

2025
Fall
49

デジタルアーカイブシステム間連携による 長期保存方式の研究

(課題番号 16500148)

平成 16 年度～平成 17 年度
科学研究費補助金 (基盤研究 (C))

研究成果報告書

平成 18 年 3 月

研究代表者 田畑孝一

筑波大学図書館情報メディア研究科教授

寄贈
田畑孝一氏

06003736

はしがき

本研究では、種々のコミュニティで構築されるアーカイブシステムは、その保存性をより高めるためには相互に連携するのがよいと考え、そのための方式を考究する。アーカイブシステムを構築する以上、当該コミュニティは組織をあげてそのシステムの長期保全に努めるはずであり、長期保存データは十分生き残りの保護を受ける。このようなアーカイブシステムが互いにコミュニティを超えて、特に重要なデータを相互に預け入れる体系を作成すれば、いっそうデータの保全が図られる。障害回復時には預けたデータの返却を求め、万一、自システムが破壊・消滅の場合には、預けたデータは他システムで生き残ることができる。

本研究では、上述の考えをデジタルアーカイブの国際標準 ISO の参照モデルである Open Archival Information System (OAIS) に適用して、考察を進める。OAIS は、任意に定められたコミュニティに対し、情報を保存する責任をもち、かつそれを利用可能にさせるアーカイブである。コミュニティは多種多様であり、それぞれが保存する情報やデータの性質は異なる。OAIS 参照モデルでは、それにもかかわらず、長期保存の観点から最低限必要な遵守すべき枠組みを定め、それ以外はコミュニティごと自由な保存形式を定めることができる。ここでは OAIS 準拠ではあるが、具体の保存形式が互いに異なるコミュニティのアーカイブシステム間の連携保存方式を考察し、下記の結果を得た。

OAIS 参照モデルにおける保存情報の基本単位である Archival Information Package (AIP) は CI (Content Information) 部と PDI (Preservation Description Information) 部からなる。源システムの AIP の内容、すなわち CI 部と PDI 部を一つの袋に封入し、その袋を目的システムの CI 部で預かる。これにより、源システムの AIP を目的システムのそれにいちいち書き換える必要がなく、保存対象のデータ形態が異なる OAIS 間においても預け入れ作業が簡単になる。

目的システムには、袋と一緒に、それを開封する方法を記した開封説明書を預けておく。源システムで預けた情報を紛失しても、目的システムから袋とその開封説明書が返却されるので、それを参照して復元できる。そして、いつか、たとえ源システム全体が消失しても、預けた情報は目的システムで生き残る。目的システムでは袋の開封説明書も一緒に預かっているので、それに則り袋を開封すれば、元の情報を復元できる。

マサチューセッツ工科大学とヒューレットパッカー社が共同開発したデジタルアーカイブシステム DSpace を OAIS とみなして、コレクション単位での預け入れのモデルをインプリメントし、このモデルを確かめた。ちなみに DSpace はその機能が OAIS 参照モデルのそれらによく対応したシステムである。

国際標準 OAIS ではその参照モデルで、アーカイブ相互間における利用面での連携モデルについては言及しているが、保存面での相互連携モデルには触れていないので、本研究の成果は本国際標準の発展に寄与するものと考えている。

研究組織

研究代表者 : 田畑孝一 (筑波大学図書館情報メディア研究科教授)
研究分担者 : 杉本重雄 (筑波大学図書館情報メディア研究科教授)
研究分担者 : 阪口哲男 (筑波大学図書館情報メディア研究科助教授)
研究分担者 : 森嶋厚行 (筑波大学図書館情報メディア研究科助教授)
研究分担者 : 永森光晴 (筑波大学図書館情報メディア研究科講師)
(研究協力者 : 岡田 武)

交付決定額 (配分額)

(金額単位:円)

	直接経費	間接経費	合計
平成 16 年度	2,200,000	0	2,200,000
平成 17 年度	1,500,000	0	1,500,000
総計	3,700,000	0	3,700,000

研究発表

(1) 学会誌等

1. Maria Luisa Calanag, Koichi Tabata, and Shigeo Sugimoto, Linking Preservation Metadata and Collection Management Policies, Collection Building, Vol.23, No.2, pp.56-63, 2004
2. Wasuke Hiiragi, Tetsuo Sakaguchi, Shigeo Sugimoto, and Koichi Tabata, A Policy-Based System for Institutional Web Archiving, Lecture Notes in Computer Science 3334, Springer-Verlag, pp.144-154, 2004
3. Koichi Tabata, Takeshi Okada, Mitsuharu Nagamori, Tetsuo Sakaguchi, and Shigeo Sugimoto, A Collaboration Model between Archival Systems to Enhance the Reliability of Preservation by an Enclose-and-Deposit Method, 5th International Web Archiving Workshop (IWAW05), <http://www.iwaw.net/05/papers/iwaw05-tabata.pdf>, 2005

(2) 口頭発表

1. Koichi Tabata, Takeshi Okada, Mitsuharu Nagamori, Tetsuo Sakaguchi, and Shigeo Sugimoto, A Collaboration Model between Archival Systems to Enhance the Reliability of Preservation by an Enclose-and-Deposit Method, 5th International Web Archiving Workshop (IWAW05), <http://www.iwaw.net/05/tabata.pdf>, Sep. 23, 2005

研究成果

目次

1. デジタルアーカイブシステム間連携による長期保存方式モデル4
田畑孝一

2. 5th International Web Archiving Workshop(IWAW05)での本研究発表
2.1 ワークショッププログラム15
<http://www.iwaw.net/05/>

2.2 A Collaboration Model between Archival Systems to Enhance
the Reliability of Preservation by an Enclose-and-Deposit Method19
Koichi Tabata, Takeshi Okada, Mitsuharu Nagamori,
Tetsuo Sakaguchi, and Shigeo Sugimoto
(本研究の発表スライド)
<http://www.iwaw.net/05/tabata.pdf>

3. A Collaboration Model between Archival Systems to Enhance
the Reliability of Preservation by an Enclose-and-Deposit Method30
Koichi Tabata, Takeshi Okada, Mitsuharu Nagamori,
Tetsuo Sakaguchi, and Shigeo Sugimoto
(5th International Web Archiving Workshop での発表論文)
<http://www.iwaw.net/05/papers/iwaw05-tabata.pdf>

4. デジタルアーカイブシステム間連携による長期保存方式に関する研究43
岡田 武
(特に、プロトタイプシステムの詳細記述)

デジタルアーカイブシステム間連携による長期保存方式モデル

田畑孝一

<抄録> 国際標準のデジタルアーカイブシステム Open Archival Information System(OAIS)は情報の長期保存のためのものであるが、それらが互いに自身の保存情報を相手に預け入れることができれば、より一層長期の保存が期待できる。ここではそのためのモデルを提案する。OAIS 参照モデルにおける保存情報の基本単位である Archival Information Package(AIP)は CI(Content Information)部と PDI(Preservation Description Information)部からなる。源システムの AIP の内容、すなわち CI 部と PDI 部を一つの袋に封入し、その袋を目的システムの CI 部で預かる。これにより、源システムの AIP を目的システムのそれにいちいち書き換える必要がなく、保存対象のデータ形態が異なる OAIS 間においても預け入れ作業が簡単になる。目的システムには、袋と一緒に、それを開封する方法を記した開封説明書を預けておく。いつか、たとえ源システムが消失しても、預けた情報は目的システムで生き残る。目的システムでは袋の開封説明書も一緒に預かっているので、それに則り袋を開封すれば、元の情報を復元できる。DSpace を OAIS とみなして、コレクション単位での預け入れのモデルをインプリメントし、このモデルを確かめた。

1. 資料保存館の連携についての基本的考察

米国の資料保存館で、日本にはもはや存在しないという日本の貴重な古い資料が“発見された”というような話をよく耳にする。このことは情報の長期保存のあり方についての有効な示唆をしている。その状況をもう少し詳しく考察すると、——日本のある資料保存館で、過去においてその資料の未来における貴重さに気づかず保存を心がけなかったか、あるいは保存に心がけたにも関わらずそれを消失してしまった。一方、米国の資料保存館では、その資料の内容はよく分からないが、何か貴重なものらしいとのことで、その当時の係りの人が、日本語は読めなかったが、資料を袋に入れ、その表面にその資料の由来を英語で簡略に記載して書庫のどこかにそれを保存した。それが、今になって発見、その貴重さが再認識されて、英語による内容説明と共にその館のものとして正式に登録された。その資料は日本にとっても大切に、その複製が改めて日本のどこかの資料保存館でも保存されることになった。——

このことを参考にして、資料保存館の相互連携による長期保存のモデルを考察する。

(1) ある資料保存館（上の例では日本の資料保存館）で、ある資料の長期保存を計画する。

しかしながら、自館のみで資料を保存していても、それがいつか消失するかもしれないので、特に大切なものは、他館（遠くは、上の例で、米国の資料保存館）にそのコピーを預かってもらうことにする。預け先の館も資料保存を目的としているところなので、預けた資料の生き残りについての確度は高い。

- (2) 後になって自館でその資料を消失したとき、預け先の館からそれを戻してもらい、復元する。
- (3) けれども、さらに後になると、預け元の館において、資料を消失したことはおろか、何をどこに預けたかも忘れ去られる。そして、いつか、預け先の館でその資料が発見され、どこから預かったが判明し、預け元の館に返却され、そこで復元が行われる。預け元の館自体がもはや存在しない場合は、預け先館でそれを自館本来の資料の一部として保存・閲覧サービスすることになり、あるいは（日本の）別の相当する館に返却する。

このことについて、預かり先の館での状況をさらに考察する。

- (a) 預かり物については保管が主目的で、また預かり元の館とはそのサービス対象コミュニティが違うのでそれについての閲覧サービスをしない。
- (b) メタデータの体系は、一般に預け元、預け先で異なる。国が違えば、その記述言語も異なる。預かった資料を自館本来の資料と同等の扱いをするには、各資料について、メタデータの自館固有の体系への書き直しや記述言語翻訳を行わなければならない。しかし、預かり物で、一般閲覧サービスをする訳ではないので、そのような作業はしたくない。
- (c) 資料の内容や預け元のメタデータにとらわれず、気軽に預かるには、預かるべき資料とそのメタデータ（上の例では、日本語）を袋にいった形でもらい、その袋に「いつ、どこの館から預かった預かり物 P」（P は単なる識別番号）ということ、自館の記述言語（英語）とメタデータ体系に従って書くことにする。例えば、[タイトル：預かり物 P]、[作成者：預け元館]、[日付：預かり日]など。係りの人は、預け元のメタデータ体系や言語（日本語）を理解する必要はない。その際、一緒にもらった袋の開け方を書いた文書（日本語）には、自館のメタデータ（英語）を作成し、袋と共に保管して置く。
- (d) 返却を要請された際には、自館のメタデータに頼って預かり物を探し出し、資料を袋に入れたまま、袋の開け方の文書も一緒に付けて、返却する。預け元の館では、その文書を参照して袋を開封し、袋の中の資料とそのメタデータを取り出し、自館に格納する。
- (e) ずっと後になって、預け元の館が存在しなくなった場合には、自館で（日本語が分かる人が）袋を開示・開封して、その資料とメタデータを理解し、必要に応じて自館の本来の資料として本格的に登録し、閲覧サービスに供する。その際、メタデータ（日本語）の自館固有の体系への書き直しやその記述言語の（英語への）翻訳が行われる。資料が作成されてから十分な歳月が経っているのでもはや著作権の問題はなく、閲覧サービスは公衆対象に行われる。

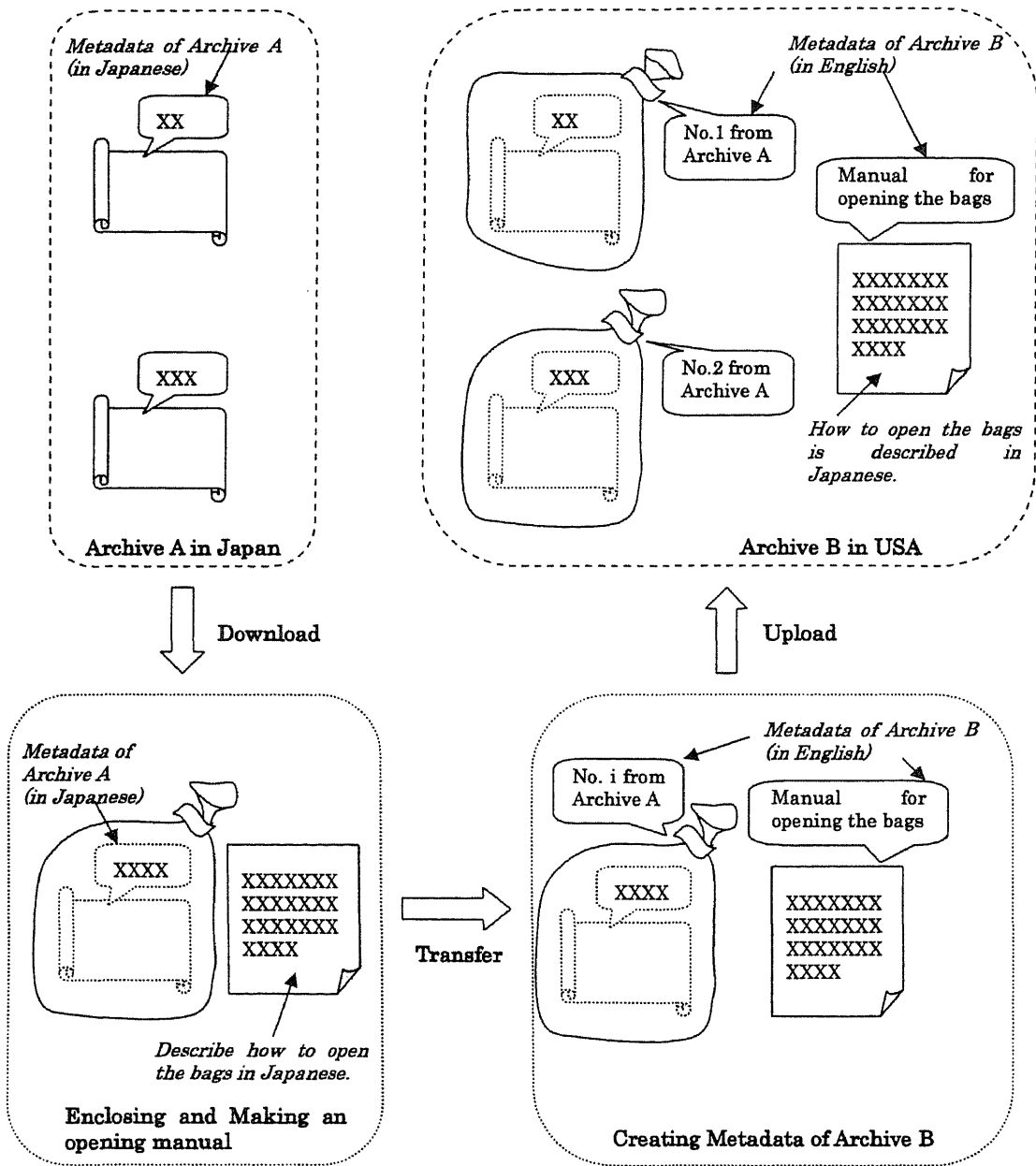


Fig.1 The basic concept of Depositing between Archives

このモデルを図1で示す。日本に存在する預け元の資料保存館Aの側で、A館から預ける資料とそのメタデータ（日本語）を取り出してその両者を一つの袋に入れ、さらにその袋の開封説明書（日本語）を作成し、それらを米国の資料保存館Bの側にする。B側では、袋とその開封説明書にB館固有のメタデータ（英語）を付けて、B館で保存する。袋に付けるメタデータは、A館からいつ預かった何番目のもの（識別子）の程度の簡単なものでよい。開封説明書のメタデータは、それが預かり物の開け方が記述してある文書であることが判るようにしておく。それを参照にして、後日、返却、開示・開封の作業が行われる。

このモデルで保存資料館を日本と米国のそれで、記述言語もそれぞれ日本語と英語としているが、これは、違いをあえて強調するためである。同じ国における異なる資料保存館の間での連携が、むしろ普通である。資料保存館は、OAISでも言及しているようにコミュニティ対応に存在するが、それらのメタデータ体系は扱うデータの性質によって互いに異なる。例えば、大学図書館と宇宙観測センター資料館のように互いにデータの性質は異なるが、それらの館で相互連携して資料の保存を考えてもよいのである。また、この例で開封説明書を日本語で書くとしているが、これは預け元でそれを書くということを強調しているだけである。預け元で袋に入れるのであるから、預け元でその開け方を書くのは当然である。

このモデルの特徴は、

- (i) 預け元、預け先の両館に対して、資料館本来のサービス以外の特別な機能を要求しない。本来準備されているダウンロード、アップロード機能しか使わない。
- (ii) 預け作業、返却作業などの作業環境を保持しなくてよい。必要なときのみその作業環境を整えればよい。預け作業のことが忘れ去られたずっと後になっても、保持されている開封説明書を参照して、返却作業、開示作業の環境を整えればよい。最終的に頼るのは、預け先の袋とその開封説明書のみである。それらが預け先で消失しない限り、保存が継続され復元が可能である。
- (iii) 連携館のメタデータ体系（および記述言語）が異なってもよく、かつそれらの体系を相互に変換しなくてよい。つまり、保存対象のデータ形態が異なるOAIS間においても預け入れ作業が簡単である。

2. OAIS間の連携保存

国際標準ISOの参照モデルであるOpen Archival Information System (OAIS)は、任意に定められたコミュニティに対し、情報を保存する責任をもち、かつそれを利用可能にさせるアーカイブである[1]。

コミュニティは多種多様であり、それぞれが保存する情報やデータの性質は異なる。OAIS参照モデルでは、それにもかかわらず、長期保存の観点から最低限必要な遵守すべき枠組みを定め、それ以外はコミュニティごと自由な保存形式を定めることができるように

なっている。一方、OAIS 参照モデルでは、こうしてできたコミュニティ対応のアーカイブシステムが相互連携して利用者にサービスを行うための、利用連携モデルを提示している。しかしながら、それはこれらが相互連携して、より長期の保存を企てるモデルを提示していない。

この論文では、OAIS 参照モデルに準拠したアーカイブシステムが互いに連携してより長期の保存を企てるモデルを 1. で論じた考察に基づき、提案する。アーカイブシステムはそれが作られる以上、コミュニティが組織をあげてその長期保全に努めるはずであり、長期保存データは十分生き残りの保護を受ける。このようなアーカイブシステムが互いにコミュニティを超えて特に貴重なデータを相互に預け入れることができれば、いっそうデータの保全が図られる。ところが、データ体系がコミュニティ対応で異なるので、他のアーカイブから預かるデータを自アーカイブの体系に書き換えるとなると、その作業が個別的かつ膨大になる。本論文ではそのような作業を行うことなく、異なるアーカイブシステム間で保管が可能となる方式を提案している。

OAIS 参照モデルでは、保存すべき情報を Archival Information Package (AIP) で表す(図 2)。

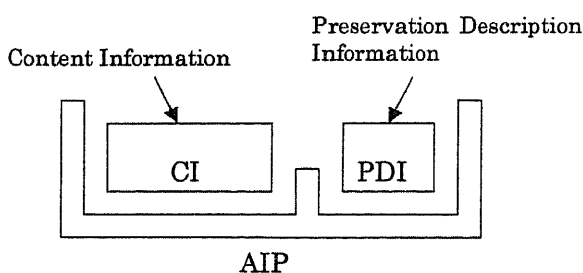


Fig.2 Archival Information Package (AIP)

AIP は、Content Information (CI : コンテンツ情報) および Preservation Description Information (PDI : 保存記述情報) と呼ばれる二つのタイプの情報を入れる概念的なコンテナである。CI はもともと保存すべき対象となっている情報である。CI は、Content Data Object (CDO) とそのための Representation Information (RI) からなっているが、この RI は、当該コミュニティがその CDO を理解できるようにするのに必要なものである[1]。

1. で論じた資料とメタデータは、AIP のそれぞれ CI と PDI に対応する。資料保存館 A 側で資料とメタデータを一つの袋に入れ、資料保存館 B 側でその袋に B 館のメタデータを付けて保管することは、次のことに対応する(図 3)。すなわち、源システム (OAIS A) の側でその CI と PDI を一つの袋に入れる。目的システム (OAIS B) 側ではその袋を自身の

AIP の CI に入れ、PDI にはその目的側のメタデータを入れる。目的システム側の PDI は、源システム側のそれを書き換えたものではなく、たとえば「OAIS A から、XX 年 YY 月 ZZ 日に、預かった物 P」（P は識別番号）という記述である。なお、目的システム側で袋の入る位置は、正確には後に述べるように、CI 中の CDO 部である。

資料保存館 A で作成された開封説明書については、次のことに対応する。すなわち、源システム側で用意した開封説明書を、目的システムの AIP の CI に入れ、PDI にはそれが開封情報であることが目的システム自身で識別できるように作成されたメタデータを入れる。上と同様、正確には説明書の入る位置は CI 中の CDO 部である。

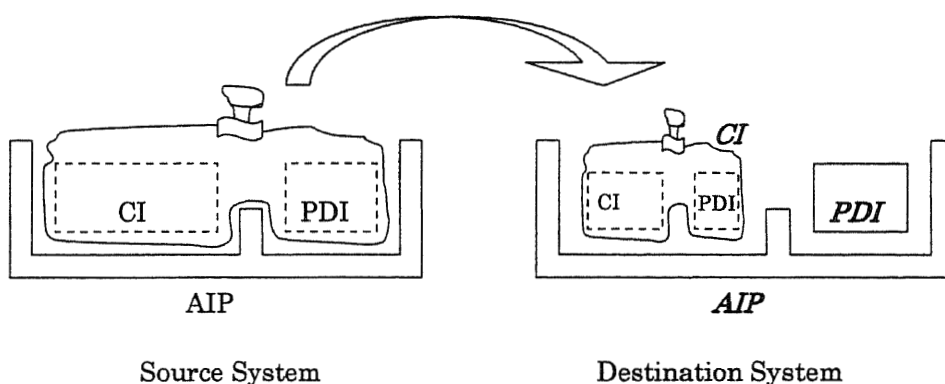


Fig.3 The concept of depositing between OAIS's

3.連携保存モデル

モデルをより詳細に述べる前に、いくつか用語を説明する。

(a)封入とは、源システムの AIP の内容、すなわちその構成要素である CI と PDI の内容を一つの XML 文書で表すこと。この文書を封入文書と呼ぶ。XML 文書にするに先立って、ビット列の部分は base64Binary によって文字列に変換する。なお、文書で使用している文字セットの符号化を、XML 宣言の encoding パラメタで指定する。

(b)開封とは、封入文書から、封入された AIP の内容、すなわちその構成要素 CI と PDI の内容を取り出すこと。base64Binary 変換されている文字列はその逆変換によってビット列に戻す。

(c)開封説明書は、封入文書がどのように作成されたかを説明することによって、封入文書の開封を可能にする文書。

(d)開示とは、目的システムにおいて、預かっている封入文書の中身を覗き見ること。封入文書は XML 文書であるので、XML 宣言の encoding パラメタで指定された文字セット環境を用意すればその中身を覗き見ることが可能である。必要に応じて、封入文書を開封する。

封入文書の作成方法をより詳しく説明する。源システムの AIP の内容、すなわちその構成要素である CI と PDI の内容を式(1)のように一つの XML 文書で表すことによって、封入文書を作成する。なお、CDO 部において、base64Binary 変換により文字列にした部分は<base64Binary>と</base64Binary >で囲む。

```
<?xml encoding = '...'?>
<API><CI><CDO>xxx</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></API>      式(1)
```

式(1)はその典型を示しているのですが、その構成は単純に見える。しかしながら実際にはもっと複雑な構成となる。CDO 部には複数の Data Object でかつ異なるデータ形式のものが含まれることが多い。たとえば、ある（一つの）HTML 文書はそれによって参照されるいくつかの JPEG データと共に構成されている。一方、CI において、CDO 部と RI 部が明確に分かれているとは限らない。たとえば RI に相当する表現情報をファイル名 abcde, prqst の拡張子として、abcde.html や prqst.jpg などと表すことも多い。なお、式(1)の生成は、テキストエディタ等を用いて手作業でも出来るが、後に述べるようにコレクション単位の場合は、プログラムを作成して一括処理する。

式(1)はアーカイブシステムによって異なるので、その構成を説明する文書が必要となり、それは源システム側で作られる。後日、この文書により CDO 部の構成が理解でき、封入文書の開封ができる。この文書を開封説明書と呼ぶ。この説明書は基本的には文字テキストである。base64Binary 変換方法の説明も行う。

<預け入れ作業> 目的システムの API の CI において、CDO 部に源システムで作成された封入文書を入れ、RI 部には「XML 文書」と記述する。RI 部の記述により、CDO 部にある情報が XML 文書であることが分かる。目的システムの AIP の PDI には、「OASIS A から、XX 年 YY 月 ZZ 日に、預かった物 P」と記述する（P は識別子）。

開封説明書は目的システム側の AIP の CDO 部に入れ、RI 部にはその表現形式を記述する。1.の例のように日本語で書いた場合は「日本語テキスト」と記述する。PDI には「OASIS A からの預かり物の開封マニュアル」と記述する。なお、開封・開示のためのプログラムを予め用意しておく場合は説明書と一緒に CDO 部に入れ、RI 部にそのプログラムの実行環境の情報も入れておく。しかしながら、それらプログラムの実行環境が、源側・目的側いずれにおいてもそれが返却・開示の時点で利用可能な状況であるかどうかどうかわからない。実行環境がもはや存在しないときは、開封説明書を参照して、改めてプログラムを作成する。

<返却作業> 目的システムから返却された封入文書を源システムで開封し、復元された AIP を源システムで再登録する。開封説明書を参照して、開封する。

<開示作業> 目的システムで封入文書を開示し、必要に応じて開封し、それを目的システムで登録する。

4. コレクション単位のモデル

連携保存するに際し、保有している保存情報を一件単位で他システムに預けることもできるが、現実的にはコレクション単位で預けることになる。OAIS 参照モデルでは、コレクションを次のように表している。AIP には、AIC (Archival Information Collection) とそのメンバーに成りうる AIU (Archival Information Unit) の 2 種類がある。AIU の CI には一件の保存情報を入れる。AIC の CI にはそのメンバーになるべき AIU を同定できる情報を入れ、それによってコレクションが形成される。

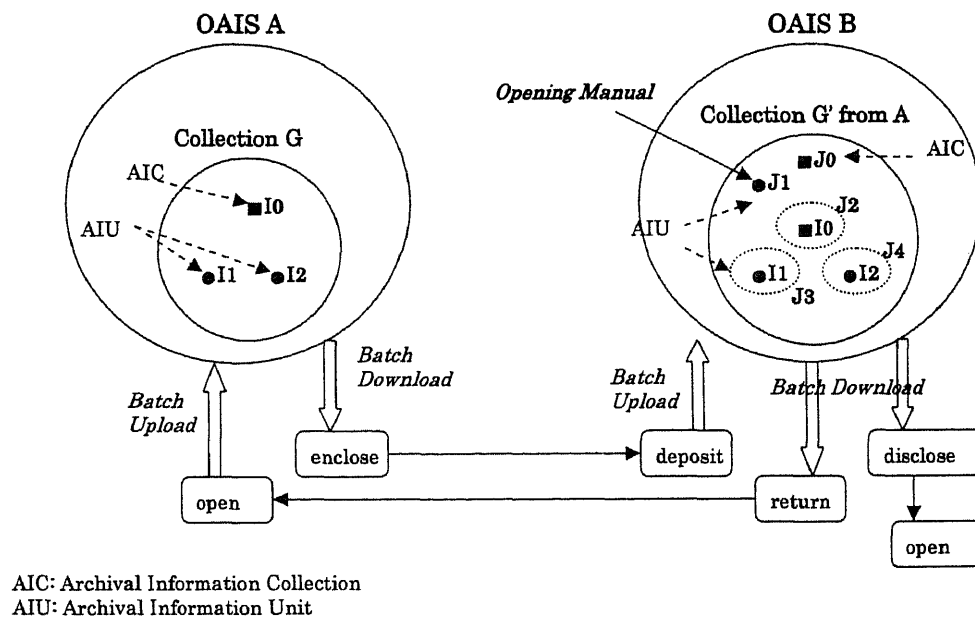


Fig. 4 Depositing a Collection between OAIS's

(1) 封入と預け入れ

源側で源システムからコレクションをダウンロードし、一連の封入文書を作成する。作成された一連の封入文書に目的側でそれらに PDI を付加して、目的システムにアップロードし、コレクションを形成する。(図 4)

(1.1) 源側：ダウンロードと封入

AIC と AIU は共に AIP であるので、封入文書の作成程はいずれも 3 で述べたのと同じである。これらを源システムから一括ダウンロードの上、一件ごと封入作業を行う。処理対象が多数で手作業は困難なので、それらを一括処理するプログラムを作成して実行する。

すべての封入文書に共通な開封説明書（返却・開示プログラムも含む）を 3 に示したように作成する。なお、コレクションの一括ダウンロード機能は、OAIS では標準装備されているものと考えられる（DIP: Dissemination Information Package の一形態として）。

(1.2)目的側：預け入れとアップロード

目的システム側において、預かるべき個々の封入文書を 3 と同様、AIP の CDO 部に入れるが、多数なので、目的側 AIP の生成を一括処理するプログラムを作成する。その際、PDI の記述には、預かった袋を識別できるように、プログラムで生成する一連番号 i を入れる。つまり、PDI の記述は「OAIS A から、XX 年 YY 月 ZZ 日に、預かった物 i」となる。なお、すべての RI 部には、3 と同様、「XML 文書」と記述する。一括処理で生成した一連の AIP を目的システムに一括アップロードする。一括アップロード機能は、OAIS に標準装備されているものと考えられる（SIP: Submission Information Package の一形態として）。一括アップロードされた AIP を AIU とみなして、目的システムで一つのコレクションを形成する。さらに、開封説明書をこのコレクションに AIU として付け加える。このコレクションの AIC は目的側で作成する。開封説明書と AIC の登録は手作業でもよい。

(2)返却と開封

まず、目的システムで預かりコレクションを同定し一括ダウンロードする。ダウンロードされたすべての目的側 AIP の CDO 部から封入文書を取り出し、その開封説明書と共に、源システムに返却される。開封説明書と（もしあれば）添付された開封プログラムにより、一連の封入文書から源側 AIP（AIC および AIU）を取り出し、源システムに一括アップロードし、コレクションの復元を行う。

(3)開示と開封

目的システムで預かりコレクションを同定し一括ダウンロードする。ダウンロードされたすべての目的側 AIP の CDO 部から封入文書を取り出す。それら封入文書（XML 文書）を接続し、一つの XML 文書に構成する。これを Microsoft Excel で取り込めば、封入文書、つまり源側 AIP の内容を表の形（Excel 文書）で一覧できる（図 5）。表示に先立って、源側 AIP の CDO 部にはデータオブジェクトに代わって、ダウンロードファイル中の封入文書の所在を示すポインタを入れておく。一覧表の PDI 部（1.の例では日本語）を読むことによって、必要なものがあればその封入文書を開封する。込み入った条件で検索するときには、Excel 文書を Microsoft Access で開く。

ID	CDO	RI	PDI
I0	C:/ppp/qqq/1/rrr0.xml	xxx	xxx
I1	C:/ppp/qqq/2/rrr1.xml	xxx	xxx
I2	C:/ppp/qqq/3/rrr2.xml	xxx	xxx

CDO: Location of the enclosing document instead of data objects
 xxx: Japanese text (in the case of Example Fig.1)

Fig. 5 Table of source AIP's obtained by disclosing

5. DSpace によるモデルの実証実験

提案したモデルを検証するために、実際にパイロットシステムを構成した(図6)。OAISに相当するアーカイブシステムとしてDSpace[2]を用い、預け入れ・返却・開示作業を行う環境としてWindows XPを用いた。源側、目的側の両方とも同じDSpaceなので、このモデルの検証に適していないと思うかもしれない。しかし、源側から取り出したAIPをそのまま目的側に預ける訳ではないので、問題はない。目的側では袋詰めされたAIP、すなわちXML文書を預かるので、源側のシステムアーキテクチャは何であってよい。同様に、源側では預け入れ時に袋詰めしたAIPを袋のまま返されるので、目的側のシステムアーキテクチャは何であってよい。

