

# ownCloud への大学教育向け機能の追加

2009SE188 村田 絵里奈 2009SE246 佐々 真由

指導教員: 後藤 邦夫

## 1 はじめに

近年, PC やスマートフォン, タブレット型 PC が普及したため, 一人で複数台のデバイスを持つことが珍しくなくなっている. それにともない, デバイス間でのファイル共有が普及し, 共有以外にも様々な機能を持つオンラインストレージが注目を集めている. また, オンラインストレージはインターネットにアクセスできる環境があれば, 簡単にデータのやり取りができる.

大学では情報通信技術を利用した講義が進んでいる. 情報通信技術を利用した講義の例として, 手書きや印刷では表現が困難な場合には, グラフ等をアニメーションで表示したり, 体験談等を動画で配信している. しかし, 学生にはその場での解り易さと多少の興味しか与えられない. その理由として, 講義中にスクリーンでアニメーションが再生しているだけであり, 後に行われる復習時には記憶を辿るしか方法はない. こうした問題を解決するため, 本研究ではオンラインストレージである ownCloud[5] を使用し, 大学教育向けの機能の追加を目的とする.

ファイル共有機能を利用すれば, 講義資料を指定の場所にアップロードするだけで講義を受講している学生全員に資料を配布することができる. また, 教員が講義資料を編集するさい, 自宅用のデバイスにデータを持ち込まなくとも, 一度デバイスを登録することで簡単に自宅で資料を編集, アップロード, 削除などができる. 同様に学生も複数のデバイスを登録さえすれば, 自宅だけでなくインターネット環境があれば通学中でも資料を作成, 保存することが可能となる.

このシステムの使用者は大学で講義を担当する教員および学生に限定する. 実現方法として, ownCloud を利用し, 追加のプログラムにより機能を追加する.

追加機能は, 課題提示を主とした分単位でのファイル公開期間や学生のログイン履歴・ファイルのダウンロードの履歴を記録・管理である. システム実現のため, 処理スクリプトの変更, 表示画面のプログラムを追加する.

それぞれの具体的な追加機能は以下の通りである. ファイルの閲覧期間の指定では, ファイルの閲覧を制限する処理スクリプトを追加する. 教師の PC から学生のログイン状態を閲覧では, データベースに学生 ID を格納するテーブル定義を追加する. 個人のダウンロード履歴では, データベースに学生 ID, ダウンロードしたファイル名を格納するテーブルを追加する. したがって, メニュー画面に入力フォーム, 表示画面を表示する. 実際に動作確認を行い, 追加システムが正常動作することを確認する. なお, システムの機能追加は佐々, 実験と評価は村田が担当する.

## 2 オンラインストレージの概要

この節では, ownCloud の概要と Dropbox[2] の概要, 採用理由を説明する. 複数あるオンラインストレージの中で Dropbox を説明する理由は, Dropbox は多くの大学で利用されているオンラインストレージ daigaku であり, また本研究で利用する ownCloud と機能が非常に似ているからである.

### 2.1 ownCloud の概要

ownCloud は, ファイル保管用にサーバマシン内でディスクスペースを貸し出すことにより, オンラインストレージとして使用できるサーバアプリケーションの一つである. さらに, サーバにデータを格納することで, セキュリティが確保される. オンラインストレージを使用することで, 他の通信デバイスと無料でファイルを自動同期することが可能となる. また, オープンソースなので, 使用者の意図に合わせた独自のシステムを改良できる. ownCloud には主に, データベースとして MySQL[4], Web サーバとして Apache[1] が使用されている. プログラミング言語については, システムとアプリケーションともに PHP で構成されている. また, コード量は約 43 万行 (php:約 42 万行, xml + txt:約 1 万行) で構成されている. クライアントは web ブラウザで ownCloud のログイン画面 (図 1) を表示し, ログインして使用する.

