Vol.2022-IS-159 No.3
2022/3/7

小学校プログラミング教育のための 個人データ利用許諾管理機構のプロトタイプ

寺西 司[1]　掛下 哲郎[1]

概要：様々な IT サービスの利用者から収集した個人データを分析することで，ビジネスや業務に役立つ知見を得る取り組みはしばしば見られる．その際には OECD 8 原則等の個人情報保護を徹底する必要がある．本論文では，小学校でのプログラミング教育を題材として，生徒から収集した学習データを分析して教育指導に役立てるラーニングアナリティスクにおいて，生徒や保護者から許諾を得た上で個人情報を利活用するための個人データ利用許諾管理機構のプロトタイプを開発する．本プロトタイプでは，著者らが提案した個人情報管理システムの技術を活用しており，生徒や保護者は当人の個人情報に対するアクセスの許諾を管理するとともに，許諾に基づく当人の個人情報に対するアクセス履歴を参照できる．Amazon 社が提供する AWS を活用してプロトタイプを開発することで，システムのアーキテクチャを単純化するとともに，実装の手間を減らすこともできる．

キーワード：自己情報コントロール権，OECD8 原則，個人情報保護，Amazon Web Services

A Prototype System of Personal Data Authorization for Programming Education at Elementary School

TSUKASA TERANISHI[†1]　TETSURO KAKESHITA[†1]

There are many efforts to analyze personal data collected from users of various IT services to obtain useful knowledge for business and operations. In such cases, it is necessary to ensure the protection of personal information in accordance with the OECD 8 Principles. In this paper, we develop a prototype of a personal data authorization system for a learning analytics system that analyzes learning data collected from students learning computer programming at elementary school. Since the system utilizes the personal data management system that we proposed, students and their parents can manage the permission to access their personal information and can refer to the access history of their personal information based on the permission. We can simplify the system architecture and can reduce the implementation effort by developing the prototype using AWS provided by Amazon.com, Inc. .

Keywords: Right to Control Self Information, OECD Basic Principles, Personal Information Protection, Amazon Web Services

1. はじめに

Web サービスには個人のデータを活用することで成り立っているものが多くある．その一方，利用者の知らないところで勝手にデータを利用していることは問題との意見が各国で挙がるようになった．近年，個人のプライバシー保護に向けた動きも活発化しており，欧州連合（EU）では一般データ保護規則 GDPR（General Data Protection Regulation）[1]，米国カリフォルニア州では消費者プライバシー法 CCPA （California Consumer Privacy Act）がそれぞれ施行されている．しかしながら，現代社会において個人データの重要性はますます高まっている．

我々はこれらの課題を解決すべく，個人情報保護の基本となる OECD 8 原則[2]をもとにして法律や契約に基づく個人データの利用に対して個人による同意を半自動化するシステムを構想している[3]．対象範囲は，小学校プログラミング教育を想定し，アクセス制御やデータ提供の手順についてプロトタイプの設計を行う． 本論文では，クラウドコンピューティングサービスの Amazon Web Services[4] を利用してプロトタイプを実装する．

2. 関連研究

文部科学省は GIGA スクール構想を通じて小中学校のデジタル教育を推進している．それに伴い，小学校でのプログラミング教育が 2020 年 4 月より義務化された[5]. 現在，小中学校へのデジタル教材の配置が進む一方で，プログラミング教育等における情報管理には課題がある．

とりわけ，プログラミング教育を担当する教師の多くがプログラミング未経験であり，教育現場で生じる学習データを収集し，収集したデータを分析する必要性が生じる．それにより，プログラミング教育を改善するとともに，授業の効率化を図る．

対象範囲は広いが同様の試みとして情報銀行が挙げられる[6]．これは，個人が自己情報を管理しながら効率的にデータ流通を行うための仕組みであり，原則として情報主体者から許諾を得た上で、個人情報を有益な情報源として活用する試みである．

これについて，日本 IT 団体連盟の情報銀行推進委員会が

1 佐賀大学
Saga University

総務省・経済産業省による情報信託機能の認定に係る指針に基づき，情報銀行関連サービスに対して情報セキュリティ対策やプライバシー保護対策に関する認定基準に適合しているか判断している[7].

