

情報処理学会研究発表会のための 発表資料アーカイブシステムの設計と実装

横木健太^{†1,a)} 山井成良^{†2,b)} 北川直哉^{†2,c)}

概要: 情報処理学会インターネットと運用技術 (IOT) 研究会では、全研究会を対象に、研究発表会で使用されたプレゼンテーション資料 (PowerPoint ファイル等) を各研究会登録会員が閲覧可能なアーカイブシステムの導入を検討している。本論文では、発表資料アーカイブシステムの導入に向けた設計と実装について報告する。本システムは、閲覧用アカウントや投稿用アカウントの追加を各研究会のメーリングリストで受信するメールによって自動的に行うことや、アップロードされた資料をクラウドストレージ上に保存することにより、管理コストを最小限に抑制することができる。さらに、アップロードされたファイルの利用情報等の統計情報を取得する機能や、閲覧権限を持たない第三者への資料漏洩のリスクを最小限にとどめるためのユニークな共有リンクの作成機能も導入した。

Design and Implementation of Presentation Material Archive System for IPSJ SIG Meetings

KENTA YOKOGI^{†1,a)} NARIYOSHI YAMAI^{†2,b)}
NAOYA KITAGAWA^{†2,c)}

Abstract: The IPSJ SIG-IOT is considering the introduction of a presentation material archive system for all SIGs that acts to allow viewing presentation materials for registered members of each SIG. In this paper, we describe a design and implementation for the introduction of a presentation material archive system. The system can be minimized management costs by automatic addition of account for access and upload using received e-mails by mailing list of each SIG, and by preserves uploaded materials to a cloud storage. Furthermore, we implemented a function to retrieve statistical information such as usage information about uploaded files, and a function to generate unique share links to minimize risks of material leakage to the third party who does not have access authority.

1. はじめに

平成 27 年 9 月に開催されたインターネットと運用技術 (IOT) 研究会運営委員会において、従来の電子図書館^{*1}において公開されている研究報告のアーカイブに加えて、発表の際に使用されたプレゼンテーション資料 (以下、発表資料) についても同様にアーカイブ化することが提案された。しかし、そのようなアーカイブシステムを運用する場合、発表者からの発表資料の回収やアーカイブシステムへのアップロード、各発表と発表資料との関連付け、サーバの保守など、様々な手間がかかる。

そこで、本稿では、閲覧用アカウントや投稿用アカウントの追加を各研究会のメーリングリストで受信するメールによって自動的に行うことや、アップロードされた資料をクラウドストレージ上に保存し、システムが保持する必要

のあるデータを最小限にとどめることなどによって、これらの運用上の手間を可能な限り省いた発表資料アーカイブシステムの設計と実装について述べる。

2. 要求要件

アーカイブシステムを設計する上で、管理の省力化のため、次に示すような要件が求められた。

- アップロードされた資料に対して、電子図書館の研究報告と関連付けて保存できること
- 大量ファイルの維持コスト軽減のために、自組織で運用するアーカイブシステム上に発表資料を保持しないこと
- 発表資料は、研究会登録会員のみが参照可能であること
- 発表者は自身の発表資料のアップロードが可能であること
- 管理者によって、ユーザや資料などの追加や編集、削除が行えること
- 発表資料へのアクセス状況を確認できること

†1 東京農工大学工学部情報工学科
Department of Computer and Information Sciences, Faculty of Engineering,
Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-8588,
Japan