実証実験では、コレクション単位での預け入れ・返却・開示を行った。DSpaceには、一件処理のためのWebインタフェースと一括処理のためのコマンドがある。Batch Item ExporterおよびBatch Item Importerは、それぞれ一括ダウンロード用、一括アップロード用のコマンドである。いずれも、コレクションを指定してその要素を一括処理する。これらBatchコマンドはDSpace simple archive format [3]を使用している。このformatは多くのアイテムからなるディレクトリで、アイテムごとにサブディレクトリがある。各アイテムのディレクトリはAIP部を含むファイルを含んでいる。

これに基づき、封入(enclose)、預け入れ(deposit)、返却(return)、開示(disclose)の各機能を作成し、図6のようにそれぞれのWindows XPにインプリメントした。開示にはMicrosoft ExcelとMicrosoft Accessを利用した。

この実証実験は、モデルの検証のみならず、モデルの形成過程に役立った。

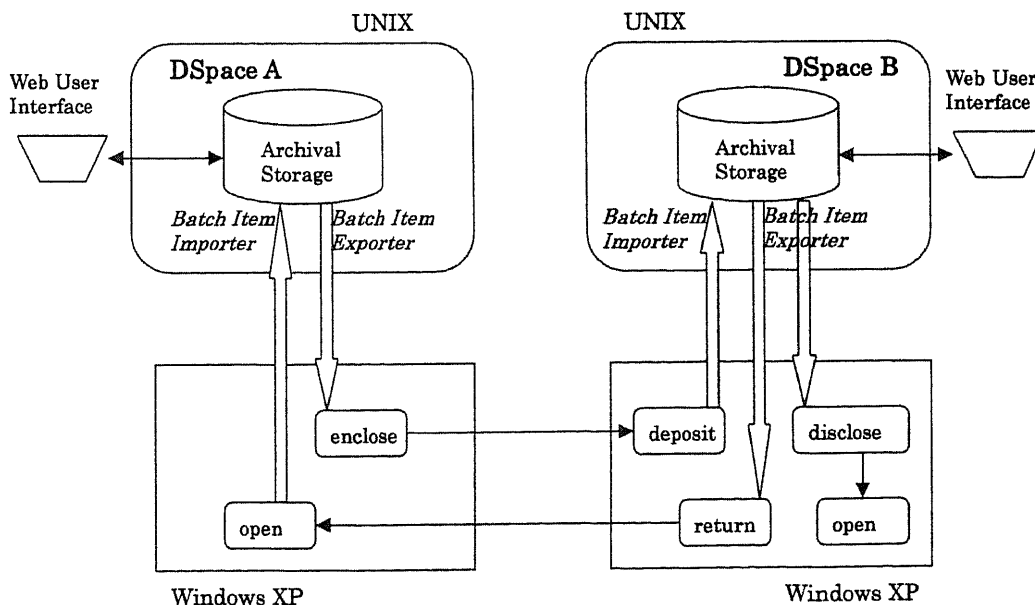


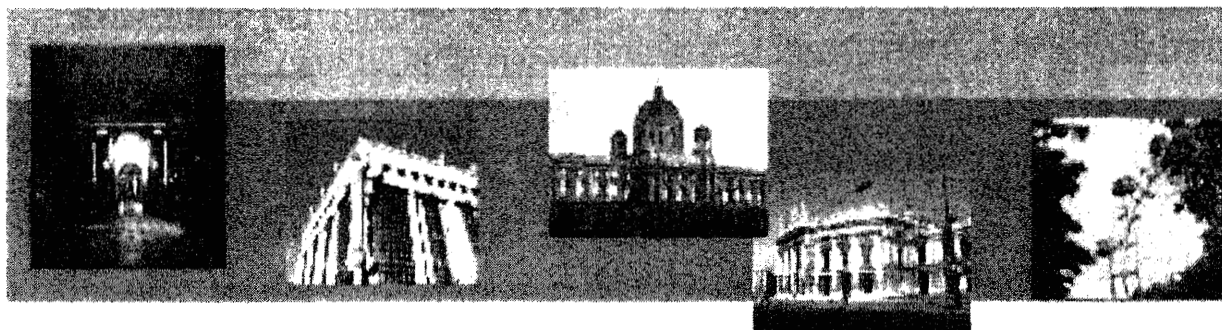
Fig.6 Verification of the Model using DSpace

参照文献

1. *Reference Model for an Open Archival Information System (OAIS)*. Blue Book. Issue 1. January 2002. This Recommendation has been adopted as ISO 14721:2003, <http://ssdoc.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf>
2. Robert Tansley, et al; DSpace as an Open Archival Information System: Current Status and Future Directions, ECDL2003, Lecture Notes in Computer Science 2769, pp446-460, 2003
3. Robert Tansley, et al; DSpace System Documentation, 2004, <http://dspace.org/technology/system-docs/>

5th International Web Archiving Workshop (IWAW05)

held in conjunction with the
8th European Conference on Research and Advanced Technologies for Digital Libraries
September 22 and 23 2005, Vienna, Austria



Objectives

Following the great success of the first four International Workshops on Web Archiving in Darmstadt, Germany in 2001 ([WebArchiving 2001](#)), Rome, Italy, in 2002 ([WebArchiving2002](#)), in 2003 ([WebArchiving 2003](#)), and in Bath, UK, in 2004 ([WebArchiving 2004](#)), we are happy to invite you to the fifth Workshop in this series.

The workshop will provide a cross domain overview on active research and practice in all domains concerned with the acquisition, maintenance and preservation of digital objects for long-term access, with a particular focus on web archiving and studies on effective usage of this type of archives.

It is also intended to provide a forum for interaction among librarians, archivists, academic and industrial researchers interested in establishing effective methods and developing improved solutions for data acquisition, ingest, and accessibility maintenance.

See the original [call for paper](#).

See details of the venue on the [conference website](#).

Agenda

([printable version](#))

Thursday 22nd of September	Friday 23rd of September
9h:00 Welcome and Introduction <i>Julien Masanès & Andreas Rauber</i>	session 4: Digital Preservation
session 1: IIPC Results	<u>Modular emulation as a long-term preservation strategy for digital objects (paper)</u> <i>Jeffrey van der Hoeven</i> De Koninklijke Bibliotheek, Netherlands
<u>The International Internet Preservation Consortium (IIPC)</u> <i>Catherine Lupovici</i> IIPC Programme Officer, Bibliothèque nationale de France	<u>Dynamic Web File Format Transformations with Grace (paper)</u> <i>Frank McCown</i> Old Dominion University, USA
<u>Web Archive's Architectural design</u> <i>Julien Masanès</i> European Archive, The Netherlands	<u>Preserving the bits of the Danish Internet (paper)</u> <i>Niels Christensen</i> <i>netarchive.dk, Denmark</i>
<u>Update on Heritrix development</u> <i>Michael Stack</i> Internet Archive, USA	<u>Selecting Preservation Strategies for Web Archives</u> <i>Stephan Strodl, Andreas Rauber</i> Vienna University of Technology, Austria
<u>Incremental crawling with Heritrix (Paper)</u> <i>Kristinn Sigurðsson</i> Landsbókasafn Íslands - Háskólabókasafn, Iceland	<u>PANIC – An Integrated Approach to the Preservation of Complex Digital Objects using Semantic Web Services (paper)</u> <i>Jane Hunter</i> DSTC, Brisbane Australia
<u>WERA: providing an Interface to Web Archives</u> <i>Michael Stack</i>	<u>A Collaboration Model between Archival Systems to Enhance the Reliability of Preservation by an Enclose-and-Deposit Method (paper)</u> <i>Koichi Tabata</i> University of Tsukuba, Japan
<u>WARC: an archiving format for the Web</u> <i>John Kunze</i> California Digital Library, USA	
<u>Metadata for Web Archiving</u> <i>Julien Masanès</i> European Archive, The Netherlands	

<p><u>Introduction to Nutch</u> <i>Doug Cutting</i> Lucene, Nutch, Internet Archive, USA</p>	<p>session 5: Current Project & Issues</p>
<p><u>Full Text Searching of Web Archive Collections (paper)</u> <i>Michael Stack</i> Internet Archive, USA</p>	<p><u>Archiving the danish web - status on netarchive.dk</u> <i>Bjarne Andersen</i> netarchive.dk, Denmark</p>
<p>session 2: Audio & Video Web Archiving</p>	<p><u>Role & utility of a technological partner for Web Archiving</u> <i>Julien Masanès</i> European Archive, The Netherlands</p>
<p><u>A technical approach for the French Web Legal Deposit (paper)</u> <i>Thomas Drugeon</i> Institut national de l'Audiovisuel, France</p>	<p><u>Creating a cooperatively-driven long-term archive of digital data in Germany - Experiences from the kopal Project -</u> <i>Olaf Brandt</i> Göttingen State and University Library, Germany</p>
<p>session 3: Time dimension</p>	
<p><u>The Availability and Persistence of Web References in D-Lib Magazine (paper)</u> <i>Frank McCown</i> Old Dominion University, USA</p> <p><u>Expression and share of temporal knowledge in archaeological digital documentation (paper)</u> <i>Tiphaine Accary-Barbier</i> INSA de Lyon, France</p>	



The workshop papers and presentations are licensed under a [Creative Commons Attribution-NonCommercial-NoDerivs 2.0 France License](https://creativecommons.org/licenses/by-nc-nd/2.0/fr/).

Workshop Officials

Organizing Committee

Julien Masanès (European Archive / e-mail : julien AT iwaw.net)

Andreas Rauber (Vienna University of Technology, Austria)

Program Committee

Andreas Aschenbrenner (SAT Research Studio, Austria)

Michael Bauer (TU München, Germany)

Birgit Henriksen (Det Kongelige Bibliotek, Denmark)

Michele Kimpton (Internet Archive, USA)

John Kunze (California Digital Library, USA)

Mark Middleton (British Library, UK)

Heike Neuroth, (State and University Library of Lower Saxony Göttingen, Germany)

Kjetil Nørvåg (University of Science and Technology, Norway)

Margaret E Phillips (National Library of Australia, Australia)

Philippe Rigaux (Orsay Paris-Sud University, France)

Steven M. Schneider (SUNY Institute of Technology, USA)

**A Collaboration Model between Archival Systems
to Enhance the Reliability of Preservation
by an Enclose-and-Deposit Method**

*Koichi Tabata, Takeshi Okada,
Mitsuharu Nagamori, Tetuso Sakaguchi,
and Shigeo Sugimoto
University of Tsukuba, Japan*

5th International Web Archiving Workshop (IWAW05)

2005-9-23

IWAW05 (ECDL'05), Vienna

1

Background and Goal

- Archiving/preservation of digital resources at many libraries, museums, archives, governmental sectors, etc.



Heterogeneity of archives

- Size of Archives
- Type of Archived Resources
- Architecture and Interface of Archival System
- Archive and Preservation Policies
- Language for Describing Metadata

2005-9-23

IWAW05 (ECDL'05), Vienna

2

Background and Goal (cont'd)

- Are the archives of digital resources really reliable?

Risks

- Shortage of human resources and funding to keep archives alive
- Organizational changes
- Natural and human-caused disasters



High risks especially in small communities

2005-9-23

IWAW'05 (ECDL'05), Vienna

3

Background and Goal (cont'd)

- Crucial issues to make digital archives highly reliable
 - Simplicity
 - Inexpensiveness in system cost and in human labor



- Interoperability

2005-9-23

IWAW'05 (ECDL'05), Vienna

4

Background and Goal (cont'd)

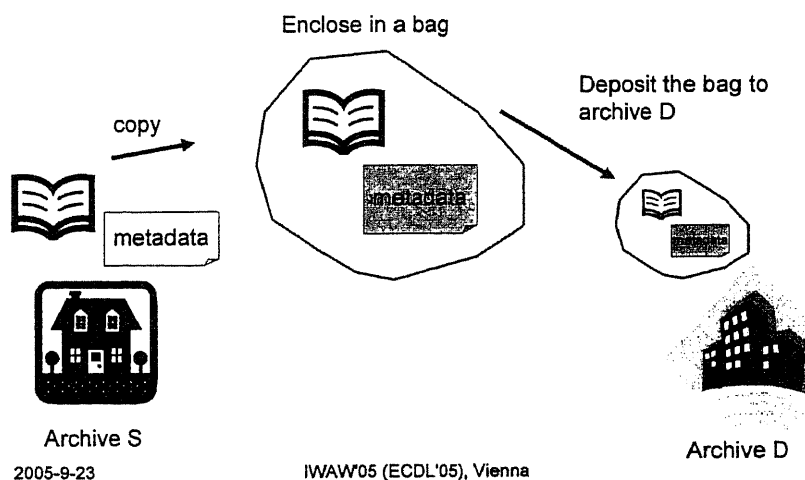
- Our approach, “Enclose-and-Deposit”
 - Enclose a preserved resource into a bag with an appropriate description and deposit the bag to a collaborating archive (or archives)
- Preservation by collaborating archives
 - Simple collaboration protocol to reduce costs and to enhance interoperability
- Use XML to realize the bag
 - Simple and stable

2005-9-23

IWAW05 (ECDL'05), Vienna

5

Enclose and Deposit

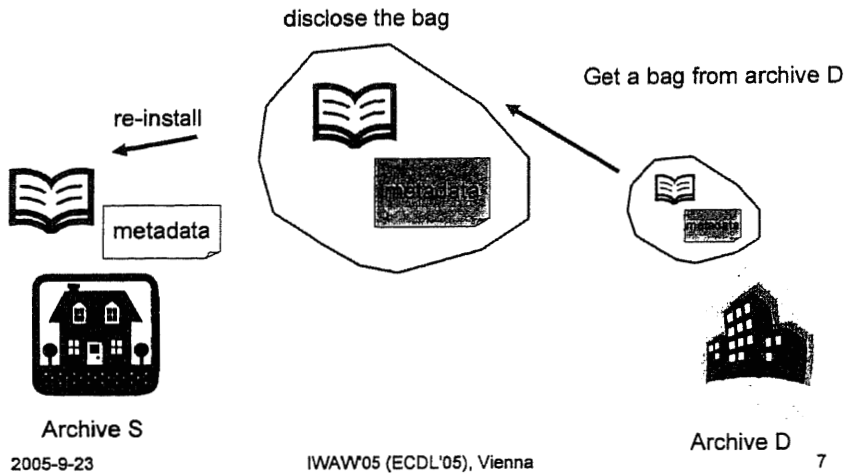


2005-9-23

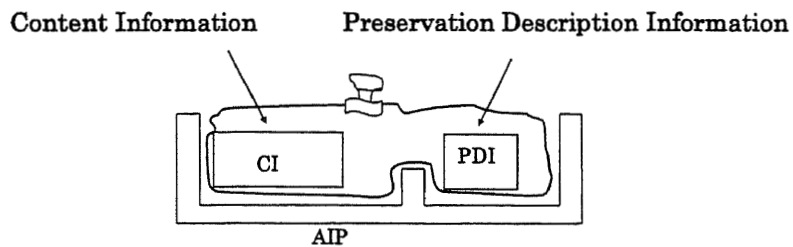
IWAW05 (ECDL'05), Vienna

6

Enclose and Deposit



Bag and AIP



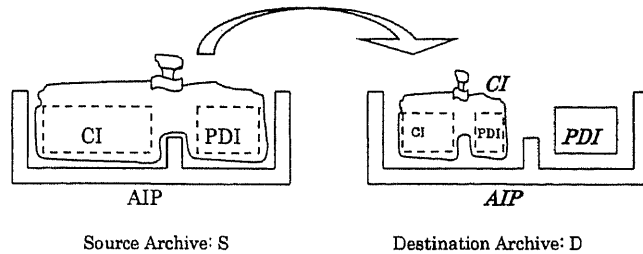
Archived Resource enclosed in a bag
Archival Information Package (AIP) of OAIS

2005-9-23

IWAW'05 (ECDL'05), Vienna

8

Bag and AIP



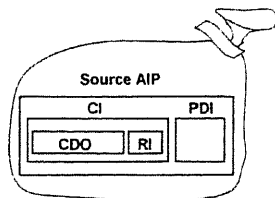
Concept of Deposition between OAISs

2005-9-23

IWAW05 (ECDL'05), Vienna

9

Bag and AIP



A Bag created
from a source AIP

```
<?xml version="1.0" encoding="xxx"?>
<AIP>
  <CI>
    <CDO>xxx</CDO>
    <RI>xxx</RI>
  </CI>
  <PDI>xxx</PDI>
</AIP>
```

XML Implementation of a Bag

A Skeleton of an XML instance which encloses Source AIP

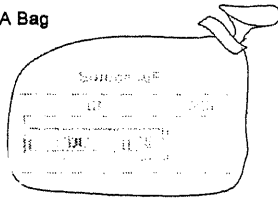
2005-9-23

IWAW05 (ECDL'05), Vienna

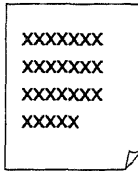
10

Bag and AIP

A Bag



A manual prepared by the source archive for archive managers in future



describes how to open and restore the bag in the source archive environment

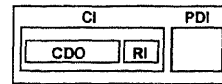
Realized as an XML instance

```
<?xml version="1.0" encoding="xxx"?>
<AIP>
  <CI>
    <subCI><CDO>xxx</CDO><RI>xxx</RI></subCI>
    <subCI><CDO>xxx</CDO><RI>xxx</RI></subCI>
    <subCI><CDO>xxx</CDO><RI>xxx</RI></subCI>
  </CI>
  <PDI>xxx</PDI>
</AIP>
```

An XML instance

(A Example of multiple Objects)

Source AIP restored



A Manual required to restore a bag

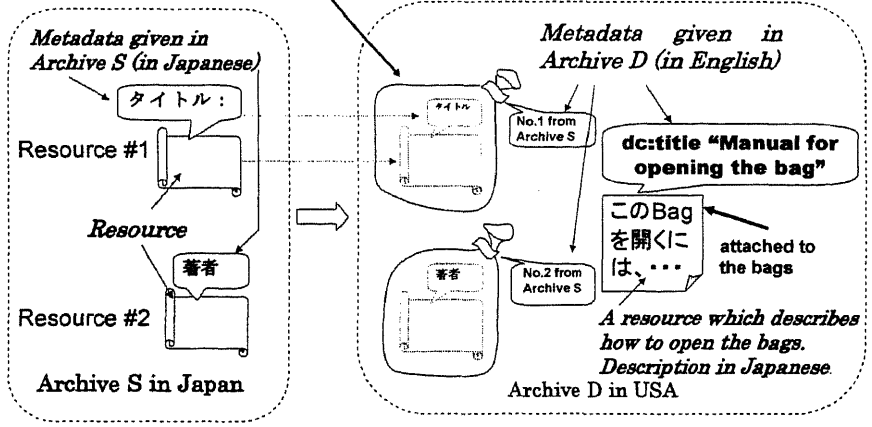
2005-9-23

IWAW05 (ECDL'05), Vienna

11

Bag and AIP

A resource and its metadata enclosed in a bag.



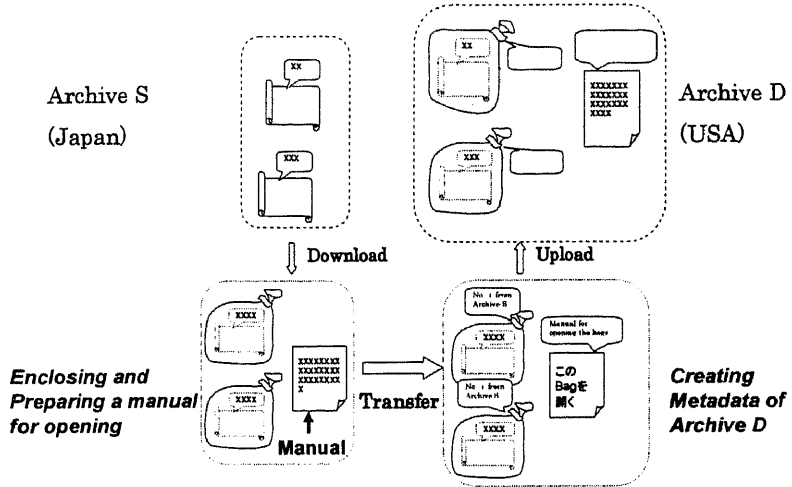
An enclose-and-deposit Model between Archives

(example: collaborating archives in Japan and USA)

2005-9-23

12

Enclose and Deposit



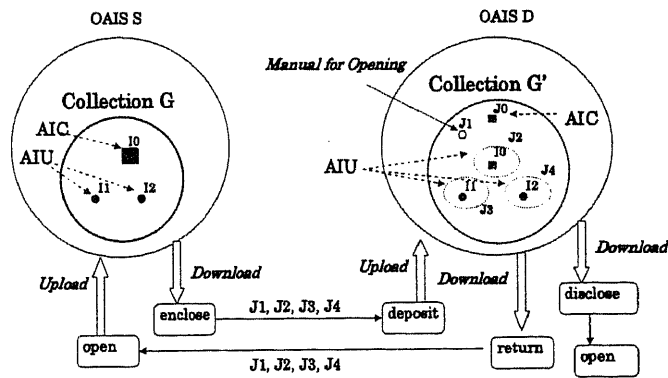
Procedure for an Enclose-and-Deposit Method

2005-9-23

IWAW05 (ECDL'05), Vienna

13

Enclosing a Collection



AIC: Archival Information Collection
AIU: Archival Information Unit

○ Enclosed instance

Depositing a Collection between OAISs

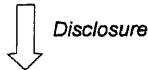
2005-9-23

IWAW05 (ECDL'05), Vienna

14

Encoding of a Collection

```
<?xml version="1.0" encoding="xxx" ?>
<Collection>
  <AIP><CI><CDO>yy0</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></AIP>
  <AIP><CI><CDO>yy1</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></AIP>
  <AIP><CI><CDO>yy2</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></AIP>
</Collection>
```

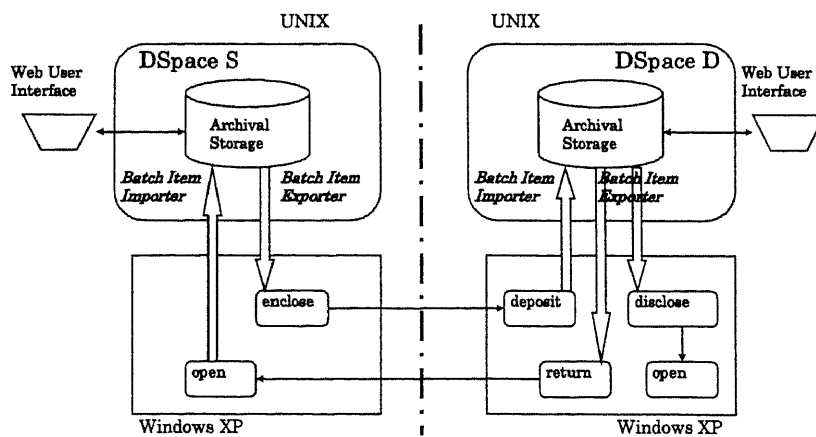


CDO	RI	PDI
C:/ppp/qqq/1/rrr0.xml	xxx	xxx
C:/ppp/qqq/2/rrr1.xml	xxx	xxx
C:/ppp/qqq/3/rrr2.xml	xxx	xxx

CDO: Location of the enclosing document instead of data objects

Table of Source AIPs obtained by Disclosing the Bag of Bags

Implementation

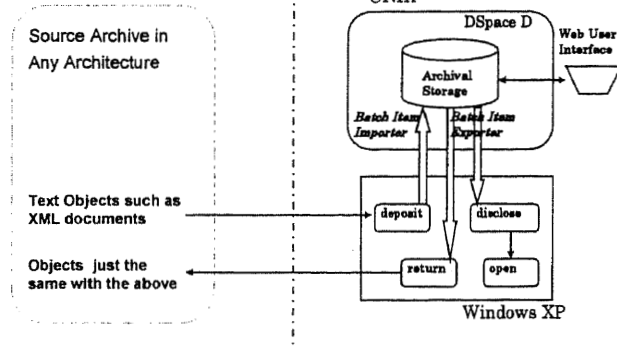


Experimental Implementation for verifying the functionality of the "Enclose-and-Deposit" model using DSpace

Implementation

Source Archive

Destination Archive



Verification of the Destination Side using DSpace

2005-9-23

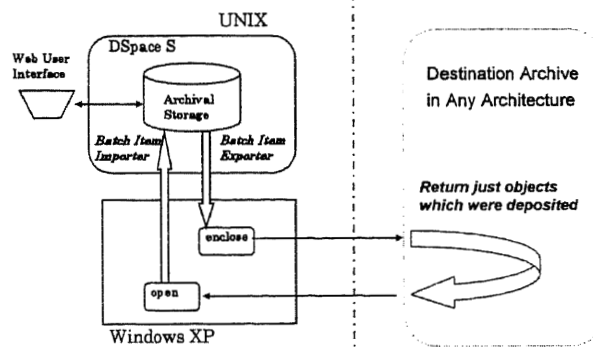
IWAW05 (ECDL'05), Vienna

17

Implementation

Source Archive

Destination Archive



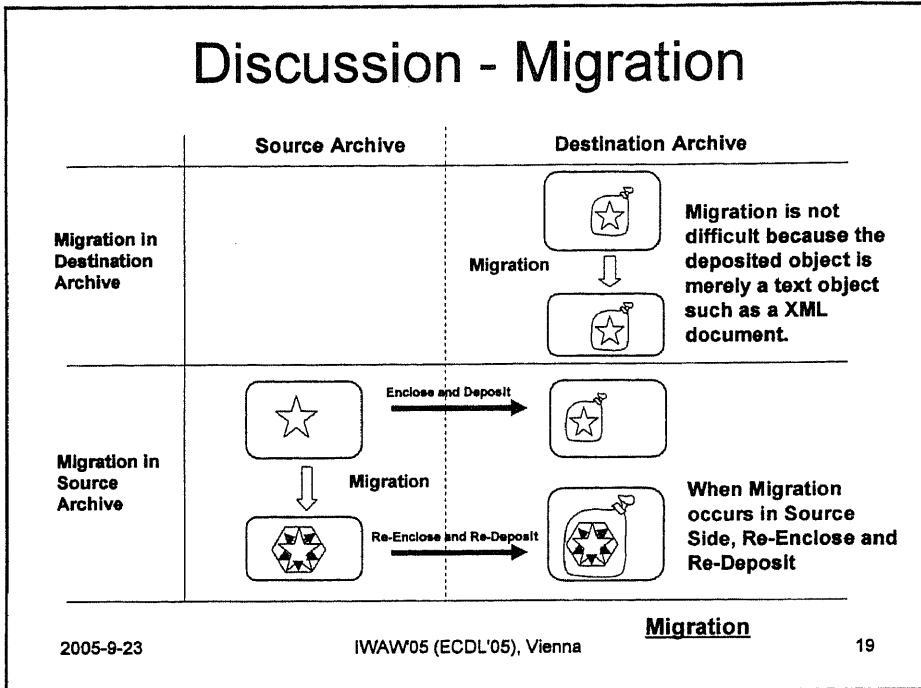
Verification of the Source Side using DSpace

2005-9-23

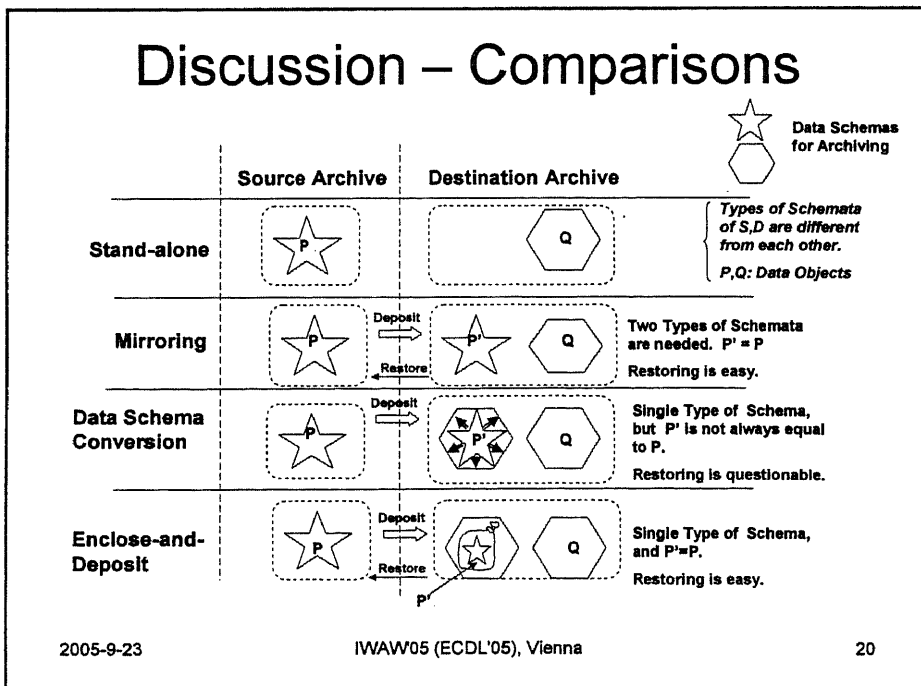
IWAW05 (ECDL'05), Vienna

18

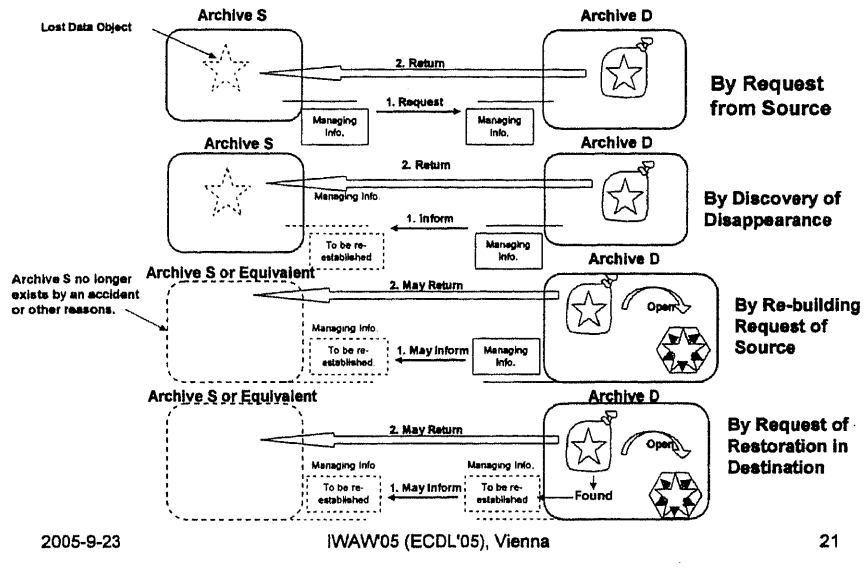
Discussion - Migration



Discussion – Comparisons



Discussion – Usage Scenarios



Summary

- A Simple Scheme for Decreasing Risks of Preservation
- Simplicity for Heterogeneous Communities

Thank You!

A Collaboration Model between Archival Systems to Enhance the Reliability of Preservation by an Enclose-and-Deposit Method

Koichi Tabata, Takeshi Okada, Mitsuharu Nagamori, Tetsuo Sakaguchi, and Shigeo Sugimoto

Graduate School of Library, Information and Media Studies, University of Tsukuba, 1-2 Kasuga, Tsukuba, 305-8550 Japan
{tabata, okadatak, nagamori, saka, sugimoto}@slis.tsukuba.ac.jp

Abstract. This paper proposes a model of cooperating archival systems that mutually deposit archived resources in order to enhance the reliability of long-term preservation. In the model, a resource and its preservation metadata in a source archive are enclosed together in a bag, and the bag is deposited in a destination archive as a resource. The proposed model is a simple approach, but it is advantageous for the following reasons: (1) Duplication of resources enclosed in a bag preserved under different environments enhances the reliability of archiving; (2) Simplicity is an essential factor in making the technology adoptable in small archives and in making preservation more reliable; (3) No archival systems are permanent, especially those created by small organizations; (4) Re-writing or transcribing metadata is not necessary. This paper first discusses the requirements for collaborating distributed archives and then proposes a simple “enclose-and-deposit” model. The model is examined for both deposit of a single resource and deposit of a collection of resources. The last section describes an implementation of the model, using XML as the encoding scheme and DSpace as the underlying architecture of the collaborating archival systems.

