平成 20 年度

プロジェクト研究報告書

大学向け OpenID 認証サーバの構築

1090374　　別府　瞳

指導教員　　妻鳥　貴彦

高知工科大学 情報システム工学科

要 旨

大学向け OpenID 認証サーバの構築

別府　瞳

近年，インターネットの発達により様々な Web サービスが提供されている．それに伴い，e-Learning や学内ポータルサイトなど，大学内でも Web サービスを利用する機会が増えて来ている．ユーザはこれらを利用することで様々なメリットを得ることができるが，利用したいサービスによってはユーザ登録をする必要が出てくる．このようなサービスはいくつもあり，ユーザはその度にアカウントやパスワードの設定，また個人情報を何度も入力する必要があり，非常に手間と時間がかかる．

本研究ではシングルサインオンの技術を用いて，ユーザ登録に関する手間やアカウント管理の負荷を軽減するシステムを提案する．シングルサインオンとは，一度の認証で許可されている複数のサービスを利用できるシステムである．シングルサインオンを実現する技術として LDAP や SAML などがあるが，今回は OpenID を利用する．OpenID とはひとつの ID で複数の Web サイトの認証を可能にするものである．

これにより，ユーザのアカウント管理の負荷や個人情報入力の手間を軽減する．結果として，1 つの OpenID で複数の OpenID 対応サービスにログインすることができる環境を構築することができた．

キーワード　OpenID，シングルサインオン，

目次

目次

図目次

第 1 章

はじめに

近年，様々な Web サービスが提供されるようになり，一般的によく利用されるのはもちろん，最近では大学内でも Web サービスを利用したシステムが導入されている．一般的に良く利用される Web サービスの例としては，Yahoo! Japan や Google などの検索エンジン，楽天や Amazon などのオンラインショップなどが挙げられる．大学内（学生生活）で利用されるものとしては，在学生の利用を対象としたポータルサイトや，e-Learning システム，SNS（Social Networking Service）などが存在している．検索エンジンなどはそのままでも利用可能となっているが，オンラインショップを利用したい場合や，学内のサービスを利用したい場合にはどちらのサービスにもユーザ登録をする必要がある．新たに Web サービスを利用するためには，その都度ユーザ登録が必要であり学生は手間と時間が必要になる．また複数のアカウントを所有することにより管理が複雑になり，Web サービスごとに別のアカウントでログイン処理を行わなければならない．

本研究では，学生のアカウント管理についての複雑さを解決するために，シングルサインオンの技術の 1 つである OpenID を用いて，アカウントの一元化を可能にする．また，学生が Web サービスを利用しやすい環境を構築する，これにより，学生が所有しなければならないアカウントを減らすことができ，学生の負担を軽減することが可能である．

第 2 章

研究の背景

2.1　大学における学生のアカウント管理の現状

近年，ICT の普及に伴い，様々な Web サービスが提供されており，大学内でも Web サービスを利用する機会が増えて来た．学生は，何かを調べたり買い物をしたりする状況や学習計画を立てるなどの状況において，多数の Web サービスを必要に応じて使い分けている．また Web サービスを効果的に利用するためには，サービスごとに氏名や住所などの個人情報を登録し，アカウントを作成する必要がある．アカウントは Web サービスごとに作成する必要がありユーザ登録を行うことにより様々なメリットを得ることが出来る．その結果，学生は複数のアカウントを所有し，管理しているという現状がある．学生が所有しているアカウントとして，2 つのタイプが挙げられる．1 つは私生活の中で一般的によく利用されている Web サービスなどで，もう 1 つは学生生活で利用される Web サービスである．

一般的によく利用されている Web サービスには，以下のようなものが挙げられる．

- 検索エンジン（Yahoo! Japan，Google など）
- オンライン・ショップ（楽天，Amazon など）
- SNS（mixi，さとあいなど）

次に，学生生活でよく利用される Web サービスは以下のようなものがあげられる．

- 学内用ポータルサイト
- e-Learning システム
- 就職支援サイト（リクナビ，マイナビなど）

これらの Web サービスのアカウントを作成する場合，Web サービスごとにユーザ登録を行う必要があり，同様の情報を何度も入力しなくてはならない．アカウントやパスワードの設定や，個人情報としては氏名や電話，郵便番号，住所などがある．個人情報に関しては基本的に同様の項目が多く，手間と時間がかかる．

これらのことから，学生は各サービスごとにユーザ登録を行い，アカウントを取得する必要がある．また，利用するアカウントが増加するとアカウントとパスワードの管理が次第に困難になってくる．

2.2　本研究の目的

本研究では 2.1 節で述べた問題を解決するために，シングルサインオンを用いてアカウントの一元管理を可能にし，また学生が Web サービスを利用しやすい環境を構築する．シングルサインオンとは，アカウントの管理や何度もログイン処理をしなけらばならないという問題を解決する為に開発された．これにより，複数のアカウントの作成やサービスごとの認証処理を何度も行わなくても複数の Web サービスが利用可能となる．

第 3 章

シングルサインオン

大学生のアカウント管理の負荷を軽減する為に，様々な Web サービスのアカウントを一元化する．本章では，シングルサインオンを用いたアカウント一元管理を実現するシステムの提案と設計について述べる．

3.1　シングルサインオンとは

シングルサインオンとは，ユーザが一度認証を受けるだけで，許可されているすべての機能を利用できるようにするサービスである．

シングルサインを利用することで，ユーザは複数のアカウントを所有する必要がなくなり，一度の認証で複数のサービスを利用できるようになるため，Web サービスごとに認証処理をする手間も軽減することが可能である．

3.2　シングルサインオン技術の比較

シングルサインオンを実現するシステムとしては，以下のようなものが挙げられる．

- LDAP（Lightweight Directory Access Protocol）

  ディレクトリサービスにアクセスするためのプロトコルである．ディレクトリサービスでは，ネットワークを利用するユーザ名やマシン名などの様々な情報を管理している．このディレクトリにアクセスすることで，認証情報の統合を行う．標準化されたプロトコルでパフォーマンスもよかったため普及したが，利用の都度になんらかの認証手続き

