

# 리치 인터넷 애플리케이션 기능의 접근성 사례 조사

최성은\*, 김석일\*

\*충북대학교 컴퓨터학과

e-mail:sweetdream0921@gmail.com

## Case Studies of RIA Function Accessibility

Seong-eun Choi\*, Suk-il Kim\*

\*Dept of Computer Science, Chungbuk National University

### 요 약

이 논문에서는 접근성 있는 리치 인터넷 애플리케이션을 구현하였을 때 우리나라에서 입수 가능한 화면 낭독 프로그램이 이들 리치 인터넷 애플리케이션의 접근성 기능을 충분히 지원하는가를 실험을 통하여 파악하였다. 실험 결과, 미국 Freedom Scientific사의 JAWS for Windows 10.0 한글판이 실험에 사용한 모든 애플리케이션의 접근성 기능을 지원하였다. 그러나 국산 화면 낭독 프로그램은 개발 환경에 따라 접근성 지원 기능이 떨어지고 있음을 확인할 수 있었다.

### 1. 서론

웹은 그동안 여러 과정을 거쳐 진화를 거듭해왔다. 처음에는 단순한 문자 정보를 제공하는데서 출발하였으나 점차 이미지, 사운드, 동영상 등의 멀티미디어를 제공하는 수단으로 발전하였다. 2000년대에 들어와서는 웹 콘텐츠의 사용자 인터페이스 개선과 상호작용의 필요성이 요구되었고, 이에 따라 새로운 리치 인터넷 애플리케이션(Rich Internet Application: RIA) 패러다임이 출현하게 되었다[1].

웹은 여러 가지 한계를 가지고 있다. 웹의 기본 단위가 페이지이기 때문에 근본적으로 복잡한 기능을 수행할 수 없는 구조를 가지고 있고, 각 페이지가 하이퍼링크를 통해 계층적으로 연결되어 있기 때문에 프로세스에 따라 연결된 모든 페이지들을 로딩하는데 많은 시간이 소요된다.

좋은 웹 콘텐츠는 내용, 표현, 기능이 유기적으로 결합하여 사용자가 편리하게 접근하여 사용할 수 있는 콘텐츠를 의미한다. 웹 콘텐츠의 내용을 나타내기 위해서 웹 페이지를 구성하는 기본 언어인 HTML(Hypertext Markup Language)[2]이 사용된다. 구조적으로 짜여진 문서(HTML, XML)에 style(글자, 여백, 레이아웃) 등을 적용하기 위해서는 CSS(Cascading Style Sheets)[3]를 사용하고 사용자 액션에 따른 반응 및 동적인 문서 구조 변경 등 기능을 제공하기 위해서는 스크립트(Script) 언어를 사용한다. 대표적인 스크립트 언어로 JavaScript가 가장 많이 사용되며 JavaScript 1.5 버전은 표준을 지원하는 응용 프로그램에서 동일하게 동작한다.

여기서 JavaScript를 DOM(Document Object Model) Script[4] 또는 ECMAScript[5]라고도 한다. DOMScript는 DOM을 다루는 스크립트라는 의미이며, ECMAScript는 유럽기구(the European Computer Manufacturers Association; ECMA)에서 표준화 한 프로그래밍 언어라는 뜻이다.

HTML과 JavaScript만으로는 윈도우 프로그램의 풍부한 유저 인터페이스의 기능을 표현하는데 한계가 있기 때문에 각 콘텐츠와 사용자가 상호 커뮤니케이션 하는데 있어서 많은 제약이 따른다. 따라서 웹의 간결함과 윈도우 프로그램의 풍부한 유저 인터페이스 기능성을 동시에 추구하는 RIA가 등장하게 되었다.

RIA는 Flash, Flex, Ajax, WPF 등 여러 기술에 의해 구현된다. 이러한 RIA는 기존의 웹 콘텐츠의 사용자 인터페이스와는 다른 새로운 사용자 인터페이스를 사용하고 있다. 뿐만 아니라 기존의 HTML 환경에서는 사용하지 않았던 여러 가지 새로운 컨트롤들을 정의하고 있다. 이러한 새로운 컨트롤들은 장애인, 특히 시각장애인에겐 처음 접하게 되는 컨트롤이므로 사용방법 조차도 생소하다. 따라서 이들 RIA의 경우에 접근성을 제공하기 위해서는 여러 가지 사항을 고려해야 한다.

