シングルサインオンシステムの導入における課題

ヤマトシステム開発株式会社

■ 執筆者Ｐｒｏｆｉｌｅ ■

永　住　健　太

2003 年　ヤマトシステム開発（株）入社
技術研究部配属
2006 年　現在 システム技術グループ所属
アイデンティティ管理システム調査，
シングルサインオンシステム開発・
運用担当

■ 論文要旨 ■

当社では，シングルサインオンの調査・研究と，検証のための開発を行ってきた．当初は検証目的で，社内の主な 11 システムのうち，２システムを対象にシングルサインオンを適用する予定だったが，社内システムのログイン統合について，社内の要望が高かったため，11 システムすべてを対象にすることになった．

テストリリースの際，『社内システムのシングルサインオン対応方法』『ユーザビリティとセキュリティのバランス』が課題になったが，非環境依存のモジュールの開発や，パスワードの管理方法を運用面/機能面で工夫し，問題の解決を図った．

結果，11 システムのシングルサインオン適用を成功させ，非環境依存のモジュールの開発により，修正モジュールの配布が容易になるなどの効果もあった．今後は，社内システムのログインを完全に一本化するフェーズに向けて，よりクライアントの環境に依存しない，可用性の高いシステムを目指していく必要がある．

■ 論文目次 ■

■ 図表一覧 ■

１．はじめに

１．１　当社の概要

当社はヤマトグループの中で，『e-ビジネス事業』を担当する情報サービス会社である．ヤマトグループの情報サービス業務を担当するだけでなく，『情報』，『通信』，『物流』という強みを活かして，トータルソリューションを提供するシステム・インテグレータである．

私の所属するシステム技術グループは，社内各部署の新サービスに不可欠な新技術の調査・研究やその導入を主な業務としている．最近では，RFIDを利用した物品トレースシステムの構築や，VMWareを利用した仮想マシン技術の検証，携帯電話OS Symbianでのアプリケーション開発方法調査，システムのユーザーIDやそれに関連する情報のライフサイクルを厳密に管理する『アイデンティティ管理システム』の調査や，複数のシステムの認証を統合する『シングルサインオンシステム』の開発・運用などを行っている．

１．２　システム導入の背景

近年，個人情報保護法の施行や相次ぐセキュリティ事故の影響により，IT 業界に限らず，多くの企業でセキュリティに対する意識が高まった．そのため，ユーザーID やアクセスログの管理が重要視されるようになり，技術的には 1990 年代後半から広まり始めた『シングルサインオン』が改めて注目されている．

当社では，2005 年からシングルサインオン(以下，SSO と略す)の調査・研究と，検証のための開発を行ってきた．当初は検証の最終段階として，社内の主な 11 システムのうち２システムに開発したモジュールを適用する予定だったが，調査の段階で，社員や各社内システムの管理者の，ログイン統合への要望が高いことが明らかになった．この要望を受けて，ユーザビリティとセキュリティレベルの向上を目的に，11 システムすべてに SSO を適用し，運用することになった．

本稿では，SSO の導入において一般的に発生し得る課題と，今回の事例でその課題に対してどのような対策を講じたかについて論じる．

２．システムの概要

SSO は，ユーザーが一度認証情報を入力するだけで，複数のシステムやサービスを利用できる仕組みである．例えば，人事システム，日報システム，メール用グループウェアなどの複数のシステムが存在する場合は，システムごとにユーザーID とパスワードの入力が必要になるが，SSO を利用すると，何れかのシステムにログインしていれば，他のシステムはログインが不要になるという仕組みである．

現在当社では，従来の各社内システムのログイン画面のほかに，鍵マークのリンクを用意し，SSO システムを使用してログインする機能を公開している．**図１**は SSO システム導入前の各社内システムのログイン方法，**図２**は現在の当社の SSO システムで人事システム，日報システム，Web メールの順でアクセスする際の使用方法のイメージである．

