

Webアプリケーションの特徴を考慮した シングル・サインオン代行入力スクリプト開発手法

劉 照深

A method to develop automatic single sign-on script for web applications

Shoushin Liu

エンタープライズ・シングル・サインオン (ESSO) は、クライアントでさまざまなアプリケーションのログオン画面に認証情報を代行入力する主要なシングル・サインオン・ソリューションの1つである。IBM が提供する ESSO 製品を Web アプリケーションに適用する際に、ブラウザ独自の進む・戻るボタンなどによる画面遷移と Web ページの表示遅延対策を考慮する必要がある。本論文では ESSO 製品の仕様、ブラウザの機能、Web アプリケーションの画面設計の特徴を踏まえて Web アプリケーションに対応した代行入力スクリプトの開発手法を提案し、適用事例の検証によりその有効性を示す。

Enterprise single sign-on(ESSO) is one of the major single sign-on solution to fill in credential automatically on logon page. There is two concern points to develop automatic sign-on script for web application based on IBM ESSO product. First is to consider page navigation function of browser and second is to prevent delay of web page display speed. This paper proposes method to develop automatic sign-on script for web application based on ESSO product design, function of browser and specification of web application page design and verifies its effectiveness through applied project.

Key Words & Phrases : エンタープライズ・シングル・サインオン, 代行入力, ブラウザー・ヘルパー・オブジェクト, Windows, インターネット・エクスプローラー
Enterprise Single Sign-On, Automatic Sign-On, Browser Helper Object, Windows, Internet Explorer

1. はじめに

エンタープライズ・シングル・サインオン (ESSO) は、さまざまなアプリケーションのログオン画面に認証情報を自動で代行入力するソリューションである。

Tivoli Access Manager for Enterprise Single Sign-On (TAM ESSO) は、IBM が提供する ESSO 製品であり、各アプリケーションへの代行入力スクリプトの開発のためにステートマシンと呼ばれる代行入力スクリプト開発が必要となる。ステートマシンは、アプリケーションの画面遷移を追跡し、代行入力対象のログオン画面やパスワード変更画面の表示を検知して認証情報を代行入力する。そのため、想定外の画面遷移により追跡が止まった場合、以降の画面で代行入力が失敗する。これを防ぐため、ステートマシンの開発時にはアプリケーションのあらゆる画面遷移を網羅しておく必要がある。しかし、アプリケーションの画面遷移は多岐にわたり、特に

Web アプリケーションの場合、Web ページ内の URL リンクやボタンによる画面遷移以外にブラウザの提供する進む・戻るボタンやブックマークなどによる自由な画面遷移を考慮する必要がある。

また、TAM ESSO はブラウザ上で画面遷移の追跡のために、ブラウザにプラグインを導入するが、このプラグインは、ブラウザ上で全画面遷移を監視することにより Web ページの表示遅延を引き起こすケースがあるため、この対策についても考慮する必要がある。

本論文ではアプリケーションの画面遷移をモデル化した標準画面遷移モデルを定義し、これを基準としてシングル・サインオンのための代行入力スクリプトを開発する手法を示す。さらにこの手法により開発された代行入力スクリプトに対して、Web アプリケーション固有の画面遷移と Web ページ表示速度遅延の軽減を考慮したカスタマイズを行う手順を示す。そして、世界最大規模の TAM ESSO 構築事例に適用した結果を詳細に分析し、その有効性を検証する。ESSO を実現するためのアーキテクチャーは、各ベンダーごとに異なるため、他社製品の開発手法を TAM ESSO には適用できない[1][2]

提出日:2011年5月9日 再提出日:2012年3月9日

[3]. また, TAM ESSO の既存の開発ガイド [4] [5] は, カスタマイズ事例の紹介に重点が置かれており, 開発品質向上のための体系的な開発手法はまだ提案されていない. 本論文はこれを補完するものである.