1. Introduction

Precious old Japanese material that no longer exists in Japan has often been reported to have been “discovered” in an archive in the United States, implying an effective method for the long-term preservation of information resources. Analysis of the specific circumstances often reveals that an archive in Japan did not try to preserve that material, not realizing its importance for the future, or lost the material despite the intention to preserve it. In the American archive, however, it is assumed that such material is important even though its contents are unclear. The person in charge at that time, despite his/her inability to understand Japanese, typically encloses the material in a storage bag and preserves the bag somewhere in the

archive after noting on its surface, in English, the origin of those materials. Re-discovered later, this material's importance is recognized anew; it is formally registered as belonging to that archive; and an explanation of the contents is provided in English. Such material may be very important for Japan as well, therefore, a decision is often made to store a copy of the material in a Japanese archive also.

Such a process occurs even if the archived material is a digital resource. Many resources are created and published on the Internet. Deposit libraries are making efforts to collect and archive these resources, but it is difficult to evaluate their importance and to describe detailed metadata as the OAIS reference model recommends. The only organization (or person) that can appropriately describe the metadata required for preservation of a resource is the one that has created the resource. However, it is difficult for the organization to run its own archival system in a secure environment and for a very long time, except in such cases as legal deposit libraries and large-scale governmental institutions. Collaboration among those organizations is a key to realizing reliable archival systems.

We borrow the phrase "Lots of Copies Keep Stuff Safe" from the LOCKSS project [1] to encourage the creation of multiple legal copies of archived resources in order to enhance the reliability of archives. The OAIS reference model does offer a highly reliable self-contained model to preserve digital resources in a single system. However, in the Internet environment, many organizations run their own archival systems using open source software such as DSpace, and not all archival systems are as reliable as they should be. Thus, in order to enhance the reliability of preservation, we need to build a network of collaborating archival systems that mutually deposit copies of their valuable resources.

2. Requirements Analysis for Collaborating Archival Systems

This section analyzes the scenario presented in the previous section in order to establish requirements for a model of long-term preservation by collaborating archives.

- (1) An archive in Japan plans long-term preservation of their valuable digital resources. However, if a resource is preserved in only the local archive, it may be lost someday, so they decide to deposit copies of important items in another archive in the United States. Because the destination archive for depositing the resources also has the responsibility of preserving digital resources, the survival of the resources deposited from Japan is highly probable.
- (2) If that resource in the local archive is lost later, it can be recovered from the destination archive.
- (3) Still later, not only the name of the resource but also the name of the destination archive is no longer in people's memory. At some point, the resource will happen to be found in the destination archive, and the resource will be restored to the source archive. If the source archive no longer exists, the destination archive may preserve

the resource and offer it for public perusal, or transfer it to an equivalent archive in Japan.

With regard to such a situation, it is important to investigate several issues for preservation by collaborating archival systems.

(a) The resources deposited into the destination archive should not be directly provided to end-users by the destination archive, because the destination archive is a back-up function for the source and the audience community of the destination may not overlap that of the source archive. Thus, the destination archive is not a mirroring archive of the source.

(b) In general, the metadata schemata of the archives will differ. In this example, the languages used to describe the metadata are different. To handle the deposited resource in a schema equivalent to that of its own archive, transcription and/or translation of the metadata from the original form into the target schema must be performed to transfer each deposited resource. However, the costs for transcription and translation should be avoided unless the destination archive directly provides deposited resources for the end-users.

(c) Regardless of the contents of the resource or the metadata from the source archive, the destination archive will receive the deposited resource as well as the metadata from the source archive (the Japanese archive in the above case), all enclosed in a bag, and will add the note "Deposited Resource P" where P is a simple identification number. Also noted on the surface of the bag will be a description of when and from what archive the resource was received, using the description language (English in this case) and the metadata system of its own archive. For example, the added notes may read [Title: Deposited resource P], [Creator: Source archive]. If the person in charge does not understand the metadata system or the language (Japanese) of the source archive, the metadata (in English) for its own archive is prepared for the document describing the method (in Japanese) for opening the bag, and is stored with the bag.

(d) When the destination archive receives a request to return a resource, the deposited resource will be located by referring to the metadata of its own archive and returned as enclosed in the bag, together with the document indicating the method of opening. The source archive will open the bag by referring to the document, take the resource and metadata from the bag, and store them in its own archive.

(e) In the future, if the source archive no longer exists, the corresponding bag will be identified and opened at the destination archive to investigate the resource and its metadata. In the previous example, it is natural to assume that the destination organization could hire a person who understands Japanese or could request assistance from another archive that has experiences in handling Japanese resources. The resource is then registered formally as an original resource of its own archive, depending on the necessity of offering it for public perusal. The metadata in Japanese will be rewritten into the proper system for its own archive, and the description language will be translated into English. Because sufficient time will likely have passed since the resource was created, there should be no copyright problem; therefore, the resource can be offered for public perusal.

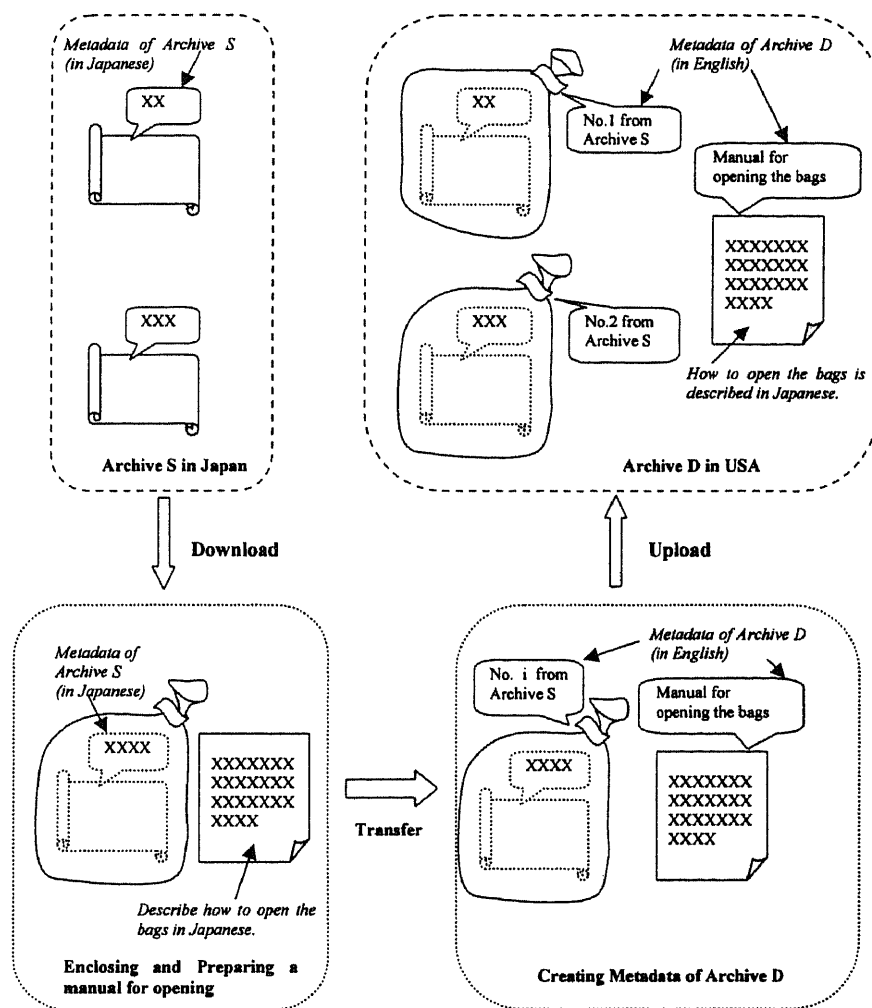


Fig. 1. Basic Concept of Depositing between Archives: On the source side, a resource and its preservation metadata in Archive S are enclosed together in a bag, and a prepared manual describes how to open the bag. The bag and the manual are transferred to the destination, and deposited in Archive D with D's preservation metadata.

This model is presented in Fig. 1. Archive S, the source archive in Japan, selects the resource and metadata to be deposited. Enclosing both of them in a bag and preparing a manual for opening the bag, the source archive sends them to Archive D in the United States. Archive D preserves the bag and its manual for opening by attaching the proper metadata. The metadata to be attached to the bag may be simple, indicating only the source archive (e.g., Archive S, the date of deposit, and the serial number or identifier of the item). The metadata for the manual for opening should

describe the method for opening the deposited resource. The return, disclosure, and opening will be performed at a later time by referring to the manual.

This scenario emphasizes the differences between archives in different countries (e.g., those in Japan and those in the United States). However, even in the same country or in regions where the same language is spoken, differences exist between archives that serve different communities. As mentioned in OAIS, their metadata schemas differ, depending on the properties of the data involved. For example, the data properties for the archive of a university library and those for that of a space observation center are not the same; however, they can collaborate to preserve resources.

Requirements determined from the above scenario are as follows.

(i) Neither the source archive nor the destination archive is required to add any special functions beyond those vital to the management functions of the archive. No functions other than download and upload will be used.

(ii) The work environment to transfer the archival information packages between the archives does not have to be retained and would be set up only when required. Even long afterwards, the work environment for deposit and return work may be re-established by referring to the manual for opening, which should be retained with the resource. Thus, the only items that have to be preserved are the bag and its manual for opening. As long as these items are maintained by the destination archive, the resource can be recovered.

(iii) The metadata schema and the description language may differ between the collaborating archives, but mutually converting their schemas is unnecessary. Deposit can easily be accomplished even between OAISs that preserve different forms of data.

(iv) The deposited resources can be re-deposited to a third archive. The bag can be placed in a new bag and deposited in the third archive. The metadata attached to the bag will describe the deposit history and the required function to recover the enclosed resource. This feature clearly delineates the difference between this model and the conventional backup functions, including mirroring.

3. Collaborative Inter-Archive Deposit of Archival Information

Package

In the ISO reference model for an Open Archival Information System (OAIS), an OAIS is defined as an archive that has accepted the responsibility for preserving information and making it available for a Designated Community [2]. These communities are diverse, and the properties of the information and data stored by individual communities differ from each other. Nevertheless, the OAIS reference model specifies a framework of guidelines to be considered as the minimum requirements from the viewpoint of long-term preservation, while allowing individual communities to establish free archival forms outside the framework. In addition, the

OAIS reference model presents several models for consumer services provided through the mutual linkages of different archive systems. However, it has not yet presented a model representing long-term preservation through the mutual linkages of these systems.

In this paper, we propose a model projecting additional long-term preservation through mutual linkages between archive systems that comply with the OAIS reference model, based on the discussions described above. Once an archive system has been established, the community will endeavor to maintain it indefinitely by involving the whole organization, so long-term archival resources must be sufficiently protected in order to survive. If such archive systems can mutually deposit particularly important resources, the resources will be more reliably preserved. However, metadata schemas differ from one community to another, so rewriting the metadata of resources deposited from a source archive into the destination archive will require a tremendous volume of work. In this paper, we propose a system to enable the mutual storage of resources between different archives without necessitating such work.

In the OAIS reference model, information to be preserved is expressed by means of an Archival Information Package (AIP) (Fig. 2). An AIP is a conceptual container holding two types of information: Content Information (CI) and Preservation Description Information (PDI). A CI consists of the Content Data Object (CDO) and its associated Representation Information (RI) needed to make the CDO understandable to the Designated Community [2].

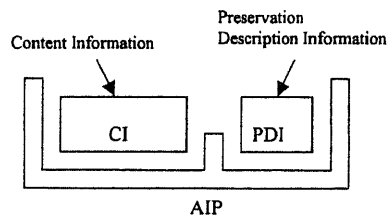


Fig. 2. Archival Information Package (AIP)

The resource and its metadata for preservation correspond to the CI and PDI of the AIP, respectively. The packaging of the resource with its associated metadata in a bag by Archive S and the preservation of the bag by attaching the metadata for Archive D by Archive D correspond to the following process (Fig. 3).

- The source archive (OAIS S) encloses the CI and the PDI in a bag,
- the destination archive (OAIS D) puts the bag in the CI of its own AIP.
- The PDI at the destination is not rewritten from that at the source but contains descriptions such as "Item P deposited by OAIS S on XX YY ZZ" (P is the identification number).
- The position where the bag is put at the destination is, to be precise, the CDO section of the CI.

The manual for opening prepared by Archive S corresponds to the following process; the manual for opening prepared at the source is put in the CI of the AIP of

the destination system, and the metadata created to enable the destination system to identify it as the information for opening is put in the PDI. As in the above case, to be precise, the manual is put in the CDO section of the CI.

This nested architecture help avoid complicated tasks to interoperate the source and destination archives and allow “on-demand disclosure” of the packaged information by the destination archive.

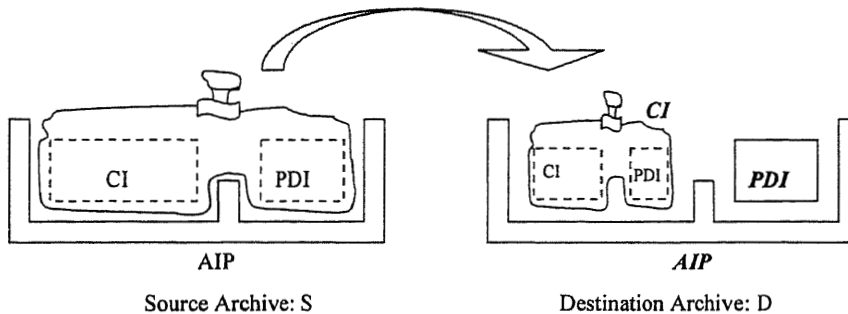


Fig. 3. Concept of Deposition between OAISs

4. Model of Inter-archive Deposit of a Single Resource

Before describing the model in more detail, we define the terms listed below.

(a) “Enclosing” means to express the contents of the AIP of the source archive, i.e. the contents of its components, CI and PDI, as a single XML document. This document is called the enclosing document. Before being enclosed in the XML document, the bit-string portions are converted into character strings by base64Binary. The character set used in the document is specified by the XML encoding declaration.

(b) “Opening” means to extract the contents of the enclosed AIP, i.e. the contents of its components, CI and PDI, from the enclosing document. Character strings converted by base64Binary are reconverted into the corresponding bit strings.

(c) “Manual for opening” refers to the document that enables opening the enclosing document by describing how the enclosing document was prepared.

(d) “Disclosure” means to view the contents of the deposited enclosing document on the destination archive. The enclosing document is an XML document, so it is possible to view its contents by preparing the character set environment as specified by the XML encoding declaration.

Let us now explain the method for preparing enclosing documents in more detail. An enclosing document is prepared by expressing the contents of the AIP of the source system, i.e. the contents of its components, the CI and the PDI, with a single XML document as shown in Expression (1). In the CDO section, the character string

being converted using base64Binary is enclosed by <base64Binary> and </base64Binary >.

```
<?xml version="1.0" encoding="xxx"?>
<AIP>
  <CI>
    <CDO>xxx</CDO>
    <RI>xxx</RI>
  </CI>
  <PDI>xxx</PDI>
</AIP>
```

(1)

Expression (1) represents a skeleton of an XML instance of an enclosing document. The <CI> contains one or multiple data objects of different formats. In the case of multiple data objects, the element of the <CI> is expressed <subCI>xxx</subCI>...<subCI>xxx</subCI>. Example of multiple data objects is a hypertext document which is composed of a single HTML text and one or more JPEG images. Sometimes the RI element may be omitted, for example, in the case that the file name of a resource has a file extension, such as 'abcde.html' and 'pqrst.jpg'.

Expression (1) shows a general form of an AIP encoded in XML for transmission between the archives. The schema to encode a resource into an XML instance depends on each archive system. A document which explains the schema of the XML instance must be prepared by the source archive. The document will be referred to by the source archive to open the enclosing document and restore the resource. This document is called the "manual for opening", which basically consists of character text, and also contains a conversion method, e.g. base64Binary. The destination archive need not interpret the document unless it is explicitly requested to disclose an AIP and to reformulate the resource.

Deposit work - The destination archive receives an enclosing document from the source archive and creates a CI which is composed of the CDO sent from the source as an XML instance and an RI which states that the CDO is an XML instance. Then, the destination archive creates an AIP by packaging the CI with a description about the CI for preservation in the destination archive, i.e. PDI. Description in PDI depends on the policy of the destination archive but a simple example would be a text which tells the fact that the destination archive received the resource from another archive, e.g. "Item P deposited by OAIS S on XX YY ZZ" (P is the identification number).

A manual for opening, without being enclosed, is put in the CDO section of an AIP of the destination archive. The description language of the manual is given in the RI section. The PDI will record "the manual for opening the item deposited by OAIS S." If programs for opening and disclosure have been prepared in advance, they are put in the CDO section. Environment information for executing the programs is put in the RI section. The archive systems guarantee that any AIP transferred to a destination archive can be rebuilt as it was but they cannot guarantee that the programs enclosed in an AIP are executable at any time.

Return work - The enclosing document returned from the destination archive is opened by the source system and the recovered AIP is registered to the source system. Opening work is performed by referring to the manual for opening.

Disclosure work – In the case that the destination archive is requested to recover an AIP transferred from a source archive, the archive discloses the enclosing document, opens it as required, and registers it to the destination system.

5. Model for Inter-Archival Deposit of an Archived Collection

“Collection” is a unit to manage a set of resources collected based on a certain policy. A collection of resources is a natural unit for preservation. In the OAIS reference model, collections are expressed as follows. There are two types of AIP: one is an Archival Information Collection (AIC) and the other is an Archival Information Unit (AIU), which can be a member of the former. The CI of an AIU holds a single item of archival information. The CI of an AIC will hold information that enables identifying individual AIUs as members of the collection.

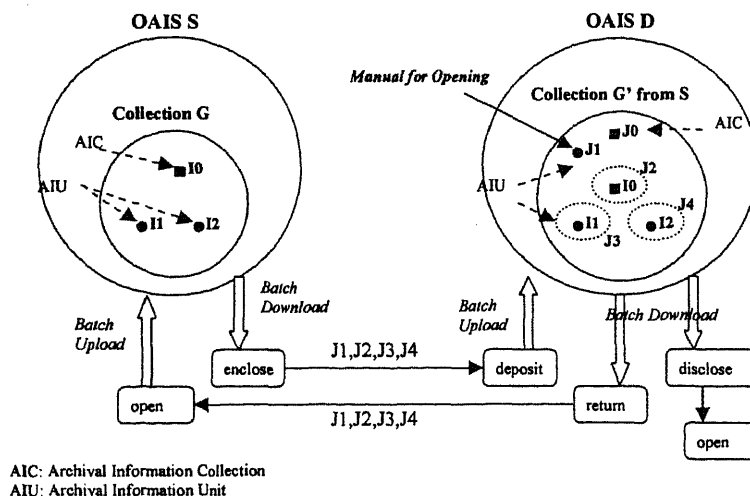


Fig. 4. Depositing a Collection between OAISs

(1) Enclosing and depositing

In Fig. 4, the source side which has a collection G prepares a series of enclosing documents J2, J3, and J4 for the AIC and AIUs. The enclosing documents J2, J3, and J4, together with a manual for opening J1, are transferred to the destination side. Then, the destination side creates AIPs (as AIUs of a collection G') from J2, J3, J4 and J1. In this process, the download/upload functions may be realized as a batch process.

(1.1) Source side: Download and Enclose

Because both the AIC and the AIU are AIPs, the process of preparing enclosing documents is the same as described in Section 4. These are downloaded as a batch from the source archive and enclosing work is performed item by item. A manual for opening applicable to all enclosing documents is prepared as described in Section 4.

(1.2) Destination side: Deposit and Upload

On the destination side, the process of creating destination AIPs from the enclosing documents transferred from the source side is the same as described in Section 4. Since a collection has a number of items, this process is realized as a batch process as well. When the destination side creates the AIPs, it assigns a unique identifier to each of their PDIs. The destination archive forms a collection G' by considering the batch-uploaded AIPs to be AIUs. The manual for opening J1 is added to this collection, also as an AIU. The AIC for this collection, i.e., J0 is prepared by the destination archive. The manual for opening and the AIC may be individually registered by operator's commands.

(2) Return and Opening

The destination archive identifies the deposited collections and downloads them as a batch. Enclosing documents J2, J3, J4 are extracted from all CDO sections in the AIPs of the destination archive and are returned to the source system, together with the manual for opening J1. The AIPs (AIC and AIUs) of the source side are extracted from a series of enclosing documents, referring to the manual for opening and any opening programs that may be attached. The AIPs are then uploaded as a batch to the source archive to recover the collections.

(3) Disclosing and Opening

The destination archive identifies the deposited collections and downloads them in a batch process. Enclosing documents are extracted from all CDO sections in the AIPs of the destination archive. The enclosing documents (XML documents) are arrayed into a single XML document as follows.

```
<?xml version="1.0" encoding="xxx" ?>
<Collection>
<AIP><CI><CDO>yy0</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></AIP>
<AIP><CI><CDO>yy1</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></AIP>
<AIP><CI><CDO>yy2</CDO><RI>xxx</RI></CI><PDI>xxx</PDI></AIP>
</Collection>
```

(2)

CDO	RI	PDI
C:/ppp/qqq/1/rrr0.xml	xxx	xxx
C:/ppp/qqq/2/rrr1.xml	xxx	xxx
C:/ppp/qqq/3/rrr2.xml	xxx	xxx

CDO: Location of the enclosing document instead of data objects

Fig. 5. Table of Source AIPs obtained by Disclosing:

For the management of archiving a collection by the collaborating archives, we need a management tool. In the implementation discussed in the next section, we use

Microsoft Excel to view the list of objects in Expression (2) as shown in Fig.5. In Fig.5, the elements in column CDO are pointers to indicate the locations of enclosing documents in the downloaded files.

6. A Pilot System of the Collaborative Archive Model using DSpace

To verify the proposed model, we have configured a pilot system (see Fig. 6). DSpace [3] is the archiving system, since it is widely used and fulfils the basic feature of the OAIS reference model. Windows XP is the environment for enclosing, depositing, returning, opening, and disclosing.

This DSpace-to-DSpace system is not a simple back-up mirroring. In the destination system, a resource deposited by the source system cannot be accessed directly by an end-user because it is enclosed in a bag. Furthermore, the function implemented in the destination DSpace is valid for any architecture of source system because the destination system merely receives a bag (i.e., an XML document) from it. Similarly, the function implemented in the source DSpace is valid for any architecture of destination system because the bag that was deposited is merely returned.

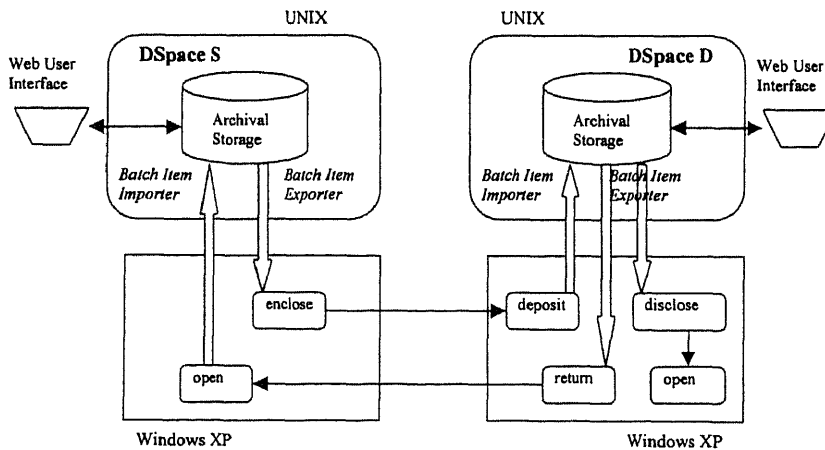


Fig. 6. Verification of the Model using DSpace

During the demonstration, enclosing, depositing, returning, opening, and disclosing were performed in units of collection. DSpace has a Web interface for item-by-item processing and a command for batch processes. Batch Item Exporter and Batch Item Importer are commands for batch downloading and batch uploading. Both specify a collection and then process its elements as a batch. These batch commands use the DSpace simple archive format [4], which is a directory full of items with a subdirectory per item. Each item directory contains files including AIP components. Here, file extensions correspond to the RI.

Each enclosing, depositing, returning, opening, and disclosing function was prepared based on the above process and was implemented under Windows XP, as depicted in Fig. 6. Microsoft Excel and Microsoft Access were used for disclosure.

7. Discussion and Conclusion

This study explored a simple approach to making archives more reliable. Such simple technology is indispensable for archives maintained by memory organizations of small communities, such as regional libraries and museums. On the one hand, each community needs its own policies to archive and preserve resources, which are valuable not only for the local community but also for the global community. On the other hand, these small communities are subject to change or may disappear over time.

The only solution is to archive the resources in another archive. However, such multiple archiving involves the challenges of intellectual property issues and interoperability between archives. Intellectual property issues are beyond the scope of this paper. Interoperability between archives is crucial in the heterogeneous Internet environment. It is unrealistic to assume that all archives use a single archiving software or that all archives use the same metadata schema. Thus, interoperability between archives is difficult to achieve.

A simple, open framework is required to solve the interoperability issue. The proposed model is simple: enclose an AIP in a bag and store the bag packaged in another AIP at a different archive. The model is open: the enclosing document format (i.e., AIP transfer format) is defined based on XML. In this model, the intellectual property issue is partially solved by restricting an end-user's access to the deposited resources on the destination side.

In the destination archive, migration of the deposited resource is not difficult because it is merely a text object such as an XML document. However, when migration occurs in the source archive, the resource needs to be re-enclosed and re-deposited in the destination archive.

Many archives are not as reliable as those maintained by large memory organizations, but their resources should be preserved as well as those of the large ones. The simplicity of the proposed model is a key to solving this issue.

Acknowledgements

This work is supported by JSPS Grants-in-Aid for Scientific Research (JSPS: Japan Society for the Promotion of Science).

References

1. LOCKSS Web Site, <http://lockss.stanford.edu/index.html>
2. Reference Model for an Open Archival Information System (OAIS). Blue Book. Issue 1. January 2002. This Recommendation has been adopted as ISO 14721:2003, <http://ssdoo.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf>
3. Robert Tansley, et al; DSpace as an Open Archival Information System: Current Status and Future Directions, ECDL2003, Lecture Notes in Computer Science 2769, pp446-460, 2003
4. Robert Tansley, et al; DSpace System Documentation, 2004, <http://dspace.org/technology/system-docs/>

デジタルアーカイブシステム間連携による
長期保存方式に関する研究

2005年3月

岡田 武

(筑波大学図書館情報メディア研究科修士論文より本人の許可を得て転載)

覚書

私、岡田武は田畑孝一氏との共同研究に基づき、執筆した下記論文の内容を科学研究費補助金（代表者：田畑孝一）の研究成果報告書に転載することを認めます。なお、同報告書がインターネットで出版されても意義ありません。

記

筑波大学大学院図書館情報メディア研究科修士論文
「デジタルアーカイブシステム間連携による長期保存に関する研究」
（岡田武：2005年3月）

2005年1月25日

田畑孝一殿

署名 岡田 武

目次

1.はじめに	1
2.デジタルコンテンツの長期保存	2
2.1 本研究で考えるデジタルコンテンツの長期保存	2
2.2 デジタルアーカイブシステムの紹介	3
3. OAIS(Open Archival Information System)参照モデル	5
3.1 OAIS 参照モデルの概要	5
3.2 OAIS 参照モデルの情報モデル	6
3.3 OAIS の機能エンティティ	9
3.4 OAIS 参照モデルの連携利用モデル	10
4 デジタルアーカイブシステム間連携によるデジタルコンテンツの長期保存	14
4.1 OAIS 参照モデルに基づいたデジタルアーカイブシステム間連携による 長期保存の概要	14
4.2 OAIS 参照モデルに基づいたデジタルアーカイブシステム間連携保存モデル	15
4.2.1 基本データモデル	15
4.2.2 コレクション単位でのデータモデル	20
4.2.3 預けから返却までの一連の流れ	23
4.2.4 連携保存モデル	25
4.2.5 環境モデル	29
5. プロトタイプシステムの開発	30
5.1 プロトタイプシステムの概要	30
5.1.1 プロトタイプシステムの開発目的	30
5.1.2 プロトタイプシステムの開発環境及び運用環境	30
5.1.3 プロトタイプシステムの構成	30
5.1.4 封入プログラム	32
5.1.5 開封プログラム	32
5.1.6 開示プログラム	33
5.1.7 返却編集プログラム及び預け入れ編集プログラム	34
5.2 プロトタイプシステムに用いるデジタルアーカイブシステム(DSpace)	34
5.2.1 DSpace の概要	34
5.2.2 DSpace の情報モデル	35
5.2.3 DSpace におけるコンテンツの提出及び取り出し	38
5.3 プロトタイプシステムの実装モデル	41

5.4 DSpace におけるデータ記述の例	49
5.5 プロトタイプシステムの運用結果	64
6.まとめ	65
7.謝辞	66
8.参考資料	67

1. はじめに

インターネットの爆発的普及により、デジタルコンテンツが世の中に溢れてきている。デジタルコンテンツは劣化がないことや処理が容易であることなど多くの長所を持っている。しかし、表示に固有のソフトウェアやハードウェアが必要であり記憶媒体の寿命も短い。そのため、紙媒体のコンテンツに比べ脆弱であり常に消失の危機にさらされている。そのような状況下において、大学や公共団体でもコンテンツの長期保存を主目的としたデジタルアーカイブシステムの導入の機運が高まっている。

デジタルアーカイブシステムの技術標準として OAIS 参照モデル (ISO 規格 14721) がある。OAIS 参照モデルは特定の実装を意図して策定されたものではなく、概念上の技術標準である。また、OAIS 参照モデルはコンテンツの長期保存に必要なコンセプトを提供している。

本研究では OAIS 参照モデルに基づいたデジタルアーカイブシステム間で互いに保存情報を預け合うことにより、デジタルコンテンツのより長期にわたる保存を図る方式を考究する。OAIS 参照モデルでは複数のデジタルアーカイブシステムを連携して情報を利用するモデルについては言及しているが、連携して情報を保存するモデルについては言及していない。そこで連携保存のモデルを提案し、実証するためのプロトタイプシステムを開発する。OAIS 参照モデルでは保存情報はコンテンツと保存メタデータから成っている。本研究では源システム側の内容情報と保存記述情報をカプセル化して、それを目的システム側のコンテンツとみなして預ける方法を提案した。源システムから目的システムへ保存情報を預け、目的システムから源システムへ保存情報を返却する一連の流れのモデルのみならず、カプセル化された保存情報を覗き見るモデルについても考究している。さらに、提案したモデルをプロトタイプシステムを用いて実証した。

2. デジタルコンテンツの長期保存

2.1 本研究で考えるデジタルコンテンツの長期保存

デジタルコンテンツの長期保存を考える上で記憶媒体の長期保存とコンテンツのメタデータを如何にコンテンツと共に保存していくかということが重要となる。ここでいうコンテンツのメタデータとはそのコンテンツを表示するためのソフトウェア環境やハードウェア環境、コンテンツの出所などを指す。コンテンツを保存している記憶媒体を長期保存してもそのコンテンツの表示方法がわからなくては意味がない。

