

Using RDF to Annotate the (Semantic) Web

Phil Cross

Institute for Learning & Research
Technology, University of Bristol,
Bristol, UK
phil.cross@bristol.ac.uk

Libby Miller

Institute for Learning & Research
Technology, University of Bristol,
Bristol, UK
Libby.miller@bristol.ac.uk

Sean Palmer

SWAG
sean@mysterylights.com

Abstract

Collections of annotations need indexes for certain of their properties. When we have a set of created annotations, we want to be able to ask questions concerning these properties such as: agency: "Who said this?"; timing: "is this the most recent annotation about this object?"; the annotated object itself: "What annotations have been made about this object?"; and content: "Which objects which fail this test?". This document examines an approach to creating annotations on the web in two projects, EARL and MedCERTAIN. Both projects use RDF (Resource Description Framework) to model annotations in a flexible and extensible way, so that these types of questions can easily be asked.

Keywords

RDF, reification, trust, XML

INTRODUCTION

When we create annotations or aggregate them together, we will want to access the information they contain in various ways. The structure of the annotation should therefore accommodate the types of questions we will want to ask. Examples of these are queries about:

The person or agent who made the annotation

"Who said this?" "What else do we know about the person who made this annotation?"

The date and time when the annotation was made

"Is this the most recent annotation about this object?"

The object of the annotation

"What annotations have been made about this object?"

The meaning of the content of the annotation

"Which annotated objects fail this test?"

"What are the descriptions of the annotated objects?"

This paper describes two projects which have used RDF (Resource Description Framework) to describe annotations of web pages or parts of web pages: MedCERTAIN [1] and EARL [2].

MedCERTAIN is a project funded by the European Union, which is looking at means for establishing an international trustmark for health information on the Web. MedCERTAIN will be a decentralised system based on the cooperation of individuals and organisations that evaluate, assess, accredit or recommend health information on the Internet.

The data model discussed in this paper has been implemented within a system that enables third-party experts to evaluate the quality of health-related Web sites, using a metadata set developed for the project. Other aspects of the project are looking at how such evaluative metadata can usefully be amalgamated with other related RDF-encoded information. The system is currently undergoing evaluation through a test-bed in Finland.

EARL, the Evaluation And Report Language, is an RDF based framework being developed by the Evaluation and Repair Tools group of the Web Accessibility Initiative; a domain of the World Wide Web Consortium. The language started out life as an experiment in producing a standardized language that could be produced by accessibility evaluation tools, but soon grew to become a generic evaluation reporting framework that can be used to make evaluations by anyone, about anything, and against any set of criteria.

Both these projects had very specific requirements about the sorts of questions that could be asked of repositories of annotations. This paper examines these requirements with respect to the four types of questions suggested above, namely: Agency, timing, annotated object and content.

AGENCY: "WHO SAID THIS?"

An annotation is an annotation because, in some sense, it is separate from the object being annotated. Its separateness can most often be distinguished by the difference in author; for example the difference between a paper and a criticism, a book and a book review. Its significance can also be determined by its author: a critique of an academic paper can itself be evaluated by the quality of the individual who wrote the criticism. Knowing the author of an annotation can provide plenty of information about the quality of the annotation, either from contextual information already known about the annotator (things written before by them, information known about their experience and interests) or from specific information that is discoverable about them (their academic qualifications, their reputation, their standing in the community).

The MedCERTAIN project faced a problem over how it should store annotations relating to health sites. The project grew out of the subject gateways community, which employs subject specialists to create metadata about Web sites. This model is based on that used for many years by traditional bibliographic services within the library community,

and uses the underlying assumptions that the data will have been entered by a trusted, trained individual, and that the data can be assumed to be correct.

This approach was not appropriate for MedCERTAIN since the expectation was that both publishers of information and expert annotators would provide data about sites. Initially the publisher-provided data is not verified but is still to be made available to end-users. MedCERTAIN consequently needed a system where the metadata output is seen as a set of separate statements, made by a particular person, on a particular date, and which may or may not be accurate. It is initially up to the end-user to choose how much trust to place in the information. For the second step in the trustmarking process, the project needed medical experts who could say further things about the web site relevant to its quality, but who could also verify that the publisher-supplied data was correct (or not), and have a means of indicating this to the end-user. MedCERTAIN is therefore concerned with the provenance of metadata, and with helping end-users to decide how much trust they may place in it.

EARL's model is based on the requirement that Web accessibility annotations should be made by humans and tools in a machine readable format.

Evaluating for accessibility is a very difficult process, involving many complicated steps and procedures, often with highly ephemeral or ambiguous results. Ratings of the efficacy of things, such as alternative textual content for images, are open to conjecture and opinion. On the other hand, certain sorts of evaluations can be done automatically with tools, which for example can test for the presence of an alt tag inside an image tag.

In EARL, the context of the evaluation contains information pertaining to the actual creator or generator of the evaluation itself, e.g. giving details of the tool or person running the test, and the platform settings of the machine on which the test was run.

Where the evaluation is automatic, the precise settings of the tools used and the hardware and software run is of crucial importance for the repeatability of the test, for the same reasons that bug-testing software has to specify these characteristics. In the human evaluated case, the complexity and subjectivity of the evaluation means that who made the evaluation - and by implication the experience, qualifications and other contextual information behind that evaluation - be as explicit as possible.

