[contents]

Evaluation and Report Language (EARL) 1.0 Schema

W3C Working Draft 10 May 2011

This version:
http://www.w3.org/TR/2011/WD-EARL10-Schema-20110510/
Latest version:
http://www.w3.org/TR/EARL10-Schema/
Previous version:
http://www.w3.org/TR/2009/WD-EARL10-Schema-20091029/
Editors:
Shadi Abou-Zahra, W3C/WAI
Previous Editors:
Michael Squillace (until October 2009 while at IBM Corporation)

The terms defined by this document are also provided in RDF Schema format.

Abstract

This document describes the formal schema of the Evaluation and Report Language (EARL) 1.0. EARL is a vocabulary, the terms of which are defined across a set of specifications and technical notes, and that is used to describe test results. The primary motivation for developing this vocabulary is to facilitate the exchange of test results between Web accessibility evaluation tools in a vendor-neutral and platform-independent format. It also provides reusable terms for generic quality assurance and validation purposes.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This 10 May 2011 Last Call Working Draft of the Evaluation and Report Language (EARL) 1.0 Schema is an update of the previous EARL 1.0 Last Call Working Draft of 29 October 2009. It meets the requirements specified in the Requirements for the Evaluation and Report Language (EARL) 1.0, and incorporates all comments received. This document is intended to be published and maintained as a W3C Recommendation after review and refinement.

The Evaluation and Repair Tools Working Group (ERT WG) believes it has addressed all issues brought forth through previous Working Draft iterations. The Working Group encourages feedback about this document, Evaluation and Report Language (EARL) 1.0 Schema, by developers and researchers who have interest in software-supported evaluation and validation of websites, and by developers and researchers who have interest in Semantic Web technologies for content description, annotation, and adaptation. In particular, the Working Group is looking for feedback on the following items which are also highlighted within the document:

- Use cases for foaf:Document as a further refinement for earl:TestSubject (see Editor Note 1)
- Use of DOAP terms and definition of earl:Software as a subclass of doap:Project (see Editor Note 2)

Please send comments on this Evaluation and Report Language (EARL) 1.0 Schema document by 10 June 2011 to public-earl10-comments@w3.org (publicly visible mailing list archive).

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document has been produced by the Evaluation and Repair Tools Working Group (ERT WG) as part of the Web Accessibility Initiative (WAI) Technical Activity.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

Appendices

---

1. Introduction

The Evaluation and Report Language (EARL) defines a vocabulary for expressing test results. It enables any person, software application, or organization to assert test results for any test subject tested against any set of criteria. The test subject might be a website, an authoring tool, a user agent, or some other entity. The set of criteria may be accessibility guidelines, formal grammars, or other types of quality assurance requirements. Thus, EARL is flexible with regard to the contexts in which it can be applied.

This document provides the core schema of EARL. Other parts of the EARL suite of specifications include:

- HTTP Vocabulary in RDF 1.0
- Representing Content in RDF 1.0
- Pointer Methods in RDF 1.0

The Developer Guide for Evaluation and Report Language (EARL) 1.0 explains how to implement and

use EARL, including conformance requirements for software tools. An Evaluation and Report Language (EARL) Overview is also available.

EARL is not a comprehensive vocabulary for describing test procedures, test criteria, or test requirements but, rather, for describing the outcomes from such testing. EARL can be supplemented by test description vocabularies or other vocabularies for different aspects of the testing cycle.

1.1. Audience of this Document

The assumed audience of this specification is developers of software tools and processes who want to express test results in a machine readable format that is semantically rich. More introductory background about EARL as well as specific guidance for quality assurance tool developers, in particular for web accessibility evaluation tool developers, is provided in the Evaluation and Report Language (EARL) 1.0 Guide.

This document assumes that the reader is familiar with the Resource Description Framework (RDF) and can read its XML serialization. Readers who wish to understand more about RDF should read a general introduction or the RDF Primer [RDF-PRIMER].

