] などである．そこで，アクセス制御モデルは，サブジェクトのこのような階層構造を表現できることが望ましい．

2.2 オブジェクト間の関係の表現

2.1 節で述べたサブジェクトの関係と同様に，アクセス制御機構において保護されるべきオブジェクトの重要性や価値にも，さまざまな度合いや関係がある．本論文ではこれを一般的に，オブジェクトの保護レベルと呼ぶことにする．たとえば，情報流制御においては，オブジェクトの保護レベルを機密種別（classification）によって分類することが一般的である[4]．アクセス制御モデルは，このようなオブジェクトの関係を表現できる必要がある．

2.3 状態依存のアクセス制御

現実的なセキュリティポリシーを実現するためには，状態に基づいたアクセス制御が必要となることが多い．たとえば，Chinese Wall セキュリティポリシー[5] では，ある企業の機密情報が，競合する他の企業に漏洩することを防ぐようなアクセス制御を行うことがその目的であり，このためには，ユーザが行ってきたアクセスに基づいてアクセスの可否を判断するような制御が必要となる．Chinese Wall ポリシーは企業間の情報の流れを対象にしてる点で興味深く，現在までにさかんな研究が行われている[6),7)]．

このことから，サブジェクトやオブジェクトの状態に応じたアクセス制御は，現実のシステムでは有用であることが分かる．

2.4 ラベルやロールを用いたアクセス制御

コンピュータシステムが大規模化するにつれ，サブジェクトとオブジェクトとの数は莫大なものになり，一貫した認可の管理を行うことが困難になってきている．このような状況においては，ラベルやロールを利用するアプローチが有効である[3),8)]．すなわち，サブジェクトやオブジェクトの識別子を直接使ってアクセス制御を行うのではなく，サブジェクトやオブジェクトにラベルを割り当て，そのラベルに基づいてアクセス制御を行う．このようにして，必要な認可の設定の数を大幅に減らすことができ，また，認可の一貫した管理が容易になる．

2.5 認可の導出

ある研究開発プロジェクトにおいて，「一般研究員に対してあるデータへのアクセスが許可されたときは，必ず主任研究員にも同じアクセス権が与えられる」といったセキュリティポリシーが採用されているとする．このようなポリシーは，多くの組織で採用されている．

このようなポリシーに適したアクセス制御を行うためには，サブジェクトやオブジェクトの関係，アクセスの履歴，アクセスモードの関係などに基づいて，ある認可から他の認可を導出することができればよい[1)~3)]．上記の例では，サブジェクトの権限の階層を考え，その階層に従って，あるサブジェクトが別のサブジェクトの権限を継承するようにすればよい．このような権限の継承関係は，半順序関係によって表現できる．

2.6 正負の認可

従来のアクセス制御モデルは，認可が存在しないことで，それに対応するアクセスが不許可であるということを表現する．このようなアプローチにおいては，そのアクセスを明示的に不許可にしたい場合と，設定を行っていないために認可が存在しない場合とを区別できない．このとき，たとえば 2.5 節で述べたような認可の導出を行っていると，例外となる認可をうまく設定できないなどの問題が生じることがある[9)]．

このような問題を解決するためには，負の認可という概念[2),9)] を導入し，アクセスの不許可を明示的に設定できるようにすればよい（これに対し，今まで考えてきた認可はアクセスを許可するものであるため，正

☆ 本論文で提案するモデルでは，認可は 3.1 節で再定義される．

の認可と呼ばれる).

2.7 まとめ

今まで議論してきたように，現在，アクセス制御モデルに対する要求は多岐にわたっている．しかしながら，そのような要求をすべて満たすアクセス制御モデルはほとんど見当たらない．そこで本論文では，この節で議論してきたような特徴を持つアクセス制御モデルを提案する．

3. デュアルラベルを利用したアクセス制御モデル

この章は，本論文で提案する，デュアルラベルを利用したアクセス制御モデルを定義し，それについて議論を行う．

3.1 基本的な概念

最初に，基本的な概念について定義する．まず，サブジェクトの集合を S とし，オブジェクトの集合を O とする．サブジェクトをオブジェクトの部分集合として扱うことも多いが，ここでは $S \cap O = \phi$ と仮定する．サブジェクトとオブジェクトとをまとめて**実体**と呼ぶ．

次に，2.1 節および 2.5 節で議論したようなサブジェクトの権限の階層構造を表現するために，各サブジェクトに，ある固定のラベルを割り当てる．これをサブジェクトの**静的ラベル**と呼び，その有限集合を C_S とする．さらに，静的ラベルの間に半順序関係 $\leq_s$ を定義する．このようにして定義された半順序集合 $(C_S, \leq_s)$ を**サブジェクト階層**と呼び，$cs_1, cs_2 \in C_S$ でありかつ $cs_1 \leq_s cs_2$ であるとき，cs_2 は cs_1 に対して**優位** (dominate) であるという．ここで，サブジェクト s_2 の静的ラベルがサブジェクト s_1 の静的ラベルに対して優位であるとき，s_2 は s_1 の権限をすべて継承するものとする．