Both projects have the requirement that the agency making the annotation should be explicit. This transparency means that individuals who use the MedCERTAIN service or the EARL tools can apply their criteria for scepticism to evaluate the quality of the annotation.

TIMING: "IS THIS THE MOST RECENT ANNOTATION ABOUT THIS OBJECT?"

Objects on the web change frequently, and any metadata about such objects has to take this into account, particularly

where the metadata creation process is decentralized as with the EARL and MedCERTAIN projects. The accuracy and therefore the trustworthiness of the annotation is very likely to change as time passes. As third-party annotations, there is no way to ensure that over time the evaluation is correct without repeated checking, which may not be feasible in terms of person effort.

In the MedCERTAIN annotation model, each time a new piece of information is created about a site, or old information is edited, a new annotation is created. When the system is queried, this new annotation replaces any information about the same aspect of the site created at an earlier date. The query that is made is essentially: "Find me the most recent annotation concerning this metadata for this site".

There are problems with this simple approach, however. What happens if someone wishes only to delete information that is no longer correct, without providing updated information to replace it? Also, what happens when multiple values can legitimately be provided for some aspect of the site - such as for creators/authors - which may be added at different times? In practice, the MedCERTAIN system simply maintains a database that has tables for 'current' annotations and 'superceded' annotations. Any annotations that are deleted or edited have their old version moved to the 'superceded' tables. Only data from the 'current' tables are shown to end users. For exporting and amalgamating data though, MedCERTAIN is experimenting with the idea of generating 'Retraction' annotations, that would annotate existing annotations, with a statement to the effect that that annotation is now considered to be false.

Nonetheless, by adding a time to an annotation, we implicitly refer to an annotation event that has occurred, and at that time we can say something about the relationship between the content of the annotation and the object annotated.

EARL attaches a date in an ISO standard format to the resource which it is evaluating. So instead of evaluating the resource as is it for all time, it evaluates how it was on a certain date, guaranteeing persistence. EARL also allows you to link to a stored version of the resource as it was on that date.

It is sometimes useful to include descriptions of how the context of the content changes throughout time. For example, what happens when an evaluation is made about a paragraph of text, and later that paragraph of text moves? EARL does have a property that lets the user assert some notion of "equivalence" between two pieces of content, but the exact semantics of this property are not stated.

These properties are not essential for EARL to be functional, but future work on EARL may include adding a range of properties that describe the way a piece of content changes through time to some finer degree of granularity.

With third party annotations the accuracy of the content of the annotation cannot usually be guaranteed; but dating the

annotation provides clues to the user about the likely accuracy of the annotation.

THE ANNOTATED OBJECT: "WHAT ANNOTATIONS HAVE BEEN MADE ABOUT THIS OBJECT?"

In order to describe the relationships between annotations and their objects accurately, it is necessary that we point unambiguously at the object of the annotation - that we can give it a name.

RDF (Resource Description Framework) allows you to say anything about anything with a URI. This means that it is particularly suited to web-based annotations applications over HTML at the scale of the HTML document or internal reference. If the document to be annotated is XML, parts of the document can be pointed to using XPointer [3], and so RDF becomes more flexible in what it can be used to annotate.

Both EARL and MedCERTAIN use RDF to model their annotations. At its simplest level, RDF is a series of typed directional links between objects represented by URI references. So one can say:

[<http://example.com/documentA.html>, *annotates*, <http://example.com/documentB.html>]

rather like the way in which the HTML tag `` is an untyped link between two documents [4]. This approach to annotating HTML documents with other HTML documents using typed links is similar to that used by the W3C's Annotea project [5], which makes an annotation object with a body (an (X)HTML document which annotates another (X)HTML document - see the example in Figure 1, taken from [6]).

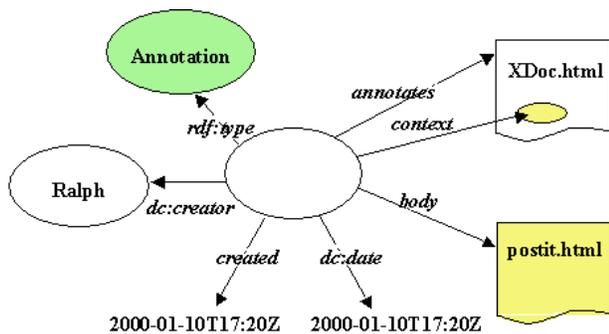


Figure 1. Annotea annotation

The difficulty with pointing into documents, or pointing at the first paragraph of the second page, say, is that Xpointer and similar methods depend on the syntactic representation of the document rather than its meaning. In practice, this means that if the document changes, the meaning of the pointers can change, which could be an accidental result of making any edits to the document. This is part of a more general problem of the changing web. Documents on the web change, and documents may change their location on the web, even though 'Cool URIs don't change' [7].

RDF is very well suited to pointing at annotated objects on the web, but because it depends on the names given to objects by their creators, it cannot guarantee that these objects will not disappear or change.

CONTENT: "WHICH ANNOTATED OBJECTS FAIL THIS TEST?", "WHAT ARE THE DESCRIPTIONS OF THE OBJECTS?"

