情報処理学会論文誌 Vol.55 No.5 1485–1497 (May 2014)

学術組織間デジタル資料分散共有システム「ARCADE」の開発

松平 拓也[1,a)] 中村 素典[2] 山地 一禎[2] 西村 健[2] 高田 良宏[1] 笠原 禎也[1]

受付日 2013年8月19日, 採録日 2014年2月14日

概要：本稿では，組織横断型の共同研究において，各ユーザがそれぞれ保持しているコンテンツを安全・安心に相互参照可能な環境の構築を目的として，学術組織間デジタル資料分散共有システム「ARCADE」の開発を行った．ARCADE のフレームワークに Shibboleth を採用し，コンテンツを中央に集中させずに各組織で分散管理できるようにした．また，GakuNin フェデレーションに適用させ，組織を超えたユーザ認証を可能にした．さらに，GakuNin mAP を利用することで，ユーザの特性に応じたきめ細かなアクセス制御を可能とした．最後に，構築したシステムの動作検証および性能評価を行い，提案システムを実運用させるにあたって十分機能することを検証した．本稿では，開発したシステムである ARCADE の設計思想と技術的な解決法，検証試験の結果を示し，今後の展望を議論する．

キーワード：Shibboleth，学認，フェデレーション，仮想組織，コンテンツ共有

Development of "ARchive system for Cross-reference Across Distributed Environment (ARCADE)"

Takuya Matsuhira[1,a)] Motonori Nakamura[2] Kazutsuna Yamaji[2]
Takeshi Nishimura[2] Yoshihiro Takata[1] Yoshiya Kasahara[1]

Received: August 19, 2013, Accepted: February 14, 2014

Abstract: In this paper we introduce the ARchive system for Cross-reference Across Distributed Environment (ARCADE). The ARCADE system was developed to provide an environment in which users can easily and safely cross-refer their contents over organizational boundaries. We applied Shibboleth as a base framework of ARCADE so as to share their contents under management of each organization without centralized storage system. Furthermore, we adapted the ARCADE to the GakuNin federation to leverage user authentication beyond the organization. Various access control according to member attribute of each user was also implemented by utilizing the GakuNin mAP. Evaluation of the architecture and performance of ARCADE demonstrates that the proposed system works sufficiently for practical use. In this paper we explain the design and technical solutions of ARCADE, describe the performance, and discuss our future work.

Keywords: Shibboleth, GakuNin, federation, virtual organization, content sharing

1. はじめに

高等教育機関や研究機関では，学術論文や紀要などの文献，実験や観測などによって得られたデータをはじめとして様々な学術コンテンツ（以下，コンテンツという）が数多く蓄積されている．最近では，複数の機関で連携して調査研究・教育交流・情報発信などを行う機会が増加しており，これらのコンテンツは保有する機関内の研究者だけにとどまらず，機関外の研究者からも相互参照の要望が高まってきている．

各機関で蓄積されたコンテンツを相互に参照するための管理手法の 1 つとして，機関リポジトリがある [1]．機関

[1] 金沢大学
Kanazawa University, Kanazawa, Ishikawa 920–1192, Japan
[2] 国立情報学研究所
National Institute of Informatics, Chiyoda, Tokyo 101–8430, Japan
[a)] takusng@kenroku.kanazawa-u.ac.jp

リポジトリに登録されるコンテンツには，文献だけではなく，教育や研究において生産される教材やデータ，ソフトウェアなど幅広い種別のものが想定されている [2]．また，機関リポジトリで多く使われるシステムを利用して，データリポジトリが運用されているケースもある [3], [4]．しかしながら，こうしたリポジトリは，コンテンツをインターネット上に公開することを目的としたものであり，複数の組織に所属する研究者間で，研究の過程で生産されるコンテンツを共有することを目的としたものではない．

組織横断型の共同研究で取り扱われるコンテンツの多くは未公表な知的財産であり，特定の研究者間のみで共有される秘匿情報である．したがって，情報公開を主目的としたリポジトリ上で，任意のユーザで構成される個々のグループが，様々な共有ポリシを設定し，このようなコンテンツを自在に扱える仕組みを実現するのは困難である．すなわち，リポジトリでの公開の前段階にあたる，研究過程における研究者間でのコンテンツの共有システムの要件を整理し，それを実現する環境を整備する必要がある．

組織横断型共同研究におけるコンテンツ共有システムに求められるべき要件には，以下の 4 つがある．1 つ目の要件は「秘匿性を保つためのコンテンツの分散管理（Confidentiality）」である．本システムで取り扱うコンテンツは秘匿性が高いがゆえに，保有者それぞれの所属組織によってコンテンツの配置場所に対するポリシが異なるケースが想定される．したがって，ある特定のシステム管理者のもとで運用される（自組織外の）中央システムで集中的に蓄積・管理するのではなく，保有者もしくは保有者が信頼するシステム管理者のもとでそれぞれ分散管理できることが望ましい．2 つ目の要件として「信頼できるユーザ情報管理（Reliability）」があげられる．所属組織が異なる集合体であったとしても，保有者が許可した研究者だけがコンテンツに確実にアクセスできる必要がある．そして，3 つ目の要件として「多様なアクセスポリシの管理（Flexibility）」があげられる．コンテンツ保有者のポリシに応じて，様々なユースケースに対応できることが望まれる．最後に，4 つ目の要件として「特定のプロジェクトに依存しない拡張性のあるコンテンツ管理（Scalability）」があげられる．研究プロジェクトの形態や規模に依存することなく，コンテンツの格納場所が広範に分散された場合でも，大規模な 1 つのシステムのようにコンテンツを一元的に取り扱うことができることが望まれる．2 章において詳しく述べるが，こうした条件をすべて備えたコンテンツの共有システムは，現在のところ報告されていない．

そこで本研究では，これら 4 つの要件を満たす学術組織間デジタル資料分散共有システム “ARchive system for Cross-reference Across Distributed Environment”（以下，ARCADE という）を開発するとともに，その動作検証と性能評価を行った．

2. システムの必要条件と関連研究

本章では，組織を越えたコンテンツの相互共有を達成するために必要な 4 つの必要条件を説明する．そして関連研究において，これらの要件がどれだけ実現されているかを議論する．

2.1 システムの必要条件

- 秘匿性を保つためのコンテンツの分散管理（Confidentiality）

組織横断型の共同研究において，そこで扱われるコンテンツの多くは未公表の知的財産であり，コンテンツの配置場所が重要になる．基本的には，コンテンツは保有者が所属する組織内で運用しているサーバに配置できるようにする．そして，自組織内で運用しているサーバにコンテンツを配置したままで，保有者が必要なユーザに対して必要な分だけを参照させることが可能な機構となるようにする．

- 信頼できるユーザ情報管理（Reliability）

コンテンツに対しては，保有者が許可した者だけが確実にアクセスできるようにする．そのためには，コンテンツにアクセスしようとしている者を正しく識別できることが重要になる．つまり，組織を超えたユーザに対しても認証が正しく行われるために，信頼性が高いユーザ情報管理が可能な機構となるようにする．

- 多様なアクセスポリシの管理（Flexibility）

コンテンツに対するアクセスポリシは，保有者個人だけがアクセス可能，研究プロジェクトメンバのみアクセス可能，特定のプロジェクトには公開などコンテンツの保有者によって様々なケースが想定される．そのため，多様なアクセスポリシに対して柔軟に対応できるようにする．

- 特定のプロジェクトに依存しない拡張性のあるコンテンツ管理（Scalability）

すべての研究プロジェクトが必ずしも十分な予算を持つわけではなく，もし十分な予算があったとしてもプロジェクトごとに共有システムを構築するのは非効率である．そのため，プロジェクトの規模や分野に依存することなく，多くのプロジェクトが利用可能な機構となるようにする．

