Vol.2014-IOT-27 No.18
Vol.2014-SPT-11 No.18
2014/10/10

管理権限を一般ユーザにも移譲できるグループ管理システム

清水さや子[†1][†2]　戸田勝善[†2]　岡部寿男[†1]

統合 ID を使用するサービスのアクセス制限には，中央の認証サーバ上のグループや所属などの属性を指定することが多いが，そこには必ずしも必要とする属性が含まれているわけではない．そのため，著者らは各サービスのアクセス制限などで必要とするユーザの集合であるグループを，統合 ID とユーザ属性を用いて，各サービスの管理者などが作成し管理できるシステムを検討中である．グループを管理する際，グループ管理で広く用いられている Grouper システムでは，ユーザ属性に対して管理権限を持っている人にグループ管理の権限を委譲し，その範囲で管理を行う．しかし，著者らは，ユーザ属性の管理権限が与えられていない一般ユーザにも，グループを作成しメンバ管理ができるよう，グループの体系化を行っている．そして，体系化したグループの管理を実際にシステムで実現する際，システム上におけるグループの操作の在り方が重要となる．そこで，本稿では，グループ操作の在り方を中心に検討し，グループ管理システムを実現することで，グループ管理者やシステム管理者に対する管理負荷を軽減する．

A Group Management System that can Delegate Administrative Privileges to an Ordinary User

SAYAKO SHIMIZU[†1][†2]　MASAYOSHI TODA[†2]
YASUO OKABE[†1]

Services that use an integrated ID, when performing an access limited by the attributes, such as group affiliation of the authentication server of the center, detailed attributes that need not have necessarily included. Therefore, a group a set of users that require access restrictions of each service, using user attributes and integrated ID, we have been considering a system that can create and manage groups such as an administrator of each service as needed. When managing a group, in the Grouper system widely used in group management, delegated authorities of a group management to people who have administrative privileges on user attributes, and manages in that range. However, authors, so that members management can create a group, general users administrative privileges of the user attribute is not also given, has made a systematization of group. Then, when realizing the system actually, a management of a group that was systemizes, way of the operation of a group for system is important. In this paper, by using discussed centered on a role of group operations, to achieve a group management system, to reduce administrative burden on the system administrator or group administrator.

1. はじめに

近年，ID を統合化し，統合認証基盤の整備をする組織が増加している 1）2）．統合 ID を使用するサービスのアクセス制限の方法はサービスごとに行うため様々であるが 3），管理の負担が少ない，中央の認証サーバ上のグループや所属などの属性で行うことが求められる．しかし，中央の認証サーバで管理される属性は，所属や身分など基本的な情報である．個々のサービスが認可で必要とする属性が含まれていない場合も多い．そのため，多くのサービスでは，アクセスを許可するユーザリストを作成し管理していた．近年では，オンラインサービスの増加に伴い，1 つのサービスで認可のために作成されるユーザリストは，別の複数のサービスでも使用されることも増えている．このような背景より，各サービスの管理者が，サービスの認可などで必要とするユーザの集合を，各サービスから切り離して管理できることが求められる．

そこで，我々は，統合認証基盤と連携したサービスの認可などで必要とするユーザの集合を「グループ」とし，統合 ID と属性を用いてグループ管理を行えるようグループの体系化を行ってきた 4）．グループのメンバを定義する際，ユーザを個々に列挙するだけでなく，属性を用いて導かれることが求められる．これは，グループ管理のシステムとして欧米で広く用いられている Grouper システムで実現されている．しかし，Grouper では，グループを管理できる人は，ユーザの属性に対して管理や閲覧権限を持っている人であり，メンバにできるのは，それぞれの権限の範囲に含まれるユーザである．そのため，作成できるグループが限定的であった．

それに対して，我々は，ユーザ属性の管理権限が無い部局のサービス管理者を含む一般ユーザも，グループを作成し，管理ができる仕組みを提案中である．

本稿では，提案する仕組みを実現する際，システム構築をするにあたって，システム上で行われるグループの操作の在り方を中心に述べる．グループ管理に必要な操作は，グループに対する操作とグループの参照に分け，それぞれ実装する際に，どれくらい必要とされる操作であるかを検討する．それらを前提に，グループ管理システムの実装を

