タスク管理を行う Single Sign-On システム

田中美里
Misato TANAKA

1　はじめに

現在，本研究室では様々なイントラサービスを提供している．続々と新しいサービスが開始される一方で，これらのサービスのアカウントは個別に管理されたままである．そのため，ユーザはそれぞれのサービスに対し，異なるパスワードを使い分けなければならない．

また，研究室側には，ユーザは割り振られたタスクを完了してからイントラサービスを利用して欲しいというニーズがある．特に本研究室ではアンケートや予定確認のメールへの返信など，軽微なタスクが割り当てられる機会が多い．これらは「やればすぐ出来る」作業であるにも関らず，研究とは直接関係がないことから，多くの場合，後回しにされてしまう．しかし，タスクが処理されないことは担当者の負担となり，研究室を運用する上での支障となってしまう．この問題を解決するためには，学生らに対し，一度だけでなく繰り返しタスクを提示し続ける仕組みが必要である．

以上のことから，まずユーザが一つのアカウントで全てのイントラサービスにアクセスできる SSO（Single Sign-On : シングルサインオン）システムを実装する．そして，認証を行うユーザにタスクの処理を促す機能を付加させたものとして，タスク管理 SSO システム ISDL Auth を提案する．

2　ISDL Auth

2.1　概要

本システムの目的は，2.2 節にて述べる SSO 機能によってユーザがイントラサービスにログインする際の負荷を軽減させると同時に，各ユーザのタスク処理を促すことで，組織全体の運営の効率化を図ることにある．

このシステムの流れを，Fig. 1 に示す．

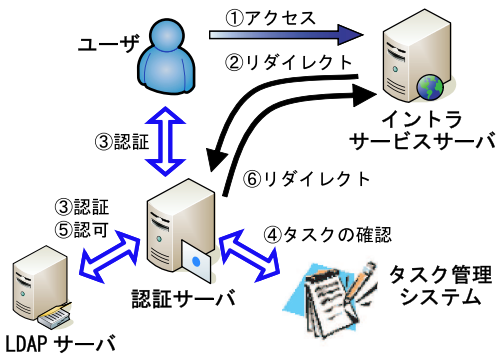

Fig.1　タスク管理認証システム (出典 : 自作)

新たにイントラサービスにアクセスしたユーザは，まず認証サーバにリダイレクトされる．そこで各イントラサービスに対する認証を受け，同時にタスクの確認が行われる．タスクの確認後，ユーザはイントラサービスに再びリダイレクトされ，その後は通常通りサービスを利用することができる．なお，認証とタスク管理の各機能の詳細については 2.2 節，2.3 節にて述べる．

2.2　認証機能

本システムでは SSO を実現している．SSO 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ソフトウェア，サービスへのアクセスが可能となるアクセス制御機能のことである．

Fig. 1 に示すように，本システムでは従来アプリケーション毎に行われる認証が，一箇所の認証サーバで行われている．ユーザはこの認証サーバで一度認証されれば，ウェブブラウザを閉じるまで，他のアプリケーションに対して認証を行う必要がない．

本システムにおける SSO の実現には，SSO 用の認証サーバの構築と，認証 API に問い合わせるモジュールをイントラサービス側に組み込む必要がある．ユーザが認証される際の，認証モジュールと認証サーバの挙動を Fig. 2，Fig. 3 のフローチャートに示す．

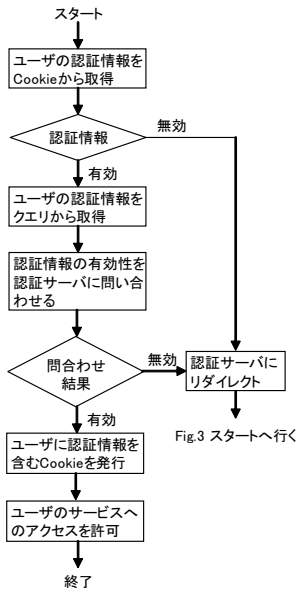

Fig.2　認証モジュール　フローチャート (出典 : 自作)

2.3　タスク管理機能

認証サーバにリダイレクトされたユーザに，新着のタスク，または期限が切れたにも関らず未完了のタスクが存在する場合，タスク管理画面が表示される．そこでユーザはタスクを確認した後，イントラサービスへと戻ることになる．