特に，研究プロジェクトの形態や規模によっては，コンテンツの格納場所が広範に分散することが想定される．さらに，容量の大きなコンテンツを多数扱うことによるリソース不足により，別途新しい格納領域を追加するケースも想定される．そのようにコンテンツの格納場所が広範に及んだり増加したりする場合においても，ユーザはあたかも大規模な 1 つのシステムにアクセスしているような感覚で，一元的にコンテンツを取り扱うことが可能な機構となるようにする．

2.2 関連研究

2.1 節で説明した 4 つの要件における先行事例について述べる．

Sasaki らは，異なる企業間で立ち上げた開発プロジェクトにおいて，ソースコードや仕様書などを安全にやりとりする手法について提案している [5]．Xen や VMware などの仮想プラットフォームを用いて，プロジェクト管理者がメンバ全員の仮想クライアントを制御ポリシの下で管理することで，安全にコンテンツをやりとりするという手法である．本手法により，プロジェクト管理者の下，秘匿性を保ったコンテンツの分散管理が可能となる．しかし，ユーザ情報やアクセスポリシはプロジェクト管理者が集中管理する仕組みのため，プロジェクト管理者を選定し，管理者にすべて設定を依頼する必要がある．そのため，設定変更のたびにプロジェクト管理者に依頼しなければならず，柔軟性に欠ける．さらに，環境構築の面においても仮想リソースをメンバ分用意し，それぞれにポリシを定義する必要があるなど金銭的および運用的コストにより拡張が難しい．

先行事例で示されている組織間連携における別の手法として Federated Identity Management（以下，FIM という）がある．FIM とは，フェデレーションと呼称される，決められたポリシに合意した組織の集合体において，ユーザの認証・認可をそれぞれの組織で行い，ユーザに対応づく情報をフェデレーション内で共有する分散サービスアーキテクチャである．代表的な認証プロトコルの 1 つとして SAML [6] が利用されており，実装されたミドルウェアとしては，Shibboleth [7] や simpleSAMLphp [8] が代表的である．Hatala らは各組織に分散するリポジトリを Edusource Community Layer（ECL）と呼ばれるミドルウェアインフラを利用して統合的に扱う手法を提案している [9]．その際，ユーザ情報管理には FIM を利用している．そのため，ユーザ情報はコンテンツを直接扱う組織ではなく，上位レベルである各機関が適切に運用しているものを利用することができる．しかし，アクセス制御は所属機関や身分などの大きな単位で行うことが想定されており，多様なアクセス制御を実現することが難しい．さらに，各組織は保有するコンテンツを ECL に対応したリポジトリフォーマットで格納する必要があり，フォーマットが異なる場合やフォーマットの変更のたびにシステムを修正する必要があるために拡張が難しい．また，Rieger らはクラウドストレージ領域を統合的に扱う方法を提案している [10]．本手法も同様に，機関レベルで管理された FIM を利用しており，コンテンツを扱うインタフェースは汎用性を持たせて設計している．しかし，クラウドストレージは個人利用のものだけを対象としており，複数のユーザでの利用は今後の課題となっている．

また最近では，Dropbox [11] や Google Drive [12] に代表される，数多くの無料のオンラインストレージが提供されている．これらのサービスを利用することは，自組織外のサーバにコンテンツを配置するということになり，クラウドを契約する外部委託の一形態になるといえる．外部委託したクラウド上に重要なコンテンツを配置することにおいては，これまでに，様々なリスクがあることが報告されている [13]．また，高等教育機関における情報セキュリティ対策におけるガイドライン [14] には，「外部委託の可否の原則として，重要な情報を取り扱う情報処理業務を外部委託により行うことは，情報漏えい等のリスクにかんがみ，これを原則として禁止する．重要な情報とは，これが不適切に取り扱われた場合に，利用者の権利利益に重大な損害を与え，あるいは，利用者及び本学の安全に重大な懸念が生ずる情報をいう」と記載されている．プロジェクトで扱うコンテンツはここで示されている重要な情報に相当する．このように，現状ではコンテンツを外部委託したクラウド上に配置することはリスクが高く，ガイドラインにおいても原則的に禁止していることから，多くの大学ではセキュリティポリシにより外部委託したクラウド上にコンテンツを配置することは認めていない．また，ユーザ情報管理においては，Sasaki らの例ではプロジェクト管理者が，FIM を利用する場合は各組織が管理するいわば「承認制」であるのに対し，Dropbox や Google Drive などの無料のオンラインストレージは，ユーザがメールアドレスを登録する「自己申告制」である．そのため，ユーザ情報の信頼性に欠ける．そして，アクセスポリシはフォルダごとに毎回ユーザに対してメールを送信して招待する必要があったり，階層的にアクセス制御を行うことができなかったりと，アクセスポリシの柔軟性に欠ける．

表 1 先行事例における必要条件の達成度
Table 1 Achievement of necessary conditions by precedent cases.

	Confidentiality	Reliability	Flexibility	Scalability
Takayuki Sasaki et al.	○	○	×	×
Marek Hatala et al.	○	○	×	×
Sebastian Rieger et al.	○	○	×	○
Dropbox Google Drive	×	×	×	○

これらをまとめたものを**表 1** に示す．この表から分かるように，我々が必要とする要件を満たす機構は存在しない．

3. ARCADE の設計

図 1 に ARCADE の動作概念図を示す．本章では，2.1 節で述べた 4 つの要件を満たすための ARCADE の設計思想について説明する．

3.1 コンテンツの分散管理（Confidentiality）

2.1 節で述べたように，取り扱うコンテンツの重要性か

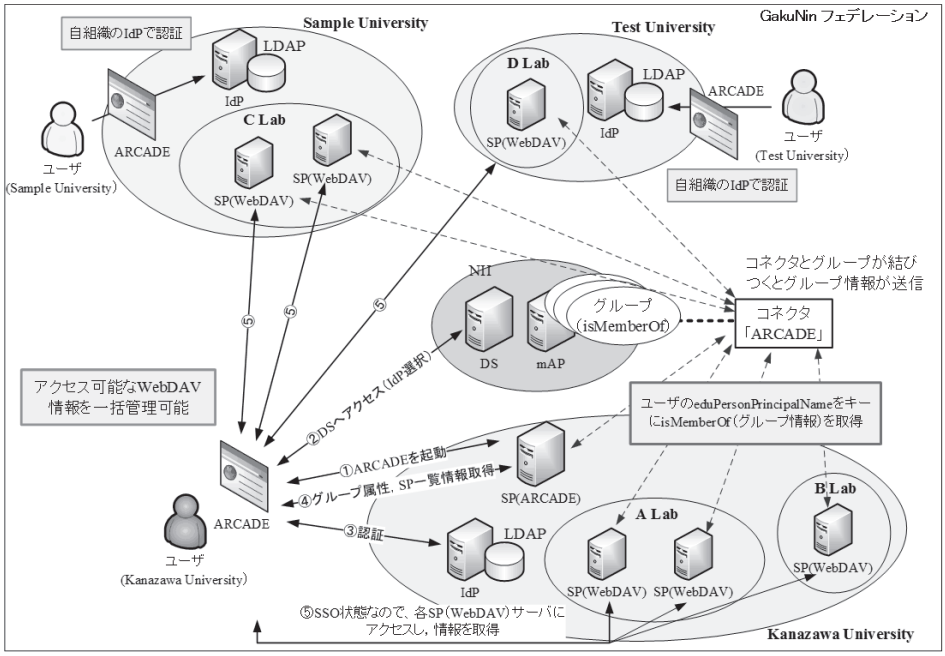

図 1 ARCADE 動作概念図
Fig. 1 Conceptual image of ARCADE.

