オブジェクトの意味に基づいたアクセス制御方式 *

5O－10

村井 洋成, 立川 敬行, 桧垣 博章, 滝沢 誠
東京電機大学
‡E-mail {hiro, tachi, hig, taki}@takilab.k.dendai.ac.jp

1 はじめに

現在、インターネットの普及に伴い、情報システムは、複数の計算実体が通信網によって相互接続された分散型の形態となっている。各実体は、実体内のデータを操作するための操作演算を提供する。複数の実体のグループ通信が協調動作することにより、分散計算が行なわれる。グループ内の実体間の協調動作では、実体間での不正な情報流を防止することが重要となる。これまで、実体に与えられた安全性クラス間の流関係により、実体に対する *read* と *write* アクセスを制御する強制アクセス制御 [2,3] が示されている。本論文では、実体に抽象的な操作演算を提供する場合の強制アクセス制御を論じる。また、利用者が利用できる操作演算は実体が属しているグループによって規定される。グループ内での実体が実行できる演算の集合をグループ内での実体の役割 (role) とする。実体の役割を制御することにより情報の流れを制御することを考える。

第2章では、アクセス制御モデルを概観する。第3章では、実体に対する意味的な強制アクセス制御を論じる。第4章では、グループ内での役割について述べる。

2 アクセス制御のモデル

2.1 アクセス制御

情報システムは、オブジェクト集合 O とサブジェクト集合 S から構成される。オブジェクトは、ある操作演算を実行する実体である。サブジェクトは、オブジェクトに対して、操作演算の実行を指示する実体である。サブジェクトとオブジェクトの関係は相対的である。オブジェクトに対する演算の集合を R としたとき、アクセス規則集合 R は、$R \subseteq S \times O \times T$ であり、R 内の各組をアクセス規則という。アクセス規則 (s, o, t) は、誰 (s) が, 何 (o) を, どのように (t) アクセスしてよいかを示す。

2.2 強制アクセス制御

2.2.1 束モデル

システム内のオブジェクトとサブジェクトをあわせて実体とよぶ。実体 e には、安全性クラス λ(e) が付与される。ここで、S を安全性クラスの集合、E をシステム内の実体の集合とすると、$\lambda : E \rightarrow S$ である。

[定義] S 内の任意のクラス s_1 と s_2 に対して、s_1 の情報をクラス s_2 に流すことができるとき、$s_1 \rightarrow s_2$ とする。□

二つの実体 e_1 と e_2 に対して、$\lambda(e_1) \rightarrow \lambda(e_2)$ であり、$\lambda(e_1) \not\rightarrow \lambda(e_2)$ とする。このとき、e_1 内の情報を e_2 に流すことができるが、逆は行なえない。$s_1 \rightarrow s_2$ で $s_2 \rightarrow s_1$ であるとき、s_1 と s_2 は同値である ($s_1 \equiv s_2$) とする。

[定義] S 内の任意のクラス s_1 と s_2 に対して、$s_1 \rightarrow s_2$ であり、$s_2 \not\rightarrow s_1$ ならば、そのときに限り $s_1 \prec s_2$ (s_2 は s_1 を支配する、また s_2 は s_1 より安全性が高い) とする。□ $s_1 \prec s_2$ または $s_1 \equiv s_2$ のときは、$s_1 \preceq s_2$ とする。半順序集合 $\langle S, \preceq \rangle$ は束 [1,2] を構成し、$\cup$ と $\cap$ を各々、*lub* と *glb* とする。二つの実体 e_1 と e_2 内の情報を流すことのできる共通の実体 e に対して、$\lambda(e_1) \cup \lambda(e_2) \preceq \lambda(e)$ でなければならない。一方、e_1 と e_2 の両方に情報を流すことのできる実体 e に対して、$\lambda(e) \preceq \lambda(e_1) \cap \lambda(e_2)$ でなければならない。

2.2.2 強制アクセス制御モデル

次に、実体をサブジェクトとオブジェクトに分けて考える。サブジェクト s によるオブジェクト o のアクセスを、支配関係 $\preceq$ により、制御することを考える。ここでは、o の操作演算として、*read*, *write*, *modify* を考える。情報流の考えでは、二つのクラス s_1 と s_2 があるとき、$s_1 \preceq s_2$ ならば、s_1 の情報を s_2 に流すことができる。o を s が *read* することを考える。*read* により、o 内の情報が s に流れる。従って、$\lambda(s) \succeq \lambda(o)$ ならば、s は o を *read* できることになる。s による o の *write* は、逆に s 内の情報が o に流れることになる。このためには、*read* とは逆に、$\lambda(s) \preceq \lambda(o)$ でなければならない。*modify* とは、o の *read* を行ない、その値を他の値に変更して、*write* することである。従って、$\lambda(s) \succeq \lambda(o)$ でかつ $\lambda(s) \preceq \lambda(o)$ であるので、$\lambda(s) \equiv \lambda(o)$ でなければならない。以上より、強制 (mandatory) アクセス規則 [2] は以下のように与えられる。

