京都工芸繊維大学における ID 管理システム midPoint の導入事例

永井 孝幸 [1)]

1) 京都工芸繊維大学 情報科学センター

nagai@kit.ac.jp

Implementation Case Study of the ID Management System 'midPoint' at Kyoto Institute of Technology

Takayuki Nagai[1)]

1) Center for Information Science, Kyoto Institute of Technology

概要

京都工芸繊維大学では 2022 年度の第 11 世代情報基盤計算機システム (System11) への更新に伴って ID 管理システムを OSS の midPoint を用いて内製した．midPoint は Evolveum 社が提供する ID ガバナンス・管理 (Identity Governance and Administration, IGA) 用ソフトウェアである．midPoint は Shibboleth や Grouper と並んで InCommon Trusted Access Platform における主要コンポーネントの 1 つであり，国外の大学では商用の ID 管理システムや独自開発システムからの置き換えに成功した事例が報告されている．

本学の既存の ID 管理システムは利用者原簿管理・アカウントプロビジョニング機能に加え，教務・人事システム連携，アカウント発行管理，利用者向けセルフサービスポータル，LMS 連携などの本学独自改修を施したベンダー製品であったが，これを midPoint を中心とした ID 管理システムとして再構築した．具体的なシステム構成や導入手順，システム更新に伴う旧システムとの並行稼働からの切り替えなど，本学における経験を共有することで高等教育機関における IAM(Identity and Access Management) 基盤の水準向上につなげたい．

1 はじめに

ID 管理システムはシステム連携の要となる部分である．セキュリティポリシーや組織運営体制と密接な関係があるため各大学により実現方法が異なり，商用システムによる実現 [1][2] や独自システムによる実現 [3][4] など様々な導入事例が報告されている．

京都工芸繊維大学では前システム (System10) において独自改修を施したベンダー製品を ID 管理システムに用い，これに OSS による統合認証基盤を組み合わせた構成でシステムを運用していた [5]．今回のシステム更新ではこの ID 管理部分を OSS の midPoint を中心に内製した．midPoint は最初のリリースから 10 年以上の歴史を持つソフトウェアであるが日本国内の大学での利用について公開された事例は見当たらず，本学の事例を共有することは価値があると考える．

本稿ではまず第 2 節で midPoint の概要について述べ，ID 管理ならびにサービス利用資格管理に関係する機能について紹介する．続く第 3 節で本学の ID およびアクセス管理システムに関する要求事項をまとめ，第 4 節で midPoint の導入結果について述べる．

2 midPoint の概要

midPoint では一般的な ID 管理機能 (原簿管理, プロビジョニングなど）に加え，セルフサービス用の Web UI も備えており (図 1)，一般ユーザ向けのサービスポータルを提供することもできる．この画面でユーザは登録情報の確認 (図 2) やパスワード変更, 利用サービス指定 (図 3) を行える．

図 1　ダッシュボード画面

midPoint の主な機能は，事前に定義されたマッピングルールに基づいて「ID 管理原簿に登録された個人（ユーザ）」と「データリソース (LDAP 等) に登録されたユーザ情報（アカウント）」の間の情報の整合性を双方向に保つことである．原簿情報を整形してデータリソースに出力するためのプロビジョニングツール

図 2　ユーザプロファイル画面

図 3　利用サービス指定画面

としての側面もあるが，定義ルールにもとづいてユーザ原簿を適切な状態に保つための機能が整備されていることが特徴である．

midPoint は CSV，リレーショナルデータベース,LDAP,ActiveDirectory 等様々な種類のデータリソースと連携し，ユーザ原簿へのデータの取り込みやアカウント情報の書き出しを行うことが出来る．データリソースの種類毎にコネクタが用意され，コネクタを追加することで対応するデータリソースの種類を増やすことが出来る．また，データリソースとの通信は標準プロトコル (JDBC,LDAP など) を用いて行われるため，連携先のリソースに追加のソフトウェアをインストールする必要が無い．