ら，それぞれの組織が保有するコンテンツを中央に集めるのではなく，保有者が信頼できる場所にそれぞれ配置できるようにしたい．

そこで我々は，ARCADE のフレームワークとして，2.2 節で議論した関連研究において実績のある FIM を採用した．そして，ミドルウェアには Shibboleth を選定した．Shibboleth は 3 つのシステムから構成される．

- Identity Provider（IdP）
  - ユーザを認証する．
  - ユーザ属性情報を SP に送信する．
- Service Provider（SP）
  - ユーザの認証を IdP に要求する．
  - ユーザの属性を IdP から受信し，アプリケーションに渡す．
- Discovery Service（DS）
  - 複数の IdP が存在する場合に，ユーザが適当な IdP を決定するための情報を提供する．

図 2 に示す動作概念図に基づいて，Shibboleth の動作を説明する．図では，A 大学所属のユーザが B 大学の SP にアクセスを試みた場合を仮定している（①）．このとき，SP はユーザの認証を促すために，DS にリダイレクトを行い，ユーザに IdP を選択させる（②）．DS は利用可能な IdP のリストをユーザに提示し，ユーザは自組織の IdP で，ID/パスワード認証やクライアント証明書認証などの方法でユーザ認証を行う（③）．IdP は SP に認証結果を返し，成功の結果を受け取った場合に，SP は必要な属性を IdP に要求し，その返却値を SP のアプリケーションに渡す（④）．SP はその情報を基に，ユーザの属性に応じたサービスを提供する（⑤）．

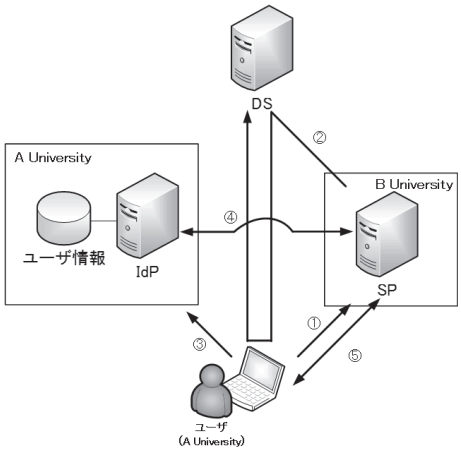

図 2 Shibboleth 動作概念図
Fig. 2 Conceptual image of Shibboleth.

各組織のコンテンツ格納場所を SP として設計することで，コンテンツの実体をそれぞれの組織に配置して管理できるように設計した．さらに，組織ごとに個別の SP を構築した場合でも，シングルサインオンによって利便性を損なわないように工夫した．

3.2 信頼できるユーザ情報管理（Reliability）

我々はこれまで ARCADE を Shibboleth 環境下で動作するソフトウェアとして実装し検証を行ってきた [15], [16]．Shibboleth を用いるうえで重要となるのは，ユーザ情報の管理をどこで誰が行うかということである．コンテンツを管理する各組織で新規にコンテンツ共有のためにユーザ情

報管理を行う場合，システム構築・運用において人的・金銭的コストが発生してしまう．

そこで我々は，ARCADE を “学術認証フェデレーション [17]”（以下，GakuNin という）上で動作させることで解決を図った．GakuNin は日本の大学など学術機関を対象としたフェデレーションである．GakuNin は 2010 年から本格運用に入り，2013 年 6 月現在で 65 の機関が IdP の運用を行っている [18]．GakuNin の参加機関では，各機関の管理者が実施要領 [19] と技術運用基準 [20] に従い IdP を運用している．そのため，GakuNin で運用中の IdP を利用することで，情報に統一的な信頼性が生まれるとともに，コンテンツ共有のために各組織で新規に IdP を構築する必要がなくなり，コストがかからないという利点も生まれる．

ARCADE における認証は GakuNin において各機関が用意している IdP で行う設計とすることで，コンテンツの利害関係から独立した信頼できる情報を利用できる．

3.3 多様なアクセスポリシ管理（Flexibility）

3.3.1 GakuNin mAP

認証後に ARCADE でコンテンツに対してアクセス制御を行う際には，アクセス対象とするユーザに関する情報が必要になる．GakuNin では認証を行った後に，各組織から GakuNin で定義されたユーザに関する情報（以下，属性という）を IdP から SP へ送信することが可能である．現在の GakuNin のポリシでは，eduPerson スキーマ [21] を軸とした 18 の属性が規定されている [22]．しかし，規定されている属性には，研究室や研究プロジェクト名称などの属性情報は存在しない．eduPersonPrincipalName（以下，ePPN という）や eduPersonTargetedID といった，GakuNin 内で個人を特定できる属性は存在するため，個人を特定可能な属性を設定していく方法もあるが，アクセス制御に用いるためには，メンバそれぞれに本人の属性値を直接聞く必要があり，手間がかかるうえ設定ミスの危険性も大きい．コンテンツを適切にアクセス制御できるように，ユーザ認証後に扱う属性として，研究プロジェクトや研究チームといった，いわば “仮想組織” の属性が必要になる．

そこで，我々は NII が GakuNin で提供している “GakuNin mAP [23]”（以下，mAP という）を利用することでこの問題の解決を図った．mAP は SWITCHtoolbox [24] のように，GakuNin 内で所属機関の異なるユーザをグループ化することが可能で，定義されたグループを属性として利用できる．このグループ属性を，GakuNin では isMemberOf 属性と定義している．つまり，仮想組織のメンバ間で同一の属性値（例：isMemberOf=”XXX”）を共有することになる．isMemberOf 属性は，プロジェクトの代表者が mAP の Web サイトへアクセスして定義する．代表者は，作成したグループに参加させたいメンバのメールアドレスに対して，招待状を送付する．メンバは招待状に記載された，メンバそれぞれ異なる URL をクリックし，所属する各機関の IdP で認証を行う．認証を行わせることで，メールアドレスに対する本人性が担保されることになる．この処理により，isMemberOf 属性と各メンバの ePPN とが結び付けられる．つまり，isMemberOf 属性を利用することで，ePPN 属性を束ねて 1 つの属性として扱うことができるようなる．

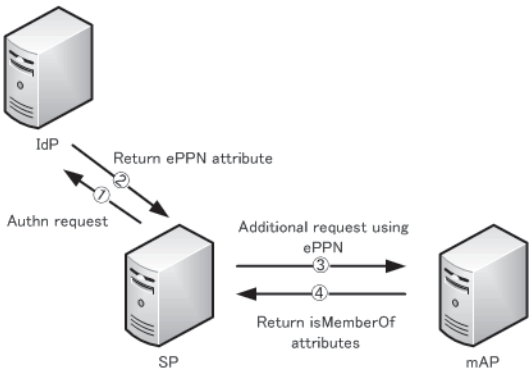

図 3 GakuNin mAP 動作概念図
Fig. 3 Conceptual image of GakuNin mAP.

```
<AttributeResolver type="SimpleAggregation" attributeId="eppn"
                   format="urn:oid:1.3.6.1.4.1.5923.1.1.1.6">
 <Entity> https://map.gakunin.nii.ac.jp/idp/shibboleth </Entity>
 <Attribute Name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1"
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
     FriendlyName="isMemberOf" />
</AttributeResolver>
```

図 4 SP における mAP の設定（一部）
Fig. 4 Setting of mAP in SP (partially).

