

Javaを用いた動画配信用著作権保護カプセル

谷口 展郎, 阿部 剛仁, 塩野入 理

NTTサイバースペース研究所

ネットワークを介した情報流通を行ううえで、流通されるデジタル情報にまつわる知的所有権の問題を避けて通ることはできない。我々は、これを解決する手段として、内包する情報の保護機能を有する、自律情報カプセルMatryoshkaを提案している。本稿では、このMatryoshkaの動画への適用を目的とした、Javaによる実装について報告する。

Java-based Copyright Protection Capsule for Movie Delivery

Noburou TANIGUCHI, Takehito ABE, Osamu SHIONOIRI

NTT Cyber Space Laboratories

As the Internet grows into the global 'public road', distributing, delivering, trading information through the network is becoming common in our life. However, the issue about protecting intellectual property rights, such as copyright, is inevitable to face when we deal with the network information distribution. For this purpose, the authors have proposed an information capsule mechanism called *Matryoshka*. A *Matryoshka* is an executable multimedia file, and can protect the information inside by itself. In this paper, we present a brief explanation of *Matryoshka*, then introduce its implementation in Java, designed for movie delivery.

1. はじめに

ネットワークを介した情報流通は、次々と実際のサービスも始まり、徐々に我々の生活の一部となりつつある。しかし、ネットワークでデジタル情報の流通を行う際、避けては通れないのが、流通される情報にまつわる知的所有権の問題である。

この問題に関しては、流通される情報の種類（音楽、画像、文章、数値など）や知的所有権の形態（著作権、著作隣接権、あるいはプライバシー

権など）に依らず、統一的に扱える仕組みがあることが望ましい。我々は、それを実現する手段として、自律情報カプセルMatryoshkaを用いた情報流通フレームワークを提案している [1]。

自律情報カプセルMatryoshkaは、流通対象となる情報（以下、コンテンツ）と、その利用制約条件などコンテンツに関する情報（以下、コンテンツ関連情報）、実行プログラム（以下、コントロール）などを、一ファイル中にまとめたものである。コンテンツへのアクセスは、内包されているコントロールを介してのみ行うようになってお

り、コントロールは内包された利用制約条件に沿って動作するよう設計されているため、ネットワークを介して流通された情報は、配信先においても、与えられた利用制約条件の下で保護することができる。

既報 [1,2] で、我々は、OLE/COM (ActiveX) 技術を利用したMatryoshkaについて報告した。このActiveX版Matryoshkaでは、コンテンツとして、静止画、音楽、テキストを扱うことができる。本稿では、新たに開発した、動画をコンテンツとする、Javaを用いたMatryoshkaについて報告する。

2. 自律情報カプセルMatryoshka

2.1. 基本的な仕組み

自律情報カプセルMatryoshkaは、

- ・コンテンツ：流通対象となる情報
- ・コンテンツ関連情報：利用制約条件などコンテンツに関する情報
- ・コントロール：実行プログラム

を、単一の格納単位（通常はファイル）にまとめたものである。

コンテンツとは、ここでは「それ自身で価値を持つ情報」と定義する。コンテンツの例としては、音楽、映画、ニュース記事、株価などの情報が挙げられる。

コンテンツ関連情報とは、文字通り、あるコンテンツに付随する情報のことである。例えば、映画の監督や出演者、あるいは小説の作者や出版日など、コンテンツに関する書誌的な情報や、キーワードや要旨、あるいは視覚的／聴覚的特徴量など、コンテンツの内容に関する情報が挙げられる。また、利用期限や同時アクセス数制限などのコンテンツ利用制約条件もこれに含まれる。言い換えると、コンテンツ関連情報は、「対象コンテンツと組み合わせることによって価値を持つ情報」と定義することもできる。

コントロールとは、コンテンツやコンテンツ関連情報の表示や管理を行う実行プログラムである。コントロールには、コンテンツの表示や再生を行う表現コントロール（例えば静止画像なら画像ビューワー、音楽ならサウンドプレーヤー、動画ならムービープレーヤーなど）や、利用制約条件に基づいてコンテンツへのアクセス制御を行

うアクセス管理コントロールなどの種類がある。

これらを一つのファイル中に格納したものが「Matryoshkaカプセル」である。Matryoshkaカプセルは、見かけ上は実行ファイルとして認識される。このMatryoshkaカプセルを実行すると、ファイル内のコントロールが、やはりファイル内にあるコンテンツ関連情報を読み込み、それに基づいて、これもファイル内にあるコンテンツを読み込み、画面／スピーカーなどへの出力を行う。

