

## 処理環境変化に耐性のあるAV同期再生・伝送方式

山口 孝雄, 荒川博, 鴨川郷, 栄藤 稔  
松下電器産業(株) 中央研究所

本論文では、端末のソフトウェア実装を前提として、処理環境の変化に耐性のある音声・動画の同期再生・伝送方式を提案する。処理環境とは負荷の状態、機種など端末や伝送路の処理能力を定める条件の集合である。音声・動画のソフトウェア処理では処理環境の変化により音声・動画の復号と再生に遅延が生じ再生品質が破綻する可能性がある。この問題に対して提案方式では、音声・動画の符号化ビットストリームに処理優先度を付加したフレーム単位の packets 構造を導入する。そして packets の廃棄を処理環境変化に応じて選択的に行うことにより再生品質の破綻を回避する。また、フレーム単位の優先度を用いて重要な動画や音声フレームを送信側で選択して伝送を行うことで、フレームの重要度を反映させて柔軟に伝送量を段階的に調節できることを示した。

### Synchronized audio and visual decoding and transmission scheme that is tolerant to variation of processing environment

Takao Yamaguchi, Hiroshi Arakawa, Akira Kamogawa, Minoru Etoh  
Matsushita Electric Industrial Co., LTD. Central Research Laboratories.

We propose a synchronized audio and visual (AV) decoding and transmission scheme that is tolerant to the variation of the processing environment, assuming that the future AV terminal is implemented by software. The processing environment means a set of conditions, such as workload and equipment, that defines AV processing ability. Due to the software implementation, the AV terminal might encounter fatal degradation in decoding quality when the environment varies worse. To avoid this, we introduce a prioritized frame packet structure to AV bitstream, where the frame packets are selectively discarded by comparing their processing priorities with the discarding threshold. We also show the possibility of adjusting traffic selectively with these priorities according to the importance of coded data.

## 1. まえがき

近年、演算処理装置の高速化により、音声・動画のメディア処理をソフトウェアにより実現しようとする研究開発が盛んに行われている<sup>1)</sup>。ソフトウェアによるメディア処理には、特定の符号化・伝送方式に限定することなく、様々な性能の端末で多様なメディアを適切な表現形式で扱える利点がある。たとえば、復号化ソフトウェアをダウンロードすることにより、放送受信方式の変更が可能な放送端末の研究開発が行われている<sup>2)</sup>。本論文ではソフトウェア実装による放送端末、通信端末機器を端末とよぶ。

音声・動画の連続メディアをソフトウェア処理の対象とすると、処理環境の変化に対してQoS (Quality of Service: サービス品質) 保証が難しいというソフトウェアメディア処理固有の問題が生じる<sup>3)</sup>。ここで処理環境とは負荷の状態、機種など端末や伝送路の処理能力を定める条件の集合である。メディアの復号・再生に遅延と不連続を生じさせない実時間制約がQoSとして重要である。しかし、処理環境が変化すると必ずしも実時間制約が満足されない。たとえば、性能の高い端末では、受信したすべての音声・動画ビットストリームの同期再生が可能であるが、性能の低い携帯端末や端末の負荷が増大した端末では、同一の音声・動画ビットストリームを遅延なく同期再生することが困難になる。

図1に動画1フレームあたりの復号処理時間(単位はmsec.)の変化を示す。動画ビットストリームはH. 263<sup>4)</sup>で、あらかじめ符号化したものを用意した(約30秒分の独自素材)。横軸上にフレーム内符号化されたフレーム番号をプロットした。フレーム内符号化された動画フレームは10フレーム毎に挿入した。図からわかるように、動画フレームによって処理時間が、かなり大きく変動する。たとえば、1フレームあたり160msec.以上かかる動画フレームを復号処理する能力がない端末の場合、復号処理が間に合わない動画フレームが存在する。図2に5フレーム毎の復号処理時間の局所平均の時間変化を示す。デコーダの入力側にバッファを設けることで、復号処理時間が平滑化される。しかし、図1