図 3 に mAP の動作を示す．IdP で認証が完了したのち（①），IdP は SP に対して，ePPN 属性を返却する（②）．次に，SP は mAP に対して ePPN をキーにして，isMemberOf 属性を問い合わせる（③）．mAP は SP に対して当該 ePPN が持つ isMemberOf 属性のリストを返却する（④）．なお，SP の mAP 対応については，mAP マニュアル [25] の「mAP 連携のための情報」の項に示されている．特に，SP の設定ファイルの 1 つである shibboleth2.xml において図 4 の記載を行うことで，図 3 の ③ および ④ の動作である，ユーザの ePPN をキーとして mAP から isMemberOf 属性を取得できるようになる．また，送付される isMemberOf 属性は，“https://map.gakunin.nii.ac.jp/gr/A” のような URI 形式で表現される．

3.3.2 mAP を利用したアクセスポリシ設定

isMemberOf 属性を利用することで，研究室や組織を超えて結成された研究プロジェクトなどの様々な組織を属性として扱え，それを当該組織に所属するユーザに対応付けることができる．ただし 3.1 節で説明したとおり，ARCADE 環境下で扱うコンテンツの格納場所は SP を想定している．Shibboleth SP の標準的な設定は，GakuNin のサイト [17] にある「技術ガイド」を参考に構築できる．しかし，実際に

SP においてisMemberOf属性をどのようにコンテンツのアクセス制御に適用するかが課題になる．ただし，プロジェクトの規模や分野に依存することなく，SP を構築できるようにしたい．そこで，我々はSP のデータプラットフォームとして，Apache の mod_dav モジュールを用いたWebDAV [26] サーバを採用した．Apache はフリーウェアであり，広く利用されている．また，Apache と Shibboleth を組み合わせて利用した場合，Apache ではisMemberOf属性をサーバ環境変数として扱うことができる．そこで，Apacheの設定ファイルの 1 つである.htaccess を SP 内の各ディレクトリに配置し，isMemberOf属性に応じて WebDAV メソッドを制限することでコンテンツのアクセス制御を行えるように設計した．Apache を WebDAV サーバとして動作させるために，httpd.conf において mod_dav 関連モジュールをインクルードする．そして，ARCADE で扱うディレクトリに対して WebDAV の制御を有効にするとともに，AllowOverride を用いて.htaccess を有効にする．また，AllowOverride の設定を “Limit” と “AuthConfig” のみ指定し，必要最低限の権限設定だけが行えるようにして，不正な設定がされないように対応している．各ディレクトリのアクセス権限は.htaccess を利用する．

.htaccess は，WebDAV プロトコルをすべて Shibbolethで制御し，限られたユーザだけが.htaccess の更新権限を有するように設計した．例として，グループ A はコンテンツの参照およびアップロードが可能なグループ，グループ Bはコンテンツの参照のみが可能なグループとする．このような場合，グループ A に属しているユーザについては，グループ B に属さなくても閲覧可能ではあるが，我々はこの状況において，プロジェクト管理者を特別な存在として A と B の両方のグループに属させ，両方に属するユーザのみが，ドットで始まるファイルにもアクセスが可能な特別な権限を持つようにした．つまり，isMemberOf属性として A および B を持つユーザだけが.htaccess を更新できることになる．このことを.htaccess で表現すると**図 5** になる．特に FilesMatch により，ファイル名がドットで始まるか否かで処理を分けている．ドットで始まるファイルを更新できるのは，A，B 両方に所属しているユーザだけである．

このように，ARCADE におけるコンテンツのアクセス制御は，isMemberOf属性を用いて Apache の WebDAV メソッドを制御することによって組織を超えた様々な形態の組織間で柔軟に行えることとした．

3.4 特定のプロジェクトに依存しない拡張性（Scalability）

3.1，3.2 および 3.3 節で述べてきたように，Shibbolethの標準環境は IdP，DS，SP など複数のサーバによって構成されている．また，SP は isMemberOf属性によってアクセス制御され，プロジェクトによってはコンテンツの格納

```
AuthType shibboleth
ShibRequireSession On
ShibRequireAll On
<LimitExcept PROPFIND GET PUT DELETE MOVE MKCOL PROPPATCH>
Deny from all
</LimitExcept>
<Limit PROPFIND>
require isMemberOf https://map.gakunin.nii.ac.jp/gr/A https://map.gakunin.nii.ac.jp/gr/B
</Limit>
<FilesMatch "^¥.">
<Limit GET>
require isMemberOf https://map.gakunin.nii.ac.jp/gr/A https://map.gakunin.nii.ac.jp/gr/B
</Limit>
<Limit PUT DELETE MOVE MKCOL PROPPATCH>
require isMemberOf https://map.gakunin.nii.ac.jp/gr/A
require isMemberOf https://map.gakunin.nii.ac.jp/gr/B
</Limit>
</FilesMatch>
<FilesMatch "^[^¥.]">
<Limit GET>
require isMemberOf https://map.gakunin.nii.ac.jp/gr/A https://map.gakunin.nii.ac.jp/gr/B
</Limit>
<Limit PUT DELETE MOVE MKCOL PROPPATCH>
require isMemberOf https://map.gakunin.nii.ac.jp/gr/A
</Limit>
</FilesMatch>
```

図 5 .htaccess 設定例
Fig. 5 Example of .htaccess settings.

SP が分散していることもある．さらに，研究プロジェクトのメンバが認証技術やコンテンツ管理技術などIT スキルに優れているとは限らない．これらのことから，ARCADEはすべての環境を一元的に取り扱うことが可能で，かつ視覚的に分かりやすいインタフェースとすべきである．

そこで，我々は ARCADE を Java アプリケーションとして設計した．その際，Standard Widget Toolkit [27]（以下，SWT という）を用いた．SWT は Eclipse [28] 開発のために設計された GUI を作成するためのツールキットである．SWT の大きな特徴として，ボタンやテキストボックスなどのウィジェットを OS ネイティブのものを採用している点がある．そのため，Java 独自のウィジェットを使用するよりも動作が軽快でかつ，見た目が OS ライクなため，利用者のユーザビリティが向上するというメリットがある．また，ARCADE では Java Web Start [29] を採用した．Java Web Start を使用することにより，Web からJava アプリケーションをダウンロードして実行することが可能になる．その利点として以下のことがあげられる．

- アプリケーションを 1 回のクリックで起動
- つねに最新バージョンのアプリケーションを実行
- 複雑なインストールやアップグレード作業が不要

さらに，我々は分散された SP 群を ARCADE で一元的に取り扱えるように設計を行った．具体的には，分散されたSP 群を ARCADE 上においてツリー構造で利用できるように設計し，IT スキルに乏しいユーザであっても簡単に操作を行えることを目指した．そして，各 SP におけるアクセス制御ファイルにおいても，ユーザが直接そのような複雑な設定を記述することなく，ARCADE で視覚的に簡単に設定できるように配慮した．

3.5 ARCADE 用 SP としての必要条件

本節では，各組織が ARCADE 用の SP として運用するための必要条件について説明する．具体的には以下の条件を満たす必要がある．

- 組織が GakuNin に参加している．
- Shibboleth SP がインストールされている．
- Apache がインストールされており，Shibboleth との認証連携が行えるとともに，WebDAV および.htaccess が動作可能な状態になっている．
- mAP との連携設定がなされている．
- 自組織以外の IP アドレスからアクセスを受け付ける場合は，443 のポートでアクセスを受け付けるようにファイアウォールなどの設定を行う．
- GakuNin の運用フェデレーションに SP として申請を行い，登録されている．

ただし，SP を運用する組織は，運用フェデレーションに SP として申請を行う前に，他組織のユーザがコンテンツを配置することに対して問題がないか確認する必要がある．もしも組織のポリシによって，自組織内のユーザのみに利用させたい場合は，SP における WebDAV のルートディレクトリに配置されている.htaccess の require で，自組織のユーザだけが持つ属性（例：o : Kanazawa University など）を指定することで，他組織のユーザにアクセスさせることなく，自組織のユーザだけに利用させることも可能である．