しかし，現状の情報銀行の運営には課題がある．個人情報収集に伴う同意確認や，ルールを遵守した個人情報管理を自動化しなければ，学習履歴データを扱う教育現場に効果的に導入することは難しい．

3. 個人情報管理システム

著者らはこれまでに個人情報管理システム（図 1）の研究・開発を進めてきた[8,9]．個人情報管理システムは個人データに対してアクセスコントロールを自動化するものであり，個人情報の保護規則である OECD 8 原則（目的明確化， データ内容，個人参加，収集制限，利用制限，安全保護，責任，公開）への対応を可能としている．

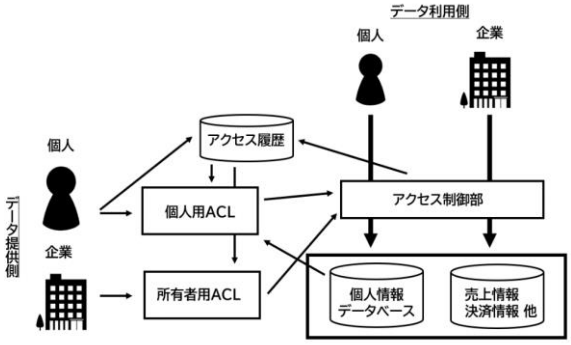

図 1　個人情報管理システム

本システムでは個人と事業者が，それぞれ複数存在する場合を想定しており，個人データは個人と事業者の取引により生じる．たとえば，個人 A が事業者 B を利用した際には，事業 B が個人 A の利用履歴を所有することになる．図 1のように，個人 A と事業者 B のアクセス制御リスト（ACL，Access Control List）を分離することにより，個人と事業者を対等な関係にする．また，個人がアクセス履歴を参照できるようにすることで OECD8 原則の個人参加の原則に適合している．

4. 個人データ利用許諾管理機構

本稿では，個人情報管理システムを用いることで個人データの利用に対して個人による同意を自動化するシステムを提案する．これにより，法令などに詳しくない個人が同意の可否を自ら判断する機会を減らし個人データの利用の効率化を図る．

小学校プログラミング教育を実施するにあたって，学習履歴等を生徒の学習状況の把握や授業の改善に役立てることは重要である．一方で，学習履歴は個人情報を含んでおり，その保存や利用等は適切に運営される必要がある．

そこで，学習履歴等を含む教育データに対する児童生徒および保護者の同意確認や教育データへのアクセス管理を行うためのオンラインでの個人データ利用許諾管理機構を構想する．これを通じて，教育データの蓄積や個人データの受け渡し方法をめぐる個人情報管理のモデル化を行う．

適切な個人データ運営を行うことで、次のような活用場面が生まれると考える．まず，生徒が自身の得意や苦手分野を把握することとそれに応じた効率的な指導を教師が行えることが挙げられる．また，大学等の研究機関が学習状況を調査・分析し，教材の効果を評価できる．他にも，行政機関が学校や学年ごとのデータを参照でき，業務の効率化や今後の施策に反映が可能となる．

本システムでは，個人データ（個人情報保護法に規定する個人情報）は当人または当人がサービスを利用した組織（提供側組織）が個人データ利用許諾管理機構によって管理する．個人データの利用を希望する利用者は利用許諾管理システムを通じて利用申請を出す．アクセス権限の情報はデータベース上で管理し，個人データ利用許諾管理機構が個別の個人データに対する利用者のアクセスを制御する．

生徒や保護者は本システムから個人データの利用履歴を取得できる．また，本システムが個人データの利用を自動判定できない場合は，個人に利用許諾の可否を問い合わせる仕組みも提供する．

利用許諾の可否を判定する際には，OECD 8 原則の中でもデータ利用先や責任の所在および利用目的に合致した内容の収集であるかが重要である．ただし，個人はいつでも許諾履歴やデータの提供履歴を閲覧可能とし，問題がある場合は異議を申し立てることができる．これにより，過去のデータに対してもデータ内容の保証および個人参加の原則を担保できる．

個人に対する利用許諾の可否の問い合わせは，通知が必要だと利用者があらかじめ設定したデータへの利用申請が出されたときに行われる．この場合は個人に通知を出した上でその情報主体者から許諾が得られない限り個人データが提供されることはない．また，このとき得られた利用許諾の可否は過去の経験として蓄積する．過去の事例を活用することで，将来の判定の自動化に役立てる．

個人データ利用許諾管理機構は，利用者に対してユーザーインターフェースを提供する．この Web サイトでは利用申請の通知の他，許諾履歴やデータの提供履歴を閲覧できる．こうすることで，情報主体者が自己情報コントロール権を行使できる．さらに，個人に通知を出す際にもデータ利用拒否および許諾の推奨オプションを提供できる．

以上のように，情報主体者である生徒が個人データの取り扱いに関与し，同意確認をオンライン上で行うシステムを提案する．それは，教育データの利活用促進や生徒や保護者が当人の個人データを管理できるところに意義がある．本システムは，個人情報に対する適切なアクセス制御を実現することが目的であり，それが達成できていることをデ

モンストレーションにより示す．デモンストレーションの流れは図 2 のように計画している．

まず，データベース上に代表的なデータを作成する．1 つは，学習履歴データであり，情報主体およびいつどこでどのように生成されたかの諸条件を含むものとする．2 つは，法令情報データであり，個人情報保護法や OECD 8 原則といった主要な情報を導入する．次に，ACL 設定の例を作成する．ACL は情報主体者がデータへのアクセスを制御するために用いる．そして，学習履歴データと法令情報データの組み合わせにより ACL 設定がどのような動作を行うのかを示す．