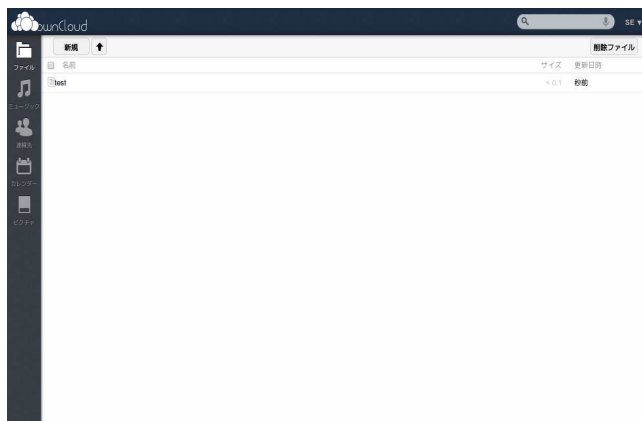


図 1 ownCloud の変更後のメニュー画面

### 2.2 Dropbox の概要

Dropbox は, オンラインストレージとローカルにある複数のコンピュータとの間でデータの共有や同期を可能とするオンラインストレージサービスである. Dropbox の有利な点は, PC の専用フォルダにデータを保存するだけで自動同期が行われることである. 他人との主な共有方法

として、リンク共有と共有フォルダがある。共有フォルダでは、他の Dropbox ユーザーとフォルダを共有でき、共有者全員が編集を行うことができる。またリンク共有においてリンクを作成したい場合、Dropbox により自動的に URL が付加され、その URL にアクセスすることでファイル内容をブラウザで表示したり、ダウンロードすることができる。

### 2.3 ownCloud の採用理由

他のオープンソースのオンラインストレージは開発や機能が不充分である中で、ownCloud は日本語が対応しており、機能も充実している。また、セキュリティ面においても、最も優れているといわれている。さらに ownCloud ではブラウザ経由でログインができ、ドラッグ&ドロップでファイルのアップロードが可能などといった、Dropbox と非常に似た機能を持ち合わせている。したがって、本研究に最も適したオンラインストレージであると考えている。また、本研究ではデータベースとして MySQL、Web サーバとして Apache を使用する。

## 3 大学教育向け機能の追加案

ここでは本研究で作成するシステムの追加案の概要を記述する。

### 3.1 ファイルの閲覧期間の指定

講義資料など、特定の期間中のみファイルを閲覧することを目的としている。主な使用方法として、教員が課題ファイルに公開期間を設定する必要がある。公開終了時間を過ぎると課題ファイルは見ることができない。しかし、閲覧期限内に課題を見ていなかった学生は当然課題内容が分からない。ただし、再度見たい場合は教員が課題ファイルに公開期間を設定することで、何度も閲覧することができる。次に、おおまかな流れを以下に示す。

1. 教員が閲覧期限を設けたいファイルに対し、事前に公開開始、公開終了の時刻入力し設定する。
2. 追加した情報は、データベースに資料のデータとともに保存される。
3. PHP スクリプトはデータベースから時刻を取り出し、時刻の比較演算をして、公開期間が開始されるまで資料のデータを非公開にする。
4. 公開開始と時刻が一致した場合は公開画面となる。同様に、公開終了と時刻が一致した場合は公開が終了し、非公開の画面に切り替わる。

上記の追加機能に必要なのが、データベースへの追加と PHP スクリプトの追加である。

### 3.2 教員の PC から学生のログイン状態を閲覧

学生 PC が同期されていることを前提とする。主な使用方法として、講義中に教員がログイン状態を確認することにより、学生の参加人数と参加している学生の ID が

把握できる。学生のログイン確認方法は、まず IP アドレスより学内と学外を判別する。南山大学では教室名を付与するドメイン名が与えられており、学生が講義教室内に居るか否かを判断することができる。さらに、学生は ID として学生番号を登録することで個人が特定できる。ここでは、データベースで ID およびパスワードの一致を判別し、サーバで IP アドレスを取得すると想定する。以上により、教員はログイン状態を確認することで、学生が教室外からログインしている場合にはその学生が講義を受けていないと簡単に判断ができる。ゆえに、出席確認で講義時間を使う必要もなく、学生側も出席が取られていると事前に通知がなされておれば、出席率が上がると判断する。ログの記録はデータベースに保存する。