[強制アクセス規則]

1. $\lambda(s) \succeq \lambda(o)$ ならば、s は o を *read* できる。
2. $\lambda(s) \preceq \lambda(o)$ ならば、s は o に *write* できる。
3. $\lambda(s) \equiv \lambda(o)$ ならば、s は o を *modify* できる。□

3 意味的強制アクセス制御

3.1 単純演算

強制アクセス制御では、オブジェクトの演算として、*read* と *write* といった物理的な演算が考えられている。これに対して、実際の分散型システムを構成する実体は、より抽象的な演算を提供している。ここでは、オブジェクト o が提供する演算を $op_1, \cdots, op_n$ とする。各演算 op_i は、以下の点により特徴付けられる。

(1) 入力 (I_i)
(2) 状態遷移
(3) 出力 (O_i)

サブジェクト s が、o_i に op_i の実行を要求し、o_i が op_i を実行する。op_i は、入力データ I_i をもとに、o_i で計算が行なわれ、あるデータ O_i を出力する。このとき、o_i の状態が変化する場合もある。情報流の考えから、入出力のデータ、状態遷移を考えると、以下となる。

(1) o_i が op_i により状態遷移するとき、s から入力データ I_i が o_i に流入する。
(2) op_i が出力 O_i を持つとき、o_i 内の情報が s に流出する。

以上からオブジェクト o_i の演算 op_i を以下の四種に分類する。[図 1]

(1) 無流型, (2) 流出型, (3) 流入型, (4) 流入出型.

無流型の演算 op_i では、o_i への情報の流入, 流出がない。o_i の状態遷移はない。流出型の演算 op_i は、o_i の情報がデータ O_i として出力される。o_i の状態遷移はない。*read* がこの種類である。流入型と流入出型の演算は、o_i の状態を遷移させる。op_i が流入型の場合には、入力データ I_i により、o_i の状態が変化するもので

*Access Control based on Objects
†Hironari Murai, Takayuki Tachikawa, Hiroaki Higaki, Makoto Takizawa
‡Tokyo Denki University

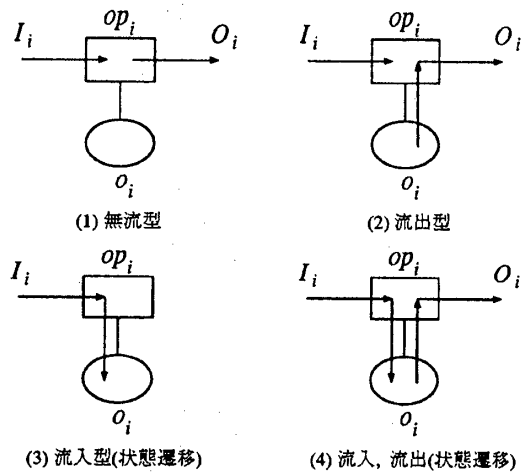

図 1: o_i への流入と流出

ある。*write* がこの種類の演算である。最後の流入出型の演算 op_i では、入力データ I_i により情報の o_i への流入があり、出力データ O_i により o_i への情報の流出がある。

以上をもとにして、強制アクセス規則を拡張した意味的強制アクセス規則を以下に与える。

[意味的な強制アクセス規則]

実体 s が o を演算 op_i により操作するとする。このとき、以下の条件を満足するときに、op_i を o に適用できる。

(1) op_i が流出型 : $\lambda(s) \succeq \lambda(o)$,
(2) op_i が流入型 : $\lambda(s) \preceq \lambda(o)$,
(3) op_i が流入出型 : $\lambda(s) \equiv \lambda(o)$, □

3.2 入れ子型演算

ここまでは、オブジェクト o_i の演算 op_i は、o_i のみを操作し、出力データをサブジェクト s に返すものであった。これに対して、op_i がサブジェクトとして、更に他のオブジェクトの演算を、実行する場合がある。このような演算を入れ子型という。ここで、実体 o_i の演算 op_i を考える。op_i の入力と出力を各々、I_i と D_i とする。実体により実行された op_i は、他の実体 op_{jh} の演算 op_{jh} $(h = 1, \cdots, l_i)$ を実行するとする [図 2]。このときのアクセス制御について考える。オブジェクト間での情報の流れ関係を考えるために、以下のグラフ (流グラフ) G を考える。