サブジェクト階層と同様にして，2.2 節で議論したような，**オブジェクト階層** $(C_O, \leq_o)$ を定義する．C_O は，オブジェクトの静的ラベルの有限集合であり，$\leq_o$ は 2 つのオブジェクトの静的ラベルの間に定義される半順序関係である．また，$co_1, co_2 \in C_O$ でありかつ $co_1 \leq_o co_2$ であるとき，co_2 は co_1 に対して優位であるという．オブジェクト o_2 の静的ラベルがオブジェクト o_1 の静的ラベルに対して優位であるとき，オブジェクト o_2 はオブジェクト o_1 よりも高い保護レベルを持つ．

さらに，2.3 節で述べたような状態依存のアクセス制御を行うために，状態を表すラベルをサブジェクトに割り当て，これをサブジェクトの**動的ラベル**と呼ぶことにする．サブジェクトの動的ラベルの有限集合を D_S とする．サブジェクトの動的ラベルは，サブジェクトが行った，あるいは受けたアクセスによって変更される．同様に，オブジェクトの動的ラベルを考え，その有限集合を D_O とする．オブジェクトの動的ラベルは，オブジェクトが受けたアクセスによって変更される．なお，ν を，実際には割り当てられることのない無効な動的ラベルと定義する．ここで，$\nu \notin D_S$, $\nu \notin D_O$ であり，以下では $D_S' = D_S \cup \{\nu\}$, $D_O' = D_O \cup \{\nu\}$ とおく．

以上の概念を用いて，**サブジェクト・デュアルラベル**を，

$$L_S = C_S \times D_S \tag{1}$$

のように定義し，また，**オブジェクト・デュアルラベル**を，

$$L_O = C_O \times D_O \tag{2}$$

のように定義する．さらに，E_L を，実際には割り当てられることのない無効なデュアルラベルと定義し，$E_L \notin L_S$, $E_L \notin L_O$ とする．E_L は，その動的ラベルが ν であるようなデュアルラベルに対応するものである．以下では $L_S' = L_S \cup \{E_L\}$, $L_O' = L_O \cup \{E_L\}$ とおく．

さらに，サブジェクトのデュアルラベルを返す関数 $dl_S : S \to L_S$ を定義し，同様に，オブジェクトのデュアルラベルを返す関数 $dl_O : O \to L_O$ を定義する．これらを用いて，実体のデュアルラベルを返す関数 $dl : S \cup O \to L_S \cup L_O$ を以下のように定義する：

$$dl(e) = \begin{cases} dl_S(e) & e \in S \text{ のとき} \\ dl_O(e) & \text{それ以外のとき.} \end{cases}$$

ここで，アクセスモードの有限集合を R とする．本論文では，$R = \{create, destroy, read, write, relabel\}$ とする（これらについては，3.2.2 項で議論する）．これを用いて，**正負のアクセスモード**の集合を $SR = \{+r, -r \mid r \in R\}$ と定義する[7]．SR は，以下で述べる，正負の認可を定義するのに用いられる．

本モデルにおける**認可**は以下のように定義される．

$$((cs_1, ds_1), (co_1, do_1), sr, ds_2, do_2) \tag{3}$$

または

$$((cs_1, ds_1), (cs_2, ds_2), sr, ds_3, ds_4) \tag{4}$$

ここで，$(cs_1, ds_1), (cs_2, ds_2) \in L_S$, $(co_1, do_1) \in L_O$, $ds_2, ds_3 \in D_S$, $ds_4 \in D_S'$, $do_2 \in D_O'$, $sr \in SR$ である．実際の認可の詳細は，3.2.2 項で議論する．

以下では，表記を簡単にするために，式 (3) と式 (4) とをまとめて，

$$((cs, ds), (cso, dso), sr, ds', dso') \tag{5}$$

と書くことがある．ここで，$(cso, dso) \in L_S$ かつ $dso' \in D_S{}'$ か，あるいは，$(cso, dso) \in L_O$ かつ $dso' \in D_O{}'$ である．

ある $r \in R$ について，$sr = +r$ であるような式 (3) および式 (4) を，**正の認可**と呼ぶ．また，ある $r \in R$ について，$sr = -r$ であるような式 (3) および式 (4) を，**負の認可**と呼ぶ．

ここで，ユーザやセキュリティ管理者が明示的に設定した認可を**明示的認可**，他の認可から導出される認可を**暗黙の認可**，その導出方法を記述したものを**導出ルール**と呼ぶことにする（導出ルールについては，3.3 節で詳しく議論する）．本論文で提案するアクセス制御モデルでは，まず最初にセキュリティ管理者が明示的認可を設定し，この後，システムが暗黙の認可を導出する．このようにして構成された認可の集合を，AS とする．また，特に断らない限り，AS は無矛盾であることを仮定する（無矛盾性の定義および AS を無矛盾にする方法については，3.4 節で述べる）．

3.2 アクセス制御

この節では，3.1 節で定義した基本概念を利用して，本論文で提案するアクセス制御モデルによるアクセス制御について定義する．

3.2.1 アクセス制御の基本

アクセス制御は，以下のとおりに行われる．

サブジェクト s が，実体 e に対するアクセスモード r のアクセスをリクエストすることを考える．ここで，$dl(s) = (cs, ds)$，$dl(e) = (cso, dso)$ と仮定する．このとき，$((cs, ds), (cso, dso), -r, ds', dso') \in AS$ のとき，r を実行することは許可されない．そのような負の認可が導かれず，$((cs, ds), (cso, dso), +r, ds', dso') \in AS$ のとき，そのリクエストは許可される．リクエストが許可され実行されたとき，s の動的ラベルは ds から ds' に変更され，また e の動的ラベルは dso から dso' へと変更される（**図 1** 参照）．