†2 東京農工大学大学院工学研究院先端情報科学部門
Division of Advanced Information Technology and Computer Science,
Institute of Engineering, Tokyo University of Agriculture and Technology,
Koganei, Tokyo 184-8588, Japan

a) s148409u@st.go.tuat.ac.jp

b) nyamai@cc.tuat.ac.jp

c) nakit@cc.tuat.ac.jp

*1 情報学広場: 情報処理学会電子図書館 <https://ipsj.ixsq.nii.ac.jp/ej/>

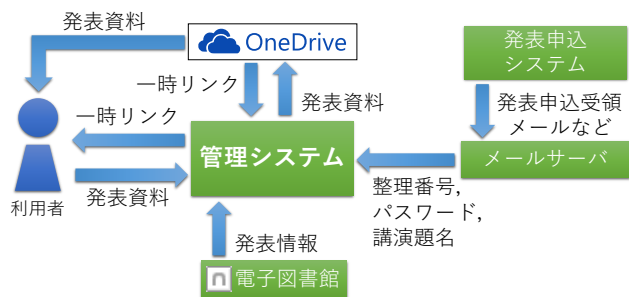


図 1 アrchiveシステムの構成

Figure 1 The Structure of This Archive System.

3. 設計

2 節で述べた要件を満たすアーカイブシステムとして、図 1 に示すようなシステム (以下、本システム) を設計した。

本システムの特徴として、運用上必要となる様々な処理をほぼ全自動で行うことによって、運用上の手間を最小限にとどめていることが挙げられる。具体的には、発表情報の取得には、研究発表会当日用サイトおよび電子図書館を使用する。また、発表資料の保存には OneDrive^{*2}を使用する。さらに、各種認証情報の取得および更新には、発表申込システムをはじめとする情報処理学会からのメールを使用する。このように、運用上必要となる情報の取得にそれぞれ外部のサービスを利用することで、自組織で運用するシステムが保持する必要がある情報を最小限にとどめられるように設計した。

なお、本システムは Web アプリケーションフレームワークとして ASP.NET MVC を、RDBMS として SQL Server を、Web サーバとして IIS をそれぞれ使用した。

4. 実装

本システムの動作およびその実装について次に示す。

4.1 ロール

本システムでは、ユーザに対して閲覧者、発表者、管理者の 3 種類のロールのいずれかを付与する。各ロールの権限と、それらに対する本システムの動作を以下の(1)から(3)に示す。

(1) 閲覧者 (アクセス権限)

閲覧者は、本システム上にアーカイブされている発表資料の閲覧権限を持つロールである。閲覧者は、本システム上にアーカイブされている発表資料へのアクセスおよび統計情報の確認のみが可能である。

*2 Microsoft 社の運営するクラウドストレージサービス

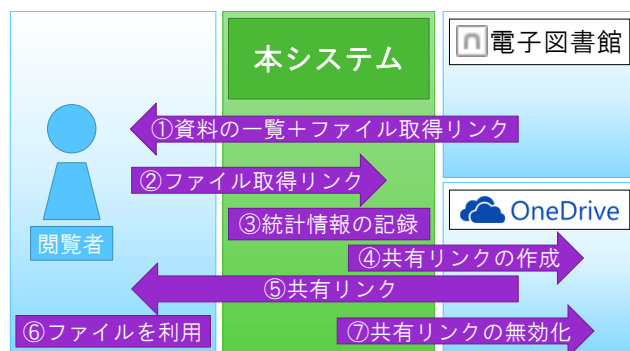


図 2 アクセスロールを持つユーザに対する本システムの振舞い

Figure 2 Behavior of the System for Users With Access Roll.

アクセスロールを持つユーザに対する本システムの振舞いを図 2 に示す。

- ① 本システムは電子図書館から資料の一覧を取得する。取得した資料の一覧の中で、本システム内に登録されている資料が存在する場合は、その資料に対するファイル取得リンクとともに、資料の一覧をユーザに提供する。
- ② ユーザは、ファイル取得リンクにアクセスする。
- ③ 本システムは、統計情報を記録する。
- ④ 本システムは、OneDrive に読み取り専用の共有リンクの作成を依頼する。
- ⑤ OneDrive にて作成された共有リンクは本システムを経由してユーザに提供される。
- ⑥ ユーザは、その共有リンクを用いてファイルをプレビュー表示したり、ダウンロードしたりすることができる。
- ⑦ 5 分経過後、本システムは作成した共有リンクの無効化を OneDrive に指示する。