が必要であったり，企業内の複数のアプリケーションがそれぞれ個別のディレクトリを持ってしまうなどの問題点がある．

- SAML（Security Assertion Markup Language）

  ユーザ認証に用いる ID やパスワードを安全に交換するためのもので，XML（eXample Markup Language）仕様である．認証情報の交換方法は SAML プロトコルとしてまとめられており，メッセージの送受信には HTTP（Hyper Text Transfer Protocol）もしくは SOAP（Simple Object Access Protocol）が使用されている．ユーザの認証情報を他のサーバと共有し，複数の Web サイトやサービスを最初の一度の認証で利用することが可能となる．

- OpenID

  1 つの ID でインターネットの様々な Web サイトの認証を実現する技術で，Web サイトの URL 形式で構成されている．URL というアイデンティティを認証する仕組みで，ユーザの本人性の認証とそれに基づいた認可は行わない．オープンソース方式で公開されているため，様々な Web サイトが採用を始めている．

上記に述べた LDAP や SAML は，システム同士がお互いに信頼関係を構築することでログイン処理を共有し，システム全体の認証の手間をなくしている．ユーザは ID とパスワードを複数回入力しなくてもよいというメリットがある．

OpenID は，これまでの LDAP や SAML とは異なり，認証サーバでアカウントを取得すれば，OpenID に対応している全ての Web サイトで使用することが可能である．また個人情報も認証サーバで管理することができ，他の Web サービスと共有することが可能である．このため，従来ならば新たな Web サービスを利用する際に必要だったアカウントとパスワードの設定，個人情報の入力にかかる手間を軽減することができる．本研究では，OpenID を用いて大学内で使用する Web サービスのアカウントの一元化を可能にするシステムを構築する．

3.3　OpenID

OpenID とは、複数の Web サイトで同じ 1 つの ID 情報による認証（シングルサインオン）を可能にするサービスのことである．米国の OIDF（OpenID Foundation）をはじめとする推進団体によって普及促進が図られている．従来の認証方式では，利用したい Web サービスごとにアカウントとパスワードを入力して認証処理を行っていた．これに対して OpenID ではユーザ情報を管理しているサーバのみが認証処理を行うため，各 Web サービスでは認証処理を行う必要がない．OpenID の発行，認証を行うサーバを OpenID Provider（OP），OpenID に対応している Web サービスは Relying Party（RP）と呼ぶ．OpenID はオープンソースなので，誰でも認証サーバを構築することが可能である．OpenID は URL 形式で構成されおり，ユーザはまず OP にユーザ登録を行い OpenID を取得する．この URL を RP に入力すると，RP は URL から OP に認証依頼を送信する．OP は受け取った認証依頼に対して結果を送信するが，この結果が認証を許可するものであればユーザは RP にログインすることが可能となる．

図 3.1 に，OpenID による認証プロセスを示す．

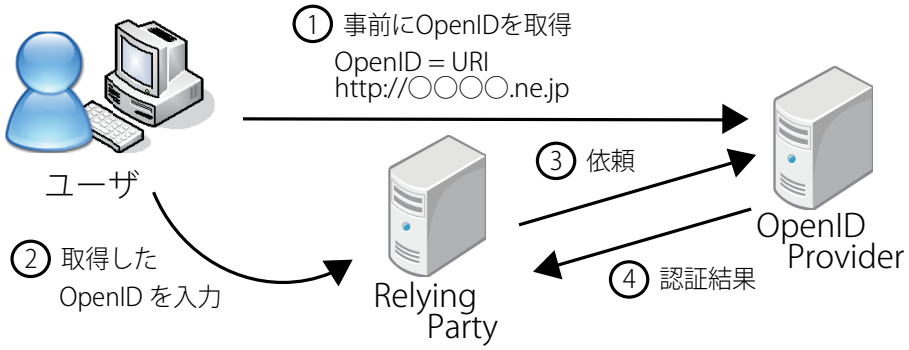

図 3.1　OpenID による認証プロセス

OpenID を利用することで，利用したい Web サービスごとに必要なユーザ登録の手間を軽減することが可能である．また，OP で管理している個人情報は，ユーザの意思によって

は RP と共有可能となっているので，今まで個人情報を入力しなければならなかった手間を軽減することも可能である．

本研究の目的は，OpenID を利用することで学生のアカウント管理や入力負荷の軽減を行い，シームレスに学内のサービスにアクセス出来るようにする．具体的には大学で信頼できる OP を構築し，大学内でのサービスに RP を対応させる．

第 4 章

システムの設計と構築

4.1 OpenID Provider の仕様

OpenID Provider（以下，OP と記す）は OpenID の認証サーバである．OpenID は，誰でも認証サービスを提供出来るという仕組みである事から，その信頼性に問題がある．そこで今回は大学で信頼できる OP を構築する．信頼性を高めるために，OpenID では SSL やハッシュ関数を用いて秘密鍵を利用可能となっている．本研究では JanRain で配布されている OpenID 対応ライブラリの PHP 版を使用した．データベースには MySQL を利用する．

4.2 Relying Party の仕様

Relying Party（以下，RP と記す）とは，OpenID の認証を望む Web サイトのことである．RP を構築するためには，RP 側のログイン画面に OpenID 用のログインフォームを設置する．ユーザはそのフォームに自分の OpenID を入力する．このとき入力された URL にはサーバの URL をメタ情報として組み込んであるので，RP はその URL にアクセスを行うことで認証依頼を送信する事が出来る．認証が得られれば RP はユーザにログインする事ができる．今回は CMS として代表的な Moodle と，SNS エンジンとして代表的な OpenPNE を RP として利用した．

