3P-3

RIA変換フレームワークを用いた Ajaxアプリケーションの Flash変換手法の実装と評価

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

現在，多くの Web アプリケーションにおいて Ajax 技術が使用されているが，Ajax の構成技術（HTML・CSS・JavaScript）に起因するブラウザ依存性・ソースコードの公開性などが，アプリケーションの用途によっては問題となる．

本論文では，前述の問題を解決する手段として，Ajax アプリケーションを Flash に自動変換する手法を提案する．Ajax アプリケーションの振舞いを記述する JavaScript が変換の難点となるが，変換処理系の実装に早川らの RIA（Rich Internet Application）変換フレームワークを利用し，JavaScript オブジェクトを OpenLaszlo で再現するエミュレーションライブラリを実装することで対処した．実装した Ajax アプリケーションの Flash 変換システムにおいて，HTML タグ・CSS プロパティ・JavaScript オブジェクトの種類毎の変換率を算出し，変換手法の実用性の評価を行った．本システムは前回の報告[1]より高い変換率を示し，課題であった DOM の UI 追加・削除などの機能の変換を実現した．

2. 関連研究

早川他[2]は異なる RIA 技術間の移植性向上のため，中間表現とフレームワークを用いて RIA を自動変換する方法を提案した．中間表現は RIA をメタ情報・ウィジェット情報・スタイル情報・振舞い情報の 4 つに分割した XML で表現し，フレームワークは RIA と中間表現の変換を支援する Java ベースの処理系を提供するものである．このフレームワークはデザインパターンを多用し，変換処理系の差し替えや拡張が容易になるよう設計されているのが特徴である．

本研究では，早川らのフレームワークの処理系を拡張して利用することで，Ajax アプリケーションの Flash 変換を実現する．

Implementation and Evaluation of Flash Translation Method for Ajax Applications by RIA Translation Framework.
†Shinya Hasegawa, Tomokazu Hayakawa, Teruo Hikita
School of Science and Technology, Meiji University.

3. 変換例

Ajax アプリケーションの変換例を図 3.1，図 3.2 に示す．図 3.1 の Ajax アプリケーションは，ボタンを押すとサーバとの非同期通信によって画像の URL を取得し，画像のタグを動的に追加する．図 3.2 は，図 3.1 の Ajax アプリケーションを本システムによって Flash に自動変換したものである．これらの図から，UI や非同期通信などの振舞いをほぼ等価に変換できていることが分かる．

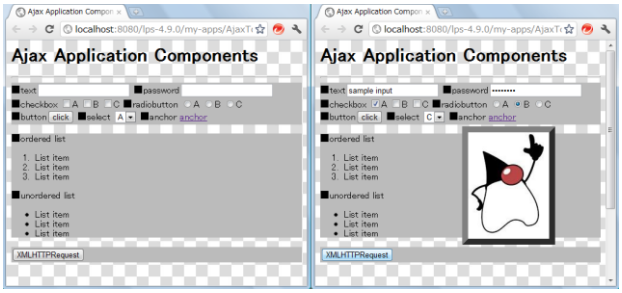

図 3.1　変換前の Ajax アプリケーション

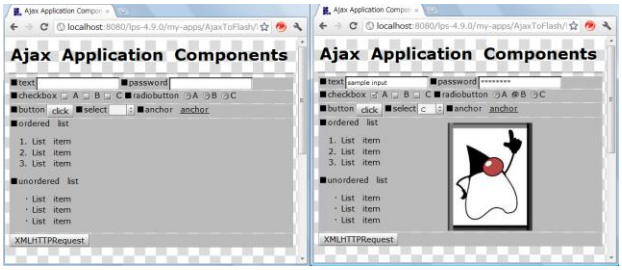

図 3.2　変換後の Flash アプリケーション

4. 変換方式

RIA 変換フレームワークを利用して Ajax アプリケーションを OpenLaszlo[3]の記述言語 LZX に変換し，Flash アプリケーションを得る．図 4.1 に変換の流れを示す．