2. TAM ESSOの代行入力モデル

開発標準を示す前提として TAM ESSO の代行入力の仕組みを示す [6]. TAM ESSO のエージェントは, 代行入力するために対象アプリケーションの起動を常時監視する. Web アプリケーションの場合は, ブラウザーに表示される Web ページの URL が監視対象となる. URL がマッチした Web ページを解析し, 代行入力スクリプト内に記憶されたユーザー ID やパスワードの入力フィールドの HTML タグなどの画面部品情報とのマッチングを行う. URL や HTML タグなどの画面部品を検知する実体を「トリガー」と呼び, ログオン画面の入力フィールドなどをトリガーで検知して代行入力する操作を「アクション」と呼ぶ. トリガーの発生と同時にアクションを実行し, ステートと呼ばれる状態を次々に遷移する. トリガーに応じてステートを次々に遷移する状態遷移図がステートマシンであり, ステートマシンの定義情報が代行入力スクリプトとなる.

図 1 に, TAM ESSO の代行入力スクリプト開発ツール Access Studio のウィザードを使用して生成した標準的なステートマシンを示す. 1 回目にログオン画面を起動した際にこれをトリガー①で検知し, ステート A からステート B に遷移する. 次にユーザーがユーザー ID とパスワードを入力し, ログオンボタンをクリックする. これをトリガー

②で検知し, 入力されたユーザー ID, パスワードを捕捉してメモリーに格納し, ステート C に遷移する. ログオンが成功した場合は業務メニュー画面などが表示されるため, これをトリガー③で検知し, メモリー上の認証情報をディスクに保存してステート A に戻る. 2 回目以降のログオンではログオン画面の表示をトリガー①で検知し, ディスク上の認証情報をメモリーに読み出し, ユーザー ID とパスワード・フィールドに代行入力する.

パスワード変更画面の場合は, 旧パスワード・フィールドに現パスワードを代行入力する. パスワードが変更された際には, 新パスワードを捕捉し, 記憶している旧パスワードを新パスワードに置換して次回の代行入力時に使用する.

例えば, 図 1 のトリガー③でログオン成功画面の検知に失敗した場合, 開始ステートに遷移できないため, ログオン画面が再度表示された場合にトリガー①で検知して代行入力することができない. 従って, このような問題を防ぐためにアプリケーションの画面遷移を網羅し, これに対応して適切なステートに遷移させることが重要となる.

3. Webページ表示遅延発生メカニズム

Web ページ表示遅延発生メカニズムを Internet Explorer (IE) を例にして説明し, IE での遅延対策の必要性を示す. IE には Browser Helper Object (BHO) と呼ばれるプラグインを導入できる [7]. IE に導入された TAM ESSO の BHO は, IE の起動と同時に起動され, IE のプロセス内で稼働し, IE にダウンロードされた Web コンテンツを監視する. BHO は, ブラウザーにダウンロードされた Web コンテンツの URL が代行入力対象画面の URL であるかを確認し, 合致した場合はメモリーにステートマシンを読み込む. 以降, ステートマシンの読み込みを「ステートマシンの起動」, 代行入力対象画面の URL を「ステートマシンの起動条件」または「起動条件」と示す.

Access Studio のウィザードでステートマシンを生成した場合, 代行入力対象画面の URL のプロトコル (http または https) とホスト名までを起動条件とし, パスまでは起動条件に含めない. このため, 図 2 に示すようにログオン画面が複数フレームで構成される場合, すべてのフレームの URL が起動条件に合致し, ログオン後に表示される代行入力対象外の業務画面の URL も起動条件に合致してしまう. TAM ESSO の BHO は, ブラウザーにダウンロードされた画面や画面内の各フレーム