RDF can do more than associate one page to another with a typed link. RDF can be used to say something about the *meaning* of any given annotation in machine-readable form. One early application of this was used in the Desire project for a shared bookmarks server [8]. In this project an annotation was a description of a web page. The annotation object represented a Web page, and the structured content of the annotation represented the Web page's title, description, date, and URL, for example, as in Figure 2:

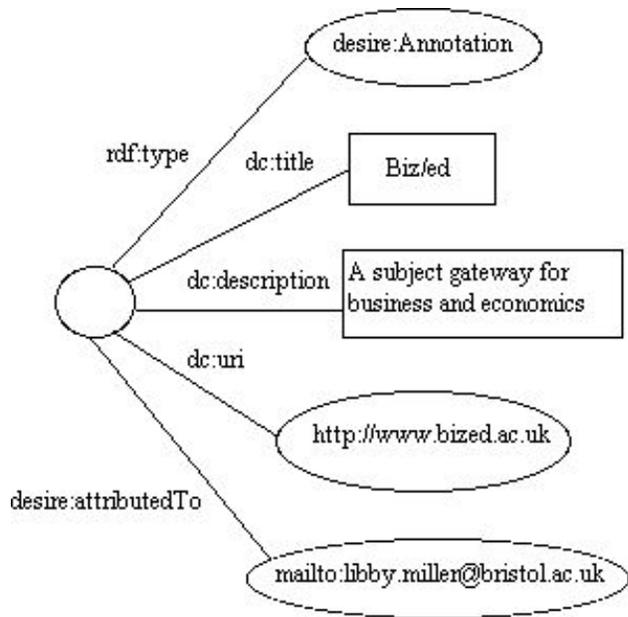


Figure 2. DESIRE annotation

The difficulty with this approach was that some of the semantics were implicit, namely, it looked as if the *annotation* had the specified title, description and so on, even though these were supposed to represent properties of the annotated Web page. Also, it was unclear that it was the person who made the annotation who gave the properties these values. This approach puts some of the interpretation of the semantics at the application level.

A more transparent approach is to use the apparatus provided by RDF to talk about objects and the links between them as objects in themselves: reification. This mechanism allows us to say:

[PersonA, stated, StatementB]

StatementB: [siteC, hasTitle, D]

Because the statement is itself an object that can be talked about in RDF, we can associate information to this very

basic atom of data, such as who made this particular statement and when.

With MedCERTAIN, an annotation object is used to link the RDF statement to the annotated object, and this is given the properties of annotator and date. This allows us to de-

fine various kinds of annotation, depending on what type of object is being annotated and the type of statement being made. For example, the publisher can annotate a page with basic metadata such as the title by using a *SiteDescription* annotation:

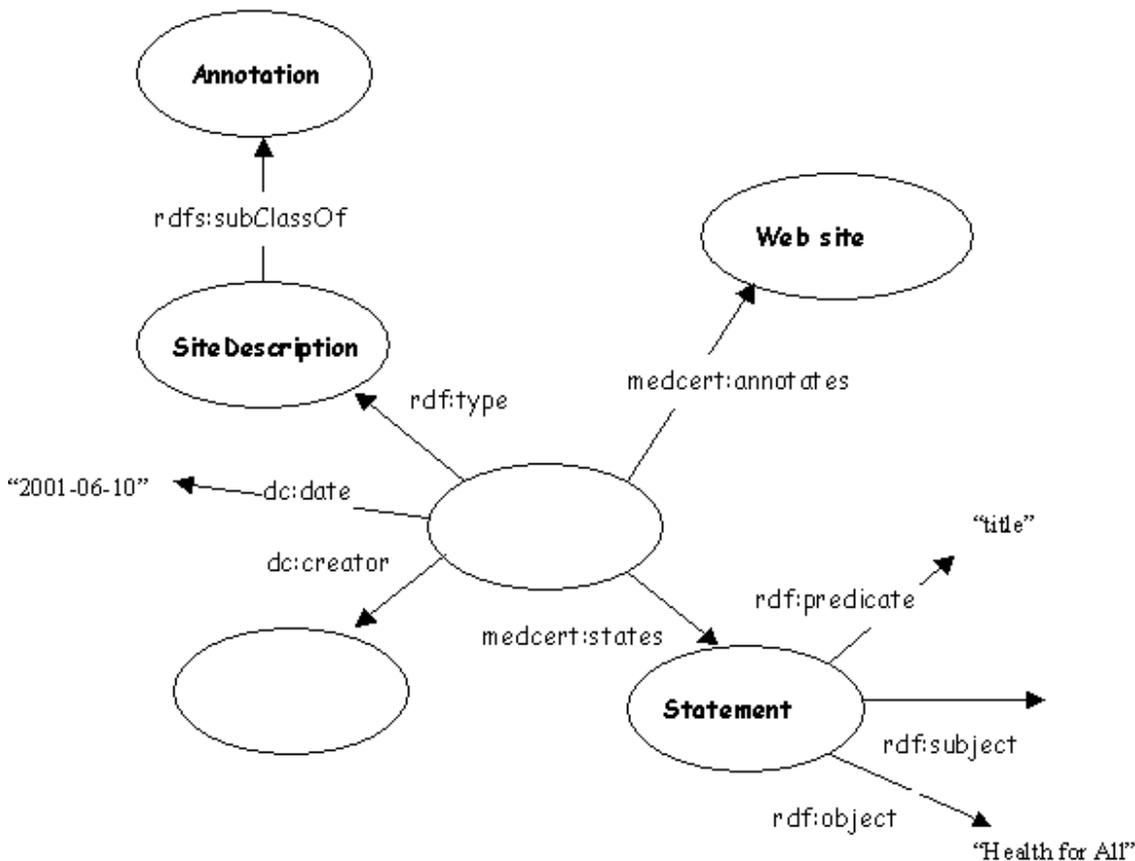


Figure 3. MedCERTAIN annotation in which the information stated is a single RDF statement

The SiteDescription annotation shown above is used for storing general information about a Web site. This could be basic bibliographic information such as that represented by the Dublin Core metadata set, but may also be information relating to the publisher's internal quality procedures, compliance to the Web Accessibility Initiative, etc. For this purpose, the MedCERTAIN project is defining a set of quality criteria metadata.

