オンライン電子文書の長期保管のための枠組*

伊戸川 暁† 川合 慧‡ 山口 和紀§
東京大学大学院 総合文化研究科 広域科学専攻 広域システム科学系¶

1998 年 12 月 1 日

概 要

現在の電子情報の管理のあり方は、即時的な情報の共有については研究が比較的進んでおり、運用上の実績もあるが、長期的な保存の問題については様々な問題点を抱えたままである。

そこで本論文では、この問題を解決するための階層的アーキテクチャである POT Architecture を紹介し、更にその実装の可能性について考察する。

1 序論

現今の情報システムは、即時的・一時的な情報の交換・共有については大きな成果をあげている。しかし、Association of Records Managers and Administrators (ARMA) 元会長の David O. Stephens が [5] において指摘しているように、電子情報の長期保管については、あまり配慮がされてこなかった。

既に、我々が有している情報の多くは電子的媒体に記録されている。しかしこれらは、何の痕跡も残さず容易に消去され得るし、消去されなくても、媒体が稼働状態から外されると、記録された情報の再現が急激に難しくなるという特徴をもつ。

作家の私的な往復書簡が後年発見されて作家研究に大きく寄与したり、江戸時代の庶民のメモがその時代の経済を理解する上で役立っているといった事実を鑑みたとき、電子媒体に記録されている情報がその特徴により完全に消滅してしまうのだとすれば、人類の記憶にとってまことに大きな損失であると言えるだろう。電子情報の世界に於ける文書館の設立は、現代の急務ではないだろうか。

本論文では、電子情報化された時代の記録を将来に留めるために、オンラインの状態で電子的に蓄積された情報を数百〜数千年のスパンで保持するための枠組を提案する。

2 問題群

本章では、電子文書を長期保管する場合の問題について概観する。

2.1 媒体の寿命

現行の媒体の寿命は非常に限られている。最も普通に使われている媒体であるハードディスクは磁気的に記憶を行っているため、寿命はせいぜい数十年である。

CD-ROM や CD-R は、磁気的な記録をしていないのでより長く寿命が取れると期待されているが、媒体の出現から十数年しか経っていないこともあり、公称されている寿命についての確証が得られているわけではない。

---

*An Architecture for Archiving On-line Electronic Documents.

†ITOGAWA Akira
‡KAWAI Satoru
§YAMAGUCHI Kazunori
¶Department of System Sciences, Graduate School of Arts and Sciences, The University of Tokyo.

2.2 命名体系

従来の紙でできた文書 (例えば書物) の場合、書物の題名は公刊時の名前が (普通は) そのまま使われ、書名が廃れてしまうことはない。ところが電子文書の配布方法として最も使われている HTTP の場合、文書の参照はファイルの物理的所在を指示することで行っており、物理的所在が変化すると参照が無効になってしまうという問題がある。

これは、紙の世界で例えて言えば、書物を書名ではなく、図書館の書架の位置のみで指定しているようなものであり、極めて脆弱である。

2.3 文書ファイル解釈方法の遺失

しかし、上に示したような方法で文書自体の長期保存ができればそれで良いかといえば、決してそうではなく、文書そのもの以外の要因でも、容易に電子文書は読めなくなってしまう。

例えば、古いワードプロセッサで作成した文書は、それ自身が残っていても、ワードプロセッサのソフトウェアが存在しなくなってしまうと、中身を見ることができなくなる。従って、或るファイルを読むのに必要な情報へのアクセスが保証されなければならない。

しかし、ファイルの書式の仕様が分かっていれば、もともとのファイルを作成したソフトウェアが存在しなくなっても、仕様を基に新たに書式を取り扱う実装を作り、当該の書式でファイルを読み書きすることが (原理上は) 可能になる。プロセッサの交替サイクルが当分の間は現状のままだと仮定すると、ソフトウェアそのものではなく、ソフトウェアの仕様を保存する方にこそ利益があると考える。