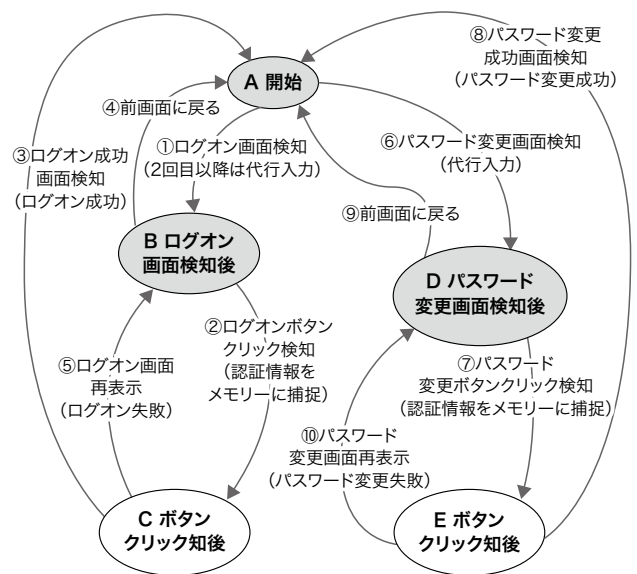


図1. 基本ステートマシン

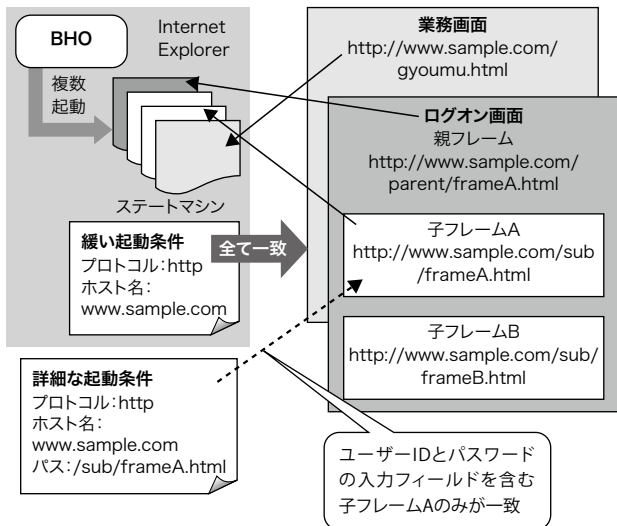


図2. 起動条件と画面構成の関係

の URL が起動条件に合致すれば、その回数だけ同じステートマシンを起動する。起動されたそれぞれのステートマシンは各画面や画面内の各フレームに代行入力対象のユーザー ID やパスワードの入力フィールドを含む場合に代行入力行いが、含まない場合は起動されずたままとなる。

ステートマシンの起動時間は、Intel Core i5 520M 2.4GHz プロセッサを搭載した PC で約 200 ミリ秒である。比較的新しいデスクトップ PC 用プロセッサでも約 200 ミリ秒かかるため、1 画面に代行入力対象外フレームが 5 枚以上含まれる場合は不要なステートマシンの起動のために 1 秒以上の遅延が発生する。また、ログイン後の業務画面においても不要なステートマシンの起動のために遅延が発生する可能性がある。秒単位での Web ページ表示遅延は一般的なユーザーが十分に認識可能なレベルであるため、Web ページ表示遅延対策が必要となってくる。

4. Webアプリケーションに対応した代行入力スクリプト開発の流れ

Web アプリケーションの画面遷移をステートマシンで網羅的にカバーするために図 3 に示す 3 つのステップが必要になる。本論文ではアプリケーションの一般的な画面遷移を標準画面遷移モデルとして定義し、ステップ 1 で標準画面遷移モデルを基準として画面遷移の大部分を網羅するステートマシンを開発する。ステップ 2 では、例えばログイン後にセキュリティレベルの高い画面を参照する時点で再度ログイン画面が表示されるなどの標準画面遷移モデルにないアプリ

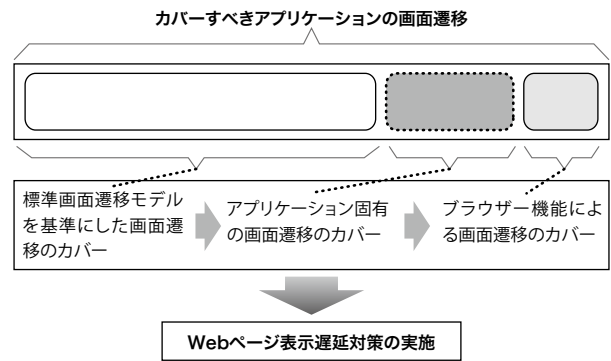


図3. 代行入力スクリプト開発手順

ケーション固有の画面遷移が存在する場合にはこれらを網羅する。そしてステップ 3 でブラウザの進む・戻るボタンなどの画面遷移機能を利用した画面遷移が可能な場合に、これらの画面遷移を網羅する。最後にステップ 3 までの手順により画面遷移を網羅したステートマシンに Web ページ表示遅延対策を実施する。また、遅延対策により幾つかの問題が発生する可能性があるため、これらに対処する。各手順の詳細については 5 章から 7 章に示す。

