情報知識学会誌 2008 Vol. 18, No. 1

論文

分割・統合可能な組織内 Web アーカイブシステムの構成方法

An Architecture of Institutional Web Archiving System that have functions to Merge and Split Archives

柊和佑*1, 阪口哲男 1, 杉本重雄 1

Hiiragi Wasuke, Sakaguchi Tetsuo, Sugimoto Shigeo

*1 筑波大学 大学院図書館情報メディア研究科

Graduate School of Library, Information and Media Studies, University of Tsukuba

〒305-8550 茨城県つくば市春日 1-2

E-mail :{ragi, saka, sugimoto}@slis.tsukuba.ac.jp

毎日、膨大な数のWebページが発信者によって作成・更新され、それに伴って消えるWebページも多く、近年は様々な組織がWebページのアーカイブに取り組んでいる。既存のWebアーカイブは独自に収集と提供を行うことが多く、発信者の意向を考慮した運用は困難である。そこで、先に我々は、発信者がWebアーカイブと連携することで、収集と提供を効率よく行う組織内Webアーカイブを提案した。この組織内Webアーカイブの研究では、アーカイブの運営組織に改組が起きた場合の対応については未検討であった。そこで、本論文では改組の際に、複数のWebアーカイブが協調的に動作し、Web アーカイブを維持する方法を提案する。改組を分割、合併、消滅に分類、それらに共通した他の Web アーカイブのデータを取り込む方法を確立し、これを統合と呼ぶ。統合では、各Web アーカイブで使われているメタデータのスキーマ変換と、データの分割を行う。本論文では統合を自動化するために、一連の処理を整理し、変換規則と分割条件を表現する記述を定義している。

Archiving Web content is an important topic for digital libraries. Usually, Web archiving systems freely can collect, preserve and provide. However, Web archiving systems have disadvantages: it is difficult to collect all versions of a resource. We proposed a Web archiving system which is designed to collect resources in accordance with a resource archiving policy determined by the person or organization which provides the resources on the Web. This system was designed without damage to the Web archive by reorganization. The Web archives should cooperate to reduce damage by the reorganization, and the cooperative task is to split archive and convert schema. We call this cooperative task the merge method. This paper describes the model of the Web archiving system which have functions of split and merge archives, and a prototype system implemented based on the model.

キーワード: 組織内 Web アーカイブ, 協調的 Web アーカイブ, イントラネット、ポリシー指向 Web アーカイブ

Keyword: institutional web archiving, cooperation of web archiving system, intranet, policy based web archiving

1 はじめに

現在、膨大な数の情報資源がWebページとして発信されている。その内容は日々作成・更新され、それに伴って消えてしまうものも多い。そのため、近年は様々な組織がWebページのアーカイブに取り組んでいる。

既存のグローバルなWebアーカイブは、運用を行う組織が、対象となるWebページ、その収集タイミング、蓄積方法、利用方法を決定している。そのため、発信者は自らのWebページがどのようにアーカイブされ、どのように利用されているのか判らない。また、運用者にとって、発信者によるWebページの更新頻度がわからないため、網羅性の高いWebアーカイブを構築することは困難である。また、イントラネットやDeep Webといった、アクセスに何らかの制限があるWebリソースを収集することも困難である。

このような現状に対し、我々が提案した組織内Webアーカイブのモデルとそのシステム(Institutional Web Archive System : IWAS)は、運用者と発信者が連携することでWebページに運用組織の必要に応じて定義された独自のメタデータを付与する。そして、そのメタデータを元に収集タイミングと提供方法の条件を指定し、網羅性と利便性を高め、グローバルなWebアーカイブでは対象としていない組織内のWebアーカイブを構築する[1]。そのため、アーカイブしたWebページに付与するメタデータはIWAS毎に異なっており、データの相互利用は考慮していなかった。

本論文では、このようなメタデータの記述規則がIWASごとに異なる組織内Webアーカイブのデータを協調、共有することで、運用者の改組が原因で起こるIWASの機能不全を解決する方法を提案する。