본 논문에서는 우리나라에서 입수 가능한 화면 낭독 프로그램을 설치한 PC에서 화면 낭독 프로그램이 RIA에 대하여 접근성을 지원하는가를 연구하였다.

## 2. 관련 연구 동향

웹 콘텐츠 저작 기법이 전통적인 HTML 기반의 코딩 방법으로부터 리치 인터넷 애플리케이션(Rich Internet Application: RIA) 방식으로 진화함에 따라, HTML을 기반으로 하는 콘텐츠의 접근성 표준은 RIA 기술을 이용하여 구축한 동적 콘텐츠에 대한 접근성 표준으로 적용될 수 없게 되었다. 이에 따라 World Wide Web Consortium(W3C) 산하 WAI(Web Accessibility Initiative)에서는 웹 콘텐츠에 대한 접근성 표준을 개정하는 한편, RIA에 관한 접근성 가이드라인(Accessible Rich Internet Applications) [6]을 제안하였다.

W3C의 웹 접근성 표준 개정판(WCAG Technical Guidelines 2.0)[7]은 오랜 보완작업을 거쳐 2008년 12월 11일자로 채택되었다. WCAG 2.0과 WCAG 1.0[7]의 가장 큰 차이점은 WCAG 2.0이 특정한 언어나 개발도구를 지칭하지 않고 포괄적인 웹 콘텐츠에 대한 접근성 개념을 제시하고 이를 준수하는 방법을 설명하고 있다는 점이다. 따라서 WCAG 2.0은 HTML 환경 뿐 아니라 RIA 기술을 이용한 웹 애플리케이션 콘텐츠의 경우에도 그대로 적용될 수 있다.

RIA에 관한 접근성 가이드라인(WAI-ARIA) 1.0에서는 접근 가능한 RIA 콘텐츠를 개발하기 위하여 고려해야 하는 규정과, HTML, CSS, JavaScript 및 Ajax 등의 기술을 이용하여 동적 콘텐츠를 개발하는 예를 제시하고 있다. 따라서 WAI-ARIA에서 제시하는 규정에 따라 RIA 콘텐츠를 구현한다면 접근 가능한 RIA 콘텐츠의 개발이 가능하다.

미국은 웹 접근성과 관련한 지침을 미국 장애인 복지법의 수정 조항인 508조[8]에 규정하고 있다. 미국도 웹 환경의 변화에 따라 508조의 개정 작업을 서두르고 있다. 그 개정방향은 W3C의 경우와 마찬가지로 기존의 HTML과 CSS에 국한된 웹 콘텐츠 개발 방법을 탈피하여 스크립트 언어의 사용 추세와 RIA 콘텐츠로의 발전 추세를 감안할 것으로 알려지고 있다.

우리나라의 웹 콘텐츠에 적용되는 접근성 지침은 2005년 12월 제정된 ‘인터넷 웹 콘텐츠 접근성 지침(Internet Web Contents Accessibility Guideline)’[9]이다. 인터넷 웹 콘텐츠 접근성 지침(이하 ‘국가표준’이라고 한다)은 주로 HTML 및 CSS로 작성된 웹 콘텐츠의 접근성에 관한 사항을 주로 다루고 있다. 하지만 풍부한 웹 환경에 JavaScript, Visual Basic 등과 같은 스크립트 코드에 의하여 추가되는 기능이나 스크립트 코드 자체에 대한 접근성 제공 방법에 대해서는 아무런 규정이 없으며, Ajax, Flash, Flex 또는 Silverlight 환경에서 개발된 RIA의 접근성 문제는 다루고 있지 않다.

## 3. 접근성 있는 RIA 제작기법

### 3.1 JavaScript의 접근성

JavaScript 프로그램을 접근성 있게 코딩하기 위하여 다음의 사항을 만족해야 한다.

