1997年度卒業研究 I
Retention Scheduleを用いた電子文書の管理
— 電子的文書館の構築に向けて —

東京大学教養学部基礎科学科第二
川合研究室 4年
61204I 伊戸川 暁

1997年11月21日

概要

人類の記憶を伝えるべき機関として、「文書館」というものがある。文書館の専門職たるArchivistは、歴史を後世に伝えるという重大な任務を負っている。

　また、紙ベースの文書管理においては、今世紀に入ってからの文書量の飛躍的増大に伴い、作成時から一貫した計画を立てて、最終的に残る文書の質や量をコントロールしようとする考え方が現れるに至った。この「計画」のことを「Retention schedule」という。

　一方、世界の情報化に伴い、歴史的に重大な意味を持つであろう情報は、これまでになく磁気ディスクなどの電子的な方法で記録されるようになった。しかし、磁気媒体には固有の寿命があるため、そのままでは永久保存には用いることができず、また、電子情報の運用技術についても、「フローとしての情報」という経済的効率の観点から語られることが多く、「ストックとしての情報」を文化財として蓄積・保存しようとする観点からの議論は少ない。

　本研究では、現代のこのような状況を踏まえつつ、「Retention scheduleに基づいた文書管理」、及び「史料としての文書」のアイディア/モデルを電子文書に持ち込み、コンピュータの上ではどのようにすればそれらが実現できるかについて考察する。

　著者は紙文書と電子文書との比較考察の結果、紙文書を基にして考えられた記録管理学のモデルの電子文書において改められるべき点を見出し、また、電子文書においては公開と保存との間に密接な関係のあることを明らかにした。

　更にそれらの結論からシステムの満たすべき要件について議論し、その上で、それらの議論に基づいた文書管理システムを提案した。

　具体的には、retention scheduleを含むメタデータ(文書に関する情報)を記述したファイルを文書そのものと並行して対象ファイルとは別の領域に記録し、定期的にretention scheduleを確認して自動的にファイルを削除や公開などを行うものである。

　加えて、その提案に基づいた簡単な実装を、実験的にSunOS4+Perl5.003上で作成してみた。

目次

第1章　序論

1.1　文書館とは

まず、論題にも現れた「文書館」なるものについて、解説をする。

「文書館 (Archives)」とは、「人々が生産した一次記録のうち、歴史的に重要と思われる記録を保存し、後世に伝えることを目的とする施設」である。

「一次記録」とは、「文字記録のうち、人々の活動から直接生産され、原則として写しが存在しないもの」のことであって、企業であれば、日誌・帳簿・辞令などがこれに相当する。

これに対して、「一次記録をもとに編集され、原則として複数の写しが存在するもの」を「二次記録」といい、やはり企業であれば、社史や会計報告書がこれに該当する。

文書館は、いわゆる「古文書」のみを扱う機関ではない。現在現役で活用されている書類も、数十年の後には歴史的資料になり得るという理由で、やはり文書館の関与すべき範囲内にある。

無論、文書館に相当する施設は古来から存在していた。けだし人類は、文字を発明して以来、歴史的記録の保存に勤めてきている。しかし、「国民一般に公開する」ことを原則とする近代的文書館の歴史は、フランス革命を以てその嚆矢とする。

フランス国民議会は、1789年に議会の記録を保存する施設を置き、これは翌年に国立文書館 (Archive Nationale) と命名された。次いで1794年の法令により、ここに旧体制及び新体制の全ての政府記録を集めることと、更に「すべての国民は、あらゆる文書館で記録の閲覧を請求することができる」こととが定められた。そして1796年には、中央及び各地方の文書館システムが制定されたのである。[1, pp456–457]。

1.2　記録管理学の起こり

かつては、文書館に納められる文書は、いちいち archivist [1]が 使用済みになった文書から 歴史的に重要と思われるものを選別していた。しかし、近代化の進行に伴う文書量の爆発的な増大により、[2] このような方法では限界のあることが明らかになった。

また、受け入れた文書記録を、文書自体に内在しない外部的な基準により断裁的に分類してしまったのでは、かえって史料として利用しづらいことも徐々に明らかになっていった。

---

[1] 文書館に関係する専門職。図書館における司書・博物館における学芸員に相当する。日本語での定訳は今だ定まっていないので、原語のままとした。

[2] 文書量の増大についてのデータの一例。アメリカ合衆国政府の生産した文書記録の体積は、建国から南北戦争までが約10万立方フィート、南北戦争から第一次世界大戦までが約150万立方フィート、第一次世界大戦から世界恐慌までが約350万立方フィート、世界恐慌から第二次世界大戦の直前までが約1000万立方フィート、そして第二次世界大戦期には、作成量は毎年200万立方フィートずつ増加したという (Schllenberg(1956) による)。

こうした状況の中、「効率的に文書の選別を行うにはどのようにしたらよいか?」といったことが理論的に考えられるようになり、記録管理に関する諸理論が案出されるに至った。

本研究で用いる記録管理の理論モデルは、今世紀中葉にアメリカで確立し、現在でも世界各国で用いられている、「記録のライフサイクル説」に基づいている。この説は、1937年に設立されたアメリカ合衆国立公文書館によって提唱され、間もなくこれに基づいた文書管理法制がアメリカ国内で施行され、しかる後に各国に広まっていったものである。

「記録のライフサイクル説」、及び文書管理学の諸概念の詳細については、第2章で記す。

1.3　わが国における文書館の状況

諸外国では文書館の機能は社会的によく認知され、それが国家をはじめ各組織のaccountabilityを明らかにするところから、「民主主義の礎」とまで考えられている。[11]

しかしわが国では、悲しいことに、社会的認知すら殆んど存在しない状態が続いている。

もっとも関係者は皆無ではなく、戦後間もない頃から歴史研究者を中心として、文書館制度の確立のための地道な努力が続けられており、1967年に国立公文書館を設置させ、1987年には「公文書館法」を制定させ、そして1997年現在、26都道府県などに文書館を建てさせるに至っているが、その成果は国民一般には余り知られていない。

1.4　電子文書の置かれている現状

翻って、電子文書のおかれている状況について概説する。

今世紀中葉にコンピューターが発明された当初は、その用途は弾道計算など、計算を大量かつ高速に行うものでしかなかったし、またそれ以外の使用法などは考えられもしなかった。コンピューターの歴史が始まったか始まらないかの頃、Vannebar Bushが“Memex”という、現在でいうところのハイパーテキストシステムの元祖のようなものを考案したが、それで閲覧できる文書は、マイクロフィルムに記されるものとされていた。

しかし、記憶容量の飛躍的な増大に代表される種々の技術革新の結果、コンピューターは文書処理には欠かせないものとなり、パソコン或いはワープロの類を備えていない事務所は、稀な存在になるまでに至った。

更に、1990年代半ばに起こったインターネットの急激な普及により、ハイパーテキストシステムは電子的な形態で実現され、電子的情報は、容易に個々のコンピューターを超えて世界的規模で流通できるようになった。

これらの一連の過程が世界を変えた様子は、しばしばグーテンベルクによる印刷術の発明と比較される。

しかしながら、電子文書の前途は決して安泰なものではない。

まず、文書における「電子革命」が現在も進行中であるがゆえに、ハードウェアがすぐに古びてしまうという問題がある。例えば、8インチのフロッピーディスクを持っていても、現在、これを読み出せるハードウェアは殆ど無い。

更に、電子メディア自体の耐久性の問題もある。現在ハードディスクなどに広く用いられている磁気メディアには、長い目で見ると磁気が徐々に劣化していき、ついには中身の情報が失われてしまうという、致命的な欠点がある。また、CD-ROMなどの非磁気的メディアも、決して将来数十~数百年にわたる耐久性が保証されているわけではない。では紙にプリントアウトすればよいかとなると、これも怪しい。プリンタが使用している用紙及びインクの耐久性も、やはり必ずしも数十~数百年ものスケールで確認されているものではないからである。

しかし、現代社会の重要な情報の多くは、間違いなく磁気テープ、或いはハードディスクの中に眠っているのである。

このような状況にもかかわらず、電子情報をめぐる研究の大勢は、マルチメディア情報の伝達がどうであるだとか、或いは電子商取引(Electronic Commerce)によって経済がどうする、といったような、「フローとしての電子的情報」に偏しているきらいがあり、「ストックとしての電子的情報」を扱った研究は、ともすると軽視されがちである。[3]

1.5 序論のまとめ

将来に向けて、電子文書の重要性はいや増すばかりであり、以上に述べた問題は放置しておくわけにはいかない。

また、文書の形態が大きく変化しつつある現在こそ、記録管理の考えを日本に根付かせ、以て薬害エイズ事件の際の「資料はございません」のような官僚の隠蔽体質をわが国から一掃する好機であるとも、著者は考えている。

そうした動機をもって、著者は、「将来にわたって電子文書を適切に管理・保存するにはどうすればよいのか?」について、研究を行うことにしたのである。

---

[3] 少なくとも著者にはそう感じられてならない。

第2章　予備知識

2.1　文書のライフサイクル説

前章で述べた通り、現代の記録管理学に於いては、文書には「ライフサイクル」があると観念されている。

文書のライフサイクル説によると、文書は次の各段階を経る。

現用 (current)　「活性 (active)」ともいう。文書が作成され、現役で使用されている状態をいう。業務の円滑な遂行のため、ただちにアクセスできる状態にある必要がある。

半現用 (semi-current)　差し当たっては必要ではないが、法律上その他の理由により、必要性が完全には無くなっていない文書の状態。これに該当する文書は保存コストを低く抑えることが優先される。その際、アクセス速度は多少犠牲にしてもよい。具体的には、この状態にあると判断された文書は、保存コストの安い倉庫など (「記録センター」という) に移されることになる。

現用と半現用の区分は実用のためにある区分であり、その分水嶺は「使用頻度」にあるとされている。

非現用 (non-current)　「不活性 (non-active)」ともいう。実務上の必要性が既になくなった文書の状態をいう。この状態になった文書は文書館などの施設に送られ、歴史的資料[1]としての「第二の人生」を歩むことになる。

2.2　文書の価値

文書の価値には、大別して「一次的価値」と「二次的価値」とがある [1]。

両者の内容は次の通り。

一次的価値　「当該機関が業務を遂行する上での文書の必要性」。

例えば、人に貸しがある時の借用証書は、一次的価値を有する。

この価値は、現場の人間が最も良く知っていることだろう。

一次的価値は、業務価値・法務価値・財務価値の 3 つに細分されることもある [9]。

二次的価値　「ある組織の過去の出来事を文書化するための文書の重要度または有用性」(ARMA の定義による)、或いは「文書の作成当初の目的以外の価値」。

---

[1] 以下、しばしばこの意味で「史料」という語を用いることがある。

> 例えば、取り立てが完了した後、かつて自分が誰に金を貸したのかを調べる際に、前述の借用証書は二次的価値を帯びることになる。
>
> 二次的価値の判断には、一次的価値とはまた別の評価の基準が必要になってくる。そしてこの価値の評価を行うところにこそ、archivist の存在理由があるとされている。