と同様、動画フレームによっては端末の性能や負荷によって処理が遅れて、決められた再生時間内に再生処理を完了させることが難しい。

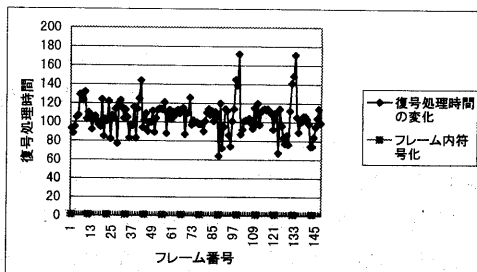


図1 復号処理時間の変化

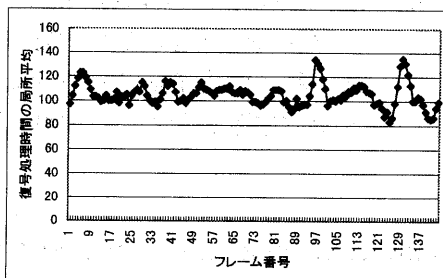


図2 復号処理時間の局所平均の変化

この課題に対して、ソフトウェア実装時の同期再生を保証するソフトウェアとしてIndeo Video<sup>5)</sup>がある<sup>5)</sup>。ベースとなる符号化はMPEG1/2で実現されているフレーム内符号化と、フレーム間符号化を組み合わせた方式である。フレーム内符号化された動画フレームを周期的(たとえば、10フレーム毎)に挿入する。動画の頭出しと、端末の過負荷時や記憶媒体からの読み出し速度が遅い場合に対して、フレーム間符号化の動画フレームを読み飛ばし、フレーム内符号化された動画フレームを再生時刻にあわせて選択、再生する。これにより端末の負荷に応じた時間同期を実現している。また、メディアの種類(テキスト、音声、動画など)に応じて優先度を付加して、過負荷状態に応じて処理するストリームを優先度に応じて選択できるファイル・フォーマットが提案されている<sup>6)</sup>。さ

<sup>1)</sup> Intel社の商標

らに、ネットワークへの伝送量を動的に調整する従来の研究には、符号化の方式を変更する方式や、タイムスロット毎に送信優先順位を決めて符号化データをパケット化し、伝送順序を制御する方式が提案されている。<sup>11)</sup>

上記の従来手法では、処理の単位は動画のフレームタイプによる処理量の調整あるいは、ビットストリーム単位のいずれかである。従って、音声フレームと動画フレームの重要度をそれぞれ反映させた選択的な処理量の調整は難しい。選択的な処理量の調整とは、たとえば、端末が無負荷時にはすべての動画を再生し、過負荷時には編集者があらかじめ定めた重要なシーンカット（たとえば、シーンチェンジで検出されるような代表画像）を選択して、再生可能にすることである<sup>2</sup>。

また、従来の研究ではシステムレベルでのQoS保証が検討されていた<sup>7), 8)</sup>。システムレベルとは伝送路における音声・動画ビットストリームの処理単位（パケット）を扱うレベルである。パケット長はネットワークやオペレーティングシステム（OS）に依存する。たとえば、イーサネットは可変長で最大1500バイト、ATMは固定長で53バイトである。

処理性能が異なる多様な端末での利用を考えた場合、利用できるネットワークやOSは同じであるとは限らない。従って、システムレベルでは一般的に処理単位が音声・動画のフレームと一致しないために、本論文で取り扱うQoS制御が煩雑になる。ネットワークやOSに依存せずに、音声・動画の処理単位のレベルで端末の負荷状態に応じて動的に処理量を調整できる同期再生・伝送方式が必要である。

本論文では、システムレベルのQoS保証ではなく、音声・動画の処理単位レベルでのQoS保証を議論する。そしてアプリケーションレベルで端末や伝送路の過負荷に耐性のある音声・動画同期再生・伝送方式を提案する。ここで音声・動画のストリーム単位、フレーム単位の処理レベルをシステムレベルの上位であることからアプリケーションレベルとよぶことにする。

<sup>2</sup> 音声の場合、たとえば有音部分を選択して再生可能にすることである。

提案手法では、端末の負荷変動に対して、アプリケーションレベルで音声・動画のデータの廃棄を選択的に行うことで、端末や伝送路の負荷変動に対して耐性を持たせる。次章で提案の音声・動画同期再生・伝送方式を述べる。最後に評価実験から本方式の有効性を示す。