1.2. Document conventions

The RDF representation of the vocabulary defined by this document uses the namespace http://www.w3.org/ns/earl#. The prefix `earl` is used throughout this document to denote this namespace. Other prefixes used throughout this document include:

- `cnt` - Representing Content in RDF namespace http://www.w3.org/2011/content# (defined by [Content])
- `dct` - Dublin Core (DC) namespace http://purl.org/dc/terms/ (defined by [DC])
- `doap` - Description of a Project (DOAP) namespace http://usefulinc.com/ns/doap# (defined by [DOAP])
- `foaf` - Friend of a Friend (FOAF) namespace http://xmlns.com/foaf/0.1/# (defined by [FOAF])
- `http` - HTTP Vocabulary in RDF namespace http://www.w3.org/2011/http# (defined by [HTTP])
- `ptr` - Pointer Methods in RDF namespace http://www.w3.org/2009/pointers# (defined by [Pointers])
- `rdf` - RDF namespace http://www.w3.org/1999/02/22-rdf-syntax-ns# (defined by [RDF])
- `rdfs` - RDF Schema namespace http://www.w3.org/2000/01/rdf-schema# (defined by [RDFS])
- `xsd` - XMLS namespace http://www.w3.org/2001/XMLSchema# (defined by [XMLS])

2. Classes

This section describes the classes defined by this document. Every test result in EARL is expressed as an assertion. An EARL Assertion contains the following information:

Assertor
: This can include information about who or what ran the test. For example human evaluators, automated accessibility checkers, or combinations of these.

Test Subject
: This can include web content (such as web pages, videos, applets, etc.), software (such as authoring tools, user agents, etc.), or other things being tested.

Test Criterion
: What are we evaluating the test subject against? This could be a specification, a set of guidelines, a test from a test suite, or some other testable statement.

Test Result
: What was the outcome of the test? A test result could also include contextual information such as error messages or relevant locations within the test subject.

EARL provides flexibility to describe different types of assertions, such as those carried out by automated testing tools or by human evaluators, or those made about generic testing requirements or specific test cases.

Examples

> Example 1: A person carries out a manual evaluation of a web page to an accessibility requirement.

Assertor
    Bob B. Bobbington
Test Subject
    A web page located at http://www.example.org/page.html
Test Criterion
    Success Criterion 1.1.1 of the Web Content Accessibility Guidelines (WCAG) 2.0
Test Result
    Passed

Example 2: A software application carries out automated validation of a web page to a technical specification.

Assertor
    The W3C Markup Validator located at http://validator.w3.org/
Test Subject
    The XHTML returned from a GET request to the URI http://www.example.org/page.html at 2004-04-14T14:00:04+1000
Test Criterion
    The validity of the XHTML code
Test Result
    Failed, the <li> element on line 53, char 7 was not closed.

2.1. Assertion Class

Assertion - a statement that embodies the results of a test.

Related Properties

- Domain of:
  - earl:assertedBy
  - earl:subject
  - earl:test
  - earl:result
  - earl:mode
- Range of: none

Examples

Example 3: Instance of an assertion expressed as an RDF/XML fragment.

```
<earl:Assertion rdf:about="#assertion">
  <earl:assertedBy rdf:resource="#assertor"/>
  <earl:subject rdf:resource="http://www.example.org/"/>
  <earl:test rdf:resource="http://www.w3.org/TR/WCAG20-TECHS/H36"/>
  <earl:result rdf:resource="#result"/>
</earl:Assertion>
```

2.2. Assertor Class

Assertor - an entity such as a person, a software tool, an organization, or any other grouping that carries out a test collectively.

Related Classes

Rather than specifying only an earl:Assertor type, it is recommended that one of the following types be employed in addition:

earl:Software
    Software - the assertor is a piece of software.
foaf:Agent
    Agent - the assertor is an agent, as defined by [FOAF].

foaf:Person
: Person -the assertor is a person, as defined by [FOAF].

foaf:Organization
: Organization - the assertor is an organization, as defined by [FOAF].

foaf:Group
: Group - the assertor is a group of agents, as defined by [FOAF].

Related Properties

- Domain of:
  - earl:mainAssertor
- Range of:
  - earl:assertedBy
  - earl:mainAssertor

It is recommended to provide additional information about the Assertor by using the following properties from external vocabularies:

dct:title
: Human readable title for the assertor.

dct:description
: Human readable description of the assertor.

foaf:name
: Name of the assertor. This could be supplemented with further refinements such as foaf:firstName or foaf:surname if the assertor is a person.

foaf:nick
: Nick name of the assertor

foaf:mbox
: E-mail address of the responsible assertor, which is preferably provided in an encrypted format using the foaf:mbox_sha1sum property.

foaf:homepage
: Homepage of the assertor.

foaf:member
: Member of the assertor, such as an individual in a group of testers or a tool used by an agent.