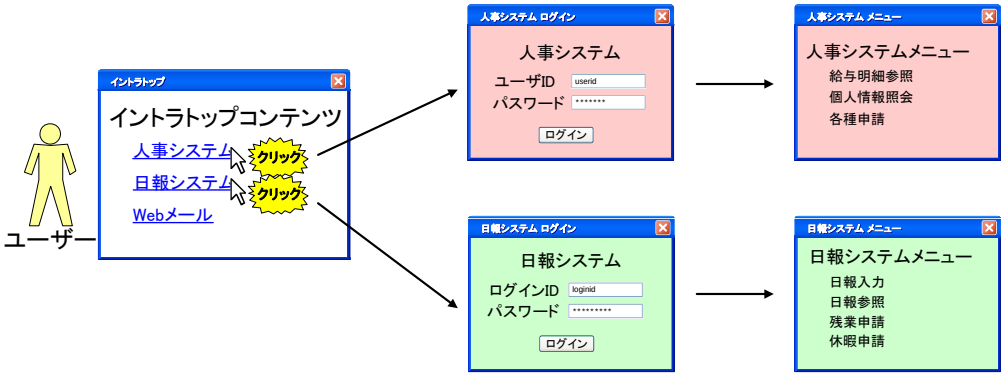

図１　SS0 システム導入前の社内システムの利用方法

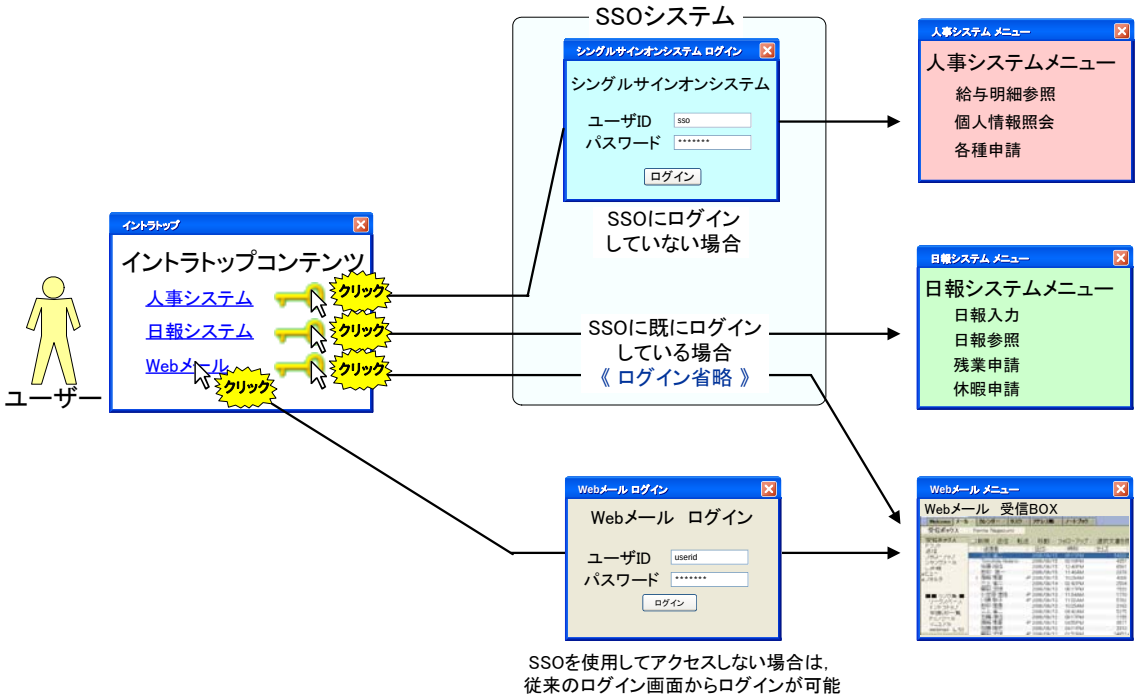

図２　現在のSS0システムの利用方法

なお，一般的に SS0 の導入におけるフェーズは，以下のようになると考えられる．

１）　適用対象となる社内システムの決定
２）　ユーザーの各社内システム使用状況調査
３）　製品の選定（もしくは開発）
４）　ユーザーや適用システムを限定したリリース
５）　課題点の洗い出し，対応
６）　本リリース（ログインを完全に SS0 に一本化）
７）　運用

当社の SS0 システムは現在，５から６の段階に入っている．そのため，次のステップとして社内システムのログインを完全に一本化するために，**図２**のような方法を用いて検証を行い，課題点の洗い出しや対応を行っている．

