

개선된 웹 접근성 평가도구의 설계 및 구현

문경실* · 최정연** · 박수현**

*동서대학교 디자인&IT 전문대학원, **동서대학교 컴퓨터정보공학부

Implementation and design of Improved Evaluation Tool for Web Accessibility

Kyung-sil Moon* · Jung-Yun Choi** · Su-hyun Park**

Dongseo University Graduate School of Design&IT, **Dongseo University Department of

Computer Engineering

E-mail : siri1210@hanmail.net, choi69hoho@nate.com, subak@dongseo.ac.kr

요 약

웹 접근성(Web Accessibility) 평가를 위하여 웹 접근성 표준 가이드라인에 기반 한 120여개의 자동화 평가 도구들이 존재한다. 국외 평가도구는 WCAG 1.0과 미국 재활법 508조를 기준으로 만들어진 A-Prompt, Bobby 등이 있으며, 국내에는 KWCAG 1.0에 기반하여 한국정보문화진흥원에서 제작한 KADO-WAH가 유일한 자동화 웹접근성 평가도구이다. KADO-WAH는 국외 평가도구와 비교하여 상세한 검사를 진행하지 않으며 자동화 평가 범위가 한정되어 있어 상당부분 전문가 평가를 필요로 한다. 이를 보완하기 위하여 개선된 웹 접근성 평가도구의 개발이 필요하다.

본 논문에서는 국내 자동화 평가도구인 KADO-WAH의 기능을 향상시키기 위하여 국외 자동화 평가도구와 비교·분석을 통해 KADO-WAH의 문제점을 제시하고, 이 문제점을 개선한 웹 접근성 평가도구를 구현한다.

키워드

웹 접근성, 웹 접근성 평가도구, W3C, WCAG, KWCAG, KADO-WAH, A-Prompt, Bobby

1. 서 론

정보 통신의 발달과 인터넷의 빠른 확산은 일상생활뿐만 아니라 교육, 금융, 행정 등 다양한 분야에서 웹의 사용을 일반화 시켰다. 웹은 이전에 경험하기 힘들었던 정보에 접근할 수 있는 강력한 도구이지만, 장애인, 노인 등 장애를 가진 사람들이 웹으로부터 정보를 획득하는데 있어 많은 문제점이 있다.

따라서 정보 일반화의 관점에서 누구나 쉽게 웹으로부터 원하는 정보를 얻을 수 있는 환경을 제공하는 것이 필요하다. 이를 위해서 웹 접근성 높은 웹 페이지를 개발하고, 기존의 웹 페이지는 접근성 평가 및 수정을 통하여 웹 접근성을 높이고자 한다.

국내에서는 W3C의 WCAG 1.0(Web Contents Accessibility Guideline1.0)을 기반으로 국내 실정에 맞게 재구성한 한국 웹 접근성 지침 KWCAG 1.0 (Internet Web Contents Accessibility Guideline 1.0)을 만들어 웹 접근성을 준수하도록 유도하고 있다. 이에 웹 접근성을 지키고 있는지

여부를 확인하기 위한 방법으로 웹 접근성 평가를 실시한다. 하지만, 웹 접근성 평가는 평가 기준이 되는 지침 항목의 모호성으로 인하여 컴퓨터 프로그램으로 구현하는데 어려움이 있고, 자동화 평가도구를 이용하여 처리할 수 있는 항목이 많지 않아, 이를 보완하는 전문가 평가에 대부분 의존하고 있는 실정이다.

국내의 웹 접근성 자동화 평가도구는 한국정보문화진흥원의 KADO-WAH가 있는데, 이는 평가시간을 단축 시켜 주는 편리함은 있지만 국외 평가도구에 비하여 상세한 검사를 진행하지 않으며 자동화 평가의 범위가 좁다. 따라서 웹 접근성 평가의 편리함과 정확한 결과를 위해서 자동화 평가도구의 개선이 필요하다.

따라서 본 논문에서는 KADO-WAH의 문제점을 분석하고, 국외 평가도구들과의 비교 및 테스트를 통하여 평가 항목을 재정립한다. 그리고 KADO-WAH의 단점인 한정된 자동화 평가 범위를 보완하고 오작동 하는 항목의 수정을 통하여 보다 개선된 기능의 자동화 평가도구를 설계하고 구현한다.