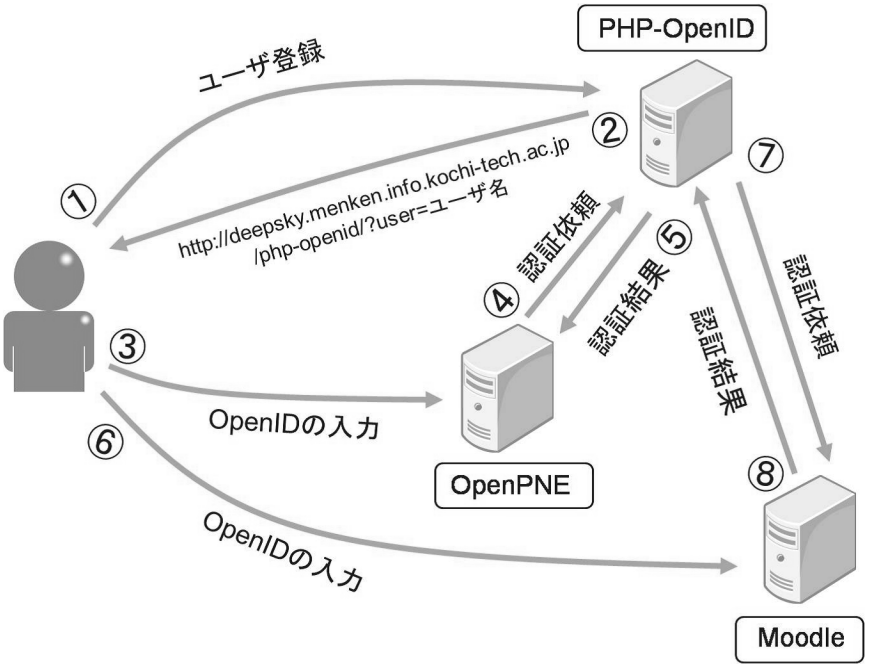

図 4.1　システム関係図

4.2.1　Moodle の特徴

Moodle は，コース管理システム（Course Management System：CMS）とよばれるソフトウェアの 1 つで，オープンソースで開発されているため無料で導入することができる．Moodle は対面式授業を補完して，学生と教師，学生と学生などのインタラクションを支援することができ，授業の資料の配布や，レポート提出，質問や会頭などのやりとりを Web を通じて行うことができる．Moodle は，Moodle1.6 から OpenID にも対応するようにプラグインが開発されている．

4.2.2　OpenPNE の特徴

OpenPNE はオープンソース方式で開発が行われてきた SNS エンジンである．オープンソースで開発されているため誰でも無償で利用することが可能であり，企業やサークルな

どの様々な場面で利用されている．今回利用した OpenPNE3.0 では国際化を実現しており，辞書ファイルを追加することで多くの言語へ対応が可能となっている．またプラットフォーム化を目指し，多彩な API を搭載している．これによりプログラマにとっても開発がしやすくなっている．OpenPNE は OpenPNE3.0 から OpenID にも対応したことから，外部の ID との連携が可能になっている．

第 5 章

動作検証

本章では，本システムの動作検証を行った結果について述べる．

5.1 OpenID Provider の検証

まずは，構築した OpenID サーバにアクセスしユーザ登録を行う．OpenID サーバにアクセスすると，図 5.1 の画面が表示されるので，新規作成ボタンを押してアカウント作成画面を開く．

図 5.1　OpenID　トップ画面

図 5.2 のアカウント作成画面が表示されたら，各項目に入力し登録を完了する．大学向け Web サービス内で利用することを想定しているので，ユーザ名には大学で使用されるアカウントを入力し，パスワードは任意のものを入力している．

