社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005—IS—92（2）
2005／5／27

シングルサインオンエキスパートシステムの検討

菊地 克朗

(株)日立製作所 中央研究所

要旨:

企業内に散在する情報システムを統合し一元的な利用を実現することにより、ユーザの業務効率向上を図る企業情報ポータルにおけるシングルサインオン実現の容易化について検討した。業務システムのログイン通信を解析し、最適なログイン代行モジュールを自動的に選択することでシングルサインオンを実現するシングルサインオンエキスパートシステムを提案する。ログイン処理を定式化し、ログイン代行モジュールを自動的に適用可能とすることで HTTP、HTML などの下位レイヤの知識を必要とせずに業務システムへのシングルサインオンを実現可能とする。これにより、低コストかつ迅速に構築可能なシングルサインオンソリューション実現の見通しを得た。

Study on Expert System for Single Sign-On Solution

Katsuro Kikuchi

Central Research Lab., Hitachi, Ltd.

Abstract:

The Enterprise Information Portal (EIP) is a Web service that serves as a single gateway to Enterprise Information Systems (EIS). A Single Sign-On (SSO) is one of the key solutions for the EIP.

In this paper, we propose the Expert System for the SSO Solution. We have formulated typical login processes of Web-based applications; furthermore, we have prepared the Login Module corresponding to formulated data. The Expert System analyzes the login process of the EIS, and automatically finds a Login Module by pattern matching. System Engineers can easily build the SSO feature without the knowledge of detailed technologies such as HTTP/HTML. As a result, we get the prospect of realizing the SSO solution built with low costs realizing.

1. はじめに

企業情報ポータル(EIP:Enterprise Information Portal)は、企業情報システム(EIS:Enterprise Information System)を統合することにより、利用者のシステム利用効率を改善し、IT システム全体の ROI(Return On Investment)を向上させる。EIP の重要な機能のひとつにシングルサインオン(SSO:Single Sign-On)機能がある。EIP における SSO 機能とは、『ユーザがポータルに認証をうける(ログインする)だけで、ポータルに統合されている企業情報システムの機能を利用できるようになる認証方式』である。

一方、多くの EIS は Web アプリケーションとして構築されている。Web アプリーションを実現するための HTTP や HTML は、元々ドキュメントの公開･閲覧システムを目的として開発されたため、アプリケーション基盤としては機能不足と言える。そのためユーザ認証機能やセッション管理機能といったアプリケーションを構築する上での基盤機能は、アプリケーション毎に独自に作りこんでいることが多い。近年、Sun Microsystems 社の提唱する J2EE や Microsoft 社が提供する.Net Framework など Web アプリケーションフレームワークが整備され、ユーザ認証、セッション管理方式はある程度集約されつつあるが、一本化されるには至っていない。

このため、EIP において SSO を実現するには、EIS で用いられているユーザ認証機能やセッシ

ョン管理機能について、それぞれ SSO 方式を検討する必要ある。Web アプリケーションで一般的に使用される単純な Form 認証や Basic 認証については、報告者が SSO 実現方法を提案している[1]。

報告者は、文献[1]で様々な SSO 方式を提案しているが、実際にシステムエンジニア(以下、SE)が EIS に対して SSO を実現しようとした場合、(i)『EIS の認証方式、セッション維持方式の解析』、(ii)『様々な SSO 方式から適用可能な方式を選択』といった手順が必要となる。手順(i)を実現するには、Web ブラウザと Web アプリケーションで交わされる認証データやセッション維持情報の転送方法を解析する必要がある。また、手順(i)で解析した Web アプリケーションの認証方式やセッション維持方式から適用可能な SSO 方式を選択する場合(ii)には HTTP や HTML 仕様を熟知しておくことが不可欠である。上記のように、SSO を実現するには、通信ログの解析など地道な作業が必要であり、また、SE が SSO を実現するには、業務ロジックに関する知識、上位レイヤの情報技術の知識以外に、Web アプリケーションの基盤技術である HTTP、HTML などの下位レイヤの深い知識が要求されるといえる。このことから、シングルサイオンの実現は SE のノウハウ(暗黙知)の固まりとなっている。