2.1　midPoint の機能

2.1.1　マッピング

midPoint ではデータリソース上のデータ項目とユーザ原簿の各項目との入出力対応付けをマッピングによって定義する．マッピングには単純な文字列処理だけでなく JavaScript や Groovy 言語を用いた関数も記述できる．マッピングによって定義された各属性について入力値・出力値の依存関係が自動的に検出され，入力値に変更があった場合は影響を受ける出力値が即座に更新される．ユーザ原簿とデータリソースの差分は midPoint 本体によって計算され，データリソース上の情報を最新状態に保つためのトランザクションが midPoint からコネクタを通じて発行される．

値の更新に対して「weak(値が未定義の場合のみ更新)」「normal(ユーザ・管理者による値の手動更新を許可)」「strong(midPoint の出力値を強制)」の 3 種類の強度を指定することが出来，例えばパスワードを保持する項目については weak マッピングを指定することで初期パスワードの登録を行うことができる．手動でのアドホックな変更を一切認めない項目に対して strong マッピングを用いることでセキュリティポリシーを強制することができる．

2.1.2　ロール, サービス, 組織と誘導 (inducement)

ユーザに種類別の権限を付与するための基本的な仕組みとして「ロール (Role)」が用意されている．ロールには「認可」と「誘導 (詳細は後述)」を設定することができ，ユーザにロールを割り当てることで，そのユーザに認可と誘導の設定内容が適用される．認可には原簿に対する操作 (read/write 等) の許可を種類別・属性単位に指定できるだけでなく，利用者ポータル上でアクセスできる画面・機能を限定することもできる．例えば，「管理者ロール」には原簿上の全ての属性と利用者ポータルの全ての機能の利用許可を設定し，「一般ユーザロール」には原簿上の自身のデータの read 権限のみを付与するといった使い方である．「誘導」は「ユーザがそのロールに所属することで自動的に登録対象となるリソース・ロール・サービス・組織」のことであり，例えばユーザを「管理者ロール」に割り当てると自動的に LDAP リソース上の管理者グループに登録されるようにする，といった使い方が出来る．

ロールに似た仕組みとして「サービス (Service)」と「組織 (Org)」が用意されている．機能的にはロールと同じであるが，管理画面や条件指定の際に別種のもの (それぞれ RoleType,ServiceType,OrgType) として区別される．サービスは「PC 端末」「プリンター」など提供サービスを表現するのに用いることが想定されており，誘導の機能と組み合わせることで「ユーザにあるサービスを割り当てることで特定の LDAP リソースにアカウントを登録し，更に LDAP 上の利用者グループのメンバーとして登録する」といった使い方が出来る．組織はユーザやリソースなどをグループとしてまとめて階層化することが想定されている．組織に誘導を設定することで，例えば「教員グループに所属するユーザに対して自動的に教員ロールを割り当てる」といった使い方が出来る．

2.1.3　アサインメント

ユーザに対してロール・組織・サービス・リソースを割り当てることを「アサインメント (assignment)」と呼び，アサインメントを通じて「ユーザが満たすべ

き状態」を定義する．アサインメントの結果とデータリソースの状態に差異があれば，その差異を埋めるためにデータリソースの状態が自動的に更新される．アサインメントにより表現されたユーザ状態がデータリソース上にどのように反映されるかは，各データリソースに対応するコネクタのマッピングによって決定される．例えば，ユーザに LDAP リソースを割り当てることで LDAP 上にアカウントを登録し，ユーザを組織に割り当てることでユーザを特定の LDAP グループに登録する，といった使い方ができる．前述の「誘導」と組み合わせることで，1 つのロールを割り当てるだけで複数のサービスや組織を芋づる式に割り当てることもでき,「複数リソースへの一括登録 (削除)」を「ロールの割り当て (解除)」として扱うことが出来る．