３．システム導入における２つの課題

SSO システム導入に際して，一般的に大きな二つの課題がある．

３．１　社内システムをどのように SSO に対応させるか

SSOシステムの導入における１つ目の課題は，『対象となる社内システムをどのようにSSOに対応させるか』ということである．例えばある実装方法を採用した結果，いくつかの社内システムをSSOに対応させることができず，ログインが統合されたシステムとされていないシステムができてしまった場合，その数が多ければ多いほど，ユーザーはSSOシステムを使用するメリットを感じなくなってしまうからである．

そこで考えなければならないのが，実際にどのような方法でSSOシステムを実装するかということである．一般的にSSOシステムには，『エージェント型』と『プロキシ型』と呼ばれる二つの実装方式がある．表１は，エージェント型とプロキシ型の特徴，**図３**と**図４**は構成のイメージである．

表１　エージェント型とプロキシ型の比較

	社内ネットワーク構成の変更	専用エージェントの配備	認証サーバに求められるスペック
エージェント型	不要	必要	低い
プロキシ型	必要	不要	高い

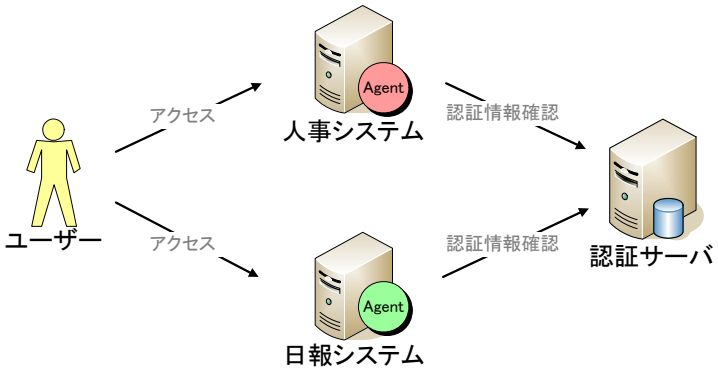

図３　エージェント型の構成イメージ

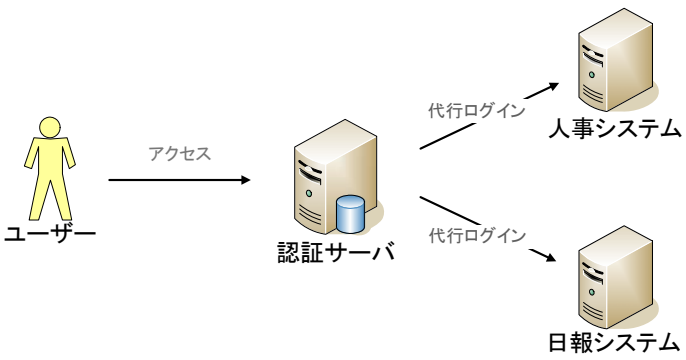

図４　プロキシ型の構成イメージ

一般的に，より多くのシステムに対応させるためには各社内システム専用のモジュールである『エージェント』を必要としない『プロキシ型』を採用することが多い．しかし，当社では既に多くのシステムが稼動しており，大幅なネットワーク構成の変更は難しいと判断し，今回はエージェント型を採用することにした．

しかしエージェント型では，例え市販のSSO製品を導入しても，専用のエージェントが用意されていないパッケージシステムや，自社で開発したシステムには対応できないという問題があった．なお，当社においては11システム中8システムが自社で開発したシステムであり，それらのシステムをどのようにSSOシステムに統合させるかが，特に課題になった．

３．２　ユーザビリティとセキュリティのバランス

SSOシステムの導入における２つ目の課題は，『ユーザビリティとセキュリティレベルのバランス』である．SSOシステムの導入の目的は，ユーザーのパスワード管理作業を簡略化してユーザビリティの向上を図ると同時に，ユーザーに対して厳重なパスワード管理を求め，システム側でもアカウント情報を厳しく管理し，セキュリティレベルの向上を図るというものだからである．

ただし，ログインの統合でユーザビリティを向上できるとしても，セキュリティレベルの向上という点で，パスワードポリシーやアカウント情報の管理方法などについて，具体的にどう実施するのかが課題であった．

