

ICRA Labelling Working Group

1 Technical outcomes of first meeting

At the initial meeting of the working group on 9th June, the following points were made:

1. De-referencing the labels and pointing to a central location would work well in many cases and, on first sight, seemed to be welcomed by large content providers
2. Work to classify content into a limited set of categories has already been done by supportive content providers.
3. In a multi-user environment, each user will need their own rating file and pointer to it, to use embedded labels or both.
4. Document level ratings must continue to be supported
5. It should be possible to override any generic label with a document level label.

There clearly is a need for flexibility. One solution won't fit all, but that's OK – RDF is designed to be flexible. The various options are set out below. Some are uncontroversial, others require rigorous examination and leave many questions unanswered.

2 Executive Summary

This paper explores the possible application of RDF to the issues surrounding content labelling in the light of the working group's discussion. Options are explored and, in many cases, reasons are given why they should probably be rejected. The strongest proposals for solutions are that

1. Link tags to RDF documents should include a query string that can convey what the description should be "about" (in the RDF sense of the word). `about= " "` in the query string should be interpreted as meaning "give me an RDF model that has the URI from which this request is being made as its subject". This is outwith the Recommendations but is less problematic than many other options. To paraphrase Churchill's comment on Democracy – it's the worst possible solution apart from all the others.
2. Encoding not only the possible ratings but the rules for their application in a single RDF instance is probably possible but needs a great deal of rigorous examination to produce a robust solution.

3. The robust solution to allowing a document level description to override a more generic description is not clear in some circumstances, although it is for many common situations.

3 Simple RDF in the page

Embedding RDF in HTML, even HTML 4.01, is not valid; however, it can be and occasionally is, done. For example, Ora Lassila, one of the architects of RDF, includes it in his homepage thus:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="Author" content="Ora Lassila">
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc = "http://purl.org/dc/elements/1.1/">
    <rdf:Description rdf:about=""
      dc:title="Ora Lassila"
      dc:description="Ora Lassila's professional home page"
      dc:date="2003-03-12"
      dc:format="text/html"
      dc:language="en"
      dc:creator="Ora Lassila"/>
  </rdf:RDF>
<title>Ora Lassila</title>
```

Note that this is declared as an HTML 4.01 document with the RDF just added in. Simple.

The problem is that, as it contravenes the (X)HTML standards, it's not recommended by the W3C. If we can "park that" for a minute and look in more detail...

ICRA Labels embedded with pages would appear in a similar way to Ora Lassila's data. Notice also the about attribute, i.e. `<rdf:Description rdf:about=""`. This is taken to mean that **the description is about the document carrying it**, whatever its URI.

This is useful as it means that a single label with the `about` attribute set in this way can be included in multiple pages without any need to specify their URLs. This might give us the RDF equivalent of "copy and paste the meta tag into the head section of your pages." An ICRA label would look something like this:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:icra = "http://www.icra.org/ratingsv03#">
  <rdf:Description rdf:about=""
    icra:cz="1"
    icra:lz="1"
    icra:nz="1"
    icra:oz="1"
    icra:vz="1"/>
</rdf:RDF>
```

If there is more metadata than just ICRA labels to add then it's a lot neater to put it all into description one like this:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc = "http://purl.org/dc/elements/1.1/"
        xmlns:icra = "http://www.icra.org/ratingsv03#">
<rdf:Description rdf:about=""
    dc:title="Ora Lassila"
    dc:description="Ora Lassila's professional home page"
    dc:date="2003-03-12"
    dc:format="text/html"
    dc:language="en"
    dc:creator="Ora Lassila"
    icra:cz="1"
    icra:lz="1"
    icra:nz="1"
    icra:oz="1"
    icra:vz="1"/>
</rdf:RDF>
```

But... remember that the learned folk of the W3C say that this isn't a good way to embed ICRA labels in (X)HTML. We'd be going against the internet community if we were to promote this as a solution. So what might be the way forward?

3.1.1 Discussion point

Dublin Core and other metadata, especially UK government e-GMS, is often written in HTML as a series of meta tags like this [DC_HTML]

```
<meta name="DC.Title" content="Welcome to the Home Office">
<meta name="DC.title.alternative" content="Home Office Homepage">
<meta name="DC.contributor" content="as supplied">
<meta name="DC.creator" content="Communications Directorate">
```

Should we make it possible to write ICRA labels in the same way? i.e.:

```
<link rel="schema.ICRA" href="http://www.icra.org/ratingsv03#">
<meta name="ICRA:cz" content="1">
<meta name="ICRA:lz" content="1">
<meta name="ICRA:nz" content="1">
<meta name="ICRA:oz" content="1">
<meta name="ICRA:vz" content="1">
```

3.1.2 Points for:

- Unlike simply including RDF, this *is* valid HTML (for XHTML add the trailing /)
- At least a section of the professional web development community will feel comfortable with it, therefore promoting its usage which is what we want.
- More people use this format than simply putting RDF in their pages (although few do either).

3.1.3 Points against:

- It means working with and supporting another method, creating more work and possibly more confusion for webmasters and others. (The questionnaire would need to include an option to a) create an RDF description; or b) create a series of meta tags – not exactly a user-friendly question!)
- It can only be applied to (X)HTML documents, not other media types
- It doesn't have quite the same rigour and flexibility of RDF.
- There is no compelling advantage over PICS. If a user wants this we may as well just give them a PICS label.

Fundamentally, neither embedding RDF in an (X)HTML page or using meta tags is attractive. An external link is the preferred method.

4 De-referencing the ratings

In this method a file is created that carries a set of ratings, call them r1, r2, r3 etc. A short block of RDF can point to the relevant rating thus:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:p="http://www.purl.org/rating_systems/04/#">
  <rdf:Description rdf:about="">
    <p:rating rdf:resource="http://example.org/allRatings.rdf#r1" />
  </rdf:Description>
</rdf:RDF>
```

Listing 1 Minimum RDF to point to a rating in an external file

<http://example.org/ratings.rdf> would be a file like that shown in Listing 2. This de-referenced solution leaves the rating in the hands of the person generating the page. It would be for them to change “r1” to r2” or whatever at the end of the URL in the rating tag. Other details in the short block of RDF within the page remain the same. A single copy of the actual ratings is maintained in a separate file so that if it were decided to change the description of “r2” it could be done very easily by editing a single file.

Again, further elements could be included in the Description as suited to the content creator. But – we’re back to talking about embedding RDF within HTML which is against the specification of (X)HTML and is not recommended in both the W3C and general sense of the word.

A further problem with this solution that won’t have escaped notice is that it is rather verbose – it takes over 200 characters to say “my rating is at <http://example.org/ratings.rdf#r1>”.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/WD-rdf-schema#"
  xmlns:p="http://www.purl.org/rating_systems/04/#"
  xmlns:movie="http://www.movieratings.org/ratings#"
  xmlns:icra="http://www.icra.org/ratingsv03/#">

  <rdf:Description rdf:ID="r1">
    <rdfs:comment>The label for the chat area of the site, see
    www.icra.org/decode/ for explanation of rating</rdfs:comment>
    <rdfs:label>Chatrooms on this site are unmoderated. They may, but
    are not known to, contain potentially offensive
    language</rdfs:label>
    <icra:ca>1</icra:ca>
    <icra:lz>0</icra:lz>
    <icra:nz>1</icra:nz>
    <icra:oz>1</icra:oz>
    <icra:vz>1</icra:vz>
  </rdf:Description>

  <rdf:Description rdf:ID="r2">
    <rdfs:comment>A label for reviews of action films. There may or
    may not be nudity but violence and weapon promotion is a given,,
    see www.icra.org/decode/ for explanation of rating
    codes</rdfs:comment>
    <rdfs:label>Reviews of action films are likely to contain
    descriptions of violence and to promote weapon use</rdfs:label>
    <icra:cz>1</icra:cz>
    <icra:lz>1</icra:lz>
    <icra:nz>0</icra:nz>
    <icra:oe>1</icra:oe>
    <icra:vb>1</icra:vb>
    <icra:vc>1</icra:vc>
    <icra:vd>1</icra:vd>
    <icra:ve>1</icra:ve>
  </rdf:Description>