- (1) 스크립트로 교체하는 이미지는 교체되는 이미지에 알맞게 대체 텍스트도 교체하여야 한다.
- (2) 키보드만으로 콘텐츠가 제공하는 모든 기능에 대한 제어가 가능해야 한다.
- (3) 키보드로 Form 컨트롤, 링크, 버튼 등 객체 사이를 이동할 때 논리적 이동 순서가 적합해야 한다.
- (4) 하위 메뉴를 제공하는 주 메뉴의 경우, 키보드만으로도 하위 메뉴를 이용할 수 있어야 한다.
- (5) 열리지 않는 팝업창을 새 창이 열렸다고 알려주지 않아야 한다.
- (6) 사용자의 입력 없이 창의 크기를 변경하거나 이동시키지 않아야 한다.
- (7) 사용자의 입력 없이 자동으로 발생하는 팝업창을 사용해서는 안된다.
- (8) 사용자의 입력에 의해서 발생하는 팝업창은 미리 팝업창임을 알려야 한다.
- (9) 온라인 서식의 구성이 논리적이어야 한다.
- (10) Tab 키에 의한 이동 순서가 논리적이어야 한다.
- (11) 키보드만으로 온라인 서식의 모든 기능을 이용할 수 있어야 한다.
- (12) JavaScript를 지원하지 않더라도 submit 기능이 동작해야 한다.
- (13) JavaScript 코드를 지원하지 않는 경우에 대체 콘텐츠를 제공하여야 한다.
- (14) 대체 콘텐츠는 간단명료하면서 원본과 동일한 기능을 수행해야 한다.
- (15) 사용하는 JavaScript 코드는 표준을 준수해야 한다.

### 3.2 접근성이 있는 Flex 제작기법

Flex 프로그램을 접근성 있게 코딩하기 위하여 다음의 사항을 고려해야 한다.

- (1) ‘텍스트가 아닌 콘텐츠’에 대해서는 적절한 대체 텍스트를 제공하여야 한다.
- (2) 동적으로 교체된 이미지는 이미지에 알맞게 대체 텍스트도 교체해야 한다.
- (3) 멀티미디어 콘텐츠는 동기화된 자막을 제공해야 한다.
- (4) 색상 이외에도 명암이나 패턴으로 콘텐츠 구분이 가능하도록 구성해야 한다.
- (5) 흰 바탕에 밝은 회색글자처럼 판독이 어려운 색조합을 피하고, 흰 바탕에 검정 글자처럼 대비 차이가 큰 색조합을 사용해야 한다.

- (6) 저시력자나 다양한 환경의 사용자도 텍스트를 쉽게 읽을 수 있도록 적절한 크기로 텍스트를 제공해야 한다.
- (7) 키보드만으로 콘텐츠가 제공하는 모든 기능에 대한 제어가 가능하도록 구성해야 한다.
- (8) Tab 키와 Shift + Tab 키에 의한 이동 순서가 논리적이어야 한다.
- (9) 초점이 주어지는 것만으로 상황이 바뀌어서는 안된다.
- (10) 하위 메뉴를 제공하는 주 메뉴의 경우, 키보드만으로도 하위 메뉴를 이용할 수 있어야 한다.
- (11) 키보드로 폼 컨트롤, 링크, 객체 사이를 이동할 때, 이동순서가 논리적이어야 한다.
- (12) 복잡한 콘텐츠에 대해서는 사용법, 구도, 레이아웃에 관한 정보를 제공해야 한다.
- (13) 콘텐츠를 사용하는데 필요한 플러그인은 사용자가 설치하여 사용할 수 있어야 한다. ※ 설치할 수 없다면 설치 방법을 제공하여야 한다.
- (14) 만일 플러그인의 설치가 불가능하다면 설치 방법을 제시하여야 한다.

### 3.3 접근성이 있는 Flash 콘텐츠 제작기법

Flash 콘텐츠를 접근성 있게 제작하기 위하여 다음의 사항을 고려해야 한다.