## 2. 処理環境変化に耐性のあるAV同期再生・伝送方式

課題を解決するために、下記のアプローチを導入した。

### (1) アプリケーションレベルでの実現

アプリケーションレベル（フレーム単位、ストリーム単位の処理）で、音声・動画の同期再生を行うことで、異なるネットワークやOS上での実装が容易になる。

### (2) フレーム、ストリーム毎に付加した優先度

音声・動画のフレーム毎に優先度（フレーム優先度：FramePriority）を定義することで、端末が過負荷でも編集者があらかじめ定めた重要な動画のシーンカットの再生が可能になる。また、音声・動画のビットストリーム単位の処理の優先度を定義するストリーム優先度（StreamPriority）を定義した（図3）。

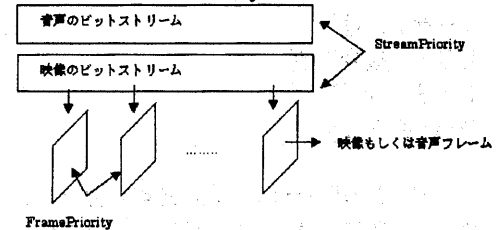


図3 StreamPriority と FramePriority の関係

図4に、本提案方式で扱うシステムデコーダモデルを示す。

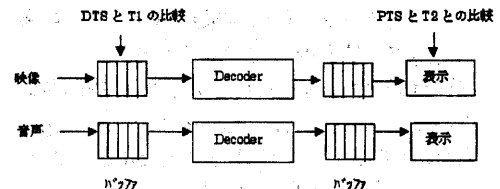


図4 システムデコーダモデル

音声・動画のデコーダはそれぞれ独立に設け

られ、その実装はマルチタスク OS の処理単位であるスレッドで構成されている。

音声・動画のビットストリームは、それぞれ独立に処理され、フレーム単位の符号化されたデータに分解、デコーダの入力側のバッファに格納される。フレーム毎にデコーダで復号されたデータは各デコーダの出力側のバッファに蓄積され、あらかじめ定められた表示時刻に合わせて表示される。符号化されたデータの復号および表示処理を管理するために、以下のような音声・動画のフレーム毎の管理情報をビットストリームとは別ファイルで用意した。

なお、伝送に関しては、フレーム優先度を用いて重要な動画や音声フレームを送信側で選択して送信、再送を行うことで、従来のフレームタイプに応じた選択的な伝送よりも、動画や音声のフレームの重要度を反映させて、柔軟に伝送量を段階的に調節できることを次章で示す。

#### ○動画フレームの管理情報

シーケンス番号

フレーム長 (バイト)

DTS (復号時刻)

PTS (表示時刻)

FramePriority

#### ○音声フレームの管理情報<sup>3)</sup>

シーケンス番号

DTS (復号時刻)

PTS (表示時刻)

FramePriority

データの復号および表示処理を管理するために、実装では復号する時刻を管理する処理タイマー (T1) と、音声・動画の表示時刻を管理する処理タイマー (T2) を独立に設けた。各タイマーは、それぞれ最初の符号化されたデータ、復号結果がバッファに到着した時点で値がゼロに初期化される。

時間情報は、MPEG1/2<sup>3)</sup>で採用されているPTS (Presentation Time Stamp: 表示時刻) と、DTS (Decoding Time Stamp: 復号時刻)を用いた。

<sup>3)</sup> 音声フレームは固定バイト (24KByte) である。

PTSは復号結果を表示する時刻を表し、実装では処理タイマー (T2) との比較からPTSよりも前に復号結果が得られた場合は、復号結果が得られるまで表示が待機させられる。PTSよりも後に復号結果が得られた場合、復号化されたデータの廃棄を行わずに表示を行う。

DTSは符号化されたデータがデコーダの入力側のバッファに格納されていない時刻を表している。しかし、いつの時点でも復号すべき符号化されたデータがデコーダの入力側のバッファに格納されている保証はない。

次に、過負荷時のシステムデコーダの動作について説明する。復号の可否を判定するために、デコーダ毎に、現在の処理対象となるフレーム優先度を決定する優先度カットオフ値 (CutOffPriority) を定義する。所定のカットオフ値より低いフレーム優先度を与えられているデータをフレーム単位で廃棄する。図5の例では、フレーム優先度は「0」が最も高く、数字が大きくなるにつれて優先度が低くなる。優先度カットオフ値が、「4」である場合にはフレーム優先度が「4」よりも大きなビデオフレームの復号は行わずに廃棄される。なお、復号の可否判定はデコーダとは独立に行なうことが可能であり、デコーダへ符号化されたデータを渡す前に予め廃棄する。

