

フォーマット非依存性を考慮した映像伝送支援技術の研究

慶應義塾大学大学院 政策・メディア研究科 修士課程 2 年 CI プログラム

学籍番号:80331285 入野 仁志 irino@sfc.keio.ac.jp

概要

本研究は現在個別的に実現が行われているフォーマット変換を行わない映像転送システムにおける、複数の異種入出力の同時利用の実現と、拡張性の確保の実現を目的とする。本研究では映像機器が扱う複数種のフォーマットの共通部分の定義、及び対象とすべき入出力における共通部分の定義を行う。個別実現がされているフォーマット変換を行わない映像転送システムの入出力に依存した処理とフォーマットに依存した処理の分離を行うことで、システムの拡張性を確保する。また複数の異なった入出力や通信方法を同時に利用することのできる自由度の高い通信を実現する。

1 背景:広帯域環境下での映像転送

通信基盤としてのインターネットの普及及び、インターネットに利用される回線の広帯域化により、大容量の映像、音声の転送が多く利用されるようになった。広帯域環境下での利用を前提とした映像転送技術の中には、高品質映像転送を目的として、計算機と接続可能なデジタル映像機器から入力されるデータを他のフォーマットに変換することなく転送する映像転送技術がある。

2 問題意識

フォーマット変換を行わない映像転送システム既存実装は DV フォーマットを扱う DVTS、DV VoD や MPEG2-TS を扱う広島大学で開発されている robst、民生用機器を用いた MPEG2-TS 転送システムなどがある。これらの既存技術及び研究は

- 対象とする入出力、またはその通信方法
- 対象とするデータフォーマット

の 2 点以外はほぼ同一である。しかし、これらは個別的に実現されている。特定の入力・出力とフォーマットの組合せ毎に個別的な実現がおこなわれると以下の問題がある。

- 機能性の問題:異種入出力の利用

既存システムの場合、映像転送システムの利用者は状況に応じてシステムを使い分ける必要がある。また複数の入出力を同時に利用するといった概念を持たない為に、オーバーレイマルチキャストのような通信に対応できない。

- 拡張性の問題

入出力や通信方法とフォーマットの組合せが固定されているシステムは、上述した新たに登場したプロトコルなど新たな入出力や通信方法、または新たなフォーマットの登場などの技術的な進歩に対して追従しにくい。現状は各種システムがフォーマットや入出力毎に個別化しているため映像転送システムの開発者に対して、新たな機能の追加時の開発量を増大させる原因となる。

3 本研究のアプローチ

第 2 節で述べた問題は、入力、出力、フォーマットが個々のシステム内で依存関係を持っていることに起因する。問題の解決には、これまで実現されて来た映像転送機構の機能を入力・出力・フォーマットの 3 項を分解し、複数の入力と出力の組合せをフォーマットに依存することなく自由に組み合わせ可能な機構が必要とされる。本研究が提案する機構は以下の特徴を持つ。

- 一律的な入出力処理インタフェースの定義による、異種入出力の利用における透過性の確保とその同時利用の実現

- 入出力処理部分とフォーマット処理部分の分離
- 入出力先毎の個別の品質管理

本研究が提案する機構の概念図を図 1 に示す。

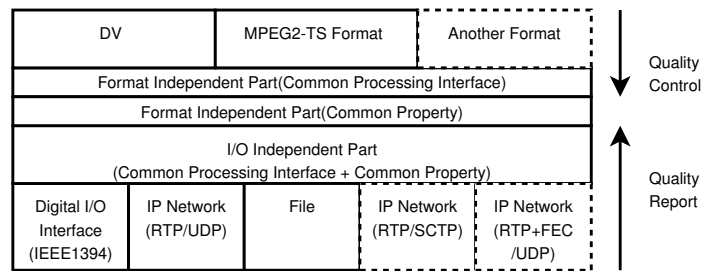


図 1: 入出力処理・フォーマット分離機構の概念図

4 設計

現在、多種多様な映像機器があるが、大衆性のある民生用映像機器の代表的な機器の比較検討を行い IEEE1394 インタフェースで接続する映像機器に焦点をあてた。またフォーマットはそれらで用いられる、固定長パケットフォーマットである DV と MPEG2-TS を対象とした。

4.1 本機構の構成要素

本論で提案する機構は以下の 4 つの機能に分離した部分から構成する。名称の右側に本論で用いる略称を記す。今後分離した機能の総称を本論ではオブジェクトと呼ぶ。

- データを送受する部分
 - － フォーマットデータオブジェクト:Format
対象とするフォーマットの実際に扱うデータの保持とその属性を持つ部分
 - － 入出力処理オブジェクト:IO
対象とする入出力に関する処理を行う部分
 - － フォーマット処理オブジェクト:Processor
対象とするフォーマットに依存した処理を行う部分
- 他のオブジェクトを管理する部分
 - － セッション管理オブジェクト:Session
全体の多重入出力管理と時間管理を行う部分