2.1.4 オブジェクトテンプレート

ID 管理システムではデータリソースから取り込んだ値をユーザ原簿に反映する際に，値を正規化したり派生値を計算することが必要になる．midPoint ではそのための仕組みとしてオブジェクトテンプレート (object template) の機能が用意されている．オブジェクトテンプレートにはユーザ原簿の属性値毎に出力マッピングを指定することができ，ユーザ原簿上の属性を入力値としてマッピング結果で属性値を更新することができる．例えば，firstName 属性と lastName 属性を組み合わせて fullName 属性を算出するような処理を記述するのに用いられる．

midPoint では更にオブジェクトテンプレート内でユーザ属性値を元にしてロール・サービス・組織・リソースを自動的に割り当てるルールを記述することもできる．この仕組みにより，例えばデータリソース上で管理者フラグが立っているユーザに対して自動的に管理者ロールを付与する，といったことができる．

3 IAM システムに関する要求事項

情報科学センターでは演習室端末，プリンタ，電子メール，無線 LAN，VPN 接続，ファイル共有，LMS など様々なサービスを提供しており利用者の種類（教員，学生など）によって提供するサービスの範囲や内容（利用可能なリソース上限など）に差を設けている．このようなサービス提供を実現するために本学の ID およびアクセス管理システム (IAM システム) が満たすべき要求事項をまとめる．

3.1 利用者原簿管理に関する要求事項

学務システムが出力する学生情報，人事システムが出力する教職員情報を取り込み，個人に一意に対応する生涯アカウントと学籍・職籍に対応する情報科学センターアカウント (CIS アカウント) の発行と管理が可能であること (**要求 1**)．なお，CIS アカウントと生涯アカウントは個人識別用 ID (KIT パーソナル ID) により相互に紐づけされていること (**要求 2**)．また，本学指定の ID 生成ルールにもとづいて KIT パーソナル ID の発行管理が行えること (**要求 3**)．さらにシステム管理者の操作でユーザ属性項目の追加が可能であること (**要求 4**).

3.2 サービス利用資格管理に関する要求事

利用者の種類ごとに利用できるサービスの範囲，初期状態で使えるサービスの範囲，割り当てるリソースの量 (ディスク容量など) を変えられること (**要求 5**). 例外的な利用者については利用可能サービス・割り当てリソースを手動で調整できること (**要求 6**)．また，一部のサービス（追加サービス）についてはオンライン研修受講をサービス利用資格付与の条件にできること (**要求 7**).

3.3 利用者ポータルに関する要求事項

パスワード変更ならびに原簿登録内容の確認をユーザ自身で行えること (**要求 8**)．また，追加認証情報 (ex.SSH 公開鍵) を登録できること (**要求 9**)．追加サービスをユーザ自身で有効化/無効化できること (**要求 9**) に加え，オンライン研修受講状況に応じて利用指定可能な追加サービスを変えられること (**要求 10**).

4 midPoint 導入結果

第 3 節で述べた要求を満たす IAM システムをmidPoint を中心に構築した．要求 4 から要求 10 は midPoint で対応し，要求 1 から要求 3 はデータ連係サーバ上の処理として自作している．

4.1 サーバ構成

System11 のサーバ群は ESXi 仮想化基盤 (HPE ProLiant DL360 Gen10/Intel Xeon Gold 5220R 2.20GHz) 上に仮想サーバとして構築されている．仮想サーバへのリソース割り当ては midPoint サーバ (CPU:4 コア, メモリ:16GB),LDAP(389ds) サーバ (CPU:2 コア, メモリ:4GB),ActiveDirectory サーバ (CPU:4 コア, メモリ:16GB) である．

System11 における利用者原簿管理システムと統合認証基盤の構成を図 4 に示す．midPoint は図中の利用者原簿管理システムに対応し，データ連係サーバを介して人事システム・学務システムとの自動連携を実現している．連携先のデータリソースには LDAP,ActiveDirectory, リレーショナルデータベース

の他，ファイルサーバ (NetApp)，メールサーバ，図書館システムなどがある．

旧システム (System10) ではサービス毎の利用者を Grouper で管理しており，現在でも Grouper のグループ情報を参照しているシステムが残っている．このため，midPoint 上での各ユーザのサービス選択状況をデータ連係サーバを介して Grouper にも反映させるようにしている．

