3N-2

Ajax アプリケーションの Flash 変換システムの提案と実装

長谷川　慎哉†　　早川　智一†　　疋田　輝雄†

明治大学理工学研究科†

1. はじめに

今日，Web アプリケーションはデスクトップアプリケーションに近い「リッチ」な見た目と機能をもつ RIA(Rich Internet Application)が主流となっており，幅広く使用されている RIA 技術として Ajax と Flash が挙げられる．Ajax はアプリケーションの閲覧にプラグインが不要である利点があるものの，閲覧に使用するブラウザの種類によって動作が左右され，またアプリケーションを提供する企業の資産であるソースコードがクライアントに公開されてしまうという問題点がある．一方，Flash はアプリケーションの閲覧に専用のプラグインが必要となるものの，アプリケーションの動作がブラウザに依存せず，ソースコードがクライアント側から閲覧不可能である利点をもつ．

本論文では，前述の Ajax がもつ問題を解決すべく，既存の Ajax アプリケーションを Flash アプリケーションに自動変換する手法を提案する．

2. 先行研究との関連

XHTML と CSS で構成される静的な Web ページを Flash に変換するシステムの研究として，後藤ら[1]がある．後藤らは，Flash 変換の処理にオープンソースの RIA 構築用フレームワークである OpenLaszlo[2]を利用した．OpenLaszlo は，独自実装の CSS, JavaScript を含む XML ベースの言語 LZX から Flash を生成することができる．後藤らが提案した変換方式は，XHTML と CSS を LZX に変換し，OpenLaszlo を通して Flash を得るというものである．

本研究では上述の Flash 変換方式を発展させ，JavaScript を含む動的な Ajax アプリケーションの Flash 自動変換を実現する．

3. 変換方式

Ajax アプリケーションを構成する XHTML, CSS, JavaScript をそれぞれ LZX, LZX の CSS, LZX の JavaScript に変換することで，OpenLaszlo を通して Flash アプリケーションを得る．

この変換において，Ajax アプリケーションの見た目や振舞いを LZX でシミュレーションするライブラリ（Ajax 再現ライブラリ）を作成，利用することで，Ajax と LZX との言語間の差異を吸収し，変換処理を単純化する．

4. 変換システムの実装

変換システムはブラウザによるリクエスト，レスポンスを処理するサーブレット，Ajax アプリケーションのパーサ，LZX へのトランスレータ，Flash を生成する OpenLaszlo，LZX による Ajax 再現ライブラリで構成される．

変換システムの処理の流れを図 4.1 に，Ajax から LZX への変換過程を図 4.2 に示す．システムは初めに，ブラウザからのリクエストを受け取ると，入力の XHTML, CSS, JavaScript をそれぞれパースし，XML の木構造にする．次に，変換規則に従ってそれぞれの木を変形し，LZX(CSS, JavaScript を含む) を生成する．最後に，OpenLaszlo にリクエストを転送することで，Flash アプリケーションを生成し，それをレスポンスとしてブラウザに返す．

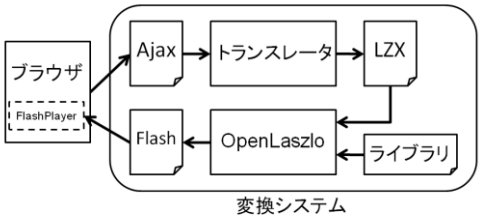

図 4.1　変換システムの処理の流れ

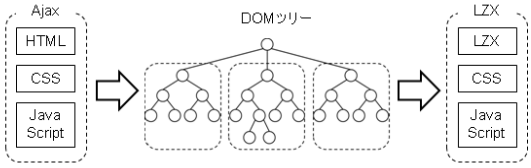

図 4.2　Ajax から LZX への変換過程

Design and Implementation of Flash Translation System for Ajax Applications.
†School of Science and Technology, Meiji University.