2 組織を指向したWebアーカイブにおける改組への適応

2.1 組織内Webアーカイブとその運用組織の改組

一般に、Webページは同一のURLでも発信者の更新作業によって日々内容が更新される。Webアーカイブはそのような Webページを対象としたアーカイブであり、ガイドラインや実装事例がいくつか示されている。

ガイドラインとしては[2]がある。これはデジタル情報の長期保存システム構築のガイドラインであり,国際標準規格ともなっている。文献[3][4]は、発信者との連携は少なく、発信者が指定できるのは収集の拒否と、アーカイブされたデータの削除要請となっている。[5][6]は発信者に更新間隔や提供方法を尋ねる手法をとっているが、その指定はサイトの一括収集についてであり、ページ毎の指定はできない。他にも、Webページの更新頻度の推定方法も研究され、ある程度モデル化されている[7]。同様のモデルを適用したクローラの開発も行われているが、Webサイトが対象であり、Webページ毎の収集は行われていない[8][9][10]。また、文献[11]では、同様のモデルを用いたWebページ毎のクローリングのスケジューリング手法が検討されている。一方、文献[12]のように発信者がWebページを運用者に提出し、運用者が手作業でWebアーカイブ構築する事例もあるが、Webページの継続的な収集等は行っていない。

我々の提案した組織内 Web アーカイブ[1]では、発信者と運用者が連携して収集・蓄積・提供を継続して行う。そのために収集機能と利用機能があり、それらを合わせて管理機能と呼ぶ。管理機能は発信者の決めたメタデータと条件に従って動作する。その条件を記述したものをポリシー記述と呼び、このポリシー記述には収集ポリシーと利用ポリシー記述がある。

ポリシー記述は IWAS で Fig2.1 のように与えられる。収集時、管理機能は収集ポリシー記述に従って Web リソースの特定と収集の可否判断を行う。その後 XML で記述したメタデータの付与が行われ、収集日時の異なる同一 URL のリソースにまとめられて蓄積される。これを収集機能と呼ぶ。利用時は、蓄積されたデータのメタデータと閲覧者のリクエストを利用ポリシー記述に照らし合わせ、利用の可否を判定する。これを利用機能と呼ぶ。

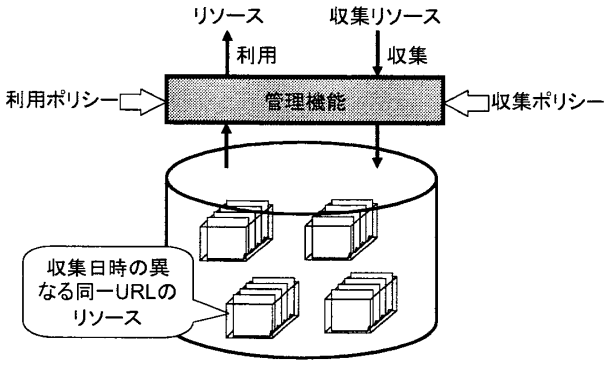


Fig2.1 IWAS 概要

ポリシー記述は各 IWAS で定義されたリソースのメタデータ記述規則に応じて決定されている。例えば、閲覧者の区別について、会社組織の場合は「部署名」や「社員ID」を使い、大学組織の場合は「学生名」と「職員名」と「教師名」を使うことなる。このようにリソースのメタデータは運用組織によって異なるため、改組によってアーカイブの構成とポリシーが変化すると、改組以前のポリシー記述と矛盾を生じ、機能不全が生じる。そこで、本論文では改組がおきた場合でも、運用が行える新たな IWAS を提案する。なお、本研究で想定した改組は以下の三種類である。

1. 分割:一つの組織が複数組織になること
2. 合併:複数組織が1つの組織になること
3. 消滅:組織自体がなくなってしまうこと

本論文では、以上の三種類の改組について IWAS にどのような障害がでるかを考察し、それを解決する組織内 Web アーカイブの新しい構成方法について提案する。

2.2 IWAS におけるアーカイブしたデータの構成

IWAS では、Web ページを Fig2.2 の構造を用いて蓄積している。これを Web Archived Object(WAO) と呼び、以下の要素をもっている。