フォーマットデータオブジェクト、入出力処理オブジェクト、フォーマット処理オブジェクトが実際のデータの送受を行うセッション管理オブジェクトはこれらの 3 つのオブジェクトを制御する。セッション管理オブジェクトが他のオブジェクトを制御する状態を、図 2 に示す。

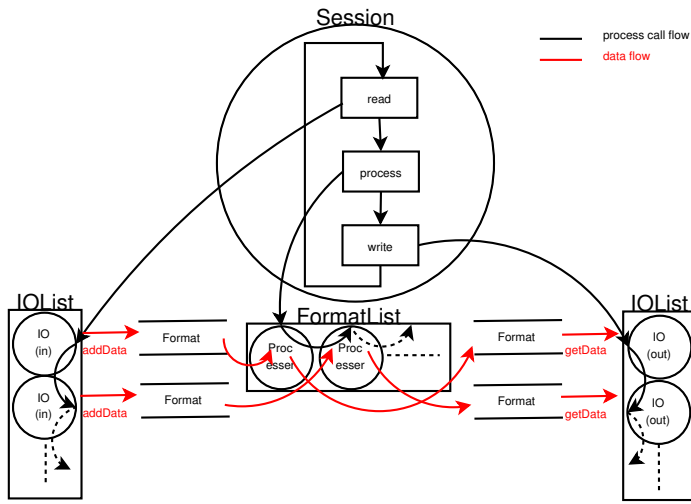


図 2: 機能呼び出しの順序とデータの流れの例

4.2 各オブジェクトの詳細

フォーマットデータオブジェクトが各フォーマット共通の属性情報として保持する情報を表 1 に示す。フォーマットデータオブジェクトが各フォーマット共通の処理インタフェースとして定義するメソッド名とその概要を表 2 に示す。入出力処理オブジェクトが各入出力処理共通の属性情報として保持する情報を表 3 に示す。入出力処理オブジェクトが各入出力共通の処理インタフェースとして定義するメソッド名とその概要を表 4 に示す。

表 1: フォーマットデータオブジェクトの保持する属性情報

属性名称
種類
PACKET サイズ
BLOCK サイズ
フレームあたり PACKET 数
BLOCK あたりフレーム数
総 BLOCK 数
映像周波数
変更データレート
変更データレート静的フラグ
エレメント数
エレメント識別子検知状態
エレメント識別子
初期バッファBLOCK 数
バッファBLOCK 最大数
現在読み込んでいる BLOCK 位置
現在書き込んでいる BLOCK 位置
データ保持領域

表 2: フォーマットデータオブジェクトの処理

メソッド名	概要
init	初期化
addData	データの追加
getData	データの取得
allocData	データ保持領域の生成
referData	データ保持領域の参照
detectId	エレメント識別子の検知

5 実装

実装環境として OS に Linux Kernel 2.4.26 を用い IEEE1394 の入出力に libraw1394 を用いた。本実装は VMT(Virtual

表 3: 入出力処理オブジェクトの保持する属性情報

属性名称
種類
実際の入出力の方向
入出力対象の名前
入出力機器内識別子
可能な転送 (同期性)
行すべき転送 (同期性)
転送のデータ量
扱うフォーマット
転送単位
転送間隔
最終送受信時間
ディレイ
ジッタ
入出力カウンタの総数
エラーカウンタの総数

表 4: 入出力処理オブジェクトの処理一覧

メソッド名	概要
init	初期化
open	開く
close	閉じる
read	読む
write	書く
allocBuffer	データ領域確保
freeBuffer	データ領域開放

Method Table) を利用して、異種の入出力に対して同一の処理インタフェースを提供した。またこの機構をライブラリとして実装し、さらにそのライブラリを用いたサンプルアプリケーションの実装を行った。図 3 にサンプルアプリケーションのコマンドラインインタフェースを示す。サンプルアプリケーションは入力用パラメータ、出力用パラメータは URI 形式を用いて表現する。各パラメータの URI 形式の書式はプロトコル名に”file”, “ieee1394”, “rtsp” を, “?” 以降に属性を指定可能である。複数の入出力を扱う場合には,” を区切り文字に用いる。

```
./gavotte -I 入力用パラメータ (URI) -O 出力用パラメータ (URI)
[-F 利用フォーマット]
```