本研究では以上のような長期保存への取り組みがなされているデジタルアーカイブシステムが存在していることを仮定する。具体的には、デジタルアーカイブシステムの国際標準である OAIS 参照モデルに基づいたデジタルアーカイブシステムが存在していることを仮定する。ここで、デジタルアーカイブシステムとはデジタルコンテンツを収集・保管・提供するためのシステムである。OAIS 参照モデルに基づいたデジタルアーカイブシステムへ格納されているコンテンツは長期保存の対象となっており、上で述べたメタデータの付与に関しても策を講じている。本研究ではさらにコンテンツの長期保存性を高めるためにデジタルアーカイブシステム間で保存情報を預けあうことを考えた。OAIS 参照モデルではデジタルアーカイブシステム間で連携して保存情報を利用するモデルについては言及しているが、連携して保存するモデルについては言及していない。そこで本研究では連携して保存するモデルを提案し、実証するためのプロトタイプシステムを開発する。複数のコミュニティで連携して情報を保存することが保存情報の長期保存性を高めることは自明であり、将来預けた側のシステムが何らかの理由でなくなった際、預かった側のシステムが預かりの保存情報を自システム内の保存情報として配布することも可能である。

本研究は、あるシステムが保存情報を他のシステムに預け、必要なときにそれを返却してもらうという一連の流れを OAIS 参照モデルに基づいて考える。預けた情報を返却してもらうということを考えてはいるが、これはバックアップの枠を超えていない。本研究でいう長期保存とは、遙か未来に、預かった側で発見、解読され再び世の中に出て行くようなことをいう。そのため、如何にして保存情報を相手側システムに預けるか、預けた保存情報が遙か未来に相手側で発見されたときに解読可能であるかどうかという点が重要であり、預けた保存情報をいかにして返却してもらうかはさほど重要ではない。そのため、預ける側がどこに何を預けたかを記憶しておくことはない。仮にそのような情報を長期保存の対象としてもシステムが壊れた際にはその情報自体も消えてしまう。その場合にも預かった側ではどこからの預かり物かを記述しているため、それを頼りに返却してもらうことは可能である。以上の事からも“どこに何を預けたかという情報”を長期保存する必要はない事が言える。さらに本研究では大きなコミュニティ間での連携を対象としているため、連携状況が複雑になることはなく“どこに何を預けたかという情報”はシステム管理者の記憶やメモ書き等で十分に事が足りると考えられる。そのような情報が消失した遙か未来に、預けた保存情報が預け先で発見され解読されることが本研究でいう長期保存である。また近い将来にシステムが故障

し保持していた保存情報が預け先にしかない状態になったとしても、預けた保存情報を返却してもらえば、再び保存情報の所在場所が増え、遙か未来にその保存情報が発見される可能性も高まるであろう。そういった観点から考えれば預けた保存情報を返却してもらうことは保存情報を預ける事と同じように長期保存性を高めることになる。

2.2 デジタルアーカイブシステムの紹介

デジタルアーカイブシステムは数多く存在しており、それぞれに様々な特徴がある。本節では代表的なデジタルアーカイブシステムを紹介する。紹介するデジタルアーカイブシステムは全てオープンソースソフトウェアであり、無料で利用できるものである。

CERN Document Server Software(CDSware)[2][3]

CDSware は CERN Document Server をサポートするために開発され、CERN(注 2.1)により提供されている。マルチメディアコンテンツ目録や博物館収蔵品目録、機密文書などの種類の異なる資料を所蔵する大規模なシステム向けに開発された。

DSpace[2][4][13][14]

DSpace はマサチューセッツ工科大学とヒューレットパッカード社が共同開発したシステムであり、研究組織の知的生成物を保存するためのシステムとして開発された。デジタルコンテンツの長期保存を視野に入れたシステムであり、OAIS 参照モデル (デジタルアーカイブシステムの技術標準) の機能によく対応している。

Eprints[2][5]

Eprints はサウサンプトン大学で開発されたシステムである。インストール及び改良が容易であり、世界中で利用されているシステムである。

i-Tor[2][6]

i-Tor(Tools and technologies for Open Repositories)はオランダ科学情報サービス研究所で開発されたシステムである。i-Tor はデータに依存しないシステムの構築を目的としており、コンテンツとユーザーインターフェース機能をシステム上で独立している。

MyCoRe[2][7]

MyCoRe(My Content Repositories)はエッセン大学の MILESS プロジェクトから派生し、現在は複数の大学からなるコンソーシアムにおいて開発中である。MyCoRe はデジタルライブラリの運営を支援するツールを提供するものであり、各機関の要件に合わせて設定を変

更できるシステムである。

GreenStone[2][8]

GreenStone はワイカト大学のニュージーランドデジタルライブラリプロジェクトにおいて開発されたシステムである。GreenStone は膨大な数の文書や音声、映像などからなるコレクションを構築、提供するシステムである。

(注 2.1)

CERN : Conseil Européen pour la Recherche Nucléaire (欧州合同素粒子原子核研究機構) の略。現在の名称は Laboratoire Europeen pour la Physique des Particules であるが呼称は CERN (セルン) のままである。

3. OAIS(Open Archival Information System)参照モデル

[1][13][15]

3.1 OAIS 参照モデルの概要

OAIS 参照モデルは宇宙データシステム諮問委員会（CCSDS : Consultative Committee for Space Data Systems)により制定されたデジタルアーカイブシステムの国際標準(ISO規格 14721)である。ここで、OAIS とは「人とシステムの一組織で構成される、指定されたコミュニティのために情報を保存し、それを利用できるようにする責任を負ったアーカイブシステム」である。

OAIS 参照モデルは、衛星から送られてくるデジタルデータを長期保存する目的で検討が開始されたが、最終的にはあらゆるコンテンツに対応したものとなっている。また、OAIS 参照モデルは Web サイト等で公開されており、幅広く利用されることを意図している。

OAIS 参照モデルはコンテンツの長期保存に必要なコンセプトと実際の保存活動に適用するためのフレームワークを提供する。OAIS は以下の要件を満たしたものとする。

- ・ 情報提供者と交渉し、情報を入手する。
- ・ 長期保存を保証する十分な管理機能を持つ。
- ・ 保存情報を理解できる指定コミュニティ(注 3.1)を確保する。
- ・ 指定コミュニティが保存された情報を専門家の力を借りずに理解できる。
- ・ 保存されている情報を保障するための方針書や手続き書を作成し、それに従う。
- ・ 指定コミュニティが保存された情報を理解できる形式で入手できる。

3.2 OAIS 参照モデルの情報モデル

OAIS 参照モデルでは情報を図 3.1 のように内容情報と保存記述情報（保存メタデータ）のパッケージとして扱う。

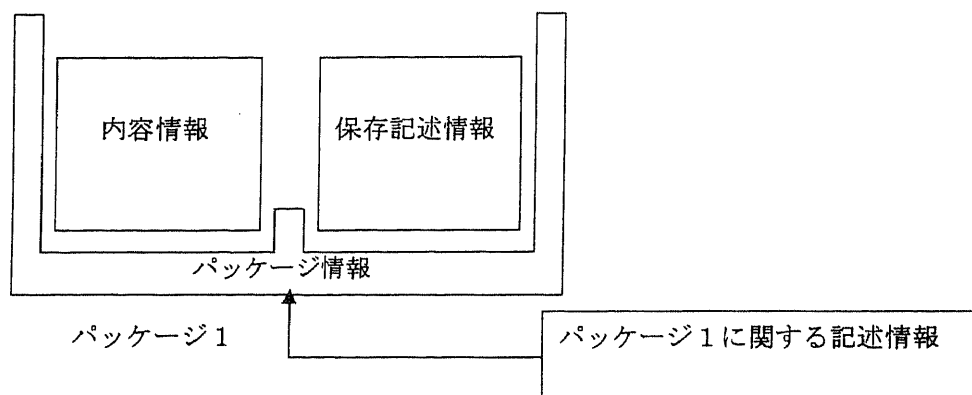


図 3.1 情報パッケージ

内容情報の分類を図 3.2 に示し、保存記述情報の分類を図 3.3 に示す。内容情報は内容データオブジェクトと表現情報に分けられる。内容データオブジェクトはコンテンツのビット列であり、内容データオブジェクトを理解する手段が表現情報である。表現情報の詳細について OAIS 参照モデルでは言及していないが、OCLC(オンラインコンピューター図書館センター)と RLG のワーキンググループが作成した A Metadata Framework to Support the Preservation of Digital Objects 内で細かく定義・分類している[9]。A Metadata Framework to Support the Preservation of Digital Objects では表現情報を図 3.4 のように分類している。表現情報は内容データオブジェクト記述と環境記述に分けられ、環境記述はソフトウェア環境とハードウェア環境に分けられる。内容データオブジェクト記述は内容データオブジェクトを表示・理解するために必要な特徴を表す情報であり、環境記述は内容データオブジェクトを正しく表示するために必要なソフトウェア環境及びハードウェア環境に関する記述である。保存記述情報は来歴、コンテキスト、参照、固定性の4つに分けられる(図 3.3 参照)。来歴は内容情報の由来、コンテキストは他の情報との関係、参照は内容情報を同定するための ID 情報、固定性は内容情報が変更されていないことを示す情報である。

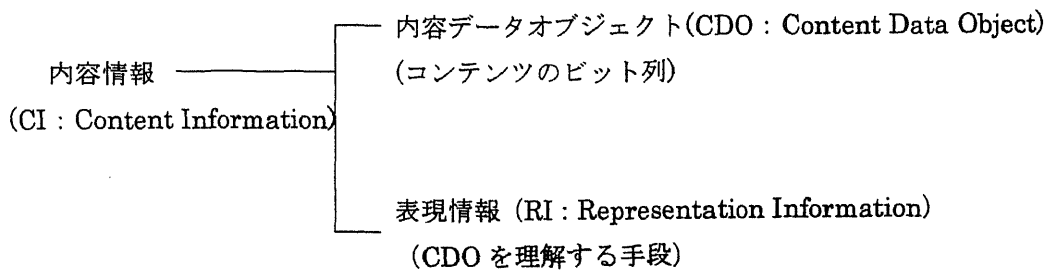


図 3.2 内容情報の分類

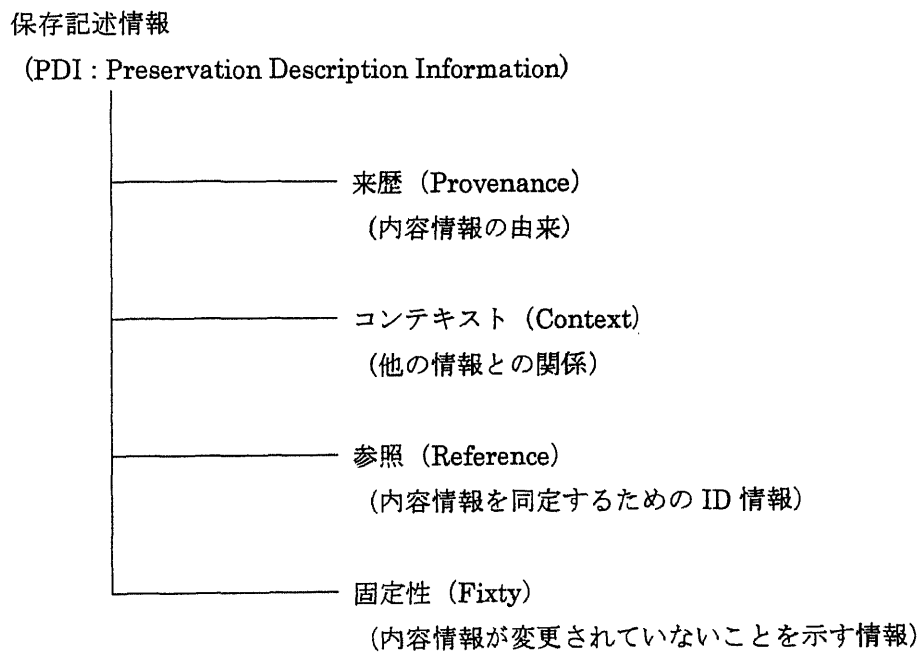


図 3.3 保存記述情報の分類

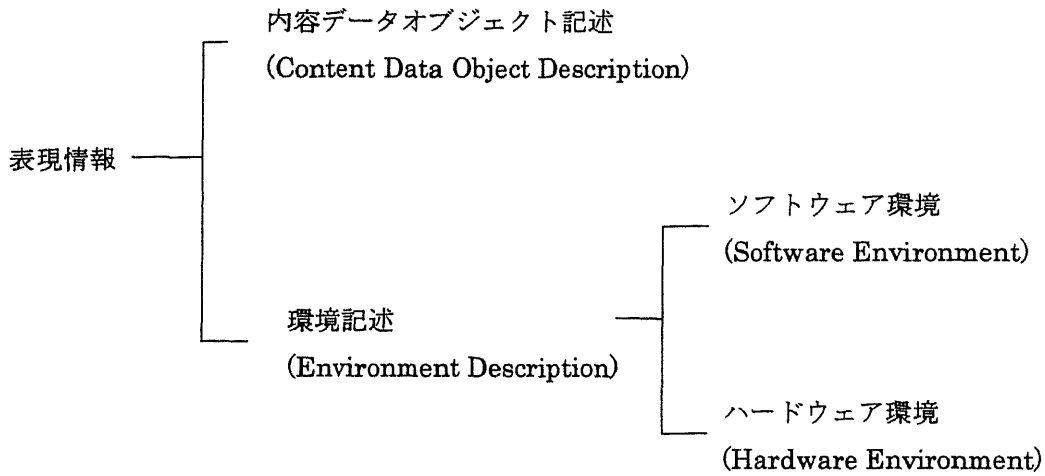
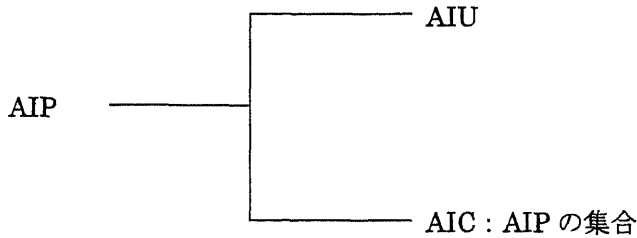


図 3.4 表現情報の分類[9]

OAIS 参照モデルでは保存情報を AIP(Archival Information Package)と呼ぶ。AIP には図 3.5 のように AIU と AIC の 2 つの型がある。AIU とは Archival Information Unit の略でありそれ以上分けることの出来ない保存情報の単位である。AIC とは Archival Information Collection の略であり、AIP の集合である。



- AIP : Archival Information Package
- AIU : Archival Information Unit
- AIC : Archival Information Collection

図 3.5 AIP の型

OAIS 参照モデルでは AIC はコレクションの要旨と構成要素である AIU へのリンク情報もしくは AIU の実体が格納される。

3.3 OAIS の機能エンティティ

図 3.6 に OAIS の機能エンティティを示す。図中の各要素について説明する。まず SIP、AIP、DIP について説明する。AIP は前節で説明した Archival Information Package の略であり、保存用の情報パッケージである。SIP は Submission Information Package の略であり、提出用の情報パッケージである。また、DIP は Dissemination Information Package の略であり、配布用の情報パッケージである。SIP は Producer によって送られる情報パッケージで、1 つ以上の AIP を生成する。また、DIP は 1 つ以上の AIP から取り出される情報パッケージで、Consumer が OAIS に要求して受け取る。

Producer は SIP を Ingest 機能に提出する。Ingest 機能は Descriptive Information を Data Management 機能へ、AIP を Archival Storage 機能へ送る。ここで Descriptive Information とは AIP を検索するための情報で AIP から抽出される。Consumer から Access 機能へ問い合わせがあれば、Access 機能は Data Management 機能から Descriptive Information を受け取り AIP の検索を行い、Consumer へ結果を返す。さらに Consumer から注文があれば Access 機能は Archival Storage 機能から AIP を受け取り、Consumer へ DIP として提供する。

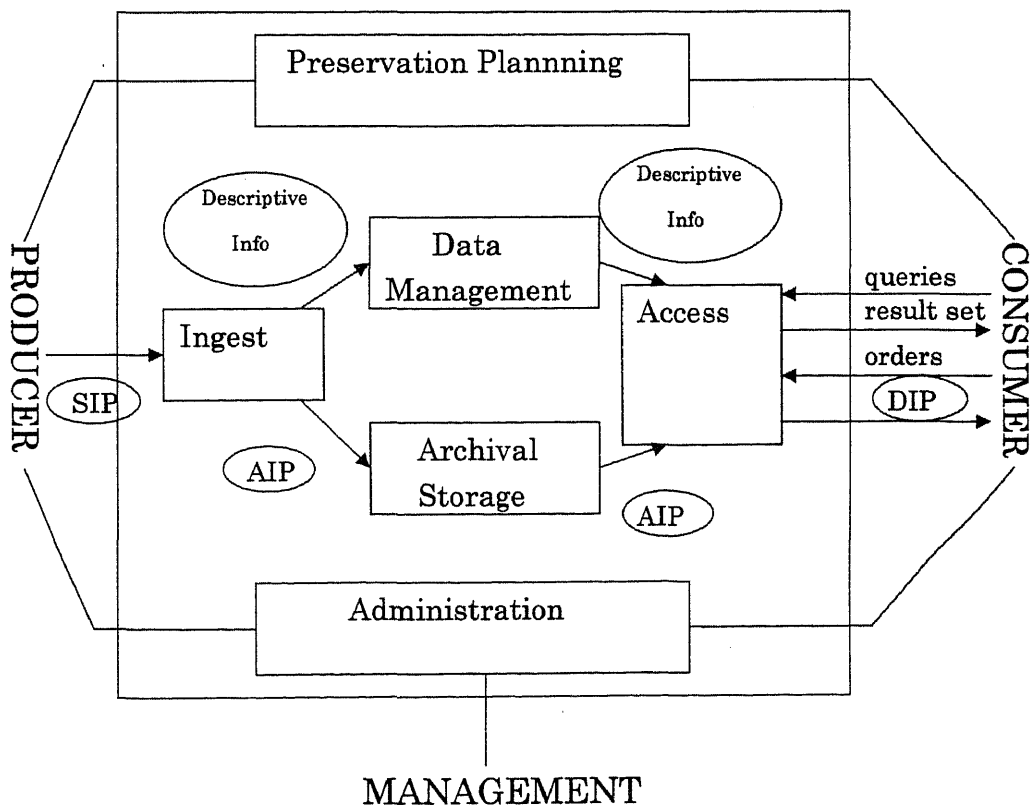


図 3.6 OAIS の機能エンティティ(1)

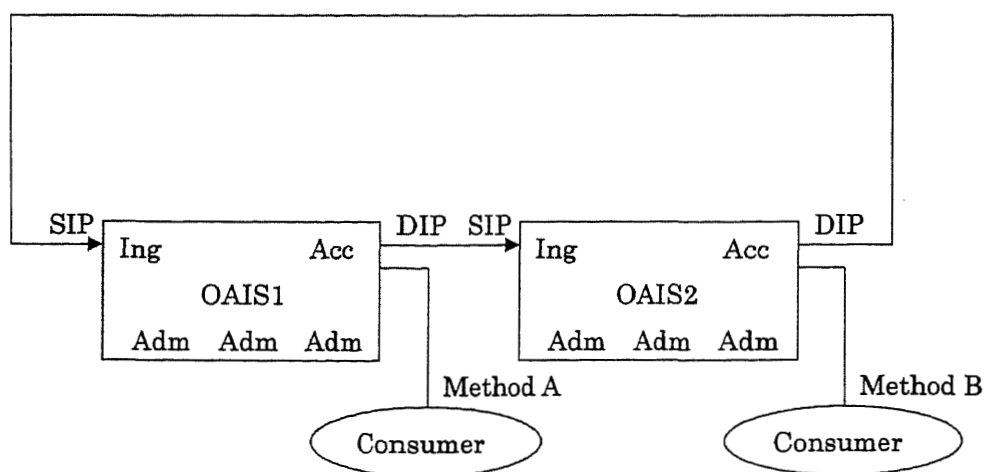
OAIS の機能エンティティにおける各機能の概要を図 3.7 に示す。

名称	機能の概要
Ingest	Producer から SIP を受け取り、保存や管理の準備を行う
Archival Storage	AIP の保管を行うためのサービスや機能を提供する
Data Management	Descriptive Information を保管・アクセスするためのサービスや機能を提供する
Access	Consumer が欲しい情報を検索し、その情報を受け取るためのサービスや機能を提供する
Administration	OAIS 全体の運用に関わるサービスと機能を提供する
Preservation Planning	長年に渡って AIP へアクセスできるようにするためのサービスと機能を提供する

図 3.7 OAIS エンティティにおける各機能の概要([13])

3.4 OAIS 参照モデルの連携利用モデル

OAIS 参照モデルではデジタルアーカイブシステムの連携保存については言及していないが連携利用については幾つかのモデルを提供している。本節では連携利用モデルを幾つか紹介する。



※一部筆者が追記している

図 3.8 相互に保存情報を交換する連携利用モデル([1])

図 3.8 は相互に保存情報を交換する連携利用モデルである。O AIS1 と O AIS2 があり、各 O AIS には Consumer がそれぞれ別のアクセス方法でアクセスしている。ここで、O AIS2 の Consumer が O AIS1 の AIP を入手したいときを考える。O AIS2 の Consumer は O AIS1 に直接アクセスするのではなく、O AIS2 の管理者が O AIS1 へアクセスして必要な保存情報を受け取り、その保存情報を O AIS2 の Consumer は受け取る。図 3.8 では、O AIS は 5 ポートデバイスとして表されており、各ポートの機能は次のとおりである。

Ing : Ingest (図 3.7 参照)

Adm : Administration (図 3.7 参照)

Acc : Access (図 3.7 参照)

図 3.9 は受入とアクセスの方法が標準化された連携利用モデルを示している。O AIS1 と O AIS2 の受入方法とアクセス方法を標準化することで Producer は O AIS1 と O AIS2 のいずれのシステムへも同じ方法で SIP を提出でき、Consumer は O AIS1 と O AIS2 のいずれのシステムからも同じ方法で DIP を受け取れる。

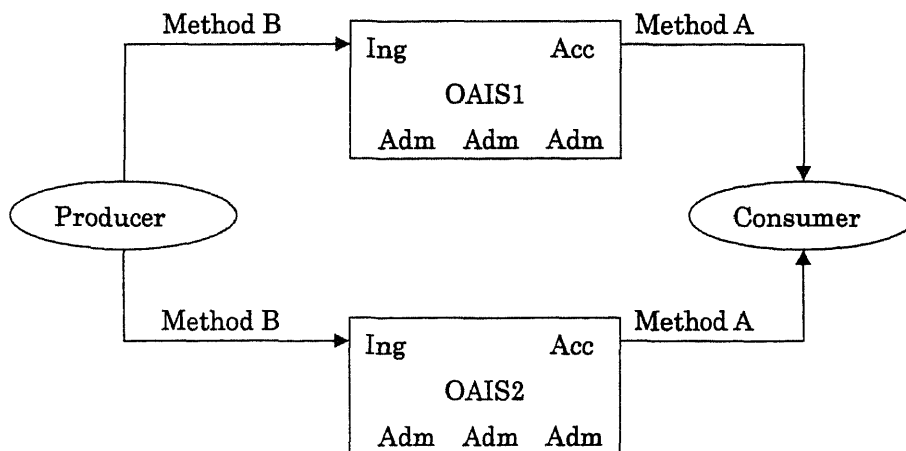
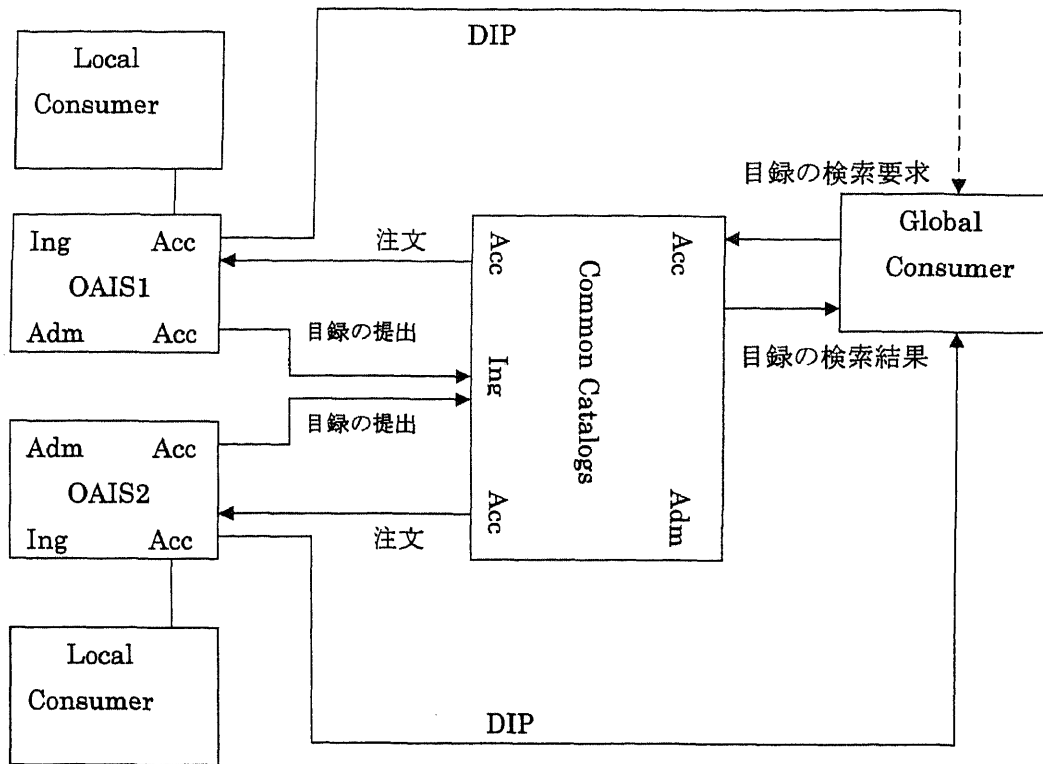


図 3.9 受入とアクセスの方法が標準化された連携利用モデル([1])

図 3.10 は共通の目録を使う連携利用モデルを示している。O AIS1 と O AIS2 にはそれぞれ Local Consumer (自身の属するコミュニティの O AIS にのみアクセスする Consumer) が存在している。一方 Global Consumer(自身の属さないコミュニティの O AIS にもアクセスする Consumer)が存在しており、Global Consumer は共通目録へアクセスすることにより欲しい AIP を検索し、各 O AIS から DIP として受け取ることが出来る。まず、各 O AIS

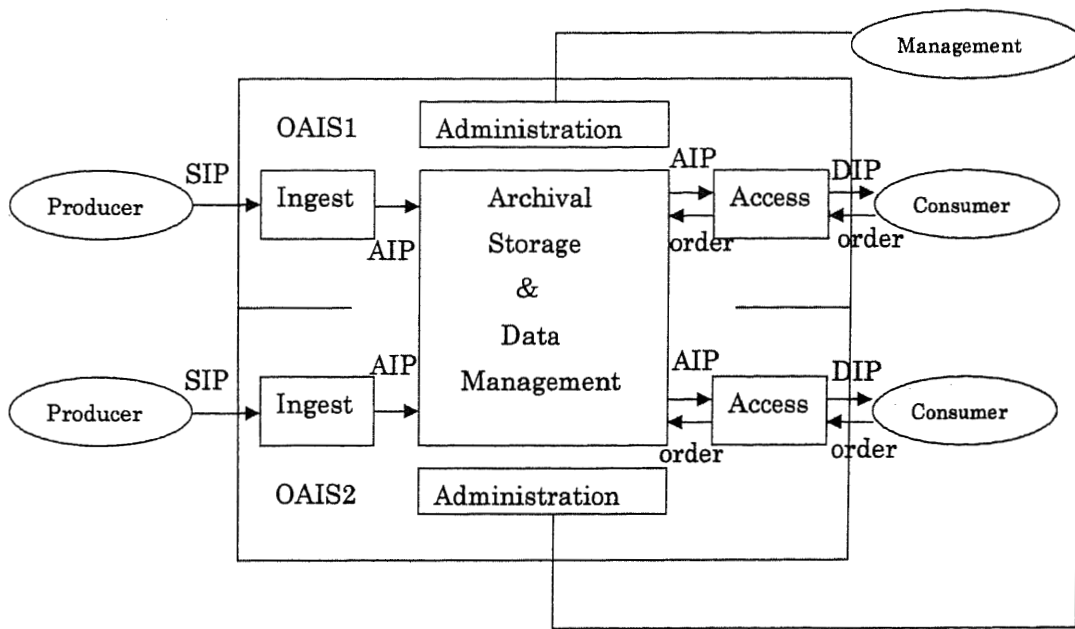
は共通目録へ Global Consumer へ提供するコンテンツの目録を提出する。Global Consumer は共通目録へアクセスして目録の検索を行う。検索の結果、欲しいコンテンツがあれば共通目録を通して各 OAIS へコンテンツを注文する。そして、各 OAIS から直接コンテンツを受け取る。



※一部筆者が追記している

図 3.10 共通の目録を使う連携利用モデル([1])

図 3.11 に保存場所を共有する連携利用モデルを示す。OAIS1 と OAIS2 の各 Producer が提出した SIP は各 OAIS の Ingest 機能を通して OAIS1 と OAIS2 共通の Archival Storage へ AIP として送られる。さらに OAIS1 と OAIS2 の各 Consumer は各 OAIS の Access 機能を通して共通の Archival Storage から DIP を受け取る。



※一部筆者が追記している

図 3.11 保存場所を共有する連携利用モデル([1])

(注 3.1)

指定コミュニティ：ある特定の情報の集合を理解できる Consumer のグループ

4 デジタルアーカイブシステム間連携によるデジタルコンテンツの長期保存

4.1 OAIS 参照モデルに基づいたデジタルアーカイブシステム間連携による長期保存の概要

本研究ではデジタルコンテンツの長期保存についてデジタルアーカイブシステム間連携の観点から検討する。具体的には、複数のデジタルアーカイブシステム間で互いに保存情報を預け合うことにより、デジタルコンテンツのより長期にわたる保存を図っていく。

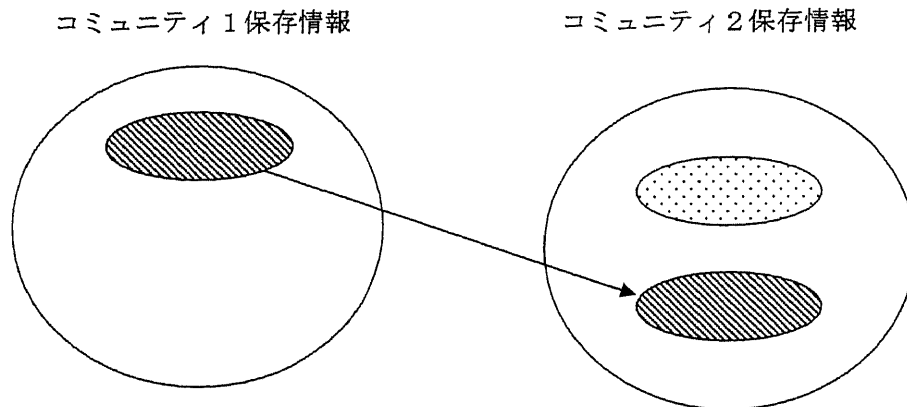


図 4.1 本研究で考えるアーカイブシステム間連携の概念図

図 4.1 は本研究で考えるアーカイブシステム間連携の概念図である。コミュニティ 1 とコミュニティ 2 が存在し、各コミュニティでは自身の保持する保存情報の長期保存を図っている。ここでコミュニティ 1 の保持する保存情報で特に大切な情報をコミュニティ 2 に預かってもらうことを考える。コミュニティ 2 では預かった情報も自身の保持する情報と同様に長期保存を図る。ただし、コミュニティ 1 から依頼されない限り預かりの情報を配布することはない。

次に保存情報の預け方について説明する。図 4.2 は本研究で提案する保存情報の預け方のモデルである。OAIS 参照モデルでは保存情報を内容情報と保存記述情報のパッケージとして扱っており、本研究では図 4.2 のように源システム側の内容情報と保存記述情報をカプセル化して目的システム側の内容データオブジェクトとして預ける方法を提案する。目的システム側ではカプセル化された内容データオブジェクトに対して表現情報と保存記述情報を記載し、自身の保存情報と同様に保存していく。ここで目的システム側では預かった保存情報に対して以下の保存記述情報を記述する。ただし、コンテキストに関しては、特に記述すべき事項はない。