４．課題に対する考察と対策

４．１　社内システムの SSO 対応方法

まず結果からいうと，社内システムの中でも，特に自社で開発したシステムを SSO システムに対応させるためには，エージェントの開発が必須となる．ただしエージェントを開発するにしても，いかに拡張性や保守性を維持するかということを意識し，課題の解決にあたる必要があると考えた．

４．１．１　オープン仕様の採用

エージェントの開発を行う場合，まずは認証サーバとエージェントの間でどのような情報がやりとりされているかを把握する，もしくは新たに定義する必要がある．

そこで今回は，SSO の標準化団体である『Liberty Alliance Project』の定めたオープンな仕様を利用することにした．このオープン仕様を利用することで，認証サーバもしくはエージェントだけをベンダ製に入れ替えるということも可能になると考え，将来的な拡張性も考慮した．

４．１．２　エージェントの開発

各システム専用のエージェントを開発する前に，自社で開発したシステムがどのような言語や環境で動作しているかを確認する必要があると考えた．表２はその調査結果の一部である．

表２　主な社内システムの環境

システム	OS	アプリケーション サーバ	開発言語
人事システム	Windows 2000	IIS	ASP
日報システム	Linux	apache	PHP
アクセス管理 システム	Linux	Tomcat	Java
Lotus Domino	Solaris	Domino 専用	パッケージ
サイボウズ office4	Windows 2000	IIS	パッケージ

結果，開発言語や環境がシステムごとに異なっており，それぞれに対応した専用のエージェントを開発する必要があることが判明した．

しかし，今後システムが増えることを考慮すると，その度にエージェントを開発するのは現実的ではないと考え，極力すべての社内システムで，エージェントの共通化を図ることにした．そのために，言語に依存しない『共通モジュール』と，どうしても言語に依存せざるを得ない各言語ごとの『エージェント』に分け，開発を行うことにした．

また，共通モジュールとエージェントの情報交換はHTTPで行い，言語の依存を極力排除するよう工夫した．**図５**は，認証サーバと情報をやりとりする各社内システムの構成例である．

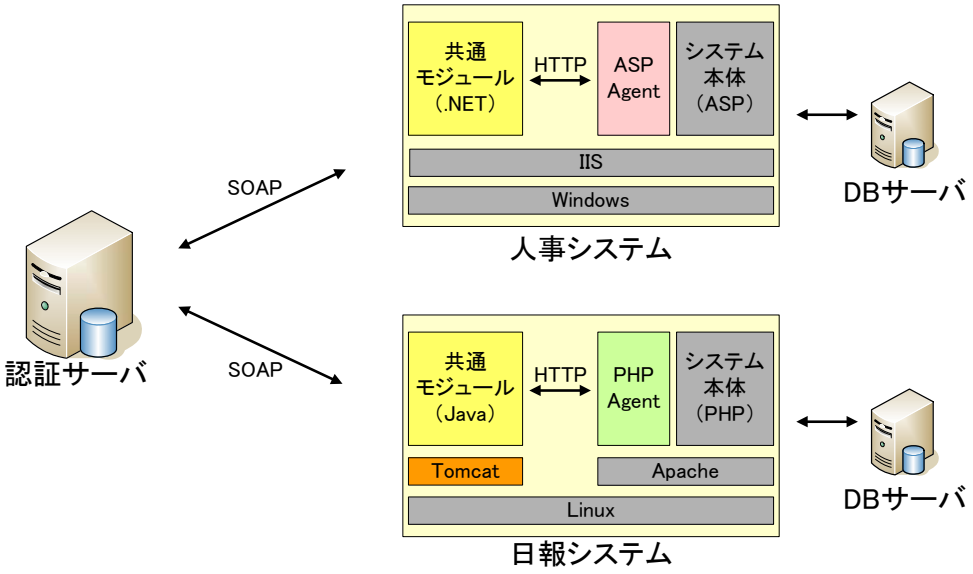

図５　社内システムの構成例

４．１．３　変更不可能なパッケージシステムの対応

前節で説明したようなモジュールの構成でエージェントを組み込む場合，システムの操作が可能なパッケージシステムであることが条件である．つまり，APIやソースが公開されていないパッケージシステムには，対応できないという問題がある．

