オブジェクト指向データベースにおけるアクセス権モデルの設計

1Q－4

森多 俊之† 石原 靖哲† 関 浩之† 伊藤 実†
†奈良先端科学技術大学院大学 情報科学研究科

1 はじめに

オブジェクト指向データベース (OODB) のデータベース管理システム (DBMS) におけるデータ保護を実現するために，種々のアクセス権モデルが提案され始めている．本稿では，これまでに提案された OODB スキーマに適用可能なモデルを提案し，アクセス権の推論をできるだけ自由に記述できるような推論規則の記述言語を定義する．そして，与えられたアクセス要求が許可されるか否かを判定する効率的な判定方法を示す．

2 アクセス権モデル

2.1 アクセス権と推論規則

OODB におけるアクセス権は，一般に，誰（アクセス主体）がどのデータ（アクセス対象）に対しどのような操作（アクセスの型）を実行できるかを表す組（基本アクセス権と呼ぶ）の集合としてモデル化できる．例えば，アクセス権 $a=$ (Julia, Member, Delete) は，アクセス主体 Julia がアクセス対象 Memeber に対してアクセスの型 Delete の操作を行なうことを許可することを表す．また，アクセスの禁止を陽に表す否定アクセス権を導入する方法 [1]，肯定アクセス権と否定アクセス権による衝突を避けるために，各アクセス権に優先度を付ける方法 [3] 等が提案されている．一般に，アクセス主体，アクセス対象，アクセスの型それぞれについてクラス階層を自然に導入できることが多い．そこで，これらのクラス階層に沿って，アクセス権を推論する規則を記述することにより，アクセス権を簡潔に指定する方法も提案されている．例えば，文献 [2] の方法では，上述のアクセス権 a を Member のサブクラスに対しても与えたい場合，推論規則を用いて次のように記述できる:

$auth(\text{Julia}, X_o, \text{Delete}) :- X_o \underset{o}{\Rightarrow} \text{Member}$
$b\_auth(\text{Julia}, \text{Member}, \text{Delete}).$

2.2 アクセス権モデルの定義

アクセス主体，アクセス対象，アクセスの型についてのクラス階層の 3 字組を，$(C, \Rightarrow) = \langle (C_s, \underset{s}{\Rightarrow}), (C_o, \underset{o}{\Rightarrow}), (C_t, \underset{t}{\Rightarrow}) \rangle$ で表す．ただし，$\underset{s}{\Rightarrow}$ は，その反射推移閉包 $\underset{s}{\overset{*}{\Rightarrow}}$ が半順序であるような C_s 上の 2 項関係であり，$s \underset{s}{\Rightarrow} s'$ は s が s' の直接のサブクラスであることを表す（$s' \underset{s}{\Leftarrow} s$ とも書く．以下同様）．また，$\underset{s}{\Rightarrow}$ の推移閉包を $s \underset{s}{\overset{+}{\Rightarrow}} s'$ と書き，$s \underset{s}{\overset{+}{\Rightarrow}} s'$ は s が s' の直接または間接のサブクラスであることを表す．$\underset{o}{\Rightarrow}, \underset{t}{\Rightarrow}$ も同様である．優先度の有限集合 P が与えられたとき，アクセス権モデル $\langle A, R \rangle$ を次のように定義する．A はアクセス権 (s, o, t, δ, p) の有限集合である．(s, o, t) は基本アクセス権，$p(\in P)$ は優先度，そして $\delta = +$ なら肯定アクセス権，$\delta = -$ なら否定アクセス権を表す．A は次の (A1) を満たす：

(A1) $(a_1 = (s, o, t, \delta_1, p_1), a_2 = (s, o, t, \delta_2, p_2) \in A)$
$\rightarrow (a_1 = a_2)$

アクセス主体，アクセス対象，アクセスの型に関するクラスを表す変数の集合を V_s, V_o, V_t，アクセス権の許可/禁止を表す変数の集合を V_δ とする．R は次のような形式の推論規則の有限集合である：

$r : auth(s_1, o_1, t_1, \delta_1) :-$
$B_s, B_o, B_t, b\_auth(s_2, o_2, t_2, \delta_2).$ (*)

ただし，$s_1, s_2 \in C_s \cup V_s, o_1, o_2 \in C_o \cup V_o, t_1, t_2 \in C_t \cup V_t, \delta_1, \delta_2 \in \{+, -\} \cup V_\delta$ であり，次の (R1), (R2) を満たす：

(R1) $\delta_1 = \delta_2$
(R2) $(s_2 \in V_s \rightarrow s_1 = s_2) \wedge (o_2 \in V_o \rightarrow o_1 = o_2) \wedge$
$(t_2 \in V_t \rightarrow t_1 = t_2)$