2.4 価値の判断

この世に生成される全ての電子情報を永久に保存することは不可能である。これはコスト的に不可能であるだけでなく、有益な情報が凡庸な情報の山の中に埋もれさせることにもなる。電子情報の分量は特に膨大なので、有益な情報を掘り当てるにためには、大変な労力が必要となってしまう。

従って、登録の際には、文書の取捨選択の基準や保存期間などを記録できる機構が必要である。

2.5 セキュリティ

電子文書は変更が容易であるので、電子文書を故意の改竄などから守るための手段を講じなければならない。

また、完全に公開されていない文書に対しては、それを流出から保護する手段が必要である。

2.6 管理組織

文書が人間の寿命を越えて存続し続けるとなると、一つの管理者権限を、人間を入れ換えつつ数百年にわたって使用し続けることになる。従って、

- 作成された文書が将来にわたっても制御できること
- 文書の管理者に対しても、永続的な識別名が登録できること

が必要である。

2.7 メタデータの標準形式の欠如

永続的に電子情報を管理するためには、文書本体の他にも様々な補助情報が必要になってくる。このような補助情報を、元の文書のメタデータという。書誌管理用には Dublin Core[7]、Web 情報のアクセス制限のためには PICS[1] のようなメタデータの標準形式が既に議論されているが、電子文書の長期保存のためにどのようなメタデータが必要かについては、未だ本格的な議論は始まっていない。

3 電子文書長期保管アーキテクチャ: POT Architecture

前章で述べた様々な問題の解決方法を位置づけるために、本章では、電子文書長期保管アーキテクチャであるPOT Architecture[1]を導入する。POT Architectureでは、長期電子文書管理の機能を5層に整理している(図1)。

本章では、下位の層より順に、各々の機能を説明していく。各層の実現可能性については、第5章で述べる。

なお、以下では、一まとまりの計算機群を「サイト」と呼び、サイト同士はネットワークを経由して互いに接続しているものとする。

3.1 物理層

各サイトで、ファイルを永続的に存在させるための処理を行う層である。媒体の違いも、この層で吸収する。

この層では、記録されたファイル(文書を幾つかに分割した単位)を、サイト内のファイルの移動で永続化する。但し、滅失の可能性を完全に零にすることはできないので、制御層から与えられる重要度に従って管理し、滅失確率のコントロールを行う。

3.2 分散層

分散層は、複数のサイトが協力してファイルを永続化させるための管理を行う。例えば、複数の箇所にファイルの複製を作り、局地的な破壊によってファイルが失われる危険を減らす、などである。

このようにして、POT Architectureでは、2.1節でふれた問題を物理層と分散層の2層のみに局所化する立場をとる。

複製の数量や配置については、制御層から与えられた重要度及び可塑性[2]、サイトごとの記憶媒体の状況、サイト間の通信線の容量や安定性などから判断する。

この層より上位の層では、ファイルの物理的な所在とは独立な「ファイル識別子」によってファイルにアクセスする。これによって、2.2節の問題が解決されることになる。

3.3 文書層

文書層は、分散層ではファイルの集積として記録された情報を、文書の集まり及び文書間のリンクの集合として利用者に提示する層である。これによって、分散層ではファイルの集合にすぎなかったものが、互いにリンクで結ばれた文書の集合に見える。

文書層はリンク関係の維持を行うので、例えば、上位の層からリンク構造を崩壊させるような削除が指示された場合、文書層はその削除指示を拒否する。

文書は、大元となるファイル(以下「外枠ファイル」と呼ぶ)と、外枠以外の構成要素であるファイル0個以上より成る。外枠のファイルは外枠以外の要素へのリンクや(存在すれば)同一種類の文書の異なるバージョン(別文書として扱う)へのリンクを含み、文書層より上では、外枠ファイルの識別子が文書全体の識別子を代表する。