MedCERTAIN also uses medical experts to validate the information provided by the site publisher; to check, if possible, whether the information is correct and sufficient. This evaluator consequently needs to comment upon, or annotate, an existing SiteDescription annotation. The type of annotation designed to do this is called a *Validation*

annotation, and takes a similar form to the SiteDescription annotation. The difference is that the annotated object will be a SiteDescription annotation and the RDF statement that is 'stated' consists of only one possible predicate: validation, and one of four possible values: 'Not checked', 'Valid', 'Invalid', or 'Cannot say'.

The creator object for any of these annotations can also have a type. In our implementation, the creator of a Validation annotation must always be of type *evaluator*, whereas a SiteDescription annotation may be created by an evaluator or a creator of type *publisher*. The creator object may, of course, be linked to further information about the creator.

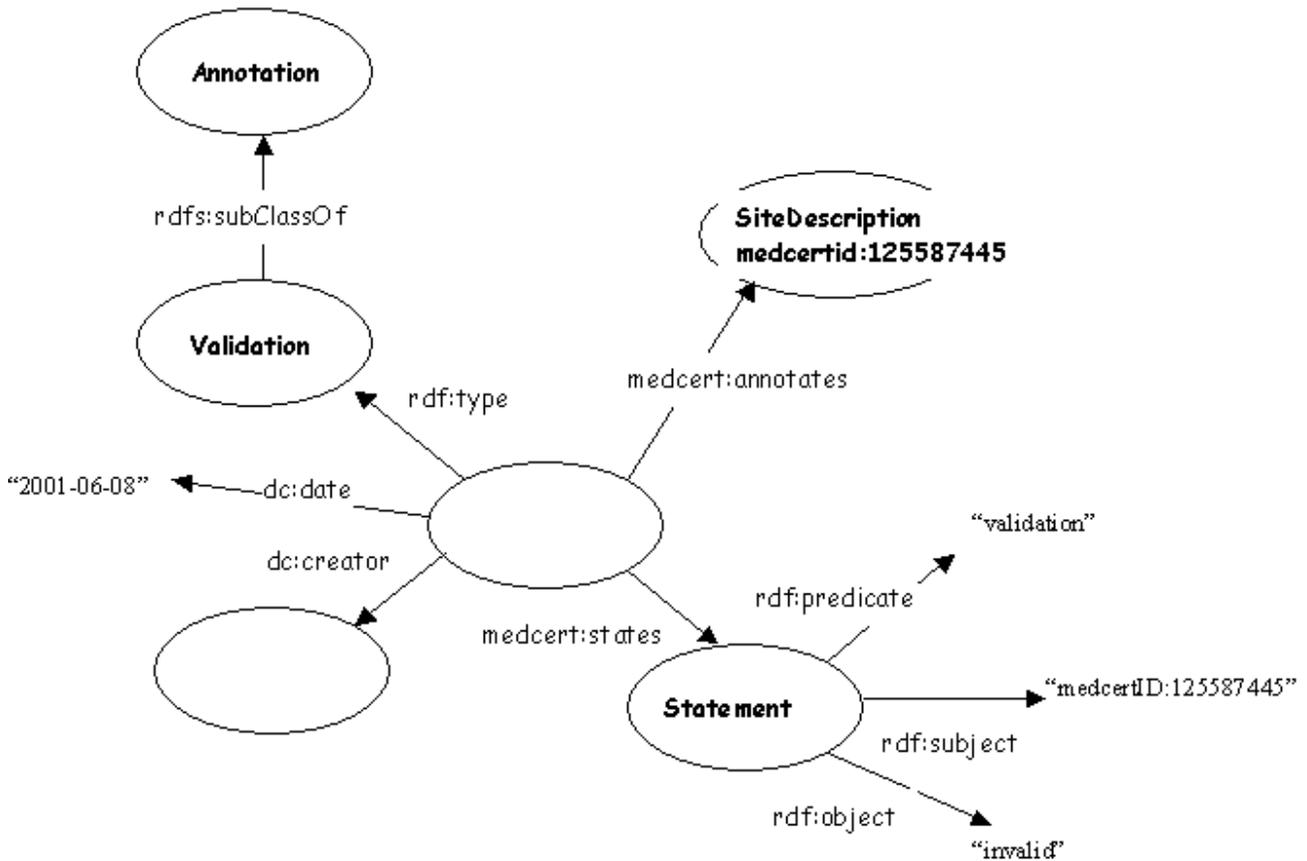


Figure 4. MedCERTAIN validation annotation

There is a third type of annotation used by the MedCERTAIN system called a Comment annotation. This is used to provide further details about the result of the validation if these are required. The Comment annotation can therefore annotate a Validation annotation. The information stated by such an annotation is also restricted to an RDF

statement with one allowable predicate: comment, that has a free-text value. Another predicate that we may use in future with this type of annotation might be called, for example, "reference" and be used to contain specific references to supporting evidence.