Examples

Example 4: An Assertor that is a person called Bob B. Bobbington.

```
<foaf:Person rdf:about="http://www.example.org/people/#bob">
  <foaf:name>Bob B. Bobbington</foaf:name>
  <foaf:mbox rdf:resource="mailto:bob@example.org"/>
  <foaf:mbox_sha1sum>1a9daad476f0158b81bc66b7b27b438b4b4c19c0</foaf:mbox_sha1sum>
</foaf:Person>
```

Example 5: An Assertor that is a piece of software called Cool Tool.

```
<earl:Software rdf:about="http://www.example.org/tools/#cooltool">
  <doap:name xml:lang="en">Cool Tool</doap:name>
  <doap:description xml:lang="en">My favorite tool!</doap:description>
  <doap:created>2011-04-27</doap:created>
  <doap:homepage rdf:resource="http://example.org/tools/cool/"/>
  <doap:release>
    <doap:revision>1.0.3</doap:revision>
  </doap:release>
</earl:Software>
```

Example 6: An Assertor that is the person from example 4 using the software tool from example 5.

```
<foaf:Group rdf:about="#assertor">
  <dct:title xml:lang="en">Bob using Cool Tool</dct:title>
  <dct:description xml:lang="en">Bob doing semi-automated testing</dct:description>
  <earl:mainAssertor rdf:resource="http://www.example.org/people/#bob"/>
  <foaf:member rdf:resource="http://www.example.org/tool/#cooltool"/>
```

```
</foaf:Group>
```

Note: According to this example, "Cool Tool" is a resource of type foaf:Agent. According to example 5, it is also a resource of type earl:Software. These are not contradictory statements and are valid RDF representations.

2.3. TestSubject Class

Test Subject - the class of things that have been tested against some test criterion.

Related Classes

Rather than specifying only an earl:TestSubject type, it is recommended that one of the following types be employed in addition:

earl:Software
: Software - the test subject is a piece of software being tested.

cnt:Content
: Content - the test subject is a representation of the content as defined by [Content].

http:Response
: HTTP Response - the test subject is the response from an HTTP server as defined by [HTTP].

foaf:Document
: Document - the test subject is a document, such as electronic file, as defined by [FOAF]. [Editor's note 1: ERT WG is considering to remove foaf:Document unless compelling use-cases can be presented; feedback on this consideration is welcome.]

Related Properties

- Domain of: none
- Range of:
  - earl:subject

It is recommended to provide additional information about the Test Subject by using the following properties from external vocabularies:

dct:title
: Human readable title for the subject.

dct:description
: Human readable descriptions of the subject.

dct:date
: Date on which the subject was created or identified.

dct:hasPart
: Reference to another subject that are part of this subject.

dct:isPartOf
: Reference to another subject of which this subject is a part of.

Examples

Example 7: A group of resources that have been tested together as a single test subject.

```
<earl:TestSubject rdf:about="http://www.example.org/">
  <dct:title xml:lang="en">example.org Web site</dct:title>
  <dct:description xml:lang="en">Each page on the example.org Web site</dct:description>
  <dct:hasPart rdf:resource="http://www.example.org/style.css"/>
  <dct:hasPart rdf:resource="http://www.example.org/page1.html"/>
  <dct:hasPart rdf:resource="http://www.example.org/page2.html"/>
  <dct:hasPart rdf:resource="http://www.example.org/image1.png"/>
  <dct:hasPart rdf:resource="http://www.example.org/image2.png"/>
</earl:TestSubject>
```

2.4. TestCriterion Class

Test Criterion - a testable statement, usually one that can be passed or failed. It is a super class for all types of tests including things such as validation requirements, code test cases, checkpoints from guidelines such as Web Content Accessibility Guidelines [WCAG], or others.

Related Classes

Rather than specifying only an earl:TestCriterion type, it is recommended that one of the following types be employed in addition:

earl:TestRequirement
: Test Requirement - a higher-level requirement that is tested by executing one or more sub-tests. For example WCAG 2.0 Success Criterion 1.1.1, which is evaluated using several Techniques for Success Criterion 1.1.1 and combining the results.

earl:TestCase
: Test Case - an atomic test, usually one that is a partial test for a requirement. For example, Technique H36: Using alt attributes on images used as submit buttons provides a partial test for WCAG 2.0 Success Criterion 1.1.1.