参照：	源 OAIS 名、シリアルナンバー
来歴：	どのシステムからいつ預かったか
コンテキスト：	
固定性：	自システム内で変化のないこと

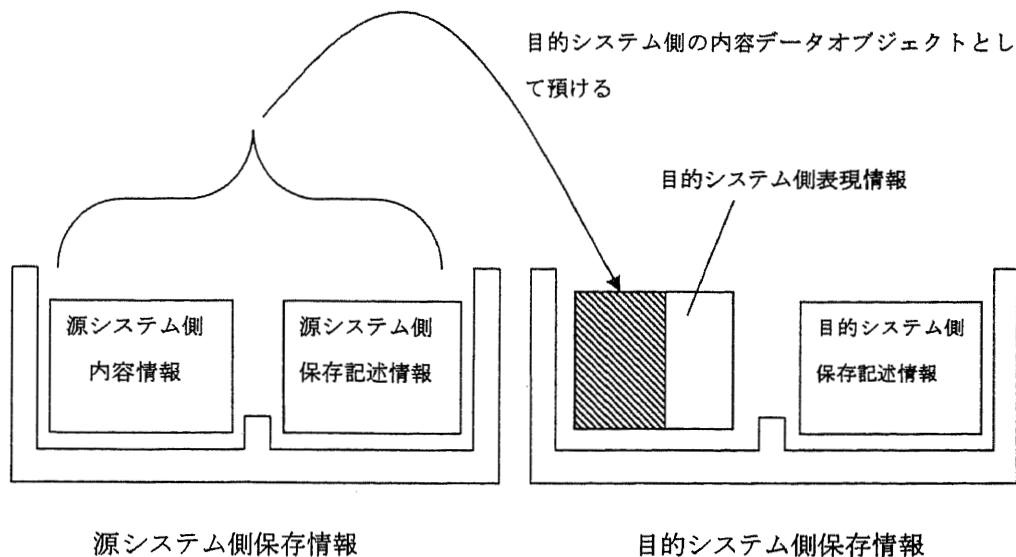


図 4.2 保存情報の預け方モデル

4.2 OAIS 参照モデルに基づいたデジタルアーカイブシステム間連携保存モデル

4.2.1 基本データモデル

ここでは預ける際の保存情報のカプセル化を如何にして実現するかをデータ表現により説明する。まず、源システムで預ける対象となる保存情報のカプセル化を行うが、本研究ではこれに XML 技法を用いることにした。具体的には、XML 文書内に内容情報と保存記述情報を格納することでカプセル化を図る。ここで XML 文書ではビット列を扱えないので、コンテンツをビット列から文字列に変換してから XML 文書内に格納する。ただし、ビット列から文字列への変換に用いた手法を必ず記述する。カプセル化した保存情報のデータ表現のモデルを図 4.3 に示す。図 4.3 のデータ表現モデルではビット列から文字列への変換に base64Binary を採用した場合を示しており、図中の略称と OAIS 参照モデルにおける情報モデルの要素との対応を以下に示す。また、図 4.3 をツリー表現すると図 4.4 のようになる。

- AIP : Archival Information Package
- CI : Content Information
- CDO : Content Data Object
- RI : Representation Information
- CDD : Content Data Object Description
- ED : Environment Description
- SE : Software Environment

HE : Hardware Environment
PDI : Preservation Descriptive Information
RF : Reference
PV : Provenance
CT : Context
FX : Fixty

```
<AIP>  
<CI>  
<CDO encode="base64Binary">  
abcdef . . . . .  
</CDO>  
<RI>  
<CDD>xxxxxx</CDD>  
<ED>  
<SE>aaaa</SE>  
<HE>bbbb</HE>  
</ED>  
</RI>  
</CI>  
<PDI>  
<RF>cccc</RF>  
<PV>dddd</PV>  
<CT>eeee</CT>  
<FX>fff</FX>  
</PDI>  
</AIP>
```

図 4.3 源システム側でカプセル化した保存情報のデータ形式

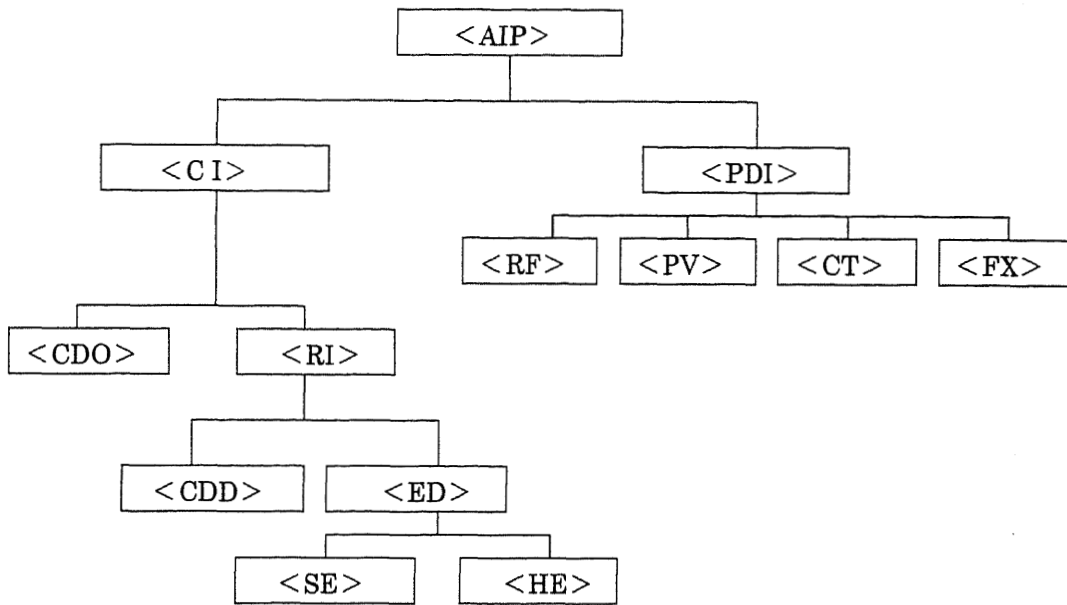


図 4.4 源システム側でカプセル化した保存情報のツリー型データ表現

次に、1つのコンテンツが複数のファイルから構成されている場合を考える。html ファイルの中に JPEG ファイルがリンクされている場合など1つのコンテンツが複数のファイルから構成されている事が多くある。複数ファイルからコンテンツが構成されている場合のデータ表現を図 4.5 に示す。また、そのツリー表現を図 4.6 に示す。

源システムがカプセル化された保存情報を目的システムへ送ると目的システムはカプセル化された保存情報を内容データオブジェクトとして扱う。目的システムでは預かった保存情報(目的システム側内容データオブジェクト)に表現情報を付与し、内容情報を形成する。さらに、その内容情報へ保存記述情報を付けて保管する(図 4.7 参照)。

```

<AIP>
<CI>
<elementofCI>
<CDO encode="base64Binary">
abcdef.....
</CDO>
<RI>
<CDD>test</CDD>
<ED>
<SE>aaaa</SE>
  
```



```

<HE>bbbb</HE>
</ED>
</RI>
</elementofCI>
<elementofCI>
<CDO encode="base64Binary">
opqrst . . . . .
</CDO>
<RI>
<CDD> testtest</CDD>
<ED>
<SE>aaaa</SE>
<HE>bbbb</HE>
</ED>
</RI>
</elementofCI>

. . . . .

</CI>
<PDI>
<RF>cccc</RF>
<PV>dddd</PV>
<CT>eeee</CT>
<FX>ffff</FX>
</PDI>
</AIP>

```

図 4.5 源システム側でカプセル化した保存情報のデータ形式 (コンテンツが複数ファイルからなる場合)

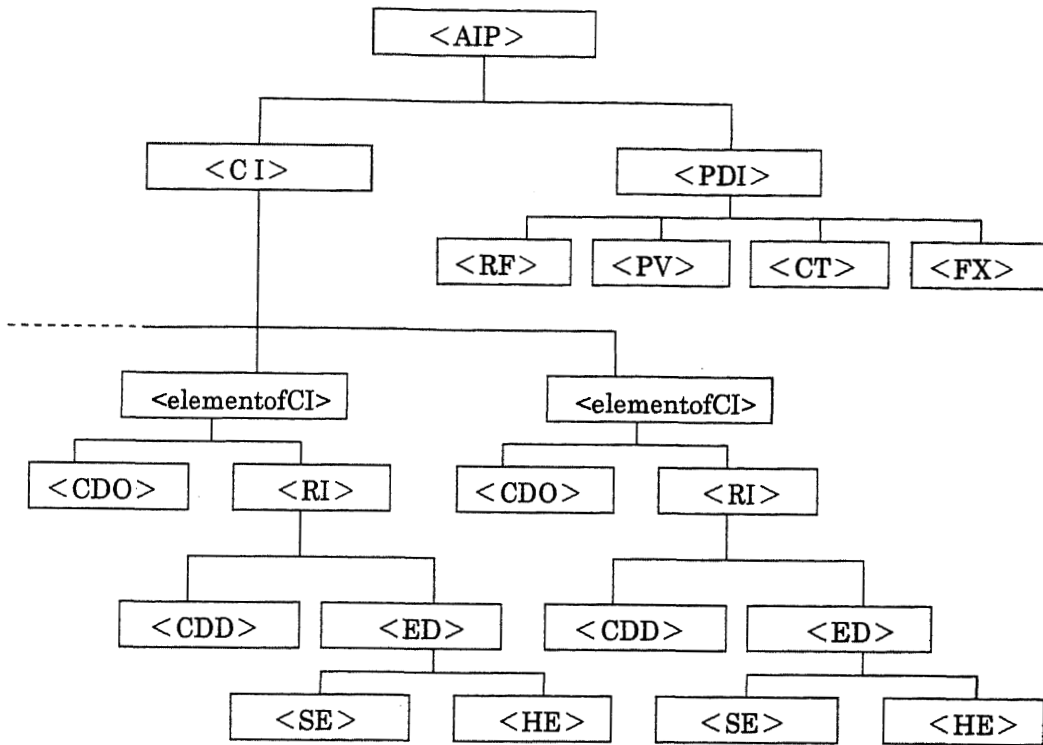


図 4.6 源システム側でカプセル化した保存情報のツリー型データ表現

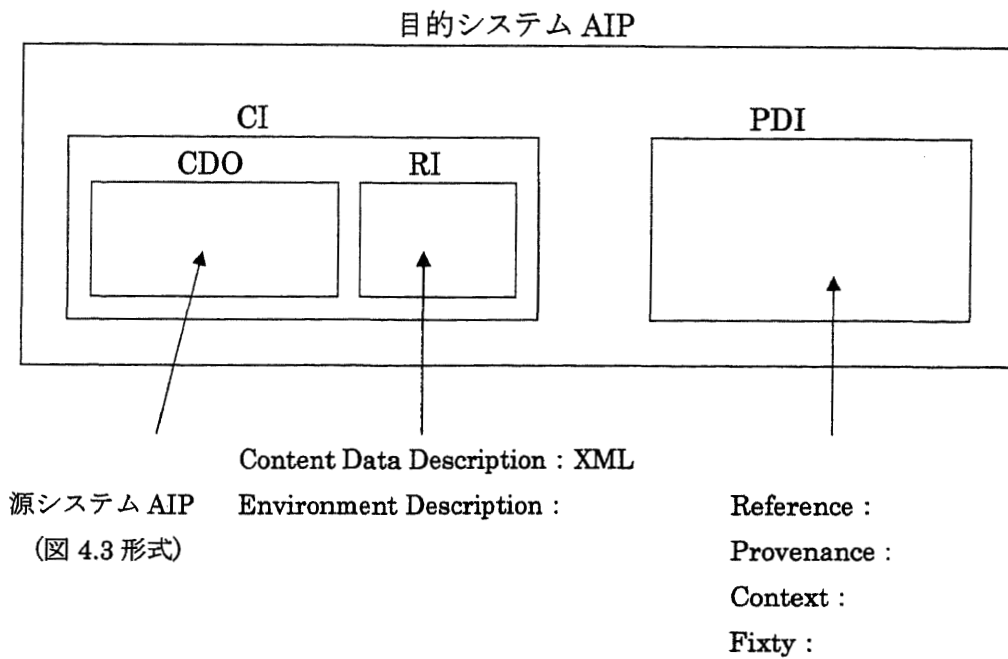


図 4.7 目的システム側 AIP データ表現

4.2.2 コレクション単位でのデータモデル

保存情報をコレクション単位で預ける際のデータモデルについて述べる。単一で保存情報を預けるケースも考えられるが、実際にはコレクション単位で預けるケースのほうが一般的であると考えられるため、本研究で提案する連携保存モデルはコレクション単位とする。

まず保存情報のカプセル化を行うがここで1つ注意が必要である。それは、1つのコレクションに対して1つのXML文書が作成されるのではなく、1つの保存情報に対して1つのXML文書が作成されるということである。また、目的システム側ではカプセル化された源システム側保存情報（目的システム側内容データオブジェクト）に対して表現情報と保存記述情報を記載し、自身の保存情報と同様に保管していく。

コレクション単位で預ける際には、通常AICのCDOにコレクションの要素であるAIUのリストを記載することでコレクションの親子関係を維持する。AICとAIUの親子関係を考慮に入れた目的システム側での源システムAIU及びAICのデータ表現を図4.8及び図4.9を示す。

本研究ではコレクションの親子関係を維持する方法として子が親の名前を記載しておく方法を採用する（図4.10参照）。つまり、各AIUがどのコレクションに属しているかを記載しておくことでAIUがばらばらになるのを予防する。AICのPDIにコレクション名を記述しておき、各AIUのPDIにそのコレクション名を記述しておく。但し、これは連携保存モデルにおける絶対的な要件ではなく、各システムが任意で行う事である。またOAIS参照モデルでもこの内容については記述されていない。

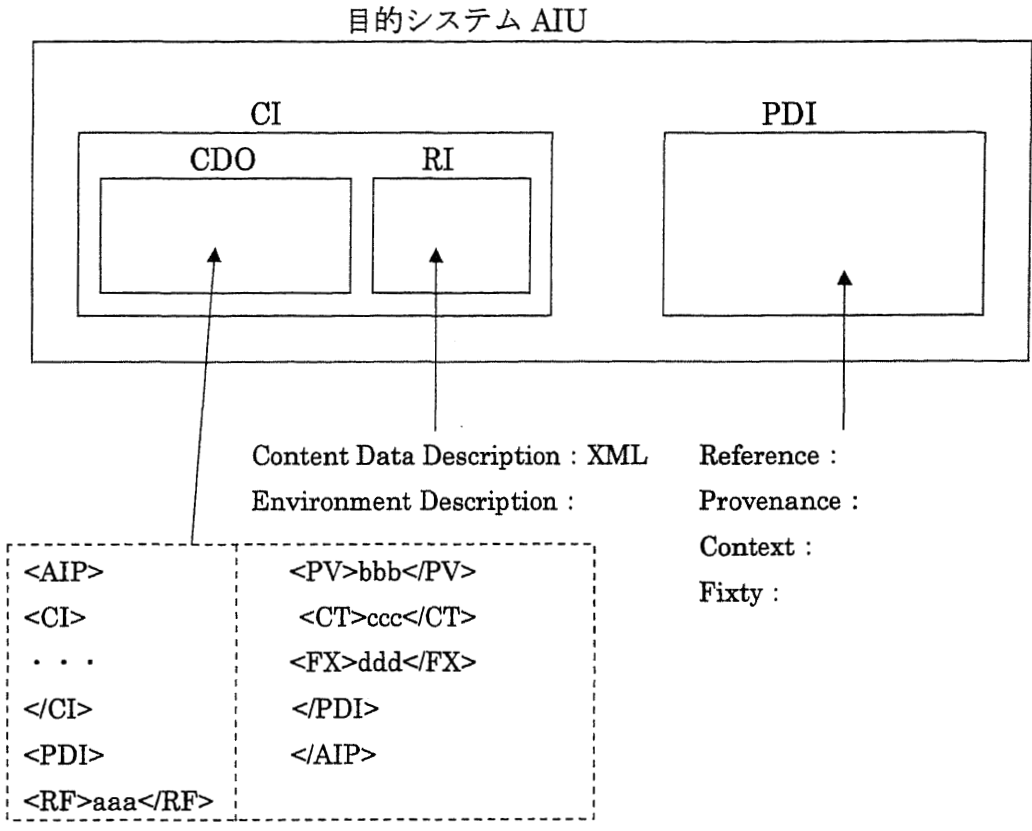


図 4.8 目的システム側での源システム AIU のデータ表現

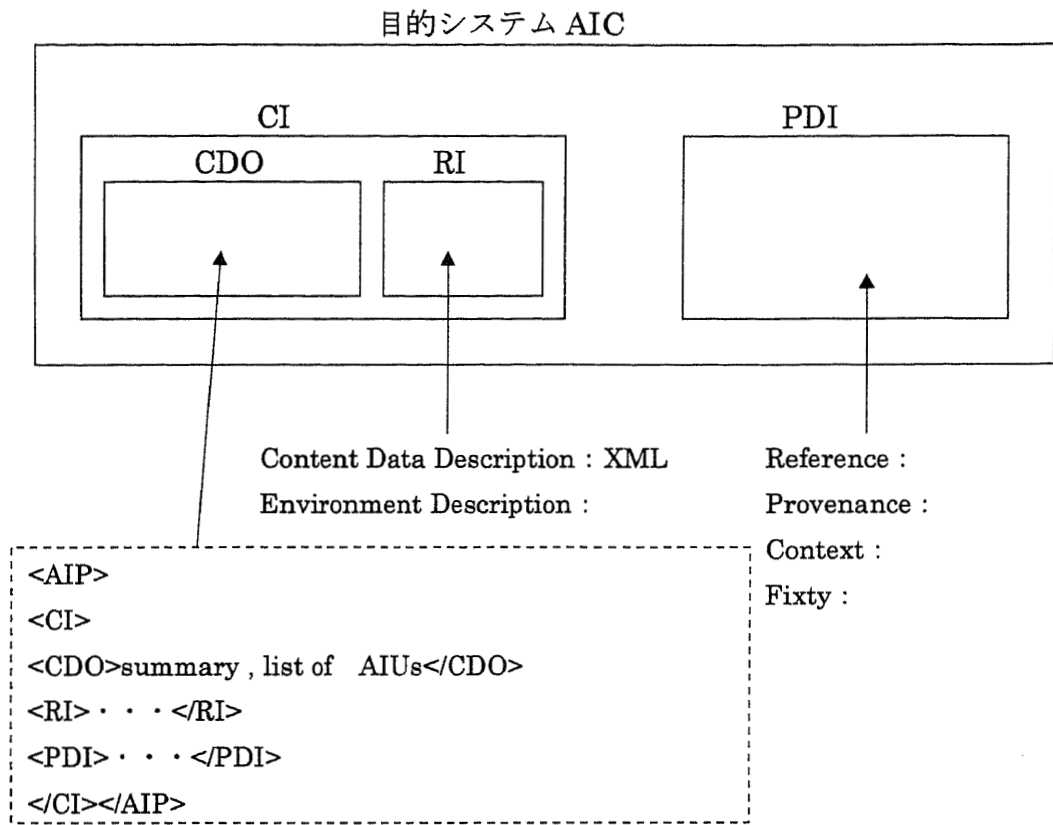


図 4.9 目的システム側での源システム AIC のデータ表現

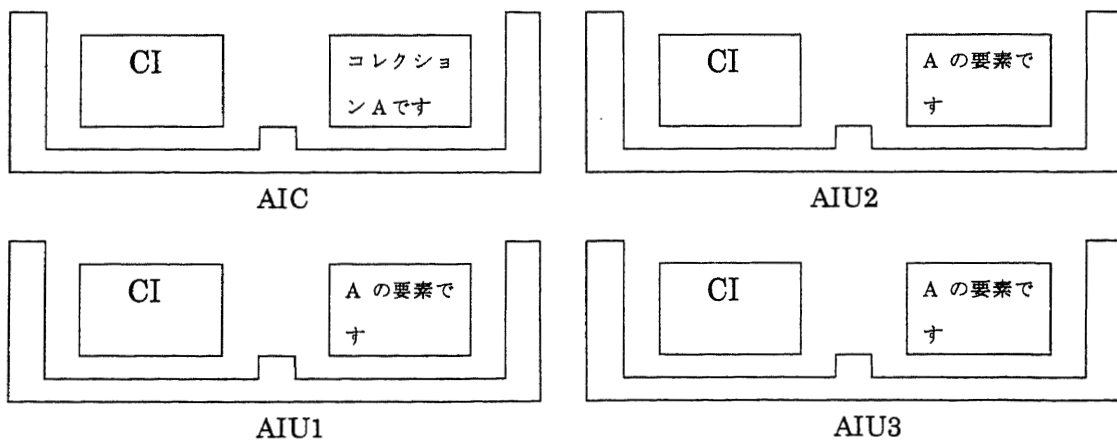


図 4.10 子が親の情報を持つ場合

4.2.3 預けから返却までの一連の流れ

ここでは預けから返却までの一連の流れについて説明する。預け及び返却の一連の流れをそれぞれ図 4.11,4.12 に示す。まず源システム側で預けたい保存情報（図 4.11 中 A-1）をデジタルアーカイブシステムから取り出し（図 4.11 中 A-2）、封入作業（カプセル化）を行う。そしてカプセル化された保存情報（図 4.11 中 B-1）は目的システム側へ送られる（図 4.11 中 B-2）。目的システム側では、カプセル化された保存情報に対して表現情報と保存記述情報を付け（図 4.11 中 C-1）、デジタルアーカイブシステムに格納する（図 4.11 中 C-2）。その後、源システム側から預けた保存情報を返却して欲しいとの依頼があれば預かっている保存情報（図 4.12 中 C-3）を源システム側へ返却する。預かっている保存情報をデジタルアーカイブシステムから取り出し（図 4.12 中 C-4）、目的システム側で返却前に目的システム側表現情報と目的システム側保存記述情報を削除しておく（図 4.12 中 B-3）。源システム側では返却された保存情報（図中 B-4）の開封作業（カプセル化された保存情報を元の状態に戻す）を行い（図 4.12 中 A-3）、デジタルアーカイブシステムに格納する（図 4.12 中 A-4）。

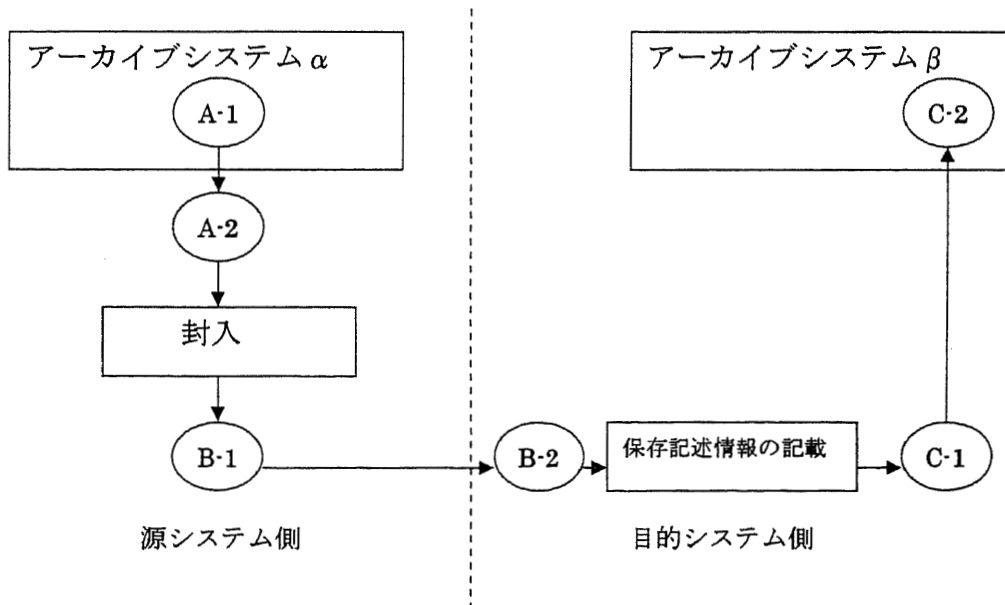


図 4.11 預け作業の流れ

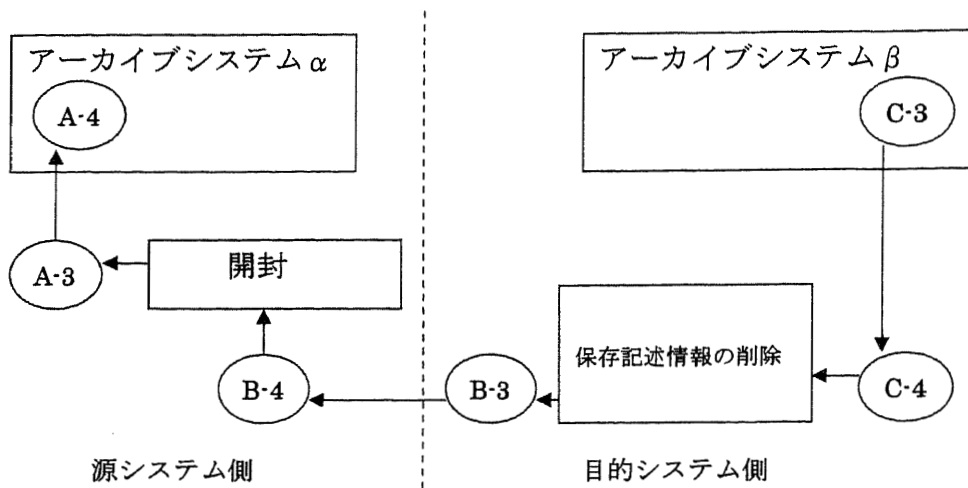
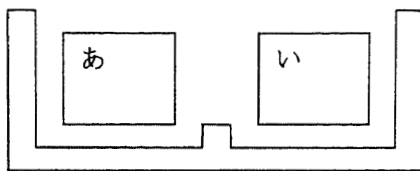
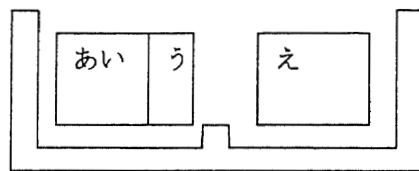


図 4.12 返却作業の流れ

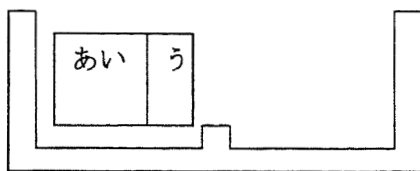
※図 4.11、図 4.12 において A-1=A-2=A-3=A-4、B-1=B-2=B-3=B-4、C-1=C-2=C-3=C-4 が成り立つ



A-1~A-4



C-1~C-4



B-1~B-4

- あ：源システム側内容情報
- い：源システム側保存記述情報
- う：カプセル化コンテンツに対する
目的システム側表現情報
- え：カプセル化コンテンツに対する
目的システム側保存記述情報

4.2.4 連携保存モデル

本研究で提案する連携保存モデルについて説明する。4.2.3 では単一保存情報での預けから預かりの一連の流れを説明した。しかし、実際には開封方法を保存情報と共に目的システム側へ預け、返却の際に預けておいた開封方法を用いて開封を行うことになるため、単一保存情報のみで預けることはない。また、実用性を考えると、単一ではなく複数の保存情報を一括して預けるケースのほうが主流であると考えられる。これら要因から連携保存モデルにおける預け及び返却の単位をコレクションとする。本研究の連携保存モデルは預けから返却までの一連の作業をコレクション単位で一括して処理するモデルとする。しかし、これは実際のシステムで一括処理ができないと預け及び返却が実現出来ないという意味ではなく、少数の保存情報では1つずつ処理しても構わない。

また、開封方法とは開封方法を文書で記述したものや開封や開示を実行するプログラムから構成されている。

図 4.13,4.14 にコレクション単位での預けモデルの構成図及びコレクション単位での返却モデルの構成図を示す。

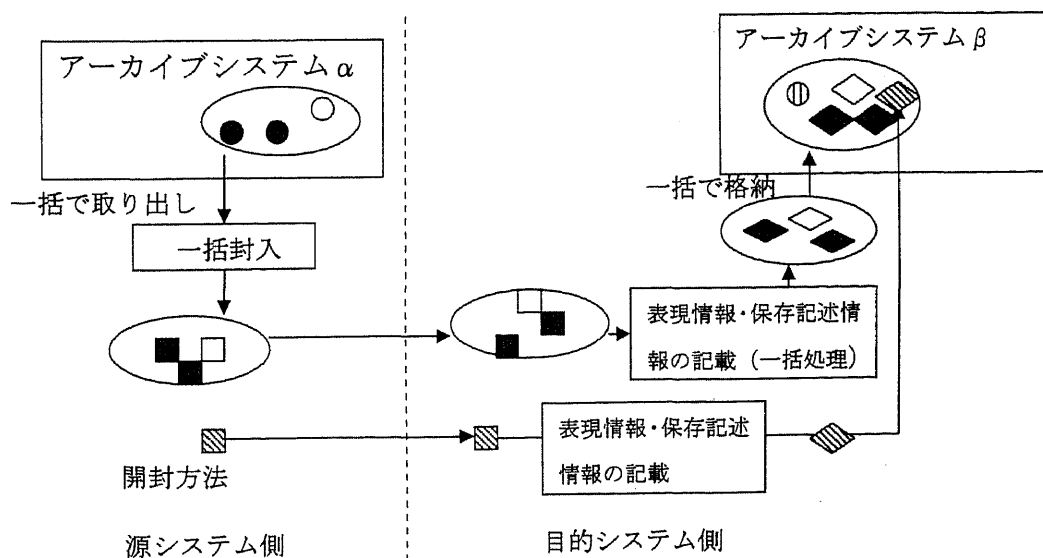


図 4.13 預けモデルの構成図 (コレクション単位)

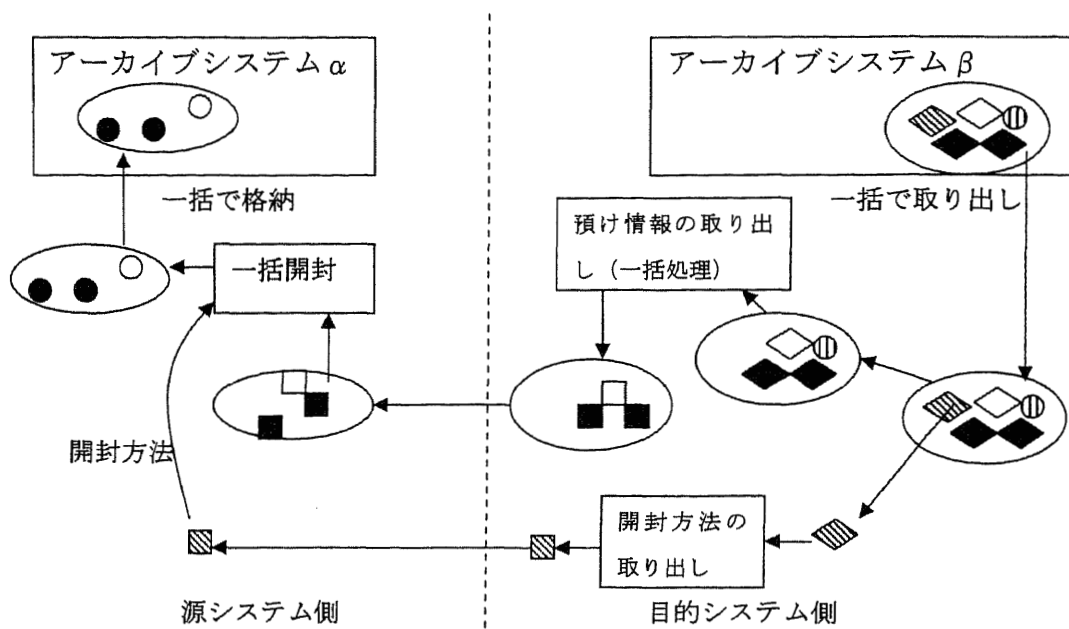
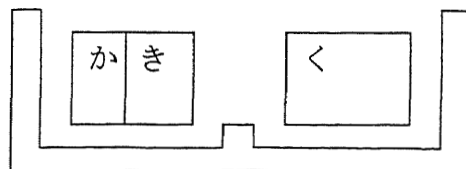