コンテンツやコンテンツ管理情報は、通常、暗号化されており、読み込む際に、内包されているコントロールを介して復号する。つまり、カプセル内の情報へのアクセスは、内包されたコントロールからのみ行えるよう制限されている。さらに、カプセル内のコントロールは内包された利用制約条件に沿って動作するよう設計されている。こうして、ネットワークを介して流通されたカプセル内のコンテンツを、配信先においても、与えられた利用制約条件下で保護することができる。

Matryoshkaカプセル動作の基本的な仕組みを図1に示す。

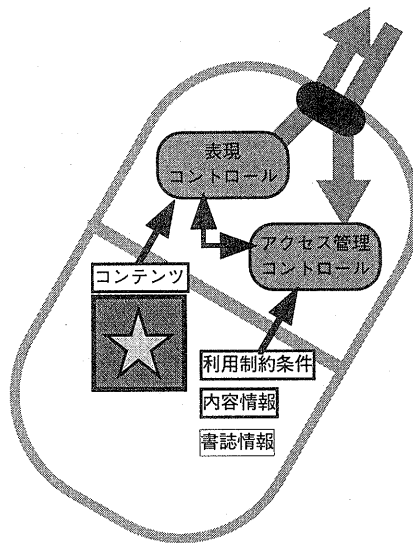


図1 Matryoshkaカプセルの仕組み

2.2. ActiveX版Matryoshkaの概要

Matryoshkaカプセルファイルは、複数の異なるデータから構成されたファイル＝複合ファイルとなる。このような複合ファイルを扱う場合、ある種のファイル内ファイルシステムが必要にな

る。

ActiveX版Matryoshka [1,2] は、OLE/COM技術を利用して実装されており、MatryoshkaカプセルファイルのデータフォーマットはOLE複合ファイル形式である。ファイル内ファイルの操作には、WindowsのOLE/COM APIを用いる。図2は、Matryoshkaカプセルファイルの構成の一例である。

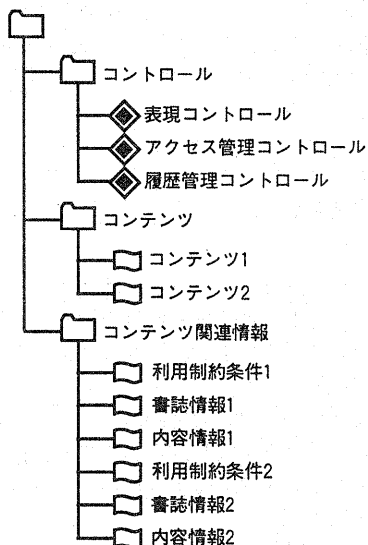


図2 ActiveX版Matryoshkaファイルの構成例

このMatryoshkaカプセルファイルをダブルクリックすると、関連づけられたMatryoshka参照プログラムが起動される。Matryoshka参照プログラムはMatryoshkaカプセルファイルをロードし、コンテンツ制御情報に従って、内包されたアクセス管理や表示/再生などのコントロールをロードし、これらに処理を委ねる。各コントロールはActiveXコントロールとして実装されており、Matryoshka参照プログラムはそのコンテナである。また、Matryoshka参照プログラム自身もActiveXコントロール化されているため、Matryoshkaカプセル内に他のMatryoshkaをコンテンツとして持つような入れ子構造も可能である。

現在のActiveX版Matryoshkaの実装では、コンテンツとして、静止画 (BMP, JPEG)、音楽 (MP3)、テキスト (プレーンテキスト) などのコンテンツを扱うプレイヤーが用意されている。また、利用制約条件として、「端末限定」「利用期限指定」「利用期間指定」「利用回数指

定」などが設定可能となっている。

3. Javaによる動画Matryoshka

従来の実装であるActiveX版Matryoshkaでは、利用プラットフォームがWindowsに限定されてしまう。そこで、プラットフォーム独立性を求めて、2.1節で説明したMatryoshkaフレームワークを、Javaを用いて新たに実装することにした。コンテンツについても、新たに動画を対象とすることとした。

