Vol.2020-CLE-30 No.4
2020/3/8

学校間連携を考慮した分散型生徒情報リポジトリに対するアクセス制御システムの試作

堀 和彦[1,a)]　永井 孝幸[2,b)]

概要：近年，教育現場の ICT 化が進められており，生徒の住所や健康診断結果などの個人情報，指導要録や成績などの情報が，デジタル化されオンラインで運用されるようになってきている．また，ePortfolio のように生徒の学習記録を学校の外で利用する試みも掲げられており，これらの生徒情報をオンライン上のデータとして学校間で連携して利用する需要は高まっている．しかし，生徒情報は利用用途に応じてそれぞれ公開されるべき対象が細かく異なるため，用途に応じてカスタマイズ可能なアクセス制御をどのように実現させるかが課題となる．
本研究では，生徒一人ひとりに割り当てられた生徒情報リポジトリに対し，生徒と学校・教師の関係性を表現するソーシャルグラフを用いてアクセス制御を行うシステムを試作した．ソーシャルグラフに格納された生徒と教師の属性情報を，属性ベースのアクセスポリシー記述言語である ALFA を用いて評価する構成とすることで，学校間連携を考慮した粒度の細かいアクセス制御を実現する．

キーワード：アクセス制御，ABAC 方式，ReBAC 方式，XACML，ソーシャルグラフ，ALFA

Prototype of access control system for distributed student information repository considering inter-school collaboration

Kazuhiko Hori[1,a)]　Takayuki Nagai[2,b)]

Abstract: In recent years, the use of ICT in educational sites has been promoted, and personal information such as student addresses and medical examination results, and information such as instructional notes and grades have been digitized and used online. Attempts to use student learning records outside schools, such as e-Portfolio, have also been raised, and demand for using such student information as online data is increasing. However, since the information to be disclosed is different depending on the use case scenario, the challenge is how to implement customizable access control according to different scenarios.
In this study, we developed a system to control access to the student information repository assigned to each student by using ALFA, which is an attribute-based access policy description language , and a social graph that expresses the relationship between students, teachers, and schools. A fine-grained access control considering the cooperation between schools is realized by evaluating the attribute information of students and teachers stored in the social graph.

Keywords: Access Control, ABAC, ReBAC, XACML, Social Graph, ALFA

1. 研究背景と目的

近年，文部科学省の主導のもとで教育の ICT 化が推進されており，IT を活用した指導環境（IT に習熟した教員や

1 京都工芸繊維大学，大学院工芸科学研究科情報工学専攻
2 京都工芸繊維大学，情報工学・人間科学系，准教授
a) k-hori18@dsm.cis.kit.ac.jp
b) nagai@kit.ac.jp

学校教室の無線 LAN 導入）の整備を 2020 年度までに完了させることを目標にしている [1]．総務省が公開している「教育 ICT ガイドブック」[2] には，教育現場の ICT 化の先行事例の代表的なものとして，クラウドを利用した学びの活性化・最適化を挙げている．また，ePortfolio として，生徒の学習の記録・成果物を進学に役立てる試みも掲げられており [3]，生徒の学習情報を小中学校・高等学校に渡り長期的に利用する需要が高まっている．

ここで懸念される事柄として，オンライン上のデータに対するアクセス制御がある．オンライン上のデータは紙媒体と比較して不正アクセスの危険性が高く，実際に 2019 年に日本国内において生徒による成績データ不正操作事件が発生しており，これらの生徒情報に対して厳密なアクセス制御を施す必要がある．同資料では外部からの不正アクセスに対するセキュリティ要件についての記述はあるが，内部の人間への情報公開範囲をどうするか，公開範囲に制限を設ける場合どのような手法で実現するかの記述はない．また，生徒情報のデジタル化によって，学校間で連携して生徒情報を取り扱うことができると期待されるが，そのためには誰がどの生徒情報にアクセスできるのかを綿密に定義しアクセス制御を行わなければならない．よって，全国的な ICT 運用を進めるためには，学校関係者によるアクセスに対するセキュリティ対策について考慮する必要があると考えられる．

本研究は，ICT を導入した学校における生徒情報について，学校間の情報連携を考慮した分散型生徒情報リポジトリに対し，学校内外の教師の所属や担当教科などの属性をもとに適切なアクセス権を設定するためのアクセス制御システムの試作を目的とする．本研究では，成績データや ePortfolio，個人情報などを生徒一人ひとりに割り当てたサーバ（生徒情報リポジトリ）で管理し，その生徒情報リポジトリに対し，生徒と学校・教師の関係性を表現するソーシャルグラフを用いて属性ベースのアクセス制御を行うシステムを試作した．ソーシャルグラフに格納された生徒と教師の属性情報を，属性ベースのアクセスポリシー記述言語である ALFA を用いて評価することで，粒度の細かいアクセス制御を実現する．このシステムによって，複数の学校間の連携を考慮したアクセス制御を実現することができる．