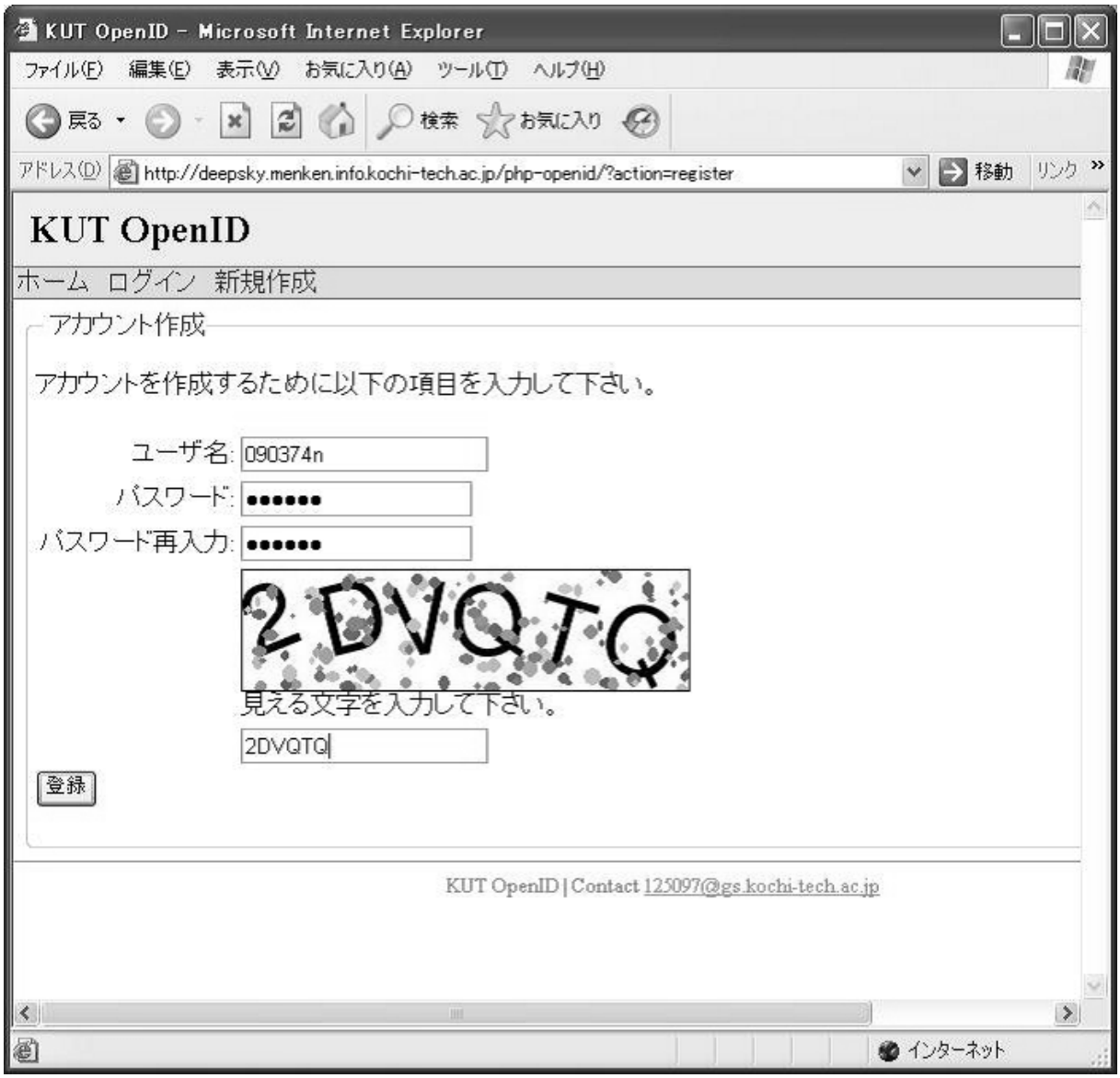

図 5.2　OpenID　アカウント作成画面

登録が完了すると，図 5.3 が表示される．ここに表示されている「あなたの OpenID」が，学生が実際に使用する OpenID となる．

5.1　OpenID Provider の検証

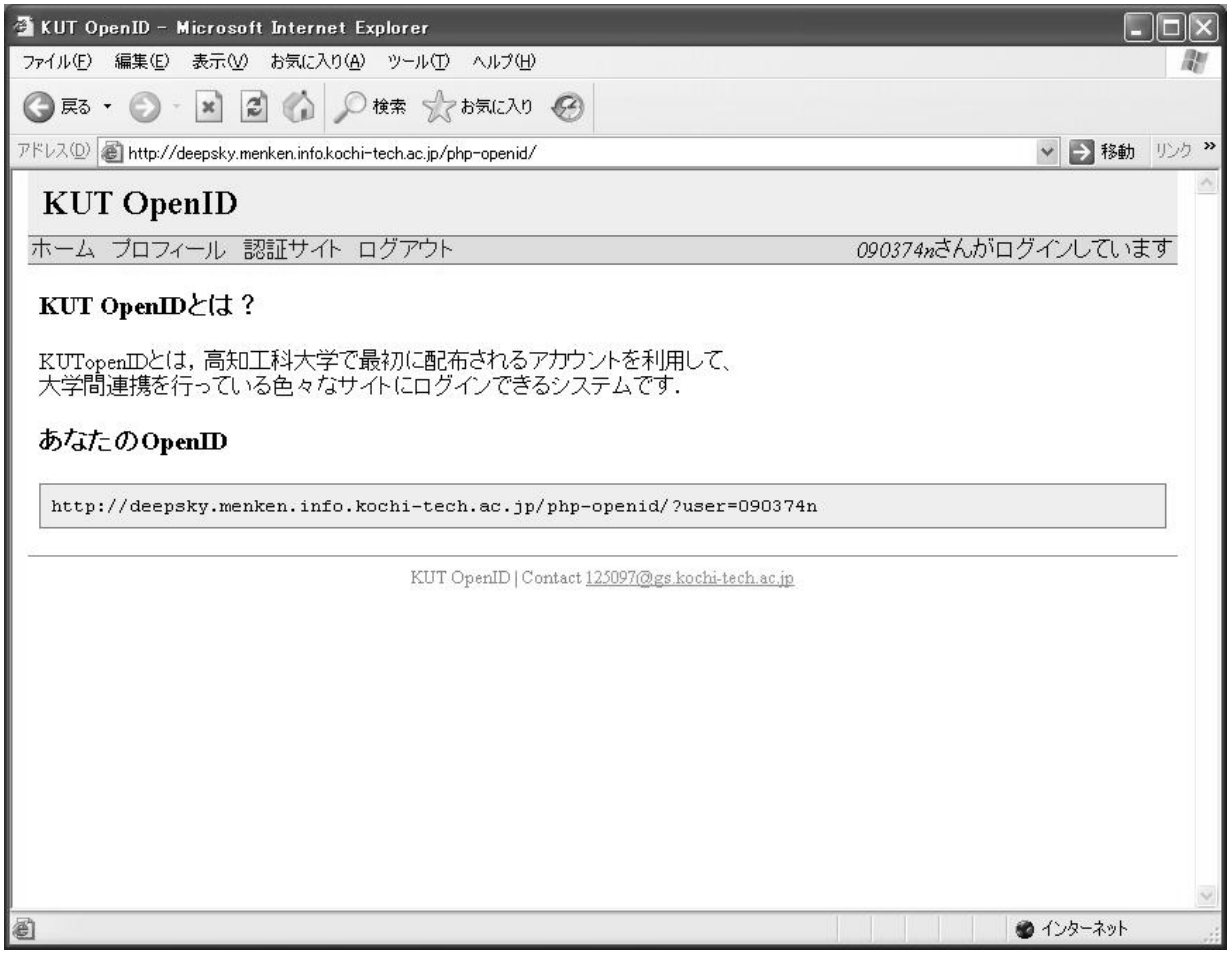

図 5.3　OpenID　アカウント取得後の画面

プロフィールボタンを押すと，図 5.4 に示すような個人情報の編集画面が表示される．個人情報として管理しているのは図 5.4 にある項目のみとなっているが，必要に応じてこの他にも項目を増やすことが可能である．