3.1. 構成

Java-Matryoshkaは、Javaアプリケーションとして動作する。従って、Javaアプレットと異なり、実行にWebブラウザなどの外部アプリケーションを使用しない。また、Javaアプレットにおけるセキュリティ上の制約も受けない。

現在のMatryoshkaカプセルは、見かけ上、Javaのクラスファイルとして実装されている。

先に述べたように、Matryoshkaカプセルは、コンテンツ、コンテンツ関連情報、及びコントロールを、一つのファイルに収めたものでなくてはならない。従って、コントロール部が複数のファイルにまたがることは望ましくない。他方、Java言語においては「1クラス1ファイル」という規約が存在する。この2つの条件を同時に満たすために、Java-Matryoshkaは単一クラス内に収まるよう設計され、各コントロールはそのクラス内のメソッドとして実装されている。

動画のデータ形式としては、現在サポートしているのはMPEG-1のみである。MPEG-1の再生には、Java Media Framework (JMF) を利用する。

現在のところ、JMFのMPEG-1デコーダーは、Pure Java のものが存在せず、Performance Pack (JMF-PP) という、OS依存ライブラリーにしか含まれていないため、この動画Matryoshkaを利用できる環境は、JMF-PPが提供されているWindows及び Solaris に限定されている。

現プロトタイプで設定できる利用制約は、

- 1) パスワード認証
- 2) 利用期限指定
- 3) 利用期間指定
- 4) 利用回数指定

の4つである。

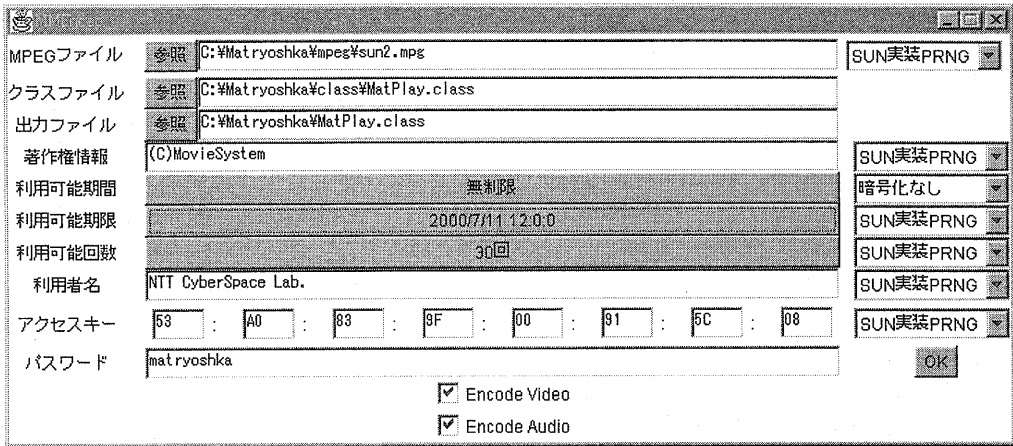


図3 カプセル化ツール画面例

パスワード認証では、Matryoshkaカプセル起動時にパスワードを要求する。このパスワードをコンテンツ及びコンテンツ関連情報の暗号化/復号化プロセスに適用され、コンテンツの利用認証を実現する。

利用期間指定は、最初の利用開始時から一定期間たったらコンテンツを利用不可能にするという制御を行う。

利用期限指定は、ある日時以降、コンテンツを利用不可能にするという制御を行う。

利用回数指定は、Matryoshkaカプセルの起動回数をカウントし、一定回数以上起動後、コンテンツを利用不可能にするという制御を行う。

これらの利用制約の一部は、利用履歴管理機能を用いて実現されている。利用履歴管理機能は、ユーザーの操作を記録し、必要に応じて参照する機能である。現プロトタイプでは、起動、終了、再生開始/停止の4つの操作履歴を記録する。これらの履歴は、利用期間制約や利用回数制約を実施する際に参照される。

3.2. 動作の流れ

●カプセルの作成

Matryoshkaカプセルを用いて情報流通を行うためには、まずコンテンツを、利用制約条件などのコンテンツ関連情報とともに、カプセル化する必要がある。我々は、このカプセル化を行うためのツールを作成した。