重要なのは、「文書の価値は実用だけにあるのではない」という点である。

故に、retention schedule (次節) の策定に当たっては、一次的価値のみならず、二次的価値にも留意しなければならない。

2.3 Retention Schedule

前述したように、今世紀の初頭あたりから、archivist が処理しなければならない記録の量は増加の一途を辿った。そこで、前 2 節の枠組みに基づいて、「文書の作成時に前もって一次的及び二次的な文書の価値を判断しておけば、archivist の仕事が軽くなるのではないだろうか」というアイディアが現れた。かくして、「文書の作成の地点から archivist が介入し、文書の流れをコントロールする」という、retention schedule の考えが出現したのである。

紙文書の場合、

1. 文書の重要度
2. 保存年限
3. 保存後の処置

などを 各々の recode series(定義は後述) に対して指定し、各文書は この計画に従って、記録センターや文書館などに移され、或いは廃棄されることになる。

記入の具体例は、「仕入伝票:決算時までオフィスで保有の後地下倉庫に移動、更に 3 年間保存した後に廃棄」という感じになる[2]。

retention schedule の策定は、一次的価値と二次的価値の両方に関わるので、実務の当事者や archivist など、当該文書に関係する全ての人々の協議の上でなされるべきである。

元々、retention schedule の考えは、増大する一方の文書をいかに効率的に処理するかというところから生じたが、現在では、文書の絶対量を減らすのみならず、事務の効率化にも有益であることが認められている。

2.4 その他の定義

最後に、この論文にしばしば出てくるであろう幾つかの重要な概念に対して、簡単な定義と解説を加えておく。

記録 この論文では、「文書の集合体」くらいの意味で用いていが、。しばしば「文書」と相通用するように使用している場合がある。

---

[2] むろん架空の例

史料　歴史的価値を有すると見なされた記録。

record series　同一の retention schedule に従う文書の集合 [9]。

記録の有機性　そもそも、文書は決して一枚のみで存在するものではない。通常、文書は幾つか集まってフォルダを形成し、更にフォルダが幾つか集まって上位の階層を作り、……という構造を為す。従って、個々の記録は、他の記録を含めた全体の一部であり、一枚の文書を、それが置かれている記録全体から切り離して論ずることは出来ない。

現秩序保存の原則　とにかく文書に付随するあらゆることが情報を持っていると考えて間違いない。

文書の文面は当然のこと、他の文書との関連、紙の材質や大きさ、はては余白の広さまでが意味を持ち得る。従って、記録自体に内在しない基準で記録を解体し、分類することは、良くないことと見做されている[3]。事実、古文書の巻物の結び目すら、解いて読み終わったら元通りの結び方で結び直すべし、とされているくらいである [14]。

文書の所在地もまた、文書に付随する情報の一部であるといえる。ここから、「文書はその発生元で保管されるべし」という考えが出てくる。これを特に「現地保存の原則」という。

---

[3] 実は昔は大いに行われていたのだが、後にその弊害が広く認められるに至ったのである。

第3章　電子文書に於いて注意すべき点に関する議論

この章では、電子文書の保存の上で何が問題になるかを示し、その解決策を探るとともに、既に存在している関連研究についても概説する。

3.1　本研究の方針

まず最初に、本研究の方針について、二・三述べておく。

- 本研究では、「今後作られる電子文書」に焦点を絞ることとし、現存(現在)の古文書・紙ベースの文書・旧式の電子文書については扱わない。即ち、対象として想定している文書は、およそ次のような性質を有するものである。
    1. 少なくとも数十年は存続するであろう組織の文書。
    2. 公共性を有するもの。（例：NetNews、ある種のMailing-list、公的機関の事務文書）
       公的機関が最終的には自らの記録を公開せねばならないのは「公僕」として当然の義務であるが、その点、私的団体の取り扱いは多少厄介である。私的文書には、著作権があったり、privacyの問題があったりするからである。しかし本研究では、「最終的には全て時間が解決する」という いささか大雑把な考えのもとに、そのあたりの問題は捨象してしまうこととする。
    3. 継続的に生産される文書。
       この際、Web-pageのような、当初から公開を目的として作成される文書も対象に含める。何故なら、それらは一般に「記録」として提示されるものではないので、記録として公開するのとは意味が異なっているからである。
- また、本研究では、software的に解決できる問題に焦点を絞る。
  本論文に述べる問題を解決するにあたっては、本質的にはハードウェア的な解決を見なければならない点も多い(例えば、磁気記憶媒体の耐久期間の問題など)。しかし本論文では、ハードウェアの発展を待たずに、現在ある技術の範囲内で問題を解決する道を探ろうと思う。

3.2　紙文書と電子文書の相違

さて、紙文書と電子文書の相違について考察を加えることにする。

体積の相違　既に言い古された感のある話ではあるが、例えば1枚のCDに収まる情報量は約540メガバイトで、A4の紙に換算すると18万枚にもなり、その体積比は1万倍以上にもなる。DVDならば、この比は更に大きくなる。

従って、電子文書に対するretention scheduleの存在意義は、「容積の節約」よりはむしろ「大量の文書を効率良く捌く」ことに主眼を置くことになるであろう。しかし、計算機の記憶容量が増えるに従って、人々は文字情報から画像・音声・動画へのように、より容量が必要な情報を扱わせるようになった、というのは厳然たる歴史的事実であって、「記憶容量を使い果たす技術は記憶容量の増大に伴って発達する」という皮相な説もあるほどである。故に、容量の増大は「昔より少し余裕ができた」くらいに捉えるべきであって、「容量の節約」という観点は、決して忘れていいわけではないと考察する。

複製が容易　基本的に計算機は、単純な作業を何兆回でも喜んでするところにその特徴がある。

改変が容易　改竄の後にいかなる痕跡も残さないのが、電子情報の特徴である。原理的には、それが0と1で表現できる限り、元の文書の文章をいかなる表現で置き換えることもできる。

これは文書管理にとって、明らかに短所となる。業務上必要な文書を改竄するわけにはいかないので、「業務領域→記録領域」はさほどの問題はないが、史料情報が流通する際に改竄され得るのが、最大の問題となる。

一次記録と二次記録との境目がぼやける　電子文書はいくらでも複製できるわけだから、電子的に公開されている限り、一次記録と二次記録(→1.1節)を「部数」で区別することは困難となる。これらはもはや「出自」で区別を立てるしかあるまい。

以上の件と、第2章で示した「文書館の要請」とが、議論の前提となる。

3.3　メタデータの必要性

文書管理の上では、文書の作成時刻、その文書が所属するrecord seriesのretention scheduleなど、文書自体以外の情報— このような「データに関するデータ」をメタデータ(**metadata**)という — を扱う必要がある。しかし、「現秩序保存の原則」に従うならば、archivistの都合だけで、文書自体に関連情報を書き込むわけにはいかない。また、ディレクトリに関する情報など、いずれのファイルに属するか決め難い情報もある。

従って、メタデータを記述する枠組が、電子的文書館システムには不可欠となる。

3.4　媒体管理の難問、及びそれのnetworkの利用による解決案

3.4.1　従来からの案

現在、コンピュータの扱う情報は、その殆どが磁気テープ・磁気ディスクなどの「磁気媒体」の形で記憶されているが、この記憶方法自体が、実は大問題なのである。

第1章にも少し書いたが、磁気記憶は、その天然として、わずか数年から数十年程度で磁気が薄れ、ついには消滅してしまうのである。

「原秩序の保存」をやかましく言う各国の Archivist も、磁気媒体は「一時的なもの」と見做すことで落ち着いているようである [8]。

勿論、主な記憶媒体の全てが磁気的である訳ではなく、CD-ROM は磁気的な記録ではない（物理的な凹凸によっている）のでより寿命が長いと考えられるが、まだ充分な「時の試練」を受けていないので、早急に判断を下すのは慎まねばならない。

つまり、磁気媒体の寿命以上に電子記録を生かそうとするならば、どうしても定期的に媒体の内容を異なる媒体に移していく作業が必要になる。情報の中身のみを別の器に移すというのは「現秩序保存原則」に反するが、そのまま放置すれば最も重要な情報自体が失われてしまうのであるから、止むを得ないとするしかない。[1]但しこの際には、媒体の遍歴の記録を取る必要があるであろう。

しかし、このような「式年遷宮」をやるとしても、以下の選択が残されている。

1. 別の種類の媒体に移すことはせず、読み出しに必要なハード (とソフト) も保存する。
   現在オランダ政府が計画している "Digital Repository" なるプロジェクトは、各種媒体に収められた電子情報は勿論、それらの読み出しに必要になるハードウェアをも保存しようというものである [6]。
2. すでに安定性が確認されているものに書き出してしまう。
   要するに、紙やマイクロフィルムに出力してしまうということである。いかにも、これも安全確実な方法ではあるが、電子文書の容量は莫大なものであるから、本当に重要なごく一部の文書に対してしか実施できないという難点があるし、電子情報の持つ柔軟性も、print out によって、当然のことながら失われてしまう。
3. 新しい媒体が開発され次第、そちらに情報を移していく。
   「情報そのものを未来に伝える」点を重視した場合、このような選択もありうる。

3.4.2 Network 案

さて、以上の案に対し、ここで著者は、ネットワークの積極的な活用による解決策を提案するものである。

Internet に於いては、「プロトコルが標準化されている」という点が重要である。

今後のコンピューターは、ネットワークに接続されるのが常態となることが予想される。外界と自由に情報を交換できるメリットは極めて大きいからである。

そして、ネットワークの上を同じプロトコルに従って流れている限り、全ての 0 と 1 は同じ形をしている。そうであるならば、問題は純粋に「データ形式の問題」のみとなる。

つまり、どのような 0 と 1 の並びを使用するかだけ決めておけば、あとはその 0 と 1 が物理的にどのような形になるかについて、深く悩む必要が無くなるということである。全く無くなる、とは言わないが、少なくなることは確かだろう。

無論、プロトコルは時代に従って変化し得る。しかし、プロトコルが変更される場合も、それは連続的に変更されるものであって、ある日突然、従来のものと全く互換性のないプロトコルが使用

---

[1] ある特定の媒体を用いていることが重大な意味を持つこともありうるので、このあたりは実に頭の痛い選択である。

されるという事態は、非常に考えづらい。[2]

以上をまとめると、ネットワークにおいて広く用いられる標準は、空間軸のみならず、時間軸においても、異なる種類のコンピューター同士を結び付けることができると考えられるのである。

従って、「公開と保存の結び付き」ということは、紙文書についても言われていたが、電子文書に於いては、ますますその結び付きが強調されることになる。

また、ネットワーク上に史料情報を流すことによって、ダウンロードによるcopyが増え、それだけ史料消滅の可能性が減少することも期待できる。

3.5 アプリケーション及びOSからの データの独立性

しかし、全ての0と1の互換性が保証されても、データの形式が特定のソフトウェアに依存していれば、やはりそのデータは「読めなく」なってしまうので、できるだけ将来にわたっても普遍的であると考えられる形式で、データを記述する必要がある。例えば、Word Perfectや一太郎などの特定のワープロソフト[3]で記された文書は、当該のソフトウェアが廃れてしまった後には読み出すことが著しく困難になることが予想されるが、plain textで記された文書ならば、その文書で用いられているコード系の使用が続いている限り、簡単に読むことができる。