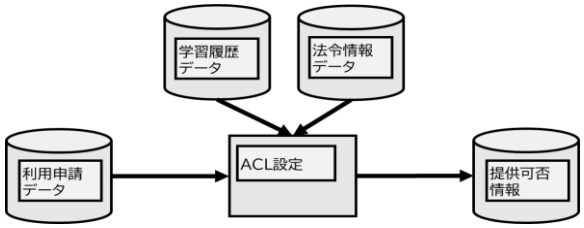

図 2　プロトタイプの概要

5. 要件定義

プロトタイプを設計するにあたって，必要とされるデータおよび実現すべき機能を挙げる．なお，今回はシステムの動作を確認することが目的であるため，ユーザーの管理やそれに伴うログインなどの機能は考慮しないものとする．

5.1 アクターの定義

想定するアクターは以下の通りである．

(a)生徒
(b)保護者
(c)教員
(d)データアクセス希望者（教員，研究機関など）

ここでいう，生徒および保護者は個人データの情報主体者である．また，教員は生徒へのデータアクセスに対して他のアクセス希望者に比べてその関係性から条件が緩和される特徴を持つ．

5.2 データの定義

サービスを提供するにあたって，データベースを構築する．データベースには，次のようなデータを保存するものとする．

- ユーザー情報：固有 ID，名前，所属，パスワード
- 学習データ情報：情報主体者，データ内容，作成期間
- 法令情報：OECD 8 原則，個人情報保護法などの一部
- 利用申請情報：申請者，利用希望データ，利用目的，申請理由，第三者提供の有無
- アクセス可否情報：個別のデータごとのアクセス権
- アクセス履歴情報：個別のデータに対する利用申請者および利用した日付

5.3 機能定義

今回，プロトタイプにより実現すべき機能を考える．主に，個人データに対する適切なアクセス制御を実現することを満たすものである．

- 利用申請を受け付ける
- アクセス制御を設定する
- データへのアクセスを制御（許可または拒否）する

また、具体的なデモンストレーションの内容として以下に示す 4 通りを考える．

- データ利用申請の受付および法令等に基づいて自動的にアクセス許可されるもの
- 自動アクセス許可のうち過去のアクセス許可に基づくもの
- 生徒および保護者の同意に基づいて許可するもの
- 自動アクセス許可されていたが，生徒または保護者が許可を取り消すもの

6. データベース設計

5. で定義した機能を実現するために，データベースを用意する．基本的に，デモンストレーションを実行するための最低限の情報を保持する．ER 図を用いてデータベースの全体構成を図 3 に示す．また，それぞれのテーブルについて属性名やキー，データ型，簡単な説明を表 1~6 に列挙する．

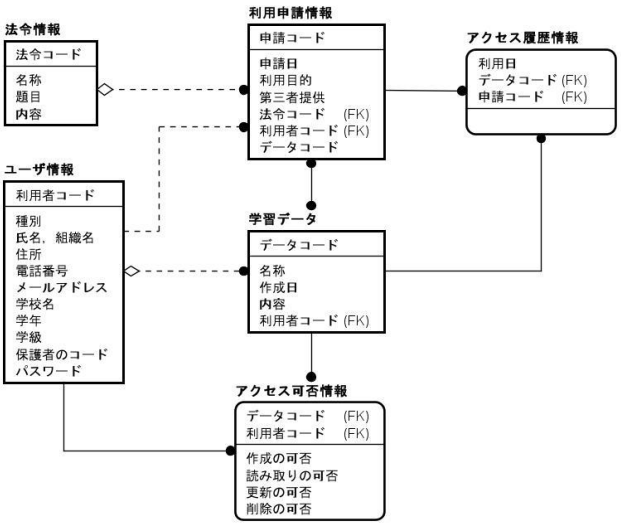

図 3　データベースの全体構成

表 1　ユーザー情報

属性名	キー	データ型	説明
利用者コード	○	整数型	管理者が利用者に付した通し番号
種別		文字列型	生徒，保護者，会社，公的機関，管理者のいずれか
氏名，組織名		文字列型	利用者の名前
住所		文字列型	利用者の住所
電話番号		整数型	利用者の電話番号
メールアドレス		文字列型	利用者のメールアドレス
学校名		文字列型	利用者の在籍学校名

属性名	キー	データ型	説明
学年		整数型	利用者の在籍学年
学級		整数型	利用者の在籍学級
子どものコード		整数型	子どもの利用者コード（保護者限定）
パスワード		文字列型	利用者が設定したパスワード

表 2　学習データ情報

属性名	キー	データ型	説明
データコード	○	整数型	管理者がデータに付した通し番号
利用者コード		整数型	情報主体者に該当する利用者コード
名称		文字列型	データの名前
作成日		整数型	データが作成された日付
内容		文字列型	データの内容