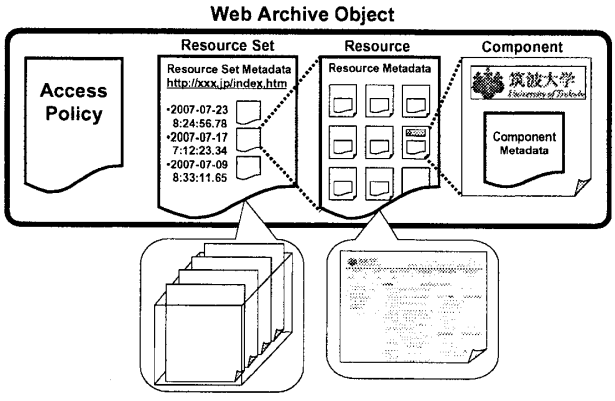


Fig2.2 Web Archive Object Model

Component:URL と日時で識別される Web ページを構成する最小単位。作成者などのメタデータが記述される。

Resource:収集した時点の Web ページ。収

集した時点の Component がメタデータに記述されている。

Resource Set:同一 URL で示される異なる Resource をまとめたもの。収集日時を基礎にしてソートする。

Access Policy:利用ポリシー記述。

利用者は、Fig2.3 のように WAO にリクエストして、WAO 内の Resource を利用する。リクエストには、URL と日時と利用目的、利用者情報が含まれる。そして、利用の条件を記述した Access Policy(AP)、日時と URL で特定した WAO 内のメタデータ、閲覧者情報に基づいて、閲覧といった利用目的について WAO が判定を行う。

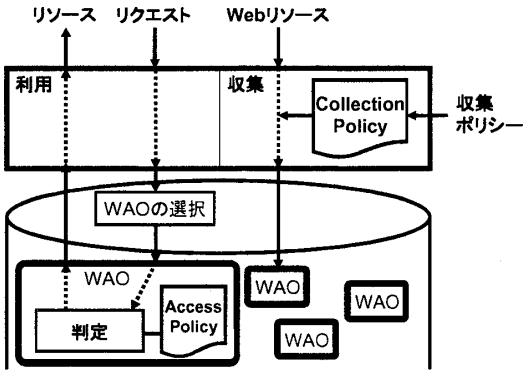


Fig2.3　リクエストと Web リソースの流れ

この判定に利用するメタデータは発信者が付与する。その記述には URL と収集日時、構造といった IWAS 共通の記述の他、運用者が個々の IWAS 用に定めたものが使われる。このメタデータが付与されるデータは大量にあるため、Web ページの収集条件とメタデータ付与のルールを発信者が記述する。これに基づいてシステムがメタデータを付与することで、収集した Web ページから WAO を生成する。このとき、収集条件である URL を、正規表現を使ってまとめて記述することで、発信者の記述する量を減らしている。これは、それぞれの Web ページが内容によってまとめて管理されており、そのまとまり毎に URL を一定のパターンで示すことができるからである。この記述が Collection Policy(CP)であり、これにより、発信者はシステムに特定人物の作成した画像は収集しない等の制限付き収集や、写真にそれぞれの撮影者名を付与する等の指定を行うことができる。

2.3 IWAS の協調とデータの統合

WAO には、生成時に用いた IWAS で定義されたメタデータが付与されているため、改組が起きた場合、改組前に生成された WAO はそのままでは使用できない。そこで、本論文では、改組前と同じように WAO の運用を続けるための IWAS の新しいモデルを提案する。以下に、2.1 で述べた三種類の改組に対応して IWAS がとる、WAO を引き続き使用するための対処手法を挙げる。

1. 分割:1つの IWAS で運用されていた全ての WAO を、改組に合わせて分割し、メタデータと AP および CP を変更することで、新しい組織毎の IWAS で利用できるようにする。
2. 合併:二つの IWAS で利用していた WAO を今まで通りに利用できるようにメタデータと AP および CP を変更し、一つの IWAS で利用する。
3. 消滅:消滅組織のアーカイブを受け継ぐための他組織の IWAS を用意し、WAO がその IWAS で利用できるようにメタデータと AP を変更する。