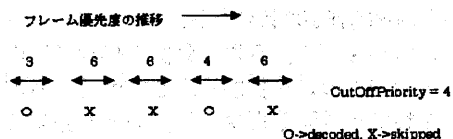


図5 デコード判定処理の例

図6に優先度カットオフ値の決定方法を示す。優先度カットオフ値はDTSと復号時刻を管理するタイマー (T1) との関係から決定される。T1の値は最初の符号化されたデータがデコーダの入力側のバッファに到着した時点でゼロに初期化され、優先度カットオフ値はフレーム優先度の最小値に初期化される (たとえば、フレーム優先度が「0」から「5」までの間の値で表現されるなら、初期値を「5」に初期化する)。T1の値と動画フレームや音声フレームに付加されたDTS

の値との比較から、 $T1$ の値がDTSの値よりも大きければ復号処理が遅れていると判断し、優先度カットオフ値を1減らす。逆に、 $T1$ の値がDTSの値よりも小さい場合、端末の処理能力に余裕があるものと判断し、優先度カットオフ値を1増やす。

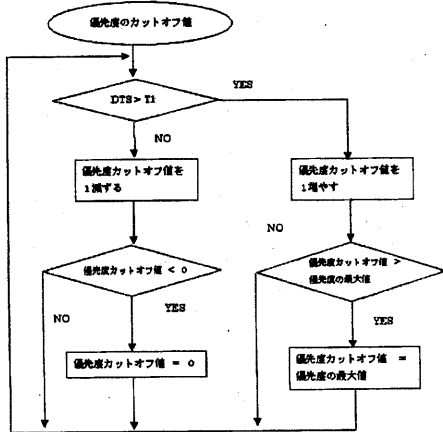


図6 優先度カットオフ値の決定方法

また、ストリーム優先度とフレーム優先度は、「0」が最も優先度が高く、数字が大きくなるにつれて優先度が低くなる。フレーム優先度が最も高い「0」のデータは必ず復号処理を行う（編集者があらかじめ定めた重要なシーンカットは優先度を「0」に割り当てればよい）。実装では、処理の優先度の高いストリームに対しては、ストリームの復号処理を行うスレッドに対して、OSレベルでより高いストリーム優先度を割り当てた。OSレベルでの優先度の割り当てにより、実行優先度の低いスレッド、すなわちストリーム優先度の低いビットストリームでは、その復号処理の実行速度の低さのため、より大きな遅延が発生し、その結果、より多くのフレームが廃棄される。

### 3. 評価実験

#### 3.1 実験環境

われわれは、プロトタイプを下記の環境で実装、評価した。

ハードウェア：DOS/V 互換機 (Pentium 166MHz, Memory 52MB), サウンドボード OS: Windows95<sup>4</sup>

<sup>4</sup> Microsoft 社の商標

なお、動画のビットストリームはH. 263で、音声のビットストリームはG. 711であらかじめ符号化したものを用意した（素材は30秒の独自素材である）。同時に音声・動画のフレームの管理情報を前述した形式で、音声・動画のビットストリームとは別のファイルとして作成した。評価実験での優先度の割り当ては下記のとおりである。

○動画：H.263, 1ストリーム, ストリーム優先度「1」フレーム優先度は4段階, フレーム内符号化されたフレーム (Iフレーム) は「0」～「2」の3段階, フレーム間符号化されたフレーム (Pフレーム) は1段階で「3」

○音声：G.711, 1ストリーム, ストリーム優先度「0」フレーム優先度は1段階すべて「0」

毎秒30フレームのCIFサイズの画像(352x288)から6フレーム毎に1フレームを取り出し、取り出された画像から10フレーム毎の画像をIフレームとした。6フレーム毎に画像をサンプリングした理由は、音声(G. 711)とCIFサイズの画像を同時に復号した場合、実験環境のコンピュータで5フレーム/秒が無負荷時に時間同期を維持できる上限であるからである。

また、端末の急激な負荷変動を考慮して、大幅なフレームスキップが可能のように、フレーム内符号化された動画フレームのフレーム優先度は下記のような繰り返しパターンで優先度を割り当てた（優先度が最も高い「0」のデータは必ず復号処理が行われる）。

0 → 1 → 2 → 1 → 0 → 1 → ……………

端末に仮想に負荷を与えるために、「cpugrab」<sup>5</sup>というCPUの処理能力を捕らえるソフトウェアを使用した。以降、負荷90%は9割のCPUの処理能力を捕らえた状態であることを示す。