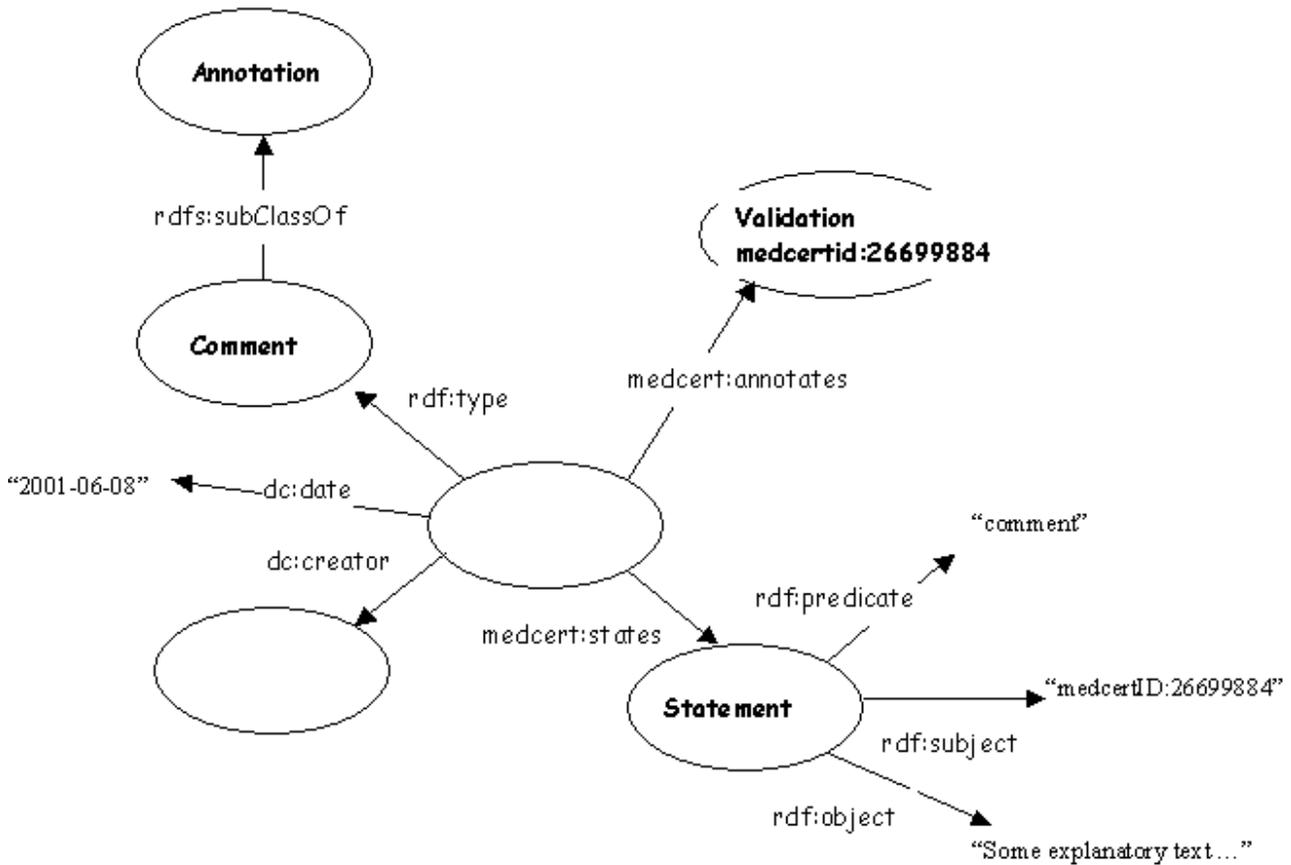


Figure 5. MedCERTAIN comment annotation

The EARL model is similar but has a slightly different focus. Although it uses the same RDF statements format as MedCERTAIN, meaning that it is extensible and flexible as RDF, EARL is focused on evaluation with respect to an evaluative principle that can itself be identified on the web. So statements concern whether a web site or part of a Web site passes or fails (or variants of these) with respect to a URL-identifiable test.

An EARL evaluation is an RDF statement, with a context and an assertion. The context of the evaluation contains information pertaining to the actual creator or generator of the evaluation itself, e.g. giving details for the tool or per-

son running the test, and the platform settings of the machine on which the test was run. This set of content information is hung off of the node called the "Assertor". In other words, the context properties are attached to the person or tools that ran the test.

The second part of the evaluation is the main assertion. This is a simple 3-ary relationship comprised of the resource being evaluated, a result property, and the evaluation criteria that the resource is being tested against. This main assertion is linked to the context using an "asserts" predicate. An example is shown in Figure 6.