これから分かるように，本モデルは，従来モデルのようにサブジェクトとオブジェクトとの識別子を利用してアクセス制御を行っているわけではなく，(デュアル) ラベルに基づいた制御を行っている．ラベルを用いた制御を行うことによって，本モデルは 2.4 節で議論したような利点を持つことになるわけであるが，1 つのデュアルラベルに割り当てられる実体をただ 1 つだけに限定した場合は，従来モデルのように識別子を直接用いてアクセス制御を行う場合に等しくなる．つまり，識別子を直接利用するアクセス制御は，本モデルにおけるアクセス制御の単なる一例にしかすぎない．

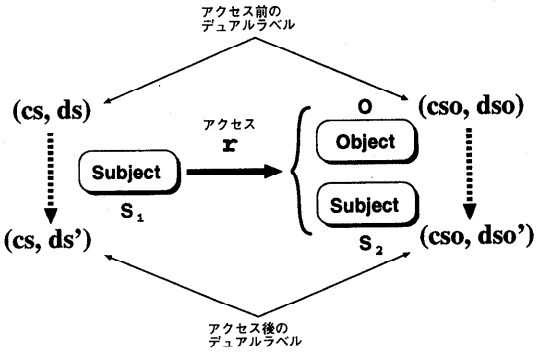


図 1 アクセス制御
Fig. 1 Access control.

3.2.2 認可の詳細

それぞれのアクセスモード $r \in R = \{create, destroy, read, write, relabel\}$ について，以下で正の認可の詳細について議論する．なお，それぞれの正の認可に対応する負の認可は，その認可で示されるアクセスを実行して該当する動的ラベルの変更を行うことが許可されないことを意味する．

- $((cs, ds), (cso, dso), +create, ds', \nu)$☆
  この認可が与えられるとき，$dl(s) = (cs, ds)$ であるようなサブジェクト s は，$dl(e) = (cso, dso)$ であるような実体 e を作り出すことが許可される．実行後の s の動的ラベルは ds' になる．$create$ 直後に e の動的ラベル dso を変更しても無意味なので，ここでは ν が用いられている．
- $((cs, ds), (cso, dso), +destroy, ds', \nu)$
  この認可が与えられるとき，$dl(s) = (cs, ds)$ であるようなサブジェクト s は，$dl(e) = (cso, dso)$ であるような実体 e を消去することが許可される．実行後の s の動的ラベルは ds' になる．$destroy$ された e の動的ラベル dso を変更するのは無意味なので，ここでは ν が用いられている．
- $((cs, ds), (cso, dso), +read, ds', dso')$
  この認可が与えられるとき，$dl(s) = (cs, ds)$ であるようなサブジェクト s は，$dl(e) = (cso, dso)$ であるような実体 e の持つ情報を読み込むことが許可される．実行後の s および e の動的ラベルは，それぞれ ds' および $dso' (\neq \nu)$ になる．
- $((cs, ds), (cso, dso), +write, ds', dso')$
  この認可が与えられるとき，$dl(s) = (cs, ds)$ であるようなサブジェクト s は，$dl(e) = (cso, dso)$

☆ この認可は，厳密には $((cs, ds), E_L, +create(cso, dso), ds', \nu)$ と書くべきであるが，本論文では他の認可との整合性や単純さを考慮して，このように書くことにする．

であるような実体 e に情報を書き出すことが許可される．実行後の s および e の動的ラベルは，それぞれ ds' および $dso'(\neq \nu)$ になる．

- $((cs,ds),(cso,dso),+relabel,ds',dso')$☆
  この認可が与えられるとき，$dl(s)=(cs,ds)$ であるようなサブジェクト s は，$dl(e)=(cso,dso)$ であるような実体 e の動的ラベルを，$dso'(\neq \nu)$ に変更することができる．実行後の s の動的ラベルは ds' になる．

relabel は，本論文で提案するモデルに特有のアクセスモードであり，以下に議論する．

サブジェクトの権限やオブジェクトの保護レベルは，状況に応じて変更される．そのため，従来のアクセス制御行列に基づくアクセス制御モデルは，grant（権限を与える）や revoke（権限を取り消す）といったアクセスモードを備えている．このようなアクセスモードを，統合的に 1 つのアクセスモードで表現するものが relabel である．

本論文で提案するアクセス制御モデルにおいては，read，write といった relabel 以外のアクセスモードも，動的ラベルを変更する性質を持っている．しかしながら，read，write のようなアクセスモードはそれぞれ固有の副作用をともなう．この一方で，relabel は純粋に動的ラベルの変更という機能しか持たない．そこでたとえば，状況に応じて，あるサブジェクトの権限の拡大・縮小を行ったり，あるオブジェクトの保護レベルの高低を変更したりするために，それらに対応する relabel の認可をセキュリティ管理者に与えるといったことが可能になる．

3.3 導出ルール

本論文で提案するアクセス制御モデルでは，2.5 節で議論したような認可の導出を行うことができる．以下では，認可 a_1 から認可 a_2 が導出されるとき，$a_1 \rightarrow a_2$ と記述する．