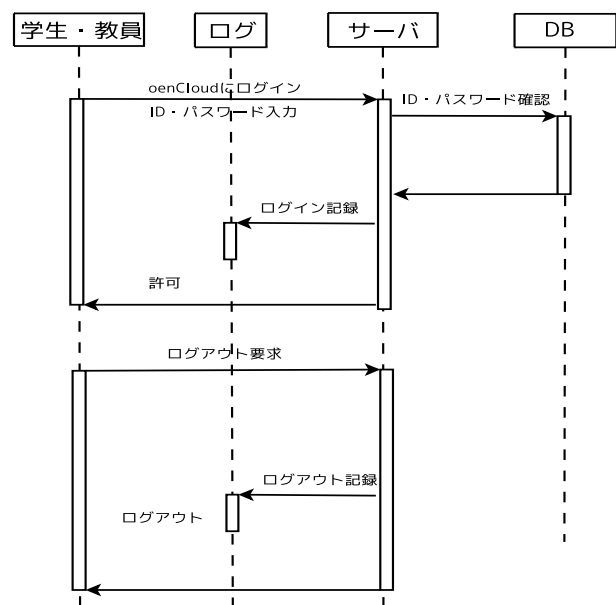


図2 ログイン・ログアウトのシーケンス図

次に、ownCloud のログの手順 (図2) を以下に示す。

1. ownCloud のログイン画面に ID・パスワードを入力する。
2. 受け取った情報はサーバを通しデータベースに送信され、データベースが判定する。
3. データベースからサーバにログイン許可の返答が送信される。
4. サーバからログに ID・ログインした時間が送信され、記録がされる。
5. サーバがログイン許可を送信する。
6. ログインが可能となる。

ログイン状態の表示方法として、学生のログイン人数、学生の個別でのドメイン情報を記載する図 (図3) を教員の PC に表示する。

	計 7名
10SE001	B201,seto-private
10SE002	B201,seto-private
10SE003	学外
10SE004	B201,seto-private
10SE005	B201,seto-private
10SE006	B202,seto-private
10SE007	B201,seto-private

図3 ログイン人数表案の図

### 3.3 ダウンロード時の履歴情報

講義資料などを学生がダウンロードしたい、履歴を残すことを目的とする。

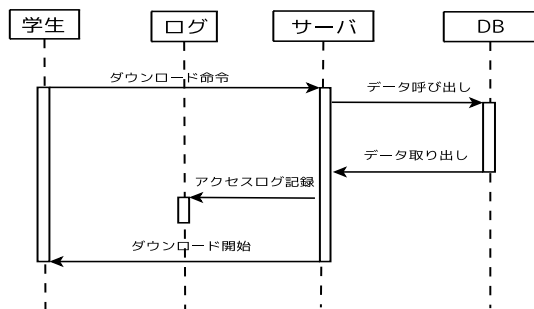


図4 ダウンロード処理のシーケンス図

資料のダウンロード時の流れを以下に示す(図4)。

1. サーバにダウンロード命令が送信される。
2. サーバはデータベースにデータを呼び出す。
3. データベースはデータをサーバに送信する。
4. ログにアクセス履歴が記録される。
5. ダウンロードが開始される。

資料のアップロード時における流れを以下に示す(図5)。

1. サーバにアップロードする。
2. サーバはデータベースにデータを保存する。
3. データベースはデータ保存の確認をサーバに送信する。
4. アップロードが完了したことをユーザに伝える。