†1 京都大学
Kyoto University
†2 東京海洋大学
Tokyo University of Marine Science and Technology

Vol.2014-IOT-27 No.18
Vol.2014-SPT-11 No.18
2014/10/10

行う．また，実装したシステムを試験運用し，その結果を評価する．

本研究で実装するグループ管理の仕組みは，Web サービスの認可などで使用することを前提とするが，作成したグループは，メーリングリストやスケジュール共有などのサービスでも応用できるものである．

2 章では，グループ管理に関する関連技術について述べ，3 章では，提案するグループ管理システムの設計について述べる．4 章では，提案するシステムの実装について述べ，5 章では，まとめを述べる．

2. グループ管理の関連技術

2.1 グループ管理の要件

組織内には様々なグループが存在する．グループは，グループごとに用途や規模が異なることや，グループの作成時期やメンバの変更時期などが異なる 5)．そのため，グループを中央で一括して管理することは，中央の管理者に非常に負荷がかかるため，グループごとに管理者を立て，分散して管理することが求められる．

本研究におけるグループの用途は，Web サービスのアクセス制限とするが，作成するグループは，メーリングリストやファイル，スケジュールの共有など，様々な用途で利用できるものとする．近年はオンラインサービスの増加に伴い，一つのサービスのために作成されたグループは，別のサービスでも利用することも多く，グループの管理はサービスごとに行うより，特定のサービスから切り離して管理することが求められる．

2.2 メンバ定義

グループのメンバを定義する際，メンバとしたい人を列挙する方法（以下，列挙型とする）が広く用いられているが，これはメンバを個々に管理することになり，グループ管理者の負荷が高くなる．グループ管理者の負荷を軽減するためには，ユーザの属性などから条件式によりメンバが導かれる方法（以下，属性型とする）が望まれる．ユーザ属性の変更がある場合，変更に合わせたメンバが導かれることで，グループ管理者の管理の負荷は非常に低くなる．さらに，別途作成済のグループに対する集合演算を用いてメンバを導く方法（以下，複合型とする）もある．その場合は，作成済グループのメンバが変更されると，変更に合わせたメンバが導かれることで，グループ管理者の管理の負荷は非常に低くなる．

これらより，メンバを定義する際には，表 1 の 3 つの種類でメンバを定義できるまたことが求められる．

表 1　メンバ定義の種類

Table 1　Patterns of members defined

列挙型	メンバのリストを列挙する
属性型	属性に関する条件式から導く
複合型	すでに定義されているグループの集合演算 (和集合，積集合，差集合，補集合)により導く

2.3 グループ管理の先行研究

前項の要件を満たしたグループ管理のシステムは，INTERNET2 が Grouper プロジェクトで開発した「Grouper」システムが，欧米で広く展開されている 6）7)．

しかし，Grouper では，グループ管理者となる人は，ユーザの属性に対して管理や閲覧できる（以下，ユーザ属性管理権限とする）人である．メンバに登録できるのは，それぞれのグループ管理者に設定されるユーザ属性の管理権限内のユーザである．そのため，作成できるグループが限定されている．また，グループの管理者は複数名設定可能であるが，グループ管理者が不在となったら，グループは削除され，例外的にグループ管理者不在のグループを継続することは対応していない．

2.4 先行研究に対する著者らの提案

組織には，事務スタッフが存在し，公式的に管理すべき人の範囲が決められており，それぞれに決められた範囲でユーザ属性の管理を行っている．しかし，人の管理を行わない一般ユーザは，ユーザの在籍有無などの情報から個人情報に当たるとのことで，閲覧できない場合が多い．部局のサービス管理者などは，人の管理を行わないため，一般ユーザに含まれる．そのため Grouper でグループ管理を実現する場合，事務スタッフはグループ管理者になることができるが，各サービスの管理者は，グループ管理者となることが難しい．