4.2 midPoint システム構成

RedHat Enterprise Linux 8 上の rootless コンテナ環境 (podman) で midPoint コンテナを動作させている．可用性向上のためにクラスタ構成としており，コンテナ同士でクラスタとして動作させるためにコンテナのネットワークを host モードにしている．Evolveum 社による配布コンテナ[*1]を用いているが，配布コンテナ内の JDK には JavaScript のランタイムが付属していなかったため，JDK を OpenJDK11 に差し替えてビルドし直した．また後述するように midPoint 本体・コネクタに改修を施している．

midPoint 用データベースには pgpool でクラスタ化した Postgresql13 を用いた．標準のロードバランス設定では動作に問題があり disable_load_balance_on_wirte に always を設定した．

Web 画面のユーザ認証は midPoint の標準機能を用いて SAML 認証に対応させ，管理者アカウントでのログイン用にローカルアカウント認証も設定している．

4.3 改修内容

技術検証ならびに旧システムとの並行稼働で判明した不具合に対応するため，以下の改修を行った．

4.3.1 サロゲートペア対応

System10 の利用者情報を midPoint に登録する技術検証の過程で一部利用者の氏名が登録できないことが判明した．氏名の文字列に「サロゲートペア」が含まれているケースが該当し，ソースコードレベルでの修正を要した．文字列処理に String.charAt メソッドを使っている箇所を String.codePointAt メソッドに修正した他，オブジェクトのシリアライズに使用する Xalan をサロゲートペアの処理で問題を起こさない OpenJDK 付属の Xalan に切り替えて解決した．

4.3.2 LDAP コネクタの 389 Directory Server 対応

midPoint 上のユーザのロック状態を LDAP 上のアカウントに反映させる機能が 389 Directory Server に対応していなかったため，OpenLDAP 用の実装を参考に nsAccountLock 属性にユーザのロック状態を反映させるよう LDAP コネクタの改修を行った．

4.3.3 Grouper コネクタの大規模グループ対応

ユーザ数が数千人規模のグループを Grouper コネクタで midPoint に取り込む際，内部的に生成される XML 文書のサイズが肥大化してグループ同期処理の時間が大幅に長くなるだけでなくグループ管理画面上での操作に支障がでた．そこで，1 ユーザにつき 1 つの XML ノードを生成する実装を修正し，1 つの XML ノードに複数ユーザをまとめて登録する方式に改修することでこの問題を解消した．

4.4 導入手順

midPoint の導入は System11 の構想段階から自力で技術検証を行い，旧システムとの並行稼働を経て最終的な利用者原簿管理システムの切替まで約 2 年を費やした．本節では導入の各段階について述べる．

4.4.1 技術検証段階 (2021)

まずは技術検証環境を立ち上げて System10 の利用者原簿を取り込み LDAP/AD にアカウントを登録するところから着手した．midPoint の標準ユーザスキーマでは属性が不足するため，ユーザ種別識別用の属性 (利用者区分コード, 職種コード, 部局コードなど) や在籍管理 (学籍離脱日，退職日など)，UID/GID，追加認証情報 (SSH 公開鍵など)，サービス選択状況を保持するための属性をカスタムスキーマに登録した．スキーマに追加した属性はユーザプロファイル画面の項目に自動的に追加される (図 5) ので，プロファイル編集画面自体のカスタマイズは不要であった．

文字列属性については注意が必要であり，スキーマ上で「インデックス対象」に指定しておくことで検索の対象とすることができるが，その代わりに最大長が 255 文字に制限される．そのため SSH 公開鍵など長い文字列を保持する属性についてはインデックス対象から外す必要がある．また midPoint では文字列属性の一致判定の際に用いる normalizer を切り替えられるようになっており，動作検証の結果 PassThroughPolyStringNormalizer を用いることにした．初期設定の AlphanumericPolyStringNormalizer では文字列中の英数字以外の文字を削除したうえで一致判定が行われるため，例えばユーザ名 hkimura と h-kimura でユーザ名が衝突し，既存アカウントの引継ができないことが判明したためである．

DatabaseTable コネクタを用いて Postgresql 上のデータを取り込むことで利用者原簿の取り込みを行った．この際，サロゲート文字の取り込みで問題が生じ