表 3　法令情報

属性名	キー	データ型	説明
法令コード	○	整数型	管理者が法令に付した通し番号
名称		文字列型	法令の名前
題目		文字列型	法令が含まれる条項の題名
内容		文字列型	法令の内容

表 4　利用申請情報

属性名	キー	データ型	説明
申請コード	○	整数型	管理者が申請ごとに付した通し番号
利用者コード		整数型	申請者の利用者コード
データコード		整数型	利用申請対象のデータコード
申請日		整数型	申請書が作成された日付
利用目的		文字列型	データを何のために利用するのかの説明
法令コード		整数型	申請が正当である根拠
第三者提供		論理型	第三者提供の有無

表 5　アクセス可否情報

属性名	キー	データ型	説明
データコード	○	整数型	アクセス権限を設定するデータのコード
利用者コード	○	整数型	アクセス者の利用者コード
作成の可否		論理型	データ作成権限の有無
読み取りの可否		論理型	データの読み取り権限の有無
更新の可否		論理型	データの更新権限の有無
削除の可否		論理型	データの削除権限の有無

表 6　アクセス履歴情報

属性名	キー	データ型	説明
データコード	○	整数型	アクセスが行われたデータのコード
申請コード	○	整数型	アクセスの根拠となる申請書のコード
利用日	○	整数型	利用者がデータへのアクセスを行った日付

なお，表 5 に示すアクセス可否情報はデータ量が多くなる場合も考えられるが，許可されている項目がある場合のみ登録するなどして縮小する．

7. 代表的なユースケース

プロトタイプを用いたデモンストレーションを実行するにあたって，本節では代表的なユースケースを示す．本プロトタイプには Web 上からアクセスする．そのために，ユーザーインターフェースとして HTML 画面を用意する．想定される機能ごとに簡易的な画面を挙げる．また，それぞれの遷移画面で実現される機能も対応づける．

7.1 データ利用申請の受付および法令等による自動アクセス許可

データ利用申請を提出する手順について説明する．ここでは，データ利用申請を行うアクターはデータアクセス希望者を想定する．実現する機能はデータ利用申請の作成である．画面遷移は次の通りである．