認可の導出は，サブジェクトやオブジェクトの静的ラベルおよび動的ラベルに基づいて行われる．以下にその詳細を述べる．

(1) サブジェクトの静的ラベルに関する導出ルール

導出ルール 1-a $\forall(cs_1,ds),(cs_2,ds) \in L_S$, $\forall(co,do) \in L_O$, $\forall r \in R$, $\forall ds' \in D_S$, $\forall do' \in D_O'$, $cs_1 \leq_s cs_2$:
$((cs_1,ds),(co,do),+r,ds',do') \rightarrow ((cs_2,ds),(co,do),+r,ds',do')$

導出ルール 1-b $\forall(cs_1,ds),(cs_2,ds) \in L_S$, $\forall(co,do) \in L_O$, $\forall r \in R$, $\forall ds' \in D_S$, $\forall do' \in D_O'$, $cs_2 \leq_s cs_1$:
$((cs_1,ds),(co,do),-r,ds',do') \rightarrow ((cs_2,ds),(co,do),-r,ds',do')$

導出ルール 1-a は，ある階層に所属するサブジェクトの正の認可は，その階層に対して優位となるような階層のサブジェクトへ伝搬していくことを表している．これとは逆に，導出ルール 1-b は，サブジェクトの負の認可は，優位でない階層のサブジェクトへ伝搬していくことを表している．

(2) オブジェクトの静的ラベルに関する導出ルール

導出ルール 2-a $\forall(cs,ds) \in L_S$, $\forall(co_1,do),(co_2,do) \in L_O$, $\forall r \in R$, $\forall ds' \in D_S$, $\forall do' \in D_O'$, $co_2 \leq_o co_1$:
$((cs,ds),(co_1,do),+r,ds',do') \rightarrow ((cs,ds),(co_2,do),+r,ds',do')$

導出ルール 2-b $\forall(cs,ds) \in L_S$, $\forall(co_1,do),(co_2,do) \in L_O$, $\forall r \in R$, $\forall ds' \in D_S$, $\forall do' \in D_O'$, $co_1 \leq_o co_2$:
$((cs,ds),(co_1,do),-r,ds',do') \rightarrow ((cs,ds),(co_2,do),-r,ds',do')$

導出ルール 2-a は，ある階層に所属するオブジェクトに対する正の認可は，その階層に対して優位ではないような階層のオブジェクトに伝搬していくことを表している．これとは逆に，導出ルール 2-b は，オブジェクトに対する負の認可は，優位な階層のオブジェクトに伝搬していくことを表している．

(3) サブジェクト/オブジェクトの動的ラベルに関する導出ルール

導出ルール 3-a $\forall(cs_1,ds_1),(cs_1,ds_2) \in L_S$, $\forall(cso_1,dso_1),(cso_1,dso_2) \in L_S \cup L_O$, $\forall ds_3 \in D_S$, $\forall dso_3 \in D_S \cup D_O$:
$((cs_1,ds_1),(cso_1,dso_1),+relabel,ds_2,dso_2) \wedge ((cs_1,ds_2),(cso_1,dso_2),+relabel,ds_3,dso_3) \rightarrow ((cs_1,ds_1),(cso_1,dso_1),+relabel,ds_3,dso_3)$

導出ルール 3-a は，あるサブジェクトが，ある実体に対して 2 回連続で relabel を適用できることが許可されているとき，それらと等価な 1 つの relabel を実行することが許可されるということを表している．つまり，導出ルール 3-a は，relabel の推移性を表現するルールである．

導出ルール 1-a，1-b，2-a，2-b，3-a をまとめて，以下のような導出ルールを定義できる．

導出ルール 1 $\forall(cs_1,ds),(cs_2,ds) \in L_S, \forall(co_1,do),$

☆ この認可は，厳密には $((cs,ds),(cso,dso),+relabel(dso'),ds',dso')$ と書くべきであるが，本論文では他の認可との整合性や単純さを考慮して，このように書くことにする．

$(co_2, do) \in L_O$, $\forall r \in R$, $\forall ds' \in D_S$, $\forall do' \in D_O'$, $cs_1 \leq_s cs_2$, $co_2 \leq_o co_1$:
$((cs_1, ds), (co_1, do), +r, ds', do') \rightarrow ((cs_2, ds), (co_2, do), +r, ds', do')$

導出ルール 2 $\forall (cs_1, ds), (cs_2, ds) \in L_S$, $\forall (co_1, do), (co_2, do) \in L_O$, $\forall r \in R$, $\forall ds' \in D_S$, $\forall do' \in D_O'$, $cs_2 \leq_s cs_1$, $co_1 \leq_o co_2$:
$((cs_1, ds), (co_1, do), -r, ds', do') \rightarrow ((cs_2, ds), (co_2, do), -r, ds', do')$

導出ルール 3 $\forall (cs_1, ds_1), (cs_1, ds_2) \in L_S$, $\forall (cso_1, dso_1)$, $(cso_1, dso_2) \in L_S \cup L_O$, $\forall ds_3 \in D_S$, $\forall dso_3 \in D_S \cup D_O$:
$((cs_1, ds_1), (cso_1, dso_1), +relabel, ds_2, dso_2) \wedge ((cs_1, ds_2), (cso_1, dso_2), +relabel, ds_3, dso_3) \rightarrow ((cs_1, ds_1), (cso_1, dso_1), +relabel, ds_3, dso_3)$