図 4.14 返却モデルの構成図 (コレクション単位)

- : 源システム側 AIC
- : 源システム側 AIU
- : 源システム側 AIC を封入したもの
- : 源システム側 AIU を封入したもの
- ▨ : 開封方法
- ◇ : 源システム側 AIC を封入したものに目的システム側で表現情報と保存記述情報をつけたもの
- ◆ : 源システム側 AIU を封入したものに目的システム側で表現情報と保存記述情報をつけたもの
- ⓪ : 目的システム側 AIC
- ▩ : 開封方法に目的システム側で表現情報と保存記述情報をつけたもの



- か : 開封方法 (文書・プログラム)、プログラムの実行環境及び実行方法
- き : 目的システム側表現情報
- く : 目的システム側保存記述情報

図 4.15 開封方法の情報モデル

源システム側では AIC 及び 1 つ以上の AIU から成るコレクションを保持している。目的システム側へ預ける際には、これらを一括して取り出し、一括して封入する。そして目的システム側へ保存情報のコレクション及び開封方法を送る。このとき、開封方法は封入しない。また、開封方法は開封方法を記述した文書や開示・開封のプログラム、プログラムの実行環境及び実行方法を記述した文書から成っている。開示プログラムとは封入済み保存情報の中身を覗き見るためのプログラムであり、後で詳しく説明する。また、開封プログラムとは封入された保存情報を開封し、元の状態に戻すプログラムである。目的システム側では預かった封入済み保存情報及び開封方法に対して表現情報と保存記述情報を記載する。図 4.15 に開封方法の情報モデルを示す。そして目的システム側が自身のデジタルアーカイブシステムへ格納する際に、預かりの保存情報を格納するためのコレクションを作成する。このとき、目的システム側の AIC を作成するが、この中に開封方法を同定するための情報を記載しておく。そして、このコレクションに預かりの保存情報及び開封方法を格納する。源システム側から返却の依頼があれば預かっている保存情報のコレクションを一括して取り出す。そして、預け情報と開封方法を別々にして、目的システム側で記述した表現情報と保存記述情報を削除する。こうすることで源システム側から受け取ったときの預け情報と開封方法を取り出せる。そして、源システム側から受け取った状態で保存情報及び開封方法を返却する。源システム側では開封プログラムで保存情報を一括開封し、自システム内に格納する。図 4.14 では目的システム側で記述した表現情報と保存記述情報を削除し、預かりの保存情報と開封方法を別々にして返却している。

本研究では預けから返却までのモデルの他に開示機能のモデルも提案する。開示機能とはカプセル化された源システム側保存情報の中身を覗き見る機能を言い、必要に応じて源システム側と目的システム側のいずれでも実行される。開示機能はカプセル化された源システム側保存情報の中の保存記述情報と表現情報を頼りに開示したい保存情報を同定し、その内容を閲覧する機能である。開示機能により、実際に表示したいコンテンツが見つければ、開封プログラムを用いて開封し表示する。

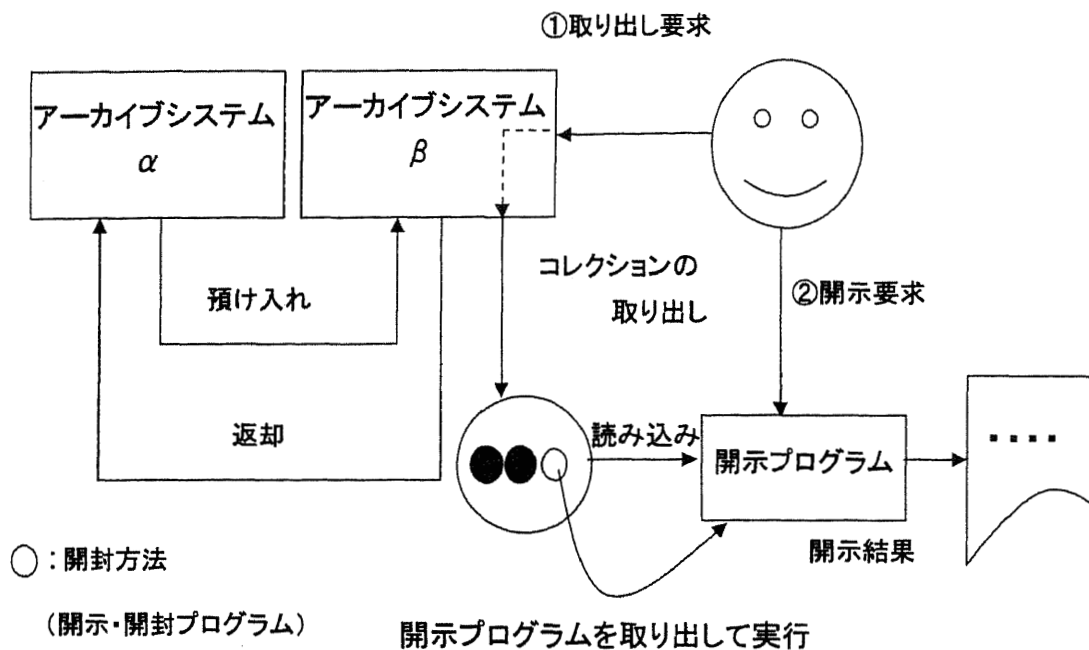


図 4.16 開示機能モデルの構成図

図 4.16 に開示機能モデルの構成図を示す。システム管理者はまずアーカイブシステムへ開示したい保存情報を持つコレクションの取り出し要求を行う。開示したい保存情報を持つコレクションの同定には目的システム側保存記述情報を用いる。開示したい保存情報を持つコレクションを取り出したらその中の開示プログラムを実行し、コレクション中の袋入りの保存情報を読み込む。このときコレクション中の白丸が開封方法であり、開封方法に開示プログラムと開封プログラムが含まれている。システム管理者は開示プログラムを用いて袋入り保存情報の開示を行う。このときの開示の手がかりは源システム側表現情報及び保存記述情報である。そして、開示結果から、開封し実際に表示したいコンテンツがあれば、開封プログラムを用いて開封し、コンテンツを表示する。

開示・開封プログラムの用途を以下にまとめる。

源システム側

- ・何を封入したかを開示プログラムで確認
- ・開示プログラムで何を封入したか確認できたら開封プログラムで所望の保存情報を開封

目的システム側

- ・預かっている保存情報の中身を開示プログラムで確認
- ・開示プログラムで何を封入したか確認できたら開封プログラムで所望の保存情報を開封
- ・源システムがトラブル等によりアーカイブシステムとしての責任を負わなくなった際に保存情報を開封して源システムの責任を引き継ぐ

4.2.5 環境モデル

本研究で提案する連携保存モデルでは封入・開封・開示の各作業はデジタルアーカイブシステムとは別の環境で実行する。それは、デジタルアーカイブシステムがインストールされている環境を変えずに連携保存を行うことが可能であることを指している（図 4.17 参照）。また、各作業を行うための環境は必要に応じて構築すればよく、常に環境を整えておく必要はない。つまり、源システムが目的システムへ保存情報を預けたいと考えたときに封入プログラム作成などの封入のための環境を整えればよい。その他の作業環境においても同様である。

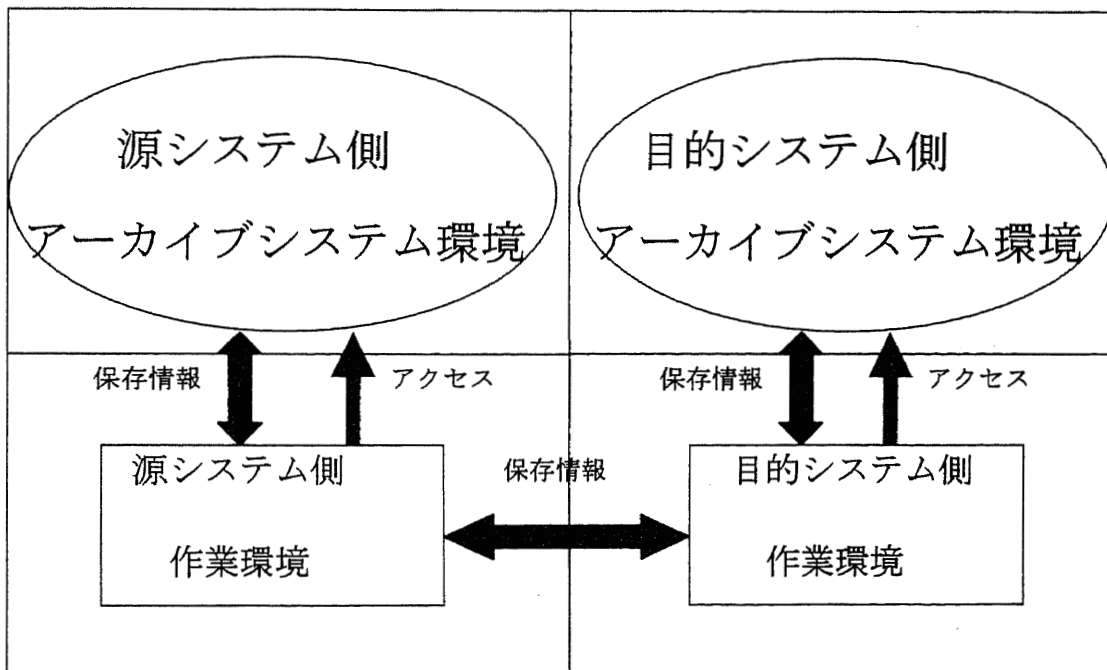


図 4.17 環境モデル

5. プロトタイプシステムの開発

5.1 プロトタイプシステムの概要

5.1.1 プロトタイプシステムの開発目的

4章で述べた連携保存モデルを実証するために連携保存モデルのプロトタイプシステムを開発する。実際にプロトタイプシステムを開発し、預けから返却の一連の流れ及び開示機能を実行することで考案したモデルが実際にシステム化して運用可能であることを確認する。

5.1.2 プロトタイプシステムの開発環境及び運用環境

プログラム開発環境

C 開発環境

- ・ Microsoft Visual C++6.0

java 開発環境

- ・ Eclipse Platform2.1.1(Eclipse java 開発ツール 2.1.1)(注 5.1)

プロトタイプシステムの運用環境

アーカイブシステム環境

- ・ 源システム：学内のサーバー-gaucho(Solaris8)
- ・ 目的システム：学内のサーバー-avalon(FreeBSD4.9)

作業環境（源システム側、目的システム側共通）

- ・ OS:windowsXP
- ・ java の実行環境

5.1.3 プロトタイプシステムの構成

プロトタイプシステムは封入プログラム、開封プログラム、開示プログラム、返却編集プログラム、預け入れ編集プログラム、預かり情報取り出しプログラムの6つのプログラムと各種ソフトウェアから構成されている。図 4.17 で示す環境でそれらソフトウェア及びプログラムを操作することで連携保存のモデルを実現する。なお、プロトタイプシステムでは開封方法を開示・開封プログラムのみとし、本文でも開封方法を開示・開封プログラムとして記述している。ただし、実用の際にはプログラムと併せて開封方法を記述した文書を作成すべきである。

プロトタイプシステムを構成する主なソフトウェアを以下に挙げる。

- ・ DSpace
- ・ PostgreSQL[10]
- ・ Microsoft Excel
- ・ Microsoft Access

・Tomcat[11]

各種ソフトウェアと封入プログラム、開封プログラム、開示プログラム、返却編集プログラム、預け入れ編集プログラムを用いて本研究で作成したプロトタイプシステムを表現すると図 5.1 のようになる。今回は DSpace を 2 つの UNIX サーバーへインストールし、それぞれのアーカイブシステム環境に対する作業環境として Windows XP の動いている PC を用意した。

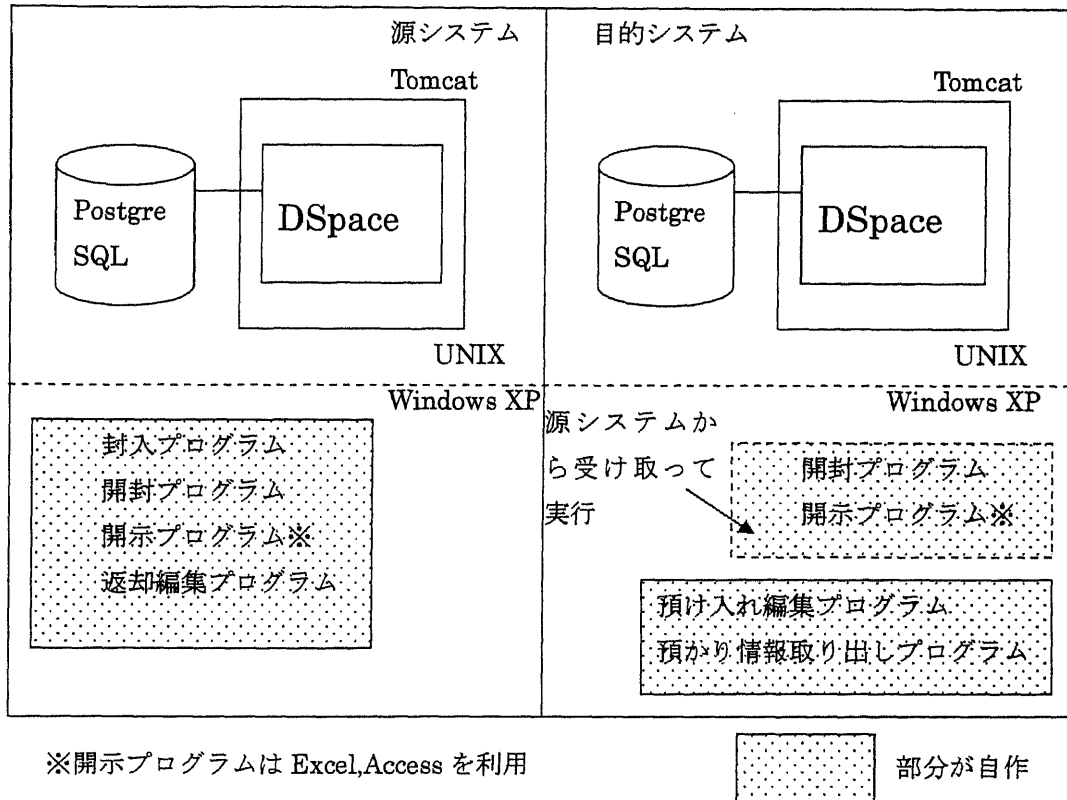


図 5.1 プロトタイプシステムのソフトウェア構成

プロトタイプシステムでは 2 つの異なる UNIX サーバー上に DSpace と PostgreSQL (注 5.2) をインストールする。ここで、DSpace は jsp (注 5.3) で記述されているためサーブレットコンテナ (注 5.4) が必要となる。プロトタイプシステムでは Tomcat (注 5.5) 上で DSpace を動作している。また、封入や開封といった作業は Windows XP 上で作業するため封入・開封・開示・返却編集・預け入れ編集・預かり情報取り出しの各プログラムは Windows XP 上に存在している。また開示プログラムで用いられる Excel や Access も同様に Windows XP 上に存在している。これら各ソフトウェアやプログラムを管理者が操作することで連携保存のモデルを実現する

5.1.4 封入プログラム

封入プログラムは源システム側保存情報を XML 技法によりカプセル化するためのプログラムである。封入プログラムの大まかな流れは次のようになる。

1. 封入するコレクションの場所およびコレクション中の保存情報の個数を指定
2. 保存情報を封入して格納するための XML ファイルを作成(encap-i とする)
3. コレクション中の保存情報-i の取り出し
4. 保存情報-i の CDO の取り出し
5. 取り出した CDO を base64Binary へ変換し、encap-i へ格納
6. 保存情報-i の RI を取り出し、encap-i へ格納
7. 保存情報-i の PDI を取り出し、encap-i へ格納
8. encap-i の整形 (XML 宣言の記述及び図 4.3、4.5 形式へ変更)
(ただし、i=1~コレクション中の保存情報の個数)

本研究で作成した封入プログラムはパラメータとして封入したいコレクションの場所とコレクションを構成している保存情報の個数を与えてやれば、上の一連作業をコレクション中の保存情報の個数だけ一括して行うものとなっている。また封入プログラムは C 言語を用いて作成しており、base64Binary への変換に MIME コンバート DLL (注 5.6) を活用した。

5.1.5 開封プログラム

開封プログラムはカプセル化された源システム側保存情報を元の形式に戻すためのプログラムである。開封プログラムの大まかな流れは次のようになる。

1. 開封するコレクションの場所およびコレクション中の保存情報の個数を指定
2. 開封した源システム側保存情報を格納するためのフォルダを作成(decap-i とする)
3. コレクション中の encap-i の取り出し
4. encap-i から源システム側 CDO を抽出し、base64Binary からビット列へ変換。さらに RI を使って元のファイル名にし、decap-i へ格納
5. encap-i から源システム側 PDI を抽出し、decap-i へ格納
(ただし、i=1~コレクション中の保存情報の個数)

本研究で作成した開封プログラムはパラメータとして開封したいコレクションの場所とコレクションを構成している保存情報の個数を与えてやれば、上の一連作業をコレクションの保存情報の個数だけ一括して行うものとなっている。また開封システムは C 言語と java 言語を用いて作成しており、XML 文書の解析には正規表現を活用し、base64Binary のデコードには javamail[12]のアプリケーションプログラムインターフェースを活用した。また、開封プログラムを実行するには java の実行環境が必要となる。

5.1.6 開示プログラム

開示プログラムはカプセル化された源システム側保存情報を覗き見るためのプログラムであり、必要に応じて Microsoft Excel、Microsoft Access を利用する。開示プログラムはまず自作のプログラムにより開示したいコレクションの源システム側保存情報(XML 文書)を接続する。ここでいう接続とは、開示したいコレクションの要素であるカプセル化された源システム側保存情報 (XML 文書) を保存情報の個数だけ繋げて1つの XML 文書にすることである。1つの XML 文書にすることにより、コレクション単位で源システム側保存情報を開示できる (図 5.2 参照)。接続してできた XML ファイルを Excel にインポートし、ツリー構造を維持したままでテーブルを作成する。その Excel ファイルを Access にインポートし、開示機能を実現する。

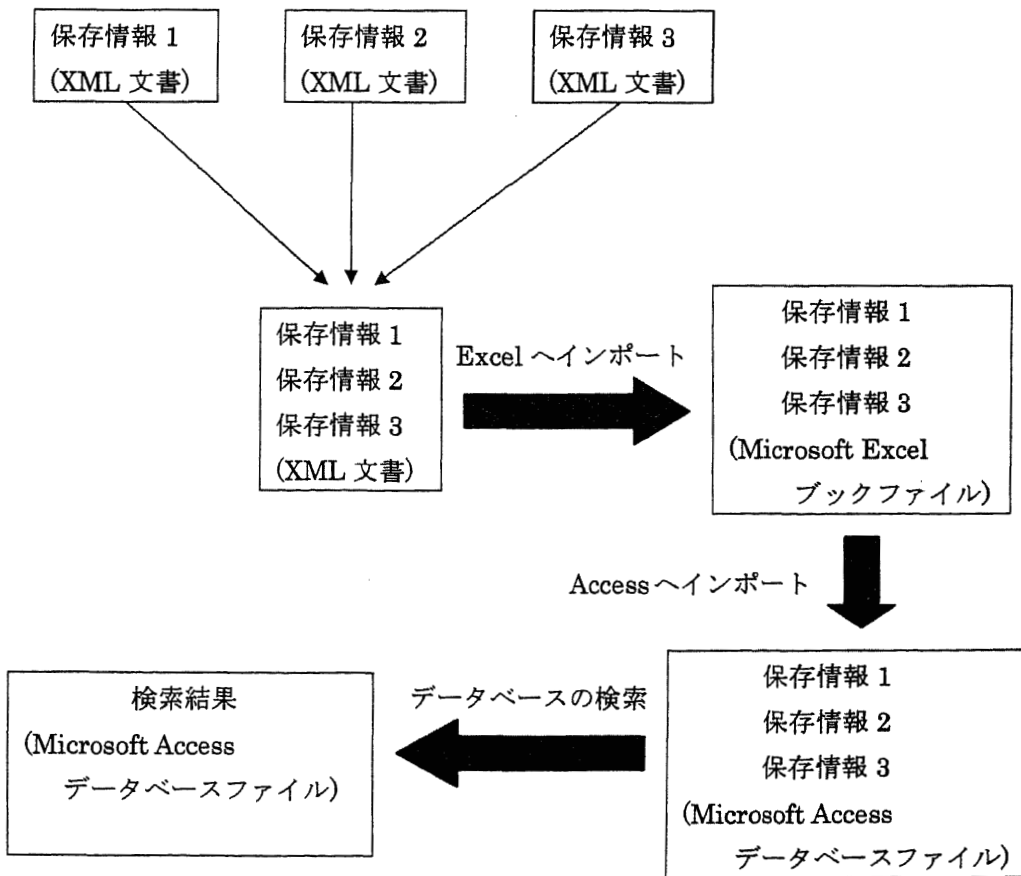


図 5.2 開示機能を実行するまでの手順

5.1.7 返却編集プログラム及び預け入れ編集プログラム

返却編集プログラム及び預け入れ編集プログラムは保存情報をデジタルアーカイブシステムへ格納する際に必要となるファイルを生成するためのプログラムである。このプログラムは完全に各システムに依存したものとなる。プロトタイプシステムでは簡単のために、返却編集プログラムを開封プログラムに組み込み、開封プログラムにより源システム側保存情報が開封される際に、DSpace へ格納するためのファイルを生成するようにしている。一方、預け入れ編集プログラムはカプセル化された源システム側保存情報を目的システムが自システムへ格納する際に必要となるため、別途作成した。預け入れ編集プログラムには目的システム側の保存記述情報を一括して記述する機能も含まれている。実際には返却編集プログラムは源システムが目的システムへ保存情報を預ける際に添付する必要はない。しかし、プロトタイプシステムでは開封プログラムとセットになっているため、目的システム側へ保存情報と共に預けることになる。

5.2 プロトタイプシステムに用いるデジタルアーカイブシステム (DSpace)

[2][4][13][14]

5.2.1 DSpace の概要

DSpace はマサチューセッツ工科大学とヒューレットパッカード社が共同開発したオープンソースのデジタルアーカイブシステムである。DSpace はその機能が OAIS 参照モデルのそれらによく対応したものになっており、DSpace を構成する各種ソフトウェアはオープンソースやフリーウェアのソフトウェアが活用されている。

DSpace はコンテンツを利用者から受け取り、データベースに格納する。そしてそれらコンテンツを保管し、要求に応じて配布するという一連の機能を実現する。

本研究では OAIS 参照モデルに基づいて研究を行い、プロトタイプシステムに DSpace を採用した。DSpace を採用した理由には DSpace 及び DSpace を構成するソフトウェアがオープンソースやフリーウェアである点と本研究で扱う OAIS の機能（表現情報や保存記述情報といったメタデータ、コレクションの概念など）を DSpace が標準で備えている点が挙げられる。

5.2.2 DSpace の情報モデル

DSpace の情報モデルを図 5.3 に示す。

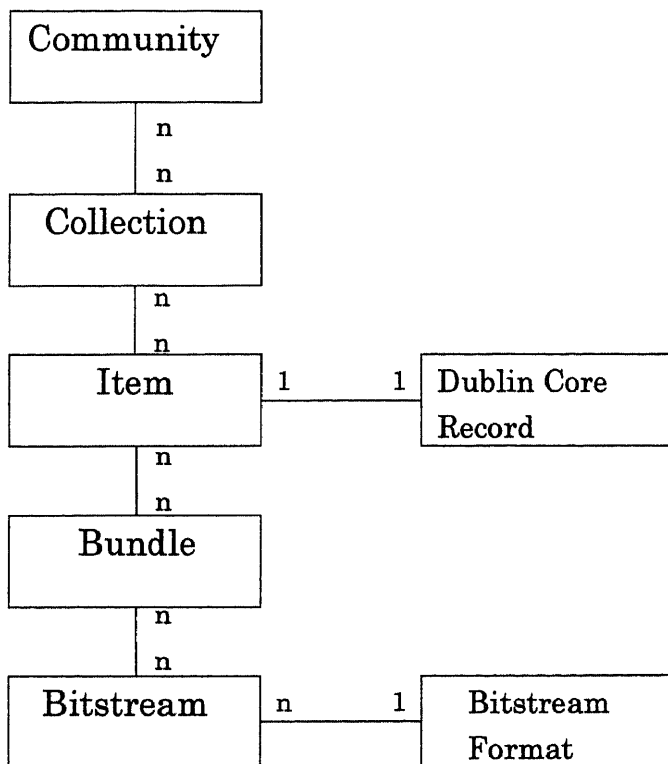


図 5.3 DSpace の情報モデル

DSpace における保存情報の基本単位は Item である。Item は 1 つ以上のコレクションに、コレクションは 1 つ以上の Community に所属する。さらに Item は複数のビットストリームをまとめる Bundle からなる。Item は Dublin Core Record を必ず持ち、Bitstream は Bitstream Format を参照する。ここで Bitstream はビット列そのものであり、そのビット列をまとめるものとして Bundle が存在する。例えば HTML 文書が html ファイルと jpg ファイルから構成されている場合、Bundle は HTML 文書であり、Bitstream は html ファイルと jpg ファイルである。さらに Bitstream は Bitstream Format を参照する。Bitstream Format はそのフォーマットのコンテンツをどのように解釈するかを示すものである。

図 5.4 に Community-Collection-Item-Dublin Core Record の関係の例を示す。Community として“工学部”を作成する。工学部は“電気工学科”、“情報工学科”、“4 年次生”の 3 つの Collection から成る。さらに、Item として A 君、B 君、C 君、D 君があり、A 君

は電気工学科のコレクションの要素であり、B君は電気工学科と4年次生のコレクションの要素である。C君、D君も同様に考える。また、A君、B君、C君、D君はそれぞれ1つの dublin_core record を持つ。

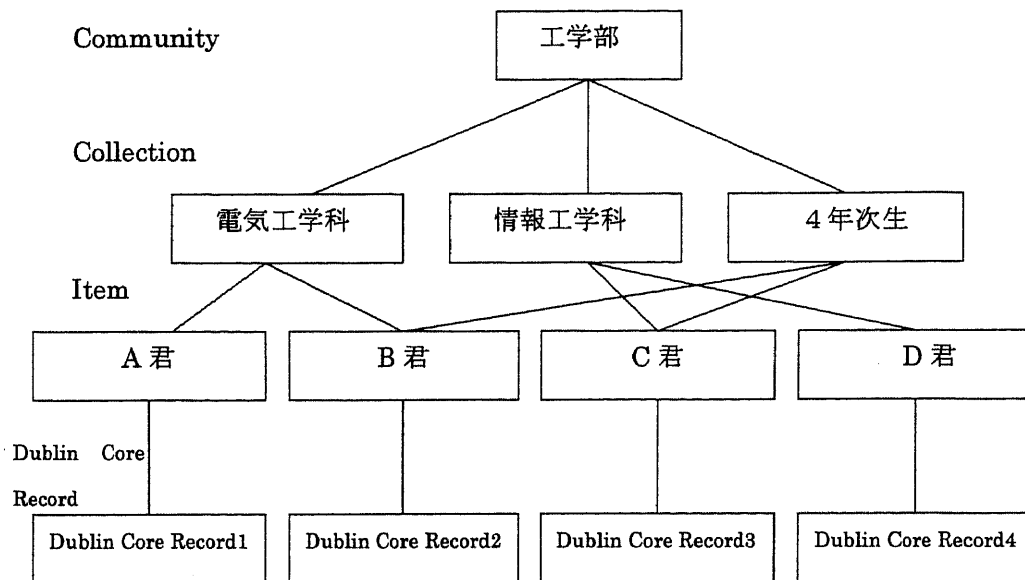


図 5.4 Community-Collection-Item-Dublin Core Record の関係

図 5.5 に Item-Bundle-Bitstream-Bitstream Format の関係の例を示す。Item1 と Item2 はそれぞれ1つの Dublin Core Record を持っている。Item1 は Bundle1 と Bundle2 からなっており Item2 は Bundle2 と Bundle3 から成っている。Bundle2 は HTML 文書であり、Bitstream1、Bitstream2、Bitstream3 から成っている。Bitstream1 は html ファイルなので Bitstream Format として”HTML Format”という情報を持っている。また、Bitstream2 と Bitstream3 は HTML 文書内の JPEG ファイルなので Bitstream Format として”JPEG Format”という情報を持っている。Bundle1 および Bundle3 も Bundle2 と同様に1つ以上の Bitstream から成る。

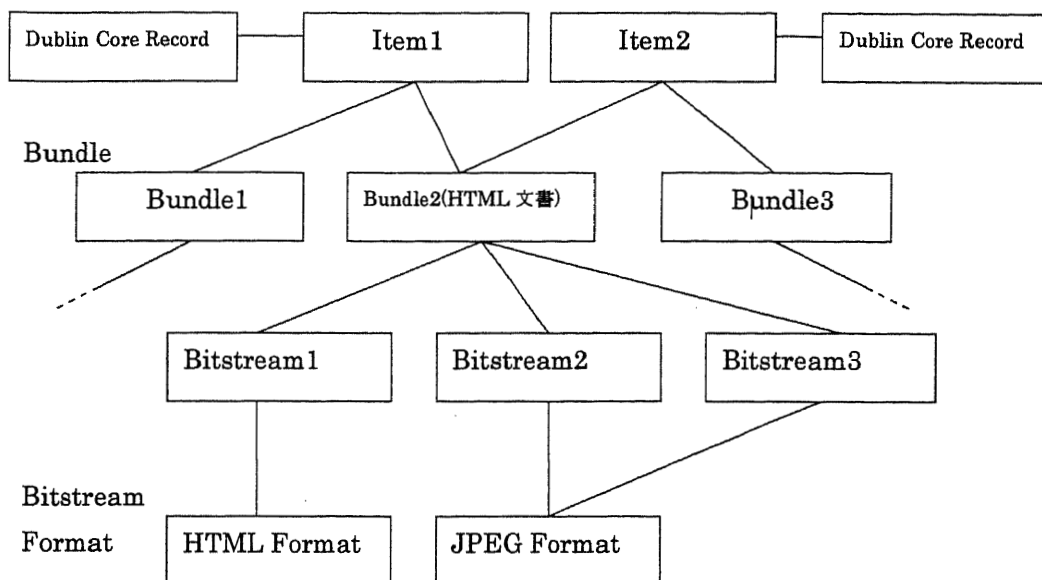


図 5.5 Item-Bundle-Bitstream-Bitstream Format の関係

各ビットストリームフォーマットはさらに、サポートレベルを持つ。これは、運営機関がそのフォーマットのコンテンツを将来にわたってどの程度保存できるかを示している。運営機関が設定できるビットストリームフォーマットのサポートレベルには Supported・Known・Unknown の3つがある。これら3つのサポートレベルの定義は以下の通りである。

Supported : フォーマットは認識されている。運営機関は、マイグレーションやエミュレーションのような技術により、このフォーマットのビットストリームを将来においても利用できることを確信している。

Known : フォーマットは認識されている。運営機関はこのフォーマットのビットストリームをそのままの形で保存し、引き出せることを約束する。また、運営機関は、このフォーマットを Supported レベルに引き上げる努力をする。