- 申請者はデータアクセス者用メイン画面（図 4）からアクセス申請提出を選択する．システムは選択された機能を提供するための画面へ遷移を行う．

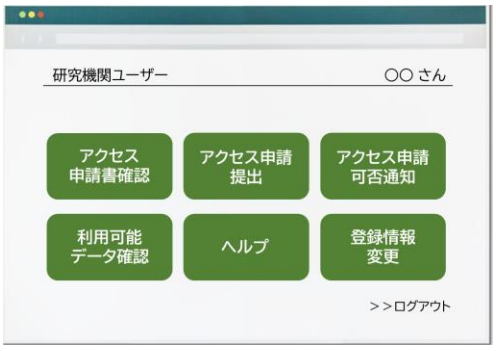

図 4　データアクセス申請者メイン画面

- 申請者はアクセス申請データ選択画面（図 5）でアクセスを希望するデータを選択する．システムは選択されたデータに対応する申請書作成情報を提示する．

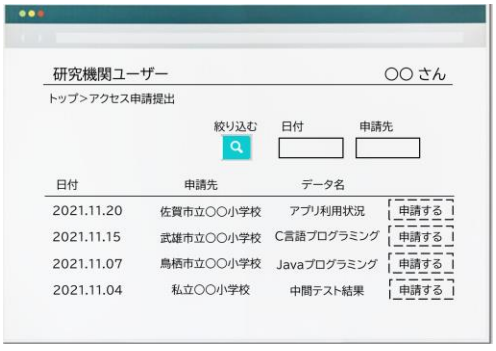

図 5　アクセス申請データ選択画面

- 申請者はアクセス申請入力画面（図 6）で利用目的や利用期間，適合法令を入力する．システムは入力された値を確認し受け付ける．

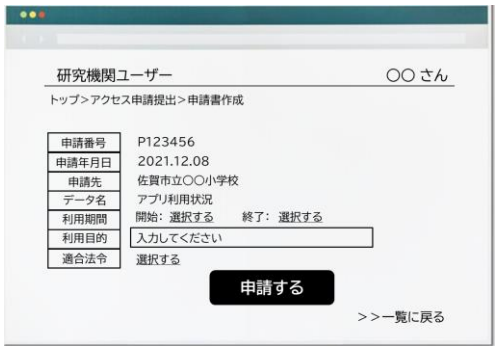

図 6　アクセス申請入力画面

- 申請者はアクセス申請可否結果通知画面（図 7）でアクセス申請の可否を確認する．

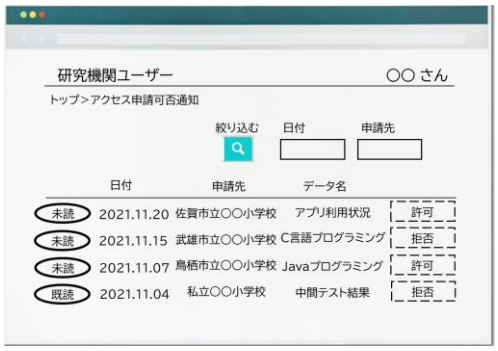

図 7　アクセス申請可否結果通知画面

7.2 過去のアクセス許可に基づく自動アクセス許可

過去のアクセス履歴に基づいて自動アクセス許可を行う手順について説明する．ここでは，データ利用申請を行うアクターはデータアクセス希望者を想定する．実現する機能は個人データへのアクセス許可である．画面遷移は次の通りである．

- 申請者は，データアクセス者用メイン画面（図 4）から利用可能データ確認を選択する．システムは，選択された機能を提供するための画面へ遷移を行う．
- 申請者は，利用可能データ一覧画面（図 8）からアクセスを希望するデータを選択する．システムは，選択されたデータに対応する申請書情報を提示する．

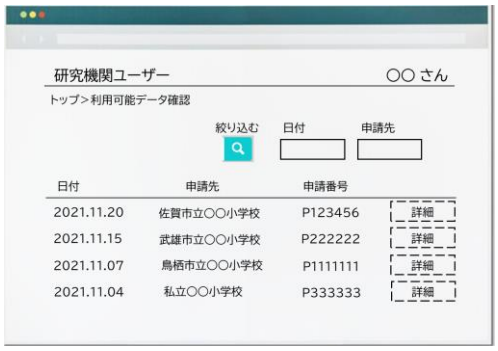

図 8　利用可能データ一覧画面

- 申請者は，利用可能データ詳細画面（図 9）でデータへのアクセスを行う．

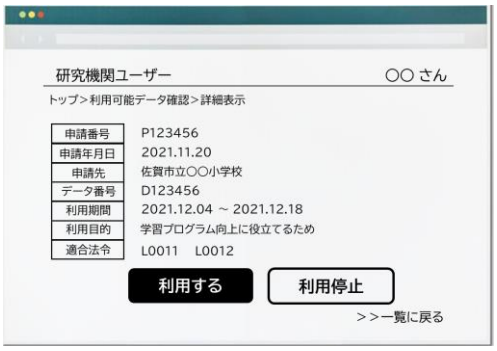

図 9　利用可能データ詳細画面

7.3 生徒および保護者の同意に基づくアクセス許可

システムにより自動アクセス許可が出来なかった場合に情報主体者に同意を求める手順について説明する．ここでは，データ利用申請を受けるアクターは生徒および保護者を想定する．画面遷移は次の通りである．

- 申請者は，7.1 の手順でアクセス申請書を提出する．システムは，データへのアクセスを許可せず，該当データの情報主体者へデータ利用申請が行われた旨の通知を行う．
- 情報主体者（生徒および保護者）は，生徒用メイン画面（図 10）からデータ利用申請通知を選択する．システムは，選択された機能を提供するための画面へ遷移を行う．

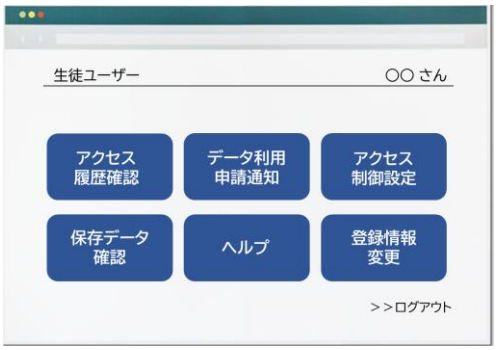

図 10　生徒および保護者用メイン画面

- 情報主体者は，データ利用申請通知一覧画面（図 11）から確認したい通知を選択する． システムは，選択された利用申請に対応する申請書情報を提示する．

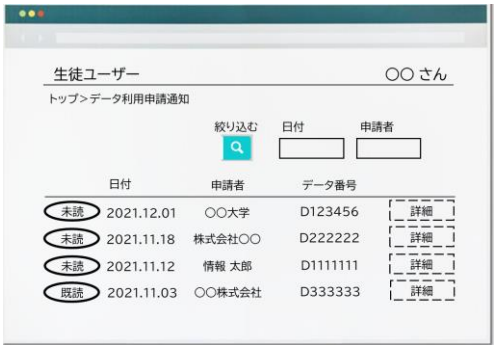

図 11　データ利用申請通知一覧画面

- 情報主体者は，データ利用申請通知詳細画面（図 12）で対象データや利用目的，適合法令が表示されるので，それを参照の上，アクセスの許可あるいは拒否を決定する．システムは，選択されたアクセス可否をデータベースに登録する．