そこで、容易に SSO を実現すべく以下を目的としたシングルサインオンエキスパートシステムの検討を行った。

(1) 解析など Web アプリケーションの詳細を SE に意識させることなく容易に SSO を実現可能とする。
(2) 未知の認証、セッション維持方式にも対応できるよう拡張性を考慮した SSO 容易化システムを検討する。

なお、SSO を実現する場合、EIS を改変することにより実現する方式も考えられるが改変にはそれなりのコストが必要となる。そこで、本稿では既存 EIS に手を入れることなく SSO を実現する方式について言及する。

2. シングルサインオン実現の一例

代表的なユーザ認証方式である Basic 認証および Form 認証について文献[1]で提案した SSO 手法を説明する。

2.1 EIS が Form 認証系の場合

Form 認証の場合に SSO を実現するためには、基本的には『エンドユーザとの窓口であるポータルサーバが EIS のセッション維持情報を取得し、その情報をブラウザに引渡し、その情報を付加して EIS にアクセスさせる』必要がある。Form認証に対する SSO 方式として、文献[1]では以下に示すポータル代行ログイン方式および直接ログイン方式を提案している。

(1) ポータル代行ログイン方式

ポータル代行ログイン方式の処理概要を図 2-1に示す。本方式では、『(i)ポータルサーバにアドインされ EIS への代行ログインを実現するログインモジュールがブラウザから SSO の要求を受け取り、ブラウザのポータルセッションすなわちログインユーザに対応する EIS の認証情報(主にユーザ ID、パスワード)を EIS 認証情報テーブルから取得、(ii)ログインモジュールが EIS へ代行ログイン、(iii) EIS のログイン応答に含まれるセッション ID を取得、(iv)ブラウザに前記ステップで取得したセッション ID を引き継がせると共に EIS に直接アクセスするよう応答を返却、(v)応答を受け取ったブラウザはセッション ID と共に EIS にアクセス』のステップにより SSO を実現する。セッション ID の引継ぎは EIS のセッション維持方法が Cookie を利用している場合には Cookie の domain 指定を利用してポータルサーバと EIS 間で Cookie を共有させることで実現する。また、URL 書き換えの場合は(iv)のブラウザから EIS へのアクセス URL 中にセッション ID を埋め込むことにより実現する。

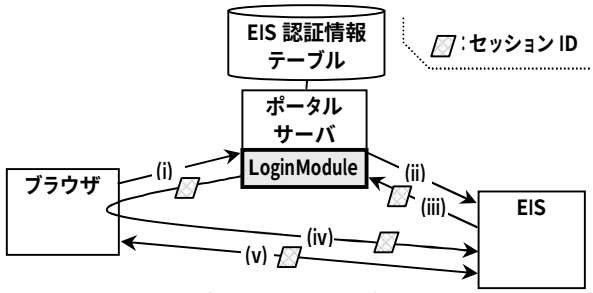

図 2-1 ポータル代行ログイン

(2) 直接ログイン方式

直接ログイン方式の処理概要を図 2-2に示す。直接ログイン方式ではログインモジュール

が EIS へのログイン代行は行わず、ブラウザに対して EIS へのログイン命令を発行して SSO を実現する。具体的には、『(i)ログインモジュールはブラウザから SSO の要求を受け取り、ブラウザのポータルセッションに対応する EIS の認証情報を EIS 認証情報テーブルから取得、(ii)前記ステップで取得した認証情報を HTML Form の hidden フィールドに埋め込み、さらに Form を自動的に EIS にサブミットするようスクリプトを埋め込んだ HTML をブラウザに応答(iii)ブラウザはスクリプトに従い EIS に Form をサブミットしてログイン(iv)ログイン完了応答』のステップにより SSO を実現する。