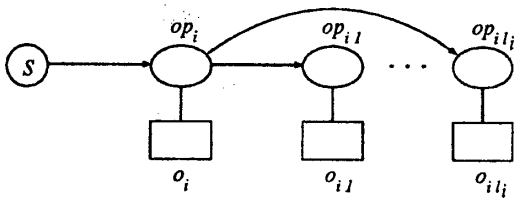

図 2: 入れ子型

[流グラフ]

演算 op_i と op_{ij} に対して、節点を設ける。節点間に以下のように有向辺を設ける。

(1) op_{ij} が無流型の場合には、op_i と op_j 間には辺を設けない。
(2) op_{ij} が流入型の場合には、$op_i \rightarrow op_j$ とする。
(3) op_{ij} が流出型の場合には、$op_i \leftarrow op_j$ とする。
(4) op_{ij} が流入出型の場合には、$op_i \leftrightarrow op_j$ とする。

このとき、以下の規則を考える。

[入れ子型強制アクセス規則]

s_i と o_i 間に意味的強制アクセス規則が成り立っているとする。

(1) op_i が流入型のとき、$op_i \rightarrow op_{ij}$ なるすべての op_{ij} について、op_i と op_{ij} 間に意味的強制アクセス規則が成り立ち、かつ $\lambda(s) \preceq \lambda(o_{ij})$ である。
(2) op_i が流出型のとき、$op_i \leftarrow op_{ij}$ なるすべての op_{ij} について、op_i と op_{ij} 間に意味的強制アクセス規則が成り立ち、かつ $\lambda(s) \succeq \lambda(o_{ij})$ である。
(3) op_i が流入出型のとき、$op_i \rightarrow op_{ij}$, $op_i \leftarrow op_{ij}$, $op_i \leftrightarrow op_{ij}$ について、op_i と op_{ij} 間に意味的強制アクセス規則が成り立ち、かつ $\lambda(s) \equiv \lambda(o_{ij})$ である。

4 グループでの役割制御モデル

実体 $e_1, \cdots, e_n$ の集合をグループ G とする。G 内の $e_1, \cdots, e_n$ で協調動作を行なう。このときの実体間でのアクセス規則について考える。実体 e_j は、データ構造 δ_j, 操作演算の集合 π_j, インテグリティ制約 ϵ_j の三つ組により表される。すなわち、$o_j = \langle \delta_j, \pi_j, \epsilon_j \rangle$。$e_j$ は、π_j 内の演算を用いて操作される。また、e_j は他の実体 e_k を π_k 内の演算により操作する。複数の実体 $e_1, \cdots, e_n$ はグループ G を構成し、互いに他の実体を操作しながら協調動作を行なう。

グループ G のスキーマを以下のように定義する。

$$G = \langle G_1, \cdots, G_n \rangle \mid \{G_1, \cdots, G_n\} \mid R \mid G^* \mid G^+$$

ここで、R は役割 (role) を示す。G^* は G 内の要素の集合を示し、G^+ は G 内の要素の組を示すとする。$[\![G]\!]$ はスキーマ G の例である。

役割とは、サブジェクトがオブジェクトに対して行なえる操作演算の集合である。従来の役割 [図 3(a)] のように、役割間での排他制御, 上下関係を定めるだけではなく、互いに関連付いている役割 [図 3(b)] を考える。

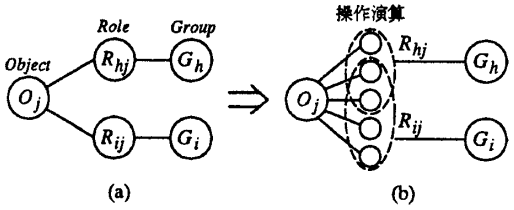

図 3: グループの役割

5 まとめ

本研究では、強制アクセス制御を一般の実体に適用できるように拡張した意味的強制アクセス規則について論じた。

参考文献

[1] Denning, D. E., "Cryptography and Data Security," Addison-Wesley Publishing Company, 1945, pp.191-327.
[2] Sandhu, R S., "Role-Based Access Control Models," IEEE *Computer*, Vol.29, No.2, 1995, pp.38-47.
[3] Sandhu, R S., "Lattice-Based Access Control Models," IEEE *Computer*, Vol.26, No.11, 1993, pp.9-19.
[4] Takizawa, M. and Mita, H., "Secure Group Communication Protocol for Distributed Systems," *Proc. of the IEEE COMPSAC'93*, 1993, pp.159−165.