(2) 発表者 (アップロード権限)

発表者は、本システムに対して発表資料の追加権限を持つロールである。本ロールを持つユーザは、本システム上に自身の発表に対する発表資料をアップロードすることが可能である。

アップロードロールを持つユーザに対する本システムの振舞いを図 3 に示す。

- ① 本システムはメールサーバより発表情報を取得し、対応するアカウントを作成する。
- ② 作成したアカウントでログインしたユーザがファイルをアップロードすると、そのファイルは本システムを経由して OneDrive 上にアップロードされる。
- ③ OneDrive からアップロードしたファイルのファイル ID を取得する。

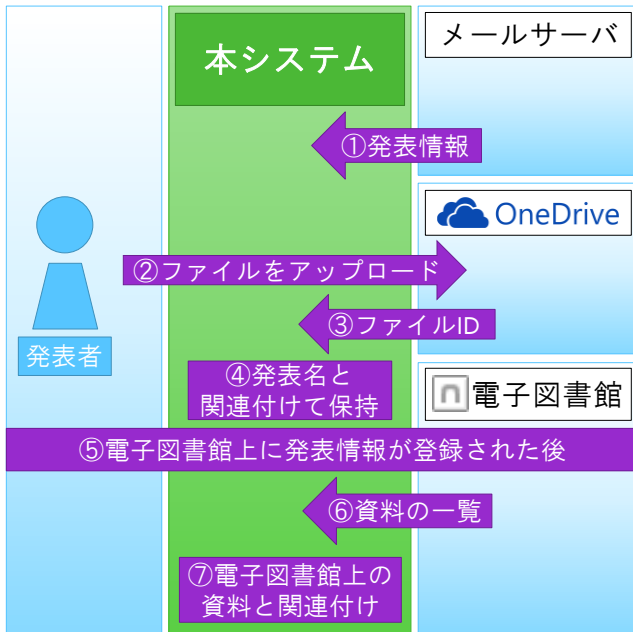


図 3 アップロードロールを持つユーザに対する本システムの振舞い

Figure 3 Behavior of the System for Users With Upload Roll.

- ④ 取得したファイル ID を発表名と関連付けて保持する。
- ⑤ その後、電子図書館上に発表情報が登録される。
- ⑥ 本システムが電子図書館から資料の一覧を取得する。
- ⑦ もし、⑥の際に保持している発表名と同一の資料が存在した場合は、その資料と保持しているファイル ID を関連付けする。

(3) 管理者（管理権限）

管理者は、本システムに対してあらゆる操作を行う権限を持つロールである。本ロールを持つユーザは、任意の資料に対する発表資料や本システム上のユーザ、資料の保存に使用する OneDrive にアクセスするための Microsoft アカウント[1]に対して、追加や削除、変更などのすべての操作を行うことが可能である。

4.2 メールを用いたアカウント生成

本システムは、IMAP 対応の任意のメールサーバからメールを取得することによって、そのメール中に記載されている情報を用いて新しいアカウントを自動的に作成することが可能である。対応するメールは、次に示すとおりである。

- 研究報告のダウンロードについて
(研究発表会当日用サイトの年間 ID およびパスワードを通知するメール)
- 発表申込完了のお知らせ



山井 成良 様

いつもお世話になっております。
情報処理学会 調査研究担当です。

情報処理学会 第 xx 回インターネットと運用技術研究発表会の発表申込を受領いたしました。

貴発表申込の整理番号及びパスワードは以下のとおりです。

1. 整理番号 : 012345
2. パスワード : hogehoge
3. 講演題名 : 発表申込受領メールサンプル

図 4 発表申込完了メールの例

Figure 4 An Example of Submission Completion Mail.