LZX による Ajax 再現ライブラリのクラスは，XHTML の要素や CSS のプロパティ，JavaScript で使用される DOM(Document Object Model)関連のオブジェクト，非同期通信のオブジェクト等を，LZX の基本クラスを拡張して実装する．ここで定義されたクラスは，トランスレータによって生成された LZX の要素として使用される．

5. 変換例と実装機能

図 5.1，5.2 にアプリケーションの変換例を示す．変換する Ajax アプリケーションは，ボタンを押したとき，非同期通信でサーバから文字列を取得し，DOM の機能を用いてテキスト入力フィールドに文字列を表示して，背景色を変更するものである．図 5.1 は変換される Ajax アプリケーション，図 5.2 は Ajax アプリケーションを変換して得られる Flash アプリケーションであり，それぞれ，左のウィンドウはボタンを押す前，右のウィンドウはボタンを押した後の画面表示である．また，図 5.3 に Ajax の，図 5.4 に LZX のソースコードの抜粋を示す．

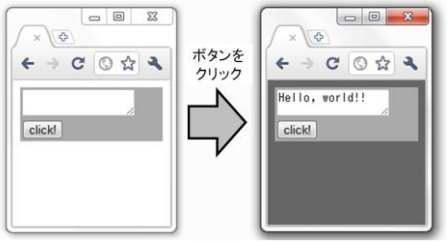

図 5.1　変換前の Ajax アプリケーション

図 5.2　変換後の Flash アプリケーション

```
<html>
 <head>
  <title>Sample</title>
  <link rel="stylesheet" href="sample.css" />
  <script type="text/javascript" src="sample.js">
  </script>
 </head>
 <body>
  <div>
   <textarea id="t"></textarea>
   <button onclick="httpRequest()">click!</button>
  </div>
 </body>
</html>
```

図 5.3　XHTML(Ajax)のソースコード（抜粋）

```
<canvas title="Sample">
 <include href="rpc/ajax.lzx"/>
 <include href="htmlclass.lzx"/>
 <stylesheet src="lzx_sample.css"/>
 <stylesheet src="lzx_default.css"/>
 <script src="lzx_sample.js" />
 <Html>
  <Body>
   <Div>
    <Textarea id="t1" text=""/>
    <Button onclick="httpRequest()">
     <Text text="click!"/>
    </Button>
   </Div>
  </Body>
 </Html>
</canvas>
```

図 5.4　LZX(Flash)のソースコード（抜粋）

図 5.4 の網掛け表示の要素が Ajax 再現ライブラリのクラスである．実装したライブラリの主な機能を表 5.1 に示す．

表 5.1　Ajax 再現ライブラリの主な機能

機能名	説明
H1	見出しを表す XHTML 要素
Button	ボタンを表す XHTML 要素
Text	XHTML のテキストノード
cssfontsize	文字サイズの CSS プロパティ
cssbackgroundcolor	背景色の CSS プロパティ
childNodes	JavaScript オブジェクト Node の子要素のリストを表すプロパティ
getElementsByTagName	JavaScript オブジェクト Node の子要素を探索するメソッド
XMLHttpRequest	非同期通信のための JavaScript オブジェクト

現在，本システムでは XHTML 要素の 66%，CSS プロパティの 55%の変換に対応している．また JavaScript については，非同期通信の機能，DOM の要素探索や属性値の書換え等の機能の変換に対応している．

6. おわりに

本論文では Ajax の問題を解決するための Flash 変換方法と，基本的な自動変換システムの実装方法を示した．現時点では，XHTML 要素を動的に追加，削除する DOM の機能が未実装であり，変換可能な Ajax アプリケーションは単純なものに限られるが，今後，Ajax 再現ライブラリの機能を拡張していくことで，一般の Web 上に公開されている実用的な Ajax アプリケーションの Flash 変換の実現を目指したい．

参考文献

[1] 後藤旭人，早川智一，疋田輝雄：OpenLaszlo による XHTML からの柔軟な Flash 生成システム，情報処理学会第 71 回全国大会，2009.
[2] Laszlo Systems：Documentation | OpenLaszlo, http://www.openlaszlo.org/documentation