[*1] https://github.com/Evolveum/midpoint-docker

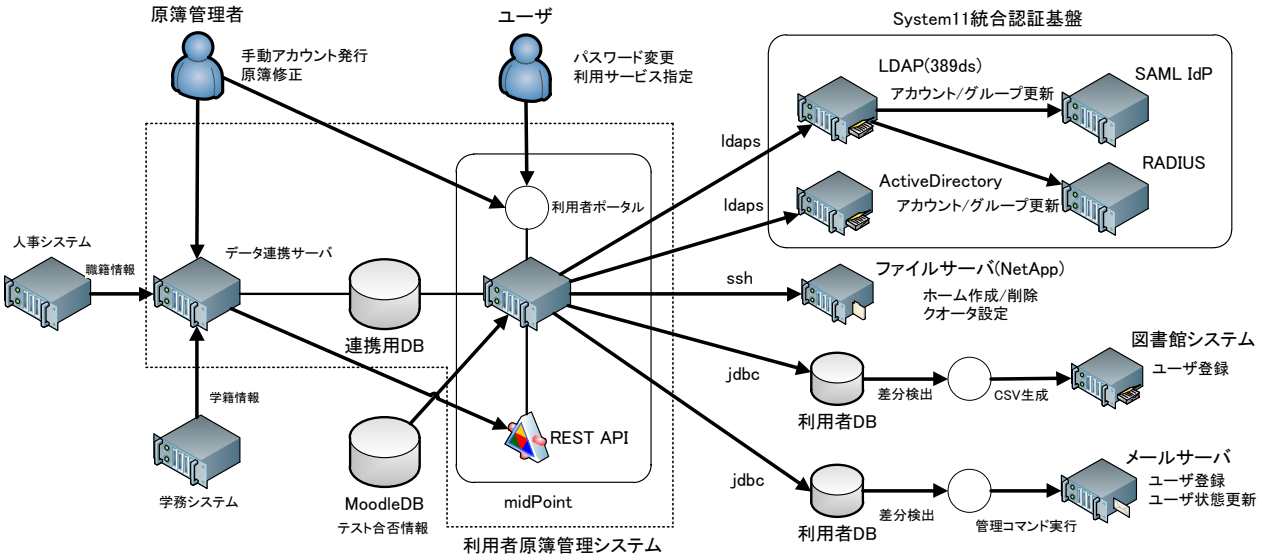

図 4　System11 構成図 (抜粋)

ること，データベースのカラム名に英大文字が使われていると正常に動作しないことが判明した．データの取り込み周期についてはインターバル指定, スケジュール指定, リアルタイム更新 (live sync) の 3 通りの方式について動作検証を行い，スケジュール指定方式とリアルタイム方式を併用することにした．インターバル指定では処理が異常終了した際に自動復旧せず，データ取り込み処理が停止してしまうことがあったためである．リアルタイム更新はデータベース上の timestamp 属性と最新のデータ取り込み実行時刻を比較して更新のあったデータエントリだけを取り込む方式であり，実用上十分な頻度 (数分間隔) で安定して最新データの取り込みが行えることを確認した．

LDAP コネクタの設定では，LDAP スキーマ通りのデータ型になるようマッピングを定義する必要があった．例えばバイナリ値を保持する項目は Base64 エンコードした文字列を出力するのではなく，バイナリ値を出力するように設定する必要がある（最終的な Base64 文字列への変換は LDAP スキーマに基づいてコネクタ自身が行う）．MSChapv2 認証に用いる NT

図 5　カスタム属性に対して生成された入力欄

Listing 1　LDAP グループへのマッピング定義例

```
<association>
 <ref>ri:maillist</ref>
 <displayName>MailList Association</
     displayName>
 <tolerant>true</tolerant>
 <exclusiveStrong>false</exclusiveStrong>
 <kind>entitlement</kind>
 <intent>maillist</intent>
 <direction>objectToSubject</direction>
 <associationAttribute>ri:member</
     associationAttribute>
 <valueAttribute>ri:dn</valueAttribute>
</association>
```