- (1) '텍스트가 아닌 콘텐츠'에 대해서는 적절한 대체 텍스트를 제공하여야 한다.
  - (2) 교체된 이미지는 이미지에 알맞게 대체 텍스트도 교체해야 한다.
  - (3) 멀티미디어 콘텐츠는 동기화된 자막을 제공해야 한다.
  - (4) 색상 이외에도 명암이나 패턴으로 콘텐츠 구분이 가능하도록 구성하여야 한다.
  - (5) 흰 바탕에 밝은 회색글자처럼 판독이 어려운 색조합을 피하고, 흰 바탕에 검정 글자처럼 대비 차이가 큰 색조합을 사용해야 한다.
  - (6) 저시력자나 다양한 환경의 사용자도 텍스트를 쉽게 읽을 수 있도록 적절한 크기로 텍스트를 제공해야 한다.
  - (7) 깜빡이는 콘텐츠가 있을 경우 사전 경고를 해야 한다.
  - (8) 키보드만으로 콘텐츠가 제공하는 모든 기능에 대한 제어가 가능하도록 구성해야 한다.
  - (9) 초점이 주어지는 것만으로 상황이 바뀌어서는 안된다.
  - (10) 무비의 움직임이 웹 페이지의 갱신을 유발해서는 안된다.
  - (11) 키보드만으로 비디오 또는 오디오 재생 컨트롤을 사용할 수 있어야 한다.
  - (12) 배경음악의 크기를 사용자가 조절할 수 있어야 한다.
  - (13) 키보드로 폼 컨트롤, 링크, 객체 사이를 이동할 때 이동 순서가 논리적이어야 한다.
  - (14) 복잡한 Flash 콘텐츠는 사용법에 대한 대체 텍스트를 제공해야 한다.
  - (15) Flash 콘텐츠가 접근성을 제공하지 못할 경우 대체 페이지를 제공해야 한다.
- (16) 불필요한 Flash 콘텐츠는 초점이 주어지지 않도록 해야 한다.
  - (17) 콘텐츠를 사용하는데 필요한 플러그인은 사용자가 설치하여 사용할 수 있어야 한다.  
※ 설치할 수 없다면 설치 방법을 제공하여야 한다.

## 4. 실험 및 고찰

### 4.1 실험

본 논문에서는 RIA의 내용구성, 화면구성, 동작(기능)의 3가지 요소 중 기능에 대한 접근성 준수 여부에 대한 사례조사를 실시하였다. 실험은 우리나라에서 구입할 수 있는 화면 낭독 프로그램인 센스리더 Professional Edition, 드립보이스 7.0 및 Jaws for Window 10.0(JAWS)의 세 가지를 사용하였다.

실험에 사용된 애플리케이션 콘텐츠는 RIA 환경 별로 자주 활용되는 콘텐츠 구현 사례를 선정하였다. 실험에서 사용된 애플리케이션은 개발 환경이 제공하는 한 접근성을 지원하는 방법으로 구현하였다.

- 가로 1단 메뉴란 주 메뉴만을 가로로 구성한 메뉴로 Tab 키를 이용하여 메뉴간의 이동이 가능하며 초점이 위치한 메뉴에서 Enter 키를 누르면 해당 메뉴와 연결된 링크로 이동한다.
- 가로 2단 메뉴는 주 메뉴에 초점이 주어지면 하위 메뉴가 자동적으로 펼쳐지며 Tab 키를 누를 때마다 하위 메뉴를 마지막까지 이동한 후에 다음번 주 메뉴로 이동하도록 구성된 메뉴이다.
- 로그인 예제는 아이디 입력상자, 비밀번호 입력상자 및 로그인 버튼 등으로 구성된 온라인 서식으로 Tab 키에 의한 초점 이동 순서가 화면에 나타난 순서와 일치한다.
- 회원가입 예제도 온라인 서식의 하나로, 회원가입 시 요구되는 정보를 Tab 키를 사용하여 차례로 입력을 할 수 있도록 구성된 예제이다.
- 롤링배너는 좌우 또는 상하로 이미지 배너를 순차적으로 교체하는 컨트롤로, 마우스를 사용할 경우에는 좌우 또는 상하 이동 버튼을 클릭하여 사용하며 Tab 키를 이용할 경우에는 배너를 하나씩 순서대로 이동하므로 이동 버튼을 이용하지 않는다.
- 자동 스크롤 배너란 마우스 포인터를 배너에 위치시키면 해당 배너 이미지가 전면에 나타나고 그것을 클릭하는 경우에는 해당 링크로 이동하는 컨트롤이다. 이것은 Tab 키를 누를 때마다 순방향으로 초점이 이동하면서 해당 배너가 전면에 나타난다.
- 차트는 Flex에만 해당하는 예제로 각종 데이터를 막대 그래프, 파이 차트, 꺾은선 그래프를 묘사할 경우에 사용된다. 차트는 색상을 배제하더라도 충분히 인지할 수

있도록 표현되고 대체 텍스트를 제공하여 차트의 정보를 보조 기술 사용자에게 알려줄 수 있어야 한다.

