

Web サイトを用いた対面業務を可能とする Web ページ共有方式

城島貴弘、子林秀明

NEC サービスプラットフォーム研究所

概要

近年インターネットの普及と Web サービスの発達により、ネットバンキングやイーコマース(EC)などのオンライン上での商取引や物販、金融サービスが既に一般化している。しかし、銀行の窓口業務における口座開設やローン相談など、対面による接客によって契約が成立するサービスについてはオンライン化がまだまだ困難である。本稿では、リモートに離れたユーザ同士が共通の Web ページを同時に共有操作可能な Web ページ共有方式を提案し、これを用いた対面業務のオンライン化手法及びその課題について検討する。

Web page sharing methods which realize online remote consultation business using a web site

Takahiro SHIROSHIMA and Hideaki NEBAYASHI

NEC Service Platforms Res. Labs.

Abstraction

As the spread of the Internet and the advances in Web services, such as trading services, net-banking services, e-commerce services and so on, online services based on World Wide Web has been a commonplace. But it is still hard to make an online service of meeting sales such that the window services on a bank: an account opening application, a consultation of loan, and so on. So we propose the web page sharing method that makes it possible for each remote user to operate a web page simultaneously, and discuss some troubles on filling out an application of web form.

1. はじめに

近年、インターネットの普及によりネットバンキングやイーコマース(EC)など、Web によるオンラインサービスの利用が一般的になっている。このような Web によるオンラインサービスは、ユーザがブラウザを用いて Web サイトにアクセスし、ブラウザ上に表示された Web ページをユーザ自身が読み取り、Web ページに設けられた入力領域に対して指示された内容を自ら入力して送信することにより完結する。一般的にこれらの作業は全てセルフナビゲーションを前提としており、その作業を通じてユーザは支援を受けることができない。

しかし、銀行の窓口業務における口座開設やローン相談、また、生命保険の契約や更新など、人同士の対面を前提としている業務を Web 上でオンラインサービス化する場合、利用者が Web ページを一読

しただけではその内容を把握することが難しく、また、契約内容に沿った入力を誤り無く最後まで完了させることが難しい。

このような対面業務のオンラインサービス化においては、Web ページのデザインの工夫や、JavaScript などの利用によりブラウザ操作を支援する、といった対応がなされてきた。しかし、千差万別であるユーザの疑問点に全て答えられる Web サイトを開発するのは至難と言える。

そこで我々は双方向コミュニケーション基盤を利用したリアルタイム Web 共有方式による Web ページ操作の遠隔支援を提案してきた^{[1][2]}。本方式では、双方向コミュニケーション基盤上で動作するプッシュサーバを介して、ネットワーク上の各ブラウザの操作情報をプッシュコマンドとして双方向にやりとりし、互いのブラウザにその操作を反映させることにより遠隔でブラウザの表示を同期させることが可能となる。こ

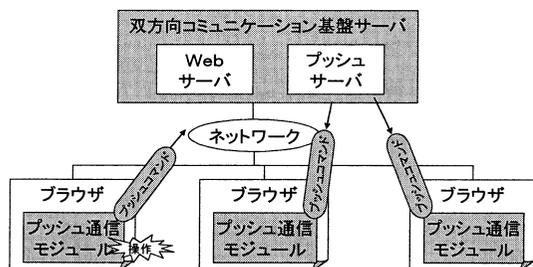


図1 双方向コミュニケーション基盤

のリアルタイム Web 共有に加え、電話や Web カメラなどの音声映像メディアを同時に用いることにより、他者の Web ページの操作を Face to Face と同等の感覚でネットワーク越しに遠隔で支援することができる。

このような Web 共有システムを利用したオペレータによる遠隔支援により、ユーザのセルフナビゲーションでは困難であった対面業務の Web ページによるオンライン化が実現可能となる。

2. 対面業務への適用における課題

しかし、Web 共有システムを対面業務へ適用するにあたり以下の課題に関する検討を要する。

(A) 双方向性における課題

Web ページの共有中にオペレータまたはユーザによるブラウザの操作を、双方向にもう一方のブラウザに伝播、適用させる同期処理において発生する課題について述べる。

(A-1) 操作の制限

対面業務においてはユーザの預金口座番号やクレジットカード番号、Web サイトへのログインパスワードなど、オペレータにも秘匿しておきたい情報を入力する必要がある。このような場合、入力操作の一部または全部を他方に送信しないといった共有の制限処理ができる必要がある。