3.6 集中管理の可能性

電子文書は複製が容易であり、また、オリジナルとコピーは、どの媒体にあるかをを除けば、本質的に同等である。

そもそも紙文書のretention scheduleに文書の移動に関する項目があったのは、正本を適切な場所(記録センターなど)に移動する必要があったからである。

また、紙文書でオリジナル(正本)が重視されているのは、オリジナルは基本的に1つしかない上に、いかなる種類の複写もオリジナルの持つ情報を完全には写し得ないからである。

しかし電子文書の場合、媒体の違いを除けば、オリジナルとコピーは全く同一である。従って、オフィスから倉庫に文書を移動するという手間をかけなくとも、複写によって十分に文書の管理を果たすことができる。加えて、複写を行うことによって業務を妨げることもない。

よって、「非現用」と他の2者はともかく、「現用」と「半現用」の文書については、一元的な管理の可能性が開けることになる。

3.7 関連する先行研究

最後に、本研究に関係のある先行研究及びプロジェクトを幾つか紹介しておく。

---

[2] 但し、プロトコルの変化に追従するには、ちょくちょくネットワーク上に史料情報を流すようにしなければならない。何十年も放っておいたのでは、プロトコルの流れに置いていかれてしまうだろう。

[3] もっとも近頃は、こうしたワープロソフトも、SGMLフォーマットをサポートするなど、データの互換性には気を遣うようになってきているのだが……。

The Internet Archives B.Kahle は、全世界の Web-pages を収集・保存するという事業を行っており [7]、精神においては共通する部分もあるが、主に対象とする文書が異なるし (向こうは主に Web 文書、こちらは主に内部文書)、それ以前に著者は、収集・保存をば人任せにすべきではないと考えているので、このプロジェクトとは袂を分かつ。

商用システム 商品として類似のことを行うシステムは、豪州[4]Towersoft 社の「TRIM」をはじめ、既に幾つか存在しているが、文書館的機能は比較的ビジネスに特化したこれらのパッケージの一部として組み込まれている場合が一般的であり、他の要素から独立して archiving のみを行うようなシステムには、ちょっとお目にかからなかった。また、これらのソフトには、データの互換性、及び特定の OS[5]への依存性といった辺りに問題点があるように見受けられる。

MSS NCAR(National Center for Atmospheric Research) は、大規模なファイル群を archival に扱うために MSS(Mass Store Server) というシステムを開発している [4]。このシステムは “Bitfile” なるメタデータの形式を用い、Retension Period によってファイルを処分できるようになっている。このシステムのもとにある情報の総量は 1996 年 2 月の段階で既に 72TB にのぼり、更に毎月 1.5TB のペースで増えているという。

類似システムの有無 組織的な電子文書の archiving を自ら行い、公開まで行っている Web-site は、管見では一つもない。例えば、「このページはこれこれの archive system をくぐってきたものです」と表示してある Webpage を、著者は未だ見たことがない。

但し、そのようなプロジェクトは一応始まってはいるが、まだ情報を公開する段階に至っていないだけであるという可能性は否定できない。

メタデータに関するもの 「情報に関する情報」を記述するメタデータの形式については、近年 (特にこの数年) になって、各所で様々な提案がなされており、それらの想定する用途も多様である。提案の代表としては、Internet 上の資源を表現するために、W3C が提案している PICS[10]、書誌情報の記述に特化しており、SGML 式の記述形式を採用している Dublin Core[13] 、IETF によるワークショップが開かれている URC[3] などがある。

しかし、記録管理に特化したメタデータの提案は、著者の見た限りでは、これといったものが無かった。

---

[4] Web で見る限り、オーストラリアでは電子文書館に関する議論が活発なようだ。

[5] 大抵は Windows 関係。

第4章　仕様要求

本章では、前章及び前々章の議論を踏まえながら、構築されるべきシステムに要求される事項について論ずる。

4.1　自主性

まず、構築されるべきシステムは、これを使用する各々の組織が自ら文書を整理し、将来に向けて公開するためのものであるべきである[1]。故に、システムは、各々のサイトが独立していても動作するものでなければならない。

4.2　利用者のクラス分け

生成されるデータの公正さを保証するため、利用者を幾つかのクラスに分類し、それらの各々について、取ることのできる行動を規定する必要がある。

例えば、一般の文書作成者が簡単にretention scheduleに手を触れられるようになっていては、文書作成者の当座の利益にとって不都合な文書が全て破棄されてしまう恐れが生ずる。

4.3　データの記述形式

データ(単にこう言った場合、メタデータを含むものとする。以下同じ)の記述にあたっては、なるべく将来に亙って多くの人が読めるコード体系と、極力互換性の高いデータ形式を採用するべきである。「将来の人が読む」ということを考えた場合、より普遍的かつ安定な形式に賭けるべきなのは見やすい道理であると考える。

ここで「極力互換性の高いデータ形式」について、例を挙げて示す。

例えば、メタデータに記される時刻は、通常の完全な暦形式で表現される必要がある。年が下2桁のみというのは論外だが、それ以外にも、例えばUnix内部の時刻表現をそのまま記述してしまったりすると、Unix以外のOSを使用している人々をも「2038年問題」[2]に巻き込んでしまう。

---

[1] こうしたことを自ら行うことによって、組織体は自らのaccountabilityを証明し、ひいては信用を増すことになると著者は信ずるものである。

[2] Unixでは時刻を1970年1月1日正子(GMT)以来の秒数で表現しており、そして通常は、時刻を表現する整数には32ビットの符号つき整数が用いられているので、1970年1月1日正子+2^{31}秒=2038年1月19日午前3時14分8秒(GMT)にカウンターが一周してしまうことになる。

4.4 データの領域を区分する必要性

4.4.1 業務領域と記録領域の分離

文書作成者が業務に用いるディレクトリ以下 (以降「業務領域」と呼ぶ) には、文書作成者にとって今必要なファイルのみが置かれ、当分の間、緊急に必要でないファイルは、文書作成者とは関係のない場所に置かれるようにしたい。また、文書作成者が必要なファイルを紛失した場合や、故意に文書を消去する場合のことをも考えると、文書記録及びそのメタデータ (以下「記録情報」と呼ぶ) を蓄える場所 (以下「記録領域 (repository area)」と呼ぶ) は、文書作成者のディレクトリとは異なる箇所に置くべきである。

記録領域は、一般の文書作成者の利便の為、バックアップとして振る舞うことが望まれるので、故に、記録領域から文書 (ディレクトリ構造を含む) を元通りに取り出せるようになっていることが望ましい[3]。

4.4.2 記録領域と史料領域の分離

Retention schedule をくぐりぬけて歴史的価値が定まった情報 (以下「史料情報」と呼ぶ) は、文書館システムの道義として公開しなければならない性質のものである。

しかし、史料情報を記録領域と同じところに置いてしまうと、記録情報の安全性が脅かされるおそれがある。なんとなれば、記録情報には、まだ外部に対して秘密にしておかなければならない事項が含まれているかもしれないからである。

よって、史料情報は、記録情報と異なるところに置かれなければならない。以下、この領域を「史料領域 (archival area)」と呼ぶ。

4.5 メタデータへの要求

メタデータ自体の必要性は第 2 章で述べた。ここでは、メタデータにどのようなことが記されるべきかについて述べる。

収集先のディレクトリと record series 名との対応 これがないと、システムは文書の収集を行うことができない。

record series ごとの retention schedule 通常、異なるディレクトリにあるデータは、異なる目的の為に供せられると考えられる。よって、複数の record series、及び record series ごとに独立した retention schedule が記述できるのでなければならない。

record series ごとの履歴 即ち、「その record series がいつ誰によって作成され、閉鎖され、公開されたか」等々を記述する部分である。文書の由来に関係する部分を記しておくと、将来の研究者が喜ぶに違いない。

---

[3] そして記録領域に対応する媒体は、上の要求を満たすために頑丈なものである必要があるが、3.1 節でも述べたように、ハードウェアの問題には立ち入らないこととする。

ディレクトリ構造の記述　記録情報のメタデータのみから、収集元の文書のディレクトリ構造が再現できるべきである。文書のディレクトリ構造は、組織体の構造をも反映していると考えられるからである。

同一名のファイルの互いに異なる版の統一的な管理　前述した各領域の構成により、システムは連続的にファイルを業務領域から記録領域に取り込むことになる。故に、必然的に、同一のファイルについて複数の異版が生ずる。

同一名のファイルがどのような経緯で変化していったのかを記録し、あとからその変化の跡を辿れるようにしておくことは、ファイルを作成した当事者にとっても、また、後世その記録を読むものにとっても、有益であると考える。

これを要するに、システムは同一のファイルの複数の異版を処理できる必要がある。

時刻に関する情報　Retention schedule の遂行の為に、時刻に関する情報が必要になる。retention schedule で指定する時刻は、しばしば「文書の作成から 3 年後」のような指示のされ方をするからである。即ち、ファイルの作成や更新の時刻が、逐一記録されるのでなければならない。

記録情報と史料情報でのメタデータの相違　記録情報と史料情報とでは、メタデータの扱いに、多少の違いが想定できる。例えば、記録情報に於けるメタデータは、効率の観点から、データ本体と分離していてもそれほど問題はないが、史料情報に於いては、流通の観点から、両者は融合していた方が望ましい。

4.6　記録情報

前章でも論じたが、システムが収集する対象となる文書は、メタデータ同様、できるだけ特定のアプリケーションに依存しないものであることが望ましい。

また、記録領域にある間の文書は、「冷却期間」にあるといえる。作成当初は秘密にしなければならない情報も、時間の経過に伴って秘密にしておく必然性は薄れ、しまいには消滅するものと考えられるが、その時が来るまでの間は、外部からのアクセスは拒否されるべきであろう。

4.7　史料情報

史料情報は、record series について単一のファイルであることが望ましい。その方が、network での移動が確実となるからである。

4.1 節と矛盾することを言うようだが、史料情報を置く場所は、必ずしも自サーバである必要はない。ひとたび責任をもって完成させた史料情報には、流通の自由が保証されるべきだからである。従って、確実に史料情報を託せるところがあるのならば、初めからそこに転送しても差し支えない。

更に、史料情報の保全のため、どこかのサーバ (必ずしも文書の発生元と同じである必要はない) が、ネットワーク上の史料情報に目を光らせていることが望ましい。

4.7.1 索引情報

構造化されたメタデータからは、容易に索引が抽出できる。以下、これを「索引情報」と呼ぶことにする。

この目的の為に、メタデータには要約やログなどの、検索のヒントとなる情報を含められるようにすることが望ましい。そのようにすれば、ftp の archie 等で用いられている ls-lR ファイルよりも優位に立つことができる。

更に、この抽出された索引情報を互いにやりとりして、「電子文書館ネットワーク」を構築することが可能になる。

また、索引情報を専門に収集する、索引サーバーを置くことも考えられる。そして索引サーバーは情報の結節点になるであろうから、

公開鍵暗号系を認証の手段として使用する場合 (次節を見よ)、索引サーバーが公開鍵サーバーを兼ねるのも有効であろう。メタデータが公開鍵を配送するのに利用できると、なお良い。

4.7.2 真正性の保証

真正性 (authenticity) とは、「文書の出自がはっきりしていること」である。紙文書においては、真正性の保証のために印鑑や署名などを用いるが、電子情報は容易に複写できるという特徴があるので、ただ複雑なビット列を作ったのでは、全く意味がない。また、電子情報の改変は容易であることにも留意しなければならない。従って、何らかの手段を用いて、史料ファイルの真正性を保証する必要がある。