5. 標準画面遷移モデル

代行入力スクリプト開発では、代行入力対象とする画面とこれに関連する画面遷移を把握して網羅することが必要である。アプリケーションでログイン画面やパスワード変更画面などのセキュリティ機能を提供する画面設計については IPA や JNSA など設計標準 [8] [9] が定義されており、各企業でもレベルの差はあるが、このような設計標準がアプリケーション開発標準に取り入れられていると考える。従って、本論文ではこれらの設計標準を参考にし、図 4 に示す標準画面遷移モデルを定義する。各アプリケーションで代行入力スクリプトを開発する際には標準画面遷移モデルを基準として以下の考慮点を踏まえる必要がある。

- (1) 代行入力対象のログイン画面、パスワード変更画面とその前後の画面との間で画面遷移した場合にこれらを検知し、ステートが適切に遷移することを確認する。
- (2) パスワード変更画面については期限切れ前後のパスワードと初期パスワードのそれぞれを変更するための画面遷移を網羅する。特に図 4 の太線で示す期限切れパスワードや初期パスワードを変更するための画面遷移は Access Studio のウィザードでは

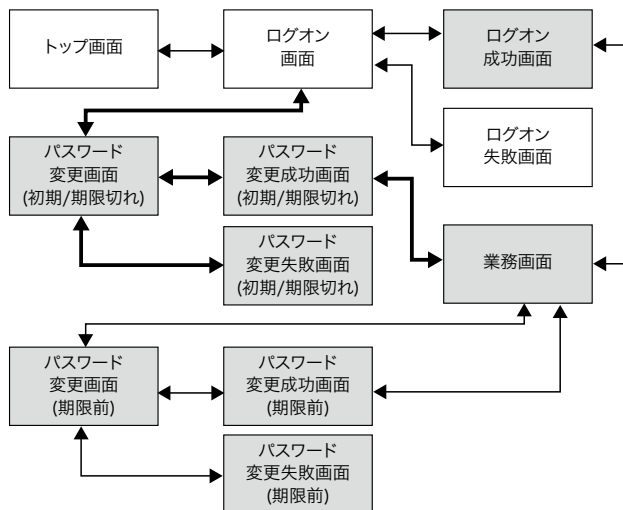


図4. 標準画面遷移モデル

網羅されないため、これらの画面遷移に対応した処理フローを状態マシンに手動で追加する必要がある。

- (3) ログオフやセッション・タイムアウトでログイン画面に遷移するケースがある場合はこれを網羅する。図4の中で灰色で示す画面を利用中にログオフやセッション・タイムアウトが起きる可能性があるため、その後ログイン画面に遷移する場合はこれを検知するトリガーが存在するかを確認する。

標準画面遷移モデルと異なる、またはこの中でカバーされないアプリケーション固有の画面遷移に対しても上記の考慮点を適用することができる。これは、これらの画面遷移の場合でも代行入力対象画面が表示され、ボタンをクリックして次の画面に遷移する基本的な画面遷移は同じであると考えられるためである。

6. ブラウザー機能を使用した画面遷移の網羅

Web アプリケーションの場合は、画面内の URL リンクで別の Web サイトの画面に遷移したり、同じ Web サイト内の別の画面に遷移できる。さらにブラウザーの進む・戻るボタンやブックマーク・URL フィールドへの直接入力を使用して自由に画面遷移できる。従って、代行入力対象画面とその前後の画面間との遷移のみでなく、任意の画面間との遷移を考慮する必要がある。これらの画面遷移を網羅するため、TAM ESSO の製品仕様に着目し、同じ Web サイト内の任意の画面との間を遷移する場合と別の Web サイトの任意の画面との間を遷移する場合の対応方法を以下に示す。