これらの項目は他の Web サービスと共有することが可能である．

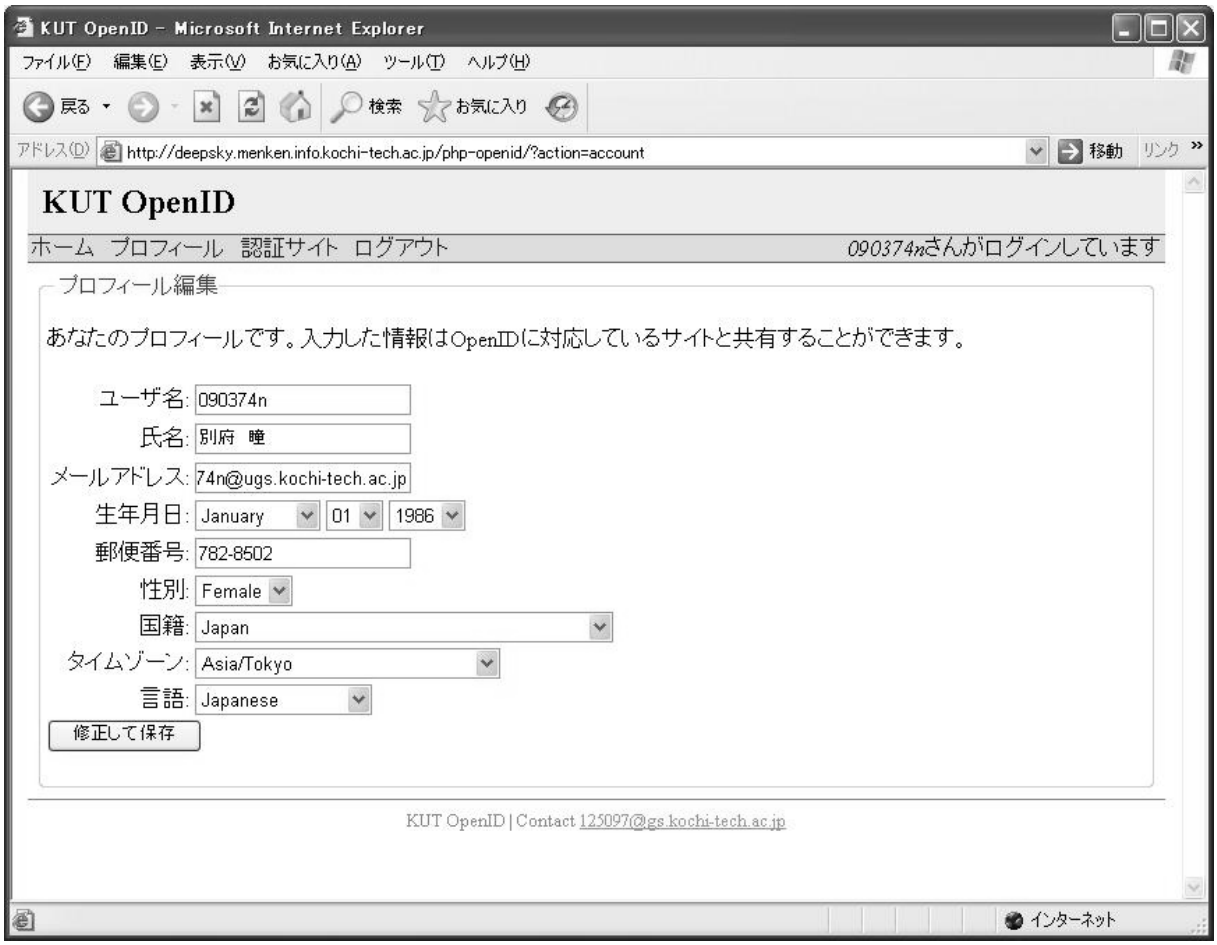

図 5.4　OpenID　プロフィール編集画面

5.2　Relying Party の検証

次に，実際に作成した OpenID で RP にログインできることを確認するために，Moodle と OpenPNE について動作検証を行った．

5.2.1　Moodle

Moodle にアクセスすると，図 5.5 の様な画面が表示される．この段階ではユーザ登録を行っていないので何もない状態である．画面の右側中央に，OpenID を入力してログインするフォームが用意されているので，このフォームに OpenID を入力してログインする．