そこで新たに，エージェントの代わりにユーザーのログインを代行する『エミュレータ』と呼ばれるモジュールを配置する方法を考案した．この方法では，エミュレータがユ

ーザーの代わりに各社内システムにログインし，ログイン済みの情報やレスポンスをユーザーに返すという仕組みを用いている．**図６**は，エミュレータを組み込んだ社内システムの構成，**図７**はエミュレータを介したログイン処理のフローである．

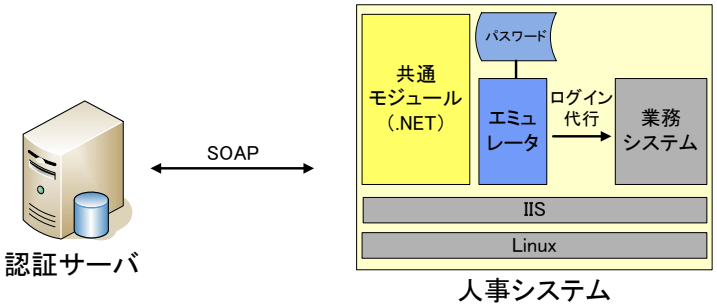

図６　エミュレータ方式の構成

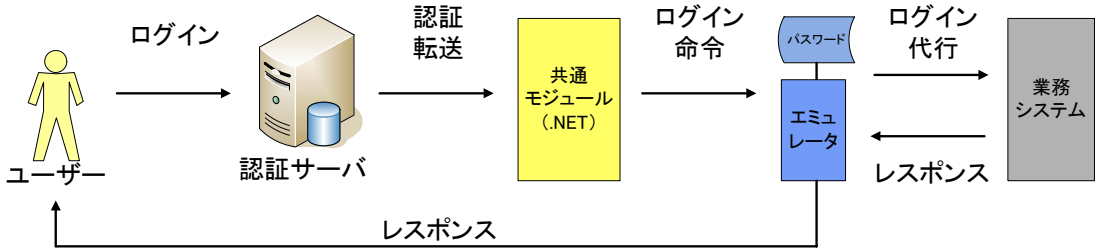

図７　エミュレータ方式の処理フロー

この実装方式では，ユーザーのパスワードが各社内システムの本体とエミュレータで二重で管理されることになるが，情報は同じサーバ上で管理されるため，セキュリティレベルが低下する心配はない．また，エージェント型と構成は変わらないため，エージェント型のメリットである『ネットワーク構成の変更が不要』『認証サーバのリソースが少なくて済む』という点を損なうことなく実装することができると考えた．

４．２　ユーザビリティとセキュリティの両立

まず，セキュリティレベルの向上には，以下の項目を検討する必要があると考えた．

１）適切なパスワードポリシーの設定

２）厳重なパスワード管理のための仕組み

３）アカウント不正使用の防止

以上のような観点から，次節以降示す具体的な策を講じた．

４．２．１　パスワードポリシーの検討

本来，すべての社内システムが一つのパスワードポリシーで統一されていれば，SSOシステムもそれと同じ設定にすべきである．しかし実際は，機能的にパスワードポリシーに対応できないパッケージシステムがあったり，あるシステムではより厳しいパスワードポリシーを設定したりしていて，すべてのシステムでパスワードポリシーを統一するのは難しい．

当社でもパスワードポリシーがシステムごとに異なっており，どのパスワードポリシー

のレベルに合わせれば，セキュリティとユーザビリティのバランスがとれるかという見極めが難しかった．

そこでまず，SSOシステム適用の対象である各社内システムのパスワードポリシーを把握することが必要であると考え，各社内システムの管理者に協力を依頼して，それぞれのシステムのパスワードポリシーの洗い出しを行った．**表３**はその結果の一部である．

表３　主な社内システムのパスワードポリシー

システム	パスワード通知方法	パスワード有効期限	最小文字数	文字列構成の制約	その他
人事システム	Web メール	最終ログインから 40 日	8 文字	英数字のみ使用可能	5 回連続して間違えた場合はロック
日報システム	Web メール 電話	なし	6 文字	英数字のみ使用可能	
アクセス管理システム	Web メール E メール	40 日	8 文字	英数字混在が必須	過去 8 回分のパスワードは使用不可
Lotus Domino	電話	なし	5 文字	なし	
サイボウズ office4	Web メール E メール	なし	1文字	なし	

