Helsinki document

Thomas Krichel

2003–12–01

0. This document is the Helsinki document. Its initial version was written by Thomas Krichel on 2003–08–31 to prepare for a meeting with Ivan Kurmanov in Moscow 2003–09–02 to 2003–09–05. The latest version may be found at http://openlib.org/home/krichel/work/helsinki.html. Thomas Krichel and Ivan Kurmanov are very grateful for the excellent hospitality of Evgenia G. Stupina during the Moscow meeting.

1. The aim of this document is to set out the requirements for stage two of the `ACIS` project. This contents of this version has been agreed with Ivan Kurmanov. As such it is a submission to the steering committee.

1 Background on reference parsing

2. Reference parsing is the interpretation of references in conventional scholarly papers. By "conventional" scholarly papers we mean papers that are not networked in an obvious way, i.e. through clickable hyperlinks. In such conventional papers, links to other papers appear as short plain strings. At the end of the linking paper, or in its footnotes, are there short strings are resolved into longer strings. These longer strings are called references. They contain descriptive metadata about the paper being linked to.

3. Autonomous reference parsing means running a computer program that finds the references in a paper, and splits them into metadata components about the paper being referred to. This process is entirely conducted by computer. "First-generation" reference parsing, as pioneered by CiteSeer, and also implemented in CiteBase and CitEc, aims to parse the string into metadata components by just looking at the string on its own.

4. It should be possible to implement "second-generation" reference parsing. Such parsing will have to be based in part, on intelligence that has been provided by humans. Examples are

- Find journal names in the reference data from a dataset about scholarly journals and the way that they are being abbreviated.
- Find author data through parsing reference using known author names.

Such a list of authors can be provided by ACIS stage one.

2 Background on ACIS

5. ACIS is a project funded by the Open Society Institute. Its basis is the Montréal document. That document may be found at http://openlib.org/home/krichel/work/montreal.html.

6. The ACIS software has two objectives

- It serves the authors' vanity.
- It improves bibliographic data.

Both objectives are intimately related. By serving the vanity, it extracts labor from the authors that improves the bibliographic data. If it was not for the improvement of the bibliographic data, the service to the authors' vanity would remain a private benefit to them that would not justify the efforts we put into creating ACIS.

7. The Montréal document has the following to say about stage 2

37. In stage two, the project will be extended from the authorship of documents to the authorship of references contained in other documents. The system will scan reference data for the occurrence of the name of an author, and ask two questions. First, is this you who is being cited in this paper? Second, is this paper part of your research profile, i.e. the list of papers that is already available? We know that authors are very interested in obtaining data on references to their works.

38. For PhysNet and RePEc, reference data is available through the Open Reference and CitEc projects, respectively. For rclis, reference data could be gathered through collaboration with CiteSeer, but for the moment this is out of the scope of the proposal. It is an option that will have to be studied. The ACIS project will fund the conversion of metadata provided by the reference data sources to a common subset of the Academic Metadata Format, that will be used for input into the database.

39. ACIS will export the value-add reference data for usage by the contributing reference indexes. A precise way of doing this will have to be agreed between participants.

8. For those authorities—i.e. people who run ACIS-based collections and services—where the visibility of services using the contributions profile—i.e. the data linking a personal record with a list of authored publications+emadash()is low, the integration of reference data provides an alternative way to attract contributors to an ACIS service. Authors have a strong desire to find out

- how many times they have been cited;
- which papers are citing their work;
- which of their own papers have made the most impact as measured by citations.

This is the first and most important motivation of integrating references data into ACIS.

9. A second motivation for reference data in ACIS is to aid automated reference parsing, i.e. prepare for "second generation" reference parsing services.

10. There are two things that need to be clarified before we can proceed to integrate references into the ACIS.

- We need to set out how we introduce the data to the users, what facilities we offer etc. These are the "presentational" issues.
- We need to set out how we read the data into the ACIS and how we write it out. These are the "representational" issues.

A common challenge to both is how to deal with references in general.

3 Basic theoretical issues

11. There are at least two types of bibliographic data in an ACIS based system.

- There are authoritative data. These are data the authority has built from trusted bibliographic sources.
- There are reference data. These are data about documents that appear in document bodies.

We assume that an authority holds authoritative data for the documents that the references appear in.

12. A document described by a record in the reference data may or may not be described by a record in the authoritative data. If a reference describes a document with authoritative data, we call it an inside reference. If a reference describes a document outside the authoritative data, we call an outside reference. Initially we have no general way to distinguish if a reference is inside or outside.

13. For the users, the difference between inside and outside reference is not always clear. Hiding the difference, or making it obsolete for the users to comprehend it, is important to stage two of ACIS.

14. A different, but important problem with references is that references are not limited to the description of only one work. Consider, for example "A.U. Thor "A work" in E. D. Itor (ed.) "An edited collection"". This string says something about the work and the collection. We need to deal with this in a simple way. Therefore we decide that the "in E. D. Itor

(ed.) “An edited collection”” part is considered to be location information. It is not part of E. I. Ditor’s citation profile. Users can only associate with references to works that they have written. They can not assaciate with reference to works that appear in a collection that they have edited. Of course, they can still associate with the collection itself, but that is a matter for the contributions profile.

15. We will devise a simple algorithm that will find if there is a similarity between a reference and the authoritative record for a document. If a reference can be found that it is similar to a authoritative record of a document, we will say that the document is an “interesting” document. If one document has more references that are similar to its authoritative record than another document in a given contributions profile, it is called a “more interesting” document.