しかし，グループは部局などで管理されるサービスの認可で使用する場合，ユーザ属性管理権限が与えられている人だけでなく，サービス管理者などの一般ユーザにもグループ管理を行えることが必要となる．一般ユーザにグループの管理権限を与える際,ユーザ属性管理権限がないため,在籍するユーザ情報を開示せずにグループ管理をすることが求められる．また，グループ管理の際には，グループ管理者とシステム管理者の管理の負荷を軽減するため，グループ管理者とメンバには，統合 ID を使用することが求められる.メンバやグループ管理者に統合 ID を用いる場合，いつの間にか削除され，グループ管理者やメンバが存在しなくなる可能性がある．Grouper では，グループ管理者が存在しなれば，グループが削除されるが，実際には継続が必要なグループがある．グループ管理者が存在しなくなった場合，原則，グループ削除でよいが，できる限りグループ管理者が存在しなくならないよう，かつ，必要に応じて

継続できるグループを残せる仕組みが求められる．

これらを元に，著者らは，グループ管理に対する体系化を行ってきた 4)（図 1)．メンバ定義の例は，表 2 のとおりである．ただし，一般ユーザには，ユーザ属性の管理権限だけでなく，ユーザの在籍確認を閲覧する権限も与えられない．そのため，一般ユーザがグループ管理を行う際，列挙型によりメンバを作成する際，グループ管理者が知り得ることができる ID を列挙することにより登録することになるが，属性などにより導いたメンバの閲覧はできない(表 3)．

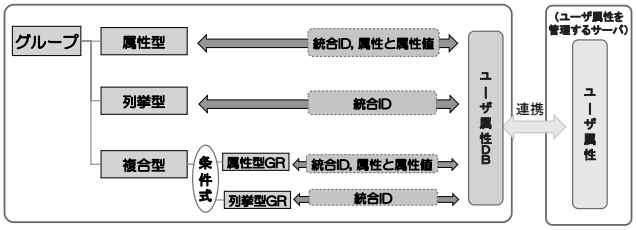

図 1　ユーザ属性の連携

Figure 1　Coordination of user attributes

表 2　メンバ定義の例

Table 2　Examples of members defined

列挙型	groupAA = userA , userB, userC
属性型	groupBB = ("N**"="n**") and ("N**"≧"0")
複合型	groupCC = groupAA and groupBB

(group**=グループ名　N**=属性名　n**=属性値)

表 3　一般ユーザがグループ管理を行う場合のメンバ参照権限

Table 3　Member reference authorities in the case of general users to perform group management

列挙型	有（グループ管理者の知る範囲に限る）
属性型	無
複合型	無（自身が作成した列挙型グループは参照可）

3. グループを管理するためのシステム設計

本章では，これまでに提案してきた統合 ID と属性を用いたグループの体系化に対して，グループ管理のシステムを実現するため，グループ管理に必要な操作を中心に，グループに対する操作とグループの参照に分け，設計を行う．

なお，ここでいう「グループ管理」とは，「メンバ管理を定義するルール」とする．

3.1 グループ管理に必要な操作

グループに対する操作は，大きく以下の 3 つに分けることができる．

グループに対する操作：

A)　**グループの作成**

B)　**グループへのメンバの追加/削除**

C)　**グループの削除**

B)は，列挙型の場合，個々のグループ管理者が，必要に応じてユーザの追加や削除を行うこととなる．属性ベースで定義されたグループは，初回のメンバ登録時だけでなく，変更や削除時においても，グループ管理者が操作しなくても，条件式に合わせて常に最新のメンバ構成になっていることが必要である．

3.2 グループ参照時の操作

グループ管理を実現する際に，属性ベースで定義されたグループは，ユーザの属性の変更に応じて，メンバが変更になる．そのため，常に最新のメンバ情報を得るためには，グループごとにメンバを格納するのではなく，問い合わせがある度にメンバが展開されることが望まれる．

しかし，本研究で管理するグループは，各サービスからの認可だけではなく，メールエイリアスなどでも応用できることとするため，グループのメンバリストが常に必要される．また，与えられたユーザが指定されたグループのメンバか否かの判定やグループのメンバの人数の参照をスムーズに行うことが必要だと考える．まとめると，グループの参照時に必要な操作は以下のようになる．