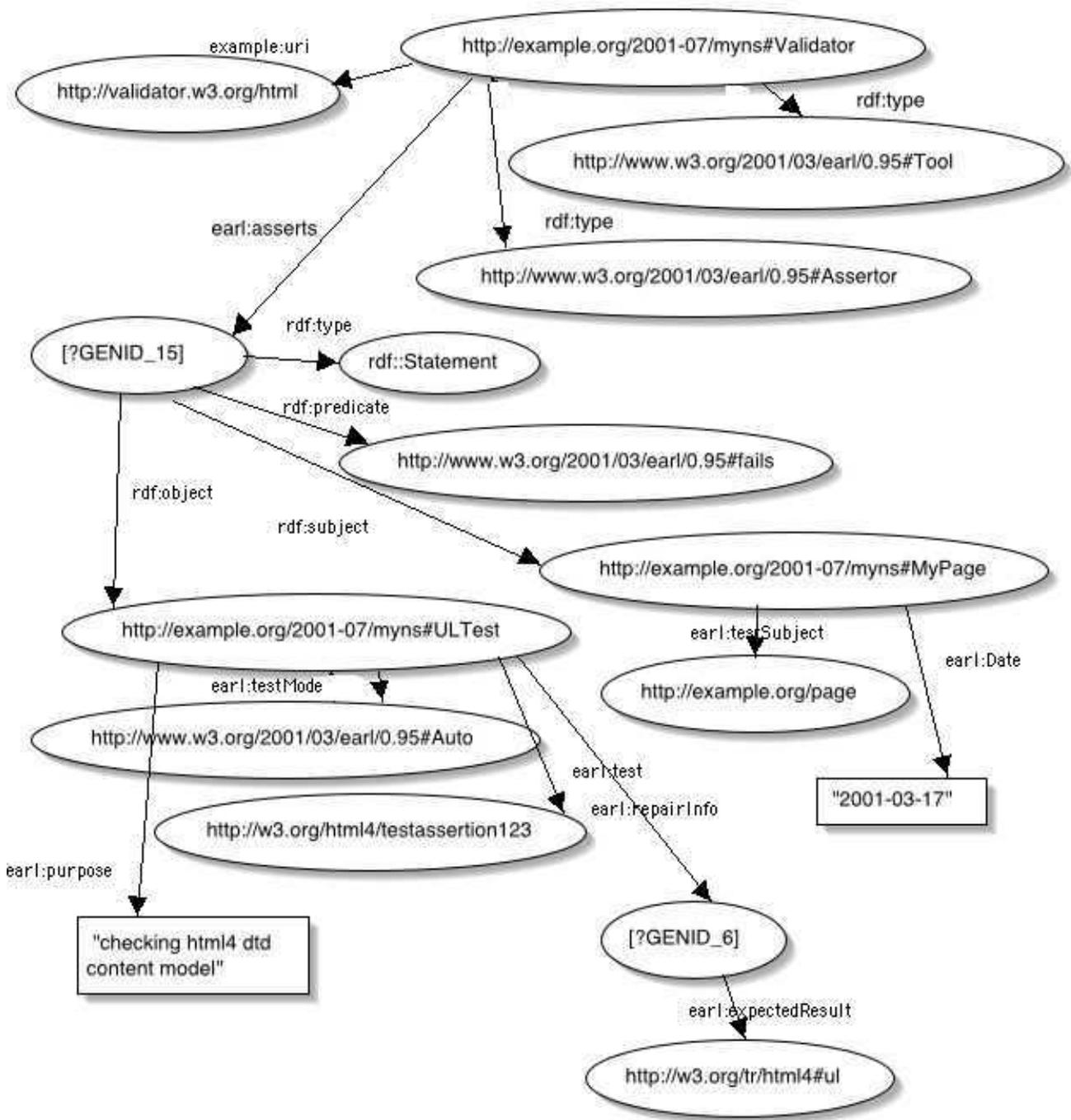


Figure 6. EARL example annotation

Because the test has a URI, RDF can be used to say more about the test, for example, machine-readable expected results information, or a human-readable purpose of the test.

In both models RDF can be used to say more about the person or agent stating the RDF statement, for example, their qualifications (if a person) or their software and hardware if a machine.

Trust models for annotations of any type are usually context based. It is easy to "trust" certain annotations servers that you know can only be accessed by a trusted entity of some sort. Likewise, if XYZ company produce a report of some content and post it on the XYZ Website, you can be certain to a fair degree of satisfaction that this report can be trusted. The question arises when annotations have an unknown state of trust. Digital signatures and Public Key

Infrastructure may help to solve some of these problems in the future.

RDF AND ANNOTATIONS

There are various disadvantages with using RDF in general. RDFS (Schema) is currently only a candidate recommendation. There are some difficulties with the semantics of RDF, which are currently being resolved by the RDFCore working group. The syntax of XML/RDF is verbose and can be difficult for humans to read and understand compared with what might be called "vanilla" XML.

However, RDF is ideally suited to *modeling* annotations because with its node-arc-node structure every RDF link is like an annotation on an object.

RDF was used in MedCERTAIN because of its flexibility and because it can be used to model higher order statements (or 'statements about statements'). It can be used to model and describe annotations about anything, not just about webpages, but about anything with a URI, (for example anything with an Xpointer or XPath) as well as things without a universal identifier (using so called 'anonymous' resources). This includes annotations themselves: RDF enables the modeller to attach provenance to statements using its implementation of higher order statements (the 'reification' mechanism), which is essential for trust for annotations.

What RDF provides over vanilla XML is its built in node-arc-node model. You can of course describe a set of structures which would describe links and provenance in arbitrary XML, which syntactically might look more simple. However, this would essentially mean inventing a new syntax for RDF, since it would be describing the same *model*, but using a different syntax. Despite the faults and verbosity of the RDF syntax, there are now many tools which can parse, store and query XML/RDF data, and this advantage is lost with an invented RDF syntax.