文書層では、文書本体と文書を読むために必要な情報、及び、文書の管理のために必要な情報へのリンク(これらもまた外枠ファイルに含まれている)を扱う。

文書を読むために必要な情報としては、ファイル形式に関する仕様書があげられる。このように、文書本体にファイル形式の仕様書を付随させることで、2.3節の問題を解決する。

また、文書の管理のために必要な情報としては、

- 文書の管理スケジュールなどを記した文書(制御層で利用。以下「管理情報文書」と呼ぶ)
- POTシステム[3]の管理に関係する人物を証明

[1] POTは Persistent On-line Text の略で、著者が将来実装しようとしている長期電子文書保管システムの仮称である。

[2] ファイルが更新される可能性をこう呼ぶことにする。可塑性の高いファイルは主に管理用の情報(後述)であることが予想される。

[3] POT Architectureを実装した任意のシステムをこう呼

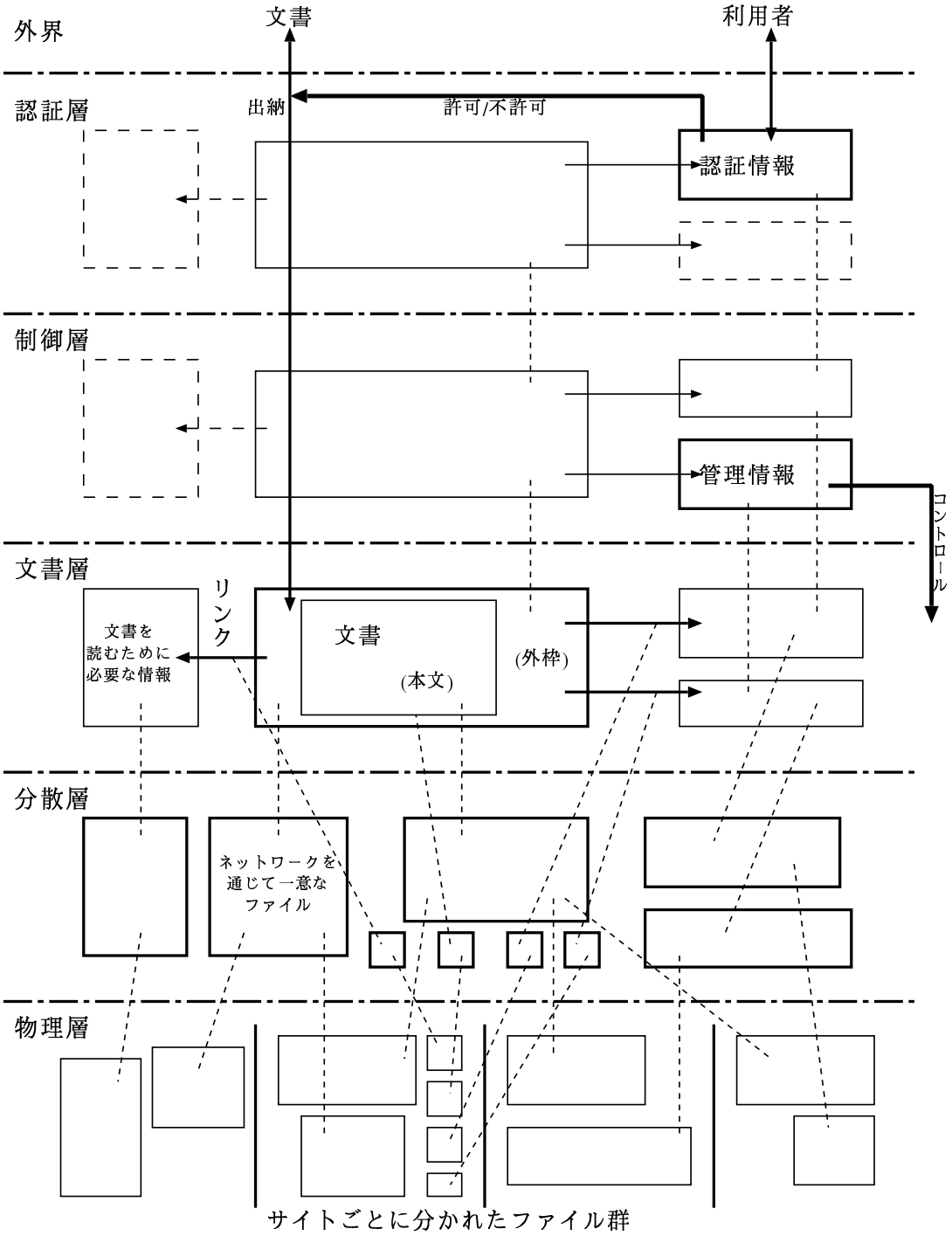

図 1: 層関係の概念図。同一の記憶内容が、各層でどのように見えるかを表している。図中の矢印のない破線は、要素の対応関係を示している。

する情報 (認証層で利用。以下「認証情報文書」と呼ぶ)

が考えられる。

なお、文書を読むために必要な情報や文書の管理のために必要な情報も、一般の文書と同様の扱いとする。

- 管理者の情報を文書の本体から切り離すこと
- 管理者の情報も一般の文書と同様に永続的な扱いをされること

の 2 点によって、2.6 節の問題が解決される。

更に、ここに挙げた種類の情報によって、2.7 節の問題も解決される。

3.4 制御層

制御層は、管理情報文書に基づいて、文書の保存年限・重要度・可塑性・公開範囲、及びそれらのスケジュールといった情報を司る層である。

但しここで、重要度とは、将来歴史的観点からその文書が重要になるであろうと期待される度合いを表現したものであり、2.4 節で述べた問題を解決するためのものである。

3.5 認証層

この層は、利用者を認証して適切な権限を与える層である[4]。システム内の文書のアクセス制御を権限に従って行うとともに、必要ならば特定の利用者にしか読めないような方法でファイルを暗号化して送付する。また、文書の真正さを保証するための電子署名の類も、この層で処理する。