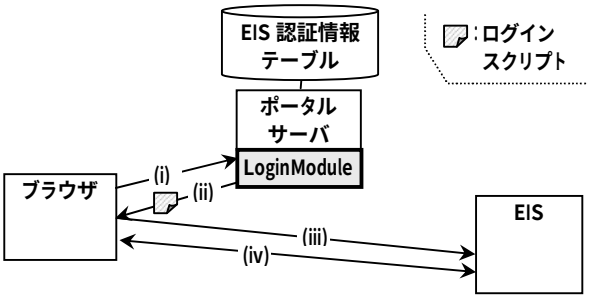

図 2-2 直接ログイン

2.2 EIS が Basic 認証の場合

Basic 認証では、全てのリクエストにユーザ ID、パスワードを付加する必要があるが、通常は、ブラウザがユーザからの入力を受け付けてブラウザ内部に保存し、サーバアクセス時に自動的にユーザ ID、パスワードを付加することにより実現されている。また、ブラウザの外部からユーザ ID、パスワードをブラウザに保存させることは基本的に不可能である。従って、Basic 認証の SSO を実現するには、ブラウザから EIS へのリクエストをトラップして、Basic 認証用ヘッダを付加する必要がある。文献[1]では、認証情報を付加する位置に応じて表 2-1示す2種類の SSO 方式を提案している。

表 2-1 Basic 認証でのシングルサインオン方式

方式	説明
SSO プロキシ	ブラウザと EIS の間にベーシック認証用ヘッダ情報を付加するプロキシサーバを設置して、SSO を実現。
エージェント	EIS を構成する Web サーバの拡張機能を利用して SSO を実現。具体的には、ブラウザのリクエストが EIS のプログラムに渡る前に Web サーバ内でリクエストをトラップしベーシック認証用ヘッダ情報を付加することにより SSO を実現。

3. シングルサインオン適用における問題点

前述のように EIS の認証、セッション維持方式に対応したログインモジュールを開発すれば SSO が実現可能である。実際には、ブラウザと EIS の間の HTTP の通信や HTML の内容を解析し、どのような認証処理、セッション維持を行っているか調査する必要がある。更に、調査した認証処理、セッション維持方式に対して SSO を行うためのプロトコルを考える必要がある。しかし、

(1) 『通信ログを見ても、どのような認証、セッション維持方式を用いているか分からない』
(2) 『通信ログから認証、セッション維持方式を特定できても、実際にどのように SSO を実現すれば良いのか分からない』

といった問題に直面することが多い。これは、SSO が、本来の SE 業務に必要な業務ロジックに関する知識や上位レイヤの情報技術とは異質の HTTP や HTML など下位レイヤの深い知識が要求されるためと言える。また、EIS がセキュリティ強化のために SSL(Secure Socket Layer)による通信路の暗号化を行っている場合、単に通信パケットを解析するだけでは認証、セッション維持方式を推定することは不可能となる。

4. シングルサインオンエキスパートシステムの提案

上記で示した 2 つの問題を解決し、SSO を容易に実現するために、以下のシングルサインオンソリューションを提案する。シングルサインオンソリューションは二つのシステムからなる。

第一のシステムは、SE のノウハウとなっている SSO 実現方式のナレッジ化を支援する SSO ナレッジ支援システム(図 4-1)、第二のシステムは SSO ナレッジ支援システムでナレッジ化された SSO 実現方式を容易に EIS に適用可能とする SSO エキスパートシステムである(図 4-2)。

SSO ナレッジ支援システムでは、SE が EIS

の認証方式やセッション維持方式を解析して開発したログインモジュールについて、ログインモジュールが適用可能な認証方式、セッション維持方式すなわちログインプロトコルの形式化を支援し、ログインモジュールを再適用しやすくするためのナレッジ化を実現する。これにより問題(2)の『認証、セッション維持方式を特定できても、実際にどのように SSO 技術を適用すれば良いのか分からない』を解決する。

一方、SSO エキスパートシステムは、SSO ナレッジ支援システムにより形式化されたログインプロトコルと EIS の実際のログインシーケンスを比較し、適用可能なログインモジュールを抽出することで、認証方式やセッション維持方式の詳細を解析することなく、容易に SSO を実現することを可能にする。これにより問題(1)の『通信ログを見るだけで、どのような認証、セッション維持方式を用いているか分からない』を解決する。

以下、本稿では SSO エキスパートシステムについて処理方式と機能の評価結果について報告する。SSO エキスパートシステムを実現するには、本来 SSO ナレッジ支援システムで構築する SSO ナレッジデータベースの整備が必要であるが、ログインモジュールと、それに対応する形式化されたログインプロトコルは手動で作成して利用するものとする。以降、Web アプリケーションのログイン方式の形式化、シングルサインオンエキスパートシステムの処理方式について説明する。