パスワードハッシュ値の生成も，マッピング定義に MD4 ハッシュ値を計算する Groovy のコードを書くことで実現した．

また LDAP コネクタの association 項目にマッピングを定義することで，サービスや組織のメンバーが LDAP 上のエントリに反映されることを確認した．リスト 1 は，グループメンバーの dn 値を LDAP グループの member 属性に登録する association の例である．あるサービス（あるいは組織, ロール）を LDAP リソースに「エンタイトルメント」として割り当てることで，association の定義に基づいて LDAP グループが更新されるようになる．

midPoint からリモート計算機上のスクリプトを実行するための仕組みとして SSH コネクタが用意されている．SSH 接続により任意のコマンドをリモート実行することができ，ファイルサーバ上にユーザのホー

Listing 2　SSH コネクタの設定例

```
<scripts>
  <script>
    <host>resource</host>
    <language>bash</language>
    <argument>
      <name>username</name>
      <path>$focus/name</path>
    </argument>
    <code>/opt/midpoint/toolbox/bin/
        createUserHome.sh $username</code>
    <criticality>partial</criticality>
    <operation>add</operation>
    <kind>account</kind>
    <order>after</order>
    </script>
</scripts>
```

ムディレクトリを作成するといった処理に用いることが想定されている．このコネクタは補助コネクタとして実装されており，他のコネクタと組み合わせて使用する必要がある．なお，補助コネクタを設定すると midPoint のコネクタ設定ウィザードが利用できなくなるため，コネクタ設定にはコネクタ定義の XML を直接編集する必要がある．

具体的な設定例が公開されていないため試行錯誤を要したが，以下の設定により LDAP コネクタとの組み合わせで動作させることができた：

- LDAP コネクタの script 実行機能を disable に設定し，additionalConnector 項目に SSH コネクタを記述する
- SSH コネクタ側の script 実行機能を enable に設定する
- LDAP コネクタの script 属性に実行するスクリプトを書く

例えば，ユーザを新規登録した際にリモートホスト上でシェルスクリプト createUserHome.sh を引数付きで実行するにはリスト 2 のように書けばよい．

midPoint には外部サービスの REST API と連携する汎用のコネクタが用意されていないが，REST API を呼び出すシェルスクリプトをこの SSH コネクタを通じて呼び出すことでユーザ原簿の更新に応じて REST API を呼び出すことが出来る．これを利用し System11 では SSH コネクタ経由で NetApp の REST API を呼び出すことでユーザディレクトリのクォータを自動設定している．

4.4.2　並行稼働段階 (2022)

System10 から System11 へ移行する際，システム切替を年度途中で行う関係から「一般ユーザ（学生，教職員）の目に触れる部分は System10 のまま，認証基盤だけを System11 に切り替える」という並行稼働を行うことになった (図 6)．アカウント発行と利用者ポータルについては System10 の利用者原簿管理システムを継続利用し，System10 の Grouper ならびにデータ連係サーバのデータを取り込むことで利用者原簿の情報を midPoint に中継する構成とした．

Grouper によるグループ管理と midPoint によるグループ管理が LDAP 上で競合しないようにするため，Grouper 上でのグループ所属については memberOf 属性，midPoint 上でのグループ所属については eduPersonEntitlement 属性にマッピングを行った．

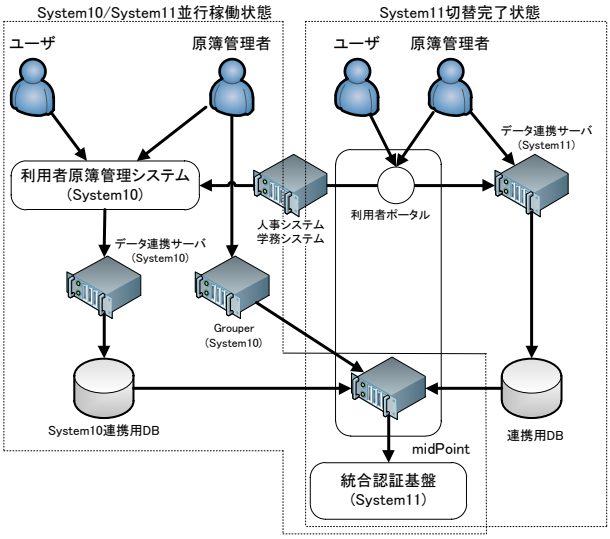

図 6　System11 切り替え時の構成

4.4.3　最終切替段階 (2023)