II. 기존 연구

2.1 웹 접근성 평가 방법

국제 표준화 기구인 W3C에서 1999년 5월 권고안으로 발표한 WCAG 1.0은 14개 지침 65개의 세부 항목으로 구성되어 있다[1]. 2005년 정보통신부에서는 WCAG 1.0와 미국 재활법 508조를 부분적으로 참고하여 국내 실정에 맞게 재구성한 KWACAG 1.0(Korea Web Contents Accessibility Guideline 1.0)을 발표하였다[2]. 이는 국내 웹 접근성 평가의 기준이 된다.

웹 접근성을 평가하는 방법은 자동화 평가도구를 이용한 방법과 전문가가 직접 평가하는 방법이 있다. 자동화 평가도구의 제한성으로 인해 만족할 만한 웹 접근성 평가를 위해서 현재 웹 접근성 평가의 대부분은 전문가에 의한 수동 평가로 이루어지고 있다. 전문가 평가는 한정된 집단에 의한 수동평가로 상세하고 정확한 결과 값이 나오지만, 평가자간의 평가 기준 차이와 과도한 평가 시간, 한정된 집단에 의존하는 단점이 있다.

자동화 평가도구를 이용한 평가는 객관성과 정량화 된 결과 값을 얻을 수 있어 평가자들이 선호하고 있다. 하지만 실제 평가에 있어 항목의 모호함으로 인한 평가 프로그램 구현에 문제가 있어 만족할 만한 평가를 수행하지 못한다[3].

현실적으로 웹접근성 가이드라인을 전부 평가할 수 있는 자동화 평가도구를 구현하는 것은 불가능하다고 볼 수 있다. 하지만 전문가 평가의 문제점을 보완하기 위해서는 보다 향상된 자동화 평가도구의 구현이 절실하다고 할 수 있다.

2.2 기존 웹 접근성 평가 도구

KADO-WAH는 한국정보문화진흥원에서 만든 국내의 대표적인 웹 접근성 평가도구이다. WCAG 1.0, Section 508, KWACAG 1.0을 기준으로 124개 체크항목을 작성하여 평가한다. 다수 페이지 및 폴더(디렉토리)별로 평가가 가능하며, 접근성 미비 위치 및 통계 값을 포함한 보고서를 제공한다[4, 5].

A-Prompt는 웹 개발자들 또는 웹 평가자들이 웹 페이지의 접근성을 평가하고 수정할 수 있게 개발된 프로그램으로 캐나다 토론토 대학의 ATRC(Adaptive Technology Resource Centre)와 미국 위스콘신 대학에 있는 TRACE Center가 공동 개발하였다[6]. A-Prompt는 56개 세부 항목으로 구성되어 있으며 평가 결과에 대한 보고서를 제공하지 않는다. 하지만, 수정하여야 할 오류항목을 부분으로 보여주고 바로 수정하여 확인 가능한 편리한 기능을 제공한다. 한국정보문화진흥원에서는 토론토 대학의 ATRC 연구소와 협약하여 KADO-WAH 이전에 A-Prompt 한국어 버전을 제작하여 배포하였다.

III. 전체 시스템 분석

3.1 KADO-WAH의 문제점 분석

KADO-WAH는 WCAG 1.0, KWACAG 1.0, 미국 재활법 508조를 토대로 만들어진 111개의 체크항목에 따라 웹 접근성 평가 결과를 출력해주는 방식으로 구현 되어있다. KADO-WAH가 가지고 있는 문제점을 분석하여 보면 다음과 같다.

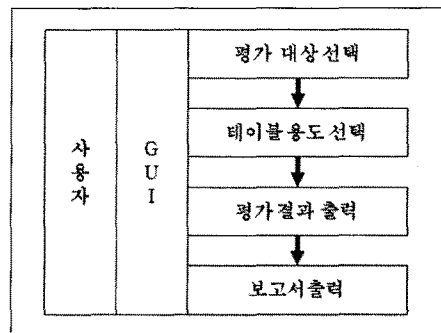
첫째, 3, 8, 61번 체크항목은 정확한 결과를 출력하지 못한다. 3번 항목의 경우 모든 alt속성의 확장자가 bmp, gif, png ... cpt 등 총 187개의 확장자를 대상으로 경고를 출력 하도록 정의 되어 있다. 하지만, 테스트 실행 결과는 bmp 확장자만 평가하여 정확하지 않은 결과값을 출력한다.