3.4 認可の競合

3.3 節で述べた導出ルールを用いると，同一の暗黙の認可が複数の認可から導出されてしまうことがある．これは，本モデルにおいては問題にならない．しかし，アクセス制御を行う際に，あるリクエストに対する実行許可/不許可を一意に決定できないような認可を導出してしまう場合は問題である．これを，**認可の競合**という．アクセス制御において，認可の競合はあってはならないものである．

この節ではまず認可の競合を具体的に定義し，次に，認可の競合をどのようにして解消するかを議論する．

3.4.1 認可の競合の定義

認可の競合とは，明示的な認可および暗黙の認可において，以下のうちいずれかの条件が成立することをいう：

競合 1 $\exists (cs, ds) \in L_S$, $\exists (cso, dso) \in L_S \cup L_O$, $\exists r \in R$, $\exists ds' \in D_S$, $\exists dso' \in D_S' \cup D_O'$:
$((cs, ds), (cso, dso), +r, ds', dso') \wedge ((cs, ds), (cso, dso), -r, ds', dso')$

これは，あるリクエストに対して，それに対応する正の認可と負の認可とが同時に存在している場合である．

競合 2 $\exists (cs, ds) \in L_S, \exists (cso, dso) \in L_S \cup L_O, \exists r \in R - \{relabel\}$, $\exists ds' \in D_S$, $\exists dso' \in D_S \cup D_O$:
$(((cs, ds), (cso, dso), +r, ds', dso') \wedge ((cs, ds), (cso, dso), -relabel, ds', dso')) \vee$
$(((cs, ds), (cso, dso), -r, ds', dso') \wedge ((cs, ds), (cso, dso), +relabel, ds', dso'))$

relabel 以外のアクセスモードへのリクエストに対する正の認可と，relabel に対する負の認可とが共存し（逆も同様），かつ，それらにおける動的ラベルの変更が一致するような状況は，そのような動的ラベルの変更を一方で許可し，他方で不許可にしていることになり，矛盾である．

競合 3 $\exists (cs, ds) \in L_S, \exists (cso, dso) \in L_S \cup L_O, \exists r \in R - \{relabel\}$, $\exists ds_1, ds_2 \in D_S$, $\exists dso_1, dso_2 \in D_S' \cup D_O'$:
$((cs, ds), (cso, dso), +r, ds_1, dso_1) \wedge ((cs, ds), (cso, dso), +r, ds_2, dso_2) \wedge \neg(ds_1 = ds_2 \wedge dso_1 = dso_2)$

あるリクエストに対する正の認可が複数（2 個以上）存在し，かつそれらにおけるアクセス後の動的ラベルの変更が一致せず，アクセス制御の挙動を一意に決定できないような状況である．競合 3 は，もし $r = relabel$ であれば矛盾ではない．

競合 4 $\exists (cs, ds) \in L_S$, $\exists r \in R$, $\exists ds_1, ds_2 \in D_S'$:
$((cs, ds), (cs, ds), +r, ds_1, ds_2) \wedge ds_1 \neq ds_2$

このとき，$dl(s) = (cs, ds)$ であるようなサブジェクト s は，自分自身に r というアクセスモードを実行することができるが，実行後の動的ラベルの変更が一意に設定できない．

上記のような認可の競合がない AS を，**無矛盾な**（consistent）AS という．

3.4.2 認可の競合の解消

今まで議論してきたような認可の競合は，アクセス制御の際にあってはならないものである．この項では，認可の競合をいかにして解消するかについて議論する．

認可の解消の方針はセキュリティポリシーごとに異なる．Jajodia らは，認可の競合の解消方針として，(1) 認可の競合を許さない，(2) 負の認可を優先する，(3) 正の認可を優先する，(4) 正負の認可のいずれも優先しない，といったものが考えられると主張している[7]．3.4.1 項で述べた認可の競合は，これらのいずれの方針によっても解消することができるが，紙面の関係から，ここでは 1 つの方針だけを述べる．

まず，認可の競合があった場合，一般的にいって，負の認可を優先することが安全であると考えられる[1]．そこで，負の認可を優先し，また，アクセス制御に条件を付加することで，認可の競合の解消を行うことができる．たとえば競合 2 では，$((cs, ds), (cso, dso), -r, ds', dso') \wedge ((cs, ds), (cso, dso), -relabel, ds', dso')$ とする．すなわち，$dl(s) = (cs, ds)$ であるようなサブジェクト s が，$dl(e) = (cso, dso)$ であるような実体 e に対して r または $relabel$ をリクエストしたとき，競合 2 が成立すれば，そのリクエストは不許可となる．同様にして，競合 1，競合 3，競合 4 も解消できる．

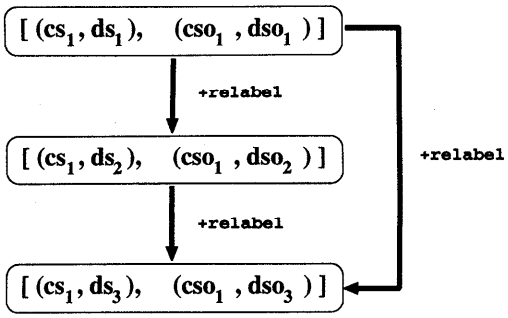


図 2 導出ルール 3-a
Fig. 2 Derivation rule 3-a.