4 Presentational issues

16. A stage two ACIS service has a citations profile. Users will manage it on the “citations profile” screen. The citations profile will contain a list of references, made by documents is the authoritative collection. These references contain a variation of the author’s name.

17. The lookup of name in the reference data is expensive in time. On initial registration of a author, the citations profile will be empty. As soon as a user has requested the addition of a personal profile, a search for the author in the reference data is being conducted. While this search is conducted, and if the user visits the citations screen during search time, the user will be informed that the search is in progress. While the search is in progress the user can request to see a list of refereneces that have been found. The user can only process these citations when the search has finished.

18. If there has been a change in the reference dataset that the citations profile is based upon, all citations profiles of all users are updated. To ensure that incremental updates are possible, the system will be able to distinguish references that are already known to the system from new ones. When the reference dataset is updated, only the updated references are being examined for similarity with author names.

19. To find references for a person, we use the name variations data of the personal profile. Every time a user changes the name variations profile, a new search for references will be performed. If the user changes an authors name variations profile, the reference data will be examined for the modified name variations. During the update, the user can not make changes to the citations profile.

20. By default, the reference parser searches only look for exact matches of the name variations. For better name recognition in references, we need to have approximate string matching. If it is not too slow, we will give users a choice between exact and approximate matching. This is an item to investigate further.

21. When the citations profile is not locked, users can perform four operations on individual references. These are

1. verification
2. identification
3. document item creation
4. revelation

Verification is the first operation that users perform. Then they perform iterations on identification and document item creation. Revelation is an operation that can be performed at any time.

22. After every reference search, users will be invited to proceed to verification if there are some new references found. Verification is the operation by which users tell us that they authored the document described in the reference. This can be done with a check box next to the reference string. When this operation is finished, the software saves the non-verified references as refused references. Therefore, at a new reference search, the already refused items will be discarded before they reach the users that have already refused them.

23. Identification is the operation by which the user associates references data with document data. We say that reference and document data are associated if the author tells us that the reference is a description of the document.

24. At the identification operation, we proceed by documents. We consider all documents in the contributions profile by decreasing level of interest for the new references found. Each document page has a representation of the document and

a list of all new similar references. Each new similar reference has a check box on the left that is checked by default. If users uncheck the box, they tell us that this reference does not describe the document that they examine.

25. Below the lists of proposed new references to a document, there is the list of already associated references. This list is clearly separated from the list of new references. Each reference will have a check box to the right side. Unchecking this removes the association between reference and document.

26. The identification operation will leave us with references that either are not similar to any of the documents, or, the similarity of which has been discarded by the users. Each of these reference has a button “new document”. This leads to the document item creation operation.

27. The document item creation operation will allow users to submit some document information about about an item found in a reference. After much debate, we resolve that since the reference is an unparsed string, users will only be able to edit the string. Ivan Kurmanov still thinks that adding a structured editing capability will be better.

28. The separate operation, that can be performed at any time, is revelation. For each reference, revelation shows users which source documents contains the reference. This can be done in a separate link for each reference.

5 Representational issues

29. Input into ACIS is represented in AMF. It is of the form

```
<text ref="id">
 <reference>
  <literal>text of reference</literal>
 </reference>
</text>
```

The attribute on the element may also be `id` instead of `ref`. Other AMF elements may be present but they will be ignored.

30. An outside reference is identified by a collated form of its string representation. If the collated form of the string changes, any data created by ACIS for that reference will be lost.

31. After verification and before identification, there are simply a range of AMF texts nouns in the personal profile. Each text appears on its own. In the case of three references

```
<person id="id">
 <isauthorof>
  <text>
   <iscitedby>
    <text ref="ref">
     <reference>
      <literal>text of reference one</literal>
     </reference>
    </text>
   </iscitedby>
  </text>
 </isauthorof>
 <isauthorof>
  <text>
   <iscitedby>
    <text ref="ref">
     <reference>
      <literal>text of reference two</literal>
     </reference>
    </text>
   </iscitedby>
  </text>
 </isauthorof>
```

```
 <isauthorof>
  <text>
   <iscitedby>
    <text ref="ref">
     <reference>
      <literal>text of reference three</literal>
     </reference>
    </text>
   </iscitedby>
  </text>
 </isauthorof>
</person>
```

32. After identification, inside references are represented in the personal profile as follows.

```
<text ref="ref">
 <iscitedby>
  <text ref="ref"/>
 </iscitedby>
</text>
```

Note that the literal of the reference could still be there, but may be omitted to save space. If its collated representation changes, the association between the texts does not disappear. There is no ambiguity, because both the source and the target of the reference are identified.

33. For documents that have been created at the document item creation operation, the edited string string data is marked up with a special `acis:user_document` tag, as in

```
<text ref="ref">
 <reference>
  <acis:user_document>
   <literal>edited reference string</literal>
  </acis:user_document>
 </reference>
</text>
```

34. Identification and document item creation groups references to the same document together. Here is a complete example

```
<person id="id">
 <isauthorof>
  <text>
   <iscitedby>
    <text ref="ref">
     <reference>
      <literal>text of reference one</literal>
     </reference>
    </text>
   </iscitedby>
   <iscitedby>
    <text ref="ref">
     <reference>
      <acis:user_document>
       <literal>edited text of reference two</literal>
      <acis:user_document>
     </reference>
    </text>
   </iscitedby>
  </text>
 </isauthorof>
 <isauthorof>
  <text ref="ref">
```

```
    <iscitedby>
      <text ref="ref"/>
    </iscitedby>
   </text>
 </isauthorof>
</person>
```