上の $b\_auth(s, o, t, \delta)$ はある $p \in P$ に対して $(s, o, t, \delta, p) \in A$ であることを表す述語，B_s は C_s 上で定まる述語 $\underset{s}{\Rightarrow}, \underset{s}{\overset{+}{\Rightarrow}}, \underset{s}{\overset{*}{\Rightarrow}}, \underset{s}{\Leftarrow}, \underset{s}{\overset{+}{\Leftarrow}}, \underset{s}{\overset{*}{\Leftarrow}}$ の有限列である（B_o, B_t も同様）．ただし，$B_s = B_1, \cdots, B_n$ は次の条件を満たさなければならない：$B_i = X_i Q_i X_{i+1} (1 \leq i \leq n) \wedge X_1 \in C_s \wedge (X_2, \cdots, X_n \in C_s \cup V_s - \{s_1\}) \wedge (X_{n+1} = s_1 \in V_s)$．ここで，$Q_i \in \{\underset{s}{\Rightarrow}, \underset{s}{\overset{+}{\Rightarrow}}, \underset{s}{\overset{*}{\Rightarrow}}, \underset{s}{\Leftarrow}, \underset{s}{\overset{+}{\Leftarrow}}, \underset{s}{\overset{*}{\Leftarrow}}\}$．$B_o, B_t$ についても同様である．なお，アクセス権 $a \in A$ 自身を推論させたいときは，次のように推論規則を記述すればよい：

$auth(s_1, o_1, t_1, \delta_1) :- b\_auth(s_1, o_1, t_1, \delta_1).$

(*) の推論規則 r に対して，「$a_j \vdash_r a$」を「ある代入 θ と $p_j \in P$ が存在し，$\theta(B_s), \theta(B_o), \theta(B_t)$ が真，かつ $a_j = (\theta(s_2), \cdots, \theta(\delta_2), p_j) \in A, a = (\theta(s_1), \cdots, \theta(\delta_1), p_j)$ が成り立つ」と定義する．(A1) より，$\theta(s_2), \theta(o_2), \theta(t_2)$ が定まると，上の条件を満たす $\theta(\delta_2)$，p_j は高々一つに定まる．このことと (R1), (R2) より，次の補題が成り立つ．

補題 1 a をアクセス権，$r \in R$ とする．$a_j \vdash_r a$ を満たす $a_j \in A$ は高々一つである． □

アクセス要求 $req = (s, o, t)$ が与えられたとき，ANT_{req} を，

$\{a_j = (s_j, o_j, t_j, \delta_j, p_j) \in A | \delta_j \in \{+, -\}, p_j \in P,$
$\exists r \in R : a_j \vdash_r (req, \delta_j, p_j)\}$

と定義する．ANT_{req} のうち優先度が最大となるアクセス権 $(s_M, o_M, t_M, \delta_M, p_M)$ を求める．(req, δ_M, p_M) は，req を許可/禁止するアクセス権のうちで優先度が最大

An Authorization Model for Object-Oriented Databases
Toshiyuki Morita†, Yasunori Ishihara†, Hiroyuki Seki†, and Minoru Ito†
† Nara Institute of Science and Technology

のものを表す．$\delta_M = +$ なら req は許可され，$\delta_M = -$ または上式を満たす (req, δ_M, p_M) が存在しないなら req は禁止される．

3 アクセス権判定

3.1 概要

ユーザが問い合わせプログラムの実行中にデータベースをアクセスする操作を要求した場合，DBMS はそのアクセス要求が許可されるか否かを判定しなければならない（アクセス権判定）．C, $\Rightarrow$, A, R 全体の記述長を N とする．アクセス要求 req が与えられたときに A, R から req が許可されるか否を判定すると $\Omega(N^2)$ 時間必要となる．一方，考えられ得るすべてのアクセス要求の判定結果を予め計算し，表として記憶する方法が考えられる．この方法では，表の検索時間だけで解を得ることができる．しかし，この表の作成に $\Omega(N^4)$ 時間，得られた表の保存に$\Omega(N^3)$ 領域必要とし，実用的ではない．本稿では，アクセス権の推論を予め部分的に計算しておき，アクセス要求 req が与えられたときにこの情報を用いて req が許可されるか否かを判定する方法を提案する．

3.2 判定方法

アクセス要求を $req = (s, o, t)$ とすると，2.2 節より，ANT_{req} を求めればよい．任意の $r \in R$ に対して，

$$ANT^r_{req} = \{a_j = (s_j, o_j, t_j, \delta_j, p_j) \in A \mid \delta_j \in \{+,-\}, p_j \in P, a_j \vdash_r (req, \delta_j, p_j)\}$$

と定義すると，

$$ANT_{req} = \bigcup_{r \in R} ANT^r_{req}$$

かつ，補題 1 より，$|ANT^r_{req}| \leq 1$ である．

まず，R に属する $(*)$ の各推論規則 r に対し，$ARG(r) \subseteq \{1,2,3\}$ を以下のように定める:

$$1 \in ARG(r) \Longleftrightarrow s_1 = s_2$$
$$2 \in ARG(r) \Longleftrightarrow o_1 = o_2$$
$$3 \in ARG(r) \Longleftrightarrow t_1 = t_2$$

各 $s \in C_s$ に対して，$INF(s) \subseteq R$ を以下のように定める: $(*)$ の各推論規則 r に対して，

$r \in INF(s) \Longleftrightarrow$「$s_1 \neq s_2$，かつ，ある代入 θ が存在し，$\theta(B_s)$ が真かつ $\theta(s_1) = s$」.

