

利用者のブラウザ操作をWebサービス化するための 開発・最適化支援環境

株式会社富士通研究所
田中 昌弘

■ 背景

- Webサービスを追加開発にコストがかかる課題

■ 関連研究

- 利用者のブラウザ操作からWebサービスを自動生成する手法

■ 提案手法

- 利用者のブラウザ操作から、不要な通信を排除し、再現処理を高速化

■ 実験

■ 考察

■ まとめ

■ マッシュアップ

- ネットワーク上のリソースや機能を組み合わせて新しいWebコンテンツを作成

マッシュアップの例: <http://www.housingmaps.com/>

For Rent For Sale Rooms Sublets

Powered by [craigslist](#) and [Google Maps](#)
(this site is in no way affiliated with craigslist or Google)

City: Price: [Show Filters](#)^{New} [Refresh](#) [Link](#) [About / Feedback](#)

pics	price	description	city	date
	\$650	cuarto independiente	Queens	6/05
	\$600	July 1st 1 bedroom available in a 3 bedroom spacious apartment	Astoria	6/05
	\$450	Looking for one roommate	Staten Island	6/05
	\$480	room for rent	Queens	6/05
	\$450	Small room for rent	Queens	6/05
	\$500	Spacious Room For Rent	Willowick	6/05
	\$500	440ft² - busy professional look for roommate to share downtown apt	White Plains	6/05
	\$750	Big, sunny unfurnished bedroom in a family house	Queens	6/05
	\$450	850ft² - Huge Room - Malden	Long Island, N.Y.	6/05
	\$650	Huge great room in a very nice place	Brooklyn	6/05
	\$350	750ft² - House Share	Bronx	6/05
	\$750	Room in Brooklyn.212-213-11-25	Brooklyn	6/05
	\$700	Beautiful room furnished apt	Brooklyn	6/05
	\$500	Large Room For Rent	Corona	6/05
	\$350	750ft² - Female Roommate Needed	New York	6/05
	\$600	Manhattan / Bronx Room Rental Cable Tv And Internet Access 	Fort Lauderdale	6/05

■ エンタープライズマッシュアップ

- 企業内の複数のWebシステムを統合し、業務を効率化することが目的
- 各WebシステムにWebサービスが公開されていればマッシュアップは容易
 - Webサービスが無い場合は、認証や画面遷移があるため、マッシュアップは困難

顧客管理システム (Salesforce.com)



認証情報と

検索キーワードを送信

社内の在庫検索システム

在庫検索用
Webサービス

在庫結果を自動入力



■ Webサービスが公開されていないWebシステム

- Webサービスの追加開発が必要になる
- Webシステムの修正に伴う仕様書修正・再テストによりコスト大

■ Webサービスの追加開発の低コスト化が課題

■ Enterprise Mashup Markup Language (EMML)

- Webラッパーサービス専用の記述言語
- Webシステムに対してブラウザと同様の通信処理を記述
- 既存のWebシステムを修正することなく、Webラッパーサービスの作成が可能