http://deepsky.menken.info.kochi-tech.ac.jp/php-openid/?user=090374n

5.2 Relying Party の検証

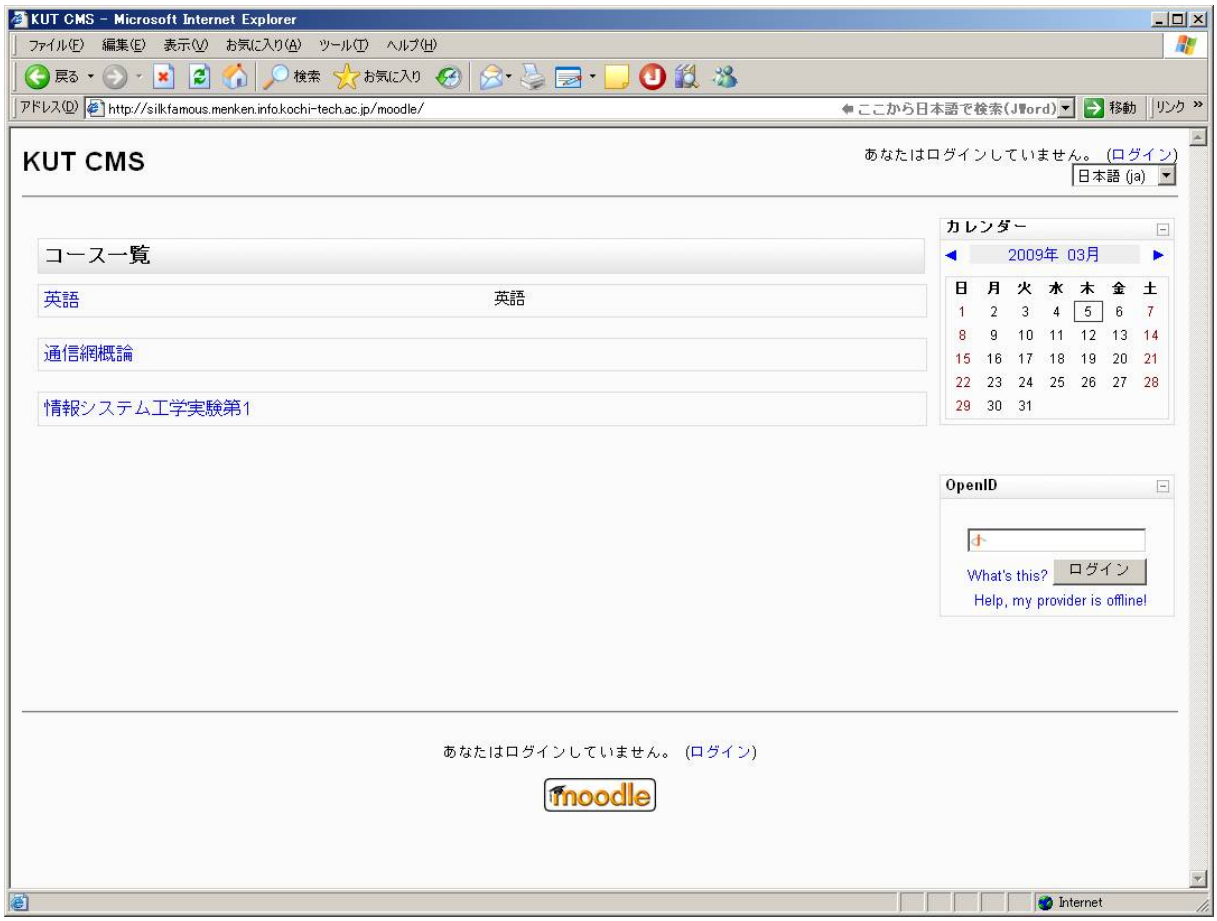

図 5.5 Moodle トップ画面

また，図 5.5 の右上にあるログインリンクからもアクセス可能である．その場合は図 5.6 のような画面が表示され，トップ画面にも表示されていた OpenID を入力するフォームがあるので，OpenID を入力してログインする．

5.2　Relying Party の検証

図 5.6　Moodle　ログイン画面

OpenID を入力してログインすると，一度認証サーバに飛び自分がログインしようとしている Web サービスへの許可を出すかどうか選択をする画面が表示される（図 5.7）．Moodle は認証サーバで管理している個人情報の共有が可能なため，認証の際に個人情報の転送をするかどうか問われる．ここは個人の判断で共有するかどうか決めれる．今回構築した Moodle は，学生のレポートの提出や講義資料のダウンロードなど大学内で利用されることを想定しているため，個人を特定する必要がある．そのため管理されている個人情報のうち，必要最低限の情報は予め必須項目となっている．

5.2　Relying Party の検証

図 5.7　OpenID　認証画面（Moodle）

認証許可を選択すると，各情報が Moodle に引き継がれ，プロフィールの入力画面には先ほど選択した情報が共有されすでに入力された状態で表示される（図 5.8）.

5.2　Relying Party の検証

図 5.8　Moodle　プロフィール編集画面

この情報を登録するとユーザ登録完了となり，Moodle へログインすることができる．次回からも同様に OpenID を入力すればログイン可能である．ログイン後の画面は，図 5.9 のようになる．

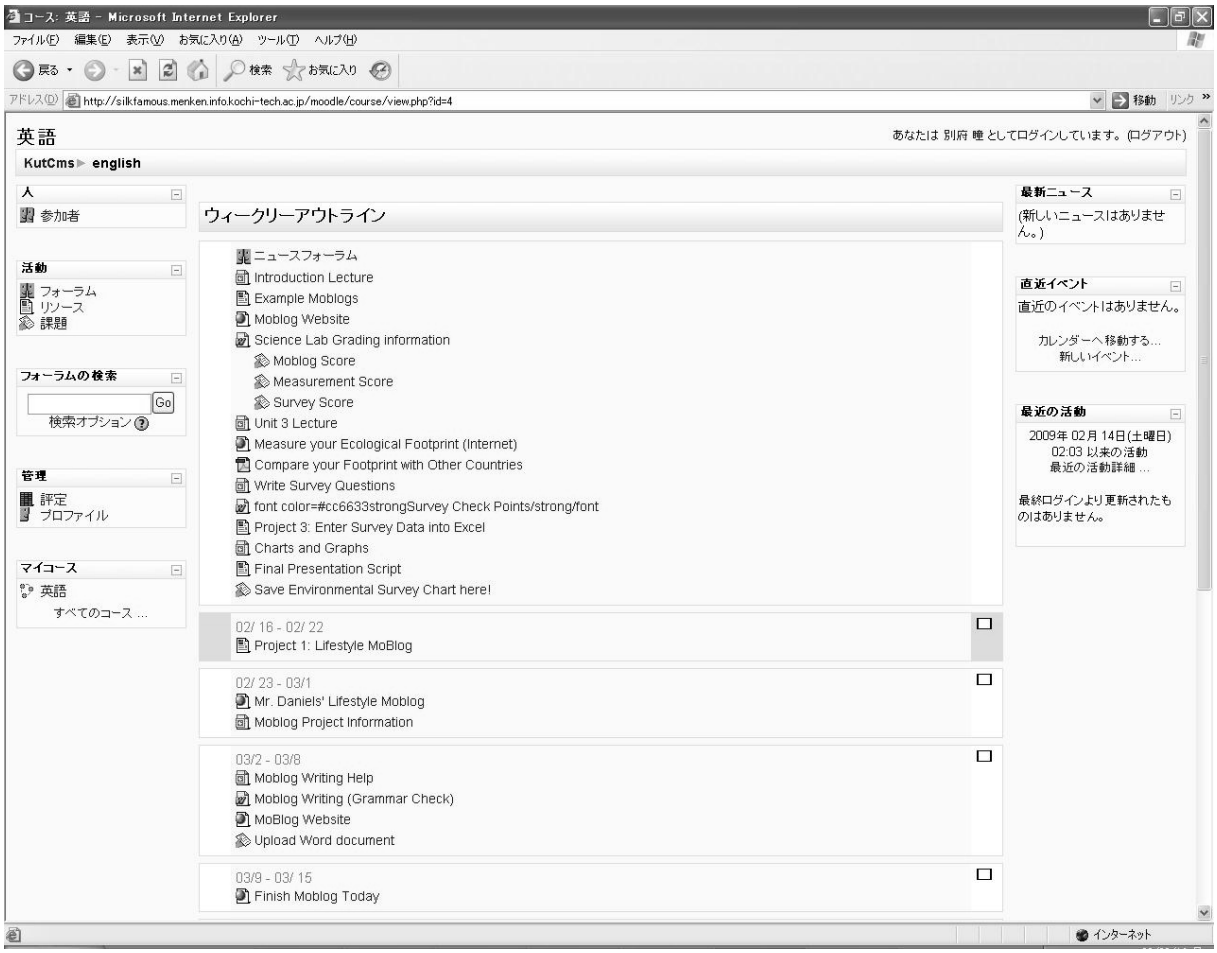

図 5.9　Moodle　ログイン後画面

5.2.2　OpenPNE

5.2.1 で述べた Moodle の場合と同様の手続きを行う.