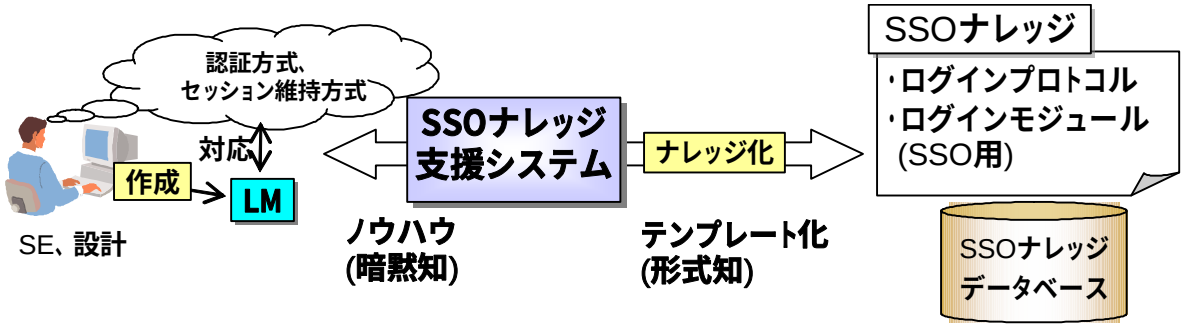

図 4-1 シングルサインオンのナレッジ化

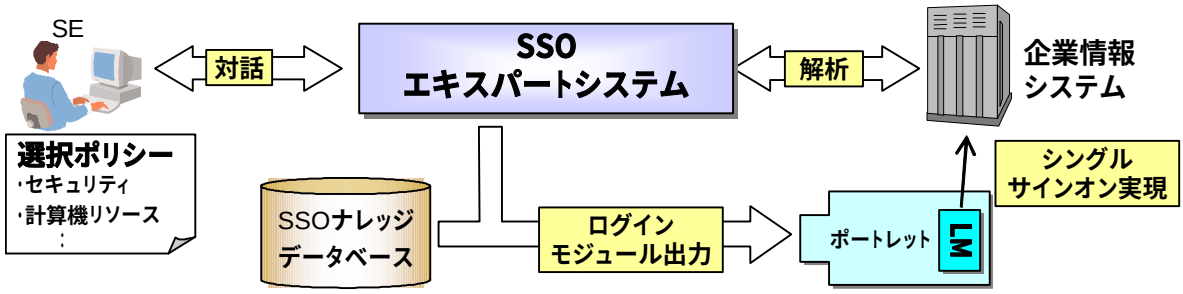

図 4-2 シングルサインオンの適用

5. Web アプリケーションのログイン方式の形式化

Web アプリケーションで一般的に利用される認証、セッション維持方式は Form 認証および Basic 認証をベースとした方式である。従って、手始めに Form 認証および Basic 認証をベースとした方式について SSO の検討を行った。Form 認証、Basic 認証方式の HTTP 通信を分析した結果、以下の特徴があることが分かった。

5.1 ログイン処理の通信パターン

Form 認証では、図 5-1のように認証情報を送るための Form のサブミットや、Cookie を用いたセッション維持のためのセッション ID の設定および送付といった特徴がある。また、セッション維持方式として一般的な Cookie ではなく URL 書き換えを行っている場合には、HTML 内のハイパーリンクについてセッション ID を示すパラメータが付加されており、結果としてリクエスト URL にセッション ID を示すパラメータが

付加された形式となる。Form 認証では、認証情報送付のための Form のサブミットおよびセッション維持情報の設定、送付のための Cookie（または URL）に特徴がある。