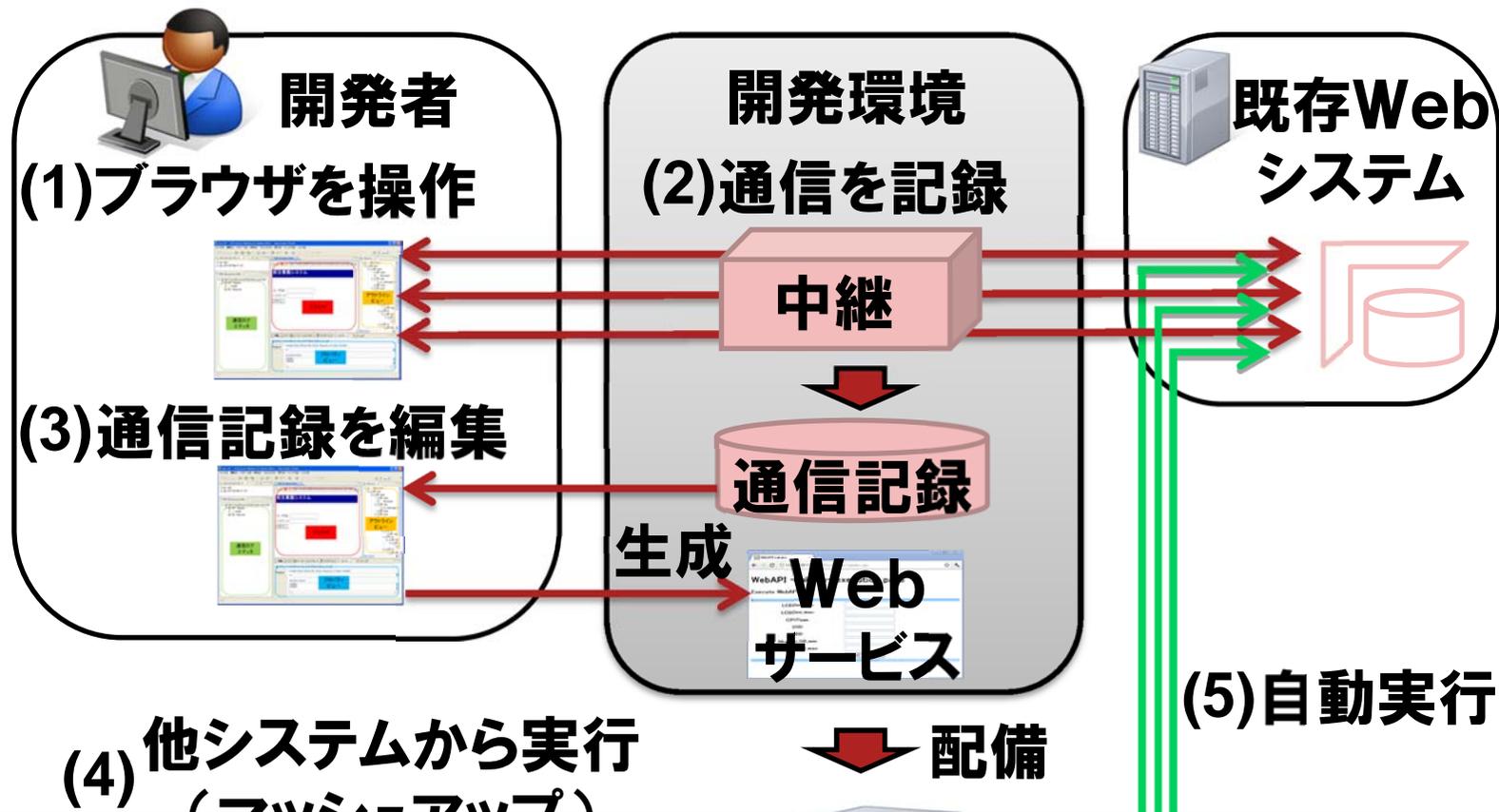
■ 通信処理の記述が煩雑

- 画面遷移や認証方式について詳細な仕様理解が必要
- 記述を容易にするだけでなく、ある程度の自動生成も必要

```
<mashup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="sampleEMML">
  <input name="id" type="string" default="" />
  <input name="pw" type="string" default="" />
  <output name="result" type="string" />
  <variables>
    <variable name="codeString" type="string" />
    <variable name="base64Authorization" type="string" />
  </variables>
  <template expr="{id}:{pw}" outputvariable="$codeString" />
  <script type="text/javascript">
    <![CDATA[
      var s = new Packages.java.lang.String(codeString);
      var b = Packages.org.apache.commons.codec.binary.Base64
        .encodeBase64(s.getBytes());
      var hash = new Packages.java.lang.String(b);
      base64Authorization = "Basic " + hash; ]]>
  </script>
  <display message="hash = " variable="base64Authorization" />
  <variables>
    <variable name="reqheaders" type="document" />
  </variables>
  <constructor outputvariable="reqheaders">
    <headers>
      <Accept>text/html;q=0.9,*/*;q=0.8</Accept>
      <Authorization>{base64Authorization}</Authorization>
    </headers>
  </constructor>
  <directinvoke endpoint="http://sample.com/index.htm"
    method="get" header="$reqheaders" outputvariable="result" />
  <display message="result = " variable="result" />
</mashup>
```