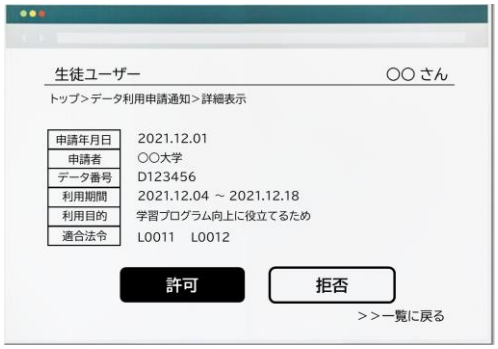

図 12　データ利用申請通知詳細画面

7.4 自動アクセス許可の取り消し

過去に自動的にアクセスが許可されているものの，情報主体者がアクセスを拒否することに切り替える手順について説明する．ここでは，データ利用申請を受けるアクターは生徒を想定する．画面遷移は次の通りである．

- 情報主体者は，生徒用メイン画面（図 10）からアクセス履歴確認を選択する．システムは，選択された機能を提供するための画面へ遷移を行う．
- 情報主体者は，当人のデータに関するアクセス履歴一覧画面（図 13）から確認したい履歴を選択する．システムは，選択されたアクセス履歴の詳細を表示する．

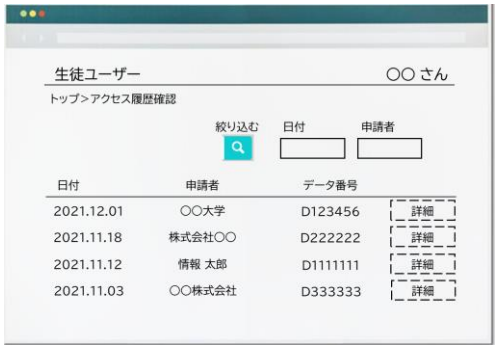

図 13　アクセス履歴一覧画面

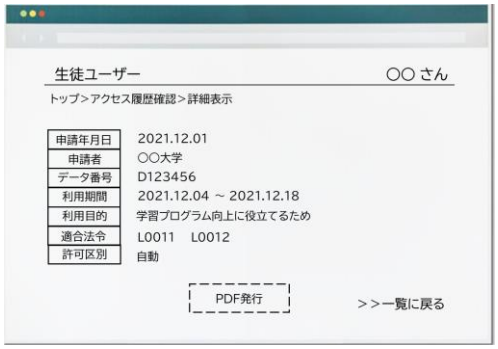

図 14　アクセス履歴詳細画面

- 情報主体者は，アクセス履歴詳細画面（図 14）で過去のアクセス履歴についての利用目的や適合法令を確認する．
- 情報主体者は，生徒用メイン画面（図 10）からアクセス制御設定を選択する．システムは，選択された機能を提供するための画面へ遷移を行う．
- 情報主体者は，アクセス制御設定画面（図 15）からアクセス設定を変更したいデータを選択する．システムは選択されたデータについて現在のアクセス可否状況を表示する．

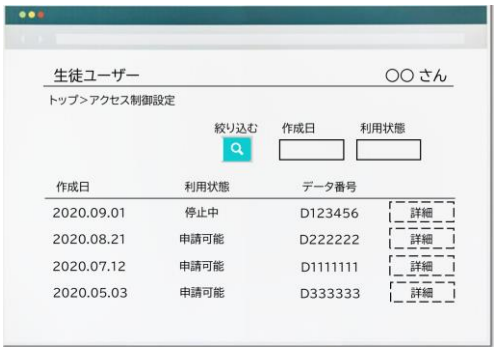

図 15　アクセス制御設定画面

- 情報主体者は，アクセス制御変更画面（図 16）でアクセスを維持するか停止するかを決定する．システムは，選択されたアクセス可否情報をデータベースに登録する．

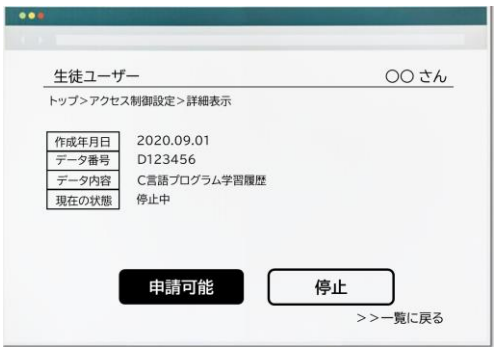

図 16　アクセス制御変更画面

8. プロトタイプの設計と実装

小学校プログラミング教育を対象とした利用許諾管理機構として作成するプロトタイプの設計を図 17 に示す．AWS を活用することで，次のような設計とした．

- 利用者のアクセス要求は API Gateway が受信する．
- API のパスごとに独立した Lambda 関数を用意し個別の機能を実装する．API Gateway は利用者の要求を適切な Lambda に転送する．
- Dynamo DB を用いて利用者の情報，データの情報，アクセス履歴を記録するためのデータベースを用意する．
- 各 Lambda 関数に，必要となるデータベースへのアク