(A-2) 操作履歴の追跡

対面業務での作業においてオペレータ、ユーザのどちらがその操作を行ったかが問題となる場合がある。例えば、サービス申し込み時における最終的な申し込みボタンの押下などはユーザが押したことを証明できる必要がある。

(B) トランザクションにおける課題

共有する Web ページを実際に設置している Web サイトと Web ページを共有する各ブラウザとの間で送受信される HTTP リクエストや、そのリクエストに基づく Web サイト側でのアプリケーション処理時に発生する課題について述べる。

(B-1) リクエストの単一性

ユーザのブラウザ操作により発生する HTTP 通信は、画像の取得やリダイレクトを除くと、基本的に 1 操作につき 1 つである。しかし、Web 共有システムのような、複数のブラウザが同じ Web ページを表示しその動作が同期されているシステムでは、ユーザ操作によるリクエストはブラウザの数だけ発生する。これは、商品購入時やサービス申し込み時など、Web サイト上でユーザのリクエストに応じたトランザクション処理が実施される場合、同一の商品を二重に購入したり、一方の申し込みが拒絶されたりといったトランザクションの重複による問題を引き起こす。

また、購入手続きなど複数のページに渡るトランザクション処理を実施する場合、ブラウザからのリクエストと利用者を一対一に関連づけるため、セッションによるブラウザの特定処理を実施する。これは大抵の場合、ログインやサービス開始時に各ブラウザに対してクッキーなどの仕組みを用いてセッション情報を付加することで実装する。よって、それぞれのブラウザに付加されたセッション情報が異なる場合、そこで表示される Web ページも異なるものが表示される。したがって、セッション制御を用いる Web サイトを共有する場合には、ユーザのブラウザと関連づけられているセッションに一時的にオペレータが参加できる必要がある。

(B-2) HTTPS リクエストのサポート

対面業務で用いる Web ページは扱う情報の機密性により、通信路を SSL で暗号化した HTTPS を利用する場合はほとんどである。この場合、SSL 通信の端点である Web サイトとブラウザ以外の第三者が HTTP リクエスト及び返信される Web ページを取得することができない。よって、Web 共有におけるオペレータのブラウザのような第三者が、ユーザのブラウ

ザで表示されている Web ページの内容を取得し、表示するための何らかの方法が必要となる。

(C) 導入時における課題

Web 共有を用いたオペレータによる対面業務の支援システムを、実際に各 Web サイトへ導入するにおいて発生する課題である。

(C-1) 既存 Web ページの流用

Web 共有を対面業務に適用するにあたり、新しく Web サイトを一から構築することはまれであり、既存の Web サイトをそのまま流用もしくは一部改変した形でのサービスインを望まれる場合が多い。この場合、Web 共有を導入しても従来の Web ページで用いられていた HTML や JavaScript がそのまま動作することが求められ、また Web サイト上の Web アプリケーションにも影響が出ないことが望まれる。

(C-2) 実運用環境への導入

通常オンラインサービスを運用する Web サイトは、可用性の観点からクラスタ構成となっている場合がほとんどであり、実際に稼働している Web サーバがかなりの数に上ることもある。このようなサイトに Web 共有を導入する場合、各 Web サーバに対し個別の設定作業が必要である場合、その作業と保守、問題発生時の対応などが課題となる。また、Web サイトがホスティングサービスを提供する運用会社で一元的に管理されている場合もあり、その場合 Web サーバに対して直接改変を加えることが難しい。

(C-3) ユーザ環境への導入

インターネットを介したオンラインサービスにおいて Web 共有を利用するユーザは不特定多数が対象となる場合がほとんどである。この時、Web 共有を開始する際に何らかの設定作業やインストールが必要となる場合、この作業が複雑で煩わしいとサービスの利用そのものが敬遠される可能性があるため、なるべくユーザ側の設定作業が少なく簡単であることが望まれる。