SSOシステムが複数の社内システムのログインを統合するシステムであることを考えると，なるべくすべての社内システムのパスワードポリシーを超えるような，最高レベルのパスワードポリシーに設定することが望ましい．なぜならば，仮にSSOのパスワードポリシーを人事システムのパスワードポリシーより低く設定してしまうと，SSO経由でアクセスする人事システムのパスワードポリシー自体が低くなるのと同等だからである．

しかし，単に厳しいだけのポリシーでは，ユーザーの利便性を損ねてしまう．そこで，ユーザーのパスワードポリシーに関する意識を把握するために，SSOシステムのテストリリースの第一段階で，設定したポリシーについて**図８**のアンケートを行った．

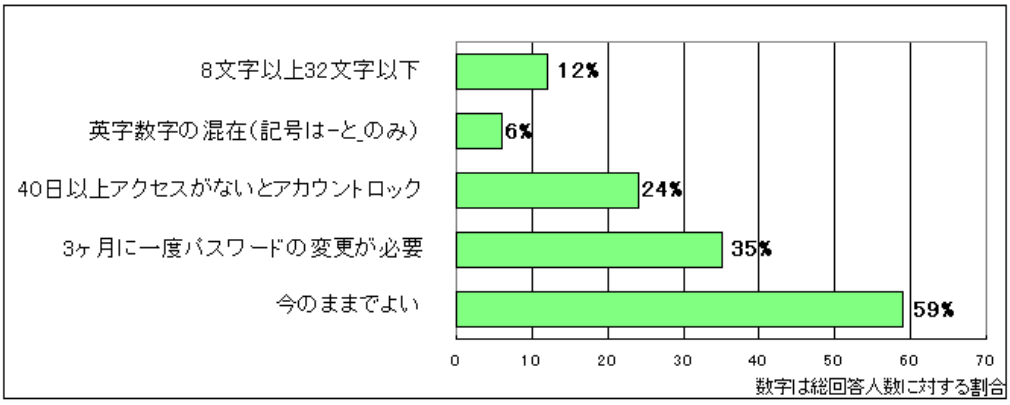

図８　アンケート『厳しいと思うパスワードポリシー』の回答

このアンケートの結果，パスワードの文字列に対する制約よりも，使っているパスワードを定期的に変更しなければならない制約に対して，ユーザーが煩わしさを感じていることが分かった．この事実を考慮し，これ以上期限に関するポリシーを厳しくするのは難しいと判断した．また，アクセス管理システムの『有効期限40日』というポリシーは，人事システムの『最終ログインから40日』という制約と，新たに設定する『有効期限3ヵ月』という制約でカバーできると考え，現段階でのパスワードポリシーを表４のように設定した．

表４　SSOシステムのパスワードポリシー

システム	パスワード 初期化方法	パスワード 有効期限	最小 文字数	文字列 構成の制約	その他
SSOシステム	Web メール E メール	基本 90日 アクセスのない 状態で40日	8文字	英数字混在 が必須	過去数回分の パスワードは 使用不可

『その他』の項目で『過去数回分のパスワードは使用不可』と具体的な回数を非公開にしたのは，回数を公開してしまうことで，一時的にパスワード変更を繰り返し，履歴をすべて削除するような操作を少しでも防ぐための工夫である．

４．２．２　管理者でも変更できないパスワード

一般的な業務システムでは，管理者機能でユーザーのパスワードを変更ができるものがある．しかし，今回のSSOシステムでは，よりセキュリティを意識した仕組みにするため，管理者すらパスワードを変更できないように，あえてそのような機能は設けなかった．

また，データベースのパスワードを『MD5』という復号化が不可能なアルゴリズムで暗号化し，仮に直接データベースにアクセスされたとしても，操作や閲覧などはできないようにした．そのため認証処理では，ユーザーの入力したパスワードを暗号化して，データベースの暗号化されたパスワードと比較するという手順を用いた．

４．２．３　通信の安全性

SSOシステムのパスワードポリシーをいくら厳しくしても，ネットワーク上を平文のパスワードが流れていては，まったく意味をなさない．そのため，認証サーバと各社内システム，ユーザー間の通信においても**図９**のようなセキュリティを意識した実装をした．