セス権を与える．

- ウェブページ配信用の S3 を用いて利用者がウェブブラウザ上で情報を入力・確認できるようにする．すなわち，S3 はプロトタイプの利用者インタフェースを提供する．

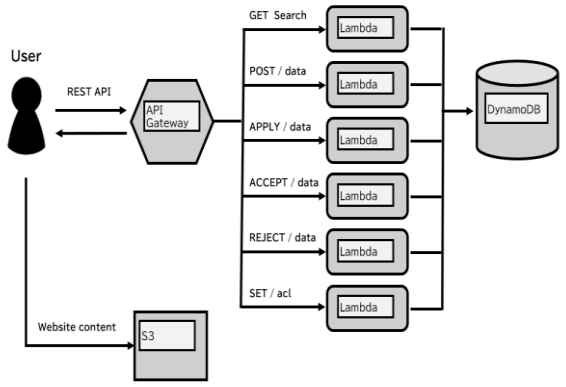

図 17　アプリケーションアーキテクチャ

8.1 API Gateway

プロトタイプの機能を実現するための API メソッドは，操作対象と動作の組み合わせによって定義する．具体的には以下の通りである．

- GET / webpage：web ページを取得する．
- GET search：検索を実行する．
- POST / data：データを入力する．
- APPLY / data：データへのアクセス申請を提出する．
- ACCEPT / data：データへのアクセスを許可する．
- REJECT / data：データへのアクセスを拒否する．
- SET / acl：アクセス制御管理を設定する．

ここでは，7.1 で示したデータ利用申請書提出を行う API の実装について紹介する．図 18 に示すように，申請情報として必要である利用目的や関係法令を利用者が入力し，それぞれの値について POST API メソッドを用いて送信を行っている．送信されたデータは，8.2 で説明する Lambda 関数が処理する．

API Gateway を介した処理を行うことで，多数のユーザーからのリクエストを同時に受け付けることが可能である．さらに，接続先を振り分ける必要がないため，処理する関数は固有の URL を指定するだけで済む．そのため，今回はプロトタイプ設計だが今後のシステム構築に向けた拡張性が高い．また，悪意のあるユーザーによって通信を操作されるリスクを低減できる．

API メソッドは入力データを検査し，その結果に応じてエラー処理や再入力等の処理も行っている．不適切な入力が行われた場合はユーザーにエラー情報をウェブサイトで表示する．これは，Lambda 関数とは独立して行われ，不正なデータ入力やデータベースへのアクセスを減らすことにつながる．

```
<script>
        // define the callAPI function
        var callAPI =
        (application_id,user_id,data_id,
        purpose,law,provided)=>{
            // instantiate a headers object
            var myHeaders = new Headers();
            // add content type header to object
            myHeaders.append
            ("Content-Type", "application/json");
            // using built in JSON utility
            // package turn object to string
            // and store in a variable
            var raw = JSON.stringify(
            {"application_id":application_id,"user_id":user_id,
            "data_id":data_id,"purpose":purpose,
            "law":law,"provided":provided});
            // create a JSON object with parameters
            // for API call and store in a variable
            var requestOptions = {
                method: 'POST',
                headers: myHeaders,
                body: raw,
                redirect: 'follow'
            };
            // make API call with parameters
            // and use promises to get response
            fetch("https://amazonaws.com/dev", requestOptions)
            .then(response => response.text())
            .then(result => alert(JSON.parse(result).body))
            .catch(error => console.log('error', error));
        }
</script>
```

図 18　利用申請書提出の API メソッド

8.2 Lambda 関数

8.1 で定義した API メソッドそれぞれに対して Lambda 関数を設定する．機能を実現するために入力および処理，出力を次のようにそれぞれ定義する．

- GET / webpage 入力：URL　出力：Web コンテンツ
- GET search 入力：検索条件　出力：データベース情報
- POST / data 入力：利用目的，適合法令　出力：利用申請書
- APPLY / data 入力：利用申請書　出力：アクセス可否判定結果
- ACCEPT / data 入力：データ ID 出力：データ内容，アクセス履歴
- REJECT / data 入力：データ ID 出力：利用申請通知，アクセス履歴
- SET / acl 入力：データ ID，アクセス可否　出力：アクセス権限情報

ここでは，7.1 のデータ利用申請書提出を行う Lambda 関数の実装について紹介する．8.1 で説明した API メソッドから送信された値をデータベースに書き込む処理を行っている．関数の内容は図 19 に示すとおりである．API メソッドから送られた値が入力として与えられ，それぞれの値をデータベースの書式へと変換する処理を行う．そして，データベースのテーブルを指定し，値を格納している．また，入力したキーが既に存在する場合は他の項目が書き換えられる．