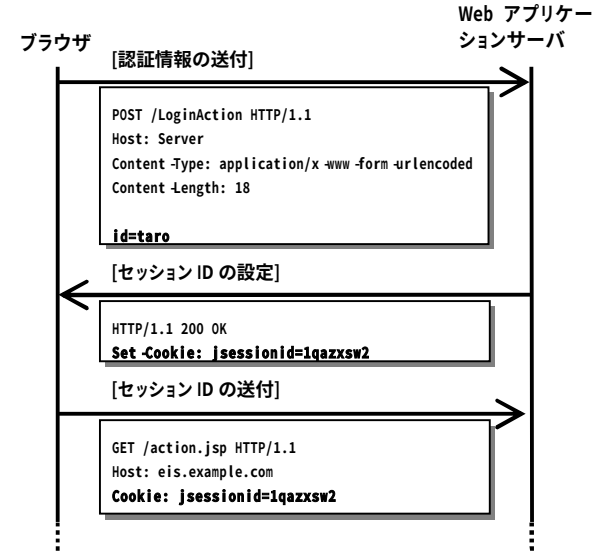

図 5-1 Form 認証、Cookie によるセッション維持の場合の通信例

一方、Basic 認証では、認証を要求するための Authenticate ヘッダ、認証情報を送付するための Authorization ヘッダが大きな特徴といえる。

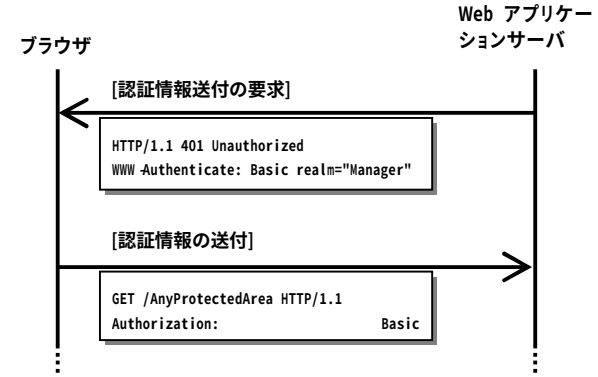

図 5-2 Basic 認証の場合の通信例

上記のように HTTP のヘッダやリクエスト URL に着目することで認証、セッション維持方式の推定が可能と結論付けられる。

ついで、ログイン処理を形式化する。具体的には、認証やセッション維持の際に現れる特徴的な HTTP リクエスト、レスポンスを抽出しパターン化する。パターン化は、図 5-3に示す XML 形式記述により行う。ログイン処理は、『ログイン画面の表示』および『ログイン情報の送信』の 2 つのステップからなるので、この2つのステップに対応して通信パターンを定義する。また、表 5-1にデータ形式の説明を示す。

```
<?xml version="1.0" encoding="UTF-8">
<pattern>
    <!-- ログイン画面の表示 -->
    <message>
        <request>
        <scheme type="${スキーム}"/>
            <method type="${メソッド}"/>
        <header name="${ヘッダ名}" value="${ヘッダ値}">
        </request>
        <response>
            <code value="${応答コード}"/>
        <header name="${ヘッダ名}" value="${ヘッダ値}">
    </response>
   </message>

    <!-- ログイン情報の送信 -->
    <message>
     <request>
        <scheme type="${スキーム}"/>
            <method type="${メソッド}"/>
        <header name="${ヘッダ名}" value="${ヘッダ値}">
    </request>
        <response>
            <code value="${応答コード}"/>
        <header name="${ヘッダ名}" value="${ヘッダ値}">
    </response>
  </message>
</pattern>
```

図 5-3 ログイン処理通信パターンデータ形式

表 5-1 ログイン処理通信パターンデータ形式説明

#	要素名	説明
1	pattern	通信パターン定義のためのルート要素。子要素として 2 つの message 要素を持つ。
2	message	一つのリクエストとレスポンスの対を定義する。1 つ目の message 要素は、ログイン画面表示時の通信パターンを、2 つ目の message 要素はログイン情報送信時の通信パターンを定義する。
3	request	リクエストの通信パターンを定義する。子要素として 0 個もしくは 1 個の scheme 要素、0 個もしくは 1 個の method 要素、0 個以上の header 要素を持つ。
4	response	レスポンスの通信パターンを定義する。子要素として 0 個もしくは 1 個の code 要素、0 個以上の header 要素を持つ。
5	scheme	通信スキームを定義する。type 属性に"http"または"https"を指定する。type 属性は必須。
6	method	通信メソッドを定義する。type 属性に"GET"または"POST"を指定する。type 属性は必須。
7	header	リクエストヘッダまたはレスポンスヘッダを定義する。name 属性でヘッダ名、value 属性でヘッダ値を指定する。name 属性は必須、value 属性は省略可能。
8	code	レスポンスコードを指定する。value 属性にレスポンスコードを指定する。value 属性は必須。