3.5 認可グラフ

今まで本論文で提案するアクセス制御モデルのさまざまな側面について議論してきた．本モデルにおける AS は，**認可グラフ** $AG = (V_A, E_A)$ によって表現することができる．ここで，V_A と E_A とは以下のように定義される：

- V_A は，AG における頂点の集合である．各頂点はデュアルラベルの 2 つ組で，$V_A \subseteq \{[(cs, ds), (cso, dso)] \mid (cs, ds) \in L_S, (cso, dso) \in L_S' \cup L_O'\}$ と定義できる．
- E_A は，AG における 2 つの頂点 v_1, $v_2 \in V_A$ の間に定義される，正負のアクセスモード sr のラベル付けがなされた辺 $v_1 \xrightarrow{sr} v_2$ の集合で，以下のように定義される：
  $((cs, ds), (cso, dso), sr, ds', dso') \in AS$ のとき，$[(cs, ds), (cso, dso)] \xrightarrow{sr} [(cs, ds'), (cso, dso')] \in E_A$ である．ただし，$((cs, ds), (cso, dso), \pm create, ds', \nu) \in AS$, $((cs, ds), (cso, dso), \pm destroy, ds', \nu) \in AS$ のときは，それぞれ，$[(cs, ds), E_L] \xrightarrow{\pm create} [(cs, ds'), (cso, dso)] \in E_A$, $[(cs, ds), (cso, dso)] \xrightarrow{\pm destroy} [(cs, ds'), E_L] \in E_A$ である．

認可グラフの例を図 **2** に示す．辺 $[(cs_1, ds_1), (cso_1, dso_1)] \xrightarrow{+relabel} [(cs_1, ds_3), (cso_1, dso_3)]$ は，導出ルール 3 によって導出されたものである．

4. 例

この章では，本論文で提案するアクセス制御モデルの有用性を示すために，その適用例について考える．ここで述べる例は，文献 10) で述べられている "Document Release Example" を発展させたものである．

まず最初に，あるプロジェクトを想定し，そこで，project manager, engineer, security officer, project member というロールを考え，それぞれ cs_p, cs_e, cs_{so}, cs_m と書く．ここで，$cs_m \leq_s cs_e$, $cs_m \leq_s cs_{so}$, $cs_e \leq_s cs_p$, $cs_{so} \leq_s cs_p$ であり，これらの関係を図 **3** に示す．

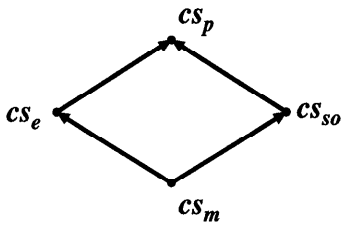


図 3 ロール束の例
Fig. 3 An example of role lattice.

ここで，プロジェクトのメンバが，そのプロジェクトの内容を文書にして公表するような状況を考える．公開文書の用意は，プロジェクトのメンバであればだれでも可能であるとする．ただし，公開する文書は，そのプロジェクトの機密情報を漏らさないように，security officer から承認を得たうえで初めて公表ができる．さらに，公正を期するため，security officer 自身が公開する文書を書いた場合は，security officer は自分ではその文書に承認を与えることはできず☆，project manager だけがその文書の公開を承認することができるものとする．また，project manager は，security officer が承認していない文書を自由に公開できる特権を持つものとする．

このときの明示的認可は以下のように与えられる．なお，以下では公開する文書の保護レベルを co_{doc} とする．ここで，記述量を減らすため，負の認可における '*' はすべての動的ラベルを表すものとする．

(1) $((cs_m, ds_1), (co_{doc}, do_1), +create, ds_2, \nu)$
(2) $((cs_m, ds_2), (co_{doc}, do_1), +read, ds_2, do_1)$
(3) $((cs_m, ds_2), (co_{doc}, do_1), +write, ds_2, do_1)$
(4) $((cs_m, ds_2), (co_{doc}, do_1), +relabel, ds_3, do_2)$
(5) $((cs_{so}, ds_4), (co_{doc}, do_2), +read, ds_4, do_2)$
(6) $((cs_{so}, ds_4), (co_{doc}, do_2), +write, ds_4, do_2)$
(7) $((cs_{so}, ds_4), (co_{doc}, do_2), +relabel, ds_5, do_3)$
(8) $((cs_p, ds_3), (co_{doc}, do_2), +read, ds_3, do_2)$
(9) $((cs_p, ds_3), (co_{doc}, do_2), +write, ds_3, do_2)$
(10) $((cs_p, ds_3), (co_{doc}, do_2), +relabel, ds_3, do_3)$
(11) $((cs_{so}, ds_3), (co_{doc}, do_2), -write, *, *)$
(12) $((cs_{so}, ds_3), (co_{doc}, do_2), -relabel, *, *)$