우리나라에서 구입할 수 있는 센스리더 Professional Edition, 드림보이스 7.0 및 Jaws for Window 10.0의 세 가지의 화면 낭독 프로그램을 사용하여 애플리케이션 콘텐츠 사례의 접근성 평가를 수행하였으며, 구현된 애플리케이션은 키보드 내비게이션에 따른 초점 이동문제와 초점이 제공된 객체에 대한 내용 읽기 기능을 위주로 평가하였다.

#### 4.2 실험 결과 및 분석

표 1은 16개의 샘플 애플리케이션에 대한 세 가지 화면 낭독 프로그램을 이용한 접근성 평가 결과이다.

<표 1> 애플리케이션 사례별 접근성 평가 결과

적용 환경	구현 콘텐츠 사례	센스 리더	드림 보이스	Jaws for window
Java-Script	가로 1단 메뉴	○	○	○
	가로 2단 메뉴	X	X	○
	로그인	○	○	○
	회원가입	○	○	○
	게시판	○	○	○
	롤링 배너	○	○	○
	자동 스크롤 배너	○	○	○
Flex	로그인	○	X	○
	회원가입	○	X	○
	게시판	X	X	○
	롤링 배너	○	X	○
	차트	○	X	○
Flash	가로 1단 메뉴	○	X	○
	가로 2단 메뉴	X	X	○
	자동 스크롤 배너	X	X	○
	롤링 배너	○	X	○

실험 결과 JAWS for Windows는 모든 애플리케이션이 필요로 하는 접근성 기능을 모두 지원하였다. 그러나 국내 화면 낭독 프로그램인 센스리더와 드림보이스는 RIA 접근성 지원 기능이 부분적으로 미흡함을 확인할 수 있었다. 특히 드림보이스의 경우에는 Flex 및 Flash 애플리케이션의 접근성 지원 기능이 크게 떨어져 이 부분의 대폭적인 기능 개선이 필요할 것으로 보인다. 센스리더의 경우에는 일부 수정으로 RIA 접근성을 지원할 수 있을 것으로 보인다.

#### 5. 결론

이 논문에서는 RIA 개발 환경인 Ajax, Flex 및 Flash에서 구현한 접근성 있는 애플리케이션을 국내에서 입수

가능한 화면 낭독 프로그램을 이용하여 접근성 지원 여부를 평가하였다.

실험에 앞서 RIA 개발 환경별로 우리나라 웹 접근성 국가표준에 의거하여 접근성 있는 애플리케이션 제작기법을 마련하였으며, 이를 토대로 평가에 사용할 애플리케이션을 구현하였다.

연구 결과, 국내에서 개발된 화면 낭독 프로그램의 RIA 접근성 지원 기능이 미국 제품에 비하여 크게 떨어지는 것을 확인할 수 있었다. 특히 드림보이스는 Flex 및 Flash 접근성 지원 기능이 거의 제공되지 못했다. 센스리더의 경우에도 부분적인 기능 개선이 필요하다.

앞으로 화면 낭독 프로그램의 RIA 지원 기능 개선에 관한 연구가 이루어져야 할 것으로 보인다.

#### 사사

이 논문은 한국정보문화진흥원의 지원을 받았음.

#### 참고문헌

- [1] 신현상 “개발자를 위한 플렉스 2 실무테크닉 148” 성안당
- [2] W3C Recommendation(1999.12.24) “HTML 4.01 Specification” <http://www.w3.org/TR/html401/>
- [3] W3C Candidate Recommendation(2007.7.19) “Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification” <http://www.w3.org/TR/CSS21/>
- [4] Jeremy Keith “다이내믹한 웹 표준 사이트를 위한 DOM스크립트” 에이콘
- [5] Standard ECMA-262(1999.12) “ECMAScript Language Specification” 3rd edition
- [6] W3C(2008.8.6) “WAI-ARIA 1.0 W3C Working Draft”
- [7] W3C Recommendation(2008.12.11) “Web Content Accessibility Guidelines 2.0” <http://www.w3c.org/TR/WCAG20/>
- [8] W3C(1999.5.5) “Web Content Accessibility Guidelines 1.0” <http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505>
- [9] The Rehavation Act “Section 508” [www.section508.gov](http://www.section508.gov)
- [10] 정보통신부(2006). “인터넷 웹 콘텐츠 접근성 지침,” 한국정보통신표준: KICS.OT-10.0003