既存研究 ブラウザ操作をWebサービス化

- ブラウザを操作して得られる通信からWebラッパーサービスを生成
 - 詳細な仕様理解はほぼ不要であり、既存のシステムを変更せずに作成可能



M. Tanaka, T. Kume, A. Matsuo, "Web API Creation for Enterprise Mashup,"
IEEE World Congress on Services (SERVICES) 2011.

■ 画像などの不要なWebコンテンツも再現することで実行時間悪化

- コンテンツの種類でフィルタリング可能だが、複数のフレームで構成されたWebページの場合に、不要なフレームどうかの自動判別は困難

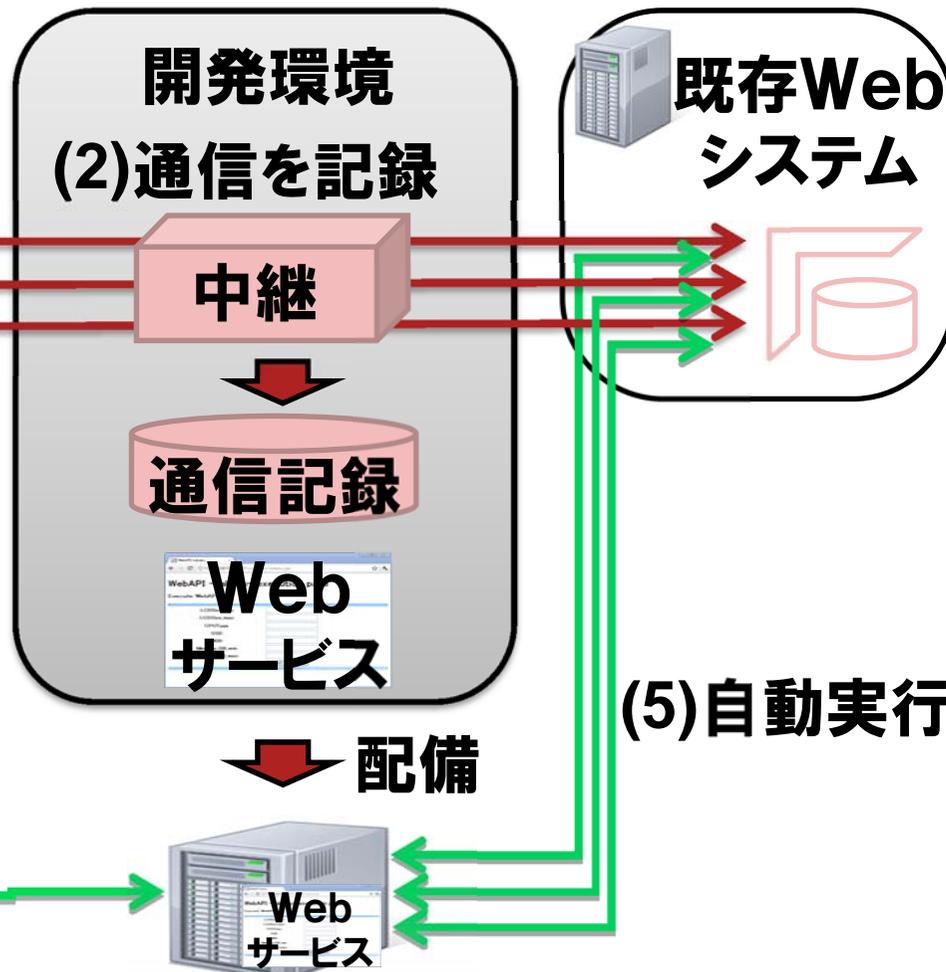
(例) ブラウザに表示された画面



ブラウザが画面表示のために
行った通信



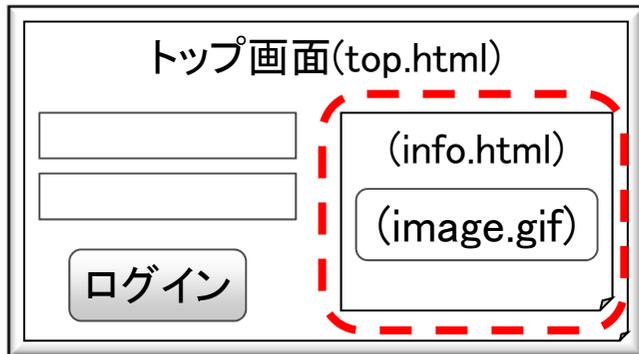
時間



課題の分析

- ブラウザに表示されたページの断片には対応したURLが含まれる

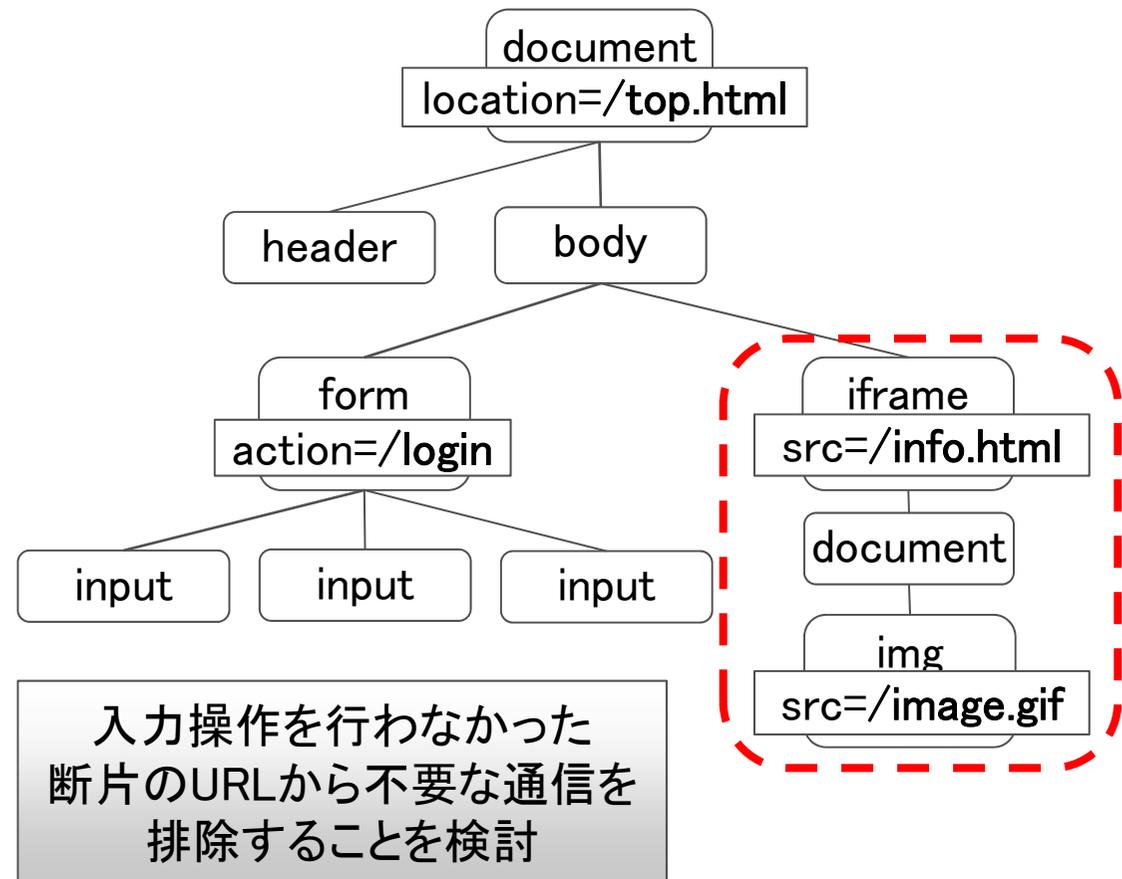