3. Web サイトへのアクセス方式

まず Web 共有システムは主に次の二つの機能システムに分けることができる。

・同期制御系

ブラウザに行われた操作を他のブラウザに伝搬、適用することで動作を同期させる機能系統

・ページ取得系

各ブラウザが同じ Web ページを Web サイトから取得するための機能系統

前記した課題において(A)双方向性における課題は同期制御系に対する課題であり、(B)トランザクションにおける課題、(C)導入時における課題はページ取得系に対する課題である。

ここで課題(A)は、双方向コミュニケーション基盤における同期制御系であるプッシュサーバ及びプッシュ通信モジュールで対応が可能である。(A-1)操作の制限については、各ブラウザのプッシュ通信モジュールにおいてブラウザの操作情報(パスワードフィールドへの入力など)の送受信を制限することにより実現可能である。(A-2)操作履歴の追跡については、ブラウザの操作情報はプッシュコマンドとして全てプッシュサーバを経由して送受信されるため、その送受信をプッシュサーバ上でログとして記録しておけば、それを証左として各ブラウザの操作を誰が行ったかを確認、証明することが可能である。

しかし、課題(B)及び課題(C)は、Web サイトに配置された Web ページ及び Web アプリケーションに、各ブラウザがどのように接続し、共有するかに関連した課題であり、これは前記したページ取得系をどのように実装するかにより、その得失が発生する。

以下、このページ取得系の実装モデルを挙げ、それぞれのモデルにおける課題(B)、課題(C)の得失について定性的に述べる。以下文中では、課題(B-1)を「単一性」、課題(B-2)を「HTTPS」、課題(C-1)を「既存流用」、課題(C-2)を「運用性」、課題(C-3)を「導入容易性」と記す。なお、導入容易性における評価はページ取得系だけに限定して議論し、同期制御系及びその他の機能(映像、音声など)については本稿では触れない。

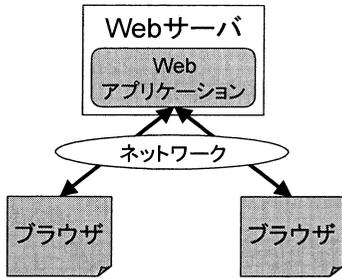


図 2 URL 同期モデル

3.1. URL 同期モデル

各ブラウザの URL のみを一致させ、Web ページの取得処理は各ブラウザで独立に行うモデルである(図 2)。ページ取得系の処理としては一方のブラウザで表示されている URL にもう一方のブラウザを遷移させるだけとなり、Web 共有システムとしては一番簡単な実装となる。

定性的評価

単一性 × Web サイトへは各ブラウザからそれぞれリクエストが送信されることになり、課題で挙げたような商品の二重購入などトランザクションの重複が発生する。また、Web サイトでは各ブラウザに対してそれぞれセッションを割り当てるため、同一のセッションを共有することができない。

HTTPS ○ 第三者が介入しないため問題ない。

既存流用 △ JavaScript などは問題なく動作するが、トランザクションの重複が発生するため、Web アプリケーションで問題が発生する可能性がある。

運用性 ○ 個々の Web サーバへの設定が必要ないため問題ない。

導入容易性 ○ 通常の Web アクセスのみであるため、ユーザの設定作業は必要ない。

3.2. ページ転送モデル

一つのブラウザ(主ブラウザ)で表示している HTML コンテンツを他のブラウザ(従ブラウザ)に直接送付、表示するモデルである(図 3)。

定性的評価

単一性 ○ Web サイトには一つのブラウザでしかアクセスしないため単一性を保証できる。

HTTPS ○ ブラウザ上で暗号解読された HTML コン

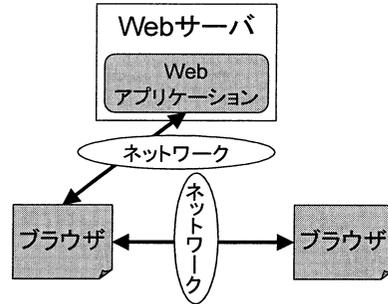


図 3 ページ転送モデル

テンツを送受信するため HTTPS ページも問題なくサポートできる。

既存流用 × Web サイトへ直接アクセスしている主ブラウザでは問題が発生しないが、従ブラウザは全てのコンテンツを主ブラウザから取得する必要があり、従ブラウザ上で動作している JavaScript や ActiveX が動作しない可能性が高い。