また ARCADE では，共同研究を行う研究者が所属する全組織において，他組織が運用するサーバ上にコンテンツを配置することをセキュリティポリシで許可している場合は，他組織のサーバにコンテンツを配置できる設計にしている．GakuNin では，各組織の IdP がそれぞれで，GakuNin で提供されている SP のうち，どの SP の利用を許可するかの設定を行っているため，ユーザ自身は自組織のセキュリティポリシを意識することなく，ARCADE で提示された SP に対してコンテンツの配置を行えばよい．このように，ARCADE は高いユーザビリティを実現しつつ各集合体のポリシに応じて研究成果を共有できるように，他の組織が運用するサーバ上においてもコンテンツを配置できるように設計している．

4. ARCADE の動作

3 章では，ARCADE 構築に際しての設計思想について説明した．本章では，3 章で示した設計を基に，実装した ARCADE の動作について，認証動作と認証後のコンテンツアクセス制御動作についてそれぞれ説明する．

4.1 認証動作

本節では，ARCADE におけるユーザ認証までの動作を説明する．3.4 節で説明したとおり，ARCADE は GakuNin 環境のサーバ群を一元的に管理するインタフェースとしての役割を持つ．なお，ARCADE を配布する Web サーバは金沢大学の SP として GakuNin に登録している．認証までの動作として，まずユーザは Web ブラウザを経由して，ARCADE の Web サーバにアクセスし，「ARCADE を起動」ボタンをクリックする（**図 6**）．ARCADE は，Java Web Start で起動し，DS の画面を表示する（**図 7**）．ユーザはプルダウン形式で表示される一覧から所属機関の IdP を選択し，所属機関で配布された GakuNin 用の ID およびパスワードを入力し，認証を行う（**図 8**）．なお，ARCADE の Java プログラムはグローバルサイン社によるコードサイニング証明書で署名を行っており，金沢大学正規のソフトウェアであることを保証している [30]．

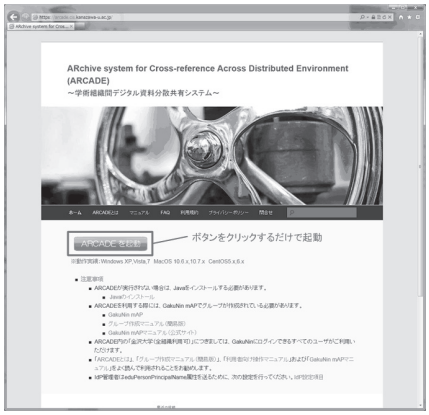

図 6 ARCADE 起動画面（Java Web Start）
Fig. 6 Snapshot of ARCADE start page (Java Web Start).

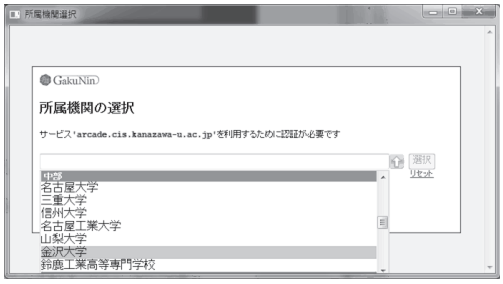

図 7 DS 画面
Fig. 7 Snapshot of DS page.

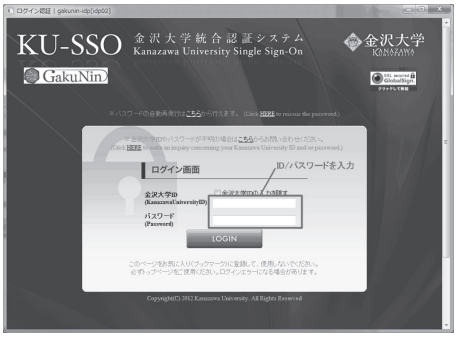

図 8 IdP 画面
Fig. 8 Snapshot of IdP page.

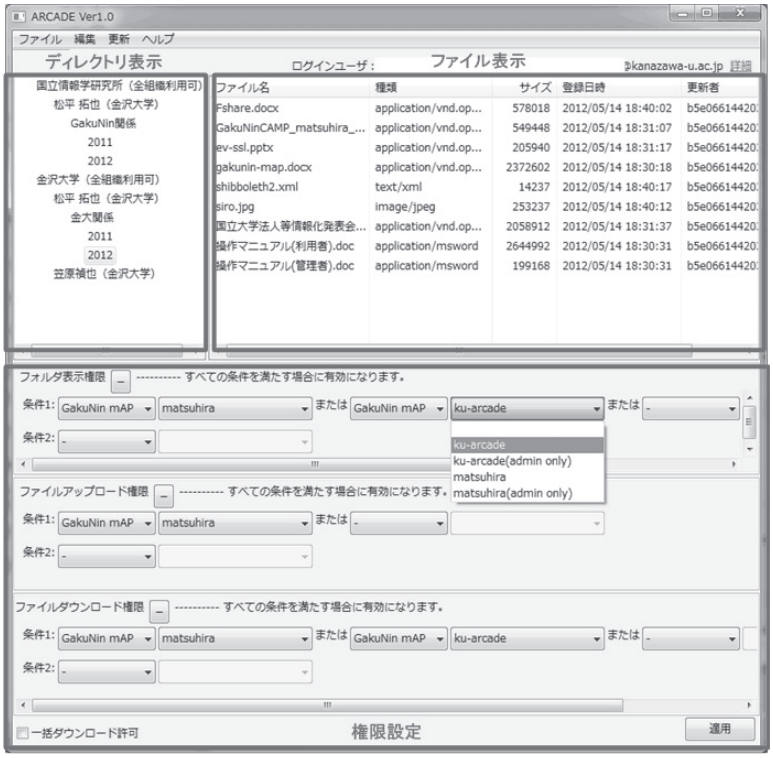

図 9 ARCADE メイン画面
Fig. 9 Snapshot of ARCADE main page.

4.2 アクセス制御動作

本節では，ARCADE におけるユーザ認証後のコンテンツのアクセス制御動作について説明する．4.1 節における認証動作が成功したユーザに対して，ARCADE は当該ユーザが利用可能なコンテンツ格納サーバ（SP）群の情報を，ARCADE のメイン画面に表示する（図 **9**）．SP 群の情報は ARCADE で一元的に管理している．ARCADE のメイン画面において，コンテンツのアクセス制御をディレクトリ単位で行うことができる．以降，ARCADE メイン画面における “ディレクトリ表示”，“ファイル情報表示”，“権限設定” の 3 つの部分について説明する．

①ディレクトリ表示

ディレクトリ表示部分には，ユーザがアクセス可能な SP 群のディレクトリツリーだけが表示される．ARCADE においては，SP 群の情報は XML で管理している．そこで各 SP の URL に対するラベルを設定し，ユーザにはラベル部分だけを提示する．そしてディレクトリ表示部分にアクセス可能なすべての SP を一度に表示することで，ユーザは OS 上で表示される異なるディスクドライブにアクセスするような感覚で簡単に SP 群を横断的に参照することができる．そして，ユーザはツリー上で右クリックし，ディレクトリの作成や削除などを行うことができる．

②ファイル表示

ユーザがディレクトリ表示部分のディレクトリを選択すると，ディレクトリ内のコンテンツ情報の一覧が表示される．コンテンツの情報として，「ファイル名」，「種類」，「サイズ」，「登録日時」および「登録者」を持つ．そして，ユーザはドラッグアンドドロップでコンテンツを操作することができる．