これについて、現在使用できる有力な技術は、公開鍵暗号であろう。

但し、暗号技術は、ハードウェアなどと同様、「古びる」ことが確実視されている。まず、暗号そのものの強度が変わらなくても、ハードウェアの速度が速くなれば、力任せに暗号を破ることは容易になる。1 種類の鍵を試すのに必要な時間を 1 ミリ秒と仮定した場合、56 ビットの鍵[4]を破るのに必要な時間は $2^{56}/1000$ 秒 $\approx 7 \times 10^{13}$秒$\approx 2 \times 10^{6}$年であり、安全性にさほどの問題は無いかのように見えるが、仮に前述の機械の 100 倍の計算速度を持つ計算機が 100 万台協力して同じ暗号の解読にかかったとすると、7 日程度で解読できてしまうことになる [12]。

また、現在使用されている暗号系の理論的な欠陥が将来発見されないとも限らないので、システムが使用する暗号系は、システム本体とは独立にしておかなければならない。

4.8 Retention Schedule

Retention schedule による文書量の統制は、電子的であるか否かを問わず、記録管理の要諦である。Retention schedule による統制を怠ったならば、ストレージが文書でパンクしてしまうばかりでなく、将来の利用者をして大量の玉石混交の文書の山に埋もれさせることになってしまい、「未来の人々に情報を提供する」という文書館の意義から考えても、はなはだ不親切である。

---

[4] これはアメリカ合衆国政府の定めた DES という暗号系の鍵の長さである。

極言すれば、Retention schedule を実行する機構が無ければ、それは「記録管理システム」とは言えない。

Retention schedule は、文書、或いは record series に付随する情報 (しかも重要な!) なので、当然これは record series につくメタデータの一部として記述できなければならない。

Retention schedule に従う動作は、明らかに自動的になされるべきである。

Retention schedule には、最低限、以下のことが記述できるべきである。

- 記録領域から文書を破棄する時期
  これは record series 内でも、ファイル単位で行えることが望ましい。
- データの公開の時期

換言すれば、最低限、「破棄」「公開」の 2 つは、自動化されなければならないということである。

ここで注意すべきは、「公開してから廃棄するかしないか」の選択がありうるので、公開と廃棄とは、独立とみなすべきことである。

時期の記述方法は、暦日によるもののほか、「record series の設定からどのくらい」のように、記録管理にまつわるさまざまなイベントからの期間として記述できるべきである。

更には、以下のようなものも指定できるとよい。

- record series の閉鎖の時期
  但し「閉鎖」とは、業務領域からの情報の収集を打ち切り、文書の破棄や公開を許すモードに移行することをいうものとする。閉鎖を行う理由は、記録領域のデータを誤って廃棄しないようにするためと、処理にけじめを付けるためである。
- record series の重要度
  システムはこれを見て、記録情報をどの媒体にどれだけ保存するかを判断するのである[5]。
- 史料情報の転送先
  これは、第 4.7 節で述べたことと関連している。

Retention schedule の作成は難事であるので、作成を支援する機能のあることが望ましい。

業務の変化などは当然文書体系にも反映するので、最初に作った retention schedule が永久に利用できるとは考えづらい。故に、retention schedule は 変更可能である必要がある。

4.9 仕様の公開

最後に、このシステムが扱うデータ、特にメタデータについては、その仕様が公開されていなければならない。さもないと、せっかく史料の保存のために作ったデータが将来読めなくなってしまうという、滑稽な事態も起こりかねない。

---

[5] 記録情報は おいそれとは network には載せ難いであろうから、こういう工夫をするのも有りかと考える。

第5章　システムの提案: Digital Archivist

この章では、前章までの議論に基づき、“Digitai Archivist”(以下DAと略)なる文書管理システムの仕様を提案する。

5.1　DAの概略

著者が提案する文書管理システムであるDAの特徴は、およそ以下の通りである。

業務領域–記録領域–史料領域」の3部構成

Retention Schedule を含む Metadata　Metadataの一部として記述されたRetention Scheduleを定期的に読みとり、それに基づいて各文書に適切な処置が為される。

公開による保存の促進　史料領域はネットワークから見えることを前提としている。但し、networkを介さない公開の形式（CD-ROM等）をとることも可能ではある。

単純なデータ形式　なるべく全てのデータがISO646(いわゆるASCII)の範囲内に収まることを目指した。

真正性の保証　DAでは、公開鍵暗号系を用いた電子署名を付することによって、史料情報の真正性を保証する。

5.2　シナリオ

DAにおける文書管理の流れは、およそ以下のようになる。

登録　DAの管理対象となるディレクトリ、及びそのrecord series名とretention scheduleを指定する。

諸般の事情により、DAではretention scheduleの作成支援をサポートしなかったので、これは手動で行わなければならない。

これによって、DAに新しいrecord seriesが登録され、記録領域に新しいrecord seriesのための場所が確保される。

収集　Record seriesが登録されてから閉鎖されるまでの間は、登録されたディレクトリ以下の内容は、定期的かつ自動的にに記録領域に取り込まれる。

ファイル或いはディレクトリ(文脈上紛れない場合、以下単に「ファイル」とのみ言う場合がある)の内容に変化があった場合、記録領域には新しいversionのファイルの内容が転写さ

れ、同時に、新しい内容に関するメタデータが追加される。しかし、変化が無かった場合、記録領域のデータは変化しない。

閉鎖 新たに収集するファイルがないと判断された地点で、record series は閉鎖 (close) される。

閉鎖は、retention schedule に期日を指定することによって、自動的に行うことができる。

これによって record series は収集対象のディレクトリから切り離され、記録領域のみの存在となる。

廃棄 閉鎖された record series 中の文書は、retention schedule で指定した期日が来次第、適宜廃棄される。

DA では、ファイル名に含まれるパターンを指定することにより、record series 中の一部のファイルのみを廃棄することができる。

公開 (やはり)retention schedule で指定した期日になると、記録領域に残っている分のデータ及びメタデータは、外部とやりとりし易い形に整形され、signature を加えられた上で史料領域に転写される。

但し、上に記したいずれの行動も、管理の都合上、手動で行える余地を残しておく。

5.3 利用者の種類とその権限

DA での利用者の分類は、以下の通りである。

文書作成者 普通に業務を行っている人たちで、DA の収集対象となるファイルの作成者。業務領域への読み書きと、記録領域及び史料領域の読み出しのみが許される。

記録管理者 DA の収集対象となるファイルの管理に当たる人。その仕事には、retention schedule の入力、記録領域の更新、史料領域へのデータの譲渡などがある。公正を期するために、組織外の人間を雇うのも一策かもしれない。

史料管理者 史料情報の出納を担当し、また、他所のサイトの史料管理者と史料情報や索引情報を交換する。史料情報の認証を行うのは、史料管理者とする。

閲覧者 受け手側の一般人。索引情報 (後述) を利用したり、史料情報を download する。即ち、史料領域からの読み出しのみが可能である。

また、各行動についての権限は、以下の通り。

record series の設定 記録管理者

retention schedule の変更 記録管理者

データの収集 文書作成者、記録管理者

記録情報の読み出し 文書作成者、記録管理者

record series の閉鎖 文書作成者、記録管理者 (、システム)

ファイルの廃棄 記録管理者 (、システム)

公開　記録管理者、史料管理者 (、システム)

認証　史料管理者

5.4　コード系について

1997 年の地点で最も普遍的な文字コードは、ISO646(いわゆる ASCII) であると考えられる。

そこで DA では、metadata を ISO646 による SGML で記述し[1]、史料領域に於いては、ファイル本体の内容にも適切な encoding(具体的には Base64) を強制的に行うことで ISO646 になることが保証されるようにした。これは、テキスト以外のデータを DA が扱えるようにするための措置である。

以下に、SGML 及び Base64 について、若干の解説を付する。

5.4.1　SGML

SGML(Standard Generalized Markup Language, ISO8879:1986) は、構造を持った文書を記述するために開発されたマークアップ言語である。DTD と呼ばれる文書型宣言によって、user が独自の文書形式を定義できるのが特徴であり、CALS や HTML など、世界各国で、各種電子文書の標準として広く用いられている。

- plain text の範囲内で
- 階層的に構造化された情報が表現できる

のが、SGML の利点である。

5.4.2　Base64

これは、マルチメディア情報を電子メールで交換するために定められた MIME Part One(RFC 2045)[5] という規格の一部で、3 バイトの binary data を 4 バイトの ISO646(ASCII) 文字に変換するものである。64 種類の ASCII 文字を基底 (Base) とするのが、命名の由来である。

DA では、文書自体のデータを外部と交換する際にこれを用いることにした。その理由は、

- あらゆる情報を ASCII の範囲内に収めることができる
- 構成文字に SGML の制御文字が含まれていないので、メタデータの記述に含んでも干渉を起こさない
- もともと電子メールのために作られた規格なので、特定の OS への依存性が低い (その点で、uuencode などに対する優位性がある)

の 3 点である。

また、「ネットワーク上の標準に依拠すべきである」という、第 3 章での主張にもかなっている。

---

[1] 本文の記述に SGML を要求しているのではないのに注意。

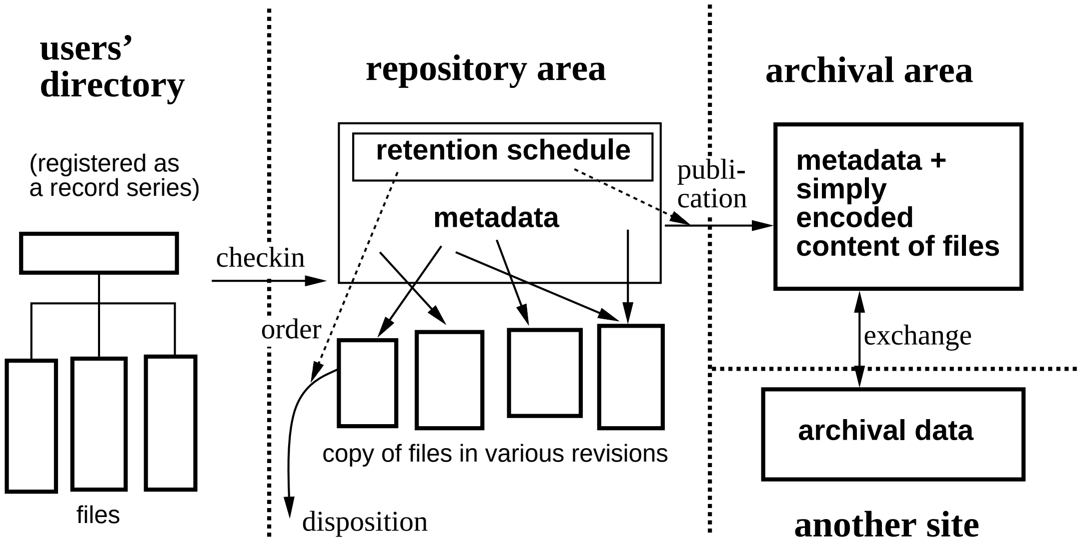

図 5.1: データ領域の構成

5.5 データ領域の構成

DA のデータ領域は、業務領域・記録領域・史料領域の 3 つに分けられる (図 5.1 を参照)。これらは、ほぼ「現用文書・半現用文書・非現用文書」、及びそれらを置く場所である「オフィス・記録センター・文書館」に対応しているが、現用文書もその写しが記録領域 (記録センター) に置かれる点が異なっている。