둘째, 53, 54, 57, 66, 67번 체크항목은 '테이블 요소의 용도를 프로그램 적으로 알 수 없기 때문에 모든 테이블 요소에 대해 결과를 출력 한다'고 되어있다[4]. 이는 국의 평가도구인 A-Prompt의 기능인 테이블 용도 선택 기법을 이용하여 모든 테이블이 아닌 데이터 테이블과 레이아웃 테이블에 따로 결과를 출력해줌으로써 중복된 결과를 줄이고 더욱 세부적인 결과를 출력할 수 있도록 수정되어야 한다.

셋째, KADO-WAH는 그들이 정의한 규칙을 수정할 수 있는 기능을 제공하지 않는다. 예를 들면 대체 텍스트의 길이가 255자가 초과 되면 메시지를 발생시키는 체크 항목이 있다. 하지만 255자로 설정된 값을 변경할 수 없고, 이러한 기준은 평가자나 평가 집단에 따라 주관적이므로 수정이 가능해야한다. 이는 국의 웹 접근성 평가도구인 A-Prompt, Wave, Bobby도 해당된다.

3.2 WAET 설계

WAET(Web Accessibility Evaluation Tool)는 KADO-WAH의 단점들을 보완하고 더욱더 향상된 웹 접근성 평가를 위해 <그림 1>과 같이 크게 4단계의 처리 순서로 설계하였다.



<그림 1> 처리 순서 설계도

제 1단계 평가 대상 선택은 웹 페이지의 URL을 입력하는 방법과 디스크상의 파일 경로를 입력하는 방법 두 가지 모듈을 지원하도록 설계하

였다.

제 2단계로는 테이블 요소의 용도 선택이다. 웹 페이지의 테이블 요소의 용도는 데이터 테이블과 레이아웃 테이블 두 가지로 구성된다. 그러나 테이블 요소는 용도를 명시적으로 표현하지 못하기 때문에 프로그램 적으로 구별을 할 수가 없다. 또한, 웹 접근성 평가 시에는 데이터 테이블과 레이아웃 테이블의 평가 기준이 다르기 때문에 테이블 요소의 용도를 구별 해주어야 한다.

KADO-WAH에서는 이러한 단계를 거치지 않기 때문에 모든 테이블에 대해 데이터 테이블과 레이아웃 테이블에 대한 평가를 함으로써 중복되고 상세하지 못한 결과를 출력 해준다. 따라서 WAET에서는 제 2단계에서 모든 테이블 요소를 리스트로 출력한 뒤 리스트의 테이블을 웹에 어떻게 나타나는지를 확인하고 테이블 요소의 용도를 선택하여 KADO-WAH보다 더욱 상세하고 중복적이지 않은 결과를 출력할 수 있도록 설계하였다.

제 3단계로는 평가 결과 출력이다. A-Prompt와 같은 도구는 평가 결과의 코드상의 위치를 나타내지 못한다. 단지 어떤 체크 항목에 대해 몇 개의 오류가 있는지를 출력해 주기 때문이다. WAET의 평가 결과는 테이블, 텍스트 필드 두 개의 컨트롤을 이용하여 출력 한다. 사용자가 출력된 평가 결과 테이블의 행을 클릭하면 소스코드 텍스트 필드에서는 어떤 부분인지를 표시 할 수 있도록 설계하였다.

제 4단계는 보고서 출력 부분으로 평가 결과를 집계하여 체크 항목별로 수치로 출력해 주도록 설계하였다. 이러한 방식의 보고서는 체크 항목별 오류 개수를 출력해주는 A-Prompt와 유사하다.