Related Properties

- Domain of: none
- Range of:
  - earl:test

It is recommended to provide additional information about the Test Subject by using the following properties from external vocabularies:

dct:title
: Human readable title for the test criterion.

dct:description
: Human readable description of the test criterion.

dct:hasPart
: Relationship to other test criteria that are part of this criterion.

dct:isPartOf
: Relationship to other test criteria of which this criterion is a part of.

Examples

Example 8: Instance of a test case that is described with a title and its relationship to a test suite.

```
<earl:TestCase rdf:about="http://www.w3.org/TR/WCAG20-TECHS/H36">
  <dct:title xml:lang="en">H36</dct:title>
  <dct:description xml:lang="en">Technique H36 - Using alt attributes
    on images used as submit buttons </dct:description>
  <dct:isPartOf rdf:resource="http://www.w3.org/TR/WCAG20-TECHS/"/>
  <dct:hasPart rdf:resource="http://www.w3.org/TR/WCAG20-TECHS/H36#H36-tests"/>
</earl:TestCase>
```

2.5. TestResult Class

Test Result - the actual result of performing the test. It includes both machine-readable values as well as human-readable description of the results (typically error messages).

Related Properties

- Domain of:
  - earl:info
  - earl:outcome
  - earl:pointer
- Range of:
  - earl:result

It is recommended to provide additional information about the Test Result by using the following properties from external vocabularies:

dct:title
: Human readable title for the result.

dct:description
: Human readable description of the result.

dct:date

Date on which the result was obtained (typically when the subject was tested).

Examples

Example 9: A test result with a validity of fail and a description of the problem in English, and encoded in XHTML format.

```
<earl:TestResult rdf:about="#result">
  <earl:outcome rdf:resource="http://www.w3.org/ns/earl#failed"/>
  <dct:title xml:lang="en">Invalid Markup (code #353)</dct:title>
  <dct:description rdf:parseType="Literal" xml:lang="en">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>The <code>table</code> element is not allowed to appear
        inside a <code>p</code> element</p>
    </div>
  </dct:description>
  <earl:pointer rdf:resource="#pointer"/>
  <earl:info rdf:parseType="Literal" xml:lang="en">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>It seems the <code>p</code> element has not been closed</p>
    </div>
  </earl:info>
</earl:TestResult>
```

2.6. TestMode Class

Test Mode - describes how a test was carried out. It reflects the information provided by the Assertor and is used to simplify some commonly used queries.

Related Instances

Where applicable it is recommended to use one of the following instances of earl:TestMode, to categorize the mode in which the test was carried out:

earl:automatic
: Automatic - where the test was carried out automatically by the software tool and without any human intervention.

earl:manual
: Manual - where the test was carried out by human evaluators. This includes the case where the evaluators are aided by instructions or guidance provided by software tools, but where the evaluators carried out the actual test procedure.

earl:semiAuto
: Semi-Automatic - where the test was partially carried out by software tools, but where human input or judgment was still required to decide or help decide the outcome of the test.

earl:undisclosed
: Undisclosed - where the exact testing process is undisclosed.

earl:unknownMode
: Unknown - where the testing process is unknown or undetermined.

Related Properties

- Domain of: none
- Range of:
  - earl:mode

It is recommended to provide additional information about the Test Mode by using the following properties from external vocabularies:

dct:title
: Human readable title for the test mode.

dct:description
: Human readable description of the test mode.

Examples

Example 10: The assertion from example 3 was carried out in semi-automatic mode.

```
<earl:Assertion rdf:about="#assertion">
  <earl:mode rdf:resource="http://www.w3.org/ns/earl#semiAuto"/>
</earl:Assertion>
```

2.7. OutcomeValue Class

Outcome Value - a value or expression that describes a resulting condition from carrying out the test.

Related Instances

Where applicable it is recommended to use one of the following instances of earl:OutcomeValue, to categorize the outcome of carrying out the test:

earl:passed
: Passed - the subject passed the test.

earl:failed
: Failed - the subject failed the test.

earl:cantTell
: Cannot tell - it is unclear if the subject passed or failed the test.

earl:inapplicable
: Inapplicable - the test is not applicable to the subject.

earl:untested
: Untested - the test has not been carried out.