Unsupported : フォーマットは認識されていない。しかし、運営機関はこのビットストリームをそのままの形で保存し、引き出せることを引き受ける。

5.2.3 DSpace におけるコンテンツの提出及び取り出し

DSpace におけるコンテンツの提出および取り出しの実行には2つの方法がある。1つは Web UI(ウェブユーザーインターフェース)を用いる方法である。これは Web ブラウザからコンテンツの提出及び取り出しを行う方法であり単一コンテンツごとに実行する。もう一方の方法は Batch Item Importer 及び Batch Item Exporter を用いる方法である。Batch Item Importer 及び Batch Item Exporter は UNIX 上でコマンドを入力することによりコンテンツの提出及び取り出しを行う DSpace の機能である。この方法を用いれば、Item をコレクション単位で一括してインポート及びエクスポートできる。なお、Batch Item Importer 及び Batch Item Exporter で Item をインポート・エクスポートする際には DSpace シンプルアーカイブフォーマットを用いる。

DSpace シンプルアーカイブフォーマットとは Batch Item Importer 及び Batch Item Exporter を実行する際の Item のデータ形式である。図 5.6 にインポートの際の DSpace シンプルアーカイブフォーマットの例を示す。コレクションのディレクトリが Item のサブディレクトリを1つ以上持っており、Item のサブディレクトリはコンテンツのファイル以外に dublin_core.xml (ダブリンコアメタデータファイル) ファイル (図 5.7 参照) と contents ファイルを持つ。contents ファイルとは各 Item ディレクトリ内のインポートしたいファイルあるいはエクスポートしたファイルを記述したファイルである。さらに、エクスポートした場合にはこのほかに handle という識別子ファイルが含まれる。

```
collection1/
  item1/
    dublin_core.xml
    contents
    abc.txt
    def.doc
  item2/
    dublin_core.xml
    contents
    ghi.jpg
    jkl.pdf
  . . . . .
```

図 5.6 DSpace シンプルアーカイブフォーマット

```

<dublin_core>
<dcvalue element="title" qualifier="none"> あいうえお</dcvalue>
<dcvalue element="date" qualifier="issued">2005</dcvalue></dublin_core>
<dcvalue element="title" qualifier="alternate" language="jp ">おかだ </dcvalue>
</dublin_core>

```

図 5.7 dublin_core.xml のフォーマット

図 5.6 で collection1 というディレクトリの下に item1、item2・・・というサブディレクトリが存在している。このサブディレクトリの中には dublin_core.xml ファイルと contents ファイル、インポートするコンテンツのファイルが格納されている。なお、item1 内の contents ファイルにはインポートするコンテンツファイル名である"abc.txt"と def.doc"が記述されている。他の contents ファイルも同様である。

図 5.7 で各ダブリンコアメタデータは、dcvalue 要素内に格納される。さらに dcvalue 要素は"element"、"qualifier"、"language"の 3 つの属性を持つ。"element"属性にはダブリンコア要素(Dublin Core Metadata Element Set で定義されている 15 の要素名)を記述し、"qualifier"はダブリンコア限定子 (Dublin Core Metadata Element Set をより詳細に記述するための要素) を記述する。また"language"にはダブリンコア要素の ISO 言語コードを記述する。

次に DSpace の Batch Item Importer 及び Batch Item Exporter の使用方法について説明する。図 5.8 は collection へ item を追加するコマンドを示している。下線部分には適切なパラメータを与えてやる。joe@user.com へは dspace へ登録した e メールアドレスを記述し、collectionID へは格納したい DSpace のコレクション ID を記述する。(コレクション ID は DSpace のウェブユーザーインターフェース等から確認できる。) また、items_dir へは DSpace へ格納したい item の場所を記述する。mapfile は後からインポートした item の削除や置き換えを行う際に使用するもので、適当な名前を付けて覚えておく。ここで items_dir へは図 5.6 における collection1 のような、コレクションのディレクトリを指定するということに注意が必要である。つまり、図 5.8 のコマンドにより複数の item を一括してインポート出来る(図 5.6 では collection1 を指定することで item1,item2・・・をインポート出来る)。

```

dsrun org.dspace.app.itemimport.ItemImport --add --eperson=joe@user.com
--collection=collectionID --source=items_dir --mapfile=mapfile

```

図 5.8 collection へ item を追加するコマンド

図 5.9 はインポートした item を削除するコマンドを示している。mapfile へ削除したい item の mapfile を記述する。

```
dsrun org.dspace.app.itemimport.ItemImport --remove --mapfile=mapfile
```

図 5.9 インポートした item を削除するコマンド

図 5.10 はインポートした item を別の item に置き換えるコマンドを示している。joe@user.com へは dspace へ登録した e メールアドレスを記述し、collectionID へは置き換えられる item の格納されているコレクションの ID を記述する。また、items_dir へは置き換える item の場所を記述する。mapfile には置き換えられる item の DSpace へインポートした際に記述した mapfile を記述する。なお、図 5.8 と同様に items_dir へはコレクションのディレクトリを指定するため、一括して複数 item の置き換えができる。

```
dsrun org.dspace.app.itemimport.ItemImport --replace --eperson=joe@user.com  
--collection=collectionID --source=items_dir --mapfile=mapfile
```

図 5.10 インポートした item を別の item に置き換えるコマンド

図 5.11 はコレクション単位で item を取り出すコマンドを示している。collID へは取り出したいコレクションの ID を記述し、dest_dir へは取り出したコレクションの保存先を記述する。また seq_num へ記述した番号から順に DSpace シンプルアーカイブフォーマットへ付番する。つまり、図 5.6 の item1、item2・・・が seq_num から始まる番号に割り当てられる。

```
dsrun org.dspace.app.itemexport.ItemExport --type=COLLECTION --id=collID  
--dest=dest_dir --number=seq_num
```

図 5.11 collection 単位で item を取り出すコマンド

図 5.12 は item を単一で取り出すコマンドを示している。itemID へは取り出したい item の ID を記述し、dest_dir、seq_num はコレクション単位での取り出しと同様である。

```
dsrun org.dspace.app.itemexport.ItemExport --type=ITEM --id=itemID  
--dest=dest_dir --number=seq_num
```

図 5.12 item を単一で取り出すコマンド

5.3 プロトタイプシステムの実装モデル

図 5.13 に預けの実装モデルを示し、図 5.14 に返却の実装モデルを示す。なお、図中の記号 A~L は図 5.14 の下に示したファイル構成内の記号 A~L に対応している。

記号 A~L は預け及び返却の過程におけるファイルの構成を示している。また、図 5.13 及び図 5.14 は図 4.13 及び図 4.14 に対応したものになっている。

預けの実装モデルについて説明する。まず、DSpace 内に格納されている保存情報 (A) をコレクション単位で UNIX 上の指定したディレクトリ内 (ここでは okadatak とする) に一括して取り出す。この際、Batch Item Exporter を用いる。そして ftp を用いて封入作業を行う環境 (今回は WindowsXP) へコレクションを一括して転送する。そして封入プログラムを用いて保存情報をコレクション単位で一括してカプセル化する (B)。カプセル化した保存情報及び開示・開封プログラム (C) を目的システム側へ送る。目的システム側ではカプセル化された保存情報を預け入れ編集プログラムで目的システム側 DSpace に格納できるような形態 (DSpace シンプルアーカイブフォーマット) にし (E)、開示・開封プログラムを手作業で目的システム側 DSpace に格納できるような形態にする (D)。この時、カプセル化された保存情報に対するダブリンコアメタデータファイルの作成 (プログラムによる一括処理) や開示・開封プログラムへのダブリンコアメタデータファイルの作成 (手作業) も行われる。そしてカプセル化された保存情報及び開示・開封プログラムは DSpace が動いている環境へ ftp により転送される (F)。そして、カプセル化された保存情報と開示・開封プログラムを同一のコレクション内に格納する (G)。この時 Batch Item Importer を用いる。また、G 内の AIC.txt は目的システム側管理者が源システム側から預かった保存情報を DSpace へ格納する際に作成する。

返却の実装モデルについて説明する。源システム側から預かっている保存情報の返却依頼を受けたら DSpace 内に保持している預かりの保存情報及び開示・開封プログラムから成るコレクションを okadatak ディレクトリへ一括して取り出す (G)。この際、Batch Item Exporter を用いる。そして ftp を用いて目的システム側で作成した情報の削除を行う環境へコレクションを一括して転送する。そして、保存情報 (H) と開示・開封プログラム (I) を分ける。目的システム側で作成した情報を削除することで源システム側から受け取った状態の預かり情報 (J) 及び開示・開封プログラム (K) を取り出し、源システム側へ返却する。目的システム側でつけた情報の削除は、保存情報に対しては預け情報取り出しプログラムで一括して行い、開示・開封プログラムに対しては手作業で行う。源システム側では開封プログラムを用いて保存情報を開封し、返却編集プログラムにより源システム側 DSpace へ格納できるような形態にする (L)。開封された保存情報は ftp により DSpace が動いている環境へ転送され、Batch Item Importer により DSpace 内に格納される (M)。

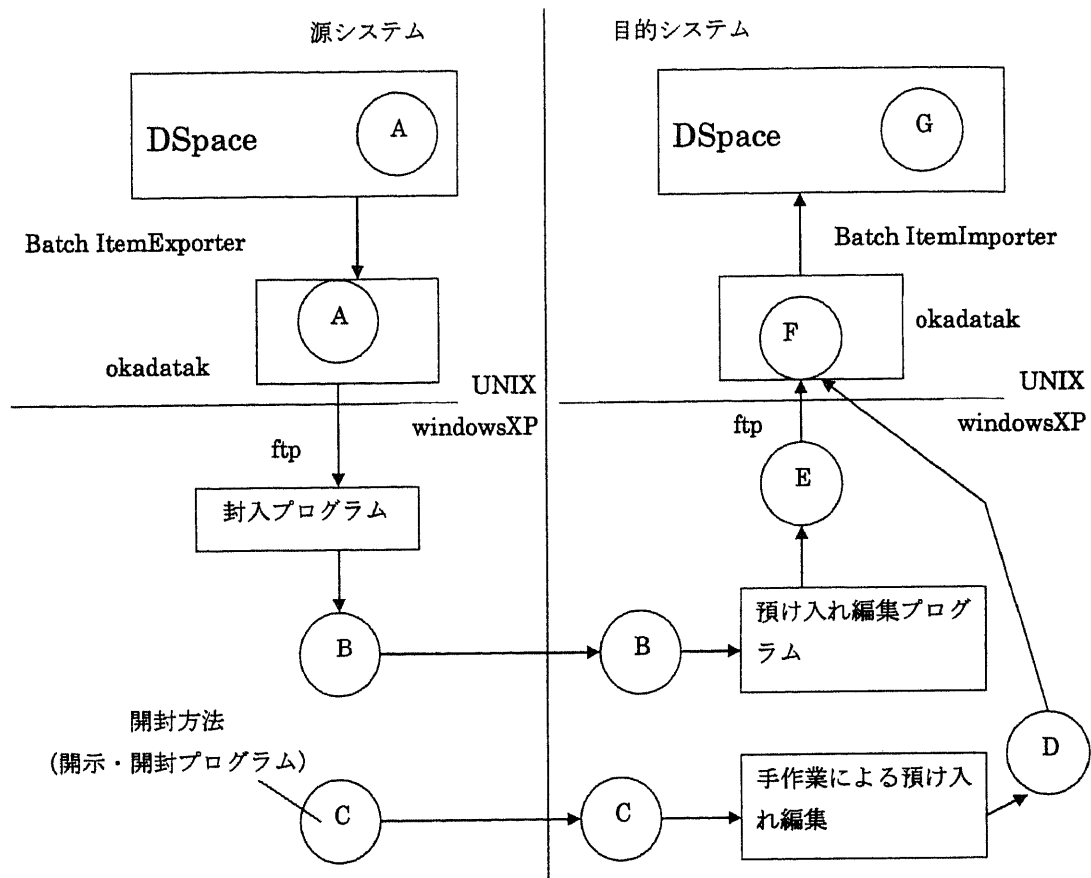


図 5.13 預けの実装モデル

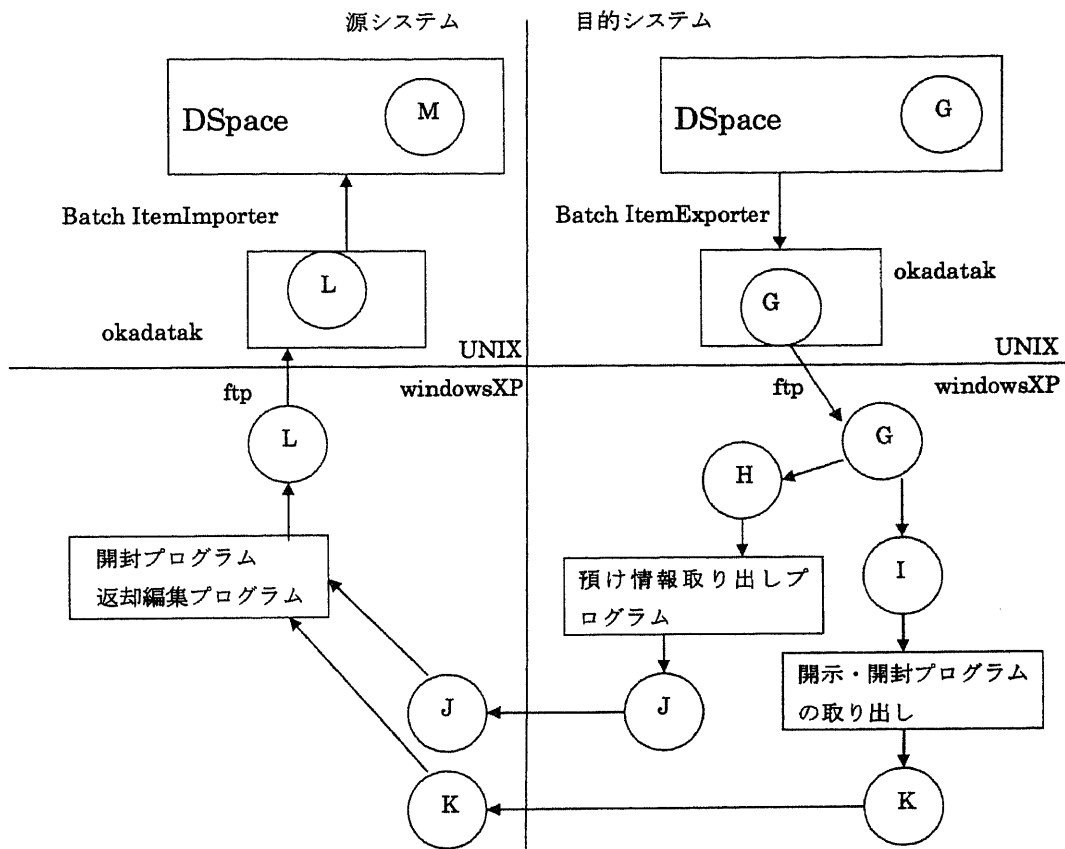
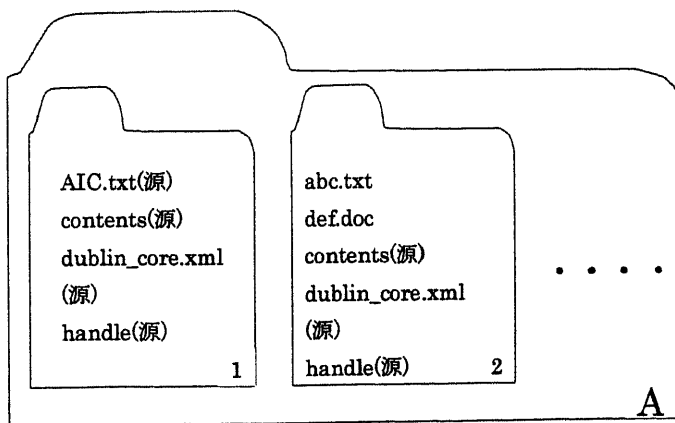
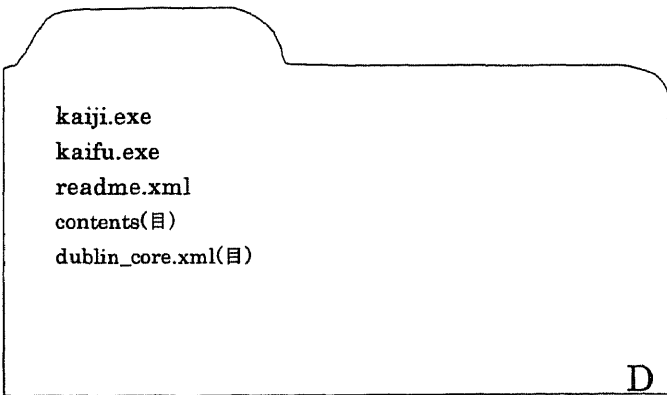
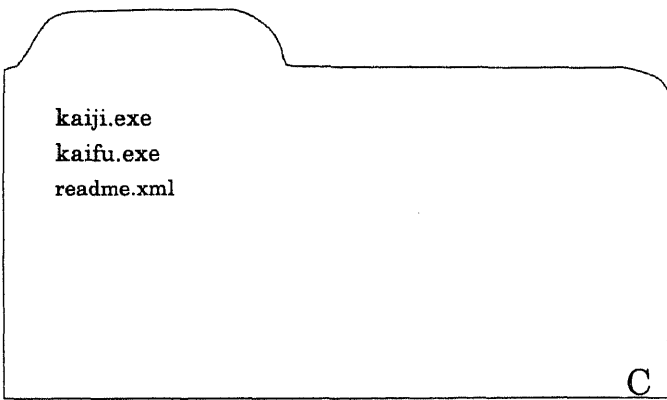
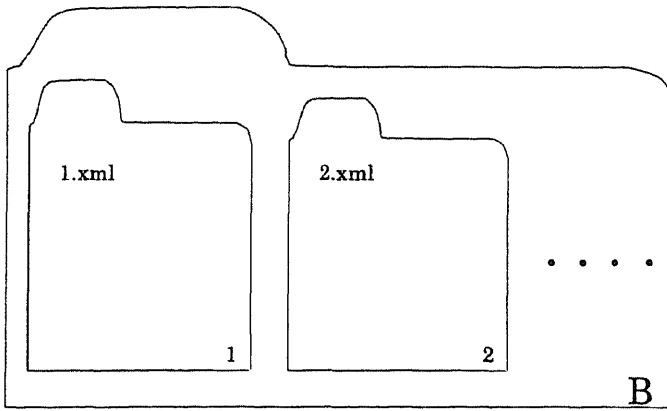
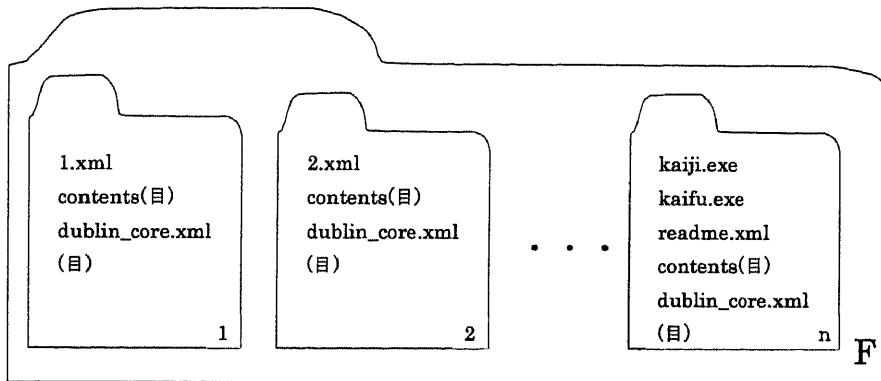
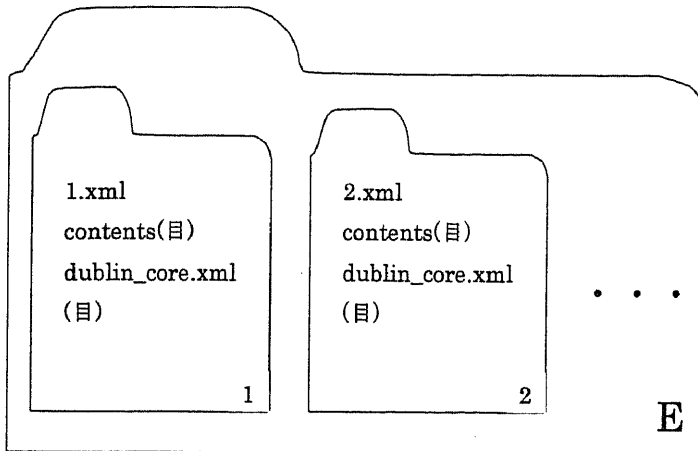


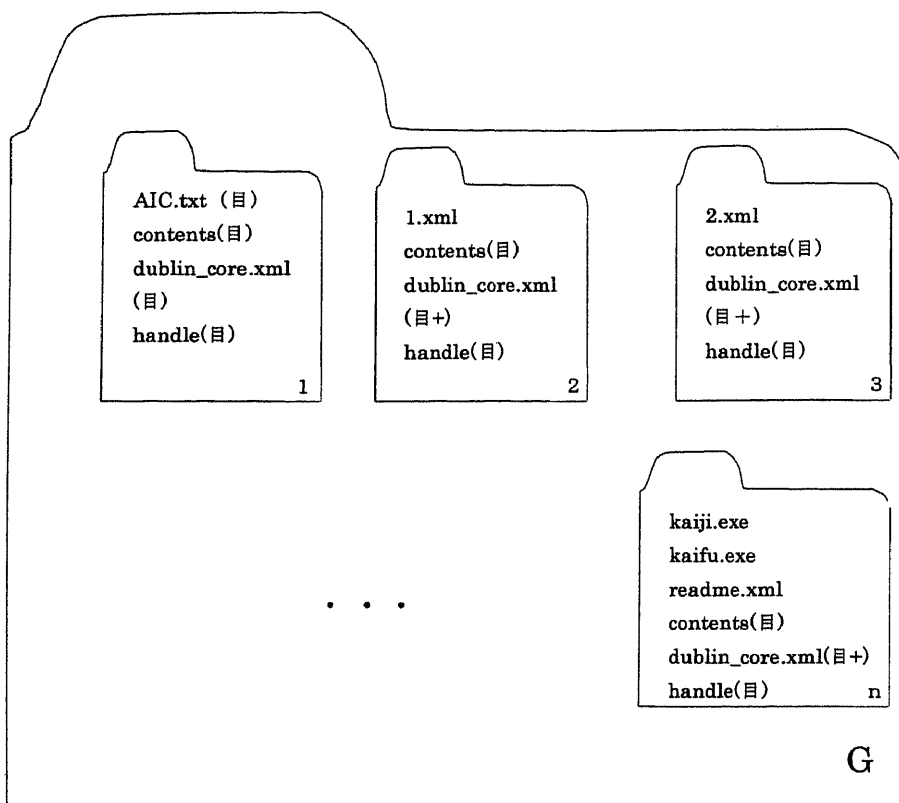
図 5.14 返却の実装モデル

ファイル構成

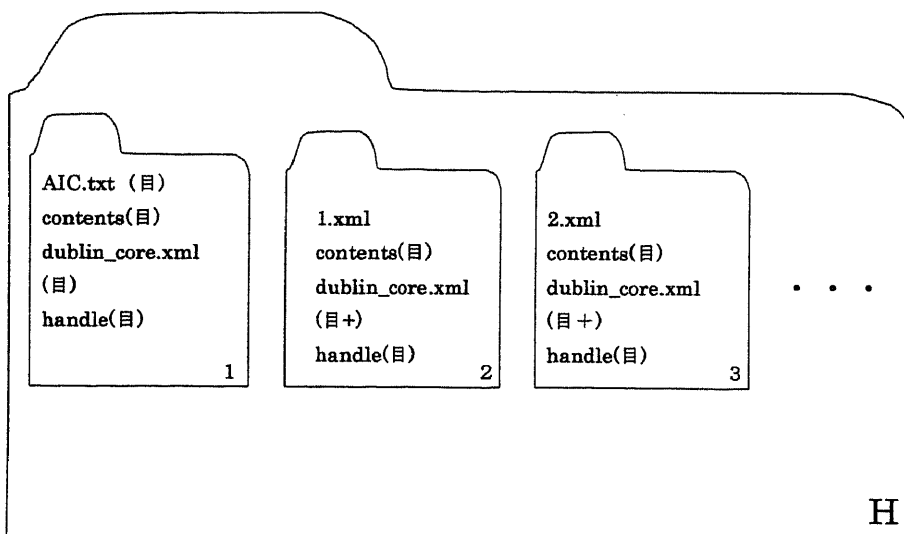


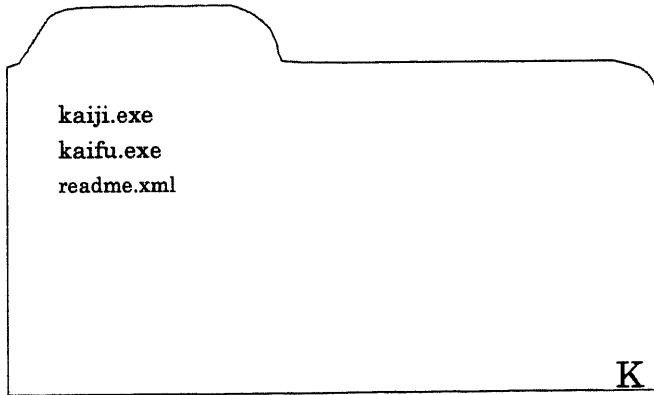
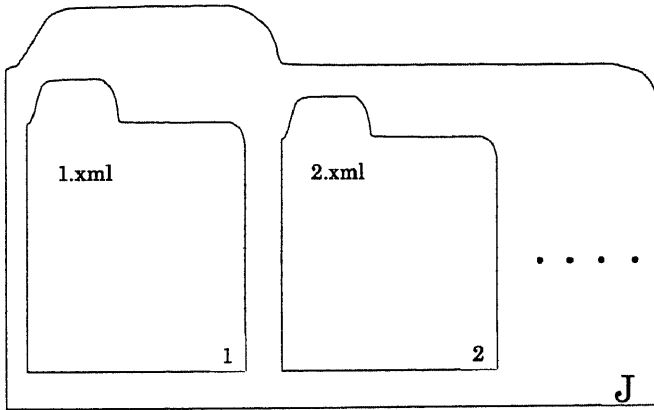
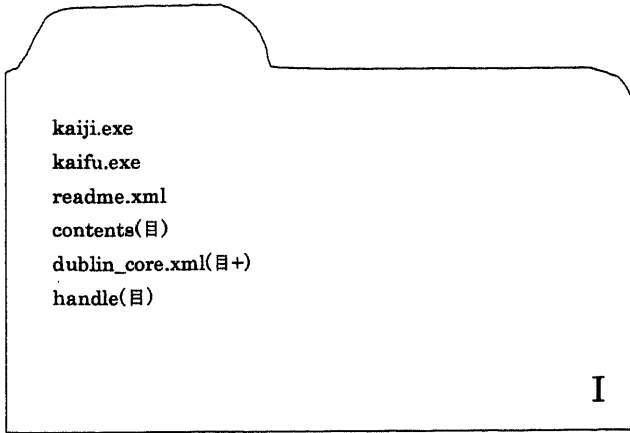


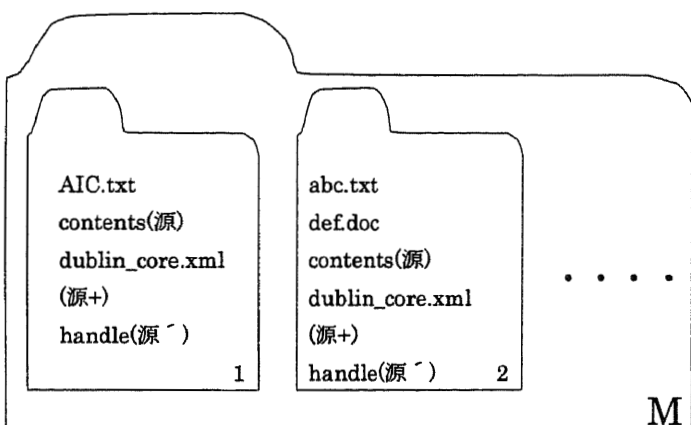
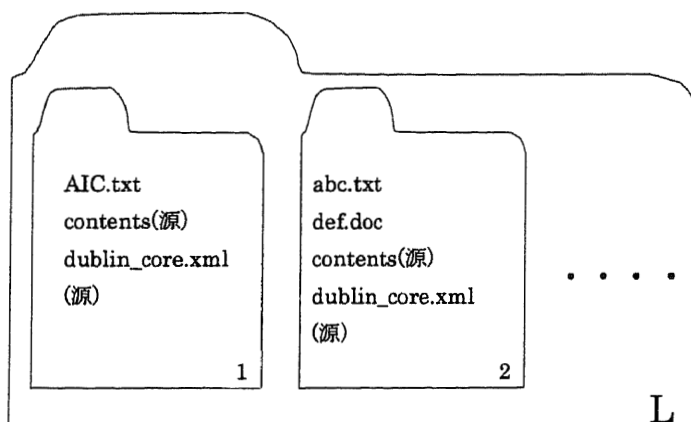




(※)dublin_core.xml(目+)とは F の dublin_core.xml(目)に DSpace が新たに dublin_core の要素を追加したことを意味する。







(※) dublin_core.xml(源+)は L の dublin_core.xml(源)に DSpace が新たに dublin_core の要素を追加したことを意味する。

(※) handle(源^)は A の handle(源)とは違うことを意味する。

(※) DSpace 中のⒶ,Ⓒ,Ⓜは DSpace シンプルアーカイブフォーマットで表現している。

次に、開示機能の実装方法について説明する。開示機能はカプセル化された源システム側保存情報を覗き見るためのものであり、必要に応じて Microsoft Excel、Microsoft Access を利用する。開示機能実行の大まかな流れは以下ようになる (図 4.16, 図 5.2 参照)。

1. DSpace のウェブユーザーインターフェース (WebUI) を用いて、開示したいコレクションの同定 (目的システム側 PDI を頼りに行う。)
2. 開示したいコレクションを取り出す(前述のファイル構成Ⓒ)。
3. 開示したいコレクションの源システム側保存情報 (XML 文書) を接続する (Ⓒ内の kaiji.exe を用いる)。

4. 3 で生成された接続ファイルを Microsoft Excel に取り込む。
5. 4 で生成された Excel ファイルを Access に取り込む。
6. 5. で生成されたファイルを Access のクエリ機能を使って開示。(源システム側 PDI 及び RI を頼りに行う。)
7. 実際に開封してみたいコンテンツがあれば開封プログラム (ⓐ内の kaifu.exe) を用いて CDO をビット列に変換し、RI を頼りにコンテンツを表示する。(5.1.3 と同様の手続きで行う)

(※) ここで開封プログラム実行の前にあらかじめ開封したい保存情報からなるコレクションを作成し、それを開封プログラムに与える。

(※) 開示機能は目的システム側に限らず源システム側でも実行され得る。ⓐに対して開示機能が実行されるというのは、ファイル構成がⓐであるものに対して実行されるという意味であり、図 5.13,5.14 内のⓐに対して実行されるという意味ではない。また、上述の開示機能実行の大まかな流れ (1~7) は目的システム側を対象とした流れであり、源システム側で実行する際には流れの 3 から始めることになる。