まず、どの場合でもメタデータと AP、CP を変更することが必要であるため、その変更方法を考える。1.では分割によってできた新しい IWAS に合わせて、メタデータのエレメント名と値を変更する。そして、AP および

CP で条件に使われているエレメント名とその値を新しいメタデータに合わせて変更し、それぞれに追加する。2.では合併によってできた新しいIWASに合わせて、1.と同様に変更する、3.では受け側の IWAS に合わせて 1.と同様に変更する。つまり、どの場合でも引き続き使用するWAOのメタデータとAP、CPを変更する点は同じである。

さらに、2.および3.では、受け側の運用者が望むデータを選択する必要がある。これは、新しくできる IWAS および受け側のIWASに不必要なデータが入ることを防ぐためである。このデータの選択機能は、1.についても、分割の結果必要となるデータを、受け側の運用者が選択すると考えることで利用することができる。つまり、(1)WAO を新しいIWASで運用できるように変更する機能、(2)データを受け取るIWASの示した条件に従ってデータを選択する機能、さらに(3)選択したことで一部の WAO に発生するリンク構造の矛盾を解決する機能、という機能を用意することで、各改組後でもWAOを利用することが可能となる。

以降、改組前の組織を送信側、改組後にデータを受け取る新しい IWAS を受信側とし、この2者間でWAOを利用できるように変更し、運用することを統合と呼ぶ。

まず、(1)を満たすために、メタデータとAP、CP の変更には、既存のメタデータのマッチングの手法を拡張して用いる[13][14]。そして、(2)を満たすために、大量の WAO から不要なデータを削除、編集するための条件の記述方法を決定する。そして、(3)で削除、編集したことにより生じたリンク等の構造の矛盾を WAO のメタデータに記述する。記述しておくことで、例えばリンク切れがあった場合、そのリンク切れが収集した時点から存在するものなのか、なんらかの都合で収集後に発生したものなのかを、利用者に伝えることができる。

我々は、以上の機能を備えた、統合機能の付いたIWASのモデルを提案する。

3 協調的アーカイブシステムにおけるデータ統合方式

3.1 IWASの協調とメタデータの統合方式

実際に統合されるWAOは膨大な量にのぼるため、その統合を手作業で行うことは難しい。そこで、WAO のメタデータと AP、CP の共通部分はそのまま利用し、各 IWAS 独自の部分にはフィルタリング等の処理を手続化して適用し、統合の効率化を行う。実際には、送信側と受信側のメタデータをエレメント毎に比べ、以下の場合にどのような処理や変換を行うかを記述する。

1. 同じエレメント、同じ記述規則:送信側のエレメントをそのまま利用する
2. 受信側に存在しないエレメント:送信側のエレメントをメタデータに併記する
3. 送信側に存在しないエレメント:メタデータに、変換した送信側エレメントや新しいエレメントを追加する

送信側 WAO は、統合後には受信側 AP にも従って利用されるため、メタデータにそのエレメントを追加しておく必要がある。また、送信側のメタデータも残しておくことで、送信側IWASで利用されていたAPを使用できるようにしている。