Related Classes

In cases where it is necessary to create further instances of earl:OutcomeValue, it is recommended that one of the following types be employed in addition:

earl:Pass
: Pass - the class of outcomes to denote passing a test. Subclasses may include ordinal, nominal, or continuous values or expressions.

earl:Fail
: Fail - the class of outcomes to denote failing a test. Subclasses may include ordinal, nominal, or continuous values or expressions.

earl:CannotTell
: Undetermined - the class of outcomes to denote an undetermined outcome. Usually this happens when an automated test requires human judgement to make a definite decision.

earl:NotApplicable
: Not applicable - the class of outcomes to denote the test is not applicable. This could be due to a mismatch between the test and the subject or for any other reason.

earl:NotTested
: Not tested - the class of outcomes to denote the test has not been carried out. This is useful for reporting as well as for other uses of progress monitoring.

Related Properties

- Domain of: none
- Range of:
  - earl:outcome

It is recommended to provide additional information about the Outcome Value by using the following properties from external vocabularies:

dct:title
: Human readable title for the outcome value.

dct:description
: Human readable description of the outcome value.

Examples

Example 11: A test result with an outcome of "Passed", using the corresponding

instance of earl:OutcomeValue.

```
<earl:TestResult rdf:about="#result">
  <earl:outcome rdf:resource="http://www.w3.org/ns/earl#passed"/>
</earl:TestResult>
```

Example 12: A test result with a non-standard outcome of "Warning", which is a type earl:Pass.

```
<rdf:Description rdf:about="http://example.org/my/warning#warning">
  <rdf:type rdf:resource="http://www.w3.org/ns/earl#Pass"/>
  <dc:title xml:lang="en">Warning</dc:title>
  <dc:description xml:lang="en">the subject passed the test but there are warnings</dc:description>
</rdf:Description>

<earl:TestResult rdf:about="#result">
  <earl:outcome rdf:resource="http://example.org/my/terms#warning"/>
</earl:TestResult>
```

2.8. Software Class

[Editor's note 2: ERT WG is looking for feedback on the use of DOAP Project to describe Software; feedback on this issue is welcome.]

A Software is any piece of software such as an authoring tool, browser, or evaluation tool. It can be used to describe an Assertor, such as a validation or other quality assurance tool, and it can be used to describe a Test Subject (for example to test compliance of an authoring tool to Authoring Tool Accessibility Guidelines [ATAG] or of a browser to User Agent Accessibility Guidelines [UAAG]).

Note: earl:Software is a sublass of doap:Project to denote the narrower meaning of executable "Software", that is an outcome of a "Project".

Related Properties

It is recommended to provide information about the Software by using the following properties from external vocabularies:

doap:name
: Human readable name of the software.

doap:description
: Human readable description of the software.

doap:homepage
: Homepage for the software.

doap:created
: Date when the software was created or released, in YYYY-MM-DD form. e.g. 2004-04-05.

doap:release
: Version information about the software release.

Examples

Example 13: Description of a software tool.

```
<earl:Software rdf:about="#cooltool">
  <doap:name xml:lang="en">Cool Tool</doap:name>
  <doap:description xml:lang="en">My favorite tool!</doap:description>
  <doap:created>2011-04-27</doap:created>
  <doap:homepage rdf:resource="http://example.org/tools/cool/"/>
  <doap:release>
    <doap:revision>1.0.3</doap:revision>
  </doap:release>
</earl:Software>
```

3. Properties

This section describes the properties defined by this document. EARL also uses properties from external vocabularies to provide additional information where necessary.

3.1. assertedBy Property

Asserted By - the assertor of an assertion.

Domain:
earl:Assertion
Range:
earl:Assertor

3.2. subject Property

Subject - the test subject of an assertion.

Domain:
earl:Assertion
Range:
earl:TestSubject

3.3. test Property

Test - the test criterion of an assertion.

Domain:
earl:Assertion
Range:
earl:TestCriterion

3.4. result Property

Result - the result of an assertion.

Domain:
earl:Assertion
Range:
earl:TestResult

3.5. mode Property

Mode - the mode in which the test was performed.

Domain:
earl:Assertion
Range:
earl:TestMode

3.6. mainAssertor Property

Main Assertor - the assertor that is primarily responsible for performing the test. It is a refinement of the term foaf:member defined by [FOAF].