③権限設定

コンテンツのアクセス制御は 3.3 節で説明したとおり，Apache の.htaccess ファイルで isMemberOf 属性に応じて WebDAV メソッドを制限することで実現する．ARCADE では各 SP において，ディレクトリ単位で「ディレクトリ参照権限」，「ファイルアップロード権限」，「ファイルダウンロード権限」の 3 種類の権限を設定することができる．なお，isMemberOf 属性は mAP の Web サイトで事前に登録しておくものとする．ディレクトリ参照権限は，isMemberOf 属性で PROPFIND メソッドを制御し，ディレクトリ表示部分にディレクトリを表示させるかどうかを設定する．ファイルアップロード権限は isMemberOf 属性で PUT，DELETE，MOVE，MKCOL および PROPPATCH メソッドを制御し，ファイル表示部分におけるコンテンツの編集の可否やディレクトリ表示部分における当該ディレクトリ以下のディレクトリ編集の可否を設定する．ファイルダウンロード権限は，isMemberOf 属性で GET メソッドを制御し，当該ディレクトリ内のコンテンツのダウンロード可否を設定する．ただし，.htaccess は記述が複雑なうえ，isMemberOf 属性は “https://map.gakunin.nii.ac.jp/gr/group-A” のような URI 形式であり，ユーザが直接編集するのは困難である．そこで，ARCADE では図 9 のように各ディレクトリにおいて，プルダウンでユーザに結び付けられている

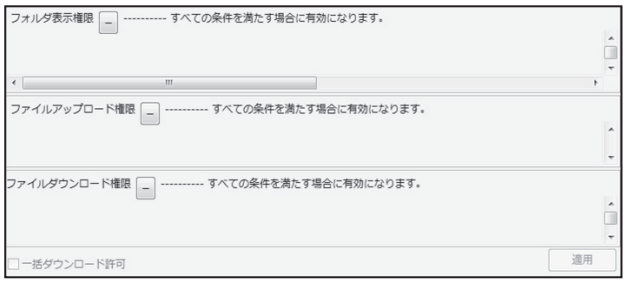

図 10 ARCADE 権限設定画面（変更権限なしの場合）
Fig. 10 Snapshot of access control page (If the user does not have permission to change).

isMemberOf 属性をすべて表示し，適当な属性を選択した後で “適用” ボタンをクリックすることで.htaccess ファイルが自動作成されるようにした．なお，権限設定を変更可能なユーザ以外には**図 10** のようにプルダウンを表示させないようにすることで権限設定の誤操作を防いでいる．さらに，ARCADE ではドットで始まるファイルをアップロード/ダウンロードできない仕様としており，ARCADE の利用により.htaccess がコントロールを失うことはない．ただし，ユーザが ARCADE 以外のクライアントから SP にアクセスする可能性も想定されるが，図 5 の.htaccess の設定にあるように，.htaccess を更新できるのはプロジェクト管理者だけである．さらに，ARCADE 以外のクライアントを利用した場合でも，.htaccess を更新できるユーザは ARCADE を利用した場合と同じである．したがって，.htaccess については，プロジェクト管理者以外は改変することができないため，プロジェクト管理者が ARCADE 以外のクライアントを用いてアクセスする必要がある場合のみ，不注意で意図しない内容に書き換えてしまわないように配慮すればよい．その際に，ユーザが誤って手元にあった ARCADE とは関係ないサーバの.htaccess をアップロードしてしまうなどの意図しない更新の事故を防ぐために，Apache の設定ファイル内の AccessFileName でファイル名を.htaccess から.arcade-access-policy に変更することで対応している．

5. ARCADE 動作検証

3 章では ARCADE の設計について，4 章では 3 章の設計思想に基づいて実装した ARCADE の動作について説明した．本章では ARCADE の検証運用について述べる．

5.1 検証運用条件

我々は ARCADE の動作を適切に評価するため，評価期間限定で，図 1 の環境を実際の GakuNin に構築した．検証運用は以下の条件を用いた（**表 2** は条件の要約）．

- Kanazawa University，Sample University，Test University の 3 つの組織が GakuNin に参加している．
- Kanazawa University のユーザ “matsuhira” は matsuhira だけが所属するグループ “matsuhira-individual” を “あけぼの衛星データ” ディレクトリにセットしている．
- matsuhira は新規プロジェクト “project-x” を立ち上げる．
- matsuhira は mAP でグループ “project-x” を作成し，Sample University のユーザ “sample01” に対して招待状を送付して参加してもらう．
- matsuhira はあけぼの衛星のデータのうち，1989，1990 および 2000 年のデータのみを sample01 に対してダウンロードのみ許可する．つまり，matsuhira は project-x の isMemberOf 属性を持つユーザに対して，これらのディレクトリにあるコンテンツのダウンロードを許可することになる．
- Test University の “test01” は project-x には参加していない．つまり，test01 に対しては，ディレクトリ自体が表示されないことになる．

表 2 検証運用条件
Table 2 Conditions for the evaluation experiment.

ユーザ	matsuhira	sample01	test01
組織	Kanazawa University	Sample University	Test University
isMemberOf 属性	matsuhira-individual project-x	project-x	(None)

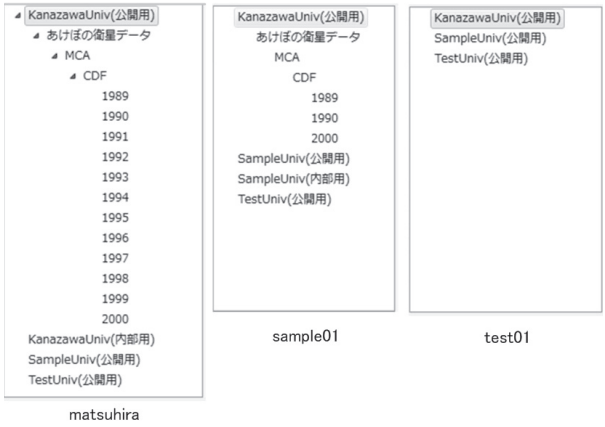

図 11 ユーザのアクセス制限に応じたディレクトリ参照状態
Fig. 11 Directory image according to user access control policies.

5.2 検証運用結果

5.1 節の条件を適用した場合の，各ユーザのディレクトリ参照状態を**図 11** に示す．

- matsuhira は，1989〜2000 年すべてのディレクトリを参照でき，コンテンツのアップロードおよびダウンロードも可能である．さらに，全ディレクトリの権限設定が行える状態になっている．
- sample01 は 1989，1990，2000 年のディレクトリのみ

参照可能で，かつコンテンツのダウンロードのみ可能な状態になっている．当該ディレクトリにおいてはコンテンツのアップロードおよび権限の設定を行うことはできない．さらに，上記以外のディレクトリは表示されない．

- test01 は，あけぼの衛星データディレクトリの参照自体が不可能になっている．

この検証運用により，ARCADE 動作を検証できた．

6. ARCADE 性能評価

本章では，ARCADE の実用性の評価について述べる．評価項目として，ARCADE でコンテンツを取り扱う際に Shibboleth 認証がどれだけオーバヘッドとなるかということがある．図 1 に示すとおり，ARCADE では，起動時にすべての SP に対して Shibboleth の認証処理を行う．Shibboleth では，認証が確立した後は Cookie を用いてセッション管理を行う．この Cookie を検証し，コンテンツを取り扱うための認可判断を行う処理は，コンテンツをアップロードまたはダウンロードするごとに，最初に 1 回必要となる．Shibboleth 認証がない場合は，上記の処理は不要であり，この差が Shibboleth 認証の有無によるオーバヘッドとなる．そこで我々は，Shibboleth 認証の有無によるデータのレスポンスタイムを計測した．計測方法としてまず，1 台の SP（WebDAV）サーバにおいて，Shibboleth 認証が必要なディレクトリ「auth」と不必要なディレクトリ「unauth」をそれぞれ用意する．そして，それぞれのディレクトリに対してコンテンツ（1 MB，10 MB，100 MB）のアップロードとダウンロードを行い処理にかかった時間を計測する．つまり，ARCADE 上に auth ディレクトリと unauth ディレクトリを表示させ，クライアント PC から ARCADE を経由して SP にコンテンツをアップロードおよびダウンロードし，それぞれの作業開始から終了までの時間を計測した．計測時間のうち，認証にかかる時間の割合を大きくするために，クライアント PC と SP の間はホップ数が 2 の距離に配置した．また，SP とクライアント PC は 1 Gbps ネットワークで接続した．SP およびクライアント PC の諸元を**表 3** に示す．