- (1) 同じ Web サイトの任意の画面との間の遷移への対応

同じ Web サイト内の任意の画面との間の遷移パターンは多数存在するが、状態マシンに着目することでより効率的に多数の遷移パターンを網羅できる。まず、標準画面遷移モデルを基準に開発した状態マシンに着目し、その中で画面が表示された状態で次に何らかのトリガーの発生待ちの状態を抽出する。例えば、図1では状態A,B,Dが該当する。ログイン後に任意の業務画面を遷移している間は状態Aに位置するため、任意の業務画面を遷移中の状態は状態Aに集約できる点がこの手法のメリットである。抽出された各状態で表示されている画面の間を遷移できる場合は、これを検知するトリガーを追加する必要がある。状態B,Dから状態Aに遷移する場合は、任意の業務画面への遷移を検知するために同じ Web サイトの任意のパスの URL を検知可能なトリガーの検知条件をワイルド・カードにより指定する。

- (2) 別の Web サイトの任意の画面との間の遷移への対応

状態マシンは、別の Web サイトに遷移した時点でアンロードされ消滅する。再度、同じ Web サイト内の画面に戻った場合は、これを検知して状態マシンを新規に起動する。状態マシンが起動する場合は、必ず図1の開始状態(状態A)から状態遷移する。従って、6.(1)で抽出した状態で表示されている画面と別の Web サイトの画面との間で遷移が可能な場合は、6.(1)で抽出した状態で表示されている画面に戻った場合に、これを検知できるトリガーを開始状態に追加する必要がある。

上記に示したケース以外に、以下に示す2つの重要な特殊ケースを考慮する必要がある。

- (3) リフレッシュボタンによる画面の再表示

アプリケーションでブラウザーのリフレッシュ・ボタンが利用できる場合、代行入力対象画面の表示後にリフレッシュ・ボタンで再表示ができる。アプリケーションの仕様により自動的に再表示されるケースもあるため、このような場合、再表示されても代行入力されるように代行入力対象画面が表示されているすべての状態で画面の再表示を検知して同じ状態に戻るトリガーを追加する必要がある。例として、図1の状態マシンの場合、状態B,Dで再表示を検知するためのトリガーを追加する必要がある。

- (4) ログイン前のブックマークから業務画面へのアクセス

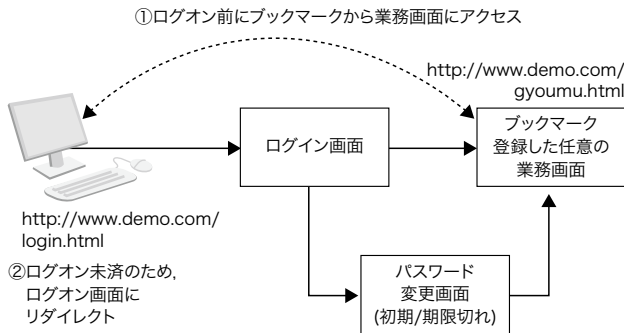


図5. ブックマークから業務画面へのアクセス

ユーザーは、ブラウザのブックマークに頻繁に利用する業務画面を登録し、アプリケーションを開く際にログオン画面ではなく、ブックマークに登録した業務画面を開く場合がある。この場合、アプリケーションによっては図5に示すようにログオン画面にリダイレクトし、ユーザーのログオン後に再び要求した業務画面にリダイレクトするといったユーザー利便性に配慮した画面遷移を実装する場合がある。ログオン時にパスワード期限切れの場合、図5に示すようにログオン後にパスワード変更画面でパスワードを変更してから業務画面にリダイレクトする。これらのケースでリダイレクトされるログオン画面が通常のログオン画面と異なる場合は、このログオン画面を検知するトリガーを追加する必要がある。また、図4の標準画面遷移モデルのログオン成功や初期・期限切れパスワード変更成功時の画面は、業務メニュー画面などを検知対象とすることが多い。従って、ブックマークから業務画面にアクセスした場合は、ブックマークから要求した任意の業務画面を成功画面として検知するトリガーを追加する。例として、図1では業務メニュー画面を検知するトリガー③以外に任意の業務画面を検知するトリガーを追加する必要がある。