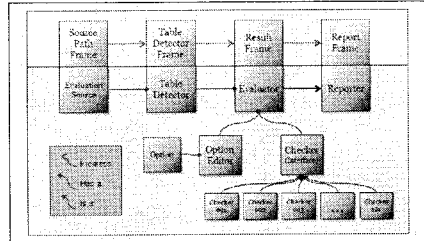
그 외의 향상된 기능으로는 설정 편집기가 있다. 설정 편집기는 각각의 체크 항목 내에서 평가 기준이 되는 값들을 사용자가 편집할 수 있도록 해주는 기능이다. 예를 들면 대체 텍스트가 midi, mid, rmi, mpg등으로 끝날 때 보고되는 체크 항목이 있다. 이러한 체크 항목은 새로운 포맷이 생기게 되면 그에 따라 값이 변경될 수 있으므로 편집이 가능하도록 설계하였다.

IV. WAET 구현

WAET의 구현을 위해 Windows XP Professional Service Pack 2의 환경에서 JDK1.6을 이용하여 구현하였다. 그리고 HTML문서의 파싱을 위해서 Java로 구현된 오픈 소스 파서인 Jericho Parser 2.5를 사용하였으며, GUI 인터페이스 구현은 Java Swing을 이용하였다.

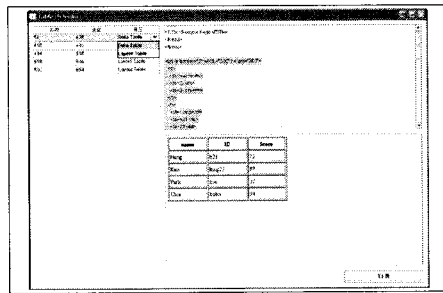
4단계의 처리 흐름과, 각각의 체크 항목에 대한 로직, 설정 편집기능을 포함하기 위해 <그림 2>와 같이 클래스를 구현하였다. 평가 대상 선택은 SourcePathFrame, 테이블 용도 선택은 Table

-DetectorFrame, 평가 결과 출력은 ResultFrame, 보고서 출력은 ReportFrame로 구현 되었고, 각각의 체크항목은 Checker로 구현 되었으며, 설정 편집기는 OptionEditor로 구현 되었다.



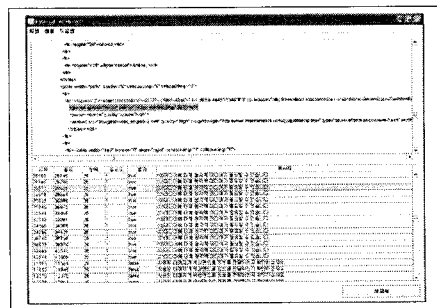
<그림 2> 클래스 다이어그램

<그림 3>은 평가 대상 페이지의 모든 테이블 요소를 출력하고, 각각의 테이블 요소의 기능을 설정하는 창이다. 기존의 A-Prompt에서도 제공하는 기능이지만 소스코드와, 테이블의 모양을 동시에 출력해주어 보다 효율적이고 정확하게 테이블 요소의 용도를 선택을 할 수 있게 해준다.



<그림 3> 테이블 요소 용도 선택창

테이블의 요소의 용도 선택이 끝난 뒤 평가 대상 페이지에서 각각의 체크 항목별로 웹 접근성을 어긴 요소에 대해서 <그림 4>와 같이 평가 결과를 출력 해준다.



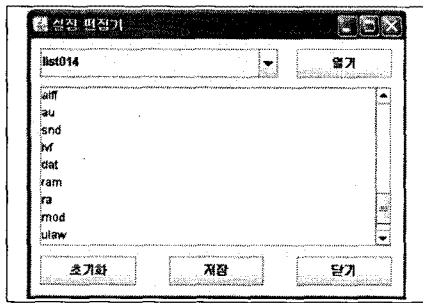
<그림 4> 평가 결과

평가 결과 보고서는 <그림 5>와 같이 각각의 체크 항목에 대해 전체, 정상, 오류의 값을 출력 해준다.

항목	전체	정상	오류
19	0	0	0
20	0	0	0
21	31	26	5
22	31	15	16
23	1	0	1
24	46	0	46
25	41	41	0
26	14	14	0
27	0	0	0
28	0	0	0
29	39	0	39
30	15	15	0
31	0	0	0

<그림 5> 평가 결과 출력

설정 편집기는 <그림 6>과 같이 체크 항목 별로 설정된 값의 편집을 지원 한다. 콤보박스로 체크 항목을 선택하고 편집한 뒤 저장을 하면 된다. 초기화가 필요한 경우 초기화 버튼을 누르면 된다.