統合では、これらの処理を記述しておくことにより、メタデータと AP へのエレメントの追加、CP の変換を行う。

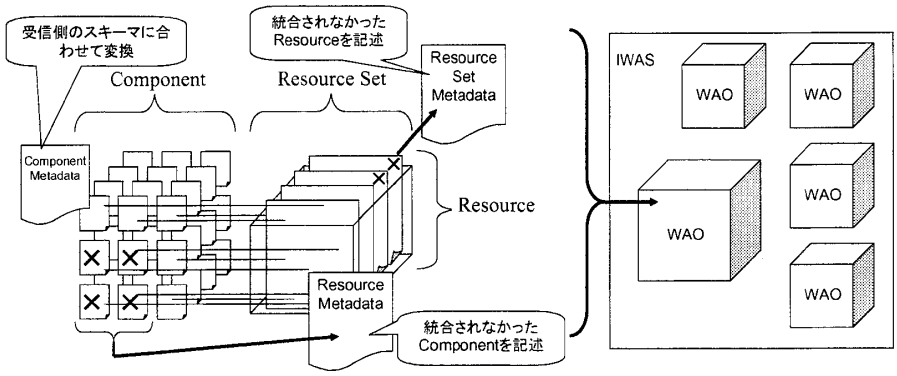


Fig3.1　Merge Policy を使った WAO の変換概要

3.2 WAO の分割とリンク構造の管理

IWAS は WAO を作成する際に、運用者の定めた条件によって収集 Web ページを選択している。そのため、統合時に受信側が望まないデータが送信側 WAO 内に存在した場合、そのデータを取り除く必要がある。そしてデータを WAO から分割し取り除いた際に発生するリンク構造の矛盾を解決する必要がある。以下では、取り除いた後どのように処理するか述べる。

前述したように、IWAS の統合時には受信側が WAO 内の要素を統合するものとしないものに分割する。そして、統合されなかったデータは、Fig3.1 のように、システムがメタデータにその旨を記録する。なお、WAO の最も大きい単位である Resource Set 自体が統合されなかった場合は、含んでいる全 Resource が統合されなかったとみなし、Resource Set のメタデータだけを残す。これは、WAO にどのような Web ページが収集されていたのか記録しておくためである。

3.3 Merge Policy を利用したメタデータの統合とアーカイブデータの分割

本論文では、統合対象の各 WAO に対して行う分割の条件と、メタデータの変更手続きを、受信側が Merge Policy(MP)として記述する。MP には AP、CP と同様に複数の WAO をまとめて表すために URL のパターンを記述する。そして、統合時は、システムが MP に従ってデータの分割を行い、次にメタデータと AP へのエレメントの追加と CP の修正を行う。

まず、分割に使う条件は URL と収集日時、その他の記述的な条件からなる。統合されるデータに関する情報は前もって統合元 IWAS から知ることができるため、システムは条件が記述されていれば分割を行える。実際には MP に記述した条件に合致したもののみを受信側 WAO に選択して加え、統合されなかったデータは 3.1 節の方法でメタデータに記録する。

条件に合致し、統合されることになったデータには Fig3.2 のような流れでメタデータの変換と付与を行う。変換する方法は、作りなおす場合と、送信側 IWAS が付与した既存のメタデータを利用する場合との 2 種類がある。作りなおす場合は、URL を条件として新たなエレメントをシステムが追加する。既存のメタデータを利用する場合は、URL 等の条件と、受信側のどのエレメントが送信側のどのエレメントと同じであるかを受信側が指定することで行う。この指定は 1 対 1 だけではなく、受信側のエレメントに複数の送信側エレメントを組み合わせることができる。シ

ステムはこのように、MP 内の条件を記述順に判定し、各 WAO のメタデータと AP へのエレメントの追加、CP の変換を行う。また、機械的に処理できない場合は受信側 IWAS の運用者に判断を仰ぎ対応する。

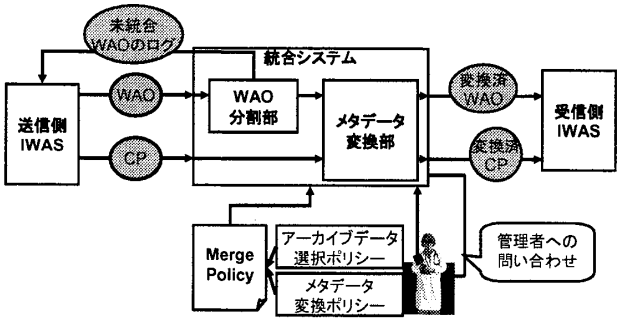
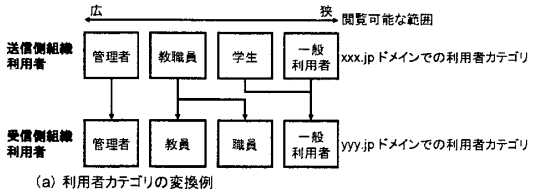