(例) ブラウザに表示された画面



ブラウザが画面表示のために
行った通信



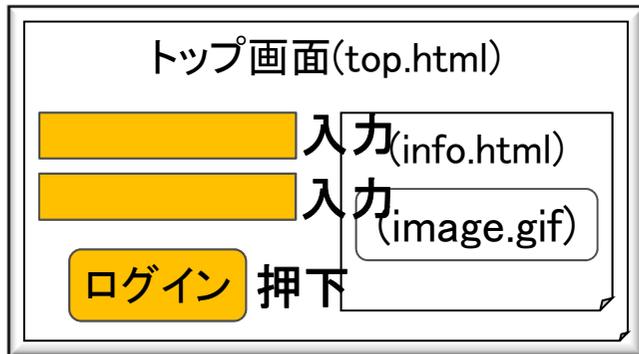
ブラウザ内部のDocument Object Model (DOM)



提案手法

- 開発者の入力操作を監視する機能を付加し、操作を行ったDOMに含まれるURLを抽出して通信のフィルタリングを行う

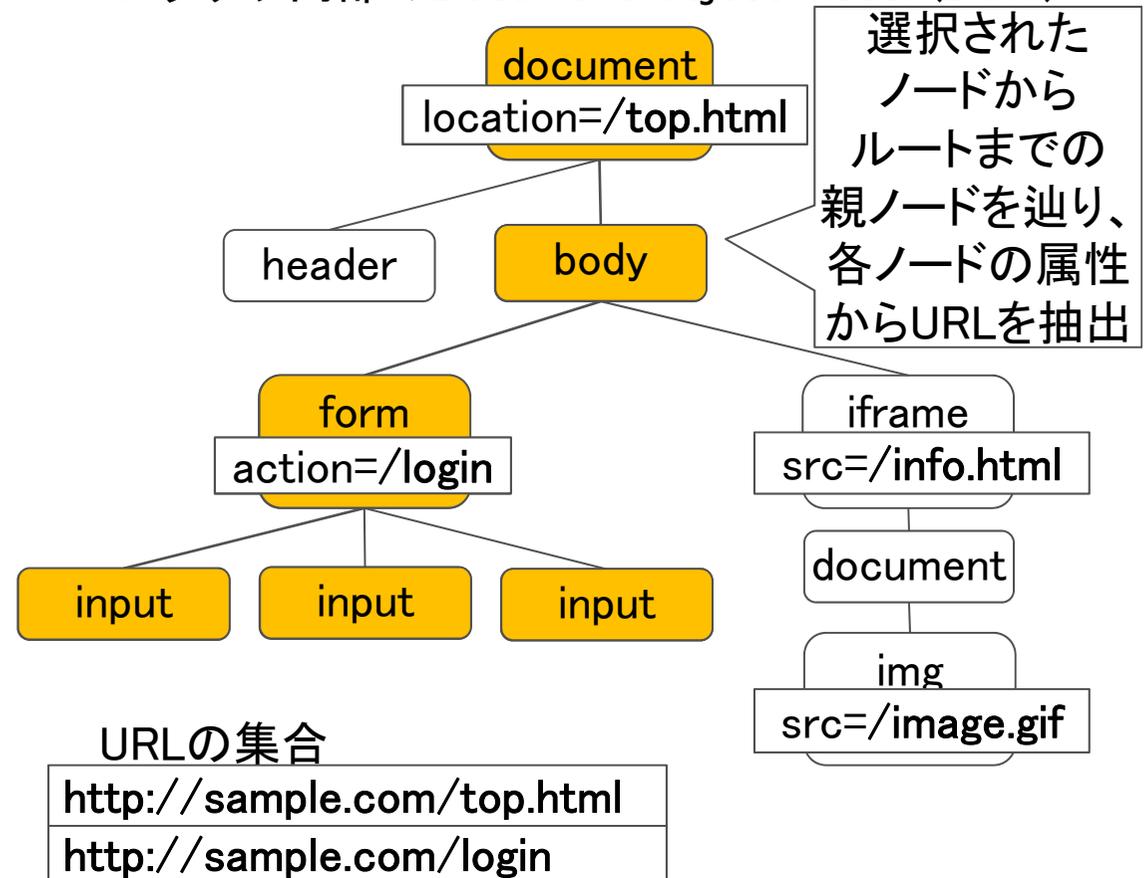
- 入力操作にはマウスの押下やキー入力がある
- (例) ブラウザに表示された画面