計測結果をグラフ化したものを**図 12** に示す．write はコンテンツのアップロード，read はコンテンツのダウンロードを表す．この実験では，各処理についてそれぞれについて 5 回計測を行った．Shibboleth 認証の有無における有意差を見るために，Wilcoxon's rank sum test で検定を行った（$n = 5$，$^*P < 0.05$）．Shibboleth 認証の有無による有意差はあったが，有意差がでている 1 MB のアップロードで認証にかかる時間が 24%，10 MB のアップロードで 11%程度であった．実際には，クライアント PC と SP はもっと距離があり，ネットワークの遅延増加により転送時間が増加することを考慮すると，処理時間としては増えているが，コンテンツを扱う処理に影響を与えない程度に小さい．この結果から，Shibboleth 認証がレスポンスタイムに与える影響は実用面で問題ない範囲であると判断できる．

表 3 SP およびクライアント PC 諸元
Table 3 Specifications of SP and client PC.

種別	諸元
SP	Hypervisor：VMware ESXi 5.0.0 OS：CentOS 6.3 (64bit) CPU：Intel Xeon 05160 3.00GHz Memory：2GB HDD：ReadyNAS 2100 RNRX4410（同一セグメント内で iSCSI による 1Gbps 接続）
クライアント PC	OS：Windows7 Ultimate (64bit) CPU：Intel i7-2640M 2.8GHz Memory：8GB

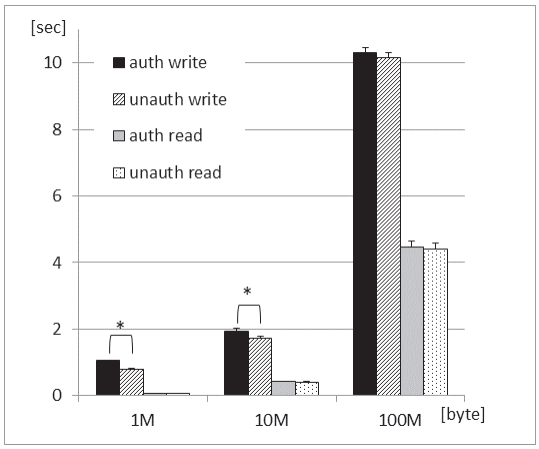

図 12 Shibboleth 認証の有無によるデータ転送速度
Fig. 12 Data transfer velocity by having Shibboleth authentication or not.

7. まとめ

今回，ARCADE を開発したことにより，実験や観測などで得られた様々なコンテンツを，組織を超えて簡単かつ安全に共有することを可能にした．ARCADE を Shibboleth 環境で動作するように設計し，かつ GakuNin 上で動作させることで，GakuNin の IdP を ARCADE の認証に利用でき，認証にかかるコストをかけずに信頼できるユーザ認証を実現した．さらに，mAP の isMemberOf 属性を利用することで，異なる組織の研究者をグループ化して扱うことを可能にし，さらにこの属性を用いたアクセス制御を実現した．また，ARCADE を Java アプリケーションとして GUI ベースで開発したことにより，ユーザの IT スキルに依存することなく，簡単に各種コンテンツのアクセス制御を行えるようにした．そして，コンテンツ格納サーバとして，Apache の WebDAV サーバを SP 化して利用することで，コンテンツを中央で管理する必要なく，コンテン

ツを自組織に保持したまま，ARCADE を介して，利用可能なコンテンツを横断的に参照でき，コンテンツのアクセス制御を行えるようにした．また，ARCADE の検証実験を行い，ARCADE が正しく動作していることを検証できた．さらに，ARCADE の性能評価も行い，実運用に耐えうることを確認できた．ARCADE は，2012/3/17 より，GakuNin の運用フェデレーションでサービスを開始しており [31], [32]，GakuNin の運用フェデレーションに IdP が構築済みの機関に所属している研究者は ARCADE を利用することが可能である．

今後の展望として，ARCADE をできるだけ多くの研究者に様々な用途で利用してもらい，可用性の評価と改善点の洗い出しを行い，その評価・要望を吸い上げるとともに，システム仕様へのフィードバックを検討している．そして，GakuNin に未参加の機関に所属する研究者も ARCADE を利用できるように，ARCADE を NII が提供する OpenIdP [33] に対応させる予定である．OpenIdP は GakuNin に参加している一部の SP を利用できる登録制の IdP である．登録制ではあるが，各プロジェクトの責任者が mAP においてメンバの登録を適切に取り扱うため，機関に IdP が立ってないユーザも収容できる．また，現状では，グループの作成や編集などの作業は mAP の Web サイトへアクセスして行う必要がある．そのため，ARCADE 上で直接作業を行えるように mAP の API 化を検討している．そして，iPhone や iPad に代表される小型端末においては Java をサポートしていない．そのため，Java 以外の実装も考慮する必要があると考えており，その 1 つとして，HTML5，CSS3，JavaScript を組み合わせた実装を検討している．また，ARCADE 上で流通する各コンテンツの検索機能を強化するために，非文献リポジトリとの相互連携を行うことを考えている．さらに ARCADE へアップロードされたコンテンツ情報が非文献リポジトリで検索できるインタフェースを開発し，コンテンツ利活用の利便性を高めることも検討している．

最後に，研究プロジェクトは国をまたぐことも多いと想定されるため，将来的には，国を超えた利用へとつなげていきたいと考えている．GakuNin のように，たとえばアメリカでは InCommon [34]，スイスでは SWITCHaai [35] という名称で，多くの国々でフェデレーションが構築されている．そして，最近ではこれらのフェデレーションをまたいで認証連携を行う動きが活発になってきている．現時点では eduGAIN [36]，Kalmar2 [37] などを用いたフェデレーション間での認証連携が主流になりつつある．eduGAIN などは，フェデレーションを超えて直接 IdP と SP が情報をやりとりするので，分散認証のモデルを破壊しない．GakuNin も将来的には世界に分散する海外のフェデレーションと相互連携を行うことが予想される．フェデレーションが相互接続されれば，接続されたフェデレーションに参加している機関の研究者たちとも ARCADE を用いてデータの共有を行うことができると考えている．なお，ARCADE の英語化は完了しており，海外の研究者も問題なく ARCADE を利用できる．さらに，海外で構築されているフェデレーションにおいても Shibboleth をベースとしているところが多いため，本稿で我々が提案する機構を容易に導入することが可能であると考えている．