2. 関連事例

2.1 ABAC（Attribute-Based Access Control）方式

ABAC 方式とは，アクセスする主体の属性（名前や所属など），アクセス対象の属性（種類や所有者など），及びアクセスしている環境の属性（時間や場所など）にもとづいて，特定のオブジェクトに対するアクセス可否を判定するアクセス制御モデルである．

ABAC モデルを定義するアクセス制御ポリシー記述言語として，OASIS（Organization for the Advancement of Structured Information Standards）で標準化されている XACML [4] が知られている．XACML では，条件付きの認可である obligation を用途に応じて定義できるのが特徴である．

また，先行事例として ABAC 方式と ReBAC（Relationship Based Access Control）方式を組み合わせたアクセス制御モデルの研究が存在する [5]．

2.2 ALFA（Abbreviated Language For Authorization）

ALFA とは，Axiomatic 社が開発したポリシー記述言語であり，XACML に変換して用いる [6]．XACML と比較して可読性が高く，プログラミング言語のように記述することができる．情報リソースがもつ属性ごとに複数のルールを定義するアクセスポリシーを記述することができる．ユーザの属性とアクセス日時などの環境情報に対し，許可・拒否する条件をルールとして記述する．また，ポリシーで評価する属性は自由に定義することができる．

2.2.1 AuthZForce

AuthZForce とは，XACML v3.0 に準拠した ABAC フレームワークで OW2 Consortium によって提供されている．AuthZForce では，ポリシー管理・アクセス承認要求の際にアプリケーションが呼び出すことができる Restful API を提供している [7]．

3. 学校間連携を考慮した生徒情報アクセスに対する要求と解決策

本節では，学校間連携を想定した生徒情報の管理とアクセス制御に対する要求とその解決策を示す．
本研究で想定する学校組織は次のとおりである．

- 小学校・中学校・高等学校が存在し，小学校は 6 年次制，中学校と高等学校は 3 年次制とする．
- それぞれの学校は学年ごとに 1 つ以上のクラスに分かれている．
- それぞれのクラスには 1 人の担任教師が存在し，教科ごとに担当の指導教師が存在する．
- すべての学校では年度ごとに進級があり，そこでクラス替えが行われる．
- すべての生徒・教師には年度中に転校・異動の可能性がある．

3.1 学校間連携を想定した生徒情報の管理に対する要求

生徒情報は小中・高等学校をまたいで活用できることが求められる．例えば特に，生徒が転校する場合や入試の合否判定に生徒の成績記録を利用する場合に，学校間で連携して生徒情報を取り扱えることが求められる．