Fig3.2　統合システム概要

実際に組織が統合し、WAO に対する利用者が変更になった例が Fig3.3 である。



(a) 利用者カテゴリの変換例

Merge Policy例

送信側APに記述された条件記述	user = 管理者	user = 教職員 and url = http://xxx.jp/~UID	user = 教職員	user = 学生 and url = http:// xxx.jp /~UID	user = 一般利用者 or user = NONE
変更後の条件記述	group = 管理者	group = 教員	group = 職員	group = 一般利用者 and id = UID	group = 一般利用者

(b) Access Policy内の条件の変換例

Fig3.3　Merge Policy を使った変換例

これは、二つの大学が統合する場合を想定して書かれた表であり、(a)は利用者の閲覧可能な範囲とその変換例、(b)はその範囲に合わせて条件を変換する手順を示している。この場合、利用者の名称の変更に合わせて各 WAO のメタデータを変換しなければならない。なお、(b)において「エレメント名＝値」という記述は「エレメントの値がその値と同じならば」という意味である。システムは(b)の上段の条件記述が AP にあった場合、下段の条件記述に変換する。

この場合の変換の内容は、送信側では教職員が教員と職員になり、学生が一般利用者と同じ扱いとすることである。しかし、送信側 IWAS では、一般利用者は学生の管理しているページについてのアーカイブを閲覧することができなかったため、その制限を統合後も維持することが望ましい。そこで、各学生が管理していたページは、統合後の条件に学生以上の権限を持つ ID を加えることで、閲覧できる範囲を制限する。これで、一般利用者は学生個人のページに関して閲覧することができなくなる。また、教職員は受信側 IWAS では教員と職員にわけられており、職員は教員の個人ページのアーカイブを閲覧できない規則になっている。その場合、送信側 WAO のうち閲覧の制限に教職員を持ち、なおかつその URL が教員の個人ページ内になっているものを、教員個人のページと判断する。なお、Fig3.3 の(b)では、左から順番に適用される。この記述方法を用いることで、複数の条件で変換を行う場合にも対応することができる。この例の場合は、エレメント名の変換を行い、それに合わせてメタデータと AP へのエレメントの追加および、CP の変換が行われ、統合が完了する。また、どの条件にも当てはまらなかった場合は、その場合の変換方法を MP に記述しておく。

3.4 ポリシーの記述形式

ポリシー記述は XML 形式で記述する。Fig3.4 のように、ポリシー記述のルートは rule 要素である。rule 要素は、その子要素として一つ以上の condition 要素と一つの default 要素をもつ。condition 要素は effect 属性をもち、子要素としてリクエストを出した

主体を表すsubject要素、対象のURLを表すtarget要素、リクエスト内容を表すaction要素、その他の条件を表すenvironment要素をそれぞれ一つ以上もつ。これらの子要素には具体的な条件が記述されている。システムはリクエストを受け取った場合に、そのリクエストが子要素内の条件と合致するか判定する。そして、合致した場合はeffect属性に従って動作する。effect 属性の値にはaccept、deny、withhold があり、accept の場合はリクエストを受け入れる、denyの場合は拒否する、withholdの場合は管理者に問い合わせるという動作を行う。

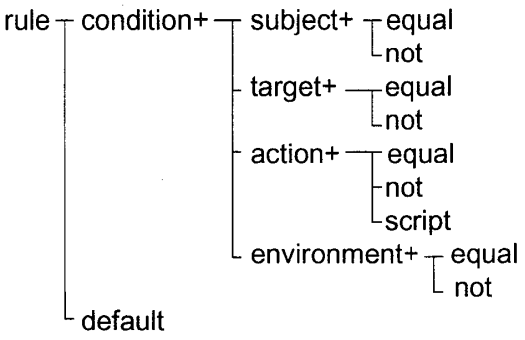


Fig3.4 ポリシーの構成要素

```
<rule>
  <condition effect="accept">
    ・・・
    <target>
      <equal element=" url">^xxx¥.jp.*</equal>
      ・・・
</rule>
```