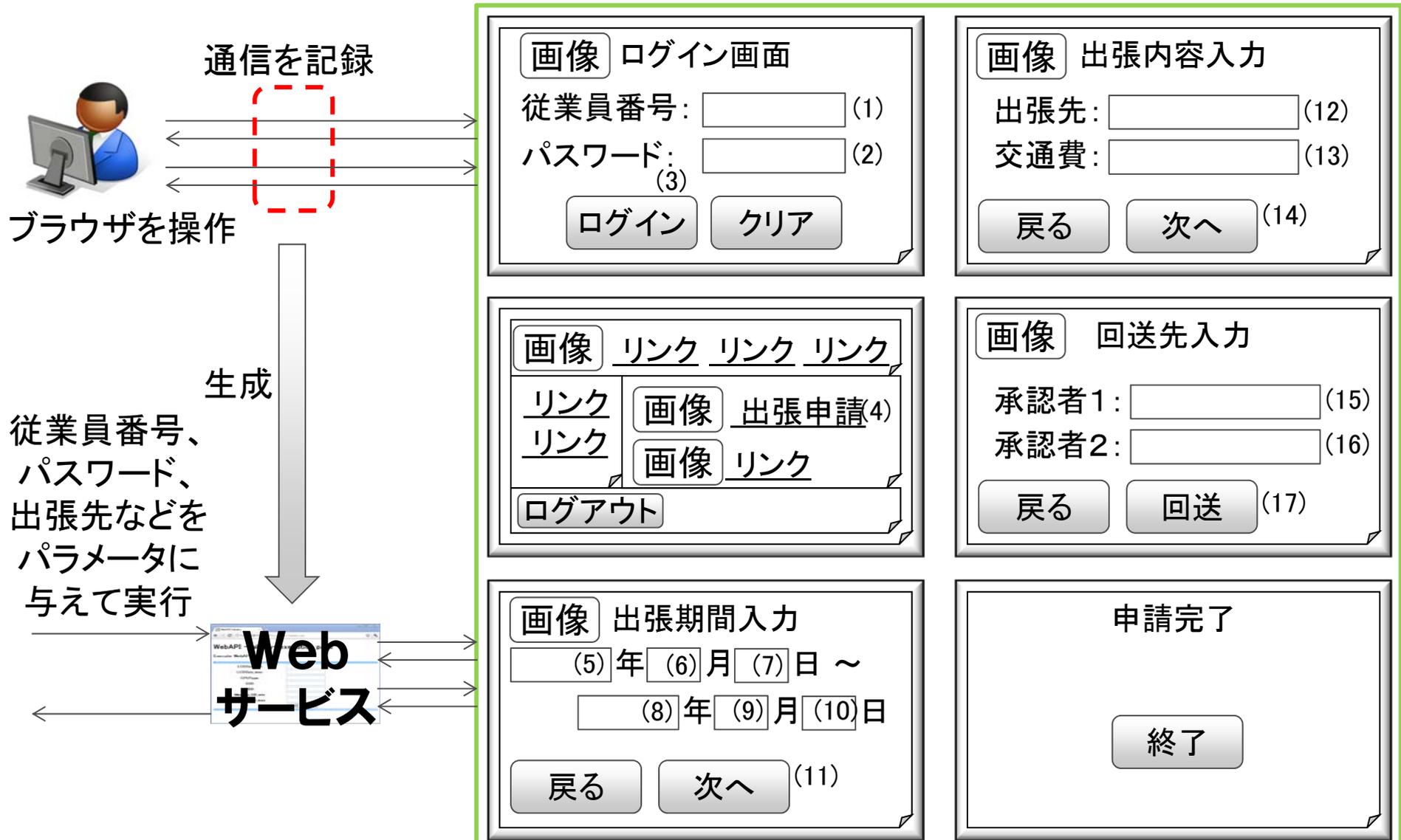
ブラウザが画面表示のために
行った通信