図 1 教師とクラスの関係性

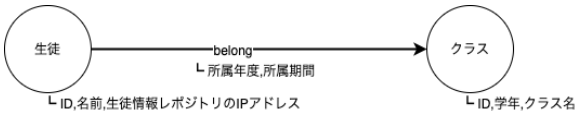

図 2 生徒とクラスの関係性

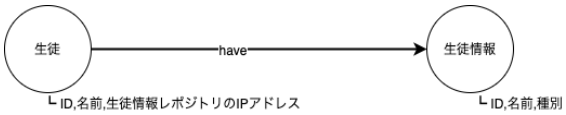

図 3 生徒が生徒情報を所有している様子

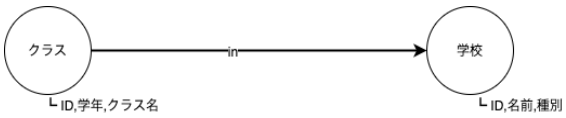

図 4 クラスが学校に属している様子

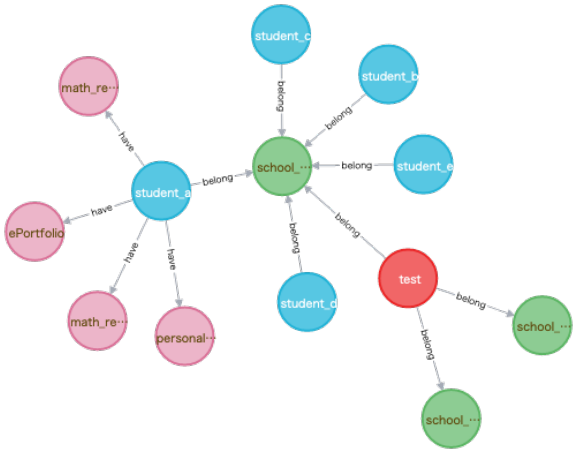

図 5 グラフ DB の構造

3.2 学校間連携を想定したアクセス制御に対する要求

原則として，教師は直接指導関係にない生徒の生徒情報は閲覧できないことが求められる．直接指導関係にある生徒の生徒情報については，アクセスポリシーに基づき，認可された生徒情報を閲覧できることが求められる．また，教師が生徒を指導する上で，自身が過去に教えていた生徒の当時の生徒情報を閲覧する必要が生じた場合，その生徒情報を閲覧できることが求められる．生徒情報の中に一部公開すべきでない情報が含まれる場合，その情報のみを隠せることが求められる．例外として，教師が直接指導関係にない生徒の生徒情報の閲覧が許可される場合（進学のために教師が生徒情報を閲覧する場合など），それが適切に許可されなければならない．

3.3 学校間連携を想定した生徒情報の管理に対する解決策

学校をまたいで生徒情報を活用するために，生徒情報を生徒個人単位で管理する．生徒一人ひとりにファイルサーバを割り当て，これを生徒情報リポジトリとする．生徒情報リポジトリには，その生徒の学校での種々の活動記録（成績や ePortfolio など学習にかかわるデータ類）や個人情報（住所，健康診断結果など）が保管される．教師はアクセスポリシーに基づいて，生徒情報リポジトリにアクセスする．

3.4 学校間連携を想定したアクセス制御に対する解決策

生徒と教師は様々な属性をもち，学校やクラスを介して複雑に結びつきあっている．教師と生徒の時系列に沿った関係性の変化を表現するために，グラフデータベース neo4j を利用する．グラフデータベースは生徒・教師・学校・クラス・生徒情報リポジトリ内のデータをノードとし，それらの結びつきをリレーションとして構成する．このグラフデータベースでつながっている教師と生徒を「直接指導関係にある・あった教師と生徒」とする．さらに，それぞれのノード・リレーションのプロパティに各要素の属性情報（教師の担当教科・生徒情報の種類など）を持たせることで，学校組織におけるアクセス制御に必要な情報の管理をグラフデータベースのみで完結させる．

ここで，グラフデータベースの定義を示す．図 1 は，教師とクラスの関係性を表す．belong はそのクラスを担任していること，teach はそのクラスで授業をしていることを表す．図 2 は，生徒とクラスの関係性を表す．図 3 は，生徒が生徒情報を所有していることを表す．図 4 は，クラスが学校に属していることを表す．

アクセス制御にはポリシー記述言語の XACML を用いる．アクセスポリシーの定義をアクセス制御のプログラムと分けることで，アクセスポリシーを変更する際にプログラムに手を加えないようにする．

3.5 グラフ DB を用いた生徒と教師の関係性と属性情報の管理

グラフ DB として使用した neo4j では，グラフ構造のデータ処理に Cypher クエリと呼ばれる言語を利用する．Cypher クエリは SQL ライクなクエリ言語で，グラフ DB から属性や関係性を柔軟に検索することができる．Cypher クエリを用いてグラフ DB から生徒と教師の関係性，属性情報の取得をどのように行うか図 5 のグラフを例に説明する．

教師’test’ と生徒 a，生徒 a の’personal_information’ のそれぞれがもつ属性と，それらを結ぶ関係性を検索する場合，Cypher クエリはクエリ 3.1 のように記述する．

クエリ 3.1 の検索結果を表 1 に示す．

Cypher クエリ 3.1 クエリ 4

```
MATCH (t:Teacher{id:"test"})-[b1{year:2019}]->(:
    Class)<-[b2:belong]-(:Student{id:"std-a
    "})-[:have]->(f:File{id:"std-a_personal"})
    RETURN t, b1, b2, f;
```

表 1 クエリ 3.1 の検索結果

返り値名	属性
t	{name:test,id:test}
b1	{year:2019,from:2019-04-01,to:2020-03-31}
b2	{year:2019,from:2019-04-01,to:2020-03-31}
f	{name:personal_information,type:Personal,id:std-a_personal}