7. Webページ表示遅延対策

3章で示したように代行入力対象画面以外の画面でステートマシンが起動するため、Webページの表示遅延が発生する。これを回避するためには図2に示すようにステートマシンの起動条件をプロトコルとホスト名のみではなくURLのパスのレベルまで指定することにより、代行入力対象画面でのみステートマシンを起動させ、その他の画面で起動するのを防げる。6章で示したように、別のWebサイト間との画面遷移を考慮した場合、開始状態から延びるトリガーで画面を検知してステートマシンを起動する必要がある。従って、開始

状態から延びるトリガーで検知する画面のURLをすべてステートマシンの起動条件に指定する必要がある。例として、図1のステートマシンの場合は開始状態からログオン画面とパスワード変更画面を検知しているため、双方のURLを起動条件に指定する。6章で示したように、ログオン前にブックマークから業務画面にアクセスした場合にリダイレクトされるログオン画面のURLが通常のログオン画面のURLと異なる場合は、このログオン画面を検知するトリガーを開始状態に追加する必要がある。このため、リダイレクトされたログオン画面のURLも起動条件に指定する。

起動条件をURLのパスのレベルまで絞った結果、2つの大きな問題が発生するため、以下にそれぞれの問題の詳細とこれらに対する対応策を示す。

7.1 起動条件の指定が難しいURLへの対応

アプリケーションの仕様によりステートマシンの起動条件に指定すべきURLが可変のケースがある。例えば、ログオン画面のURLにセッションIDが含まれ、毎回ログイン画面にアクセスするたびにURLが変わる場合は、可変部分の文字列をワイルドカードなどにより起動条件に指定する必要がある。ここで指定した条件が代行入力対象外の画面にも合致した場合、ステートマシンが起動してWebページの表示遅延を引き起こす可能性があるため、これらの画面のURLに合致しないようにワイルドカードの範囲を絞る必要がある。また、URLのパスの部分全体が可変で一定のルールがない場合やログオン画面と業務画面のURLの区別がつかないなどの場合はワイルドカードなどによる条件指定ができないため、起動条件の調整による遅延対策が効かないケースもある。

7.2 ステートマシンのアンロード防止

図6は、ログオン画面からパスワード変更画面までの画面遷移とこれらを検知するステートマシンの相関図である。ステートマシンの起動条件をパスのレベルまで絞った結果、図6の③のように、ログオンボタンをクリック後にメニュー画面が表示される間に検知対象外の画面に遷移した場合にステートマシンがアンロードされる。起動条件がプロトコルとホスト名のみの場合、このような画面も同じWebサイト内の画面とみなされアンロードされないが、起動条件をパスのレベルまで絞った結果、パスに合致しない画面は別のWebサイトとみなされステートマシンのアンロードが起きる。このような場合、図6の③のトリガーのように、状態にループする

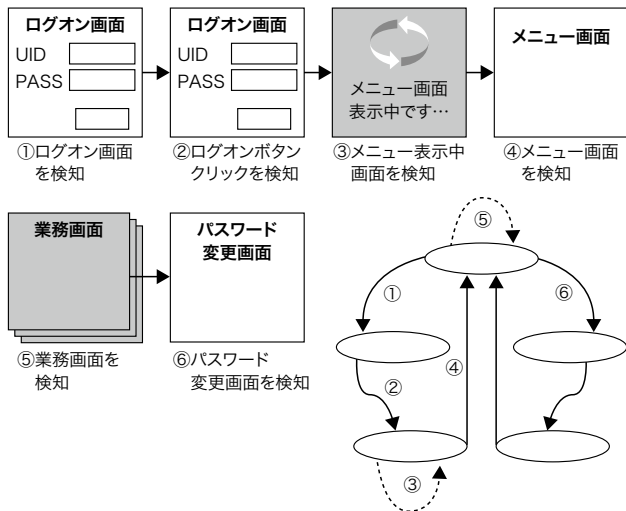


図6. ステートマシンのアンロード防止

トリガーを追加し、間に入った画面を検知する必要がある。また、図6の⑤のように、ログオン後に遷移する任意の業務画面を検知するトリガーを追加することにより、これらの画面を遷移している間のステートマシンのアンロードを防げる。これによりパスワード変更画面が表示された場合やログオン画面が再び表示された場合にステートマシンを新規に起動せずに、起動中のステートマシンでこれらの画面を検知できる。このように、ステートマシンの起動回数を低減することによっても遅延を防止できる。