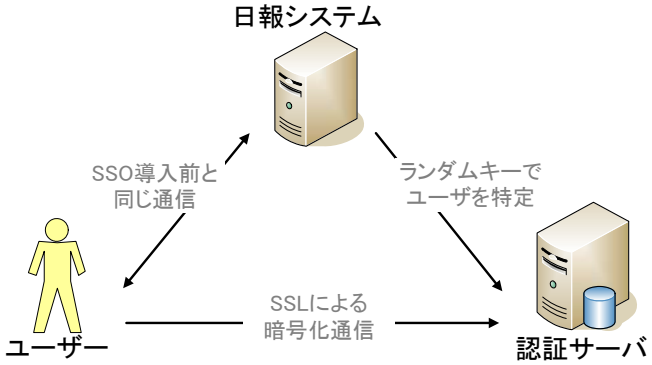

図９　SSOの通信（日報システムの例）

図９のポイントは，日報システム（各社内システム）と認証サーバとの通信である．一般的にSSOというと，認証サーバから各社内システムに対して，ユーザーIDやパスワードを送信するようなイメージを持たれる場合が多い．しかし，今回の実装方式ではユーザーの特定には，キーとなるランダムな文字列を用いている．これにより，通信内容だけではユーザーの特定や，パスワードの取得はできないような仕組みにし，SSO導入前よりも安全な通信ができるようにした．

４．２．４　アクセスログの閲覧機能

今回のSSOシステムでは，ユーザーが自分のアクセスログを閲覧できるような仕組みを設けた．この仕組みでアカウントの不正使用を，ユーザー自身が警戒することができるとともに，アクセスログが取られているという認識を与えることで，不正アクセスの防止にもなると考えた．

４．２．５　不要アカウントの自動削除機能

不正アクセスの温床になる不要アカウントを削除するため，人事部の管理する社員マスタとSSOシステムの社員マスタを連携させ，日次で社員情報を更新する仕組みを設けた．この方法で，不要アカウントの削除と同時に，新たに入社した社員のアカウントを追加することもできるため，私達SSOシステム管理者の作業も軽減できると考えた．

５．評価

５．１　社内システムの SSO 対応方法について

今回の対策によって，SSO システムの適用対象とした 11 システムを統合できたことの他に，以下のような点でメリットを見出すことができた．

- 最近 Liberty Alliance の仕様に対応した製品が市場に出回るようになり，開発したエージェントを活用しつつ，今後の製品選択の幅が広がった．
- 各社内システムのエージェントを共通化することで，配布する修正モジュールも共通化され，メンテナンス性の向上につながった．
- ほぼ一通りの言語依存エージェントが揃ったため，今後システムが増えた際にも，新たな開発なしに，SSO システムへの対応ができるようになった．
- エミュレート方式の考案により，API やソースが公開されていないシステムにも簡単に対応できるようになった．
- エミュレータ方式は，既存の社内システムの修正を必要としないため，重要度の高い社内システムを SSO に対応させる際のリスクを軽減することができた．