図 3: サンプルアプリケーションコマンドラインインタフェース

6 評価

6.1 開発量の比較

開発にかかるコストの比較として、ソースコードの行数の比較と初期手順数の比較を行う。DVTS は IEEE1394 からデータを受信しネットワークへ送信する dvsend, ネットワークからデータを受信し IEEE1394 へ送信する dvrecv IEEE1394 からデータを受信しファイルへ保存する dvsave ファイルからデータを読み出し IEEE1394 へ送信する dvplay が含まれる、本実装はこの全ての用途に利用するため本基盤部分の実装及びサンプルアプリケーションの行数の総和と、DVTS に同梱される dvsend, dvrecv, dvsave, dvplay の総和を比較した。

その結果本ライブラリ実装は約 4950 行、及びサンプルアプリ

ケーションとしてライブラリ外での実装が 134 行となった．既存実装である DVTS は約 5340 行であり，ライブラリを利用すると既存実装と同等以上の機能をライブラリ利用者は既存の方法と比べて非常に少ない開発量で実現できる．

また同様に初期手順数を比較した．本実装は DVTS を構成する 4 アプリケーションと同等以上の機能を統一された 7 手順で実現可能である．

1. 入力用の入出力オブジェクトリストの初期化
 2. 出力用の入出力オブジェクトリストの初期化
 3. 入出力処理オブジェクトインスタンス毎に初期化．フォーマットデータオブジェクトのデータ領域確保と関連付け，リストへの追加
 4. フォーマット処理リストの初期化
 5. フォーマット処理オブジェクトのインスタンス毎に，インスタンス生成，リストへの追加，終端フラグの設定
 6. セッション管理オブジェクトの生成と初期化と入力用入出力オブジェクトリスト，出力用入出力オブジェクトリスト，フォーマット処理リストの関連付け
 7. セッションの開始

図 4: 本機構を用いた場合の処理の手続きの流れ

一方既存アプリケーションである DVTS に同梱される 4 つのアプリケーションはその初期手順もそれぞれ個々独立しており，その手順数の総和は 19 手順となっている．本機構は入出力・フォーマットのオブジェクトの独立性が高くかつ拡張が可能な設計をしており，それらのオブジェクトの再利用性が高いことが本評価から示された．

本機構が実現する入出力の組合わせを表 5，既存の複数の実装で実現できる入出力の組合わせを表 6 に示す．

表 5: 本実装の入出力の組合せ

	ファイル	RTP/UDP	IEEE1394
ファイル	-		
RTP/UDP			
IEEE1394			-

表 6: 既存実装の入出力の組合せ

	ファイル	RTP/UDP	IEEE1394
ファイル	-		
RTP/UDP			
IEEE1394			-

本機構は約 4950 行で実現で全ての入出力を実現しているが，既存実装は DVTS が約 5340 行，ファイルから RTP/UDP への送出を実現する DV VoD システムは約 920 行である．本機構は機能を拡張しつつ，少ない開発量で実現している．

7 研究成果と今後の課題

7.1 研究成果

本研究は，映像機器が用いるデータフォーマットを中間的なフォーマットに変換することなく転送する映像転送システムにおいて，入出力処理とフォーマットに関する処理の分離による入出力機能の同時利用と拡張性の確保を実現した．本研究において実現された機構は複数の異種の入出力の対象に異なった品質で入出力を可能にするために，1) 入出力の属性及びを保持し

処理を担当する部分，2) フォーマットに依存した属性を保持する部分，3) フォーマットに依存した処理を行う部分，4) 全体の多重入出力と時間管理を行う部分に機能を分割し設計，実装を行った．この機構を利用して映像転送システムを開発する場合，本機構の利用により，開発が軽減される．本機構を拡張する場合追加するフォーマットや入出力固有の部分を作成すれば良い．映像転送システムの利用者が本研究の成果を利用する場合は，複数の入出力の多重化可能であり，その操作が統一される利点がある．本研究の設計上の制限として，対応可能なフォーマットに一部制限がある．対応可能なフォーマットの条件として，固定長パケットを基本単位としたフォーマットであり入出力部分に自己のデータ構造を写像しないフォーマットである．

7.2 今後の課題:外部の情報の利用

本論は特定のフォーマットと入出力の組合せによる映像転送の実現に問題の焦点をあて，それを分離する方法に関して述べた．実際の利用時には様々な情報の指定が必要となる．特に一般に映像転送アプリケーションの情報は SDP によって記述され RTSP や SIP などを用いて交換される場合が多い．それらのプロトコルとの連係や親和性の確保に関して今後取り組む必要性がある．