5.2 未知のログイン処理への対応

上記では、Form 認証と Basic 認証の基本的なパターンについてログイン処理通信の形式化について説明した。一方、EIS によっては上記のような基本的な認証方法を拡張して認証を実現している場合もある。例えば、『認証情報送付に先立って予めセッションを確立する』、『特定のページを経由して認証情報を送付する』などの拡張形がある。このような未知の認証、セッション維持方式に対して、HTTP のデータ構造に沿った通信パターンの定義を可能とするために、message 要素を複数列挙可能とした。これにより、未知のログイン処理方式に対しても通信パターンを柔軟に定義可能とした。

6. SSO エキスパートシステム処理方式

SSO エキスパートシステムは、『SSO 設定フェーズ』と『SSO 実行フェーズ』の 2 つのステップからなる。SSO 設定フェーズでは、SE と SSO の対象となる EIS の間に SSO エキスパートシステムが介在し、対象 EIS に最も適したログインモジュールを選択する。SSO 実行フェーズでは、SSO 設定フェーズで選択されたログインモジュールを用いて EIS への SSO を行う。以下、2つのフェーズについて順に処理方式を説明する。

6.1 SSO 設定フェーズ

SSO エキスパートシステムは、図 6-1に示すように、クライアントと EIS の間に介在し、EIS に対するログイン操作の通信ログを取得する HTTP/HTTPS プロキシ部および通信ログ取得部、取得した通信ログとログインモジュールの通信パターンを比較し、EIS に適用可能なログインモジュールを選択するログインモジュール選択部、EIS のアダプタの役割を担うポートレットから呼び出され実際に SSO を実行する SSO 実行部からなる。また、SSO を行うログインモジュールを格納したログインモジュール管理テーブル、EIS(ポートレット)と適用可能なログインモジュールの関連付けが格納されている EIS-SSO 対応テーブル、ポータルサーバのアカウントに対応した EIS のユーザ情報が格納されている EIS 認証情報テーブルを持つ。

以下、SSO 設定フェーズの処理の流れを図 6-1を用いて説明する。

(i) SSO エキスパートシステム経由で EIS のログイン操作を SE が行う。
(ii) (i)のログイン操作中にリクエスト URL、リクエストメソッド、リクエストヘッダ、リクエストボディ、レスポンスコード、レスポンスヘッダ、レスポンスボディを通信ログとして取得する。
(iii) (ii)で取得した通信ログとログインモジュール管理テーブルに格納されているそれぞれのログインモジュールについて適用通信パターンとのマッチングを取り、適用可能ログインモジュール候補を選択する。
(iv) ログインモジュールの設定として、ログインモジュールが EIS に認証情報として送付するユーザ ID、パスワードのフィールド名をポートレットのパラメータとして設定する。
(v) (iii)で選択した適用可能ログインモジュール一覧について、SSO 実行部により実際に SSO を試行する。
(vi) 試行の成否を SE に指定してもらい、成功した場合、EIS-SSO 対応テーブルに対応するログインモジュールを設定する。

6.2 SSO 実行フェーズ

次いで、SSO 実行フェーズの処理の流れを図 6-2を用いて説明する。

(i) SSO に必要な EIS ごとのユーザ ID/パスワードを設定する(初回のみ)
(ii) ポートレット中の SSO 用リンクをクリックし、ポータルサーバに SSO の要求を行う。
(iii) SSO の要求を受け取ったポートレットは SSO 実行部に SSO の実行を依頼する。SSO 実行部は呼出元ポートレットからログインすべき EIS の識別子を取得する。次いで EIS 識別子をキーに EIS-SSO 対応テーブルから、対応するログインモジュールを選択し、ログインモジュールをロードした後にログインモジュールに対して SSO の実行を依頼する。
(iv) SSO 実行の依頼を受けたログインモジュールは EIS ユーザ情報テーブルから EIS に対応するユーザ ID、パスワードを取得し、設定フェーズで指定された変数名で EIS に認証情報として送付し、EIS へのログインを行う。