本システムは、前者に対してはアクセスロールを付与したアカウントを、後者に対してはアップロードロールを付与したアカウントをそれぞれ作成する。

メールによるアカウントの自動生成は、まず IMAP を用いて指定されたメールサーバに接続し、未読のメールをすべて取得する。この際、メールは既読状態となる。そして、取得したメールのうち、件名に特定の文字列を含むものに対して、本文中から ID およびパスワード、講演題名を抽出し、メールに応じたロールを付与したアカウントを作成する。具体的に、図 4 に示す発表申込完了メールを例として説明する。図 4 のメールは、件名に「発表申込完了のお知らせ」という文字列を含むため、発表者（アップロード権限）のロールを持つユーザを作成するために使用される。この例では、ID は整理番号の「012345」、パスワードは「hogehoge」、講演題名は「発表申込受領メールサンプル」となる。現時点ではメールの From アドレスなどのチェックを行っておらず、不正なメールによって勝手にアカウントが作成されてしまうセキュリティホールの原因になりうるため、本システムが使用しているメールアドレスが外部に漏れないように適切に管理する必要がある。

4.3 発表情報の取得

本システムは、発表情報を情報処理学会の電子図書館および研究発表会当日用サイトから取得する。発表情報を情報処理学会の電子図書館からだけでなく、研究発表会当日用サイトからも取得しているのは、発表に関する情報がまだ電子図書館に掲載されていない状態でもそれらの発表資料を閲覧できるようにするためである。また、当日用サイ



図 5 スクレイピングに用いる電子図書館の Web サイトにおける発表情報の例

Figure 5 An Example of the Digital Library Web Site's Presentation Information for Scraping.

ト以外の発表情報は常に電子図書館から取得しているので、仮に電子図書館から資料が削除された場合には、本システム上にも該当する資料は表示されなくなる。

電子図書館からの発表情報の取得には、電子図書館の Web サイトをスクレイピングすることによって実現した。また、一度に多数のアクセスがあった場合を想定し、システムはスクレイピングした情報を一定時間キャッシュして利用する。図 5 および図 6 に、発表情報をスクレイピングする元となる電子図書館の Web サイトと、それに対して本システムが出力する資料の一覧を示す。

4.4 OneDrive

本システムは、ユーザによってアップロードされた資料を本システム内では保持せず、OneDrive 上にアップロードする。これにより、本システムのストレージの拡張やバックアップなどの手間を最小限にとどめることが可能である。また、単一の OneDrive アカウントではアップロードできる容量に限りがあるため、複数の OneDrive (Microsoft) アカウントを登録し、適宜使用するアカウントを切り替えることが可能である。また、OneDrive に保存されている資料を閲覧する際には、Office Online を利用する。これにより、例えばアップロードされた資料が PowerPoint プレゼンター

Contents List

タイトル	PDF	DL数	スライド
Overlay Cloud で構成する論文再現環境	PDF	0	追加
DMATでの活用を見据え災害時にも手動設定変更不要で平常時に近い接続性を提供するネットワークの実現	PDF	0	追加
Annoteaサーバ「Wasabi」でのウェブアプリケーション全文検索手法の試験実装	PDF	0	追加
録画画像を用いた攻撃に耐性を持つパズル型認証方式の一検討	PDF	0	追加
表面筋電位計を用いた個人認証システムの実現に向けた検討	PDF	0	追加
多数決に基づく公開鍵決定プロトコルによる中間者攻撃への対応とその評価	PDF	0	追加
nmapを利用した学外からの学内ネットワーク監視と対策	PDF	0	追加
関西大学におけるSSHアクセスの収集と分析	PDF	0	追加
宮崎大学無線LANの利用状況の解析	PDF	0	追加
全学無線LAN利用ログ情報の解析と応用	PDF	0	追加
宮崎大学におけるパソコン必修化の取り組み	PDF	0	追加
琉球大学情報工学科における教育研究用情報システムの更新に関する研究	PDF	0	追加
Netspec: OpenFlowネットワークのテスト基盤の提案と試作	PDF	0	追加
OpenFlowネットワークにおけるトラフィック測定方法の提案	PDF	0	追加
IPv6時代におけるネットワーク状態評価手法の提案	PDF	0	追加
コネクション分割によるTCPスループット向上システムの構築と評価	PDF	0	追加

図 6 スクレイピングした発表情報に基づいた本システムによる資料リストの出力例

Figure 6 An Example of Documents Output List by This System Based on Scraped Presentation Information.