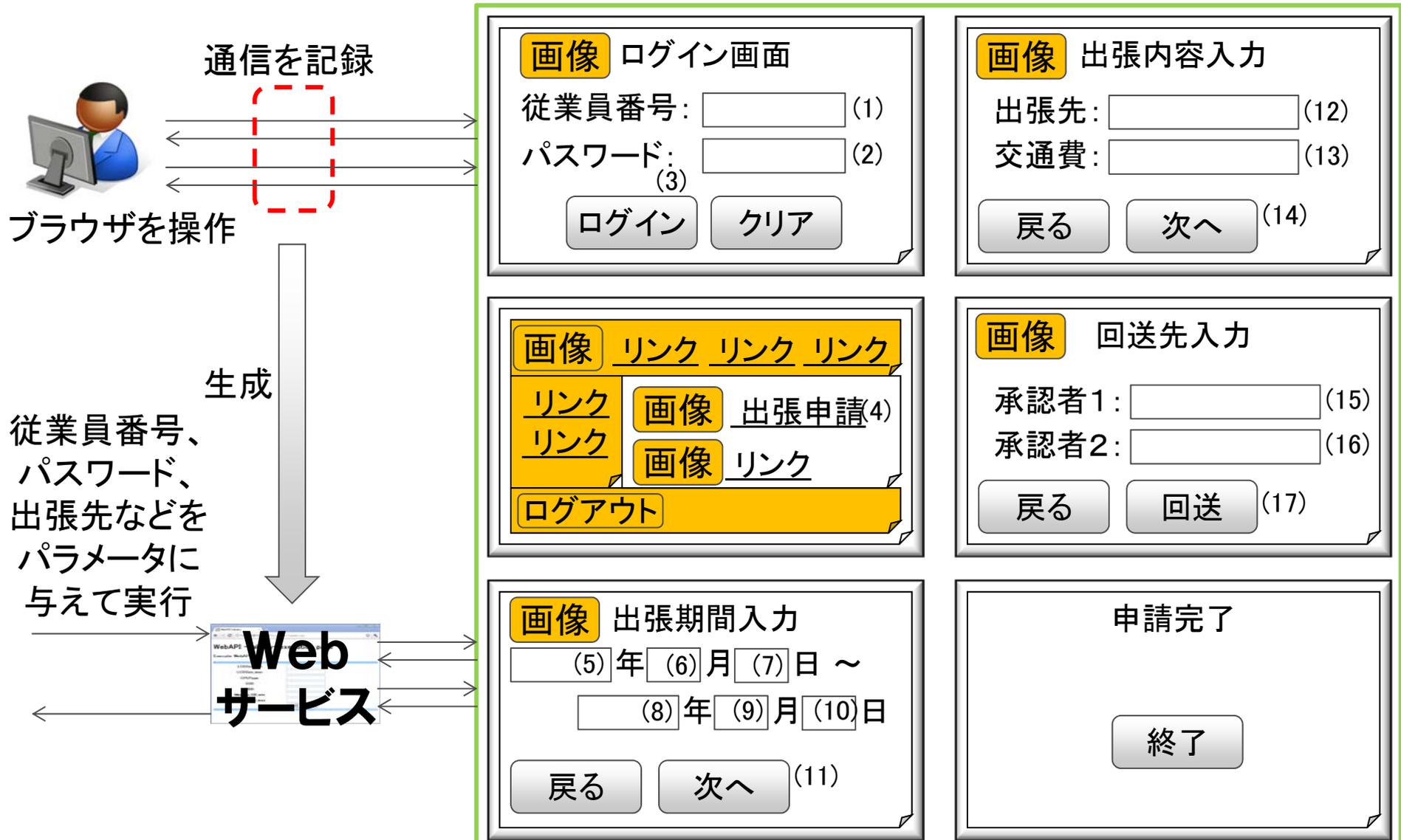
ブラウザ内部のDocument Object Model (DOM)