業務領域 実務上のファイルが置かれている場所をいう。ここの領域には、DA 固有のファイルは一切置かれない。ただ、「ここを業務領域と指定する」という記述が、記録領域に記されるのみである。

記録領域 DA が管理する文書は、記録管理者のみがアクセスできる一定の領域、即ち「記録領域」に集められる。

「記録領域」は、作成間もない文書から公開済みの文書までを、一元的に管理するものとする。

記録領域には、定期的に業務領域のファイルが取り込まれる[2]。

文書を取り込んだ場合のデータ及びメタデータの変動は、図 5.2 のようになる。

本システムでは文書の形式について特に制約を設けないこととしたが、文書自体の形式は、今後考慮すべき問題ではある。

記録領域からファイル及びディレクトリ

を取り出すことができる。

---

[2] 注意:「定期的な取り込み」は、retention schedule とは関係が無い。

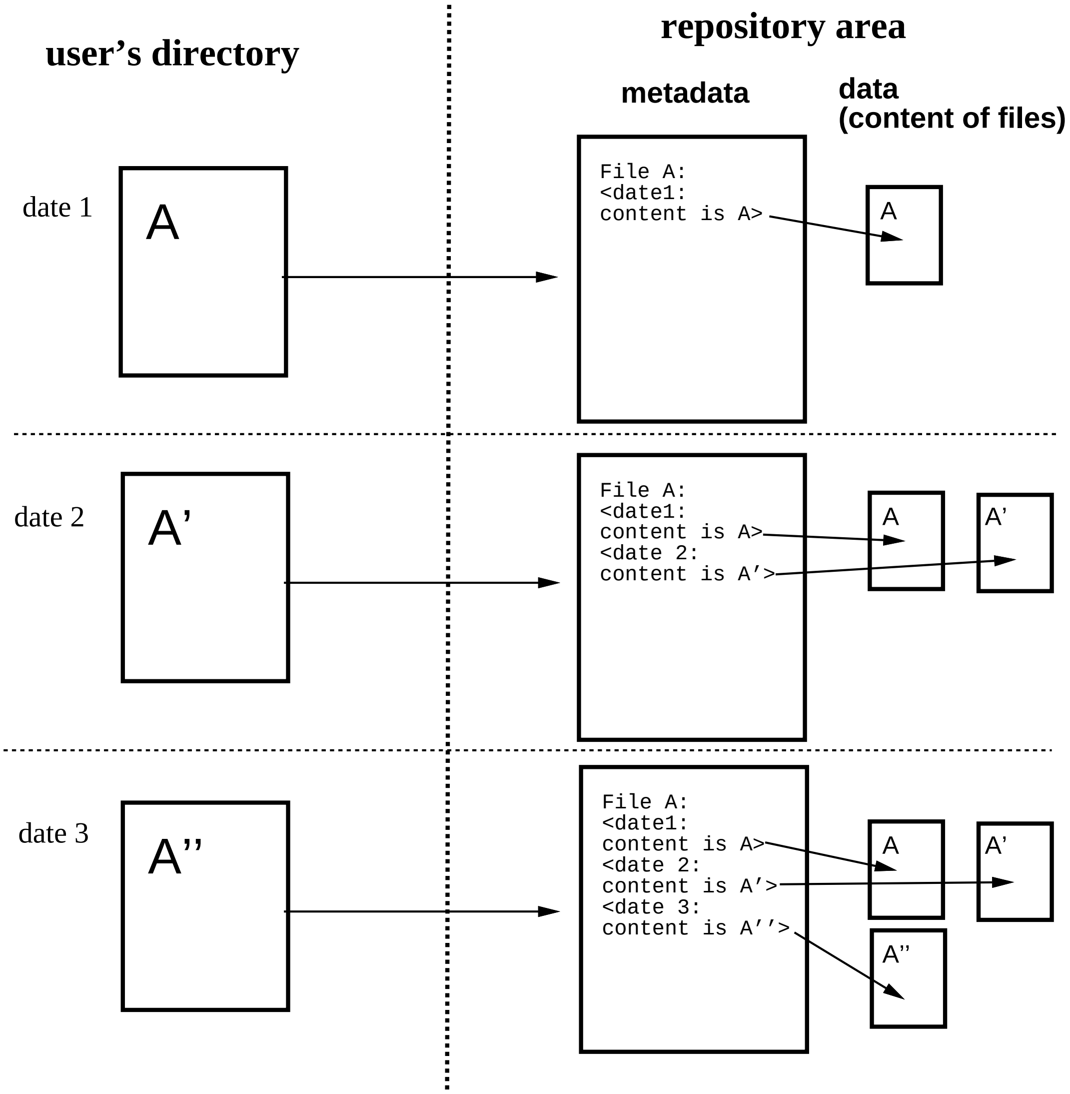

図 5.2: checkin によるデータ及びメタデータの変動

史料領域　Retention Schedule をくぐり抜けて史料となることが確定した記録情報は、ネットワークを通じて配布しやすい形に整えられた後に、史料領域に置かれる。

具体的には、史料領域を、ftp サーバーや http サーバーなどの、外部にファイルを転送するサーバーから見えるようにする。

史料領域に於いては、各々の record series の情報は、1 つの (或いは巨大な) ファイルの形を取る。各ファイルの内容については、前述の通り、有無を言わさず Base64 で encode し、メタデータの中に放り込むものとする。

通信はみな「1 つのファイル」の形で行い、その内容の展開は、各クライアントが行うものとする。

真正性の保証の為、史料情報には、signature を付することとする。signature は移動の度に更新され、受けとった側は、signature が正しいと判断した上で、文書を展開して読むことになる。

5.6　メタデータの構成

DA で使用するメタデータの構造は、以下の通りである。

5.6.1　記録領域のデータ構造

まず、記録領域に於けるメタデータには、大別して 以下の種類がある。

- record series と対応するディレクトリ名との対応表
- record series のメタデータ
- ファイル及びディレクトリのメタデータ

以上のメタデータは、効率の関係上、各々が別個のファイルになっているものとする。

対応表の中身は、登録時に記され、閉鎖時に除かれる。

その他のメタデータの内容は、およそ以下の通りである。

record series のメタデータ:

- 履歴
- retention schedule
- その他は「ディレクトリのメタデータ」に同じ。

履歴:

- record series の作成にかかるメタデータ
- record series の閉鎖にかかるメタデータ

いずれもタイムスタンプ[3]を含むほか、「record series の作成にかかるメタデータ」については、対応するディレクトリのパス名をも記す。

Retention schedule:
第 5.7 節で触れる。

ファイル或いはディレクトリのメタデータ:

- 各異版ごとの内容を記したメタデータ

各異版毎のファイル或いはディレクトリの内容を記したメタデータ:

- タイムスタンプ
- 作成者・更新時刻・permission など、ファイルの状態に関するデータ
- ディレクトリの場合は含まれるファイルのメタデータへのポインタ、ファイルの場合は 記録領域に copy されたファイルの本文へのポインタ

なお、大まかな見取図を、図 5.3 に示した。但し、この図に於いて、太線はファイルの境界を、細線はタグの境界を、太破線は領域の境界を、それぞれ表している。(他の図もこれに準ずる)。

5.6.2 史料領域のデータ構造

史料情報のファイルは、record series に関わる、(廃棄されていない) 全ての情報を含む。

その構成は、以下の通りである。

- signature
- 「record series のメタデータ」から取り出された履歴に関するメタデータ
- record series のメタデータ (当然、「履歴に関するメタデータ」を含まない。)
- 各ファイル/ディレクトリのメタデータ
- Base64 で encode された、各ファイルの各々の異版の本文