The syntax of RDF must be clearly distinguished both from its model and from the storage implementation. MedCERTAIN makes use of the RDF model to describe annotations, but stores the annotations in an SQL database optimised for the data and queries that are made on the data. The XML/RDF syntax is used for transferring the data.

In the case of EARL, the group responsible for EARL took some months in weighing up the options for data representation, and eventually the choice was narrowed down to two models: a proprietary XML schema based model, or an RDF based model. Investigating the trade-offs between the two, it was found that the RDF based solution was more appropriate. The benefit of using the RDF model with respect to the efficiency of the deployment of EARL is highly obvious: there are many generic RDF parser implementations available that can thus be used to handle EARL.

EARL is demographically targeted to a wide range of disparate entities: corporations, Web accessibility organizations, and even the general public, and so it was important that an interoperable model was chosen. Because the RDF model is already very accurately documented, and discussions clarifying the structure have been made over many years, by using RDF, the group was able to forgo the usual operation of deciding upon a generic framework onto which EARL would fit. In other words, for EARL the question was: why invent a data model when RDF provides one? Why repeat the work?

The third major reason for choosing RDF for the EARL data model was that the group behind EARL (a chartered Working Group of the W3C) hopes to use tools related to the Semantic Web activity of the W3C where possible. This was actually a decidedly useful step to take; in the development of the EARL schema, we proved that it was possible to roughly map version 0.9 of the language into version 0.95 using a forwards chaining query/inference engine written in Python (Tim Berners-Lee's CWM [12]). Although version 0.9 of the language had not been widely deployed, this proved importantly that the EARL model, thanks to it being based on RDF, is evolvable and extensible. Although there is always some trade-off when upgrading a language, Semantic Web technologies make it easier.

EARL have made sure that the model is as syntax independent as possible, and takes a wary standpoint with respect to much-debated model constructs such as reification. EARL does use higher order statements, but these can be expressed as N3 [13] contexts, RDF reification, or something else entirely. What matters is that they are higher-order statements, these will always be around when you have 3-ary relationships, because it is so easy to invent properties.

SUMMARY

EARL and MedCERTAIN are two examples of projects where the agency, the timing, the objects and the meaning of the content of annotations must be clearly defined. The trustworthiness of evaluative applications of annotations such as these depends on the unambiguous identification of the objects of the annotations. Trustworthiness also depends on the possibility of making the annotations distinguishable on the grounds of the agency creating them and the time and date on which they were created.

Both EARL and MedCERTAIN chose to use RDF to model their annotations, for three principal reasons:

- the usefulness of associating objects with URIs as the natural choice for identification of objects on the web;
- the flexibility and extensibility of RDF in creating relationships between arbitrary objects such as agents and dates;

- the ability of RDF to model statements about objects as objects in themselves, enabling the content of an annotation to be made machine-readable.

In addition there are many tools available for processing and storing RDF, and many projects using RDF, enabling interoperability between systems.

RDF is a natural tool for modelling annotations because it is all about describing the properties of objects (such as the title of a webpage, the date changed for this part of a webpage). This includes the ability to describe the properties of the assignment of a property to an object (such as the creator of an annotation). RDF allows the modeling of all the aspects of annotations, such as who made the annotation, when they made it, and what the content is, because it allows both the content of the annotation and the properties of the annotation itself to be modeled in the same system.

ACKNOWLEDGMENTS

Thanks to Dan Brickley and Martin Poulter for reading earlier versions of this paper

The MedCERTAIN project is funded under the EU Safer Internet Action Plan, and consists of: the University of Heidelberg, Dept. of Clinical Social Medicine; the University of Bristol, Institute for Learning and Research Technology (ILRT); and The Finnish Office for Health Care Technology Assessment (FinOHTA) at the Finnish National Research and Development Centre for Welfare and Health (STAKES).

Thanks to the IMesh toolkit which Libby Miller is partially funded by.

REFERENCES

- [1] MedCERTAIN
<http://www.medcertain.org>
- [2] EARL
<http://www.w3.org/2001/03/earl>
- [3] Xpointer
<http://www.w3.org/TR/xptr>
- [4] Brickley, D. Nodes and Arcs 1989-1999
<http://www.w3.org/1999/11/11-WWWProposal/thenandnow>
- [5] Koivunen, M. Annotea
<http://www.w3.org/2001/Annotea>
- [6] The W3C Collaborative Web Annotation Project ... or how to have fun while building an RDF infrastructure
<http://www.w3.org/2000/Talks/www9-annotations/Overview.html>
- [7] Berners-Lee, T. Cool URIs don't change
<http://www.w3.org/Provider/Style/URI>
- [8] Miller, L. Rudolf Project
<http://ilrt.org/discovery/2000/09/rudolf/>
- [9] Lassila, O., Swick, R. (eds.) RDF model and Syntax
<http://www.w3.org/TR/REC-rdf-syntax/>
- [10] Lassila, O., Swick, R. (eds.) RDF Model and Syntax Reification
<http://www.w3.org/TR/REC-rdf-syntax/#higherorder>
- [11] RDF Interest Group Daily Chump
<http://rdfig.xmlhack.com>
- [12] Berners-Lee, T. CWM
<http://www.w3.org/2000/10/swap/cwm.py>
- [13] Berners-Lee, T. Primer: Getting into RDF & Semantic Web using N3
<http://www.w3.org/2000/10/swap/Primer>