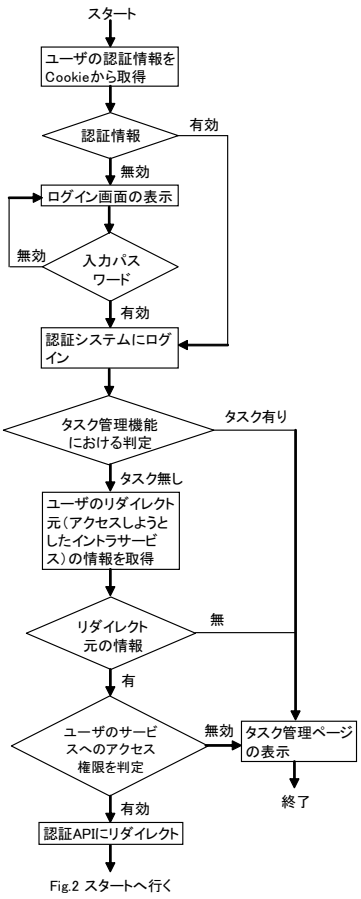

Fig.3　認証サーバ　フローチャート (出典 : 自作)

なお，イントラサービスにログインした後も，一定の期間をおいてユーザは認証サーバへとリダイレクトされ，タスクがチェックされる．従って，以下の 3 つの場合においてユーザはタスクの確認を行うこととなる．

- 認証サーバへのログイン時
- 新しいイントラサービスへのアクセス時
- 一定期間ごとに行われるリダイレクト時

一定期間毎にリダイレクトを行うのは，タスクを確認するためである．期限切れのタスクがあると，リダイレクトの度にタスク確認ページが表示される．これによって，ユーザに対してタスクを処理するよう促す．

タスクの通知だけであれば，デスクトップアプリケーションでも実装は可能である．しかし，未完了のタスクを何度も通知し続けるアプリケーションを，普通，ユーザは常駐させない．その点，このシステムにおいては，ユーザの任意でタスク通知のイベントを回避できない．従って，イントラサービスを利用する以上，タスクの確認だけは必ず行われることになる．

3　現在の実装状況

3.1　システム

現在，ISDL Auth の認証モジュールには PHP 版のみ存在し，近日中に β 版をリリースする予定である．今後は各プログラミング言語に対応したモジュールを提供していく．

認証サーバには現在，タスク管理機能は用いられていない．しかし，API を組み込んだシステムさえあれば，SSO の機能を利用することはできる．なお，ユーザ認証情報やイントラサービスの情報を格納するバックエンドには，LDAP サーバを利用している．

3.2　本システムを用いたイントラサービスの開発手法

イントラサービスの開発者が本システムを利用する場合の手順は，以下の通りである．サービス情報の登録は専用の登録ページを利用するが，ユーザの登録についてはフリーの LDAP クライアントソフトウェアを利用する予定である．

1. ISDL Auth システムにサービス情報を登録する
   - アプリケーション名
   - トップページの URL
2. ISDL Auth システムが発行した ID を，認証モジュールの設定ファイルに書き込む
3. サービスに認証モジュールを組み込む
4. サービスへのアクセスを許可するユーザを登録する

4　今後の展望

4.1　認証機能

本システムは，今後，Google Apps（Google Apps for Your Domain）と SSO による連携を行う．Google Apps は Google が提供するホスティングサービスである．

Google Apps はそれ自体に，ローカルのユーザアカウントを認証に用いる SSO 機能が備わっているため [1]，その機能を本システムに組み込むことによって，同一アカウントでのアクセスが可能となる．

4.2　タスク管理機能

本システムではユーザに対し，SSO とタスクの提示という機能を提供するだけのものである．ユーザが実際にそのタスクを片付けているのかどうかを判断することはできない．従って，ユーザが実際にタスクを完了していなくても，リダイレクトを回避するために，タスクを処理したと虚偽の申告をすることも可能である．

ユーザに割り振られるタスクについては様々なものが想定される．そのため，システムによって虚偽申告を起こさない仕組みを用意することは難しいと考えられる．よって，どのような対策を採るか，また，本システムでそれを実現すべきかを検討する必要がある．

また，本研究室で研究されているマッピング型タスク管理システム ZoomToDo との連携も，今後の検討課題として挙げられている．

参考文献

1) Google apps saml-based single sign-on `http://code.google.com/apis/apps/sso/saml_reference_implementation.htm%l`.