OpenPNE にアクセスすると，図 5.10 のような画面が表示される．OpenID を入力する項目があるので，OpenID を入力してログインを押す.

5.2　Relying Party の検証

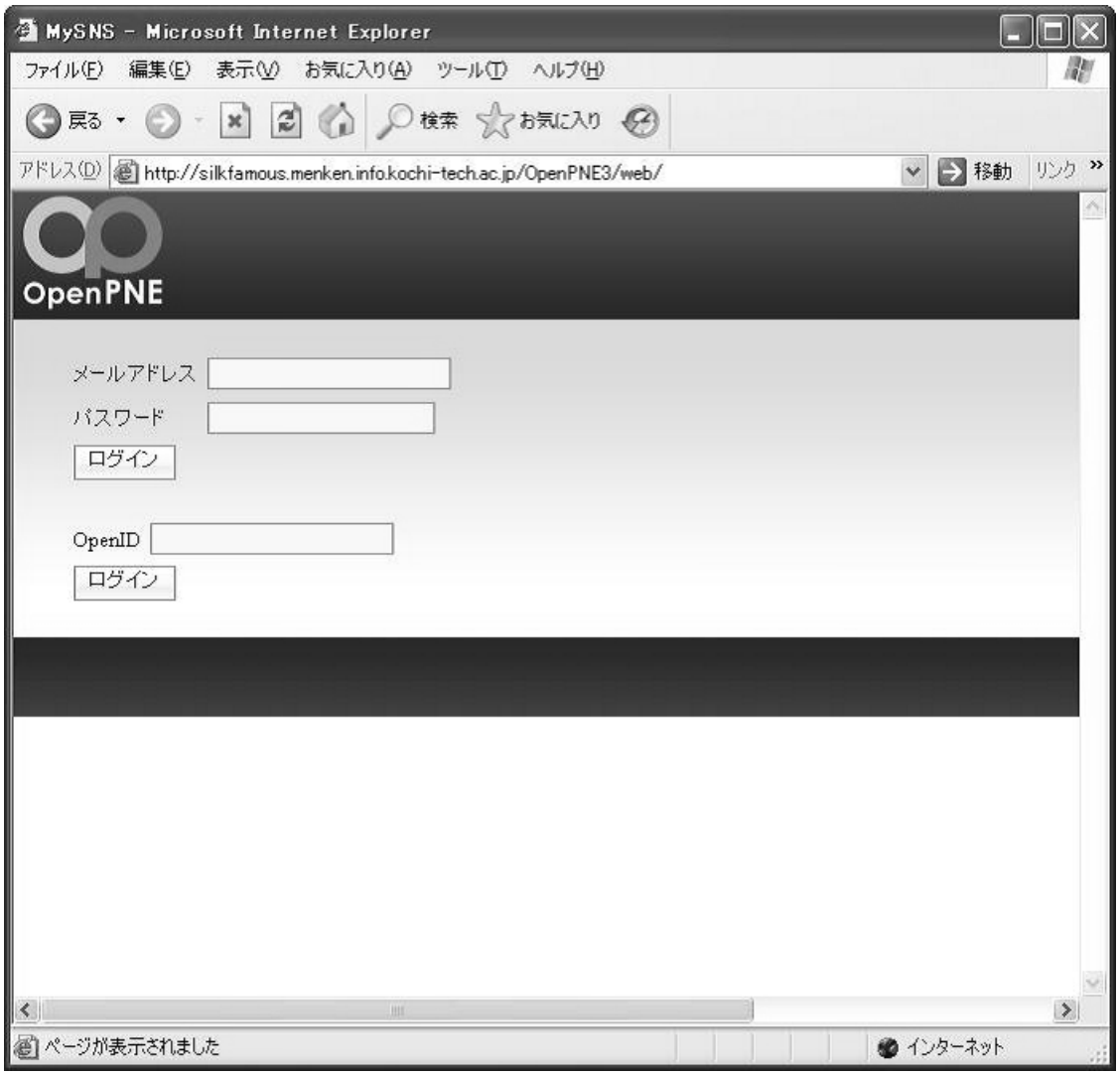

図 5.10　OpenPNE　ユーザ登録

Moodle の場合と同様に，認証許可を出すか問われるので許可の選択をする（図 5.11）．Moodle では個人情報の共有を選択する画面が表示されるが，OpenPNE では学生情報を必要としないため，学生情報の共有は行われない（図 5.11）．