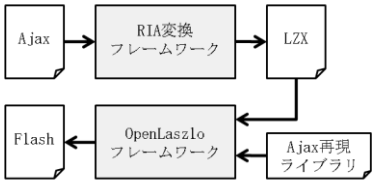

図 4.1　Ajax の Flash 変換方式

Ajax アプリケーションから変換された LZX では，HTML・CSS・JavaScript のエミュレーションを行うライブラリ（Ajax 再現ライブラリ）を使用する．前述した変換例（図 3.1，図 3.2）における，変換前（Ajax）と変換後（LZX）の JavaScript ソースコードの抜粋を図 4.2，図 4.3 に示す．図 4.3 における網掛け表示の部分が，Ajax 再現ライブラリが提供する DOM 関連のオブジェクトである．このように，ライブラリによって Ajax・LZX 間における言語の差異を吸収し，変換処理を単純化する．

```
var httpObj = new XMLHttpRequest();
var target_url = "./imgInfo.xml";
httpObj.onreadystatechange = function(){
  if(httpObj.readyState == 4
   && httpObj.status == 200){
    var img = document.createElement("img");
    img.style.left = "240px";
    img.style.top = "-230px";
    img.style.border = "outset 10px gray";
    var xml = httpObj.responseXML;
    var url = xml.getElementsByTagName("url");
    img.src = url[0].childNodes[0].nodeValue;
    document.body.appendChild(img);
  }
}
```

図 4.2　変換前（Ajax）の JavaScript（抜粋）

```
var httpObj = new lz.XMLHttpRequest();
var target_url = "./imgInfo.xml";
httpObj.onreadystatechange = function(){
  if(httpObj.readyState == 4
   && httpObj.status == 200){
    var img = document.createElement("img");
    img.style.setAttribute("left", "240px");
    img.style.setAttribute("top", "-230px");
    img.style.setAttribute(
       "border", "outset 10px gray");
    var xml = httpObj.responseXML;
    var url = xml.getElementsByTagName("url");
    img.setAttribute("src", url[0].childNodes[0]);
    document.body.appendChild(img);
  }
}
```

図 4.3　変換後（LZX）の JavaScript（抜粋）

5.　変換システムの実装

Ajax アプリケーションから LZX へのソースコードの変換は，RIA 変換フレームワークを拡張した Java ベースの処理系（ApplicationReader・Translator・ApplicationWriter）を用いて行う．RIA 変換フレームワークが行う処理の流れを図 5.1 に示す．ApplicationReader は Ajax アプリケーションのソースコードを読み込み，HTML・CSS を XML 表現に，JavaScript を抽象構文木（AST）表現にそれぞれ変換する．Translator は，生成された XML・AST に対して，HTML の要素名・CSS のセレクタ文字列・JavaScript の演算子やオブジェクト名の置換などを行い，LZX アプリケーションの XML・AST に変換する．ApplicationWriter は，XML・AST 表現から LZX のソースコードを生成する．

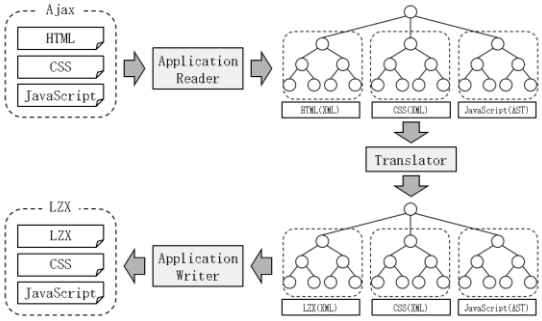

図 5.1　Ajax から LZX への変換過程

6.　変換手法の評価

本システムが変換に対応する主な HTML タグ・CSS プロパティ・JavaScript オブジェクトを表 6.1 に示す．HTML タグは XHTML 1.1 における非推奨タグを除いた 83 種中 64 種（77%），CSS プロパティは CSS 2.1 の音声読み上げや印刷用プロパティを除いた 91 種中 64 種（71%），JavaScript オブジェクトは JavaScript 1.5 で使用可能なオブジェクト（ECMAScript 3・DOM レベル 2・ブラウザ独自の主要なオブジェクト）74 種中 30 種（41%）の変換に対応する．これにより，基本的な Ajax アプリケーションの Flash 変換をほぼ実現できた．

表 6.1　変換可能な主な HTML・CSS・JavaScript

HTML タグ	CSS プロパティ	JavaScript オブジェクト
a	width	Node
div	padding	Document
span	height	CharacterData
img	display	Text
li	color	Element
button	background	HTMLElement
script	margin	CSS2Properties
p	font-size	Window
option	position	XMLHttpRequest

7.　おわりに

本論文では，RIA 変換フレームワークを用いた Ajax アプリケーションの Flash 変換手法の実装と，その実用性を示した．今後，変換処理系やライブラリを拡充することで，実用性を一層高めていきたい．

参考文献

[1] 長谷川慎哉，早川智一，疋田輝雄：Ajax アプリケーションの Flash 変換システムの提案と実装，情報処理学会第 73 回全国大会，2011.
[2] 早川智一，長谷川慎哉，吉賀祥太，疋田輝雄：中間表現とフレームワークを用いた Web アプリケーションのメンテナンス法の提案と評価，DPS149，2011.
[3] Laszlo Systems：Documentation | OpenLaszlo, http://www.openlaszlo.org/documentation