続いてアクセス記録の確認の流れを示す。

1. アクセス記録をログに要求する。
2. ログはアクセス記録を表示する。

以上により、いつどの資料を誰がダウンロードしたかの情報を教員はログとして確認することができる。そのため、サーバ側でのアクセス履歴を保存するシステムを作成する。

主な使用方法として考えられるのは、学生のダウンロード状況を把握したい場合が挙げられる。課題資料など、ダ

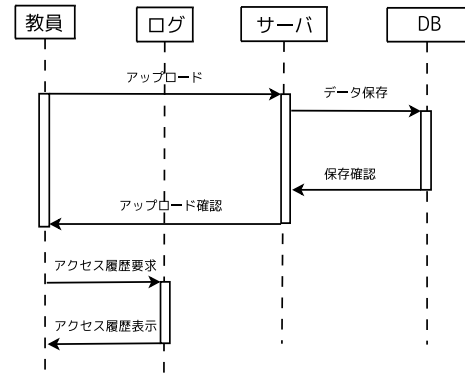


図5 ダウンロードファイルの保存とアクセス記録のシーケンス図

ダウンロードによる資料配布のさい、ダウンロードをしていない学生は課題などを見ていないことになる。また、前述の閲覧期間の指定やログイン状況と組み合わせることにより、教員側はより課題確認を正確に行うことが可能となる。

## 4 機能の追加

この節では、ファイルの閲覧期間の指定を実現する。ファイルを公開・非公開にする期間を指定し、閲覧機能を変更可能にした。実現するにあたり、データベースのフィールドの追加、PHP スクリプトでデータベースへ入力内容を登録、閲覧期間を設定する入力画面を作成し、ファイルの公開・非公開処理を変更する。ただし、プログラムの追加先は ownCloud のソースコードのみとする。

アップロード時に閲覧期間を設定する予定であったが、システムの追加が困難であるため、アップロード後にシステムを追加する。アップロード後では、ownCloud フォルダ内 (apps/files/templates) の part.list.php にてアップロードファイル一覧が表示される。個々のファイルでそれぞれ閲覧期間の設定をするシステムにするため、名前、サイズ、更新日時を表示する欄に閲覧期間の設定を追加するシステムを考察する。

以下に追加プログラム [3] と説明を述べる。はじめに、閲覧公開時刻と閲覧終了時刻の入力フォームを作成する。form タグの中に、input タグにて閲覧公開時刻 (timeno1) と閲覧終了時刻 (timeno2) の変数を作成し、それぞれ時刻の値を格納する一行テキストボックスを作成する。PHP の時刻を取得するさい、時刻表記の設定がグリニッジ標準時間であったため、日本時間に変更した。次に、if 文で閲覧開始時刻から閲覧終了時刻までファイルを表示する条件を付ける。if 文のプログラムは以下に示す。

```
< ?php if($_POST["timeno1"] <=
date("Y/m/d/H/i")
&& $_POST["timeno2"]): ?>
ファイルの中身を表示するプログラム
< ?php else: >
< ?php endif; ?>
```

本研究では、ファイルの閲覧期間の指定で、データベースのフィールドの追加、および PHP プログラムによるデータベースへの接続は、調査とプログラムの試行錯誤を進めていくにあたり、データベースへの接続は困難と判断した。

## 5 実験方法

本システムに置ける実験方法について述べる。

### 5.1 システムの動作確認

まず、本システムが正確に動作するかを以下の手順で確認する。

1. 教員用 PC でログインを試す。
2. 教員用 PC で閲覧期限指定のファイルをアップロードする。
3. 学生用 PC でログインを試す。
4. 学生用 PC で閲覧期限中のファイルが閲覧や非表示となるかを確認する。
5. 学生用 PC のログアウトを試す。
6. 教員用 PC でログアウトを試す。

手順どおり動作確認を実施し、追加機能が正常に動作することが確認された。次に、ダウンロードや削除など既存の機能についても動作を確認し、追加機能によるデータの破損がないことを確認した。

### 5.2 追加前後の比較

閲覧制限について、開始時刻と終了時刻の入力フォームを追加した。(図 6)