運用性 ○ 個々の Web サーバへの設定が必要ないため問題ない。

導入容易性 × 主ブラウザ及び従ブラウザ共に HTML コンテンツを送受信するモジュールを組み込む必要があり、インストールやネットワーク設定作業が必要である。

3.3. Web サーバ制御モデル

Web サーバ上にリクエスト処理モジュールを実装し、各ブラウザからのリクエストを Web サーバ上で一つにまとめ、あたかも一つのブラウザからのリクエストとして処理するモデルである(図 4)。リクエスト処理モジュールの処理は主に以下の二つとなる。

- ・Web アプリケーションでの処理に先立ちブラウザからのリクエストを横取りし、各ブラウザからの同じ URL へのアクセスを一つにまとめる。
- ・URL に対応するレスポンスを受け取り、その URL にアクセスしたブラウザにコピーして返す。

定性的評価

単一性 ○ Web アプリケーションでは一つのブラウザからのリクエストとしてのみ認識され、単一性を保証できる。

HTTPS ○ SSL 通信の終端である Web サーバ上にモジュールを配置するため、通信内容は復号化さ

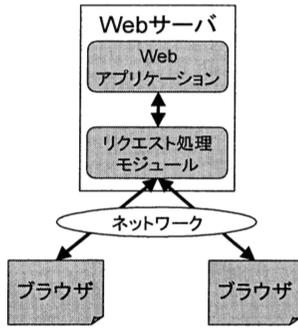


図 4 Web サーバ制御モデル

れており、問題なく HTTPS をサポートできる。

既存流用 △ ブラウザでの Web ページの表示自体には問題はないが、リクエスト処理モジュールの実装によっては OS や Web サーバの変更が必要となり、Web アプリケーションの改変が必要となる可能性がある。

運用性 × 各 Web サーバに対して設定作業が必要となるため導入時の障害となる可能性が高い。また、Web サーバの送受信部分の改変が必要なため、運用中の他の Web アプリケーションに障害が発生する可能性がある。

導入容易性 ○ Web サイト上の処理だけですみ、ユーザの設定作業はほとんど必要ない。

3.4. プロキシ制御モデル

Web サーバ制御モデルで用いたリクエスト処理モジュールをプロキシサーバ上に配置し、各ブラウザのリクエストをプロキシ内で一つにまとめるモデルである(図 5)。

定性的評価

単一性 ○ Web サイトに到達する前にリクエストを一つにまとめるため、Web サイト上のアプリケーションでは一つのブラウザからのリクエストとしてのみ認識され、単一性を保証できる。

HTTPS × プロキシ上では通信内容が暗号化されているため通信を解析、変更することができず、ブラウザ、Web サイト間の HTTPS 通信を処理することができない。

既存流用 ○ 通常のプロキシ処理とほぼ同等であり、問題ない。

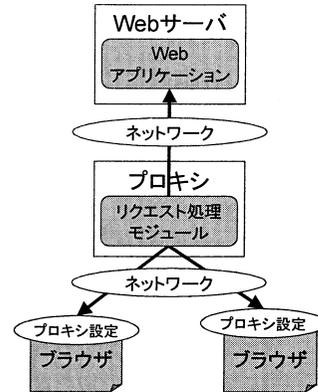


図 5 プロキシ制御モデル

運用性 ○ Web サーバへの設定は必要ないため問題ない。

導入容易性 △ 各ブラウザのプロキシ設定を特別なプロキシサーバ宛にするよう設定しなければならない。また、ブラウザとプロキシ間は直接アクセスできなければならないが、ファイアーウォールなどによりアクセスできない場合もある。また、Web 共有とは関係ない他の通信も特別なプロキシを経由するためセキュリティ上の懸念が発生する。

4. Web スプリッティングモデル

前章で述べた Web サイトへの各アクセス方式を実際のオンラインシステムに適用するにあたり、導入時の課題の観点から URL 同期モデル及びプロキシ制御モデルがその候補に挙げられるが、URL 同期モデルは単一性の問題によりトランザクションを必要とする Web サイトには適用できず、また、プロキシ制御モデルは HTTPS がサポートできないため、セキュリティへの対応に問題がある。