Appendix

Sample Rudolf Annotation

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:r="http://rdf.desire.org/vocab/recommend.rdf#">
  <r:Annotation>
    <dc:title>Biz/ed</dc:title>
    <dc:description>A subject gateway for Economics and Business
    </dc:description>
    <dc:identifier rdf:resource="http://www.bized.ac.uk" />
    <r:attributedTo rdf:resource="mailto:libby.miller@bristol.ac.uk" />
  </r:Annotation>
</rdf:RDF>
```

Sample MedCERTAIN Annotations

Please note that the annotation schema and namespace has not been finalized.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:an="http://rdf.desire.org/vocab/recommend.rdf#" >

  <an:SiteDescription rdf:about="http://id.medcertain.org/annotations/1981930821-995021535103">
    <an:annotator rdf:resource="http://medcertain.org/publishers/pub8779" />
    <an:annotates rdf:resource="http://www.acme-health.com" />
    <dc:date>12-07-01</dc:date>
    <an:states>
      <rdf:Statement>
        <rdf:subject rdf:resource="http://www.acme-health.com" />
        <rdf:predicate rdf:resource="http://medcertain.org/hiddel/Sitespecific_content_purpose_target" />
        <rdf:object>adult patients or consumers</rdf:object>
      </rdf:Statement>
    </an:states>
  </an:SiteDescription>

  <an:Validation rdf:about="http://id.medcertain.org/annotations/1981930821-995056925671">
```

```
<an:annotator rdf:resource="http://medcertain.org/evaluators/eval8996" />
<an:annotates rdf:resource="http://id.medcertain.org/annotations/1981930821-995021535103" />
<dc:date>13-07-01</dc:date>
<an:states>
  <rdf:Statement>
    <rdf:subject rdf:resource="http://id.medcertain.org/annotations/1981930821-995021535103" />
    <rdf:predicate rdf:resource="http://rdf.desire.org/vocab/recommend.rdf#validation" />
    <rdf:object>invalid</rdf:object>
  </rdf:Statement>
</an:states>
</an:Validation>

<an:Comment rdf:about="http://id.medcertain.org/annotations/1981930821-995059887654">
  <an:annotator rdf:resource="http://medcertain.org/evaluators/eval8996" />
  <an:annotates rdf:resource="http://id.medcertain.org/annotations/1981930821-995056925671" />
  <dc:date> 13-07-01</dc:date>
  <an:states>
    <rdf:Statement>
      <rdf:subject rdf:resource="http://id.medcertain.org/annotations/1981930821-995056925671" />
      <rdf:predicate rdf:resource="http://rdf.desire.org/vocab/recommend.rdf#comment" />
      <rdf:object>Much of the material is not suitable for a lay audience</rdf:object>
    </rdf:Statement>
  </an:states>
</an:Comment>

</rdf:RDF>
```

Sample EARL Annotation

```
<rdf:RDF xmlns="http://example.org/2001-07/myns#"
  xmlns:earl="http://www.w3.org/2001/03/earl/0.95#"
  xmlns:log="http://www.w3.org/2000/10/swap/log#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:about="http://example.org/2001-07/myns#MyPage">
    <earl:date>2001-03-17</earl:date>
    <earl:testSubject rdf:resource="http://example.org/page"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/2001-07/myns#ULTest">
    <earl:purpose>checking HTML4 dtd content model</earl:purpose>
    <earl:repairInfo rdf:parseType="Resource">
      <earl:expectedResult rdf:resource="http://w3.org/tr/html4#ul"/>
    </earl:repairInfo>

    <earl:test rdf:resource="http://w3.org/html4/testassertion123"/>
    <earl:testMode rdf:resource="http://www.w3.org/2001/03/earl/0.95#Auto"/>
  </rdf:Description>

  <earl:Assertor rdf:about="http://example.org/2001-07/myns#Validator">
    <rdf:type rdf:resource="http://www.w3.org/2001/03/earl/0.95#Tool"/>
    <uri rdf:resource="http://validator.w3.org/html"/>
    <earl:asserts rdf:parseType="Resource">
      <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
      <rdf:subject rdf:resource="http://example.org/2001-07/myns#MyPage"/>
      <rdf:predicate rdf:resource="http://www.w3.org/2001/03/earl/0.95#fails"/>
      <rdf:object rdf:resource="http://example.org/2001-07/myns#ULTest"/>
    </earl:asserts>

  </earl:Assertor>
</rdf:RDF>
```

And, in N3:

```
@prefix earl: <http://www.w3.org/2001/03/earl/0.95#> .
@prefix : <http://example.org/2001-07/myns#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:Validator earl:asserts
  [ a rdf:Statement; rdf:subject :MyPage;
    rdf:predicate earl:fails;
    rdf:object :ULTest ];
  a earl:Tool, earl:Assertor;
  :uri <http://validator.w3.org/html> .
:MyPage
  earl:testSubject <http://example.org/page>;
  earl:date "2001-03-17" .
:ULTest
  earl:test <http://w3.org/html4/testassertion123>;
  earl:testMode earl:Auto;
  earl:purpose "checking html4 dtd content model";
  earl:repairInfo
    [ earl:expectedResult <http://w3.org/tr/html4#ul> ] .
```