5.2　Relying Party の検証

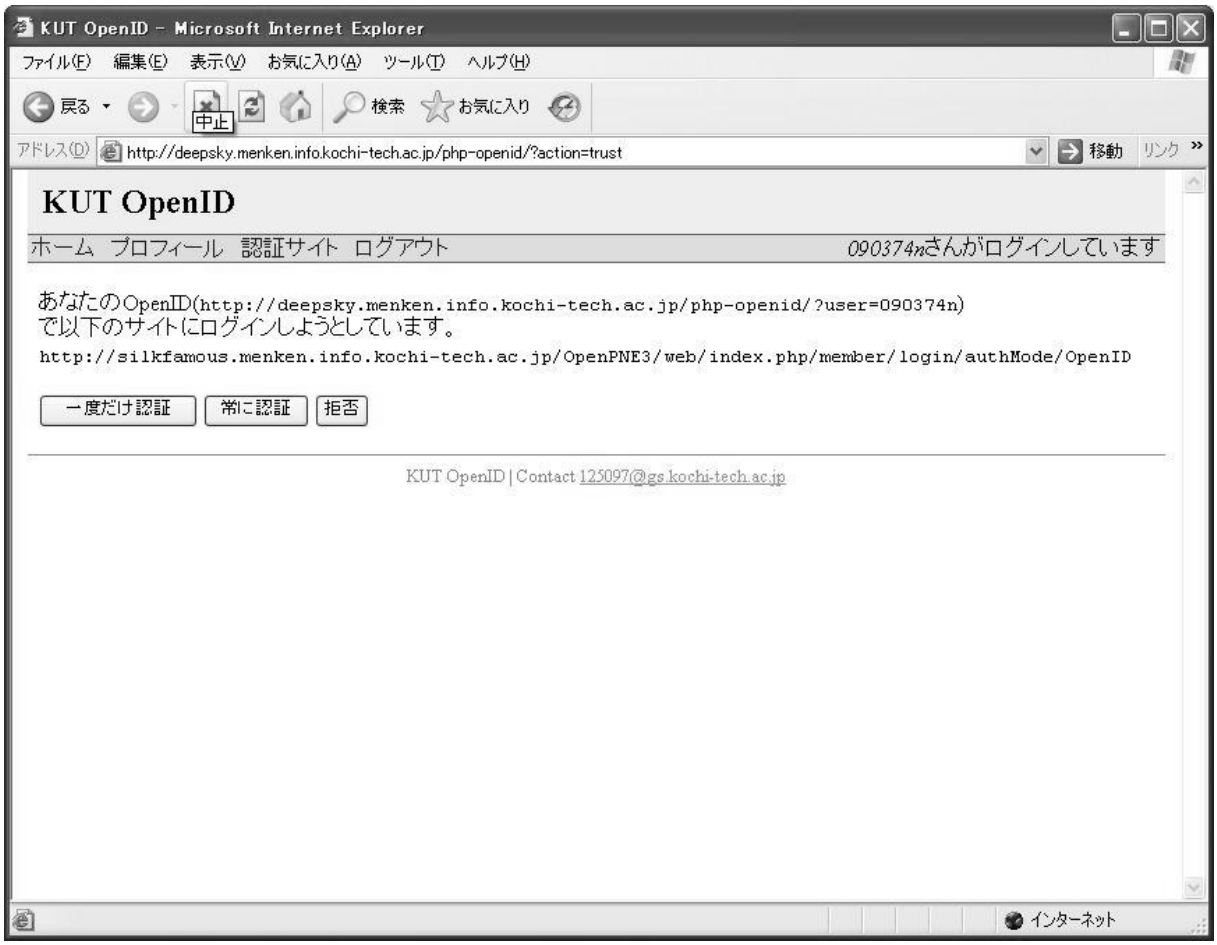

図 5.11　OpenID　認証画面（OpenPNE）

ログイン完了後は図 5.12 に示すような画面が表示される.

図 5.12　OpenPNE　ログイン画面

5.3　考察

今回の検証では，OP として php OpenID サーバ，RP としては代表的な CMS である Moodle と代表的な SNS である OpenPNE を利用した．

高知工科大学内で OpenID を使用することを想定して環境を構築し，検証を行った．本研究では OpenID の取得をする段階から検証を行ったが，実際に OpenID を利用するにはすでに大学側で学生の OpenID を取得し，入学時に配布する学内用のアカウントと同じように配布するのが望ましい．なぜならば学生個人にユーザ登録を任せていたのでは，アカウ

ントの統一性や，管理者側の学生管理が困難になってしまうことが考えられるからである．また，本学ではすでに LiveCampus というポータルサイトで学生の成績情報や履修登録などに関する Web サービスを提供していることから，本学で OpenID を利用するためにはすでに利用されている LiveCampus を OpenID に対応させる必要がある．しかし現段階では LiveCampus は OpenID に対応していないため，今後新たにポータルサイトを構築する際には，OpenID に対応するできるように構築することが望ましい．

第 6 章

まとめ

本研究では，学生のアカウント管理の問題点を解決する為に，OpenID 認証サーバを構築し，複数の Web サービスに 1 つのアカウントを利用してアクセスできるようにした．OpenID 認証サーバを構築したことで，学内サービスで利用されるアカウントが一元化でき，学生のユーザ登録する手間や，今まで何度も個人情報を入力しなければならなかった手間を軽減した．また個人情報については，OpenID 認証サーバで一括で管理し，他の Web サービスと共有することで学生のユーザ登録時の負荷を軽減した．

本研究の段階では大学間連携を想定しているが，今後は大学間だけに留まらず一般的な Web サイトとの連携や就職活動支援サイトとの連携が可能になれば，更に有益なシステムを構築することが可能である．就職支援サイトなどは学生が利用する際に学生情報の入力を求めてくることが殆どである．利用する Web サイトがひとつの場合は一度の入力のみだが，個人情報の入力と学生情報の入力の両方を行うのはそれなりの時間と手間が必要である．また情報収集のために複数の Web サイトを利用する場合，各 Web サイトごとに個人情報と学生情報を入力しなければならず非常に手間が必要となる．OpenID により各連携が可能になれば，それらの負担を軽減することが可能である．

また，大学内で使用する OpenID は大学間連携のみでしか利用できないため，大学外でのアカウントとの関連付けや情報の引き渡しなどをどう行うかも今後の課題である．大学で発行する OpenID は，卒業後は学内のポータルサイトなどを利用する権限がなくなるため自然と利用不可能となる．そのため，在学中の情報と在学前後の情報の管理をどう行うかが今後の課題となってくる．

参考文献

[1] OpenID Foundation Japan, “ OpenID ファウンデーションジャパン ”, http://www.openid.or.jp/.

[2] OpenID Authentication 2.0 - Final, “ OpenID Authentication 2.0 - Final ”, http://openid.net/specs/openid-authentication-2_0.html.

[3] JanRain, “ JanRain ”, http://www.janrain.com/.