アカウント発行と利用者ポータルについても System11 に完全に切り替え，一般ユーザが midPoint を直接利用する状態になった．利用者原簿ならびに利用サービス指定は並行稼働段階における各ユーザの状態を引き継ぎ，設定作業のやり直しを一般ユーザに強いることがないようにした．

2023 年の 4/1 に前年度ユーザの一括削除 (約 1,200 件)・更新 (約 2,600 件) と新年度ユーザの一括登録処理を学務システムとの自動連携により行い，1215 ユーザの新規登録・プロビジョニングを 50 分程度で終えることが出来た．

4.5 midPoint を用いた利用資格の管理

本節では第 3.2 節で述べた要求事項を満たすように midPoint 上でロール・サービス・リソースをどのように組み合わせたか説明する.

4.5.1 ロール・サービス・リソースの自動割り当て

情報科学センターでは発行するアカウントに対して利用者区分コードを割り当て，アカウントを職種や所属に応じて 52 通りに分類 (教職員・学生・その他) している．提供するサービス内容は利用者区分コードと対応しているため, midPoint 上でも利用者区分コードをもとにロール・サービスを自動割り当てする方式とした．そのためにロール・サービスのスキーマを拡張し，自動割り当て対象となる利用者区分コード（複数）を保持できるようにしている．リソースについては認証基盤側の構成に変更があった時に備え，直接ユーザに割り当てるのでは無く，サービスからの誘導として間接的に割り当てる方式とした.

オブジェクトテンプレート内のマッピングで利用者区分コードに合致するロール・サービスを検索し，ユーザの assignment 属性に登録することで自動割り当てが完了する (リスト 3)．ここでマッピングの強度に strong を指定することで，利用者区分に変更があった場合にロール・サービスの割り当ても修正されるようにしている.

4.5.2 手動割り当てと自動割り当ての共存方法

midPoint 上でユーザに割り当てるサービス・ロール・組織には自動割り当てによって強制的に登録 (解除) したいものと，ユーザ・管理者の操作によって手動で登録（解除) したいものがある．midPoint において割り当て内容はユーザの assignment 属性に保持されるが，単純に strong マッピングによって assignment 属性を強制的に更新してしまうと，手動で割り当てた内容も合わせて更新されてしまい，手動設定内容が失われてしまう．この問題に対応するために midPoint には出力マッピングの値域を定義する mapping range の機能が用意されている.

この機能を用いるためには，assignment に登録する値の subtype 属性に値域を区別するための識別子を登録しておき (リスト 4 の例では system11default を登録)，これに対応するように出力マッピングの値域を「subtype 属性に含まれる文字列」で定義しておく (リスト 5)．この設定により，マッピング強度に strong を指定しても更新対象となる assignment 属性の値は subtype によって分離される．なお手動で設定した割り当てについては subtype は空になっている.

Listing 3　利用者区分によるサービスの自動割り当て

```
<expression>
 <assignmentTargetSearch>
  <targetType>ServiceType</targetType>
  <filter>
    <q:equal xmlns="">
     <q:path>extension/
         autoassignUserCategory</q:path>
     <expression>
      <path>$userCategory</path>
     </expression>
    </q:equal>
  </filter>
 </assignmentTargetSearch>
</expression>
```

Listing 4　assignment に subtype を指定する例

```
<expression>
  <assignmentTargetSearch>
    <targetType>ServiceType</targetType>
      <filter>
        <q:equal xmlns="">
          <q:path>identifier</q:path>
            <expression>
              <path>$defaultServices</path>
            </expression>
          </q:equal>
      </filter>
      <assignmentProperties>
        <subtype>system11default</subtype>
      </assignmentProperties>
    </assignmentTargetSearch>
</expression>
```

System11 では「system10(System10 から引き継いだ利用サービス指定)」「system11default(System11 上で強制的に登録するサービス・組織・ロール)」という subtype を設けることで System11 上の手動設定と自動割当が共存するようにしている.