この層によって、2.5 節の問題が解決される。

4 ケーススタディ

この章では、具体的な事例において各層がどのように働くかを示すことによって、POT Architecture の妥当性を示す。

4.1 閲覧

一般の利用者 P が POT システムのサイト S にアクセスして文書 R を読むときに起こることを、時間順に説明する。

- 認証層は、P の名乗る識別子を用いて P の認証情報を下位の層から取り出し、それを用いて P の身元を確認する。
- P が認証されたら、制御層は文書層に文書 R の管理情報を請求する。
- 文書層は文書の外枠ファイルの持つリンクから管理情報を取得して制御層に渡す。
- 制御層は、利用者の身元と管理スケジュールとを比較し、見せていけなければアクセス不可能であることを認証層に伝え、見せてよいことが分かれば、改めて文書層から本文の要素を取ってくる。
- 文書層は先ほど取得した外枠ファイルに記されたリンクの識別子を読み、残りのファイルを集め、文書を構成する。
- 認証層は受け取った文書にサイト S の管理者の署名を加え、必要ならば P に対して内容を暗号化して、P に渡す。

4.2 登録

A という人物が、ファイル形式 B を持つ文書 X をサイト S に登録するとき、以下のようなことが起こる。

- 認証層は前節に述べた方法で A を認証する。
- 制御層は受け取った文書 X のための管理情報文書 Y を作成する[5]。例えばそれは、『重要度 N、10 年後 E の範囲に対して限定公開、20

ぶことにする。

[4] なお、本アーキテクチャでは、一般の閲覧者も管理に関係する者も同一の方法でアクセスするものとする。

[5] むろん可能ならば、A が直接に与えてもよい

年後完全公開、廃棄年限なし』というようなものである。

- 文書層はファイル形式 B の仕様書が存在することを確認する (確認できなければ登録を拒絶する)。

  プレーンテキスト以外の形式の文書を登録しようとした場合、使用しようとするファイル形式の仕様書は、完全に公開されているか、将来的に公開される予定がなければならない。これは、ファイルの仕様書がなければ、電子化された文書の場合、利用が著しく困難になるからである。