グループの参照時に必要な操作：

D)　**ユーザのグループメンバーシップの参照**
（与えられたユーザが指定されたグループのメンバか否かを YES/NO で判定するため）

E)　**グループのメンバリストの参照**
（与えられたグループのメンバのリストを得ることができるか）

F)　**グループのメンバ数の参照**（あるいは空グループか否かの参照）

ここでは，これらに対して，それぞれ属性などの変更によりメンバが更新される頻度と，それに対してどれくらいリアルタイム性を求めるか，ということが検討課題となる．

3.3 グループ管理の操作

本研究では，ユーザ属性は，別途ユーザ属性を管理するサーバと同期するが，ユーザ属性の変更は 1 日に何度も行われないことより，全グループのメンバリストの更新は，1 日 1 度行えばよいと考える．ただし，ユーザ属性の変更によりアクセス制限の許可がされなくなったユーザが約 1 日の間，利用できることは，アクセス制限のセキュリティ面の観点から考えると好ましくない．そのため，ユーザ属性の変更により，グループのメンバではなくなる場合に限り，

メンバリストから即時削除することとする．

具体的な動きは，グループ管理システムのユーザ属性 DB には，ユーザ属性を管理するサーバから得る情報だけでなく，所属するグループの情報も格納する．それにより，ユーザ属性の変更時，参加するグループの条件式を確認し，条件に当てはまらなくなった場合は，グループのメンバリストから削除する．これにより，サービスの認可でアクセスを許可する場合に，不要なユーザの利用を避けることができる．

さらに，ユーザ属性 DB に所属するグループ情報を格納することで，与えられたユーザが指定されたグループのメンバか否かを即時に知ることができ，利便性の向上につながる．

4. グループ管理システムの実装

4.1 システム概要

本稿で実装するシステムは，グループの用途を，統合 ID を使うサービスの認可を前提としている．中央の認証サーバは，LDAP で構築されていることが多いことより，本稿では，LDAP の Proxy を用い，その上で実現することとする 8）9）．

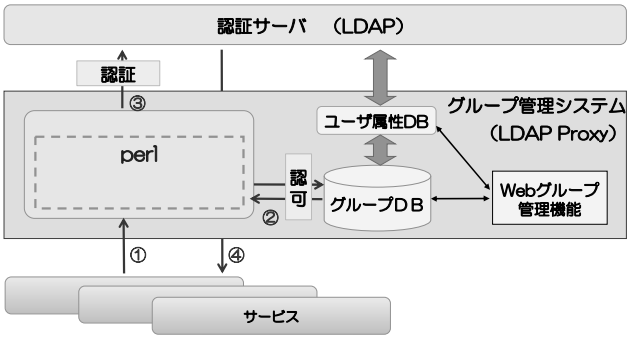

図 2　グループ管理システムの構成

Figure 2　a constitution of group management system

グループの作成，メンバの登録や管理をする際は，グループ管理者が Web 上から操作を行えることとし，ログイン時には，グループ管理者の統合 ID にてログインを行う 10）．

4.2 ユーザ属性 DB に格納する情報

本実装で使用する属性は，統合 ID，所属 1，所属 2，身分，職位などとする（表 4）．ユーザ属性 DB で必要とするユーザの属性情報は，別途，LDAP サーバから受信し，同期することで，ユーザ属性を最新の状態に保つ．なお，属性を管理するサーバは，組織の構成員の身分や所属に応じて，管理部局が異なり，それぞれの担当の管理部局によって，最新情報に更新されるものとする．

ユーザ属性 DB には，ユーザ自身が所属するグループ，自身がグループ管理者であるグループが分かるよう，所属グループの ID，グループ管理者であるグループの ID も格納する．所属グループ ID は，ユーザ属性の変更時，所属グループに問合せを行い，そのグループの条件式に該当しなくなった場合にメンバリストから削除するため，必要とする．また，ユーザごとにグループのメンバであるか否かの判定にも使用する．

表 4　ユーザ属性 DB に格納する情報

Table 4　Data to be in user attributes DB

格納情報	データ例	作成方法
氏名	情報　太郎	属性管理サーバから取得，同期
統合 ID	taro	属性管理サーバから取得，同期
所属 1	事務局	属性管理サーバから取得，同期
所属 2	人事課	属性管理サーバから取得，同期
身分	常勤職員	属性管理サーバから取得，同期
職位	課長	属性管理サーバから取得，同期
所属グループ ID	groupAA	グループのメンバとして追加される都度，グループ ID を追加
グループ管理者であるグループ ID	groupBB	グループ管理者に設定される都度，グループ ID を追加