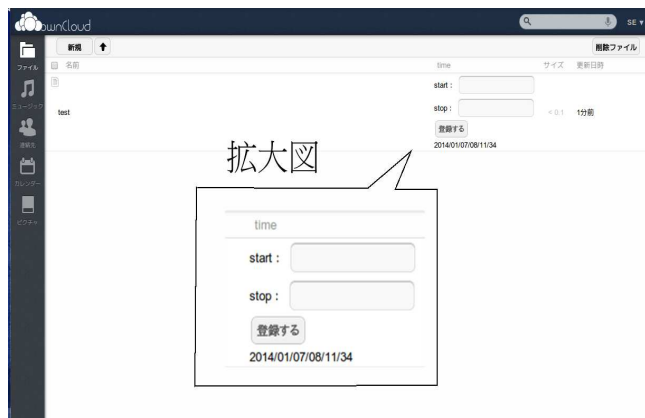


図 6 ownCloud の変更後のメニュー画面

図 1 と比較し、入力フォームが追加されたことが確認できる。図 6 の入力フォームに、start は閲覧開始時刻、stop は閲覧終了時刻を入力する。入力する時刻は分単位であり、半角英数字で日本時間の年/月/日/時/分の形式で入力し、入力後に登録ボタンを押して設定作業を終了する。設定後にブラウザの更新をすると、閲覧期間外ではファイルの中身が閲覧不可能となり、閲覧期間中はファイルの中身が閲覧可能となる。また、設定期間中は入力フォームが

非表示となり、閲覧終了後は入力フォームが表示され閲覧期間が繰り返し設定できる。ファイルをアップロードした後はファイルの中身が非表示となり、閲覧制限が必要ない場合はファイルをアップロードした時刻を開始時刻とし、終了時刻を年単位に設定することで長期間の閲覧が可能となる。

これにより、教員は閲覧期限を設定したいファイルに任意の時間で閲覧期限を設定することが可能となった。さらに学生は登録がされたデバイスを利用し、インターネット環境のある場所にて閲覧時間中にファイルを開くことが可能となった。よって、既存の機能より追加機能の方が大学教育向けの追加であると判断する。

## 6 おわりに

本研究では、大学教育向けの機能の追加であるファイルの閲覧期間の指定を可能にした。この結果より、任意の時間で閲覧期間を指定することが可能となった。しかし、ownCloud の膨大なコード量により解読が困難であるため、必要なソースコードが書かれているファイルの検索方法や読み方など、実装前の手段について様々な方法で調べる必要があった。

より良い機能追加の実現には今後の課題も多く、現段階では次のような課題がある。

1. PHP スクリプトとデータベースとの連結。
2. 本研究で未実装に終わったログイン状況の一覧、ダウンロード時の履歴情報。
3. どの程度の規模に耐えられるのかの負荷実験。

ownCloud は膨大なコード量で構成されているため、解読に時間がかかり、いくつかの追加案が未実装に終わった。また、追加プログラムの試行錯誤により、PHP スクリプトから入力する指定時間をデータベースに格納することはできなかった。本研究では少人数の学生がいる小規模の講義を想定としている。そのため、多人数の学生がいる大規模の講義を想定とした負荷実験を試行することで、大学の教育現場における利便性が向上すると考えられる。

## 参考文献

- [1] Apache Software Foundation: The Apache HTTP Server Project (accessed Jan. 2014). <http://httpd.apache.org/>.
- [2] Dropbox Inc.: Dropbox (accessed Jan. 2014). <https://www.dropbox.com/>.
- [3] naito: PHP Labo (accessed Jan. 2014). <http://www.php-labo.net/tutorial/mysql/php.html/>.
- [4] Oracle Corporation: MySQL (accessed Jan. 2014). <http://dev.mysql.com/>.
- [5] ownCloud Inc: ownCloud (accessed Jan. 2014). <http://owncloud.com/>.