3.6 ポリシー記述言語 ALFA を用いたアクセスポリシー定義とアクセス制御

ポリシー記述言語 XACML を用いることで，グラフ DB 上で表現された属性情報と関係性をもとに属性ベースアクセス制御を実現する．標準規格とされている XACML を用いることで，学校組織においてシステムを長期的に運用できると考える．アクセスポリシーの定義にはポリシー記述言語 ALFA を用い，XACML にコンパイルしたものを実際のアクセス制御に用いる．

4. アクセス制御システムの設計

試作アクセス制御システムは，生徒一人ひとりに割り当てられた生徒情報リポジトリに対し，グラフ DB で表現された教師・生徒の属性を利用して，学校間連携を考慮した ABAC 方式のアクセス制御を実現する．生徒と教師がもつ名前，ロール，所属クラスなどの属性をもとにアクセス要求を評価し，認可されたユーザのみアクセスを許可する．

教師と生徒の属性情報・関係性をアクセス制御に利用するために，ABAC 方式を採用したオープンな処理系をもつポリシー記述言語である XACML を採用する．アクセス要求を評価する PDP には AuthZForce を使用する．情報リソースとなる生徒情報は，学校組織に所属する生徒一人ひとりに割り当てられた生徒情報リポジトリ内のデータとする．アクセス制御を行う PEP は Python を用いて実装する．ユーザが生徒情報にアクセスするための Web サイトは Flask を用いて実装する．

4.1 教師用機能一覧

- ユーザが生徒情報リポジトリ内の任意の生徒情報を閲覧する機能
- ユーザの学校所属記録に基づき，閲覧する生徒情報を，学校名→クラスと所属期間→生徒名→生徒情報名の階層型のリンクをもつ Web ページとして生成する機能（図 6）
- ユーザと生徒情報の属性情報（ファイルの種類，教科，更新日時など）と関係性に基づき，アクセス制御を行う機能

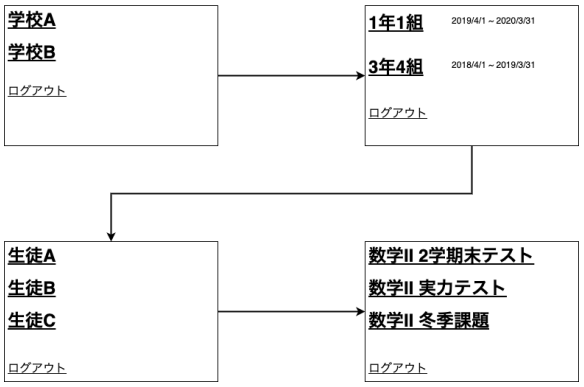

図 6 階層型リンクをもつ Web ページの構成

```
math 2018 record

試験日：2019年12月4日

<mask tag="score">
65 / 100
</mask>

評定：可

<mask tag="comment">
補講の必要あり
</mask>
```

図 7 生徒情報ファイルの例

- 生徒情報にマスク処理を施す機能

4.2 生徒情報リポジトリ

生徒情報リポジトリは，生徒情報を保管するためのファイルサーバである．WebDAV プロトコルを用いて PEP モジュールと通信を行う．生徒情報は，主に担任教師がアクセスする住所録や健康診断結果などの個人情報を Personal タイプ，主に授業担当教師がアクセスするテスト結果や指導要録などを Record タイプ，ePortfolio を ePortfolio タイプと定義する．また，生徒情報はすべてテキストファイルとする．マスク処理を行う箇所は <mask> </mask> のタグで指定されている（図 7）．