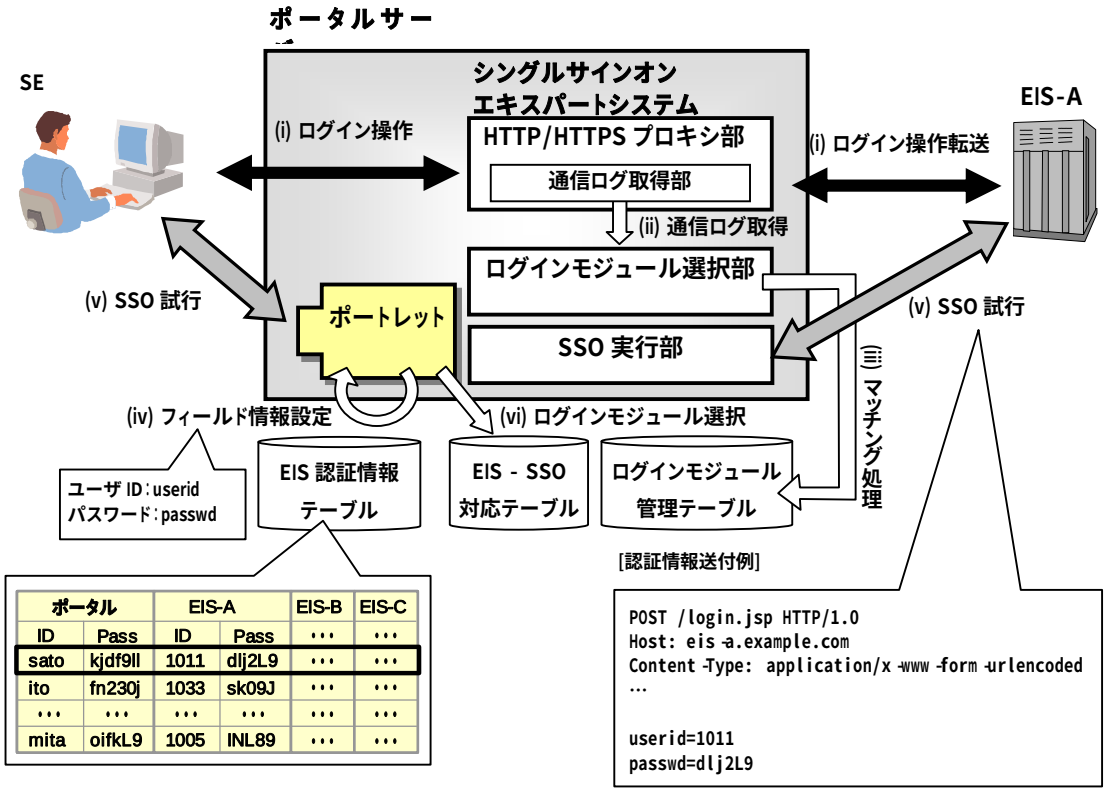

図 6-1 処理手順 –設定フェーズ-

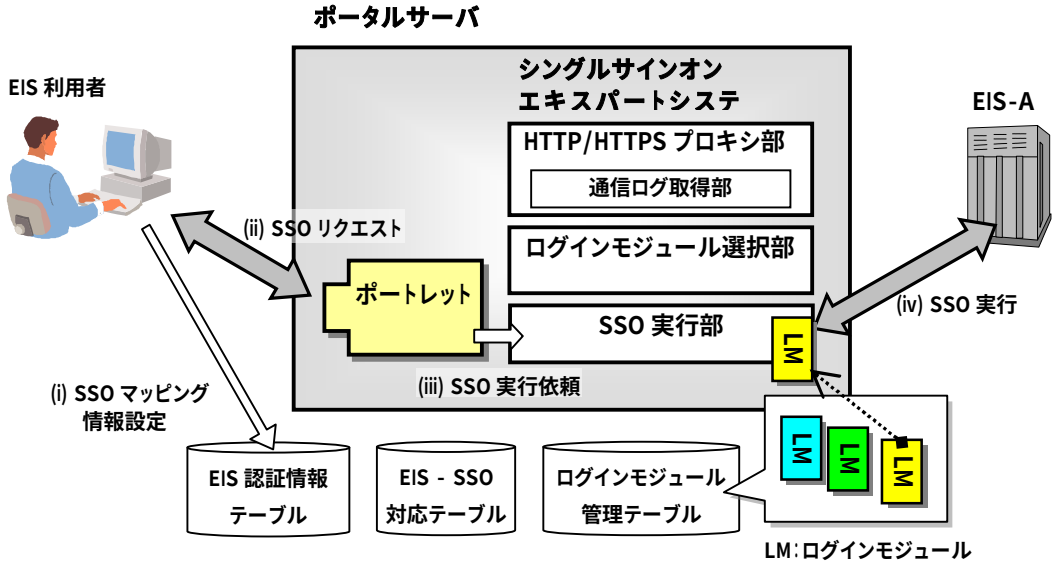

図 6-2 処理手順 –実行フェーズ-

7. 評価

上記で提案したシングルサインオンエキスパートシステムについて、ある企業における EIS に適用し評価を行った。なお、ログインモジュールとしては、2章で説明した基本的な実現方式を用いた。表 7-1に評価結果を示す。#10、#16 を除いた 14 のシステムが直接ログイン方式およびポータル代行ログイン方式(domain 指定)のログインモジュールにより SSO が実現できることを確認した。

#10 は、クライアント(ブラウザ)側でスクリプト

が動作するなど、2章で提案している SSO 方式では SSO が実現不可能なログイン方式であった。#16 の方式ではログインの前処理としてセッション確立が必要だったため2章で提案している素朴な SSO 方式では SSO が不可能であった。しかし、上記のように形式化した SSO については、的確に SSO の自動選択が実現できる見通しを得た。

また、#2 のシステムは SSL により通信データが暗号化されているため机上での検証は困難である。しかし、提案する SSO エキスパートシステムでは HTTPS プロキシもサポートしており、適切なログインモジュールを選択することが出来た。

表 7-1 適用結果

#	システム名	適用結果
1	資材発注システム A	直接ログイン方式 / ポータル代行ログイン方式(domain 指定)
2	資材発注システム B (SSL にて暗号化)	直接ログイン方式
3	開発情報システム	直接ログイン方式 / ポータル代行ログイン方式(domain 指定)
4	社内英語検定募集システム	同上
5	コンサルティングマネジメントシステム	同上
6	経営情報システム	同上
7	プロジェクト従事日誌管理システム	同上
8	試作部門情報統合化システム	同上
9	社員ポータル	同上