#### 3.2 実験結果と考察

図7に動画フレームの優先度の時間的な遷移を示す。横軸はフレーム番号で、◆はフレームスキップが起こったかどうかを示す情報

<sup>5</sup> Microsoft 社から開発者向けに提供されている。

報で、値が「1」のときはフレームスキップの状態である。■は動画フレームに付加されたフレーム優先度を示し、▲は優先度カットオフ値を示す。負荷を無負荷、90%負荷の場合で評価した。無負荷時にはPフレームだけがスキップされているが、90%負荷時には最も優先度の高いIフレームと優先度「1」のIフレームのみが再生される。したがって、優先度「2」のIフレームとPフレームはスキップされている。

図からわかるように、本実験で用いた優先度カットオフ値は、端末の負荷変動の様子をあらわしている。

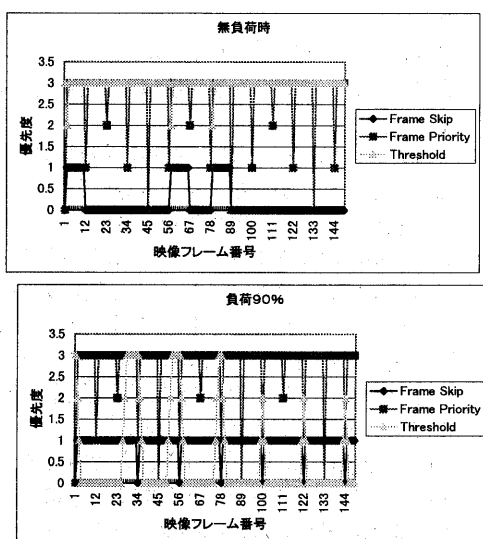


図7 フレーム優先度の遷移

図8に音声・動画フレームのPTSと再生時刻との関係を示す。◆は音声・動画フレームのPTS、■は動画フレームの再生時刻(復号が完了した時刻)、▲は音声フレームの再生時刻を示している。グラフには復号された動画フレームのみをプロットした。なお、動画フレームの再生時刻がPTSよりも値が下回る場合があるのは、動画のフレーム廃棄により、処理に余裕が発生し、表示時刻以前に復号が完了したためである。

また、音声はストリーム優先度により動画よりもシステムレベルで処理の優先度が上げられているため、音声フレームの廃棄は発生しなかった。なお、音声フレームの再生時刻とPTSはほぼ同一直線にある。

さらに、図からわかるように、音声・動画

の同期のずれに関しては、無負荷、負荷90%時とも、同期のずれは生じていない。

以上より、負荷が増大しても音声・動画の同期再生が維持でき、処理環境の変化に対して実時間制約を考慮したQoS保証が実現できた。なお、音声の場合も有音区間は高く、無音区間は低い優先度を割り当てるような段階的な優先度割り当てが可能である。

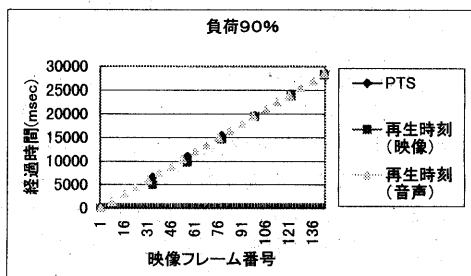
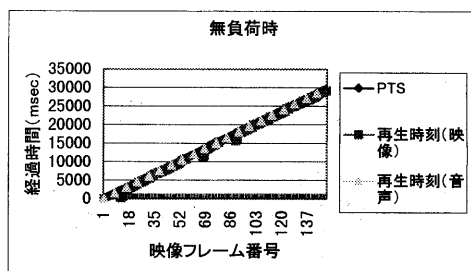


図8 PTSと再生時刻の関係

図9にフレーム優先度に応じた動画の伝送量の変化を示す。前述同様、フレーム内符号化された動画フレームのフレーム優先度は下記のような繰り返しパターンで優先度を割り当てた。

0 → 1 → 2 → 1 → 0 → 1 → .....