このツールはやはりJavaで実装されており、コンテンツとなる動画ファイルや、利用制約条件や

著作権情報などのコンテンツ関連情報、パスワード設定、各情報の暗号化の有無などを、GUIあるいはコマンドラインインターフェイスを用いて設定することができる。

図3はこのツールのGUI画面である。この図のフィールドに適切な値を設定したのち、[OK] ボタンを押すと、動画Matryoshkaカプセルファイルが生成される。

●カプセルの利用

作成されたカプセルは、ネットワークを通じて配信される。配信先でのカプセルの起動は、コマンドラインから、"java" の引数としてクラスファイルを指定する。Javaクラスファイルに対して適切な対応付けがされている場合は、ファイルをダブルクリックすることでカプセルを起動できる。

カプセルが起動されると、まず利用制約条件管

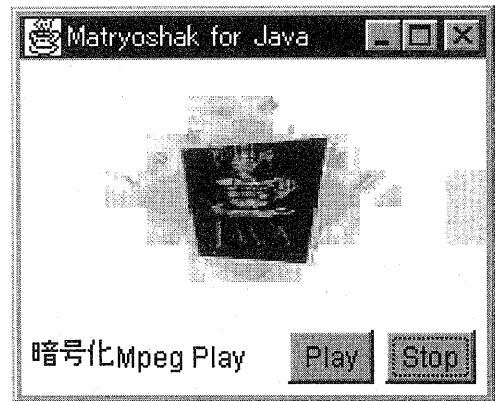


図4 カプセル実行画面例

理メソッドが呼びだされ、利用制約条件のチェックを行う。ここで、パスワード認証が要求されている場合は、ユーザーに対してパスワードの入力を要求する。また、利用期限/利用期間/利用回数の制限が設定されている場合は、これをチェックして、制限範囲を超えている場合はその旨メッセージを表示して終了する。

利用制約条件チェックをクリアすると、プレイヤーウィンドウが表示される(図4)。ここで、[Play] ボタンを押すと暗号化して格納された動画コンテンツを復号しながら再生する。[Stop] ボタンを押すと再生を停止する。ウィンドウのクローズボタンを押すと、カプセルの実行を終了する。こうしたボタン操作は、全て履歴として記録される。

3.3. 検証

作成したプロトタイプを用いて、実際に動画カプセルを作成/実行する簡単なテストをおこなった。テスト環境は、CPU: Pentium II 400MHz, メモリ: 128MB, OS: Windows 98, Java実行環境: Java2 Standard Edition 1.2, JMF 2.0 Windows Performance Pack である。

画面サイズ 352×240, フレームレート 30fps, ファイルサイズ 5111KB の MPEG-1 ファイルから、カプセル化ツールを利用して、さまざまな利用制約条件の下で、動画カプセルを作成した。カプセル化に要した時間は約4~5秒、カプセルファイルのサイズは5134KB であった。

次に、これらのカプセルを実行してみた。再生速度(フレームレート)は、体感的にはオリジナルの MPEG-1 ファイルを Windows Media Player を利用して再生したものと同程度であった。

最後に、このカプセルファイルを、Solaris マシン (CPU: UltraSPARC 167MHz, メモリ: 128MB, OS: Solaris 2.6, Java実行環境: Java2 Standard Edition 1.2, JMF 2.0 Solaris Performance Pack) 上にコピーし、実行したところ、問題なく動作した。再生速度(フレームレート)は、体感的には Windows 上とあまり変わらないように感じられた。

4. 考察

4.1. 現プロトタイプにおける問題点

今回のプロトタイプの作成を通じ、Javaを用いた動画カプセルについて、以下のような問題点が明らかになってきた。

●Javaバイトコードの脆弱性

Javaのソースをコンパイルしてできるクラスファイルのフォーマットであるバイトコードは、比較的逆コンパイルが容易なため、リバースエンジニアリングされやすい。このことは、コンテンツやコンテンツ関連情報を守るうえでも不利に働くため、これらの情報を守っているコントロール自身を保護したり、あるいはコントロールがリバースエンジニアリングされてもコンテンツの保護には影響が及ばないような仕組みを取り入れていく必要がある。

●ハードウェア情報が取得できない

ActiveX版 Matryoshka では、ハードウェア固有の情報を利用して、カプセルを実行できるマシンを限定する「端末限定」という利用制約を行うことができる。しかし、Javaはプラットフォーム依存性を排するため、プラットフォーム依存であることの多いハードウェア固有情報を取得する手段を標準では殆ど提供していない。このため、特定のマシン上でしか実行できないような利用制約を実現することが難しい。これについては、将来パーソナルICカード等のタンパーフリーのIDが普及し、こうしたIDへのJavaからのアクセスが標準的に実装されるようになることに期待している。