■ 社内の業務システムで出張申請の操作をWebサービス化



■ 社内の業務システムで出張申請の操作をWebサービス化



■ 本手法の適用前と後で通信回数を比較

- 最小値(動作確認を行いながら通信処理を手動で省いて求めた値)とも比較

コンテンツの種類	通信回数				最小値
	適用前	コンテンツ フィルタ適用	本手法 適用	本手法+ コンテンツ フィルタ適用	
HTML	10	10	6	6	5
画像,スタイル, スクリプト	15	0	2	0	0
合計	25	10	8	6	5

最小値と比較して
余分に抽出したWebページ

画像 ログイン画面
 従業員番号: (1)
 パスワード: (2)
 (3)
 ログイン クリア

入力操作をしたDOMの中に
画像のURLが含まれていたために
余分に抽出した画像コンテンツ

画像 リンク リンク リンク
 リンク 画像 出張申請(4)
 リンク 画像 リンク
 ログアウト

- 開発者の操作に関係の無いHTMLの通信を自動で排除可能
 - とくにフレームやポップアップ画面の通信が多いWebシステムに対して有効
 - 操作したDOMに画像などのコンテンツが含まれている場合は、別途コンテンツの種類で排除
- 手動で求めた最小の通信と比較して余分な通信を含む場合がある
 - 省略してもWebシステムに影響を与えない通信処理は存在する場合がある
 - 通信を省略して影響を与えるかどうかはWebシステムの仕様に依存するため自動的な排除は困難
- 制限事項
 - HTMLで書かれたWebシステムに対して有効
 - JavaアプレットやFlashなどで画面作成が行われたものには無効
 - HTMLの仕様に沿った認証や画面遷移を行うWebシステムに対して有効
 - JavaScriptでセッション情報を更新するような処理が多く記述されている場合、認証に必要なWebコンテンツも、操作に関係が無いと判断して排除する可能性がある
 - 記録した通りの通信のみの再現が可能
 - 途中で画面遷移に分岐が入るものは、別途Webサービスの修正が必要

■ 背景

- エンタープライズマッシュアップ
- Webサービスを追加開発にコストがかかる課題

■ 関連研究

- 利用者のブラウザ操作からWebサービスを自動生成する手法
 - ブラウザの通信を記録し、その通信を再現する処理をWebサービスとして生成
 - 利用者のブラウザ操作に関連しない通信まで再現する課題

■ 提案手法

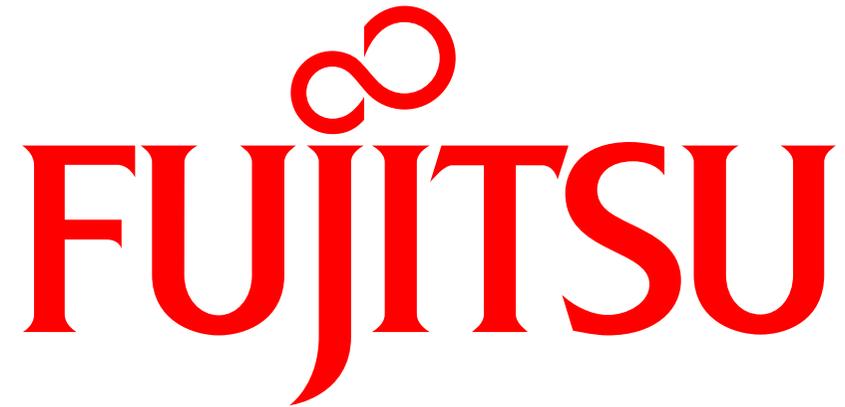
- 利用者(開発者)の入力操作から、不要な通信を排除し、再現処理を高速化

■ 実験

- 社内のOAシステムで出張申請の操作をWebサービス化し、通信回数を比較

■ 考察

- サブページで構成されたHTMLのページに有効
- ログイン画面のような入力画面を表示するだけの通信は排除不可



shaping tomorrow with you