A Common Vocabulary to Facilitate the Exchange of Web Accessibility Evaluation Results

Shadi Abou-Zahra
W3C Web Accessibility Initiative
Tel: +33 492 38 5064
Fax:+33 492 38 7822
shadi@w3.org

Abstract:

Web accessibility evaluations are significantly different from document validations; they are not based on determining conformance with specifications of formal grammars but are often more rule-based methodologies encompassing sequences of atomic tests to determine conformance to guidelines. While some of these tests may not be automatable by current computer technology and require human judgment to determine the results, evaluation tools often vary considerably in their coverage and reliability for tests that are automatable. These differences in tool performances are probably unavoidable in practice; however, they cause notable discrepancies in the efficiency and quality of the conformance evaluations. This paper discusses the practical implications of the *Evaluation and Report Language*; a vendor neutral vocabulary to facilitate the aggregation of test results generated by different evaluation tools in order to maximize the benefit from their respective features. The paper will also highlight the importance of this language to the objectives of the Dublin Core Metadata Initiative.

Keywords:

Web Accessibility, Automated Evaluation, Conformance Testing, Quality Assurance, Validation, Content Exchange.

1. Introduction

The Evaluation and Repair Tools Working Group (ERT WG) is part of the Web Accessibility Initaitive (WAI) at the World Wide Web Consortium (W3C) and maintains a comprehensive list of Web accessibility evaluation tools. Studying the entries of this list shows broad diversity between the features provided by the currently available tools. These differences can be roughly summarized into the following characteristics:

- **Type of User Interface**
  User interfaces highlight the overall intended usages of the evaluation tools. For example, tools that generate listings of evaluation results are typically focused on a high degree of automation while others that modify the appearance of the content (e.g. reading out loud, displaying without colors, removing images, etc.) are typically focused on assisting evaluators in assessing non-automatable tests. The following types of user interfaces can be found, some tools provide more than one mode of operation:
  - **Report Generating Tools**
    generate listings and reports of evaluation results
  - **Wizard Interface Tools**
    guide users through evaluation processes step-by-step
  - **In-Page Feedback Tools**
    assist evaluations by inserting icons and markup into the content
  - **Transformation Tools**
    modify the appearence of the content to assist manual evaluation

- **Coverage of Accessibility Guidelines**
  While some evaluation tools only focus on specific accessibility tests to develop algorithms for, others attempt to cover whole sets of guidelines such as the Web Content Accessibility Guidelines (WCAG). The degree of coverage, automation,

and reliability of the testing varies significantly between different tools; sometimes also between versions of the same tool.

- **Support for Web Technologies**
  Not all evaluation tools support common Web technologies (for example XHTML or CSS, etc.) equally well. While some evaluation tools are specialized for specific technologies (sometimes these are proprietary formats), others aim to be more all-round.

- **Other Features and Options**
  Besides functional characteristics like the ones listed above, there is a multitude of features that evaluation tools sometimes provide with varying degrees. These features include repair capabilities, support for internationalization, compatibility with other tools (e.g. editors or content management systems), support for different operating systems and configurations, customization features, as well as others.

At first, this jungle of Web accessibility evaluation tools seems chaotic but in fact it is a natural reaction to the equally diverse user requirements. There is a wide spectrum of tool users and other case specific factors that determine which evaluation tools may best assist in accomplishing different tasks. The following are some of the considerations that may be made while selecting evaluation tools:

- **What is being evaluated?**
  Is the content being evaluated a specific page, a whole Web site, or a path of pages within a site? What technology is being used to implement the content? Is the content static or generated dynamically? Is the overall layout and design, the programming and markup, or the actual content being evaluated?

- **Who is evaluating it?**
  Is it a single person or a team of reviewers? Can it be assumed that reviewers have sufficient background skills and experience in Web accessibility? Can skills and experience in the underlying implementation and Web technologies be assumed? Is the evaluation being conducted independently of the development process?

- **Why is it being evaluated?**
  For debugging by the Web developers during the design, implementation, or maintenance phases of the content? For third-party evaluation services? To comply with legal policies or legislations? To monitor the accessibility status of the content?