●クラスファイルの利用による制限

今回のプロトタイプは、Javaクラスファイルを用いて実現されているが、前述したように、Javaクラスファイルには、

- ・1クラス1ファイル
- ・ファイル名とクラス名が一致

という規定がある。この規約と、カプセル化=コンテンツ+コンテンツ関連情報+コントロールの一体化を両立させるためには、実装上の困難が大きくなり、ファイル名選択の柔軟性も失われる。この問題を解決するため、現在、Javaのもう一つの実行形式であるJAR (Java ARchive) 形式を用いた Matryoshka カプセルの開発を行っている。

●JMFによる制限

今回、MPEG-1の復号/再生にJMFを利用したが、これも前述したように、Performance Pack が特定プラットフォーム向けしか提供されていないため、プラットフォーム独立性という点で問題がある。これについては、現在、Pure Java の JMF で codec が提供されている他の動画フォーマット

へのMatryoshkaの対応を進めているが、将来的にはJMFのMPEG-1 codecがPure Java化されることを期待している。

4.2. その他

Javaによるコンテンツカプセルとしては、木俣らによる画像データカプセル [3] がある。主な違いは、木俣らのカプセルが、Javaアプレットとして実装され、静止画コンテンツを対象とするものであったのに対し、Matryoshkaは、Javaアプリケーションとして実装され、動画コンテンツを対象とするものであるという点にある。

Javaアプリケーションとして実装したことにより、Matryoshkaは、アプレットが受ける、

- ・ロードされたサーバー以外へのネットワーク接続の禁止
- ・ローカルリソースへのアクセス禁止

などの制約を受けないため、ネットワークが切断された状態でも動作できる。また、Webブラウザのような外部アプリケーションを必要としないので、実行環境の自由度が高い。

しかし、こうした特性は、逆に、悪意のあるカプセルが流通した際には、カプセルが、配信先のマシン環境に対するセキュリティ上の脅威となることを意味する。従って、Matryoshkaのような自律情報カプセルの流通を普及には、悪意あるカプセルからローカルリソースを守る仕組みも同時に提供する必要がある。我々は、このような仕組みを、[1]で述べている「サイトマネージャー」として実現することを目指している。

また、アプレットの場合、外部ドキュメントへの埋め込みはHTMLを用いると容易に実現できるが、Javaアプリケーションの場合、このような埋め込み手段は提供されていない。これについては、JavaBeansのようなコンポーネントインターフェイスの組み込み、Matryoshkaカプセルへの通信インターフェイスの追加などの拡張を行って対処することを考えている。

5. おわりに

コンテンツ、コンテンツ関連情報、及びコントロールを一つのファイルに収めた、コンテンツ保護のための自律情報カプセルMatryoshkaの、Javaによるプロトタイプを開発した。

Java-Matryoshkaは、Javaクラスファイルを利用しており、Javaアプリケーションとして動作し、コンテンツとして扱えるのはMPEG-1動画である。MPEG-1動画の再生にはJMF 2.0 PPを利用しており、同ライブラリーの提供されているWindows及びSolaris上で動作する。また、コンテンツ保護機能として、パスワード認証/利用期限指定/利用期間指定/利用回数指定の各機能を備えている。

さらに、検証のため、このプロトタイプを用いて実際に動画のカプセルの作成/実行を行い、コンテンツの利用制約条件を守りつつ動画の再生を行えることを確認した。

一方、このプロトタイプについて、Javaバイトコードの脆弱性、ハードウェア情報の取得、クラスファイル形式による制限、JMFによる制限などの問題点が明らかになった。これらの多くはJava固有であり、今後のJava標準の進化に期待するしかない点もあるが、その他については今後の取り組みを通じて解決していく予定である。

参考文献

- [1] 谷口, 森賀, 久松, 櫻井: "マルチメディア情報ベースとその格納単位Matryoshka", 情処 DICOMO '99 シンポジウム, 1999
- [2] 加賀美, 森賀, 塩野入, 櫻井: "コンテンツ流通における自律管理を目的としたカプセル化コンテンツMatryoshka", 情処 DPS 97-18, 2000
- [3] 木俣, 田中, 上原: "版權管理のためのJavaによる画像データカプセル化", 情処 DBS 111-11, 1997