  <rdf:Description rdf:ID="r3">
    <rdfs:comment>A label for video clips</rdfs:comment>
    <rdfs:label>Film clips on this site have been chosen so as not to
    contain any potentially offensive material</rdfs:label>
    <icra:cz>1</icra:cz>
    <icra:lz>1</icra:lz>
    <icra:nz>1</icra:nz>
    <icra:oz>1</icra:oz>
    <icra:vz>1</icra:vz>
    <movie:rating>U</movie:rating>
  </rdf:Description>

  <rdf:Description rdf:ID="default">
    <icra:cz>1</icra:cz>
    <icra:lz>1</icra:lz>
    <icra:nz>1</icra:nz>
    <icra:oz>1</icra:oz>
    <icra:vz>1</icra:vz>
  </rdf:Description>
</rdf:RDF>

```

Listing 2 Simple RDF file with multiple ratings. Referred to in examples as <http://www.example.org/allRatings.rdf>.

5 Using a link tag or HTTP header to point to a rating

This is rather shorter...

```
<link rel="meta" type="application/rdf+xml" href="ratings_1.rdf" />
```

The MIME type of application/rdf+xml has now been accepted [RFC-3023]

The same link can also be encoded in an HTTP response Header thus:

```
Link: <ratings_1.rdf>; /="/"; rel="meta"; type="application/rdf+xml"
```

Such a link might point to a short document that contained the information in Listing 1 or a slightly longer document that contained the actual rating information as well as the details of what it was about. In other words, ratings_1.rdf could be something like this that continues to point to another file that contains several ratings, the one we want being identified by the #r1 fragment identifier:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:p="http://www.purl.org/rating_systems/04/#">
  <rdf:Description rdf:about="">
    <p:rating rdf:resource="http://example.org/allRatings.rdf#r1" />
  </rdf:Description>
</rdf:RDF>
```

Listing 3 A repeat of Listing 1 presented as possible contents of ratings_1.rdf

or all the information can be in one file like this:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:p="http://www.purl.org/rating_systems/04/#">
  <rdf:Description rdf:about="">
    <p:rating>
      <rdfs:comment>The label for the chat area of the site, see
        www.icra.org/decode/ for explanation of rating</rdfs:comment>
      <rdfs:label>Chatrooms on this site are unmoderated. They may,
        but are not known to, contain potentially offensive
        language</rdfs:label>
      <icra:ca>1</icra:ca>
      <icra:lz>0</icra:lz>
      <icra:nz>1</icra:nz>
      <icra:oz>1</icra:oz>
      <icra:vz>1</icra:vz>
    </p:rating>
  </rdf:Description>
</rdf:RDF>
```

Listing 4 A complete description about "". This is an alternative structure for "ratings_1.rdf".

5.1 *Is this the magic solution? (Sadly no)*

And this is where we hit to big problem. RDF is about triples:

Subject – Predicate – Object

e.g.

This person – Has the name – John.

The problem, the whole problem from ICRA's point of view, is specifying the Subject. In Listing 3, the Predicate – the rating – is specified by the tag

```
<p:rating ...
```

The Object, i.e. the value of the rating, is given by the remainder of that tag:

```
rdf:Resource="http://example.org/allRatings.rdf#r1" />
```

This “resource” is actually a set of Predicate/Object pairs. In Listing 4 the Predicate is still `rating` but the subsequent Predicate/Object pairs follow directly below.

But what's the Subject?

This is given in the Description tag:

```
<rdf:Description rdf:about="">
```

As noted earlier, this means that the description is *about the document itself*. Well, in our example the document is `http://www.example.org/ratings.rdf`, **not** the HTML document that linked to it. The information at `http://www.example.org/ratings.rdf#r1` or given directly below it – i.e. the Predicate/Object pairs we call rating 1, are describing the RDF document, not the thing we actually want to describe.

We need a way to “write in” the about attribute. (An alternative would be to supply an XML Base URI – it seems to amount to the same thing.)

5.2 *Writing in the about attribute dynamically*

This forces us to look at some sort of dynamic method. Perhaps the Link/HTTP header could point to a dynamic page that returned a version of Listing 3 with the about string written in? This would not be hard to do – but it means putting a bit more load on the server (never popular) and that we'd be generating the whole of Listing 3 for every resource that pointed to it, changing just one small part which is the URI of the thing that called it in the first place – which has a ring of pointlessness about it.

Some example data might help to visualise the problem.

I have two HTML pages like this:

```
<html>
<head>
  <title>Page 1</title>
  <link rel="meta" type="application/xml+rdf" href="ratings_1.php" />
</head>
<body>...
```

Example 1 "Page 1" points to ratings1.php

```
<html>
<head>
  <title>Page 2</title>
  <link rel="meta" type="application/xml+rdf" href="ratings_1.php" />
</head>
<body>...
```

Example 2 "Page 2" points to ratings1.php

Both pages point to the same URI for the metadata which, you'll notice, I've now changed to a .php document so I can write the about attribute dynamically.

The browser/filter visits page 1, notices the RDF link and makes a request for which it gets back.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:p="http://www.purl.org/rating_systems/04/#">
  <rdf:Description rdf:about="http://www.example.org/page1.html">
    <p:rating rdf:resource="http://example.org/allRatings.rdf#r1" />
  </rdf:Description>
</rdf:RDF>
```

Listing 5 RDF header-load to point to rating for specified page

Note the about attribute which is the URI of the document from which we learned about the existence of this bit of RDF.

The user now goes on to page 2 and another request is sent for which it receives

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:p="http://www.purl.org/rating_systems/04/#">
  <rdf:Description rdf:about="http://www.example.org/page2.html">
    <p:rating rdf:resource="http://example.org/allRatings.rdf#r1" />
  </rdf:Description>
</rdf:RDF>
```

Listing 6 RDF header-load to point to rating for specified page

Everything is the same except the about attribute which is now the URI of the second document.

That's a whole HTTP request and server task to generate a file that contains data we've already got to wrap around a variable we already know!

Surely we need this:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:p="http://www.purl.org/rating_systems/04/#">
  <rdf:Description rdf:about="$HTTP_REFERER">
    <p:rating rdf:resource="http://example.org/ratings.rdf#r1" />
  </rdf:Description>
</rdf:RDF>
```

Listing 7 RDF header-load to point to rating for generalised page

A user agent would be capable of doing this without more than one round trip to the server, but how would it know to do it?

Maybe we could promote a new relationship for link tags:

```
<link rel="dynamic_meta" type="application/rdf+xml"
href="http://www.example.org/allRatings.rdf#r1" />
```

Example 3 Link tag with new "REL" of dynamic_data

The new "Rel" of "dynamic_meta" would imply that the RDF at the given URI will require the insertion of the URI of the document holding the link. There is no specification for Rel, it's open to do what you like with – which makes it a very flexible system but also a non-standard one that may or may not have any credibility.

An alternative approach might be to encode the same "keep hold of this URI and add it into the variable field in this bit of RDF" by defining a new MIME type, say "d+rdf+xml". (MIME types are subject to international standardisation through the IETF). A link tag would then look like:

```
<link rel="meta" type="application/d+rdf+xml"
href="http://www.example.org/allRatings.rdf#r1" />
```

Example 4 Link tag with a new MIME type

Finally, the "treat this differently" information might be encoded in a query string after the URL of the RDF data thus:

```
<link rel="meta" type="application/rdf+xml"
href="http://www.example.org/ratings.rdf?about=%22%22" />
```

(NB. Double quotes cannot be included in URLs directly and have to be encoded as %22).

Example 5 Link tag with query string on href

Either the server or, more likely, the client, will interpret this as a request for RDF *about the requesting URI*. Of the three approaches outlined, this one seems to have the edge.

5.2.1 Points for:

- It uses the existing Recommended Link tag “Relationship” and MIME type.
- It passes information to a URL from the client in line with the principles of web architecture/HTTP requests.
- By putting `about=""` in the link tag we are stating there and then that we want RDF about the resource we’re on at that point, not the one we’re about to request.
- Content providers could include multiple Link tags/HTTP headers like this in which they would specify URIs for which RDF should be generated, rather than/as well as “”.

5.2.2 Points against:

- If the RDF client that receives the data does not process the query string, the resultant RDF is meaningless and may result in the generation of an error message. The other examples would be less likely to be processed by such a client in the first place.

All of the above “solutions” have problems but let’s imagine that one or other proves acceptable. How would it work?