<그림 6> 설정 편집기

KADO-WAH에서 잘못된 결과를 출력하는 항목과 사용자가 수동으로 평가해야 하는 항목을 WAET에서 최대한 구현하고, 성능 평가를 위하여 샘플 페이지를 작성하였다. 평가한 결과로 <표 1>과 같은 결과를 출력하였다.

<표 1> 성능비교

보고기준	KADO-WAH		WAET	
	항목 번호	결과	항목 번호	결과
187(개)	3	1	12	187
8(개)	8	1	13	8
동작유무	14	X	7	O
동작유무	33	X	59	O
보고범위	37	모든테이블	53	데이터테이블
보고범위	38	모든테이블	54	데이터테이블
보고범위	39	모든테이블	64	레이아웃테이블
보고범위	40	모든테이블	67	레이아웃테이블
자동여부	50	수동	50	자동
자동여부	51	수동	51	자동
13(개)	61	11	89	13
보고범위	89	모든테이블	57	데이터테이블
동작유무	90	X	46	O
보고범위	92	모든테이블	66	레이아웃테이블

KADO-WAH의 3번 항목은 WAET의 12번 항목과 동일하며, 187개의 확장자에 대해 오류를 출력해주어야 하지만 KADO-WAH에서는 단 한 개의 오류만을 나타내고 WAET에서는 187개의 오류를 나타낸다. KADO-WAH의 37, 38, 89번 체크 항목은 데이터 테이블, 39, 40, 92번 체크항목은 레이아웃 테이블일 경우에 출력되어야 하지만 테이블의 용도를 알 수 없기 때문에 모든 테이블에 대해 경고 한다고 명시되어 있다[5].

그러나 WAET은 테이블의 용도를 미리 선택했기 때문에 테이블 용도별로 보다 자세한 결과를 출력해준다. 이와 같이 나머지 항목들에서도 WAET의 기능이 KADO -WAH에 비해 개선된 것을 볼 수 있다. 또한, WAET은 대체 텍스트 길이 제한과 11, 12, 13, 14, 25, 26, 45번 체크항목은 <그림6>과 같이 설정 편집기를 이용하여 평가 기준 값을 수정할 수 있어 보다 사용자에게 한정된 평가 도구가 아닌 폭넓고 다양한 평가를 가능하게 해준다.

V. 결론

본 논문에서는 국내·외 웹 접근성 평가도구의 문제점들을 비교, 분석하고, 이를 이용하여 국내 자동화 평가도구인 KADO-WAH보다 자동화 기능을 개선한 평가도구 WAET을 구현하였다.

WAET은 현재 타 평가도구에 비해 자동화 기능이 개선되긴 하였지만 앞으로 자동화 평가 기능뿐만 아니라 편리한 수정기능과 더불어 웹 개발과정에서 검사하고 수정할 수 있는 도구 등 다양한 기능을 추가로 구현 할 계획이다. 또한, 이번 연구를 통해 구현된 웹 접근성 평가도구는 향후 개발자간의 웹 접근성 평가 결과를 모아 결과를 산출하고 값의 조절이 가능한 협업 시스템과 같은 연구로 나아갈 수 있다.

참고 문헌

- [1] W3C Recommendation, "Web Contents Accessibility Guideline 1.0", 1999.5
- [2] 한국통신기술협회, "한국형 웹 콘텐츠 접근성 지침 1.0", 2004.12
- [3] 홍순규, 박지용, 이대형 "한국과 미국 정부기관의 웹 접근성 평가를 통한 자동화 평가 도구의 문제점 및 개선 방안에 관한 연구", 한국산업정보학회논문지, Journal of Korean Industrial Information Systems Society, v.11 no.1, 2006, pp.38-48
- [4] <http://www.iabf.or.kr>
- [5] 한국정보문화진흥원, "KADO-WAH 사용자 가이드", 2005.3
- [6] <http://aprompt.snow.utoronto.ca/>
- [7] <http://wave.webaim.org>