From the observations made above, it can be concluded that the diversity between the features provided by Web accessibility evaluation tools is not a problem per-se but rather a welcome response to the diverse requirements. However, the diversity in reliability and performance amongst the evaluation tools is an unwanted side effect that can cause inaccurate conformance evaluations and therefore the publishing of potentially inaccessible Web content. In some countries, this could also have legal implications.

In order to compensate the disadvantages of some evaluation tools but still make use of their stable features, the Evaluation and Report Language 1.0 (EARL 1.0) proposes a vocabulary to express test results. The language is designed to be simple and abstract enough for generic quality assurance testing. Also, the design of the language acknowledges that Web accessibility evaluations will not be fully automatable in the foreseeable future and therefore encourages the usage of combinations of automated and manual evaluation tools for different purposes (for example, at different stages of the Web development process). The following describes the features of EARL 1.0 as well as possible models for aggregating the evaluation results from different sources.

2. Core Classes

The current Working Draft of the RDF Schema for the Evaluation and Report Language 1.0 (EARL 1.0) lists all currently proposed classes and properties. Basically, the EARL 1.0 proposes a simple model made of a collection of assertions that have the following structure:

Evaluator *claims* ***Result*** *after* ***Test*** *on* ***Subject***

In other words, each ***assertion*** contains information about the ***subject*** that is being evaluated; the ***test case*** against which it is being evaluated; the ***result*** of the test; as well as the ***assertor*** that is claiming this assertion. The following is a description of some of the core classes as well some of their properties in order to highlight the overall scope of the EARL 1.0 Schema:

- **Assertor**
  Generic class to describe the evaluator that claims an assertions. The assertor can be sub-classed in order to describe the tool or human evaluators more specifically.

- **TestSubject**
  Abstract class to describe the subject being tested. EARL 1.0 proposes a sub-class for Web content but tools can introduce their own as well.

- **TestCase**
  Basic class that only contains a URI property to represent a test case. EARL 1.0 does not attempt to introduce more properties about the descriptions or nature of these test cases in order to remain independent of any specific domain processes or vocabulary.

- **TestResult**
  Basic class that contains the actual claim of the assertion. Currently EARL 1.0 proposes the following three properties for the TestResult class:
- **result**
  one of the following values: Pass, Fail, NotApplicable, or NotTested;
- **message**
  additional information such as error or success messages for the users;
- **confidence**
  a High, Medium, or Low value indicating the level of confidence.

- **TestMode**
  Basic class describing the mode in which an assertion was made. The possible values in EARL 1.0 are Automatic, Manual, or Heuristic.

The language has been intentionally designed to be simple and transparent in order to remain generic enough for the usage for other quality assurance purposes, as well as to be extensible enough for the usage in the context of Web accessibility. However, some implementations of EARL 1.0 seem to indicate that the language requires more optimization. For example, the confidence claims of the test results are ambigous and need to be revised before EARL 1.0 can mature to a W3C Recommendation. A more in-depth discussions about open search questions is discussed in the section 4 “Future Work” further below.

3. Use Cases

In the context of evaluating Web sites for accessibility, the following use cases illustrate some of the ways in which EARL can be utilized. Some of the use cases describe how the language can contribute to more efficient and effective evaluations while other use cases outline was of providing accessibility features by actively using the EARL as metadata to describe the accessibility of the content.

3.1. Combine Reports

Web accessibility evaluation tools vary greatly in their performance. For example, while some evaluation tools have more advanced color contrast analysis algorithms, others perform better in text analysis. EARL provides a standardized data format which allows test results from automated or semi-automated evaluation tools to be collected into a single repository. This allows reviewers to collaborate and to integrate several evaluation tools into the review process.

3.2. Compare Test Results

EARL allows the test results from Web accessibility evaluation tools to be compares to known test files. For example, the HTML test suites for the Web Content Accessibility Guidelines 2.0 (WCAG 2.0). This helps tool developers to ensure that their evaluation tools implement a correct interpretation of the guidelines. Tools could also be compared against each other statistically using EARL output. Such benchmarking indicators could help the users of the tools select specific evaluation results from different tools and compose more precise accessibility reports.