8. 提案手法の検証

提案手法を実事例に適用した結果を分析し、その有効性を検証する。本手法は、あるある親会社1社とそのグループ会社4社の計5社の28のWebアプリケーションの代行入力スクリプト開発に適用した。アプリケーション数は国内では最大規模である。独自開発に加え

てSaaSやパッケージのアプリケーションが利用されており、画面仕様もさまざまであり、最新のWeb画面表示技術を採用したアプリケーションも存在する。従って、多くの企業で利用されているWebアプリケーション種別を網羅しており、評価対象事例として適切と考える。以下では適用結果の分析により、工数削減や品質向上の達成度やWebページ表示遅延対策により画面表示速度がどの程度改善されたかを定量的に示す。また、本手法の汎用性について検証する。

8.1 開発効率と品質の検証

表1に示す提案手法の適用結果から開発効率と品質を検証する。適用事例では標準画面遷移モデルのもとに開発したステートマシンにブラウザ機能による画面遷移の網羅とWebページ表示遅延対策を考慮したカスタマイズを行った。最初に取り組んだ2アプリケーションでは試行錯誤を重ねたため、1アプリケーションあたりの代行入力スクリプトの開発・デバッグ・テストに平均10日を要したが、提案手法を確立して適用した結果、表1のように開発期間が80%短縮されている。また、表1のバグ発生率と実運用における実績から、以下の理由で十分な品質が確保できたと考える。

- (1) カスタマイズ後の1回目のテストでバグ発生率が最大でも1%と低い点は、提案手法により画面遷移を十分に網羅していることを示している。
- (2) 実運用で発生した代行入力スクリプトに依存する不具合は数件のみであり、実運用でもバグ発生率が低い結果となっている。

8.2 改善後の画面表示速度の検証

表2に3種類の画面の表示時間と1ステートマシンの平均起動時間の比較表を示す。適用事例では最

表1. テスト結果とカスタマイズ内容

1アプリケーションあたりの開発期間	
提案手法未適用ケース	平均10日
提案手法適用ケース	平均2日
1アプリケーションあたりのテスト項目数と1回目の単体テストでのバグ発生率	
テスト項目数	平均35ケース
バグ発生率	1% (最大ケース)
ステートマシン停止対応で追加した処理フロー数	
処理中停止対応	6アプリケーション
起動回数低減対応	22アプリケーション

表2. 改善後の起動時間と画面表示時間の比較

	起動時間 (秒)	画面表示時間 (秒)
CASE1	V8.1.2.0028 (遅延対策を未適用)	
画面A	2.488	3.4
画面B	2.523	14.2
画面C	1.872	24.6
CASE2	V8.1.2.0028 (遅延対策を適用)	
画面A	2.479	3.2
画面B	0.000	2.0
画面C	0.000	2.8

テスト PC : ThinkPad T410(Windows XP SP3)

新の V8.1.2.0028 を採用した結果、1 ステートマシンの起動に平均 2.5 秒かかるために Web ページの表示遅延が発生したため、遅延対策を実施した。CASE1 と CASE2 は遅延対策前後の計測結果である。画面 A がログオン画面、画面 B,C は業務画面で画面 C はより遅延が発生しやすい複雑な画面設計となっている。表 2 の結果から、遅延対策により代行入力対象外の業務画面ではステートマシンが起動されておらず、画面の表示速度が改善されている。