- 次に文書層は、X 自体・A の認証情報・Y・B の仕様書など、必要な文書へのリンクを埋め込んだ外枠ファイル X' を作成し、X とともに分散層に渡す。

- 分散層は制御層から与えられた重要度に従い、適当なサイト (通常は自分を含む) の物理層に X と X' を送付する。

4.3 管理情報に基づいた行動

ここでは、保存年限が定められた文書 X を削除する場合について考える。

- 文書 X の管理者 A[6]は、X について削除の期限が来たら、POT システムに削除を命ずる。

- 文書層は、削除によって矛盾が生じないことを確認した上で、本文を含むファイルの削除を分散層以下に命じる。但し、文書が存在していたことを記録に残すため、外枠ファイルは消してはならない。

- 制御層は更新された管理情報ファイルに基づき、引き続き下位の層に対して適切なコントロールを行う。

[6]実装的にはエージェントが代行することになるだろうが、その場合も認証上は A の資格になる。

4.4 管理情報の更新

A が管理する文書 X の管理情報を置き換えるときに起こることは以下の通りである。

- 認証層は文書層に、X の外枠ファイルから管理情報へのリンクを取り出すよう命じる。

- 認証層はそのリンクに記された識別子から管理者の認証情報を取り出し、X の管理者が A に一致することを検証する。

- 検証に成功したら、文書層は古い管理情報文書を (当然外枠ではなく本体のみ) 削除し、新しい管理情報文書を加え、X の外枠ファイルの管理情報文書へのリンクを更新する。このとき、新旧双方の管理情報文書の外枠ファイルには、更新関係を示すリンクが張られる。

4.5 文書管理者の交替

旧管理者 A が文書 X の管理権限を新管理者 B に譲り渡すときに起こることは以下の通りである。

- 認証層は前節と同じ方法で A が X の管理者であることを認証する。

- 次に認証層は B の識別子を用いて、B がシステムに登録されている利用者であることを確認する。

- 認証層は B に変更されようとしていることを通知し、返事を待つ。

- B の許諾があった場合、文書層は、文書 X の持つ管理者の情報を新管理者の情報に置き換える。

5 実現可能性

本章では、第 3 章に示した POT Architecture が、どのようにしたら実現できるかについての可能性について検討する。

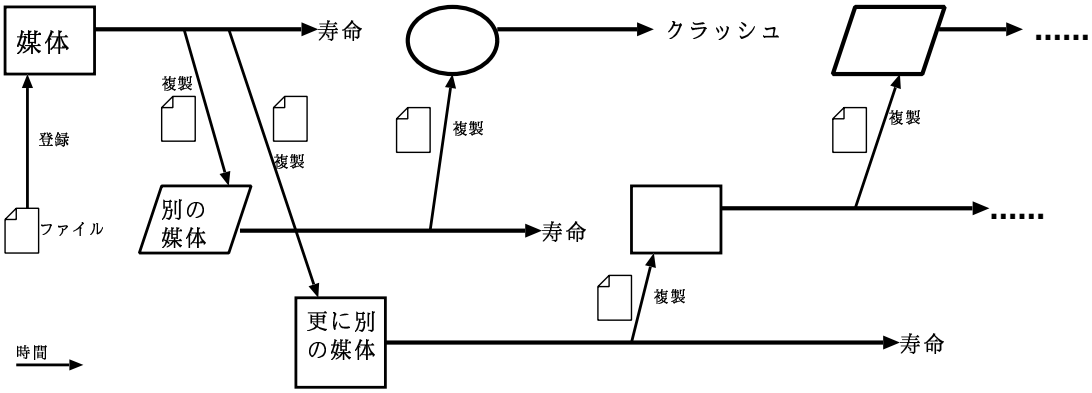

図 2: 媒体間の複写によるファイルの永続化

5.1 物理層

記憶媒体の物理的な寿命を超えて情報を永続させるには、適当な時間間隔で、古い媒体から新しい媒体に内容を写せばよい (図 2)。複写の際には、ファイルの識別子と媒体上の所在との関係を定義し直すことになる。このとき、複写先の媒体を適宜新しい種類のものにすることによって、媒体の技術的進歩にも対応が可能になる。また、複製や定期的なバックアップや RAID などの方法を用いてもよい。