以上のような手順でデータベースへの入力を行うため，利用者がデータベースの値を直接的に書き換えることはできない．すなわち，Lambda 関数により管理者の意図した

入力値に変換できることからシステムのセキュリティを高める効果が期待できる．

```
// Include the AWS SDK module
const AWS = require('aws-sdk');
// Instantiate a DynamoDB document client with the SDK
let dynamodb = new AWS.DynamoDB.DocumentClient();
// Use built-in module to get current date & time
let date = new Date();
// Store date and time in human-readable format in a variable
let now = date.toISOString();
// Define handler function,
// the entry point to our code for the Lambda service
// We receive the object that triggers the function as a parameter
exports.handler = async (event) => {
    // Extract values from event and format as strings
    let application_id = JSON.stringify(`${event.application_id}`);
    let user_id = JSON.stringify(`${event.user_id}`);
    let data_id = JSON.stringify(`${event.data_id}`);
    let purpose = JSON.stringify(`${event.purpose}`);
    let law = JSON.stringify(`${event.law}`);
    let provided = JSON.stringify(`${event.provided}`);
    // Create JSON object with parameters
    // for DynamoDB and store in a variable
    let params = {
        TableName:'application_form',
        Item: {
            'application_id': application_id,
            'user_id': user_id,
            'data_id': data_id,
            'purpose': purpose,
            'law': law,
            'provided': provided,
            'date': now
        }
    };
    // Return the response constant
    return 0;
};
```

図 19　利用申請書提出の Lambda 関数

8.3 S3 バケット

利用者に対してユーザーインターフェースを提供するためにデータを格納する必要がある．そこで，本プロトタイプでは AWS が提供するサーバーレスストレージの Simple Storage Service（S3）を利用する．HTML ファイルなどのウェブサイト用コンテンツを保存しておき，利用者からは API を介してアクセスしてもらう．

利用者インタフェースの例として，生徒ユーザーがアクセス申請通知を確認する際の画面を図 20 に示す．

図 20　アクセス申請通知確認画面

9. おわりに

本稿では，個人情報管理システムを活用し，小学校プログラミング教育向けの個人データ利用許諾管理機構のプロトタイプを提案した．本プロトタイプには，以下からアクセスできるようにしてある．

https://demonstrationforprograming-education20220204001.s3.ap-northeast-1.amazonaws.com/index.html

今後は，小学校プログラミング教育に関するデータ管理用としてのサービスを提供し，実現性や効果を確認する予定である．

8．で設計・実装したプロトタイプには，複数の Lambda が含まれる．Lambda は，サーバーレスコンピューティングの中心を担う存在である．これは，計算を実行するためのサービスである．API からのリクエストに対して起動し，計算結果をデータベースなどに出力する．リクエストに対する処理を定義しておく必要があるため，それぞれの関数が必要となる．現在，これらの実装を進めているが，評価結果を踏まえて機能の充実を図ってゆきたい．

プロトタイプの構想に当たっては，情報主体者と利用申請者を公平に扱う点を特に考慮した．また，データ利活用に対する社会的なニーズの高まりを考えると判定の自動化率を高めることが望まれる．個人の参加が増えると当然ながらデータ利活用の効率性が阻害される可能性はある．ただ，個人の権利は保証すべきであることはすでに多くの国において共通認識である。それらを両立させることが本システムの目標である．

また，将来的にはディープラーニングなどの技術を活用し利用許諾の自動化精度を高めたいと考えている．具体的には，多数の利用者から得られたデータ利用許諾状況の情報を基に法令法律や規制面からだけでなく人間の感情や世論を参考にすることで自動化が強化できると期待される．

参考文献

[1] General Data Protection Regulation (GDPR) – Official Legal Text（2018）, https://gdpr-info.eu/.
[2] OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data（2013）, https://www.oecd.org/sti/ieconomy/privacy.htm .
[3] 寺西司，掛下哲郎，個人情報管理システムを活用した情報銀行における利用許諾自動化の試み，第 20 回情報科学技術フォーラム（2021）．
[4] Amazon Web Service, https://aws.amazon.com/jp/.
[5] 文部科学省: GIGA スクール構想の実現パッケージ（2020），<https://www.mext.go.jp/content/20200219-mxt_jogai02-000003278_401.pdf>
[6] 砂原秀樹，山内正人，金杉洋，柴崎亮介，「情報銀行」構想とその技術的課題，マルチメディア・分散・協調とモバイルシンポジウム（2014）．
[7] 一般社団法人日本 IT 団体連盟 情報銀行推進委員会，「情報銀行」認定制度について（2021）．
[8] 掛下哲郎，新井康平，大月美佳，吉田豊昭，個人情報管理システム，特願 2004-4779 号, 2004 年 1 月．
[9] T. Kakeshita, K. Arai, M. Ohtsuki, “A personal data control mechanism for digital society”, Proc. Int. Proc. Int. Conf. on Advanced Information Networking and Application (AINA 2004), pp. 524-527, 2004.