☆ これは，任務の動的な分離 (dynamic separation of duty) と呼ばれるものである[7]．

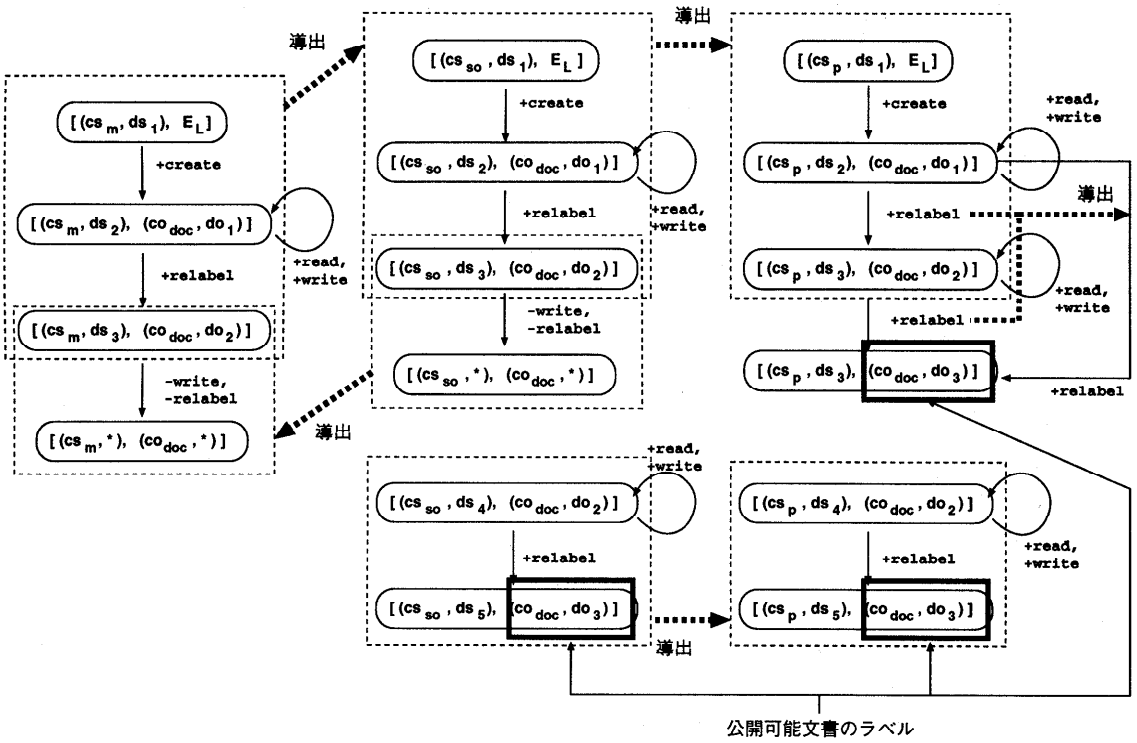


図 4 認可の関係の例
Fig. 4 An example of authorization relationship.

(13) $((cs_e, ds_3), (co_{doc}, do_2), -write, *, *)$
(14) $((cs_e, ds_3), (co_{doc}, do_2), -relabel, *, *)$

また，この例における認可グラフを図 4 に示す．ただし，図 4 では，engineer の記述は省略されている．

まず，認可 1 によって project member は文書を作成することが許可され，その文書を編集することが認可 2 と認可 3 とによって許可される．文書の編集を終えたとき，その文書の動的ラベルを do_2 に relabel することによって（認可 4），その文書は security officer がレビューしたり編集したりすることが可能になる（認可 5, 6）．認可 7 によって，security officer は，レビューおよび編集を終えた後その文書の動的ラベルを do_3 に変更することが許可される．こうして，デュアルラベル (co_{doc}, do_3) を持つ文書は公開可能となる．

ここで，3.3 節で述べた 導出ルール 1 によって，認可 1, 2, 3, 4 から暗黙の認可を導出することが可能になるため，project member だけでなく project manager, engineer, security officer も文書を作成・編集することができるようになっている．また，認可 5, 6, 7 と導出ルール 1 から，security officer と project manager が公開文書をレビューすることができる．また，認可 8, 9, 10 によって，project manager はその動的ラベルが ds_3 のときも文書をレビュー・編集できるようになる．このようなきめ細かな認可の設定が可能になる点が，本モデルにおける大きな特徴である．ここで，認可 4 から暗黙の認可 $((cs_p, ds_2), (co_{doc}, do_1), +relabel, ds_3, do_2)$ を導出できるが，これと認可 10 とにおいて導出ルール 3 を適用すると，$((cs_p, ds_2), (co_{doc}, do_1), +relabel, ds_3, do_3)$ を導出できる．このことは，project manager は文書を作成・編集し，security officer のレビューなしでその文書を公開できることを意味している．これによって，導出ルール 3 の有用性が分かる．

さらに，負の認可 11, 12, 13, 14 と導出ルール 2 によって，文書を作成した security officer, engineer, project member は，レビュー段階にある文書に変更を加えることができないようになっている．ここで，認可 6, 7 と認可 11, 12 との違いによって，文書を作成していない security officer のみがレビューできるようになっている．

5. 関連研究

本論文で提案したモデルでは，サブジェクトやオブ

ジェクトのさまざまな関係や，その状態に基づいた柔軟なアクセス制御が可能であり，導出ルールなども含めた統一的なモデル化が行われている．このようなモデルはほとんど見当たらない．しかし，本モデルのさまざまな側面については，部分的に関連する研究は多い．これらの代表的なものについて，この章で議論する．