1. Client makes a request to the desired URL for the page or whatever it is
2. Client notices the RDF link, either in a Link tag or an HTTP header and makes a GET request to the URL specified in the href which includes the query string
3. One way or another, the URL used in step 1 is inserted into the received RDF instance as the value of the `about` attribute(s). The RDF would need to include a variable name where the `about` information should go.
4. RDF processing continues in the normal way.

5.2.3 Points for:

- We’re within RDF Recommendations, only the method of generating it is new
- Therefore none of the flexibility and power of RDF is compromised.

5.2.4 Points against:

- RDF wasn’t designed with this kind of dynamism in mind. It was designed to provide a layer of static data that could be interpreted by dynamic systems to create the Semantic Web. But – we have to do *something!*.

5.3 Server side or client side processing?

All of the solutions above allow for the `about` attribute to be written by either the client or the server. It seems most likely that the client would do it since it is the client that “wants” the information which is provided as an option. But, there’s nothing to stop the dynamic elements being created server side. As with other web technologies, server side solutions are less prone to errors being caused by clients that don’t necessarily follow all the rules. Since we’re suggesting a solution that we know *could* lead to an error – creating an RDF model that is only completed by dynamic processing – the possibility of a server side solution should probably not be overlooked.

5.4 Addressing the objectives

The possible solutions described in section 5.2 all provide a means for linking any number of resources with a single specified description. The Predicate/Object pairs may be contained in the same block as the one that defines the XML namespaces and the Subject of the Description (Listing 4), or held in a separate file that might contain different sets of data identified by URL fragments (`#r1`, `#r2` etc. see Listing 3) – the system is entirely flexible.

In a situation like `blogwise.com` where many users can be asked to select one of a number of ratings for their blogs, their selection can be translated into a link to the relevant RDF document, either through a Link tag or an HTTP header pointing to one of the available ratings.

A multiple-user server can be configured so that a given user’s site always points to a given RDF document, which may contain one or multiple ratings and other stuff, it’s all pretty straightforward. Once a Link has been associated with an RDF instance, it can be cached and applied by the client without further HTTP traffic and server load.

The drawback for some users is that it’s the content producer or server set up that is delivering the pointer. Similar functionality is possible now with PICS. It’s possible now to set up a server to deliver a rating with a given set of URLs. What some content providers – including prime internet properties – are asking is that not just the labels are de-referenced but so are rules for deciding which rating applies to what. Asking content production staff to choose and include ratings, or server engineers to configure servers to do extra processing is not popular! What’s demanded is a method of making *everything* point at exactly the same place with the same URI.

This takes us back to the example in the working group primer although now with a revised approach bearing in mind the W3C’s Data Access Working Group’s recent publication.

6 Ratings and rules in one

This is where we're getting into potentially deep water.

Listing 8 is a reworking of Listing 2 with two additional pieces of data. First, the `about` attribute is written as a variable. Second, there's an added piece of data for each rating – a pattern to match. The suggestion is that this can be a simple text string/substring match or a regular expression, either way the intention is straightforward “if the pattern matches, use this rating.”

In XML documents the order of the data matters and is generally preserved. We should be able to specify that a filter should go through the data in the order provided and apply the first match it comes to. The default rating therefore is at the end. Using “RDF Sequence” may be a way to specify this of itself – further input required!

The attractions of this method were discussed in the primer. What's needed now is consultation with a variety of experts to see whether it is a workable solution.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/WD-rdf-schema#"
  xmlns:p="http://purl.org/rating_method/"
  xmlns:movie="http://www.movieratings.org/ratings#"
  xmlns:icra="http://www.icra.org/ratingsv03/#">

  <rdf:Seq rdf:about="$var">
    <rdf:li rdf:resource="http://example.org/ratings.rdf#r1" />
    <rdf:li rdf:resource="http://example.org/ratings.rdf#r2" />
    <rdf:li rdf:resource="http://example.org/ratings.rdf#r3" />
    <rdf:li rdf:resource="http://example.org/ratings.rdf#default" />
  </rdf:Seq>

  <!-- NB. RDF specifies that if a resource is another URI, it must be
  written in full, URIrefs can't be used. The above resources point to
  the ratings lower down the page -->