伝送パターンを縦軸に、平均伝送量(単位はKByte)を横軸に示す。伝送パターンは番号順に、1)すべての優先度の動画フレームを送送した場合、2)優先度「1」から次の優先度「1」の間の区間の動画フレームがスキップした場合、3)優先度「0」から次の優先度「0」の間の区間の動画フレームがスキップした場合を表している。

このように、フレーム優先度を用いて重要な動画フレームを送信側で選択して送信、再送を行うことで、動画や音声のフレームの重要度を反映させて、柔軟に伝送量を段階的に調節することが可能である。音声に関して

も同様な方法で伝送量を調整することが可能である。

特に、固定の伝送レートである公衆網を介した伝送の場合、再送回数が増加すると輻輳の原因となる。本論文で提案したフレーム優先度を用いて重要度に応じて動画や音声フレームを送信側で選択して再送を行うことで、柔軟に伝送量を段階的に調節できるものと思われる。具体的な評価に関しては今後の課題としたい。

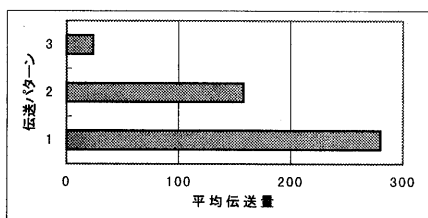


図9 フレーム優先度と伝送量の関係

#### 4. むすび

われわれはアプリケーションレベルでの処理環境変化に耐性のある音声・動画同期再生方式を提案した。提案方式では、音声・動画のビットストリーム毎にストリーム優先度を、ビットストリームを構成するフレーム毎に、フレーム優先度、復号時刻、表示時刻を対応づけて同期再生処理を行った。また、廃棄対象となるフレーム優先度を示すカットオフ値を定義し、この値により低いフレーム優先度を与えられているフレームを廃棄対象として選択的に廃棄した。同時に、フレーム優先度のカットオフ値を、どの復号時刻においても遅延が検出されない最も低いフレーム優先度に維持した。以上の結果、負荷変動に対して遅延が起きない必要最小限の packets 廃棄（最大限の再生品質）をアプリケーションレベルで確保することが可能になった。また、フレーム優先度を用いて重要な動画や音声フレームを送信側で選択して伝送を行うことで、フレームの重要度を反映させて柔軟に伝送量を段階的に調節できることを示した。

本提案手法を用いることにより、負荷に耐性のあるソフトウェア実装の放送端末、通信端末機器の実現が期待できる。なお、本論文で提案したフレーム優先度は MPEG4 システムで "degradationPriority" として採用され

た<sup>9),10)</sup>。

複数のビデオや複数のオーディオストリームでの定量的な評価が今後の課題である。なお、実装方法に関してはデコーダを追加することで可能である。また、ソフトウェアのダウンロードを含めて、Java<sup>6)</sup>のようなプラットフォームに依存しないフレームワーク上での実装および検証が課題である。

本研究は通信・放送機構 (TAO) からの受託研究「次世代放送方式技術に関する研究開発」の一環として行われたものである。

#### 参考文献

- 1) T. Turletti, C. Huiterna: "Videoconferencing in the Internet", IEEE/ACM Transactions on Networking Journal, pp.340-351, June 1996.
- 2) K. Rath: "Set-top box control software: a key component in digital video", Philips Journal Res(1996).
- 3) H. Arakawa, T. Maeda and M. Etoh: "Software architecture for flexible and extensible image decoding", Signal Processing: Image Communication, 1997, 10, pp.235-248.
- 4) ITU-T Recommendation H.263, "Video Coding for Low Bitrate Communication"(May 1996).
- 5) Intel Indeo Video 5:  
<http://developer.intel.com/ial/indeo/video/compress.htm>
- 6) ASF(Advanced Streaming Format):  
<http://www.microsoft.com/asf/good.htm>
- 7) ISO/IEC 13818-1: "INFORMATION TECHNOLOGY-GENERIC CODING OF MOVING PICTURES AND ASSOCIATED AUDIO: SYSTEMS", International Standard, November, 1994.
- 8) 藤原 洋監修: "最新MPEG教科書", アスキー出版(1995).
- 9) 柴藤 稔: "MPEG-4 システム", 動画情報メディア学会誌, Vol.51, No.12, pp.1996-1973(1997).
- 10) ISO/IEC JTC1/SC29/WG11/N1901, Text for CD 14496-1 Systems, November, 1997.
- 11) <http://www.hil.ntt.co.jp/SoftwareVision/us/htrn/svcom02.html>

<sup>6)</sup> Java は Sun Microsystems 社の商標