ちなみに「索引情報」は、上から、「Base64 で encode された、各ファイルの各々の異版の本文」を除いたものとなる。

履歴に関する部分には、以下の情報が更に付け加えられる。

- 史料情報を複製したというメタデータ
- 索引情報を作成したというメタデータ

これらのメタデータは、内部にタイムスタンプを含む。

なお、大まかな見取図を、図 5.4 に示した。

---

[3] 以下、「DA 或いは DA の使用者がが或る行為を行った時刻と実行責任者」を記したものを、「タイムスタンプ」と呼ぶことにする。

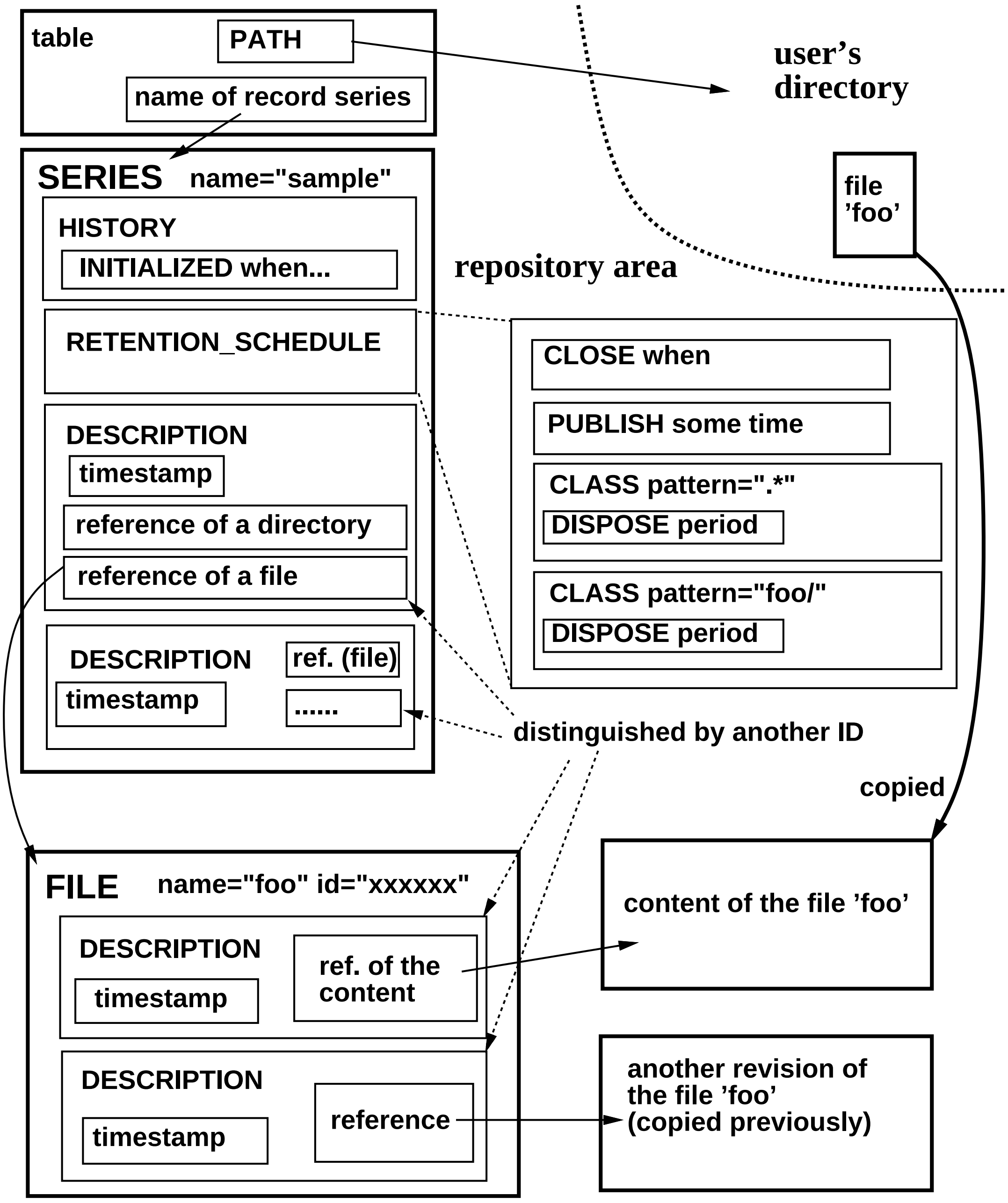

図 5.3: 記録領域のデータ構造

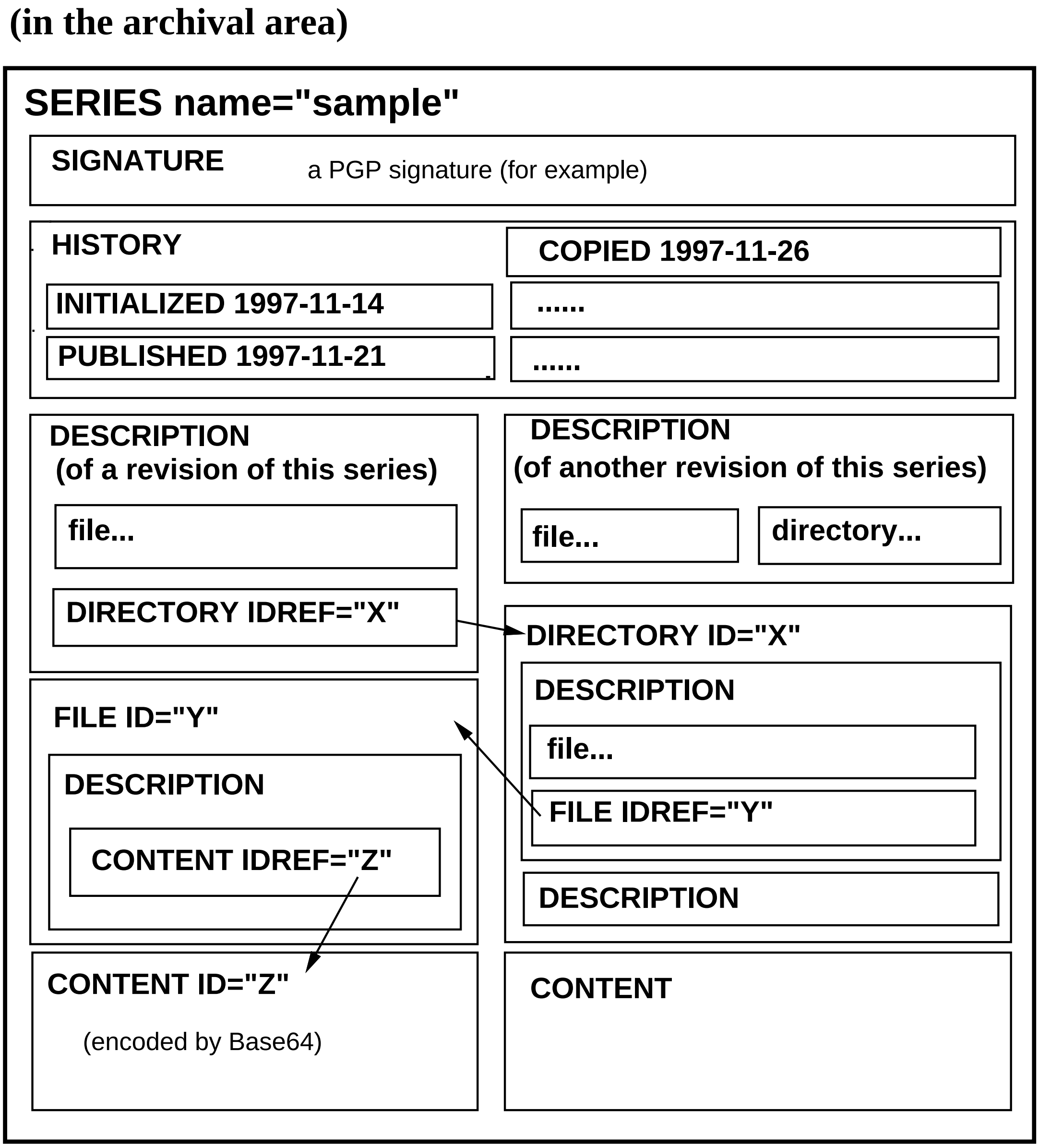

図 5.4: 史料領域のデータ構造

5.7 Retention Schedule

5.7.1 記述方法

Retention Schedule は、record series の設定時にこれを与える。記述は、記録領域の record series に関する metadata に転写される。なお、record series の設定は、記録管理者の仕事である。

以下の記述は、今回行った実装 (第 6 章を見よ) における記述方法の具体例を示したものである。

Retention schedule の種類は、以下の 3 種類とする。

`CLOSE` record series を閉鎖する時期を記す。

`DISPOSE` record series の各ファイルを廃棄する時期を記す。

`PUBLISH` record series を公開する時期を記す。

これらのうち、`CLOSE` と`PUBLISH` は`<RETENTION_SCHEDULE>`の真下に、`PUBLISH` は`<CLASS>`の中に記述する。

`CLASS` タグは、record series 内の細分類を正規表現で定義する。これにパス名がマッチするファイルの集合の各々を、以下「クラス」と呼ぶことにする。

`TEMPLATE` には、ファイルの種類など、細分類に共通する情報を記す。今回の実装では具体的に利用していないが、こういう情報を与えておけば、将来的に助かることが多いだろうと考えられる。しかしこれには、作成された文書の形式や内容を確認するものではないという難点がある。

処置をすべき時期の表現は、

```
<PUBLISH>c2y:all,revise</PUBLISH>
```

のように記す。ちなみにこれは、「ファイル或いディレクトリの作成の 2 年後に、当該ファイルの全ての異版を廃棄するかどうかを考える」という意味である。

左端のアルファベットの意味は、以下の通りである。

`c` ファイル作成時刻から

`m` 最終更新時刻から

`i` record series の作成時から

`C` record series の閉鎖時から

(数字) 絶対時刻。
上の 4 つに挙げた相対的な時刻ではなく、「1998 年 3 月 31 日」のような確定の日付を与えることもできる。

どの異版を廃棄の対象とするかについての情報の与え方は、以下の通り。

`all` 全ての異版 「を捨てる (以下同じ)」。

`backs` 最新の異版以外

`mods` 最初の異版以外 「各異版 s 共通」と共通の書式 (或る時刻以降)

`revise` DA は記録管理者に注意を促し、それを見た記録管理者が判断の上、手動で処置を決定する。Retention schedule の策定時に文書の価値を量り難い時のために用意した。

但し、廃棄が行われても、記録領域には「かつてかくかくの record series が存在した」という事実が残る。

5.7.2 実施方法

廃棄・公開などの文書の処置を行うプログラムは、定期的に以下の動作を行う。

「閉鎖」「公開」の場合:

1. 指定された record series の retention schedule を読み込む。(以下各々のファイル (或いはディレクトリ) について)
2. retention schedule に記された行動時期の記述を読み出す。
3. メタデータ内の適切なタイムスタンプと前項で読み出した行動時期の記述から、行動を起こすべき時刻を算出する。
4. 現在時刻が行動を起こすべき時刻を過ぎているならば行動を起こし、過ぎていなければ起こさない。

「廃棄」の場合:

1. 指定された record series の retention schedule を読み込む。(以下各々のファイル (或いはディレクトリ) について)
2. ファイル名から、適用すべき retention schedule のクラスを見つける。
3. クラスに記された廃棄時期の記述を読み出す。
4. ファイルの適切な異版のタイムスタンプと前項で読み出した廃棄時期の記述から、廃棄すべき時刻を算出する。
5. 現在時刻が廃棄すべき時刻を過ぎているならば指定されたファイルの異版を捨て、過ぎていなければ捨てない。

(図 5.5 を見よ)

但し、処置の中には手動で行わなければならないものも含まれているので、その場合は、自動処理プログラムは、然るべき管理者に すべき仕事を通知する。

最後に、retention schedule の設定ファイルの例を示す。

```
<SERIES_INIT>
  <SERIES> MIHON </SERIES>
  <PATH> /home/users/itogawa/sk1/DA/mihon </PATH>
  <RETENTION_SCHEDULE>
    <CLOSE> i6m </CLOSE>
    <PUBLISH> i2y </PUBLISH>
    <CLASS PATTERN=".+">
      <DISPOSE> i18m </DISPOSE>
```