4.3 グループ DB に格納する情報

本実装では，グループ DB にはグループ ID やメンバ定義の種類の他，メンバのリストや連携するグループ ID などを格納する（表 5）．

連携するグループ ID は，複合型で作成されたグループの条件式に含まれる場合に追加する．グループの変更に応じてスムーズに反映させるためである．

表 5　グループ DB に格納する情報

Table 5　Data to be in groups DB

格納情報	データ例	作成方法
グループ ID	groupAA	グループ管理者が登録
グループ名	本部事務局課長	グループ管理者が登録
メンバ定義の種類	属性型	グループ管理者が登録
グループ管理者 ID	taro	グループ管理者が登録
条件式	("所属 1"="事務局") and ("職位"="課長")	グループ管理者が登録
メンバリスト	taro , jiro , hanako	条件式により追加される都度，メンバに含まれる ID を追加
連携グループ ID	groupBB	複合型で指定される都度，グループ ID を追加

4.4 グループ作成から管理の流れ

グループの作成は，グループを作成したいユーザが Web ブラウザより該当 URL へアクセスし，自身の ID とパスワードを入力する．認証サーバに問合せを行い，成功すれば

ログインする．ログイン後，メンバ定義を行うための種類を列挙型，属性型，複合型より選択する．メンバ定義の種類により動作が異なるため，それぞれ以下にまとめる．

A） **列挙型**

列挙型が選択されると，グループ管理者は，メンバとして登録したいユーザの ID を列挙する．システム側では，列挙された ID の存在を確認し，存在する ID であればメンバリストに登録する．グループの変更時は，グループ管理者が，登録されている ID に対して削除したり，新たに ID を列挙することで追加を行う．それに伴い，システム側では，メンバリストの更新を行う．登録されているメンバの ID がユーザ属性 DB から削除されると，メンバリストからも削除する．

B） **属性型**

属性型が選択されると，グループ管理者は，属性に関する条件式を設定する．システム側では，条件式に基づき，ユーザ属性 DB から必要なメンバを導き，メンバリストに登録する．メンバリストに登録されているユーザの属性が変更になった際，そのグループの条件式を確認し条件式に当てはまらなれば，メンバリストから削除する．また，登録されているメンバの ID がユーザ属性 DB から削除されると，メンバリストからも削除する．グループが，複合型で作成されたグループの条件式に含まれている場合，メンバの削除時には，そのメンバが，該当の複合型のグループに含まれている場合は削除を行う．

ユーザ属性の変更やユーザの新規登録により条件式に当てはまるユーザが増える場合がある．それに対応するため，夜間に一度，全グループに対して，メンバリストの更新を行うことで，必要なメンバの追加を行う．

なお，一般ユーザがグループ管理者となる場合は，導かれたメンバのリストを閲覧することは不可とする．

C） **複合型**

複合型が選択されると，グループ管理者は，既存のグループに対して条件式を設定する．システム側では，条件式から導かれたユーザの ID をメンバリストに登録する．条件式で使用する既存グループに対しては，連携グループとして，作成したグループ ID を登録する．

既存グループのメンバリストから ID が削除されると，そのグループのメンバリストからも ID を削除する．

既存グループのメンバが追加される場合は，即時対応を行わず，夜間に全グループに対して，メンバリストの更新を行う際に，必要に応じてメンバの追加を行う．

なお，一般ユーザがグループ管理者となる場合は，属性型と同様，導かれたメンバのリストを閲覧することは不可とする．

4.5 グループ管理システムの試験稼働と評価

本研究において，実装したシステムの試験稼働を行った．その評価を述べる．

本研究で実装したシステムは，一般ユーザが，統合 ID を用いて，列挙型だけではなく，属性型や複合型のグループを作成する．

本研究では，属性型や複合型のように集合演算で定義されるグループは，都度展開を行うのではなく，それぞれのグループに対して，メンバリストを持たせている．メンバリストを持つことで，グループのメンバリストやメンバ数の参照が容易に行うことができる．また，サービスのアクセス制限だけでなく，メーリングリストなどにも応用可能である．