10	旅費システム	×
11	就業管理システム	同上
12	行事日程･募集申込システム	同上
13	資産情報システム	同上
14	プロジェクト情報システム	同上
15	情報機器セキュリティ対策チェックシステム	同上
16	ワークフローシステム	×

8. おわりに

ポータルソリューション強化を目的に企業情報システムのシングルサインオンを容易に実現可能とするシングルサインオンソリューションの検討を行った。本稿では、提案するシングルサインオンソリューションの 1 システムである SSO エキスパートシステムについてプロトタイピングを行い、評価を行った。以下、結論を示す。

(1) 通信の解析など Web アプリケーションの詳細を意識させることなくシングルサインオンを実現

Web アプリケーションの認証方式、セッション維持方式を調査、分析し、通信パターンとして形式化した。更に形式化した通信パターンと、通信パターンに対応したシングルサインオンを実現するログインモジュールをデータベース化した。企業情報システムのログイン通信を解析し、データベース化した通信パターンと照らし合わせることで企業情報システムに適用可能なログインモジュールを自動選択することによりシングルサインオンを容易に実現可能とした。これにより、シングルサインオンの自動化が図られ、低コストかつ迅速に構築可能なシングルサインオンソリューション実現の見通しを得た。

(2) 未知の認証、セッション維持方式にも対応できるよう拡張性を考慮したシングルサインオンシステムの検討

動的にログインモジュールをリンケージ、呼出しすることでログインモジュールの追加、変更に柔軟に対応できるよう設計した。また、HTTP のデータ構造に沿って通信パターン定義を可能とすることで、未知の認証、セッション維持方式に対しても通信パターンを柔軟に定義できるよう設計した。これにより、拡張性の高いシングルサインオンシステム実現の見通しを得た。

参考文献

[1] 菊地、”シングルサインオン実現方法に関する考察”、FIT 2004 第 3 回情報科学技術フォーラム、O-008 (2004)