文献2)では，ロール束，認可オブジェクト束，認可型束に基づいた導出ルールを提案しているが，そのモデルでは，実体の状態に依存した柔軟なアクセス制御を行うことができず，また検証も困難である．本論文で提案したモデルは，単純さを優先したために認可型束については考えていないものの，それを組み込むことは容易である．

Jajodia ら[7]は，さまざまセキュリティポリシーを記述するための論理的な言語，ASL（Authorization Specification Language）を考えた．ASL では，"Done Rule" によって，過去のある時点に行われたアクセスに基づいた制御方針を記述することができるものの，本モデルのように，過去に行われたアクセスの順序に基づいた制御を記述することはできない．

Foley ら[6]は，情報流制御モデルにおける，セキュリティラベルの変更に関する研究を行った．Foley らが提案したモデルでは，基本的にはセキュリティラベルの間に定義される束に従う形でしかラベルの変更を行うことはできない．しかしながら，本論文のモデルでは，動的ラベルの任意の変更を記述することができる．また，本論文のモデルは，固定のラベル（静的ラベル）と変更されるラベル（動的ラベル）とを厳密に区別しているため，ラベルの意味するセマンティクスはより明らかになり，直感的にも分かりやすいといえる．もちろん，Foley らが提案したさまざまなラベル変更ポリシーは，本論文で提案したアクセス制御モデルに適用することも可能である．

6. 結　　論

近年，コンピュータの性能の飛躍的な向上や分散環境の一般化などにより，コンピュータで処理可能な応用分野は急激に広がりつつある．それにともない，利用されるアプリケーションはますます高度化の一途をたどっており，同時に，アクセス制御モデルや機構に対する要請も複雑化・高度化してきている．しかしながら，従来のアクセス制御モデルはこのような状況に十分に対応できていない．

そこで，本論文では，サブジェクトやオブジェクトのさまざまな関係やその状態などに基づいた柔軟なアクセス制御を行うことを可能にするアクセス制御モデルを提案した．このために，本モデルにおいては，デュアルラベルが定義され，これを利用したアクセス制御が行われる．さらに，そのデュアルラベルに基づいた認可の導出が行われ，また，認可の競合の解消方法も提案されている．本モデルにおいては，それらは統一的にモデル化が行われており，このようなモデルはほとんど見当たらないといっていいだろう．

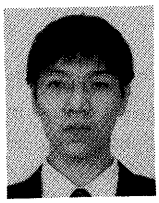
参 考 文 献

1) Bertino, E., Jajodia, S. and Samarati, P.: Supporting Multiple Access Control Policies in Database Systems, *Proc. IEEE Symposium on Security and Privacy*, pp.94–107 (1996).
2) Rabitti, F., Bertino, E., Kim, W. and Woelk, D.: A Model of Authorization for Next-Generation Database Systems, *ACM Trans. Database Syst.*, Vol.16, No.1, pp.88–131 (1991).
3) Sandhu, R.S., Coyne, E.J., Feinstein, H.L. and Youman, C.E.: Role-Based Access Control Models, *IEEE Computer*, Vol.29, No.2, pp.38–47 (1996).
4) Bell, D.E. and LaPadula, L.J.: Secure Computer Systems: Mathematical Foundations and Model, Technical Report ESD-TR-73-278-I, MITRE Corp., Bedford, MA (1973).
5) Brewer, D.F. and Nash, M.J.: The Chinese Wall Security Policy, *Proc. IEEE Symposium on Security and Privacy*, pp.206–214 (1989).
6) Foley, S.N., Gong, L. and Qian, X.: A Security Model of Dynamic Labeling Providing a Tiered Approach to Verification, *Proc. IEEE Symposium on Security and Privacy*, pp.142–153 (1996).
7) Jajodia, S., Samarati, P. and Subrahmanian, V.S.: A Logical Language for Expressing Authorizations, *Proc. IEEE Symposium on Security and Privacy*, pp.31–42 (1997).
8) Moffett, J., Sloman, M. and Twidle, K.: Specifying Discretionary Access Control Policy for Disributed Systems, *Computer Communications*, Vol.13, No.9, pp.571–580 (1990).
9) Woo, T.Y. and Lam, S.S.: Authorization in Distributed Systems: A Formal Approach, *Proc. IEEE Symposium on Security and Privacy*, pp.33–50 (1992).
10) Sandhu, R.S. and Suri, G.S.: Non-Monotonic Transformation of Access Rights, *Proc. IEEE Symposium on Security and Privacy*, pp.148–161 (1992).

(平成10年 5 月25日受付)
(平成10年11月 9 日採録)

双紙　正和（正会員）

1968年生．1993年，東京大学大学院理学系研究科情報科学専攻修了．1997年，電気通信大学大学院情報システム学研究科後期博士課程を単位取得退学後，現在まで同研究科助手．セキュリティモデル，分散システムの研究に従事．



加藤　丈治

1973年生．現在電気通信大学大学院情報システム学研究科情報システム設計学専攻博士前期課程に在学中．連合システムにおけるアクセス制御に関する研究に従事．

前川　　守（正会員）

1942年生．1965年，京都大学工学部数理工学科卒業．同年東京芝浦電気（株）入社．東京大学理学部情報科学科助教授等を経て，現在電気通信大学大学院情報システム学研究科教授．主として分散システム，ソフトウェア開発環境等の研究に従事．Ph.D.