4.3 対象とするアクセス制御ポリシー

試作アクセス制御システムで対象とするアクセス制御ポリシーを以下に示す．

(1) クラスの担任教師は現時点でそのクラスに所属する生徒の Personal タイプと ePortfolio タイプの生徒情報を閲覧できる．

(2) 授業を教える教師は，担当するクラスに所属する生徒の Record タイプの生徒情報の中でその年のものを，自身の担当する教科に限り閲覧できる．

(3) また，その年のものでない場合でも，現年度から過去 3 年度分までの生徒情報を閲覧できるが，閲覧できる

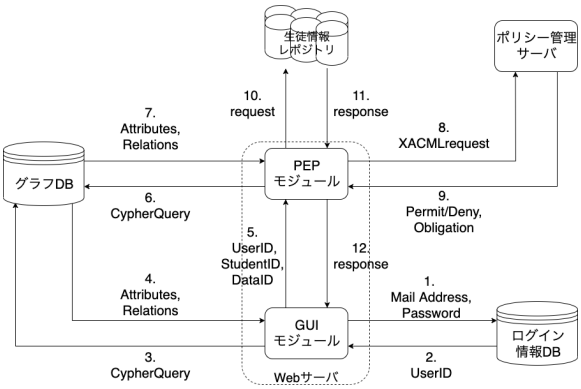

図 8 試作アクセス制御システムの全体構成

データ内容は制限される．

(4) 授業を教える教師は，自身が授業を担当した生徒の Record タイプの生徒情報について，3 年前までのものを閲覧できる．

(5) 入試担当の教師は，自身が所属する学校へ出願している生徒の ePortfolio タイプの生徒情報を閲覧できるが，閲覧できるデータ内容は制限される．

5. 試作アクセス制御システムの実装

試作アクセス制御システムは，教師のもつ属性（担任クラスや指導科目）に応じて生徒情報のアクセス制御を実現し，生徒情報リポジトリから生徒情報を取得するアプリケーションである．試作システムは WebDAV プロトコルで生徒情報リポジトリにアクセスし，教師が Web ブラウジングの感覚で生徒情報へアクセスすることを可能にする．

試作システムは，GUI モジュールと PEP モジュールからなる Web アプリケーションと，ログイン情報 DB，グラフ DB，生徒情報リポジトリ，ポリシー管理サーバで構成されている．試作システムの全体構成を図 8 に示す．なお，ポリシー管理サーバは PDP・PAP の役割を果たす AuthZForce である．ALFA をコンパイルして生成した XACML ポリシーを AuthZForce で利用するために，AuthZForce にポリシー結合アルゴリズム OnPermitApplySecondCombiningAlg を追加している．

GUI モジュールは教師が Web 上で生徒情報を閲覧する際の GUI を提供するモジュールであり，Python 用の Web アプリケーションフレームワークである Flask で実装している．ログイン後，GUI モジュールは Cypher クエリを用いてグラフ DB から教師の関係性情報を取得し，所属履歴のある学校名→クラスと所属期間→生徒名→生徒情報の階層型のリンクをもつ Web ページを生成する（図 6）．生徒情報名のリンクがクリックされたら，PEP モジュールに教師の ID と生徒情報の ID，生徒情報を所有する生徒の ID を引数として渡す．

PEP モジュールは生徒情報リポジトリへのアクセス可否を決定し，生徒情報リポジトリから生徒情報を取得・編集するモジュールである．GUI モジュールから受け取った教師・生徒・生徒情報の ID をもとにグラフ DB からそれらの属性情報を取得する．取得した属性情報から XACML 承認リクエストを生成し，ポリシー管理サーバに送信する．ポリシー管理サーバからのレスポンス（以下，XACML 承認レスポンス）に従い，生徒情報リポジトリへのアクセス可否を決定する．XACML 承認レスポンスが Permit の場合，GUI モジュールから受け取った生徒情報の ID と一致する生徒情報を生徒情報リポジトリから取得し，キャッシュ用ディレクトリに取得した生徒情報を保存し，GUI モジュールに認可を返す．XACML 承認レスポンスに obligation が含まれていた場合，obligation の ID に応じてマスクなどの処理を行う．XACML 承認レスポンスが Deny の場合，GUI モジュールに否認を返す．ObligationID が mask の場合，生徒情報内で <mask> タグに挟まれた箇所を <masked> 文字列に置換する．

グラフ DB は教師・生徒・生徒情報の属性と関係性を表すグラフデータベースであり，neo4j で実装している．生徒・教師・学校・クラス・生徒情報リポジトリ内のデータをノード（表 2）とし，それらの結びつきをリレーション（表 3）として表現している．

表 2 グラフ DB のノード

ノード名	属性
Teacher	教師の ID，教師の名前
Student	生徒の ID，生徒の名前，生徒情報リポジトリの IP アドレス
School	学校の ID，学校の名前，学校の種類
Class	クラスの ID，クラスの名前，学年，クラス番号
File	データの ID，データの名前，データのタイプ

表 3 グラフ DB のリレーション

リレーション名	始端	終端	属性
belong	Teacher,Student	Class	所属期間，所属年度
teach	Teacher	Class	所属期間，所属年度，指導科目
manage	Teacher	School	所属期間
choice	Student	School	所属期間
in	Class	School	
have	Student	File	

5.1 試作アクセス制御システムのアクセス制御ポリシー

試作システムのアクセス制御ポリシーの (1) と (5) をポリシー記述言語 ALFA で記述したものが ALFA ポリシー 5.1，5.2 である．アクセス制御ポリシーが評価する属性として ALFA に新たに定義した属性を表 4 に示す．