$INF(o), INF(t) \subseteq R (o \in C_o, t \in C_t)$ も同様に定義する．また，T_ϵ, T_1, T_{12} 等を次のように定める:

- $ARG(r) = \phi$ のとき．(R2) より $s_2, o_2, t_2 \in C$ である．
  「$\exists \delta_j, p_j : a_j = (s_2, o_2, t_2, \delta_j, p_j) \in A$」ならば $T_\epsilon(r) = \{a_j\}$ (補題 1 よりこのような a_j が存在すれば一意に定まる．以下同様),
  さもなければ $T_\epsilon(r) = \phi$.
- $ARG(r) = \{1\}$ のとき．(R2) より $o_2, t_2 \in C$ である．
  「ある代入 θ が存在し，$\theta(B_s)$ が真かつ $\theta(s_2) = s$, かつ $\exists \delta_j, p_j : a_j = (s, o_2, t_2, \delta_j, p_j) \in A$」ならば $T_1(r, s) = \{a_j\}$,
  さもなければ $T_1(r, s) = \phi$.
- $ARG(r) = \{1,2\}$ のとき．(R2) より $t_2 \in C$ である．
  「ある代入 θ が存在し，$\theta(B_s), \theta(B_o)$ が真かつ$\theta(s_2) = s, \theta(o_2) = o$, かつ $\exists \delta_j, p_j : a_j = (s, o, t_2, \delta_j, p_j) \in A$」ならば $T_{12}(r, s, o) = \{a_j\}$,
  さもなければ $T_{12}(r, s, o) = \phi$.
- $ARG(r) = \{1,2,3\}$ のとき．
  「ある代入 θ が存在し，$\theta(B_s), \theta(B_o), \theta(B_t)$ が真かつ $\theta(s_2) = s, \theta(o_2) = o, \theta(t_2) = t$, かつ $\exists \delta_j, p_j : a_j = (s, o, t, \delta_j, p_j) \in A$」ならば $T_{123}(r, s, o, t) = \{a_j\}$,
  さもなければ $T_{123}(r, s, o, t) = \phi$.
- 他の場合も同様.

定理 1 各 $T_\epsilon(r)$, $T_1(r, s)$, $T_{12}(r, s, o)$, ..., $INF(s)$, $INF(o)$, $INF(t)$ は，全体で $\mathcal{O}(N^2)$ 時間で計算できる．□

なお，アクセス権や推論規則の追加，削除などを行なう場合，INF, T を局所的に変更するだけで良い．

アクセス要求 $req = (s, o, t)$ が与えられたとき，INF, Tを用いて，ANT_rを以下のように表すことができる:

- $ARG(r) = \phi$ のとき．

$$ANT^r_{req} = \begin{cases} T_\epsilon(r) & r \in INF(s) \wedge INF(o) \wedge INF(t) \text{ のとき,} \\ \phi & \text{それ以外.} \end{cases}$$

- $ARG(r) = \{1\}$ のとき．

$$ANT^r_{req} = \begin{cases} T_1(r, s) & r \in INF(o) \wedge INF(t) \text{ のとき,} \\ \phi & \text{それ以外.} \end{cases}$$

- $ARG(r) = \{1,2\}$ のとき．

$$ANT^r_{req} = \begin{cases} T_{12}(r, s, o) & r \in INF(t) \text{ のとき,} \\ \phi & \text{それ以外.} \end{cases}$$

- $ARG(r) = \{1,2,3\}$ のとき．
  $ANT^r_{req} = T_{123}(r, s, o, t)$
- 他の場合も同様.

定理 2 アクセス要求 req が与えられたとき，ANT^r_{req} は INF, T を用いて $\mathcal{O}(N)$ 時間で求められる．また，各 ANT^r_{req} から ANT_{req} は $\mathcal{O}(N)$ 時間で計算できる．□

4 おわりに

今後は，推論規則によって述語 $auth$ が再帰的に定義される場合について検討する予定である．

参考文献

[1] E. Bertino and P. Samarati, *"Research Issues in Discretionary Authorizations for Object Bases,"* Proc. OOPSLA-93 Conference Workshop on Security for Object-Oriented systems, pp.183–199, 1994.

[2] E. Bertino and H. Weigand, *"An Approach to Authorization Modeling in Object-Oriented Database Systems,"* Data and Knowledge Engineering, 12, pp.1–29, 1994.

[3] H. H. Brüggemann, *"Object-Oriented Authorization,"* Advances in Database Systems - Implementations and Applications, CISM 347, Springer-Verlag, pp.139–160, 1994.