この問題に対応するため我々が考案した手法が Web スプリッティングモデルである(図 6)。システム構成の点ではプロキシ制御モデルとほぼ同じであるが、本モデルではスプリッティングサーバが Web サーバのように振る舞う点異なる。スプリッティングサーバでは、ブラウザからのリクエストがスプリッティングサーバ宛となるよう Web サーバから取得した Web ページ中の URL をスプリッティングサーバ宛のものに変更する。この時の変換規則は概ね下記のように

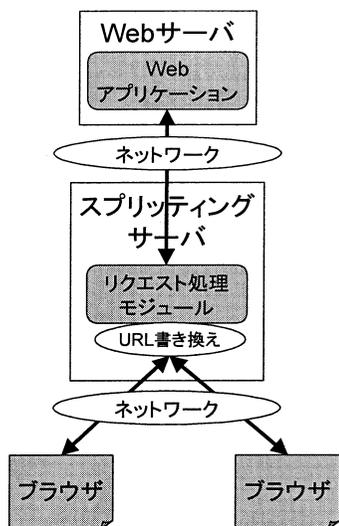


図 6 Web スプリッティングモデル

動作する。

- ・ ページ中に"http[s]://"で始まる URL がある場合、それを"https://スプリッティングサーバ/フラグ/http[s]/元のサーバ名/元のパス/..."の形式を持つ URL に変更する。
- ・ その他の URL(パス指定のみの URL 及び http 以外の URL)に関しては処理しない。

この処理によりブラウザからの HTTP リクエストは全てスプリッティングサーバに送信される。スプリッティングサーバでは HTTP リクエストを受信した際以下の処理を実行する。

- ・ リクエスト URL に前記フラグが含まれている場合、その中から元の URL を取り出し、その URL に従って Web サーバにアクセスする。
- ・ フラグが含まれていない場合、以前のリクエストに基づきアクセスする Web サーバを決定する。

これによりブラウザのリクエストをスプリッティングサーバでまとめることができ、また、HTTPS 通信においてもスプリッティングサーバが SSL 通信の終端となるため問題なく処理可能である。

定性的評価

単一性 ○ 上記した理由により単一性を保証。

HTTPS ○ 同じく上記した理由により問題ない。

既存流用 △ スプリッティングサーバから返される Web ページの URL は全てスプリッティングサーバ宛に変更されているため、ブラウザ上で動作する JavaScript やクッキーで URL の操作をした場合、問題が発生する可能性がある。

運用性 ○ プロキシ制御モデルと同じく個々の Web サーバへの設定が必要ないため問題ない。

導入容易性 ○ ブラウザはページ中に記述された URL に従い、Web スプリッティングサーバにアクセスするだけであり、特別な設定は必要ない。また、他のモデルと違い、URL 表記がスプリッティングサーバを指す URL となるため、共有中かそうでないかが一目で判別可能であるという利点もある。

5. おわりに

以上、Web 共有システムを用いた遠隔支援による対面業務のオンライン化について、主に Web サイトとブラウザ間のページ取得処理についてモデルを提示し、その得失について定性的な検証を実施した。

対面業務のオンライン化に際し、実際に運用可能なモデルは導入時における課題に対応可能な URL 同期モデル及びプロキシ制御モデルであるが、単一性の点で URL 同期は使用できず、また HTTPS への対応の点でプロキシ制御モデルは問題がある。Web スプリッティングモデルは、このような欠点を克服したモデルであるが、本モデルには前記したように既存 Web ページを利用する場合に問題が発生する可能性がある。特に、HTML 中の URL をスプリッティングサーバ経由のものに変更したことによる、ブラウザ上での JavaScript や Ajax, Flash などのローカル処理の動作に異常が出る場合がある。今後このようなローカル処理にも対応可能な Web スプリッティングモデルについて検討する予定である。

参考文献

- [1] 中島、川本、大芝、吉坂、田淵: リアルタイム Web 共有方式による双方向コミュニケーション基盤, 情報処理学会研究報告, Vol.2004, No.2(20040115), pp.57-64
- [2] SHIROSHIMA, SUGIYAMA, ISHIKAWA: Development of Web Consultation System for Cellular Phones, NEC Technical Journal, Vol.1 No.1, March 2006