5.4 DSpace におけるデータ記述の例

源システムが次の 3 つの item を目的システムへ預ける場合を例にとって DSpace におけるデータ記述について述べる。源システム側 item 名とはカプセル化する前の item のタイトルで源システム側コンテンツファイルがその item を構成するコンテンツファイルである。また、目的システム側コンテンツファイルが、カプセル化して出来た XML 文書ファイルであり、そのファイルから構成される item のタイトルが目的システム側 item 名である。

item1

源システム側 item 名 : 概要

源システム側コンテンツファイル : AIC.txt

目的システム側 item 名 : fromtsukuba3

目的システム側コンテンツファイル : 3.xml

item2

源システム側 item 名 : あおあじ

源システム側コンテンツファイル : aoaji.jpg, aoaji.txt

目的システム側 item 名 : fromtsukuba2

目的システム側コンテンツファイル : 2.xml

item3

源システム側 item 名 : あおりいか

源システム側コンテンツファイル : aoriika.jpg, aoriika.txt

目的システム側 item 名 : fromtsukuba1

目的システム側コンテンツファイル : 1.xml

図 5.15 に item1 に対する源システム側 dublin_core.xml の記述例を示す。

```
<dublin_core>
<dcvalue element="contributor" qualifier="author">たけし, おかだ</dcvalue>
<dcvalue element="date" qualifier="accessioned">2004-11-24T06:25:08Z</dcvalue>
<dcvalue element="date" qualifier="available">2004-11-24T06:25:08Z</dcvalue>
<dcvalue element="date" qualifier="issued">2004-11-24T06:25:08Z</dcvalue>
<dcvalue element="identifier"
  qualifier="uri">http://hdl.handle.net/123456789/190</dcvalue>
<dcvalue element="description" qualifier="provenance">Submitted by Takeshi Okada
  (okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:25:08Z No. of bitstreams: 1 AIC.txt:
  141 bytes, checksum: 5137b69df0e6d8eae263f29632d8618d (MD5)</dcvalue>
<dcvalue element="description" qualifier="provenance">Made available in DSpace on
  2004-11-24T06:25:08Z (GMT). No. of bitstreams: 1 AIC.txt: 141 bytes, checksum:
  5137b69df0e6d8eae263f29632d8618d (MD5)</dcvalue>
<dcvalue element="format" qualifier="extent">141 bytes</dcvalue>
<dcvalue element="format" qualifier="mimetype">text/plain</dcvalue>
<dcvalue element="language" qualifier="iso">ja</dcvalue>
<dcvalue element="relation" qualifier="ispartofseries">魚;1</dcvalue>
<dcvalue element="title" qualifier="none">概要</dcvalue>
<dcvalue element="type" qualifier="none">Preprint</dcvalue>
</dublin_core>
```

図 5.15 源システム側 dublin_core.xml

図 5.16 にカプセル化した源システム側保存情報(item1)の記述例を示す。OAIS 参照モデルの情報モデルに従って XML 形式で記述している。図 4.5 に OAIS 参照モデルに基づいた記述例を示したが、ここでは図 4.5 を DSpace のフォーマットに書き換えたものになっている。図 5.16 中に license.txt というファイル名が記述されているが、これは DSpace からコンテンツを取り出した時にコンテンツに付与されてくる、コンテンツのライセンスに関するファイルである。

```

<?xml version="1.0" encoding="euc-jp" ?>
<AIP>
<CI>
<elementofCI>
<RI>license.txt</RI>
<CDO encode="base64Binary">TGljZW5zZSBncmFudGVkIGJ5IFRha2VzaGkgT2thZGEgKG9rY</CDO>
</elementofCI>
<elementofCI>
<RI>AIC.txt</RI>
<CDO encode="base64Binary">grGCzINSg4yDTtoNWg4eDk4LNk/qWe4LJgqKC6YubgsyCsoKti</CDO>
</elementofCI>
</CI>
<dublin_core>
<dcvalue element="contributor" qualifier="author">たけし, おかだ</dcvalue>
<dcvalue element="date" qualifier="accessioned">2004-11-24T06:25:08Z</dcvalue>
<dcvalue element="date" qualifier="available">2004-11-24T06:25:08Z</dcvalue>
<dcvalue element="date" qualifier="issued">2004-11-24T06:25:08Z</dcvalue>
<dcvalue element="identifier"
qualifier="uri">http://hdl.handle.net/123456789/190</dcvalue>
<dcvalue element="description" qualifier="provenance">Submitted by Takeshi Okada
(okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:25:08Z No. of bitstreams: 1 AIC.txt:
141 bytes, checksum: 5137b69df0e6d8eae263f29632d8618d (MD5)</dcvalue>
<dcvalue element="description" qualifier="provenance">Made available in DSpace on
2004-11-24T06:25:08Z (GMT). No. of bitstreams: 1 AIC.txt: 141 bytes, checksum:
5137b69df0e6d8eae263f29632d8618d (MD5)</dcvalue>
<dcvalue element="format" qualifier="extent">141 bytes</dcvalue>
<dcvalue element="format" qualifier="mimetype">text/plain</dcvalue>
<dcvalue element="language" qualifier="iso">ja</dcvalue>
<dcvalue element="relation" qualifier="ispartofseries">魚;1</dcvalue>
<dcvalue element="title" qualifier="none">概要</dcvalue>
<dcvalue element="type" qualifier="none">Preprint</dcvalue>
</dublin_core>
</AIP>

```

図 5.16 カプセル化した源システム側保存情報

図 5.17 にカプセル化した源システム側保存情報 (item1) に対する目的システム側 dublin_core.xml の記述例を示す。預け入れ準備プログラムにより作成する要素に下線をつけており、その他の要素は DSpace で自動的に付けられる要素となっている。ここで、"<dcvalue element="relation" qualifier="ispartofseries">series:fish</dcvalue>"はシリーズ名の要素であり、コレクション内の親子関係を維持するために AIU が AIC の名前を覚えていることを示している (図 4.10 参照)。ただし、4 章でも述べたようにこれは各システムが任意で行うことである。

```
<dublin_core>
<dcvalue element="date" qualifier="accessioned">2005-01-19T02:41:46Z</dcvalue>
<dcvalue element="date" qualifier="available">2005-01-19T02:41:46Z</dcvalue>
<dcvalue element="date" qualifier="issued">2005-01-19T02:41:46Z</dcvalue>
<dcvalue
element="identifier" qualifier="uri">http://hdl.handle.net/123456789/212</dcvalue>
<dcvalue element="description" qualifier="provenance">Made available in DSpace on
2005-01-19T02:41:46Z (GMT). No. of bitstreams: 1 3.xml: 4421 bytes, checksum:
2a75c235d8a34cd735e4a2825bb18ef1 (MD5)</dcvalue>
<dcvalue element="format" qualifier="extent">4421 bytes</dcvalue>
<dcvalue element="format" qualifier="mimetype">text/xml</dcvalue>
<dcvalue element="language" qualifier="iso">en_US</dcvalue>
<dcvalue element="relation" qualifier="ispartofseries">series:fish</dcvalue>
<dcvalue element="title" qualifier="none">fromtsukuba3</dcvalue>
</dublin_core>
```

図 5.17 目的システム側 dublin_core.xml (カプセル化した源システム側保存情報に対して)

図 5.18 に開示機能を実行する際にエクセルへインポートするファイルの記述例を示している。これはカプセル化した保存情報(item1,item2,item3)の接続ファイルであり、3つの保存情報を1つのXML文書にしている。また、CDOの代わりに各保存情報へのリンク情報を追加している。カプセル化した保存情報のCDOはコンテンツのビット列を文字列に変換したもので、コンテンツの種類によっては非常に長い文字列になってしまう。Excelでは文字列の上限を定めていることもありCDOの文字列は割愛する。その代わりに接続前の個々の保存情報のアドレスをリンク情報として接続ファイルへ追加することにした。個々の保存情報とはカプセル化されてできたXML文書であり、CDOの文字列も含まれている。開示機能を実行した人はRIやPDIにより表示したいコンテンツを決定し、リンク情報を頼りに実際に表示したいコンテンツの保存情報へアクセスする。リンク情報は開示プログラムの接続機能が、開示者により与えられた、開示するコレクションのアドレスから個々の保存情報のアドレスを解析し、接続ファイルへ付加する。接続ファイルのCDO要素にはCDOの文字列

の代わりに”look at the link”という文字列を格納し、新たに link 要素を追加した。図 5.19 に図 5.18 のツリー型データ表現を示す。

```
<?xml version="1.0" encoding="euc-jp" ?>
<collection>
  <item>
    <AIP>
      <CI>
        <elementofCI>
          <CDO>look at the link</CDO>
          <RI>license.txt</RI>
        </elementofCI>
        <elementofCI>
          <CDO>look at the link</CDO>
          <RI>AIC.txt</RI>
        </elementofCI>
      </CI>
      <dublin_core>
        <dcvalue element="contributor" qualifier="author">たけし, おかだ</dcvalue>
        <dcvalue element="date" qualifier="accessioned">2004-11-24T06:25:08Z</dcvalue>
        <dcvalue element="date" qualifier="available">2004-11-24T06:25:08Z</dcvalue>
        <dcvalue element="date" qualifier="issued">2004-11-24T06:25:08Z</dcvalue>
        <dcvalue
          element="identifier" qualifier="uri">http://hdl.handle.net/123456789/190</dcvalue>
        <dcvalue element="description" qualifier="provenance">Submitted by Takeshi Okada
          (okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:25:08Z No. of bitstreams: 1 AIC.txt:
          141 bytes, checksum: 5137b69df0e6d8eae263f29632d8618d (MD5)</dcvalue>
        <dcvalue element="description" qualifier="provenance">Made available in DSpace on
          2004-11-24T06:25:08Z (GMT). No. of bitstreams: 1 AIC.txt: 141 bytes, checksum:
          5137b69df0e6d8eae263f29632d8618d (MD5)</dcvalue>
        <dcvalue element="format" qualifier="extent">141 bytes</dcvalue>
        <dcvalue element="format" qualifier="mimetype">text/plain</dcvalue>
        <dcvalue element="language" qualifier="iso">ja</dcvalue>
        <dcvalue element="relation" qualifier="ispartofseries">魚;1</dcvalue>
        <dcvalue element="title" qualifier="none">概要</dcvalue>
        <dcvalue element="type" qualifier="none">Preprint</dcvalue>
```

```

</dublin_core>
</AIP>
<link>C:/1-19demo/hiroshima/1-19/1/3.xml</link>
</item>
<item>
<AIP>
<CI>
<elementofCI>
<CDO>look at the link</CDO>
<RI>license.txt</RI>
</elementofCI>
<elementofCI>
<CDO>look at the link</CDO>
<RI>aoaji.txt</RI>
</elementofCI>
<elementofCI>
<CDO>look at the link</CDO>
<RI>aoaji.jpg</RI>
</elementofCI>
</CI>
<dublin_core>
<dcvalue element="contributor" qualifier="author">たけし, おかだ</dcvalue>
<dcvalue element="date" qualifier="accessioned">2004-11-24T06:26:34Z</dcvalue>
<dcvalue element="date" qualifier="available">2004-11-24T06:26:34Z</dcvalue>
<dcvalue element="date" qualifier="issued">2004-11-24T06:26:34Z</dcvalue>
<dcvalue
element="identifier" qualifier="uri">http://hdl.handle.net/123456789/191</dcvalue>
<dcvalue element="description" qualifier="provenance">Submitted by Takeshi Okada
(okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:26:33Z No. of bitstreams: 2 aoaji.txt:
116 bytes, checksum: 870284a526529737f48a2690a3948797 (MD5) aoaji.jpg: 24501 bytes,
checksum: cc45238a2ca7b458671e29968b4dd792 (MD5)</dcvalue>
<dcvalue element="description" qualifier="provenance">Made available in DSpace on
2004-11-24T06:26:34Z (GMT). No. of bitstreams: 2 aoaji.txt: 116 bytes, checksum:
870284a526529737f48a2690a3948797 (MD5) aoaji.jpg: 24501 bytes, checksum:
cc45238a2ca7b458671e29968b4dd792 (MD5)</dcvalue>
<dcvalue element="format" qualifier="extent">116 bytes</dcvalue>

```

```

<dcvalue element="format" qualifier="extent">24501 bytes</dcvalue>
<dcvalue element="format" qualifier="mimetype">text/plain</dcvalue>
<dcvalue element="format" qualifier="mimetype">image/jpeg</dcvalue>
<dcvalue element="language" qualifier="iso">ja</dcvalue>
<dcvalue element="relation" qualifier="ispartofseries">魚;2</dcvalue>
<dcvalue element="title" qualifier="none">あおあじ</dcvalue>
<dcvalue element="type" qualifier="none">Image</dcvalue>
</dublin_core>
</AIP>
<link>C:/1-19demo/hiroshima/1-19/2/2.xml</link>
</item>
<item>
<AIP>
<CI>
<elementofCI>
<CDO>look at the link</CDO>
<RI>license.txt</RI>
</elementofCI>
<elementofCI>
<CDO>look at the link</CDO>
<RI>aoriika.txt</RI>
</elementofCI>
<elementofCI>
<CDO>look at the link</CDO>
<RI>aoriika.jpg</RI>
</elementofCI>
</CI>
<dublin_core>
<dcvalue element="contributor" qualifier="author">たろう, やまだ</dcvalue>
<dcvalue element="date" qualifier="accessioned">2004-11-24T06:27:33Z</dcvalue>
<dcvalue element="date" qualifier="available">2004-11-24T06:27:33Z</dcvalue>
<dcvalue element="date" qualifier="issued">2004-11-24T06:27:33Z</dcvalue>
<dcvalue
element="identifier" qualifier="uri">http://hdl.handle.net/123456789/192</dcvalue>
<dcvalue element="description" qualifier="provenance">Submitted by Takeshi Okada
(okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:27:33Z No. of bitstreams: 2

```

```

aoriika.txt: 142 bytes, checksum: a558ee224627bf72f61fdfd1dac22c5a (MD5)
aoriika.jpg: 20847 bytes, checksum: 2f45087b2b3f2b011f2ad95e3da2924b (MD5)</dcvalue>
<dcvalue element="description" qualifier="provenance">Made available in DSpace on
2004-11-24T06:27:33Z (GMT). No. of bitstreams: 2 aoriika.txt: 142 bytes, checksum:
a558ee224627bf72f61fdfd1dac22c5a (MD5) aoriika.jpg: 20847 bytes, checksum:
2f45087b2b3f2b011f2ad95e3da2924b (MD5)</dcvalue>
<dcvalue element="format" qualifier="extent">142 bytes</dcvalue>
<dcvalue element="format" qualifier="extent">20847 bytes</dcvalue>
<dcvalue element="format" qualifier="mimetype">text/plain</dcvalue>
<dcvalue element="format" qualifier="mimetype">image/jpeg</dcvalue>
<dcvalue element="language" qualifier="iso">ja</dcvalue>
<dcvalue element="relation" qualifier="ispartofseries">魚;3</dcvalue>
<dcvalue element="title" qualifier="none">あおりいか</dcvalue>
<dcvalue element="type" qualifier="none">Image</dcvalue>
</dublin_core>
</AIP>
<link>C:/1-19demo/hiroshima/1-19/3/1.xml</link>
</item>
</collection>

```

図 5.18 開示機能を実行する際にエクセルへインポートするファイル (カプセル化した保存情報の接続ファイル)

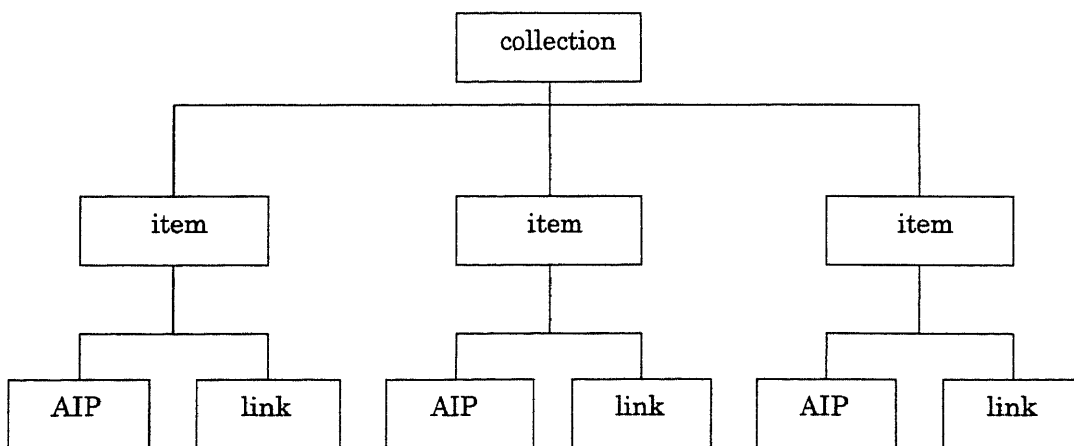


図 5.19 エクセルへインポートするファイルのツリー型データ表現

図 5.18 の XML 文書を Excel に取り込み、Excel で生成されたファイルを Access に取り込むと図 5.20 のような表が生成される。図 5.20 の 1 行目に ID,1,2,3,4,5,6,7 とあるがそれぞれ何を示すかについて説明する。なお、図 5.20 の 2 行目、3 行目は XML 文書でのディレクトリ構造を示している。2 行目にはルートとなる”/collection”を、3 行目には XML 文書の各データの要素名及び属性名を示している。ただし、”@”は属性を示している。まず ID は表中の各レコードを一意に識別するための番号である。”1”は各レコードがどの item に属するかを示すもので、”1”の値が等しいレコードは同じ item に属することを示す。”2”は CDO が <link>内にあることを示しており、”3”に記述されているファイル名の CDO の実体は”7”に示されているアドレスのファイル内に存在していることを示す。”4”は dcvalue 要素のデータを示しており、”5”がそのデータの element 属性を、”6”が qualifier 属性を示している。

ID	1	2	3	4	5	6	7
1	/collection						
2	/item/ /id	/item/A IP/CI/el ementof CI/CDO	/item/AIP /CI/eleme ntofCI/RI	/item/AIP/dublin_core/dcvalue	/item/AIP/d ublin_core/dc value/@elem ent	/item/AIP/d ublin_core/dc value/@qualif ier	/item/link
3	1	look at the link	license.txt				C:/1-19demo/hiroshima/1-19/1/3.xml
4	1	look at the link	AIC.txt				C:/1-19demo/hiroshima/1-19/1/3.xml
5	1			たけし, おかだ	contributor	author	C:/1-19demo/hiroshima/1-19/1/3.xml
6	1			38315.267454	date	accessioned	C:/1-19demo/hiroshima/1-19/1/3.xml
7	1			38315.267454	date	available	C:/1-19demo/hiroshima/1-19/1/3.xml

ID	1	2	3	4	5	6	7
8	1			38315.267454	date	issued	C:/1-19demo/hi roshima/1-19/1 /3.xml
9	1			http://hdl.handle.net/123456789 /190	identifier	uri	C:/1-19demo/hi roshima/1-19/1 /3.xml
10	1			Submitted by Takeshi Okada (okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:25:08Z No. of bitstreams: 1 AIC.txt: 141 bytes, checksum: 5137b69df0e6d8eae263f29632d8 618d (MD5)	description	provenance	C:/1-19demo/hi roshima/1-19/1 /3.xml
11	1			Made available in DSpace on 2004-11-24T06:25:08Z (GMT). No. of bitstreams: 1 AIC.txt: 141 bytes, checksum: 5137b69df0e6d8eae263f29632d8 618d (MD5)	description	provenance	C:/1-19demo/hi roshima/1-19/1 /3.xml
12	1			141 bytes	format	extent	C:/1-19demo/hi roshima/1-19/1 /3.xml
13	1			text/plain	format	mimetype	C:/1-19demo/hi roshima/1-19/1 /3.xml
14	1			ja	language	iso	C:/1-19demo/hi roshima/1-19/1 /3.xml
15	1			魚;1	relation	ispartofseries	C:/1-19demo/hi roshima/1-19/1 /3.xml
16	1			概要	title	none	C:/1-19demo/hi roshima/1-19/1

ID	1	2	3	4	5	6	7
							/3.xml
17	1			Preprint	type	none	C:/1-19demo/hi roshima/1-19/1 /3.xml
18	2	look at the link	license.txt				C:/1-19demo/hi roshima/1-19/2 /2.xml
19	2	look at the link	aoaji.txt				C:/1-19demo/hi roshima/1-19/2 /2.xml
20	2	look at the link	aoaji.jpg				C:/1-19demo/hi roshima/1-19/2 /2.xml
21	2			たけし, おかだ	contributor	author	C:/1-19demo/hi roshima/1-19/2 /2.xml
22	2			38315.268449	date	accessioned	C:/1-19demo/hi roshima/1-19/2 /2.xml
23	2			38315.268449	date	available	C:/1-19demo/hi roshima/1-19/2 /2.xml
24	2			38315.268449	date	issued	C:/1-19demo/hi roshima/1-19/2 /2.xml
25	2			http://hdl.handle.net/123456789/191	identifier	uri	C:/1-19demo/hi roshima/1-19/2 /2.xml
26	2			Submitted by Takeshi Okada (okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:26:33Z No. of bitstreams: 2 aoaji.txt: 116 bytes, checksum:	description	provenance	C:/1-19demo/hi roshima/1-19/2 /2.xml

ID	1	2	3	4	5	6	7
				870284a526529737f48a2690a39 48797 (MD5) aoaji.jpg: 24501 bytes, checksum: cc45238a2ca7b458671e29968b4 dd792 (MD5)			
27	2			Made available in DSpace on 2004-11-24T06:26:34Z (GMT). No. of bitstreams: 2 aoaji.txt: 116 bytes, checksum: 870284a526529737f48a2690a39 48797 (MD5) aoaji.jpg: 24501 bytes, checksum: cc45238a2ca7b458671e29968b4 dd792 (MD5)	description	provenance	C:/1-19demo/hi roshima/1-19/2 /2.xml
28	2			116 bytes	format	extent	C:/1-19demo/hi roshima/1-19/2 /2.xml
29	2			24501 bytes	format	extent	C:/1-19demo/hi roshima/1-19/2 /2.xml
30	2			text/plain	format	mimetype	C:/1-19demo/hi roshima/1-19/2 /2.xml
31	2			image/jpeg	format	mimetype	C:/1-19demo/hi roshima/1-19/2 /2.xml
32	2			ja	language	iso	C:/1-19demo/hi roshima/1-19/2 /2.xml
33	2			魚:2	relation	ispartofseries	C:/1-19demo/hi roshima/1-19/2 /2.xml
34	2			あおあじ	title	none	C:/1-19demo/hi

ID	1	2	3	4	5	6	7
							roshima/1-19/2/2.xml
35	2			Image	type	none	C:/1-19demo/hiroshima/1-19/2/2.xml
36	3	look at the link	license.txt				C:/1-19demo/hiroshima/1-19/3/1.xml
37	3	look at the link	aoriika.txt				C:/1-19demo/hiroshima/1-19/3/1.xml
38	3	look at the link	aoriika.jpg				C:/1-19demo/hiroshima/1-19/3/1.xml
39	3			たろう, やまだ	contributor	author	C:/1-19demo/hiroshima/1-19/3/1.xml
40	3			38315.269132	date	accessioned	C:/1-19demo/hiroshima/1-19/3/1.xml
41	3			38315.269132	date	available	C:/1-19demo/hiroshima/1-19/3/1.xml
42	3			38315.269132	date	issued	C:/1-19demo/hiroshima/1-19/3/1.xml
43	3			http://hdl.handle.net/123456789/192	identifier	uri	C:/1-19demo/hiroshima/1-19/3/1.xml
44	3			Submitted by Takeshi Okada (okadatak@slis.tsukuba.ac.jp) on 2004-11-24T06:27:33Z No. of bitstreams: 2 aoriika.txt: 142	description	provenance	C:/1-19demo/hiroshima/1-19/3/1.xml

ID	1	2	3	4	5	6	7
				bytes, checksum: a558ee224627bf72f61dfd1dac22 c5a (MD5) aoriika.jpg: 20847 bytes, checksum: 2f45087b2b3f2b011f2ad95e3da2 924b (MD5)			
45	3			Made available in DSpace on 2004-11-24T06:27:33Z (GMT). No. of bitstreams: 2 aoriika.txt: 142 bytes, checksum: a558ee224627bf72f61dfd1dac22 c5a (MD5) aoriika.jpg: 20847 bytes, checksum: 2f45087b2b3f2b011f2ad95e3da2 924b (MD5)	description	provenance	C:/1-19demo/hi roshima/1-19/3 /1.xml
46	3			142 bytes	format	extent	C:/1-19demo/hi roshima/1-19/3 /1.xml
47	3			20847 bytes	format	extent	C:/1-19demo/hi roshima/1-19/3 /1.xml
48	3			text/plain	format	mimetype	C:/1-19demo/hi roshima/1-19/3 /1.xml
49	3			image/jpeg	format	mimetype	C:/1-19demo/hi roshima/1-19/3 /1.xml
50	3			ja	language	iso	C:/1-19demo/hi roshima/1-19/3 /1.xml
51	3			魚:3	relation	ispartofseries	C:/1-19demo/hi roshima/1-19/3 /1.xml

ID	1	2	3	4	5	6	7
52	3			あおりいか	title	none	C:/1-19demo/hiroshima/1-19/3/1.xml
53	3			Image	type	none	C:/1-19demo/hiroshima/1-19/3/1.xml

図 5.20 Access に図 5.18 の XML 文書を読み込んで生成される表

XML 文書を図 5.20 のような表にした後、Access のクエリ機能を使って開示を行う。図 5.21 に開示結果の例を示す。図 5.21 は”dcvalue”の”qualifier”属性が”author”であるレコードの一覧である。結果の一覧には ID、属性値である”author”、属性値が”author”である”dcvalue”要素の値、CDO の実体がある場所のアドレスを出力している。この他にも”author”が”たけし, おかだ”であるレコードを検索したり、検索結果の出力を編集したりと様々な検索を行うことが出来る。

ID	6	4	7
5	author	たけし, おかだ	C:/1-19demo/hiroshima/1-19/1/3.xml
21	author	たけし, おかだ	C:/1-19demo/hiroshima/1-19/2/2.xml
39	author	たろう, やまだ	C:/1-19demo/hiroshima/1-19/3/1.xml

図 5.21 Access による開示結果の例

5.5 プロトタイプシステムの運用結果

前節までに述べたように、連携保存モデルの預けから返却の一連動作及び開示機能をプロトタイプシステムで実証できた。また全ての作業をコレクション単位で一括して行うことにより現実的な実証を行うことが出来た。

(注 5.1) Eclipse Platform2.1.1

IBM が開発したオープンソースの統合開発環境である。プラグインにより、java や C など多種言語の開発環境として利用できる。

(注 5.2) PostgreSQL

オープンソースのリレーショナルデータベース管理システム

(注 5.3) jsp

java server pages の略。java 言語を利用して web サーバーで動的に web ページを生成し、クライアントに送信する技術。html ファイルの中に Java プログラムを埋め込んでおき、クライアントの要求に応じてプログラムを実行、処理結果のみをクライアントに送信する。サーブレットコンテナ上で動作する。

(注 5.4) サーブレットコンテナ

サーブレットや jsp を実行するためのアプリケーションサーバー

(注 5.5) Tomcat

オープンソースのアプリケーションサーバー

(注 5.6) MIME コンバート DLL

文字列から Base64Binary 形式への変換及びその逆変換を行う為のダイナミックリンクライブラリ。(株) 日本アートギアシステムズ

ライセンス番号 : 001-0501210001

6. まとめ

本研究では OAIS 参照モデルに基づいてデジタルアーカイブシステム間連携による長期保存のモデルを考案した。本研究では源システム側で預けたい保存情報の内容情報と保存記述情報をカプセル化して目的システム側の内容データオブジェクトとして預け、目的システム側ではその情報に表現情報と保存記述情報をつけて保管するという手法を提案した。その手法を用いて源システムから目的システムへ保存情報を預け、目的システムから源システムへ預かった保存情報を返却するという一連の動作について言及した。さらに、カプセル化した保存情報を覗き見るモデルについても言及した。カプセル化には XML 技法を採用し、1つの保存情報を1つの XML 文書に格納することとした。

本研究で考案した連携保存モデルはコレクション単位で一括して処理するものとなっているため、実現には OAIS に保存情報の一括格納及び一括取出しの機能が必要である。それら機能は標準機能として備わっていると考えられる。OAIS 参照モデルで AIC を定義している以上、OAIS はコレクション単位で保存情報を出し入れする機能を持つと考えるのが一般的である。

さらに考案した連携保存モデルのプロトタイプシステムを DSpace を用いて開発し、サンプルデータを用いてモデルの検証を行った。プロトタイプシステムを実際に開発し、預けから返却の一連の流れ及び開示機能を実行することで考案したモデルが実際にシステム化して運用可能であることを確認した。また、プロトタイプシステムの開発はモデルの形成過程・精緻化において有益であった。

OAIS は長期保存を実現するシステムである。しかし1つの OAIS システムが単独で保存情報を保存するよりも他の OAIS システムと連携して保存する方が長期保存性が高まることは自明である。保存情報を預けておくことで、源システムが災害等で保存情報を消失したときに目的システムから預けている保存情報を返却してもらうことが可能である。さらに、源システムが何らかの理由で運用を終了した際には預けている保存情報を目的システム側で提供してもらうことも可能である。しかし、保存情報を預けることの最大の利点は別の所にある。それは源システムが預けた保存情報のことを忘れてしまった遙か未来に、目的システムが源システムから預かった保存情報を発見しそれを解読し、世の中にそのコンテンツが提供されることである。カプセル化された源システム側保存情報は XML 文書化されており、その中には文字列化されたコンテンツの他にビット列から文字列への変換に用いた手法、保存記述情報や表現情報といったメタデータが記述されている。さらにカプセル化された源システム側保存情報には目的システム側で表現情報と保存記述情報が付けられている。このような保存情報に解読方法を記述した文書や開示・開封を実行するプログラムを併せて預けることになる。遙か未来、預かった保存情報が発見されたとき、これらの情報を元に目的システム側の人々が預かった保存情報を解読することになる。解読のために必要な情報をコンテンツと共に目的システム側へ将来解読できる形態で預ける手法を考案した点が本研究の最大のポイントである。

7. 謝辞

本研究を進めるにあたり共同研究をした指導教官田畑孝一教授に深く感謝を申し上げます。また、有益なアドバイスをいただいた杉本重雄教授、阪口哲男助教授、森嶋厚行助教授、永森光晴講師、本研究室の皆様に深くお礼申し上げます。

8. 参考資料

[1]CCSDS,Reference Model for an Open Archival Information System(OAIS)Blue Book,Issue1.

<http://ssdoo.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf>

(アクセス日 : 2005/01/19)

[2]機関リポジトリ構築ソフトウェアガイド 第3版

http://www.nii.ac.jp/metadata/irp/osi_guide_3/#DSpace

(アクセス日 : 2005/01/19)

[3]CERN,CDSware HomePage

<http://cdsware.cern.ch/>

(アクセス日 : 2005/01/19)

[4]MIT Libraries,Hewlett-Packard Company,DSpace Homepage.

<http://www.dspace.org/>

(アクセス日 2005/01/19)

[5]University of Southampton,eprints HomePage

<http://www.eprints.org/>

(アクセス日 : 2005/01/19)

[6]i-Tor HomePage

<http://www.i-tor.org/en/>

(アクセス日 2005/01/19)

[7]MyCoRe HomePage

<http://www.mycore.de/content/below/index.xml?lang=en>

(アクセス日 2005/01/19)

[8]GreenStone HomePage

<http://www.greenstone.org/cgi-bin/library>

(アクセス日 2005/01/19)

[9] OCLC/RLG WorkingGroup,A Metadata Framework to Support the Preservation of Digital Objects

http://www.oclc.org/research/projects/pmwg/pm_framework.pdf

(アクセス日 2005/01/19)

[10]PostgreSQL

<http://www.jp.postgresql.org/>

(アクセス日 2005/01/19)

[11]The Apache Jakarta Project

<http://jakarta.apache.org/tomcat/index.html>

(アクセス日 2005/01/19)

[12]JavaMail

<http://java.sun.com/products/javamail/index.jsp>

(アクセス日 2005/01/19)

[13]国立国会図書館,電子情報保存に係る調査研究報告書

http://www.ndl.go.jp/jp/aboutus/preservation_02_01.html

(アクセス日 2005/01/19)

[14]DSpace Technology & Architecture

<http://www.dspace.org/technology/architecture.pdf>

(アクセス日 2005/01/19)

[15]栗山正光.長期保存型電子図書館と OAIS 参照モデル

<http://www.tulips.tsukuba.ac.jp/pub/dlsympo/hobunshu/kuriyama.pdf>

(アクセス日 2005/01/19)

[16]安田悦子.デジタルアーカイブシステム間の連携保存に関する研究.筑波大学.2004.学士論文

[17]石井康子.デジタルアーカイブシステムの連携利用に関する研究. 筑波大学.2004.学士論文