Domain:
earl:Assertor
Range:
earl:Assertor

3.7. outcome Property

Outcome - the outcome of performing the test.

Domain:
earl:TestResult
Range:
earl:OutcomeValue

3.8. pointer Property

Pointer - the location within a test subject that are most relevant to a test result.

Domain:
earl:TestResult
Range:
ptr:Pointer

3.9. info Property

Info - additional warnings or error messages in a human-readable form.

Domain:
earl:TestResult
Range:
Literal

Appendix A: Terms

This section summarizes the terms defined and used by this EARL 1.0 Schema specification.

Classes

EARL 1.0 Classes

Class Name	Label	Comment	Refinements	Related Properties
earl:Assertion	Assertion	a statement that embodies the results of a test	-	earl:assertedBy earl:subject earl:test earl:result earl:mode
earl:Assertor	Assertor	an entity such as a person, a software tool, an organization, or any other grouping that carries out a test collectively	earl:Software foaf:Agent foaf:Person foaf:Organization foaf:Group	earl:assertedBy earl:mainAssertor dct:title dct:description foaf:name foaf:firstName foaf:surname foaf:nick foaf:mbox foaf:mbox_sha1sum foaf:homepage foaf:member
earl:TestSubject	Test Subject	the class of things that have been tested against some test criterion	earl:Software cnt:Content http:Response foaf:Document	earl:subject dct:title dct:description dct:date dct:hasPart dct:isPartOf
earl:TestCriterion	Test Criterion	a testable statement, usually one that can be passed or failed	earl:TestRequirement earl:TestCase	earl:test dct:title dct:description dct:hasPart dct:isPartOf
earl:TestRequirement (subclass of earl:TestCriterion)	Test Requirement	a higher-level requirement that is tested by executing one or more sub-	-	earl:test dct:title dct:description dct:hasPart dct:isPartOf

		tests		
earl:TestCase (subclass of earl:TestCriterion)	Test Case	an atomic test, usually one that is a partial test for a requirement	–	earl:test dct:title dct:description dct:hasPart dct:isPartOf
earl:TestResult	Test Result	the actual result of performing the test	–	earl:result earl:info earl:outcome earl:pointer dct:title dct:description dct:date
earl:TestMode	Test Mode	describes how a test was carried out	–	earl:mode dct:title dct:description
earl:OutcomeValue	Outcome Value	a discrete value that describes a resulting condition from carrying out the test	earl:Pass earl:Fail earl:CannotTell earl:NotApplicable earl:NotTested	earl:outcome dct:title dct:description
earl:Pass (subclass of earl:OutcomeValue)	Pass	the class of outcomes to denote passing a test	–	earl:outcome
earl:Fail (subclass of earl:OutcomeValue)	Fail	the class of outcomes to denote failing a test	–	earl:outcome
earl:CannotTell (subclass of earl:OutcomeValue)	Undetermined	the class of outcomes to denote an undetermined outcome	–	earl:outcome
earl:NotApplicable (subclass of earl:OutcomeValue)	Not applicable	the class of outcomes to denote the test is not applicable	–	earl:outcome
earl:NotTested (subclass of earl:OutcomeValue)	Not tested	the class of outcomes to denote the test has not been carried out	–	earl:outcome
earl:Software	Software	any piece of software such as an authoring tool, browser, or evaluation tool	–	dct:title dct:description foaf:homepage dct:hasVersion dct:hasPart dct:isPartOf

Properties

EARL 1.0 Properties

EARL 1.0 Properties

Property Name	Label	Comment	Domain	Range
earl:assertedBy	Asserted By	assertor of an assertion	earl:Assertion	earl:Assertor
earl:subject	Subject	test subject of an assertion	earl:Assertion	earl:TestSubject
earl:test	Test	test criterion of an assertion	earl:Assertion	earl:TestCriterion
earl:result	Result	result of an assertion	earl:Assertion	earl:TestResult
earl:mode	Mode	mode in which the test was performed	earl:Assertion	earl:TestMode
earl:mainAssertor (subproperty of foaf:member)	Main Assertor	assertor that is primarily responsible for performing the test	earl:Assertor	earl:Assertor
earl:outcome	Outcome	outcome of performing the test	earl:TestResult	earl:OutcomeValue
earl:pointer	Pointer	location within a test subject that are most relevant to a test result	earl:TestResult	ptr:Pointer
earl:info	Info	additional warnings or error messages in a human-readable form	earl:TestResult	Literal