8.3 汎用性の検証

適用事例では、標準画面遷移モデルをすべての Web アプリケーションの画面設計に適用し、標準でカバーすべき画面遷移だけでなく、標準でカバーされていないアプリケーション固有の画面遷移も抽出できることを確認した。ブラウザの画面遷移機能による遷移を網羅する手法については、ステートマシンに着目し、その中で画面表示状態のステートを抽出し、これらのステート間の遷移を分析するため、どのような Web アプリケーションのステートマシンにも適用できた。

Web ページ表示遅延対策は、7 章で述べたように、アプリケーションの仕様上、ステートマシンを起動する URL をパスのレベルで特定することが難しい場合以外は適用できる。適用事例で遅延対策を適用できなかったケースは 2 ケースのみであった。ステートマシンのアンロード防止への対応については、表 1 に示すように約 20% のアプリケーションでアンロード防止対策が必要であり、ステートマシンの起動回数低減のための対策が約 80% のアプリケーションで有効であった。TAM ESSO の仕様上、アプリケーションの各フレームでステートマシンが個別に起動するため、ログオン画面とパスワード変更画面のフレームが異なる場合は、ログオン画面のステートマシンのアンロードを防いでも、このステートマシンをパスワード変更画面に流用できない。しかし、同じフレームである場合やフレームを使用していないアプリケーションではアンロード防止による起動回数低減の対策が有効である。

9. おわりに

本論文では標準画面遷移モデルを定義し、これをもとにして開発されたステートマシンに対してブラウザ機能による画面遷移と Web ページ表示遅延を考慮したカスタマイズ手法を提案した。さらに、適用事例を検証してその有効性を示した。

Web ページ表示遅延対策でステートマシンの起動条件に設定した URL が代行入力対象外の業務画面の URL と重複した場合、遅延につながるため、起動条件をパスのレベルに絞る過程でアプリケーション担当者にアプリケーション仕様を確認しながら、起動条件を調整する作業が非常に重要である点を最後に述べておきたい。

本論文で提案する手法を活用いただき、開発工数削減と品質向上に役立てていただきたい。

謝辞

本手法の考案に際し、多くの有用なご指摘をいただいたプロジェクト・メンバーにあらためて深謝いたします。

参考文献

- [1] Gartner, "MarketScope for Enterprise Single Sign-On", ID Number: G00219310 (2011.09.30).
- [2] Hitachi ID Systems, Inc, "成功するエンタープライズ・シングル・サインオン: 展開上の課題に対処", <http://hitachi-id.com/login-manager/ja/docs/next-generation-ssso.pdf> (2012.02.27).
- [3] Hitachi ID Systems, Inc, "従来の E-SSO における問題点", <http://hitachi-id.com/login-manager/ja/docs/next-generation-ssso.pdf> (2012.02.27).
- [4] "AccessProfiles - Beyond the wizard" : <https://www.ibm.com/developerworks/wikis/display/tivoliaccessmanagerforesso/AccessProfiles+-+Beyond+the+wizard> (2011.05.09).
- [5] RedPaper: "A Guide to Writing Advanced Access Profiles for IBM Tivoli Access Manager for Enterprise Single Sign-On", <http://www.redbooks.ibm.com/abstracts/redp4767.html?Open> (2011.08.15).
- [6] "Access Studio Guide", SG23-9956-03, http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itamesso.doc/IBM_TAM_ESSO_AccessStudioGuide_pdf.pdf
- [7] "Browser Helper Objects: The Browser the Way You Want It" : [http://msdn.microsoft.com/en-us/library/bb250436\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/bb250436(v=vs.85).aspx) (2011.05.09).
- [8] IPA: "安全な Web サイトの作り方" (2009.01.).
- [9] JNSA: "セキュアシステム開発ガイドライン" (2009.12.05).



日本アイ・ビー・エム
システムズ・エンジニアリング株式会社
システムズ・マネジメント
IT スペシャリスト

劉 照深 Shoushin Liu

[プロフィール]

1999年、日本IBM入社。主にTivoliの認証・ID管理製品の技術支援を担当。2008年からTivoli Access Manager for Enterprise Single Sign-Onの技術支援も行っている。
shoushin@jp.ibm.com