単独の媒体の寿命を延ばす研究としては、HD-ROM[5] がある。これは、媒体に物理的に微細な傷をつけることで情報を記憶するというもので、磁気を使用しないため、耐用年数が非常に長く取れるという。

5.2 分散層

永続的な命名体系を提供する枠組として現在提唱されているものには、URN[4] がある。現在はまだ仕様の詳細について議論が進められている段階であるが、実用に供されるようになれば、永続的命名の標準になることが期待されている。

本稿執筆の地点で、すぐに利用できる URN 以外の手段としては、PURL[6] がある。これは HTTP を利用したサービスで、特定のサイトに属する URL に実際の URL をリンクさせて実際の URL の移転を吸収させるというものだが、将来的には URN ベースに移行することも可能だという。

5.3 文書層

ファイルの構造、メタデータ、及びファイル間のリンク関係を表現する手段としては、XML[2] 及び XLink[3] が適当であろう (登録される文書の本文が XML でなければならないという意味ではない)。その理由は、

- XML はプレーンテキストの形で表現でき、人間にも機械にも読み易い
- XML を用いると、自由に文書形式を定義することができる
- XLink は、リンク元やリンク先の文書から独立したリンク情報を持たせることができる

などである。

5.4 制御層

文書自体と管理情報は文書単位としては独立なので、複数の文書が 1 つの管理情報を共有することが可能である。従って実装上は、文書の集合ごとについてポリシーを設定し、そのポリシーに従って文書ごとのスケジュールを設定することにしてもよい。

5.5 認証層

システムにアクセスする者を認証する方法としては、幸いにして公開鍵暗号方式が既に発明されており、近年に至って十分な強度を持った暗号が一定の普及を見るに至ったので、実装上の困難はあまりない。

但し、具体的な暗号方式については、現在通用している方式が将来も安全であるとは限らない(むしろそうならない公算の方が高い)ので、特定の暗号方式のみしか使えないような実装をしてはならない。

6 まとめ

本論文では、電子文書を長期的に保管するための枠組であるPOT Architectureについて述べ、基本的な動作を見ることによってアーキテクチャの妥当性を検証した。更に、現存の技術によるPOT Architectureの実装にどれだけの現実性があるかについて考察し、要素技術はかなり揃っていることを確認した。

現在、本論文で示したアーキテクチャの実験システムを構築することを予定している。これによって、様々な事態に正しく対応できるアーキテクチャであるかを検証し、また、システムのコストと文書のコストについても、定量的な分析を行う予定である。

謝辞

著者に発表の機会を与えて下さった、S. Lagoonの中谷多哉子氏に感謝します。また、非常に多くの助言や指導を下さった、東京大学大学院総合文化研究科広域科学専攻KTYYゼミの皆様に感謝します。

参考文献

[1] W3C PICS Interest Group. Platform for internet content selection (pics). http://www.w3.org/PICS/.

[2] W3C XML Working Group. Extensible markup language (xml). http://www.w3.org/XML/.

[3] Eve Maler and Steve DeRose. Xml linking language (xlink). http://www.w3.org/PICS/.

[4] R. Moats. Urn syntax. Technical report, IETF, May 1997. RFC2141.

[5] David O. Stephens. 文書管理における電子化の世界的動向. ARMA International東京支部講演会, July 1998.

[6] The PURL Team. Persistent url home page. http://www.purl.org/.

[7] Stuart Weibel and Eric Miller. Dublin core. http://purl.oclc.org/metadata/dublin_core/.