謝辞 本研究は科研費（若手研究 B）の助成を受けたものである（22700809，25750080）．

参考文献

[1] 学術機関リポジトリ構築連携支援事業，入手先 ⟨http://www.nii.ac.jp/irp/⟩（参照 2013-08）．
[2] 国立情報学研究所学術機関リポジトリ構築連携支援事業 “メタデータフォーマット junii2”，入手先 ⟨http://www.nii.ac.jp/irp/archive/system/junii2.html⟩ (accessed 2013-08).
[3] 高田良宏，笠原禎也，西澤滋人，森 雅秀，内島秀樹：非文献コンテンツのための可視性と保守性に優れた学術情報リポジトリの構築，情報知識学会誌，Vol.19, No.3, pp.251–263 (2009).
[4] Rice, R. and Haywood, J.: Research Data Management Initiatives at University of Edinburgh, *The International Journal of Digital Curation*, Vol.6, No.2, pp.232–244 (2011).
[5] Sasaki, T., Nakae, M. and Ogawa, R.: Content oriented virtual domains for secure information sharing across organizations, *Proc. 2010 ACM Workshop on Cloud Computing Security Workshop*, DOI: 10.1145/1866835.1866838 (2010.10).
[6] SAML2.0, available from ⟨http://www.oasis-open.org/standards#samlv2.0⟩ (accessed 2013-08).
[7] Shibboleth, available from ⟨http://shibboleth.net/⟩ (accessed 2013-08).
[8] simpleSAMLphp, available from ⟨http://simplesamlphp.org/⟩ (accessed 2013-08).
[9] Hatala, M. Eap, T.M. and Shah, A.: Unlocking repositories: Federated security solution for attribute and policy based access to repositories via Web services, *The 1st International Conference on Availability, Reliability and Security, 2006, ARES 2006*, DOI: 10.1109/ARES.2006.140 (2006.4).
[10] Rieger, S., Richter, H. and Xiang, Y.: Introducing Federated WebDAV Access to Cloud Storage Providers, *CLOUD COMPUTING 2011, The 2nd International Conference on Cloud Computing, GRIDs, and Virtualization*, ISBN:978-1-61208-153-3, pp.46–51 (2011.9).
[11] Dropbox, available from ⟨https://www.dropbox.com/⟩ (accessed 2013-08).
[12] Google Drive, available from ⟨http://www.google.com/drive/about.html⟩ (accessed 2013-08).
[13] 原田要之助：クラウドコンピューティングのリスクとガバナンスに関する調査・研究について，情報処理，Vol.51, No.12, pp.1591–1601 (2010).
[14] 高等教育機関の情報セキュリティ対策のためのサンプル規程集，入手先 ⟨http://www.nii.ac.jp/csi/sp/doc/sp-sample-2010-2.pdf⟩（参照 2013-08）．
[15] Matsuhira, T., Kasahara, Y. and Takata, Y.: Development of a file-sharing system for educational collaboration among higher-education institutions, *International Journal of Education and Information Technologies*,

Vol.5, Issue 2, pp.149–156 (2011).
[16] Matsuhira, T., Kasahara, Y. and Takata, Y.: ARchive System for Cross-Reference Across Distributed Environment (ARCADE) Applicable to Sharing of Educational Materials among Inter-University Consortium, *9th WSEAS International Conference on Education and Educational Technology* (*EDU '10*), pp.167–170 (2010).
[17] 学術認証フェデレーション（GakuNin），入手先 ⟨http://www.gakunin.jp/⟩（参照 2013-08）.
[18] GakuNin IdP リスト，入手先 ⟨http://www.gakunin.jp/participants/⟩（参照 2013-10）.
[19] 学術認証フェデレーション実施要領（Ver.1.0），入手先 ⟨http://id.nii.ac.jp/1149/00000210/⟩（参照 2013-10）.
[20] 学術認証フェデレーション技術運用基準（Ver.2.0），入手先 ⟨http://id.nii.ac.jp/1149/00000212/⟩（参照 2013-10）.
[21] eduPerson schema, available from ⟨http://middleware.internet2.edu/eduperson/⟩ (accessed 2013-08).
[22] GakuNin 属性リスト，入手先 ⟨https://meatwiki.nii.ac.jp/confluence/pages/viewpage.action?pageId=12158166⟩（参照 2013-10）.
[23] GakuNin mAP, available from ⟨https://map.gakunin.nii.ac.jp/map/⟩ (accessed 2013-08).
[24] SWITCHtoolbox, available from ⟨http://www.switch.ch/de/toolbox/about/index.html⟩ (accessed 2013-08).
[25] GakuNin mAP マニュアル，入手先 ⟨https://meatwiki.nii.ac.jp/confluence/display/gakuninmappublic/Home⟩ (accessed 2013-08).
[26] WebDAV, available from ⟨http://www.webdav.org/⟩ (accessed 2013-08).
[27] Standard Widget Toolkit (SWT), available from ⟨http://eclipse.org/swt/⟩ (accessed 2013-08).
[28] Eclipse, available from ⟨http://www.eclipse.org/⟩ (accessed 2013-08).
[29] Java Web Start, available from ⟨http://www.oracle.com/technetwork/java/javase/javawebstart/index.html⟩ (accessed 2013-08).
[30] コードサイニング証明書，入手先 ⟨https://jp.globalsign.com/service/codesign/objectsigning/⟩（参照 2013-08）.
[31] ARCADE, available from ⟨https://arcade.cis.kanazawa-u.ac.jp/⟩ (accessed 2013-08).
[32] GakuNin SP リスト，入手先 ⟨http://www.gakunin.jp/participants/⟩（参照 2013-10）.
[33] OpenIdP, available from ⟨https://openidp.nii.ac.jp/⟩ (accessed 2013-08).
[34] InCommon, available from ⟨http://www.incommonfederation.org/⟩ (accessed 2013-08).
[35] SWITCH, available from ⟨http://www.switch.ch/⟩ (accessed 2013-08).
[36] eduGAIN, available from ⟨http://www.geant.net/service/edugain/Pages/home.aspx⟩ (accessed 2013-08).
[37] Kalmar2, available from ⟨https://www.kalmar2.org/kalmar2web/front_page.html⟩ (accessed 2013-08).

松平 拓也 （正会員）

2004 年信州大学工学部情報工学科卒業．2006 年信州大学大学院工学系研究科博士前期課程情報工学専攻修了．2011 年金沢大学大学院自然科学研究科博士後期課程電子情報科学専攻修了．博士（工学）．2004 年 4 月より金沢大学総合メディア基盤センター技術職員．認証基盤の構築および組織間認証連携に関する研究開発に従事．電子情報通信学会会員．

中村 素典 （正会員）

1994 年京都大学大学院工学研究科博士後期課程単位取得退学．立命館大学理工学部助手，京都大学経済学部助教授，京都大学学術情報メディアセンター助教授を経て，2007 年より国立情報学研究所特任教授，現在に至る．博士（工学）．コンピュータネットワーク，ネットワークコミュニケーション，認証連携等の研究に従事．IEEE，電子情報通信学会，日本ソフトウェア科学会各会員．

山地 一禎 （正会員）

2000 年豊橋技術科学大学大学院博士課程修了．同年日本学術振興会特別研究員．2002 年より理化学研究所脳科学総合研究センター研究員．2007 年より国立情報学研究所准教授，現在に至る．データシェアリングならびにその認証基盤に関する研究開発に従事．電子情報通信学会，情報知識学会各会員．

西村 健

1998 年東京大学大学院理学系研究科情報科学専攻修士課程修了．2001 年東京大学大学院理学系研究科情報科学専攻博士課程単位取得退学．同年東京大学人文社会系研究科助手，同情報基盤センター特任助教を経て，2009 年より国立情報学研究所特任研究員．認証基盤の構築，認証技術および認証連携技術の研究開発を行う．

高田 良宏 （正会員）

2010 年金沢大学大学院自然科学研究科博士後期課程電子情報科学専攻修了．博士（工学）．現在，金沢大学総合メディア基盤センター准教授，高度情報処理による大規模データベースの参照・配信技術に関する研究，非文献資料公開のための共通プラットフォームの研究に従事．情報知識学会，電子情報通信学会，コンピュータ利用教育学会各会員．

笠原 禎也 （正会員）

1989 年京都大学工学部電気工学第二学科卒業．1992 年京都大学大学院博士後期課程を中退し，京都大学工学部助手着任．同大学院情報学研究科助手，金沢大学工学部助教授，同総合メディア基盤センター准教授を経て，現在，同センター教授．科学データからの知的信号処理，科学衛星搭載ソフトウェア受信器の開発と宇宙空間のプラズマ波動特性の研究に従事．博士（工学）．電子情報通信学会，地球電磁気・地球惑星圏学会，米国地球物理学会連合各会員．