6. 試作アクセス制御システムの動作検証

本節では，試作アクセス制御システムの動作検証で想定したシナリオ，検証の流れを示す．紙面の都合上，ある生徒の進学志望先の学校の教師がその生徒の ePortfolio を閲

ALFA ポリシー 5.1 allowAccessByBelongTeacher

```
rule allowAccessByBelongTeacher {
    permit
    target clause Attributes.subjectType == "belong"
    condition Attributes.subjectInteractFrom <= Attributes.currentDate
    && Attributes.subjectInteractTo >= Attributes.currentDate
    && Attributes.resourceOwnerInteractFrom <= Attributes.currentDate
    && Attributes.resourceOwnerInteractTo >= Attributes.currentDate
}
```

ALFA ポリシー 5.2 allowAccessToPortfolio

```
rule allowAccessToPortfolio {
    permit
    target clause Attributes.subjectType == "manage"
    condition Attributes.subjectInteractFrom <= Attributes.currentDate
    && Attributes.subjectInteractTo >= Attributes.currentDate
    && Attributes.resourceOwnerInteractFrom <= Attributes.currentDate
    && Attributes.resourceOwnerInteractTo >= Attributes.currentDate
    on permit{
        obligation Attributes.mask
    }
}
```

表 4 アクセス制御ポリシーが評価する属性定義

属性名	内容
subjectType	教師とクラス・学校の関係
subjectTeachingArea	教師の指導科目
subjectInteractFrom	教師とクラス・学校の関係の開始日
subjectInteractTo	教師とクラス・学校の関係の終了日
resourceType	生徒情報のタイプ
resourceOwnerInteractFrom	生徒とクラス・学校の関係の開始日
resourceOwnerInteractTo	生徒とクラス・学校の関係の開始日
resourceArea	生徒情報の科目
resourceDate	生徒情報の更新日
currentDate	現在の日付
currentYearFirst	現在の年度の 4 月 1 日
currentYearLast	現在の年度の 3 月 31 日

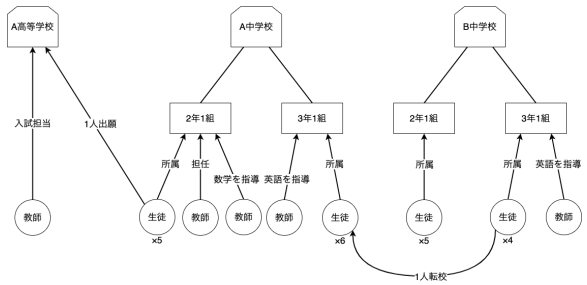

図 9 想定シナリオで扱う学校組織

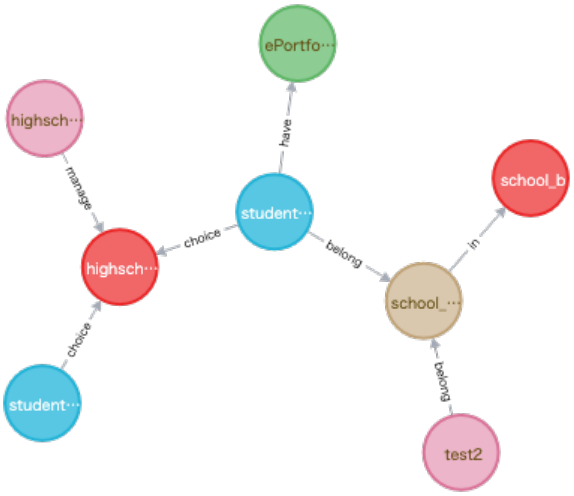

図 10 生徒 p の関係性グラフ

覧するシナリオについてのみ説明する．

動作検証を行うにあたり，5 人の教師と 20 人の生徒，2 つの中学校とそれぞれ 2 つずつのクラス，1 つの高等学校をグラフ DB に登録した．2 つの中学校にはそれぞれ 2 年次と 3 年次のクラスがあり，1 つのクラスに 5 人の生徒が所属している（図 9）．

WebDAV サーバの Docker イメージをもとに 20 台の WebDAV サーバコンテナを作成し，これを 20 人の生徒それぞれの生徒情報リポジトリとする．