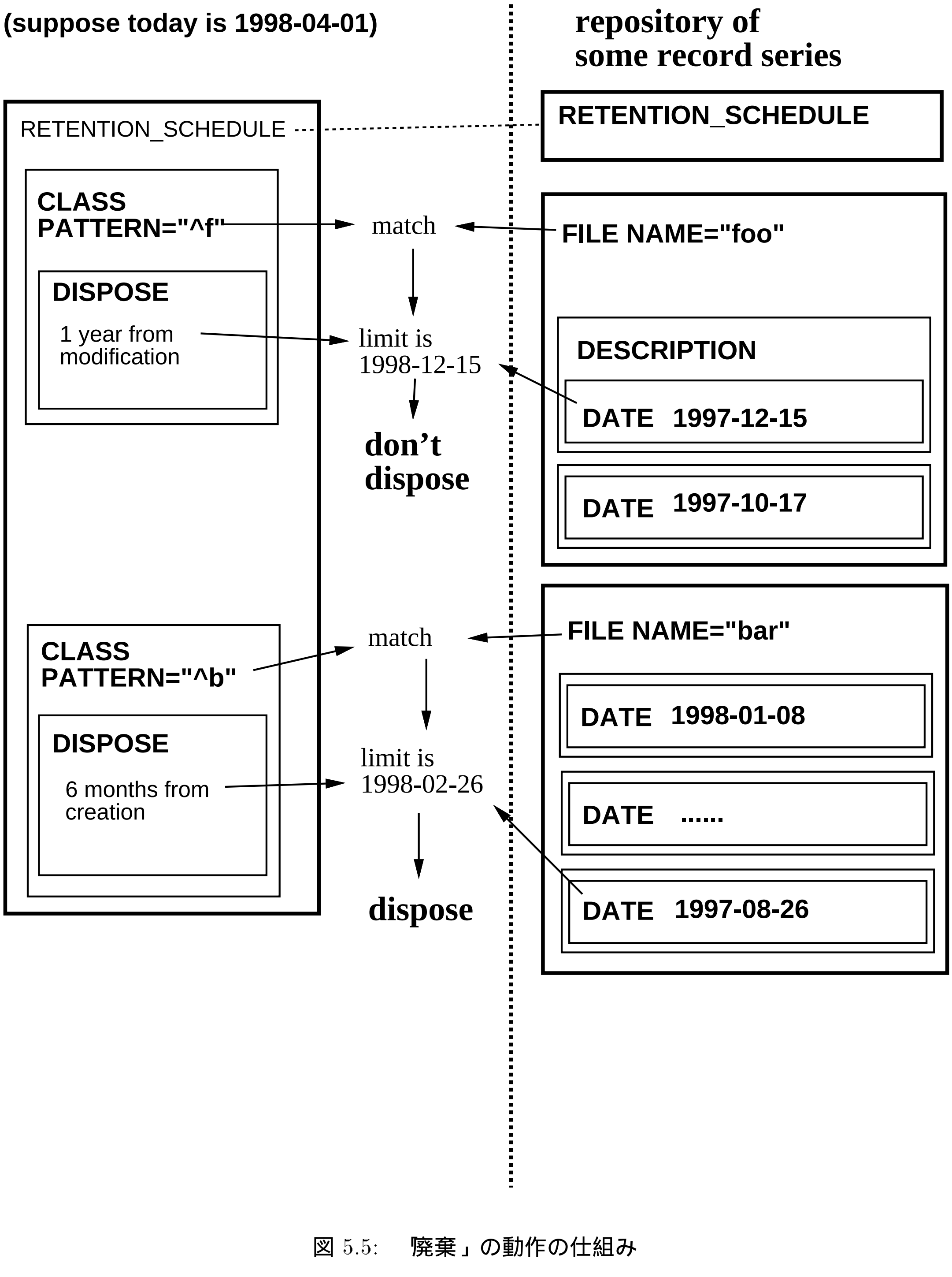

図 5.5: 「廃棄」の動作の仕組み

```
      <TEMPLATE>
        <CONTENT_TYPE>text/plain, euc-japan</CONTENT_TYPE>
      </TEMPLATE>
      <!-- TEMPLATE はここに「書くだけ」でよく、
下位の metadata に転写する必要はない。 -->
    </CLASS>
    <CLASS PATTERN="~$">
      <DISPOSE> c1d </DISPOSE>
    </CLASS>
  </RETENTION_SCHEDULE>
</SERIES_INIT>
```

第6章　実装

この章では、前章で著者が述べたDAの一部の機能の、試験的な実装について述べる。

6.1　概要

まず、今回行った実装の外延的な特徴は次の通りである。

- 使用言語：Perl5.003
- Linux及びSunOS4で動作確認
- shell command中心のinterface

コードは、`http://www.graco.c.u-tokyo.ac.jp/~itogawa/DA_prog`以下にある[1]ので、ご参照願いたい。

但しこの実装は、前章までの議論の成果を具体的に説明するために作られた、単なるプロトタイプに過ぎない。

また、前章の記述に比べて、以下の部分が未実装となっている。

- 記録管理者と史料管理者の区別
- 記録情報取り出し時のタイムスタンプ[2]の復元が完全ではない。

前章通りでない部分は、各コマンドの説明でもふれている場合があるので、そちらも参照のこと。

6.2　技術的注意

実装の簡略化のため、メタデータに使用したSGMLの記法については、

- 終了タグは必須
- 空タグ(中身のないタグ)は`<FOO/>`のように表す

のような、XML[2]式の記法を採用している。

また、`<FOO ATTR='value'>`の=の前後に空白を入れることができないなど、本来のSGMLが要求しないような構文的制約も存在している。

それから、公開鍵暗号の利用については、本実装では、史料情報にPGPを利用して署名を施すのみとした。但し、今回の実装では、史料情報はsignatureを1つしか持てないので、データの信

---

[1] 1997年11月21日現在。なお、「参考文献」のURLも同様である。

[2] ここでは、OSのタイムスタンプの意。

憑性は、直前の発行者を信用する場合に限られる。この辺りの実装の工夫も、今後のちょっとした課題である。

6.3 コマンド

各コマンドの詳細は、以下の通り。

series_config record series の諸設定、及び retention schedule の変更を行う。具体的には、record series 名と retention schedule、それと初期設定時には対応するパス名を記述したファイルを作り、それを記録領域に読み込ませる。

前述の通り、reention schedule の設定の自動化についてはいい案がないので、差し当たっては、記録管理者に煩雑な手続きをしてもらう風にしておいた。

checkin 業務領域から記録領域へ文書を取り込む。本実装では、record series の一部のみを取り込めるようにもした。

また、新しい異版ができたときに、異版のメタデータにログを加えるオプションも存在する。

checkout 記録領域から、指定した時刻のディレクトリツリーを再現する。

デフォルトでは、指定した時刻の record series の内容を指定したディレクトリの下に作る。オプションにより、メタデータを添えることができる。

また、時刻を省略した場合、現在の時刻に対応するものが作られる。

‘checkout’ の作るディレクトリでは、1 つのファイルは複数の異版を含み得るが、その場合、各々の異版は、“foo,1997-11-12/bar/baz” や “foo/bar,1997-11-11/baz,1997-11-15” のような、カンマで始まる拡張子で区別される (実際には時刻は秒単位まで書き出される)。

但し、前者は特定の日付の record series の内容を取り出した場合の後者は或る record series の全ての異版を取り出した場合の、パスの形式である (どちらにするかは option で選択できる)。更に言うと、後者の拡張子は、ファイル或いはディレクトリの更新時刻 (今回の実装では、実は取り込み時刻) を表している。

なお、上の記法は、extract でも共通である。

close 指定した record series の閉鎖を行う。

dispose 指定した record series の内容について、期限が切れたものの廃棄を行う。

publish 指定した record series の記録情報を史料情報の形式に変換し、signature を付けたものを史料領域に置く。

本実装での難点は、いちいち passphrase を打たなければならないことである (extract も同じ)。

従って、システムが自動的にすることは、「史料管理者に公開時期の到来を知らせる」ことだけとなってしまった。

もっとも、passphrase をオプションに書いてしまうこともできるのだが、これは危険に過ぎるであろう。

extract 史料情報からディレクトリツリーを抽出したり、史料情報を複写したりする。このコマンドを実行するたびに、史料情報の履歴部分 (第 5.6.2 節を参照) には「取り出された」事実が記録され、取り出された方のファイル或いはディレクトリツリーにも、それが「写し」である事実が記述される。履歴部分の更新に伴って、signature も新しいものに置き換わる。また、履歴部分にログを加えることも可能である。

retain checkin, close, dispose, publish は、手動でも実行できるが、基本的には、定期的かつ自動的に実行するために作られている。

それらのコマンドをを定期的かつ自動的に実行するのが 'retain' である。とはいえその実体は、上述のコマンドを並べた shell script を crontab で動かしているだけであるが[3]。

6.4 使用例

DA の動作見本を以下に示す。

まず、 `itogawa/sk1/DA/mihon` というディレクトリの下に、

```
Hello, World!
```

という 1 行が記された、`foo` というファイルがあるものとする。

ここで、以下のような設定ファイル (名前は`MIHON.config` とする) を作成し、

```
  <SERIES> MIHON </SERIES>
  <PATH> /home/users/itogawa/sk1/DA/mihon </PATH>
  <RETENTION_SCHEDULE>
    <CLOSE> i5M </CLOSE>
    <PUBLISH> C2M </PUBLISH>
    <CLASS PATTERN="bar">
      <DISPOSE> c3M:all </DISPOSE>
    </CLASS>
  </RETENTION_SCHEDULE>
</SERIES_INIT>
```

次に

```
> series_config MIHON.config
```

というコマンドを実行すると、記録領域 (その所在は `itogawa/sk1/DA/rc` であるとする) のもとに、

```
<!DOCTYPE ITOGAWA_METADATA>
<ITOGAWA_METADATA VERSION="0.4">
  <SERIES_PATH>
```

---

[3] 幸いにして Unix には cron[tab] があるので、この手の「定期的実行」の実装で思い煩うことはない。

```
    <SERIES>
      MIHON
    </SERIES>
    <PATH>
      /home/users/itogawa/sk1/DA/mihon
    </PATH>
  </SERIES_PATH>
</ITOGAWA_METADATA>
```

というファイル(itogawa/sk1/DA/rc/.serieslist)と

```
<!DOCTYPE ITOGAWA_METADATA>
<ITOGAWA_METADATA VERSION="0.4">
  <SERIES NAME="MIHON" ID="000000">
    <RETENTION_SCHEDULE ID="000001">
      <CLOSE>
        i5M
      </CLOSE>
      <PUBLISH>
        C2M
      </PUBLISH>
      <CLASS PATTERN="bar">
        <DISPOSE>
          c3M:all
        </DISPOSE>
      </CLASS>
      <OPERATOR>
        itogawa@globe
      </OPERATOR>
      <DATE>
        1997-11-21 11:52:46+09
      </DATE>
    </RETENTION_SCHEDULE>
    <HISTORY>
      <INITIALIZE>
        <OPERATOR>
          itogawa@globe
        </OPERATOR>
        <DATE>
          1997-11-21 11:52:46+09
        </DATE>
```

```
        <PATH>
          /home/users/itogawa/sk1/DA/mihon
        </PATH>
      </INITIALIZE>
    </HISTORY>
  </SERIES>
</ITOGAWA_METADATA>
```

というファイル(`itogawa/sk1/DA/rc/MIHON/000000`)が出来る。
次にここで、

```
> checkin MIHON
```

と入力すると、`itogawa/sk1/DA/mihon`以下の内容が取り込まれ、itogawa/sk1/DA/rc/MIHON/以下の内容は次のようになる。

```
::::::::::::::
000000
::::::::::::::
<!DOCTYPE ITOGAWA_METADATA>
<ITOGAWA_METADATA VERSION="0.4">
  <SERIES NAME="MIHON" ID="000000">
    <DESCRIPTION STATUS="latest" ID="000002">
      <AUTHOR>
        itogawa@
      </AUTHOR>
      <FILE NAME="foo" IDREF="000003"/>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
        1997-11-21 11:57:37+09
      </DATE>
    </DESCRIPTION>
    <RETENTION_SCHEDULE ID="000001">
      <CLOSE>
        i5M
      </CLOSE>
      <PUBLISH>
        C2M
      </PUBLISH>
      <CLASS PATTERN="bar">
```

```
        <DISPOSE>
          c3M:all
        </DISPOSE>
      </CLASS>
      <OPERATOR>
        itogawa@globe
      </OPERATOR>
      <DATE>
        1997-11-21 11:52:46+09
      </DATE>
    </RETENTION_SCHEDULE>
    <HISTORY>
      <INITIALIZE>
        <OPERATOR>
          itogawa@globe
        </OPERATOR>
        <DATE>
          1997-11-21 11:52:46+09
        </DATE>
        <PATH>
          /home/users/itogawa/sk1/DA/mihon
        </PATH>
      </INITIALIZE>
    </HISTORY>
  </SERIES>
</ITOGAWA_METADATA>
::::::::::::::
000003
::::::::::::::
<!DOCTYPE ITOGAWA_METADATA>
<ITOGAWA_METADATA VERSION="0.4">
  <FILE NAME="foo" ID="000003">
    <DESCRIPTION STATUS="latest" ID="000004">
      <AUTHOR>
        itogawa@
      </AUTHOR>
      <SIZE>
        14
      </SIZE>
```

```
      <CONTENT IDREF="000005"/>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
        1997-11-21 11:57:38+09
      </DATE>
    </DESCRIPTION>
  </FILE>
</ITOGAWA_METADATA>
::::::::::::::
000005
::::::::::::::
Hello, World!
```