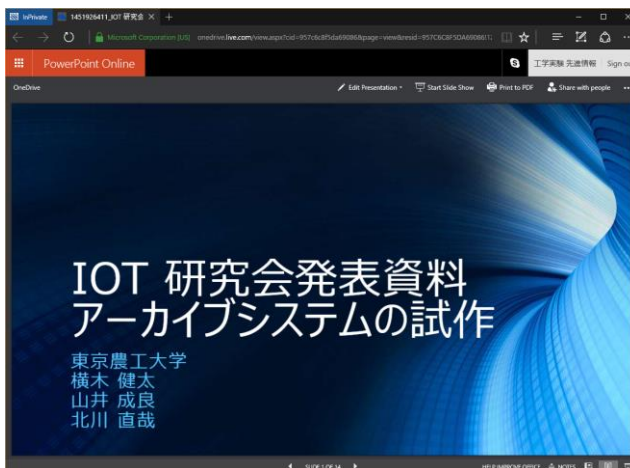


図 7 Office Online を用いた.pptx ファイルのプレビュー例
 Figure 7 A Preview Example of .pptx File with Office Online.

ション・スライドショー形式のファイルであれば、に示すようにブラウザ上でスライド内のアニメーションなども含めたほぼ完全な状態のプレビューを表示することが可能である。

また、ユーザが発表資料を閲覧する場合は、本システムは OneDrive 上に保存されているファイルに対する共有リンクを作成し、ユーザに提供する。万が一この共有リンク

が外部に漏えいした場合に備え、本システムは共有リンクが作成されてから5分後にこの共有リンクを無効化する。なお、OneDriveは一度作成した共有リンクを無効化した後に、再び共有リンクを作成しても、最初に作成したものとは異なるリンクになることを確認した。

OneDriveとの連携には、OneDrive API[2]を利用した。OneDrive APIを利用するためにはOAuth認証を行う必要があるが、一度のOAuth認証によって取得できるAccess Tokenには有効期限があるため、有効期限が切れた場合は再び再認証を行う必要があり今回のようなバックエンドサービスとしての利用には適さない。そこで、Refresh Tokenを利用して、操作ごとに新しいAccess Tokenを取得しなおしている。Refresh Tokenは最低でも6か月間は有効[3]であり、その間に新しいRefresh Tokenを取得すれば有効期間は延長され続けるため、実用上の問題は生じない。

OneDriveにアップロードされるファイルは、ファイル名の衝突を防ぐために、「<UNIX時間>_<元のファイル名>」というファイル名に付け替えている。

4.5 発表資料

本システムにアップロードすることができる発表資料のファイル形式は、PDF、ZIP、PowerPointプレゼンテーション・スライドショー形式のいずれかである。プレゼンテーションソフトウェアとしては、このほかにもMacユーザに広く使用されているKeynoteなどが存在するが、それらを含む他形式のファイルのアップロードを許可していないのは、該当するソフトウェアを持たないユーザが資料を閲覧することができなくなるからである。そのため、それらのソフトウェアを使用している場合はPDF形式に変換してからアップロードしてもらうことを想定している。なお、PowerPointプレゼンテーション・スライドショー形式に対応しているのは、先述の通りOffice Onlineを使用してブラウザ上でほぼ完全なプレビュー表示が可能であることに加え、Office Onlineの機能としてファイルをPDFに書き出すことが可能であるからである。ZIP形式は、発表資料が複数ある場合や、デモの動画などが別ファイルとして存在する場合を想定している。ZIP形式で資料をアップロードする場合は、PowerPointプレゼンテーション・スライドショー形式であってもOffice Onlineを利用できないため、あらかじめPDF形式等に変換してもらう必要がある。