6.1 検証内容

生徒 p の関係性を表すグラフを図に示す．

生徒 p は’school_b’ 中学校の 3 年 1 組に所属していて，’highschool_a’ 高等学校へ出願している．本システムのアクセスポリシーに基づくと，生徒 p の ePortfolio について’school_b’ 中学校の 3 年 1 組の担任教師’test2’ は閲覧でき，’highschool_a’ 高等学校の入試担当教師’high-school_teacher’ は制限付きで閲覧できる．教師’high-school_teacher’ のもつ関係性を表すグラフを図 10 に示す．

教師'highschool_teacher' は'highschool_a' 高等学校の教師で，今年度の入試担当教師である．生徒 a は'highschool_a' 高等学校へ出願している．
本システムのアクセスポリシーに基づくと，教師'highschool_teacher' は生徒 a の ePortfolio を制限付きで閲覧できる．
ここで，本シナリオについて次の 2 つのケースで検証を行う．

ケース 1
教師'test2' が生徒 p の ePortfolio へアクセスする．

ケース 2
教師'highschool_teacher' が生徒 p の ePortfolio へアクセスする．

各ケースにおける教師・生徒・生徒情報の属性は表 5 のようになる．

表 5 検証用シナリオの教師・生徒・生徒情報の属性

Attribute	Case1	Case2
subject_type	belong	manage
subject_teaching_area	none	none
subject_interact_from	2019-04-01	2019-04-01
subject_interact_to	2020-03-31	2020-03-31
object_type	ePortfolio	ePortfolio
object_owner_interact_from	2019-04-01	2019-11-01
object_owner_interact_to	2020-03-31	2020-02-28
object_area	none	none
object_touched_in	2019-12-14	2019-12-14

アクセスポリシーにより，2 つのケースはそれぞれ以下のように処理されることが期待される．

ケース 1
ALFA ポリシー 5.2 をすべて満たすため，制限付きのアクセスが許可される．

ケース 2
ALFA ポリシー 5.1 をすべて満たすため，アクセスが許可される．

実際に検証を行った結果を表 6 に示す．

表 6 検証用シナリオの検証結果

Case	Access	elapsed time [ms]
Case1	succeed	48.76
Case2	succeed(masked)	37.17

以上より，検証用シナリオにおいて，期待された通りのアクセス制御が行われたことがわかる．

7. 評価

本節では，試作アクセス制御システムの評価について述べる．

7.1 アクセス制御手法に対する評価

ここで，試作アクセス制御システムで実装したアクセス制御手法（以下，提案方式）について，RBAC 方式と比較した評価を行う．

表 7 提案方式と RBAC 方式の比較

項目	提案方式	RBAC 方式
細粒度のアクセス制御	○	△
多様なロール・属性定義	○	×
ロール・属性の更新	△	○
過去のロール・属性の活用	○	△
アクセスポリシーの更新	△	○

表 7 に示すように，RBAC 方式はロールやアクセスポリシーの更新が柔軟であるが，多様なロールを定義することは難しい．また，過去の指導関係に基づいたアクセス制御を実現する場合，ロール更新時に更新前のロールを記録・管理しておく必要が生じる．一方，提案方式は属性とアクセスポリシーの更新は RBAC 方式と比較すると複雑であるが，細粒度のアクセス制御が可能で，過去の属性を含めて多様な属性を定義できる．よって，多様な属性をもつ学校組織におけるアクセス制御において，提案方式は有効であるといえる．

7.2 アクセス制御システムの所要時間に対する評価

6 節で取り扱ったシナリオについて，属性情報取得・XACML 承認プロトコル・生徒情報取得・マスク処理のそれぞれの所要時間を表 8 に示す．

表 8 シナリオの各処理の所要時間

case	get attribute [ms]	xacml request [ms]	get data [ms]	obligation [ms]
case1	26.33	10.50	13.33	
case2	12.83	10.80	7.901	0.01740

表 8 から，グラフ DB からの属性情報取得にかかる時間がアクセス制御の所要時間の 4 割以上を占めていることが伺える．グラフ探索において，その所要時間はノードに繋がる枝の数・始端ノードと終端ノードとの距離に比例して増加するため，グラフ DB が肥大化することでアクセス制御の所要時間が膨大になる可能性がある．グラフ DB から属性情報を取得する際の始端ノードと終端ノードはそれぞれ教師ノードと生徒情報ノードである．例えば，クエリ 3.1 によるグラフ探索の結果は図 11 のようになる．なお，クエリ 3.1 はシナリオ 1 のケース 1 で PEP モジュールがグラフ DB に送信するクエリである．