Fig3.5 基本的な記述例

この4種類の要素には条件を記述するための子要素としてequal要素とnot要素があり、それぞれがelement属性をもつ。element属性には WAO のメタデータに使われている要素名が入る。実際には、Fig3.5 のように正規表現を使って記述することでxxx.jpから始まるurl要素をもったリクエストを受け取った場合、condition要素のeffect属性に従って、そのリクエストを受けいれ、action要素を実行する、という動作をおこなう。

subject 要素にはリクエストを出した利用者についての条件が、各IWASで規定したメタデータの記述規則に従ってXML で記述される。target要素にはリクエストの対象となるWebページのURLが同様に記述される。environment 要素には、WAO のメタデータ内の条件が一つ以上記述される。action要素にはその WAO が対応できる動作が一つ以上記述される。現在は、view、delete、collect、merge の4種類の動作がある。そのうち、collect と merge の場合は、動作の記述の後に、収集リソースに付与するメタデータの変換、生成方法を記述しておく。変換、生成方は子要素である script 要素にXQueryで記述する。

```
<rule>
  ・・・
  <action>
    <not element="behavior">^(view|delete)</not>
    <script>
      for $vn in document($INPUT)//viewer
      let $in := document($OUTPUT)//data
      where count$vn > 0
      return
      <viewerlist>
        {$vn/text()}
      </viewerlist>
      ・・・
</rule>
```

Fig3.6 メタデータ変換のための記述例

Fig3.6のようにaction要素を記述することで、behavior要素に閲覧と削除以外が記述されたリクエストが出された場合、WAO内のメタデータのすべてのviewer要素の記述を、data要素の子要素であるviewerlist要素に列挙して記述する、という判定を行うことができる。これにより、メタデータにviewer要素と同じ内容のviewerlist要素をメタデータに追加することができる。なお、$INPUTと$OUTPUTはWAO内の変換元メタデータと変換後メタデータを示す統合機能付きIWASで定義している変数である。

なお、default要素はeffect属性をもち、子要素はない。Default要素はWAOに出されたリクエストがどの条件にも当てはまらなかった場合の判定を行う。

4 統合システムの実装

我々は、この統合をLinux上で実装し、テスト用のWebサイトを作って確認を行った。実装ではFig3.2のWAO分割部および、メタデータ変換部にJava2SE 5.0を利用した。

また、WAOはPostgreSQL7.4.7を使用しComponent、Resource、Resource Set用のテーブルを用意し、Webページを構成しているデータおよびIWAS間で共通の記述規則のメタデータを格納している。IWASごとに異なるメタデータ、および必要なAPはXMLで記述し、テーブルに格納している。これにより、IWASごとに要素の数が異なるメタデータやAPであっても、それぞれ一つのテーブルに格納できる。

ポリシー記述であるCPとAPはテキストファイルで収集システムに与えられる。CPは収集時に参照され、APはWAOを作成、更新する際に必要な部分だけがWAOに記述されるようになっている。統合に利用されるポリシーであるMPも、テキストファイルで統合システムに与えられる。

MPは、その記述にXML、変換用の記述にはXQuery1.0を使用し、その記述に従ってWAO内のメタデータの変換を行う。なお、XQueryの変換エンジンにはSAXON SA-8.9. for Javaを利用している[15]。

5 まとめと今後

本研究では、改組が起きてもIWASをできるだけ安定して運用するために、IWAS同士の協調的な作業としてアーカイブの統合を提案した。組織の改組は三種類としたが、実際にはアーカイブの分割とメタデータスキーマの変換という手段で統合が可能になった。また、その統合方法も一定の処理方法が定義できた。本研究によって、IWASの運用組織の改組における問題を解決する方法を示した。これにより、IWASを運用する組織の変化にWebアーカイブを合わせることができるようになり、Webアーカイブの運用性を高めることができると考える。