Instances

EARL 1.0 Instances

Instance Name	Title	Description
earl:automatic (instance of earl:TestMode)	Automatic	where the test was carried out automatically by the software tool and without any human intervention
earl:manual (instance of earl:TestMode)	Manual	where the test was carried out by human evaluators
earl:semiAuto (instance of earl:TestMode)	Semi-Automatic	where the test was partially carried out by software tools, but where human input or judgment was still required to decide or help decide the outcome of the test
earl:undisclosed (instance of earl:TestMode)	Undisclosed	where the exact testing process is undisclosed
earl:unknownMode (instance of earl:TestMode)	Unknown	where the testing process is unknown or undetermined
earl:passed (instance of earl:Pass)	Passed	the subject passed the test
earl:failed (instance of earl:Fail)	Failed	the subject failed the test
earl:cantTell (instance of earl:CannotTell)	Cannot tell	it is unclear if the subject passed or failed the test
earl:inapplicable	Inapplicable	the test is not applicable to the subject

(instance of earl:NotApplicable)		
earl:untested (instance of earl:NotTested)	Untested	the test has not been carried out

Appendix B: References

This section provides references to related documents and specifications.

[ATAG]
Authoring Tool Accessibility Guidelines
[Content]
Representing Content in RDF
[DC]
Dublin Core (DC) Metadata
[DOAP]
Description of a Project (DOAP) Vocabulary
[FOAF]
Friend of a Friend (FOAF) Vocabulary
[Guide]
Evaluation and Report Language (EARL) 1.0 Guide
[HTTP]
HTTP Vocabulary in RDF
[Pointers]
Pointer Methods in RDF
[RDF]
Resource Description Framework (RDF): Concepts and Abstract Syntax
[RDFS]
RDF Vocabulary Description Language 1.0: RDF Schema
[RDF-PRIMER]
RDF Primer
[RDF/XML]
RDF/XML Syntax Specification
[RDF-XML-DIFFS]
Why RDF model is different from the XML model
[RFC 2119]
Key words for use in RFCs to Indicate Requirement Levels
[UAAG]
User Agent Accessibility Guidelines
[WCAG]
Web Content Accessibility Guidelines
[XMLS]
XML Schema Part 0: Primer

Appendix C: Contributors (Non-Normative)

EARL is the result of the work of many people over the past. The editors would particularly like to thank Wendy Chisholm, Sean B Palmer, and Daniel Dardailler, whose contributions have included editing the first versions of the EARL specifications, and Leonard Kasday who set the work in motion to develop EARL. The editors apologise for any names left out of this list, and will endeavour to rectify any errors noted in comments.

Contributors to this and/or previous Working Drafts

Shadi Abou-Zahra, Philip Ackermann, Chrisoula Alexandraki, Shane Anderson, Myriam Arrue, Gabriele Bartolini, Giorgio Brajnik, Dan Brickley, Dan Connolly, Karl Dubost, Nick Gibbins, Al Gilman, Emmanuelle Gutiérrez y Restrepo, Dominique Hazaël-Massieux, Nadia Heninger, Sandor Herramhof, Ian Hickson, Björn Höhrmann, Carlos Iglesias, Nick Kew, Johannes Koch, Jim Ley, William Loughborough, Rui Lopes, John Lutts, Charles McCathieNevile, Libby Miller, Tom Martin, Yehya Mohamed, Daniela Ortner, Dave Pawson, Eric Prud'hommeaux, Pierre Queinnec, Chris Ridpath, Romain Roure, Christophe Strobbe, Michael Squillace, Aaron Swartz, Olivier Thoreaux, Carlos Velasco, Konstantinos Votis, and Rob Yonaitis.

Appendix D: Document Changes

Besides several minor editorial changes, the most significant changes from the 29 October, 2009 Working Draft include:

- Added examples to section 2.7 OutcomeValue Class, to clarify the use of the terms;
- Made earl:Software a subclass of doap:Project, and reused several terms from DOAP
- Moved entire conformance section into the Developer Guide for EARL 1.0 document

A detailed listing of the comments, resolutions, and changes made is provided in the Open Issues for EARL 1.0 Schema listing.