4.6 ロールとサービス組織の設計

利用サービス指定に関する要求を満たすためにサービスを組織化し，ロール毎にアクセスできるサービスグループを制限している.

- CIS 初期サービス
  全ての CIS アカウントに対して初期状態で有効化するサービスを組織化したもの (ex. LDAP,Moodle, 共用端末, ホームディレクトリ)
- CIS 基本サービス
  確認テスト合格後の CIS アカウントに対し，利用サービス指定により有効化/無効化することを許

Listing 5　subtype を用いた出力マッピング対象の制限

```
<target>
  <path>assignment</path>
  <set>
    <condition>
      <script>
        <code>
          assignment?.subtype.contains('
              system11default')
        </code>
      </script>
    </condition>
  </set>
</target>
```

可するサービスを組織化したもの (ex. 電子メール, 共用プリンタ, 無線 LAN 接続, 多要素認証)

- CIS 追加サービス
  管理者により手動で有効化するサービスを組織化したもの (ex.VPN 接続,eduroam)

ロール内で「ある組織に属するサービスへのアクセスを許可する」というアクセス権限を設定することで, ユーザがサービス利用指定画面で閲覧することのできるサービスを制限することができる．System11 では利用指定可能サービスを制限するために以下のロールを設けた：

- CIS Initial User
  アカウント初期登録済みで確認テスト合格前の状態に対応するロール．このロールを持つユーザは「CIS 初期サービス」に含まれるサービスを参照できる．
- CIS Standard User
  確認テスト合格済みで標準サービスを利用可能な状態に対応するロール．このロールを持つユーザは「CIS 基本サービス」に含まれるサービスを参照できる．
- CIS ID Admin
  アカウント発行・名寄せ作業担当者用のロール．このロールを持つユーザは原簿全体の閲覧が可能になる．

CIS Initial User ロールをユーザの利用者区分に基づいて自動的に付与することで，統合認証基盤へのアカウントプロビジョニングと必須サービスの有効化を行う．確認テストの合格状況を全学 LMS のデータベースから live sync で取り込み，合格者に対して CIS Standard User ロールを自動付与することで要求 10 を実現している．

5　まとめ

midPoint を用いることで要求事項を満たす ID 管理システムを内製することができた．midPoint には各種管理作業を行うための GUI が用意されているが，本学の要求事項を実現するには midPoint 固有の概念を理解した上で XML 定義ファイルを直接記述することが必須であった．なかでも属性マッピングの処理は Groovy 言語で柔軟に記述できるが，midPoint の関数ライブラリやデータ構造を理解する必要があり実質的にプログラミング作業が必要であった．

定常利用において安定稼働しており新システムへの移行はひと段落した状態であるが，年度またぎ処理などの業務フローを確立することが今後の課題である．

謝辞

本研究の一部は JSPS 科研費 JP20H04297 の助成を受けたものである．

参考文献

[1] 大勝瀬川, 隆彦辻澤, みゆき石橋．学内の様々な運用形態に対応したコンパクトなアカウント管理システムの実現．学術情報処理研究, No. 16, pp. 59–70, 2012.

[2] 中山仁, 甲斐郷子．Sun identity manager を用いた「小さな」統合 id 管理システムの構築．*View Point*, Vol. 10, pp. 80–83, 2010.

[3] 田中克明, 山中達哉, 松村芳樹, 高見澤秀幸. 大学構成員情報の管理・連携システムの構築. 学術情報処理研究, Vol. 19, No. 1, pp. 50–57, 2015.

[4] 櫻田武嗣, 三島和宏, 石橋みゆき, 萩原洋一ほか. 管理運用システム 「salut」 の概要．研究報告インターネットと運用技術 (IOT), Vol. 2016, No. 3, pp. 1–6, 2016.

[5] 永井孝幸, 山岡裕美, 桝田秀夫．京都工芸繊維大学における利用者原簿管理基盤の強化と連携サービスの構築. 情報処理学会研究報告 第 25 回 CLE 研究発表会, Vol. 2018-CLE-25, No. 9, pp. 1–8, jun 2018.