ちなみに、御覧の通り、記録領域に於いて、メタデータのファイル同士は、番号で参照するようになっている。

いちいちメタデータを載せるのは繁雑なので途中の掲示は省略するが、次のような一連のコマンドを入力した後には(但し、各コマンドの実行には、十分な 間」を置いているものとする)、

```
> cat > ~itogawa/sk1/DA/mihon/bar
Howdy, world!
> checkin MIHON
> close MIHON
> dispose MIHON
> publish MIHON
```

史料領域に次のようなファイル(`MIHON,1997-11-21_12:23:56+09`)が生成されている[4]。

```
<!DOCTYPE ITOGAWA_METADATA>
<ITOGAWA_METADATA VERSION="0.4">
  <SIGNATURE VERBATIM ENCRYPTER="pgp">
-----BEGIN PGP MESSAGE-----
Version: 2.6.3i

iQCVAwUANHT+yZwiYmhe1r61AQG6jQQAyNGqi94Vhm177SAUVRadNpZIL4tHjQUL
0j5nIcwfbxqKWKg8FpvnXJfPoZr8kI41xoKPg2H6qAb2lnjA1fTHY1z1E77Zkmn0
fpD+wHJU1oIEddPDAxnlS+qjS8DeGjaaeeWr/GopUMbsqB9DhmL73RLAaMFS73Fy
gCHX7EAcIxI=
=BuIp
```

---

[4] 以下のファイルの履歴部分にPUBLISHが2つあるのは、1回バグによる失敗作を作ってしまったからである。申し訳ない。

```
-----END PGP MESSAGE-----
  </SIGNATURE>
  <HISTORY>
    <INITIALIZE>
      <OPERATOR>
        itogawa@globe
      </OPERATOR>
      <DATE>
        1997-11-21 11:52:46+09
      </DATE>
      <PATH>
        /home/users/itogawa/sk1/DA/mihon
      </PATH>
    </INITIALIZE>
    <CLOSE>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
        1997-11-21 12:08:35+09
      </DATE>
    </CLOSE>
    <PUBLISH>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
        1997-11-21 12:15:14+09
      </DATE>
      <NAME>
        :MIHON,1997-11-21_12:15:14+09
      </NAME>
    </PUBLISH>
    <PUBLISH>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
        1997-11-21 12:23:45+09
```

```
    </DATE>
    <NAME>
      :MIHON,1997-11-21_12:23:45+09
    </NAME>
  </PUBLISH>
</HISTORY>
<SERIES NAME="MIHON" ID="000000">
  <DESCRIPTION STATUS="latest" ID="000006">
    <AUTHOR>
      itogawa@
    </AUTHOR>
    <FILE NAME="bar" IDREF="000007"/>
    <FILE NAME="foo" IDREF="000003"/>
    <OPERATOR>
      itogawa@gene.c.u-tokyo.ac.jp
    </OPERATOR>
    <DATE>
      1997-11-21 12:05:51+09
    </DATE>
  </DESCRIPTION>
  <DESCRIPTION ID="000002">
    <AUTHOR>
      itogawa@
    </AUTHOR>
    <FILE NAME="foo" IDREF="000003"/>
    <OPERATOR>
      itogawa@gene.c.u-tokyo.ac.jp
    </OPERATOR>
    <DATE>
      1997-11-21 11:57:37+09
    </DATE>
  </DESCRIPTION>
  <RETENTION_SCHEDULE ID="000001">
    <CLOSE>
      i5M
    </CLOSE>
    <PUBLISH>
      C2M
    </PUBLISH>
```

```
      <CLASS PATTERN="bar">
        <DISPOSE>
          c3M:all
        </DISPOSE>
      </CLASS>
      <OPERATOR>
        itogawa@globe
      </OPERATOR>
      <DATE>
        1997-11-21 11:52:46+09
      </DATE>
    </RETENTION_SCHEDULE>
  </SERIES>
  <FILE NAME="foo" ID="000003">
    <DESCRIPTION STATUS="latest" ID="000004">
      <AUTHOR>
        itogawa@
      </AUTHOR>
      <SIZE>
        14
      </SIZE>
      <CONTENT IDREF="000005"/>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
        1997-11-21 11:57:38+09
      </DATE>
    </DESCRIPTION>
  </FILE>
  <CONTENT VERBATIM ID="000005">
SGVsbG8sIFdvcmxkIQo=
  </CONTENT>
  <FILE NAME="bar" ID="000007">
    <DESCRIPTION DISPOSED>
      <OPERATOR>
        itogawa@gene.c.u-tokyo.ac.jp
      </OPERATOR>
      <DATE>
```

```
        1997-11-21 12:22:12+09
      </DATE>
    </DESCRIPTION>
  </FILE>
</ITOGAWA_METADATA>
```

6.5 運用と評価

この種のシステムの評価は、少なくとも数カ月から数年間はかけて行うべきものであり、この研究に与えられた僅かな時間で早急に結論を下すべきではないのは承知している。しかし著者は敢えて、実装の性能評価のために、試験的にこの論文の原稿をDAにストアしてみた。

データの回収は、通常の使用ならば1日に1回程度で良いと思うが、この論文の原稿については、1日に3ないし4回ずつ、データの回収を行った。

そこで気づいた結果を述べると、今回の実装は、大量にディスクを消費する。原稿のディレクトリ自体は高々400キロバイト程度の大きさしかないのに、記録領域の大きさは、1週間で4メガバイトもの大きさに膨れ上がってしまった。

史料領域に於けるデータは、更に大きくなる。これの理由としては、ファイルの差分を取るなどの措置を講じていないというのもあるが、更に、ファイルの種類に関わらず、encodingにbase64を一律に適用しているため、ファイル本体に関する情報量が、確実に4/3倍になってしまうことも挙げられる。いくら何でも、数メガバイトにのぼるファイルを転送するのは大変なことであると考える。

やはりretention scheduleの記述に、「特定の名前を持つファイルは、システムの管理下にあっても収集しない」という項目が必要であったかも知れない。実は、その「特定の名前を持つファイル」が収集を逃れる隠れ蓑に使われることを恐れて、仕様には盛り込まなかったのであるが。

第7章　まとめ

7.1　結論

本論文では、紙文書と電子文書との相違、及び記録管理の諸理念から出発して考察を行い、電子文書を扱う上で注意すべき点として、以下の結論を明らかにした。即ち、

- 情報の保全のためにはメタデータが必要になること、
- ネットワークを活用すればより長い文書の寿命が望めること、
- (メタ) データの表現はできる限り平易であるべきこと、
- 電子文書では文書の状態に応じて細かく保管場所を変える必然性が薄く、一箇所で文書の集中管理が可能であること、
- 公開すべき文書には何らかの認証が必要なこと、

などである。

また、

- 以上の結論に基づいて「電子文書館システム」が持つべき諸々の性質を考察し、
- それらの諸性質を踏まえた上で “Digital Archivist” なるシステムの仕様を提案し、
- 更にその実装を行った

のだが、試みに実装を稼働させてみた結果は、

- 生成されるファイルが冗長である
- 必要な retention schedule の要素が足りないかもしれない

など、まだまだ検討すべき箇所が山ほどあることを示唆していた。

7.2　今後の展望

今回の研究では、主にデータ構造の議論に終始し、媒体の取り扱いの問題 (例えば、記録領域からデータが溢れたらどうするか、など)、及びネットワーク周りのあるべき仕様などについては、あまり深い分析ができなかった。

また、実装についても、すべきことは大いにある。

まず、このシステムが実用的になるためには、ファイルの圧縮/diff/patch 機能の追加、インターフェースの洗練、メタデータを解析する機構の充実など、多くの付随的な課題をクリアーしなければならない。

更にその上で、ギガバイト級のデータを扱っても、実用上充分な速度と耐久性で動く実装を作らなければばならない。具体的には、C などの「ちゃんとした」言語で実装を行う必要があるだろう。

その上、実装が終了した後も、少なくとも数カ月から数年の使用実績を積んで、不具合があるか否かが検討されるべきである。

以上の事項は、大学院に於ける著者の研究課題としたい[1]。電子文書館への道は、まだまだ先が長いようだ。

最後に、この拙論が、わが国の文書管理の発展に僅かでも寄与することを願いつつ、ここに筆を擱く。

---

[1] 但し、無事に卒業でき、大学院に進めたらの話ですが……　:-)

謝辞

この論文を著すにあたっては、実に様々な方々のお世話になった。

著者の指導教官である川合慧教授は、遅々として進まぬ研究を暖かく見守って下さり、また、適切な時に適切なアドバイスを下さった。

KTYY ゼミの皆様からは、様々な助言を頂いた。特に、山田康弘さん、金子知適さん、原田昌紀さんには、この実に至らない原稿の査読をして頂いた。

広域システム専攻の丹羽清助教授には、有益な参考書 [9] を紹介して頂いた。

全史料協をはじめ日本全国で逆境にもくじけずに頑張っていらっしゃる文書館員の皆様方には、著書を通じて大いに励まされた。

Perl の作者である Larry Wall 氏、MIME 規格の提唱者である N. Freed 氏及び N. Borenstein 氏、PGP の作者である Philip Zimmermann 氏などなど、ソフトウェア科学・工学における先覚者の皆様方の成果無しには、システムの実装は有り得なかった。

参考文献のページをご覧になればお分かりのように、この論文を執筆する上での参考資料の多くは、net 上の文献に依った。従って、持てる知識をを World Wide Web の上に置いて下さっている全世界の皆様方には、本当に沢山の情報を提供して頂いたことになる。

そして最後に、このような拙い論考につきあって下さった査読者の皆様には、まことに頭も上がらない。

以上の方々に、改めて感謝の意を表明するものである。

参考文献

[1] 安藤正人, 青山英幸（編）. 記録史料の管理と文書館. 北海道大学図書刊行会, 1997.

[2] Dan Connolly and Jon Bosak. Extensible markup language (xml). http://www.w3.org/XML/, 1997.

[3] Ron Jr. Daniel. Urc usage scenarios and requirements. http://www.acl.lanl.gov/URI/Scenarios/scenarios_3.asc, 1995.

[4] dig@ucar.edu. Mass storage system (mss). http://www.scd.ucar.edu/docs/mss/, 199?

[5] N. Freed and N. Borenstein. Multipurpose internet mail extensions (mime) part one: Format of internet message bodies. RFC 2045, 1996.

[6] info@digiduur.archief.nl. Digital longevity. http://www.archief.nl/digilong/, 1997.

[7] Brewster Kahle. The internet archive. http://www.archive.org/.

[8] library@archives.ca. Archivia enterprises archival and records management - electronic records. http://www.islandnet.com/ archivia/electronic.html, 1997.

[9] 作山宗久. 文書のライフサイクル. 法政大学出版局, 1995.

[10] PICS-ask@w3.org. Platform for internet content selection (pics). http://www.w3.org/PICS/, 1997.

[11] 高橋実. 文書館運動の周辺. 岩田書院, 1996.

[12] 辻井重男. 暗号 ポストモダンの情報セキュリティ. 講談社, 1996.

[13] Stuart Weibel and Eric Miller. Dublin core metadata. http://purl.org/metadata/dublin_core, 1997.

[14] 国立研究資料館史料館（編）. 史料の整理と管理. 岩波書店, 1988.