  <rdf:Description rdf:ID="r1">
    <rdfs:comment>The label for the chat area of the site, see
    www.icra.org/decode/ for explanation of rating
    codes</rdfs:comment>
    <rdfs:label>Chatrooms on this site are unmoderated. They may, but
    are not known to, contain potentially offensive language
    </rdfs:label>
    <p:pattern>
      <p:contains>"/chat/"</p:contains>
    </p:pattern>
    <icra:ca>1</icra:ca>
    <icra:lz>0</icra:lz>
    <icra:nz>1</icra:nz>
    <icra:oz>1</icra:oz>
    <icra:vz>1</icra:vz>
  </rdf:Description>

  <rdf:Description rdf:ID="r2">
    <rdfs:comment>A label for reviews of action films. There may or
    may not be nudity but violence and weapon promotion is a given.
    </rdfs:comment>
    <rdfs:label>Reviews of action films are likely to contain
    descriptions of violence and to promote weapon use</rdfs:label>
    <p:pattern>
      <p:regEx>/.*\//filmreview\//action.*</p:regEx>
    </p:pattern>
    <icra:cz>1</icra:cz>
    <icra:lz>1</icra:lz>
    <icra:nz>0</icra:nz>
    <icra:oe>1</icra:oe>
    <icra:vb>1</icra:vb>
    <icra:vc>1</icra:vc>
    <icra:vd>1</icra:vd>
    <icra:ve>1</icra:ve>
    <icra:vk>1</icra:vk>
  </rdf:Description>

```

```

<rdf:Description rdf:ID="r3">
  <rdfs:comment>A label for video clips</rdfs:comment>
  <rdfs:label>Film clips on this site have been chosen so as not to
  contain any potentially offensive material</rdfs:label>
  <icra:cz>1</icra:cz>
  <icra:lz>1</icra:lz>
  <icra:nz>1</icra:nz>
  <icra:oz>1</icra:oz>
  <icra:vz>1</icra:vz>
  <movie:rating>U</movie:rating>
</rdf:Description>

<rdf:Description rdf:ID="default">
  <rdfs:comment>The default label that declares "None of the above"
  in all categories, see www.icra.org/decode/ for explanation of
  rating codes</rdfs:comment>
  <rdfs:label>Unless otherwise stated, there is no potentially
  offensive material</rdfs:label>
  <p:pattern>
    <p:regEx>/.*</p:regEx>
  </p:pattern><icra:cz>1</icra:cz>
  <icra:lz>1</icra:lz>
  <icra:nz>1</icra:nz>
  <icra:oz>1</icra:oz>
  <icra:vz>1</icra:vz>
</rdf:Description>
</rdf:RDF>

```

Listing 8 RDF instance with multiple ratings and pattern matching data

7 Overrides

An issue not specifically addressed above is how a description pointed to by a resource would be interpreted as one that should override any more generic label available by processing information that may already held in cache.

One possible route would be to consider that an RDF description that included a matching URI in the `about` attribute of the Description tag, i.e.

```
<rdf:Description rdf:about="http://example.org/specific_page.html"
```

should override any RDF instance in which an `about` attribute were specified dynamically.

This would, no doubt, present serious problems. It's non-standard, it goes against not just the implementation of RDF but the central ideas that make it such a strong system. Section 5.2 goes to a lot of trouble to create correct RDF for a specified resource that can then be processed by standard means.

The call from the relevant experts will be that the generic labels mustn't be pointed to in the first place for these "special case" URIs.

If a single file contains both the ratings and the rules for their application (section 6) then it is easy to add in a specific rating at the top of the sequence.

Where multiple resources point to a URL for a single rating, the situation may be more complicated.

How would a set-up like blogwise.com “switch off” the template element that wrote in usual link and provided a custom link for that page? How much extra programming would be required within Helm or how much effort would be required to adjust the Apache configuration to achieve this for HTTP Response headers?

For web pages that are created mostly by hand, or that can be edited individually, it’s easy. If we have a single file for ratings and rules, it’s easy – it’s the ones between the two where rating application is semi-automatic that poses the biggest challenge.

8 References

[DC_HTML] Powell, Andy, Expressing Dublin Core in HTML/XHTML meta and link elements. <http://www.dublincore.org/documents/dcq-html/>

[RFC-3023] See <http://www.ietf.org/rfc/rfc3023.txt> section 8.18.