これらの結果から，各社内システムを SSO に対応させる今回の対策は，非常に有効な手段であったと評価している．

５．２　ユーザビリティとセキュリティの両立について

ユーザビリティとパスワードポリシーについて，ユーザーの意識を把握するために**図 10**, **図 11** のような二つのアンケートを行った．

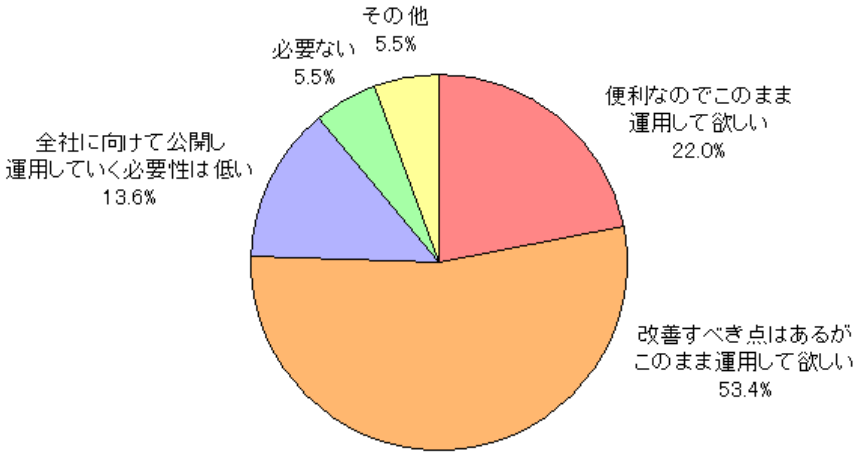

図 10　アンケート『SS0 システムの有効性について』の回答

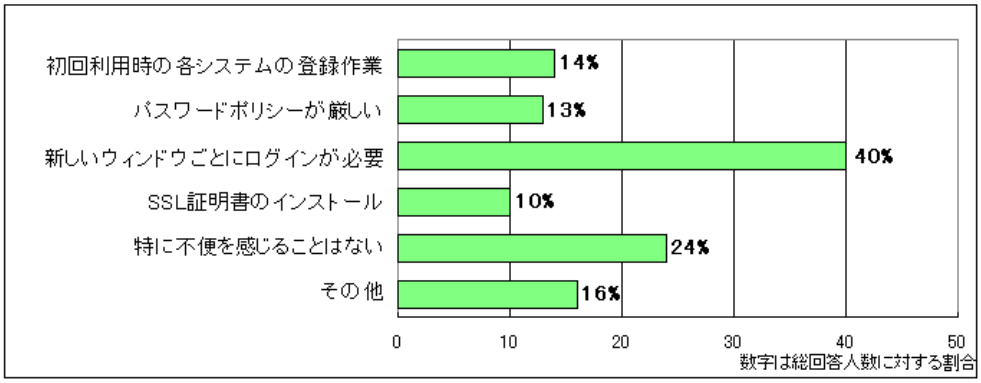

図 11　アンケート『SS0 システムの不便なところ』の回答

図 10 の結果を見ると，『便利なのでこのまま運用して欲しい』『改善すべき点はあるがこのまま運用して欲しい』という意見の合計が約 75%を占めており，多くのユーザーが SS0 システムの利便性を評価している．また，『SS0 システムがないと不便で困る』『使ってみたら，非常に便利だった』とコメントを記入したユーザーもいた．

次に，**図 11** の結果を見ると，SS0 システムに対する不満の中で，パスワードポリシーに関するものは 13%に留まっており，現行の社内システムの中では厳しいパスワードポリシーに設定したのにも関わらず，87%のユーザーが不満を感じていないことが分かった．また，アカウント情報が不正に使用された形跡や報告もなく，『SS0 システムにセキュリティ上の不安を感じる』という声もない．

このことから，今回 SS0 システムに講じたセキュリティ向上のための対策は，ユーザーの利便性を損ねることなく，ユーザビリティとセキュリティのバランスをとることができたと感じている．

６．今後の課題

６．１　環境依存性の排除

今回の SSO システムは特に『クライアント環境に依存しない構成』を意識して開発した．結果，バージョンや種類の異なるブラウザでも，正常な動作を確認することができた．しかし，ある特定のクライアント環境では，認証処理が正常に動作しないという現象が報告されている．

ログインを統合する SSO システムでは，『SSO システムを使うことのできないユーザー』は『すべての社内システムを使うことのできないユーザー』となってしまい，大きな問題となる．今後は，同様の現象が発生している環境の把握と，より環境に依存しない実装方法の追及をしていく必要がある．

６．２　可用性の向上

現在の SSO システムでは，当初の研究用サーバを継続して使用しており，冗長化構成を全くとっていない．すべての社内システムのログインを統合することを考えると，SSO システムは『絶対に止まってはならないシステム』になる．その点を考慮して，今後はより可用性の高いシステム構成にしていかなければならない．

７．おわりに

冒頭にも述べたとおり，SSO の仕組み自体は 1990 年代後半から既にあり，その頃から有効性が謳われている．しかし，費用対効果が見えにくいことや，少なからず現行の社内システムの修正が必要なこともあり，『多くの企業で広く普及している』とはいえない状況である．ただし，市場のセキュリティ意識はより高まる傾向にあり，SSO がこれからますます注目されることは間違いない．

当社では今後，実際に社内システムのログインを完全に一本化し，運用フェーズに入っていく．そこで新たな問題や課題が発生することも考えられるが，SSO システム導入の目的である『ユーザビリティとセキュリティのバランス』を常に意識し，解決を図っていきたい．また，導入後の効果を定量的に評価し，今後の機能向上に役立てたいと考えている．