3.3. Processing Results

A standardized format such as EARL encourages the development of data processing tools that analyze, sort, prioritize, or infer test results according to different policies. For instance, it may sometimes be desirable to sort the test results according to their corresponding severity (for example by matching them to the respective impact on accessibility). In other cases, the relative cost of repair for accessibility barriers may be the criteria by which processing tools may be configured to sort the test results by. Data processing tools can also output their reports in EARL format to allow cascades of EARL enabled tools with different specializations.

3.4. Customized Reports

Test results can contain comprehensive information for different end-users. For example line numbers and detailed error messages for Web developers, or less verbose technical detail for project managers and executives. Repair suggestions and educational resources may sometimes be helpful to educate developers new to Web accessibility, but may also be tedious for more experienced ones. The well defined structure of EARL allows customized data views to be made from the same set of test results in order to suite the preferences of the end-users.

3.5. Integration into Authoring Tools

EARL provides a standardized, royalty-free, and vendor neutral interface between Web accessibility evaluation tools and authoring tools. Instead of generating reports with test results, authoring tools could directly process these machine readable results and assist Web developers in finding and fixing errors through appropriate prompts and dialogs. This features also benefits evaluation tool vendors who want to focus on developing specialized accessibility testing algorithms rather than on implementing full-blown tools with user interfaces; these modules could easily export their results to EARL enabled authoring tools.

3.6. Integration into Web Browsers

Web accessibility evaluation tools could add significant features to Web browsers by assessing the accessibility of the content and providing detailed results. Web browsers could then compare the user preferences to the encountered accessibility results and readapt the content accordingly. For example, a Web browser may be configured to re-render complex tables or to suppress moving content. On each of these occasions, evaluation tools could provide information about the existence and location of such content on Web sites.

3.7. Integration into Search Engines

Similar to Web browsers, search engines could also make use of EARL reports to respond to user queries according to their preferences. However, search engines may prefer to out-source such features to third party evaluation services that publish accessibility reports for Web sites according to specific guidelines (for example national policy requirements). This is another example of where EARL benefits from the powerfull querying mechanisms provided by other Semantic Web technologies such as OWL, RDQL or SPARQL.

3.8. Justifying Accessibility Claims

Currently, there are many accessibility icons and labels that can be used to indicate the accessibility level of Web sites. However, often the accessibility level indicated by such marks is over claimed or outdated. While EARL 1.0 can not directly address the issue of reports becoming outdated over time, it can assist in supplementing the respective marks with more comprehensive reports of what has been tested in order to provide more credibility for the usage. The reports are metadata stored outside the direct Web content and are therefore transparent to the users. However, EARL aware browsers or other user agents could process these reports according to preferences.

4. Future Work

While EARL is slowly maturing to become a stable and widely deployed standard, there are still several challenging research questions which the Evaluation and Repair Tools Working Group (ERT WG) is currently working on. The following are some of the aspects of EARL which the Working Group hopes to improve in the next version:

4.1. Location of Results

The current EARL schema proposes a model in which the subject of the assertion sufficiently describes the location of the result. For example, if the subject being tested is a specific markup element in a Web page, it could be described using an XPath expression in the subject of the assertion so that it could later be found again. However, in several situations it is desired to have a more generic subject as the overall context of the assertion, and more specific location pointers (such as XPath or XPointer expressions, line and position numbers, as well as other methods) separately. For example, the subject of the assertion could be Web content, and additional location pointers could identify each location within that subject where the test case of the assertion yields the same results. The main issue to resolve in this effort is locating results in Web content that is not markup- or even text based. For example Macromedia Flash or Adobe PDF.

4.2. Persistency of Results

In EARL 1.0, assertions are tied to date-time stamps which makes the results only valid for a snapshot in time. For example, on the Web the life spans of the assertions are tied to the creation date of the content and are therefore in average quite brief. At the same time, some of these results may be *expensive* (for example when manual reviews are required to make a judgement). While it is probably not possible to achieve absolute persistency of the assertions with respect to changes in the subject, there is some room for potential enhancement of this aspect. There are several possibilities to enhance the persistency aspect of EARL assertions, for example by studying which types of changes to a subject imply which consequences to the assertions in respect to test cases. For example, for tests that are not context-sensitive (typically validation-type tests), changes outside the direct scope of the test usually do not affect the result. However, most accessibility tests are somewhat context-sensitive.