別サーバで管理されているユーザ属性と連携することにより，属性の変更時，該当属性を条件式に使用しているグループでは，メンバリストを変更するため 1 日 1 回更新を行う．更新には，約 1 日の誤差はあるものの，毎日数十回にわたりユーザ属性の変更が行われるわけではないため，大きな問題ではないと考える．

また，属性が変わることによって，グループのメンバリストから削除が必要な場合は，更新を待たずして即時に行う．そのため，Web サービスに対して，アクセス許可されなくなったユーザを即時にアクセスさせないようにできることにより，安全性の向上につながる実装となっていると考える．

本システムを実装するにあたって，中央の属性管理サーバの属性が曖昧な設計である場合，属性型によるグループ作成が難しい．属性に対する設計は非常に重要なものであるため，実運用していく際には，ユーザ属性の設計を，念入りに行うべき必要があると考える．

5. まとめ

本研究では，統合 ID を使ったサービスのアクセス制限に，中央で管理される認証サーバの属性を使用する際，各サービスに対する詳細な属性が必ずしも含まれていないことより，各サービスの管理者などがアクセス制限のためのグループを作成し管理できるシステムが必要であった．

これまでのグループ管理のシステムでは，グループを管理する人はユーザ属性の管理権限が与えられている人に限定されていたが，本研究では，ユーザの管理権限がなかった一般の構成員でも，グループを作成できることにより，グループ作成の自由度が高まったといえる．また，統合 ID と連携することにより，グループのメンバリストから不要な ID がいつまでも残ることを避けることができる．

実装においては，作成されるグループに対して，ユーザのグループメンバシップの参照やグループのメンバリスト

の参照，メンバ数の参照が行えるよう，グループごとにメンバリストを作成した.ユーザ属性の変更に対応するため，メンバリストは 1 日 1 度更新をすることで，1 日遅れではあるが，ほぼ最新状態に保つことができる．

本研究で実装したグループ管理システムは，グループごとにメンバリストを作成することで，サービスのアクセス制限だけでなく，メーリングリストなどのさまざまな用途に応用できる．本研究で実装したシステムでは，組織内におけるグループに限定していたが，組織を超えた場合でも応用可能である．今後，本研究で提案するグループの用途やメンバ範囲の拡大に向けて，検討していく予定である．

謝辞 本研究におけるシステム開発にご協力頂いた皆様に，謹んで感謝の意を表する．

参考文献

1） 江原康生「大阪大学における新全学 IT 認証基盤システムの構築と運用」電子情報通信学会論文誌 D,Vol.J95-D,No.5,1172-1182,2012
2） 松平 拓也，笠原 禎也，高田 良宏，東 昭孝，二木 恵，森 祥寛「大学における Shibboleth を利用した統合認証基盤の構築」情報処理学会論文誌 52(2), 703-713, 2011
3） 飯田勝吉，新里卓史，伊東利哉，渡辺治「キャンパス共通認証認可システムの構築と運用」電子情報通信学会論文誌 B,Vol.J92-B No.10 pp.1554-1565,2009
4） 清水 さや子，戸田 勝善，岡部 寿男「統合 ID と属性を用いたグループの体系化」マルチメディア，分散，協調とモバイル (DICOMO2014)シンポジウム 3G-4, 2014
5） 伊吹塁「”クラウド型”グループマネジメント～グループ価値最大化に向けた戦略的グループマネジメントのあり方～」, FAS Group Newsletter Vol.30, May, 2011
6） Ineternet2「Grouper」 http://www.internet2.edu/products-services/trust-identity-middleware/grouper/ last visited September. 1, 2014.
7） Grouper Wiki Home https://spaces.internet2.edu/display/Grouper/Grouper+Wiki+Home last visited September. 1, 2014.
8） The Proxy Cache Engine – Open LDAP http://www.openldap.org/doc/admin23/proxycache.html last visited September. 1, 2014.
9） Dr. Dobb's Journal The Open LDAP Perl Backend http://www.drdobbs.com/the-openldap-perl-backend/199102060 last visited September. 1, 2014.
10） Apache https://httpd.apache.org/ last visited September. 1, 2014.