図 11 のそれぞれのノードについて，リレーションをもつノードの種類と 1 年間に増加するリレーション数の予測を表 9 に示す．

ここで，教師ノードの 1 年間のリレーション増加数を

図 11 クエリ 3.1 によるグラフ探索の結果

表 9 リレーションをもつノードと 1 年間のリレーション増加数予測

ノード	リレーションをもつノード	1 年間のリレーション増加数予測
教師	クラス，学校	授業担当クラス数（未知数） +担任するクラス数（0 または 1）
クラス	教師，生徒，学校	所属する生徒数（約 40 以下） +授業担当教師数（科目数+α） +担任教師数（1） +所属する学校（1）
生徒	クラス，学校，生徒情報	所属するクラス数（1+α） +出願校数（受験生の場合） +生徒情報（未知数）
生徒情報	生徒	0

10，クラスノードの 1 年間のリレーション増加数を 60，生徒ノードの 1 年間のリレーション増加数を 100 と仮定して，シナリオ 1 のケース 1 について属性情報取得の所要時間の 5 年分の推移を計測した（表 10）．

表 10 属性情報取得の所要時間の推移

経過年数 [年]	get_attribute [ms]
0	23.83
1	32.57
2	23.90
3	40.47
4	39.31

表 10 から，リレーションの増加により緩やかに所要時間が増加しているように見えるが，グラフ DB との通信で発生するラグによる影響を考慮すると，リレーションの増加がグラフ探索の所要時間に及ぼす影響は軽微であると考えられる．

8. 考察と今後の課題

8.1 試作アクセス制御システムの検証結果について

7 節で述べたように，試作アクセス制御システムは本研究の要求に対する解決策となり得ると考える．表 6 より，全てのケースにおいて生徒情報取得にかかる時間は 100 ミリ秒以下であり，アクセス制御の所要時間に関しても，本シナリオにおいては十分な速度であると考える．また，年度経過によりグラフ DB 内の属性情報が増加しても，十分なアクセス制御速度を保つことができると考える．

8.2 マスク処理の応用について

PEP モジュールはポリシー管理サーバが発行する obligationID に応じて処理を決定するため，マスク処理に関する obligationID を複数定義し，その obligationID に対応するタグでマスク箇所を指定することで，教師の属性ごとに異なる箇所にマスク処理を施すことが可能となる．

また，本システムの実装では，<mask> タグで指定された文字列を <masked> タグに置換することで生徒情報のマスク処理を実現しているため，マスク処理が可能な生徒情報はテキストファイルに限定されてしまう．例えば pdf 文書であれば，文書内にタグ付けを行うことで同様にマスク処理を施すことが可能だが，画像ファイルの特定の箇所にマスク処理を施す場合はタグ付けによるマスク処理ができない．今後の課題として，画像ファイルの特定の箇所へのマスク処理に対応することが挙げられる．

9. 結言

本研究では，学校間連携を考慮した分散型生徒情報リポジトリに対するアクセス制御システムを試作した．今後の課題として，生徒情報をリポジトリに送信するモジュールとグラフ DB を自動で更新するモジュールの追加が挙げられる．

謝辞 本研究は JSPS 科研費 18K11568 の助成を受けたものです．

参考文献

[1] 文部科学省：日本再興戦略 2016（抜粋），文部科学省（オンライン），入手先 ⟨https://www.soumu.go.jp/main_content/000425445.pdf⟩（参照 2020-02-06）．
[2] 総務省：教育 ICT ガイドブック (2017).
[3] 文部科学省：「JAPAN e-Portfolio」について，文部科学省（オンライン），入手先 ⟨https://www.mext.go.jp/b_menu/shingi/chousa/shotou/143/shiryo/__icsFiles/afieldfile/2019/02/20/1413594_001.pdf⟩（参照 2020-02-06）．
[4] Organization for the Advancement of Structured Information Standards: eXtensible Access Control Markup Language (XACML) Version 3.0, Organization for the Advancement of Structured Information Standards (online), available from ⟨http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html⟩ (accessed 2020-02-06).
[5] Aktoudianakis, E.: Relationship based access control, PhD Thesis, University of Surrey, Guildford (2016).
[6] Axiomatics: Abbreviated Language for Authorization (ALFA) Plugin for Eclipse, Axiomatics (online), available from ⟨https://ma.axiomatics.com/acton/fs/blocks/showLandingPage/a/10529/p/p-0060/t/page/fm/0⟩ (accessed 2020-02-06).
[7] FIWARE Open Specification License: FIWARE Authorization PDP API Specification, FIWARE Open Specification License (online), available from ⟨https://authorizationpdp.docs.apiary.io/#⟩ (accessed 2020-02-06).