4.3. Relationship between Assertions

As briefly mentioned earlier, EARL 1.0 proposes a model in which assertions are independent of each other and can only be sometimes related by processing the subject. However, directly describing the relationships between both the subjects and the test cases allows EARL assertions to address subjects or test cases which are composed of several related parts. Examples include:

paths of pages within Web sites that are required in order to commit a transaction

sets of source code files that are compiled in order to build a software application

expressing statements that are based on the results gathered from other sub-tests

However, a carefull balance has to be maintained to avoid over-featuring EARL and possibly giving away some of the flexibility of the language towards generic testing processes.

4.4.Confidence Claims

While the confidence property that is proposed by EARL 1.0 potentially provides a mechanism to prioritize assertions, it has not shown the desired effect in practice. The main reason for that is that currently EARL does not provide sufficient guidance on how to make use of that element. This lead to different interpretations between implementations and therefore a lack of compatibility and reliability on the value of this element. It is vital to refine the model for expressing confidence claims in EARL assertions and adjusting the processing model accordingly. It may be necessary to extend or tweak other related EARL elements, such as the test case for example, in order to support a more robust model for conformance claims.

5. Relevance to Dublin Core

Describing the accessibility of Web content in a reusable vocabulary is significantly in-line with work pursued by the Accessibility Working Group of the Dublin Core Metadata Initiative. Systems such as the IMS e-learning platform can develop higher level protocols to deliver content that matches the user preferences. However, as discussed in section 3 “Use Cases”, this is not the only use case provided by EARL. Since EARL is not restricted to Web accessibility evaluations, it can be used to describe more generic information about the profiles of objects that are not necessarily available on the Web. For example, to describe the conformance of products with requirements and specifications. In this sense, EARL can be used to justify the usage of quality marks.

6. Conclusion

EARL provides a mean for different types of tools to exchange test results amongst each other. These tools include (but are not limited to) Web accessibility evaluation tools, authoring tools, user agents, search engines, assistive technologies, and data processing tools. EARL facilitates the interoperability of these tools and their integration into existing development or browsing environments. For Web developers, the integration of Web accessibility evaluation tools into existing development environments such as editors or content management systems may reduce the time and effort required to carry out comprehensive evaluation reviews. For Web users, the integration of evaluation tools into search engines or browsers can significantly enhance their experience on the Web.

While existing implementations of EARL highlight its benefits, EARL is still at a relatively early stage without a wide main stream support in Web accessibility evaluation tools. There are also some key challenges and research questions open which need to be addressed and resolved before EARL can become a stable W3C standard. The Evaluation and Repair Tools Working Group (ERT WG) is actively developing EARL in coordination with several related activities within and outside W3C in order to deliver a mature standard which can be of relevance outside the realm of Web accessibility.

References

1. Web Accessibility Initiative
   http://www.w3.org/WAI/
2. Web Content Accessibility Guidelines
   http://www.w3.org/TR/WCAG10/
3. Authoring Tool Accessibility Guidelines
   http://www.w3.org/TR/ATAG10/
4. User Agent Accessibility Guidelines
   http://www.w3.org/TR/UAAG10/
5. Evaluation and Repair Tools
   http://www.w3.org/WAI/ER/
6. Quality Assurance Activity
   http://www.w3.org/QA/
7. Semantic Web Activity (SW)
   http://www.w3.org/2001/sw/
8. Resource Description Framework (RDF)
   http://www.w3.org/RDF/
9. Evaluation and Report Language
   http://www.w3.org/TR/EARL10/
10. Web Ontology Language (OWL)
    http://www.w3.org/2004/OWL/
11. Evaluation, Repair, and Transformation Tools for Web Content Accessibility
    http://www.w3.org/WAI/ER/existingtools
12. DCMI Accessibility Working Group
    http://www.dublincore.org/groups/access/
13. IMS Global Learning Consortium
    http://www.imsproject.org/accessibility/
14. Annotea Project
    http://www.w3.org/2001/Annotea/