4.1節で述べた通り、ユーザによって資料がアップロードされるのは主に研究発表会の前後であり、その時点ではまだ電子図書館上に発表情報が掲載されていないため、アップロードされた資料と電子図書館上の発表情報との関連付けをすぐに行うことができない。そこで、本システムはまずアップロードされた資料を講演題名と関連付けて保存する。そして、その後電子図書館の発表情報を表示する際に同名の発表情報がないかを調べ、同名の発表情報が見つ

かった場合はその場で関連付けを行う。

4.6 統計情報

本システムは、統計情報としてそれぞれの発表資料のダウンロード数を記録および表示することが可能である。

統計情報の取得には、資料のリンクを取得する際にアクセスされるクッションページを使用した。

4.7 セキュリティ対策

本システムでは、次に示すようなセキュリティ対策を行った。

まず、HTMLインジェクションやクロスサイトスクリプティング(XSS)などをはじめとする、ユーザの入力をそのまま出力する場合に発生する可能性のある脆弱性に対しては、ASP.NETのRequest Validation機能[4]を用いて対策した。この機能は、リクエスト中に危険な可能性のある値が含まれている場合に、エラーを返し、リクエスト自体を受け付けられない機能である。

また、SQLインジェクション攻撃への対策として、データベースへのアクセスには、すべてLINQ(統合言語クエリ)式[5]を使用したLINQ to Entitiesを用いた。LINQ to Entitiesでは、クエリの構築に文字列操作や文字列連結を行わないため、SQLインジェクション攻撃に対して十分な抵抗力を持っている[6]。

クロスサイトリクエストフォージェリ(CSRF)攻撃に対しては、本システムに対して何らかの変更を伴うリクエストを全てHTTP POSTリクエストとし、ページを表示する際にAntiForgeryTokenを発行してリクエスト時に送信される値と比較することによって対策した。

5. 今後の予定

3節でも述べた通り、現時点では本システムはIISおよびSQL Serverを用いて動作している。しかし、本システムを運用しようとしているIOT研究会のサーバはOSにFreeBSDを使用しているため、そのままでは動作させることができない。そこで、今後はMonoをはじめとする互換環境やFreeBSD上で動作可能なRDBMSを用いて本システムを動作させる予定である。

6. まとめ

本稿では、情報処理学会研究発表会のための発表資料アーカイブシステムの設計および実装について述べた。本システムでは、閲覧用アカウントや投稿用アカウントの追加を各研究会のメーリングリストで受信するメールによって自動的に行うことや、アップロードされた資料をクラウドストレージ上に保存し、システムが保持する必要のあるデータを最小限にとどめることなどによって、管理コストを最小限に抑制することが可能である。さらに、アップロー

ドされたファイルの利用情報等の統計情報の取得機能や、閲覧権限を持たない第三者への資料漏洩のリスクを最小限にとどめるための、ユニークな共有リンクの作成機能も導入した。なお、本システム的设计および実装は東京農工大学工学部情報工学科 2 年次配当の特別プログラム「先進情報工学実験 III」の一部として実施している。

参考文献

- [1] "ホーム | Microsoft アカウント".
<https://www.microsoft.com/ja-jp/msaccount/>
- [2] "OneDrive/onedrive-api-docs · GitHub".
<https://github.com/OneDrive/onedrive-api-docs>
- [3] "Does onedrive refresh token expire? · Issue #124 · OneDrive/onedrive-api-docs · GitHub".
<https://github.com/OneDrive/onedrive-api-docs/issues/124>
- [4] "Request Validation in ASP.NET".
<https://msdn.microsoft.com/en-us/library/hh882339.aspx>
- [5] "LINQ クエリ式 (C# プログラミング ガイド)".
<https://msdn.microsoft.com/ja-jp/library/bb397676.aspx>
- [6] "セキュリティに関する注意事項 (Entity Framework)".
<https://msdn.microsoft.com/ja-jp/library/cc716760.aspx#mt93>