だが、統合されずに削除されたWAOについては問題が残っている。まず、Resourceが削除されてしまった場合、そのページからしかリンクされていないResourceが参照できなくなる。それは、IWASとしては削除されたものとして扱うのか、何らかのアクセス方法を残し、アーカイブに残すべきなのか考える必要がある。また、統合時に削除されたWAOは、その旨が統合元の組織に伝えられ、統合先の組織にはメタデータに統合時に削除されたという記述が残るだけである。今後、削除されたアーカイブデータを削除するのか、なんらかの方法で他

の組織に再び統合をするのか、削除しないですむような方法があるかを考察し、その場合のリンク等の参照関係をどのように扱うかが今後の課題である。

本論文によって技術的には運用者の意向に基づいてWebアーカイブを統合できるようになった。しかし、Web アーカイブに含まれる知的財産権に代表される諸問題については、運用者がMPを正確に記述することで対応する必要がある。そのためには、運用組織は自ら責任をもってWebページを残そうと考え、その上でどのようなWebアーカイブを構築するか検討しておくことが必要となる。本論文で述べたIWASの統合機能で、その検討に基づく統合を行うための技術的な方法は示すことができたが、個々のWebアーカイブが統合後に本当に検討した通りのWebアーカイブになっているか、という検証は運用者が行うことになる。

また、AP や CP のように順次追加していけるポリシー記述と異なり、MP は統合を行うと決まった時点で全てを記述しなければならないポリシー記述である。現在は条件の組み合わせや、条件の記述に正規表現を用いることでその手間を減らしてはいるが、今後は実運用を通して実際にどの程度のコストがかかるか考える必要がある。

参考文献

1. Wasuke Hiiragi, Tetsuo Sakaguchi, Shigeo Sugimoto, Koichi Tabata: "A Policy-based System for Institutional Web Archiving", Z.Chen et al.(Eds):ICADL 2004,LNCS 3334, pp.144-154,2004.
2. Consultative Committee for Space Data Systems. Reference Model for an Open Archival Information System (OAIS). Blue Book, Issue 1 (CCSDS 650.0-B-1). 2002. <http://wwwclassic.ccsds.org/documents/pdf/CCSDS-650.0-B-1.pdf>. (accessed 2007-10-01)
3. Internet Archive. <http://www.archive.org/>. (accessed 2007-10-01)
4. The Kulturarw^3 Heritage Project. <http://kulturarw3.kb.se/>. The Royal Library (National Library of Sweden). (accessed 2007-10-01)
5. WARP. <http://warp.ndl.go.jp/>. National Diet Library Japan. (accessed 2007-10-01)
6. PANDORA Archive. <http://pandora.nla.gov.au/index.html>. PANDRA. (accessed 2007-10-01)
7. J. Cho and H. Garcia-Molina: "The evolution of the web and implications for an incremental crawler", Proc. of VLDB, pp. 200-209, 2000.
8. J. Cho and H. Garcia-Molina: "Estimating frequency of change", ACM TOIT, 3, 3, pp. 256-290, 2003.
9. J. Cho and A. Ntoulas: "Effective change detection using sampling.", Proc. of VLDB, pp. 514-525, 2002.
10. 熊谷, 山名: "リンク構造を利用したWeb ページの更新判別手法", DEWS2004 論文集, 2004.
11. 田村 孝之, 喜連川 優: "大規模Webアーカイブのための更新クローラの設計と実装", DEWS2007 論文集, 2007.

12. KB archive Nederland websites. <http://www.kb.nl/nieuws/2006/webarchivering.html>. (accessed 2007-10-01)
13. Atsuyuki Morishima, Toshiaki Okawara, Jun'ichi Tanaka, Ken'ichi Ishikawa: "SMART: A Tool for Semantic-Driven Creation of Complex XML Mappings", SIGMOD Conference 2005: 909-911, 2005.
14. XQuery 1.0: An XML Query Language. <http://www.w3.org/TR/xquery/>. W3C. (accessed 2007-10-01)
15. SAXONICA. <http://www.saxonica.com/>. SAXONICA. (accessed 2007-10-01)

(2007年10月23日受付)
(2008年2月3日採択)