

フレキシブル制作システム

竹内真也

Flexible Program Production System

Shinya TAKEUCHI

ABSTRACT

Broadcasting operations have relied on VCR tape. Progress in hard disk and memory technology is prompting a move to file-based non-linear editing systems and program material transfer systems on IP networks. We are doing research on an advanced file-based production system that streamlines the production workflow. With this system, producers can easily access large amounts of program materials and edit decision lists from anywhere on the network. The system's image processing functions and capacity are scalable.

1. まえがき

放送局では、これまで番組の制作や保存にVTRテープを用いてきたが、最近是非ンリニア編集機を用いた番組制作やネットワークを利用した番組素材転送など、ハードディスクやメモリーを利用したファイルベースシステムへの移行が進んでいる¹⁾。ファイルベースシステムによって番組制作のワークフローは変化したが、現状では以下のような課題がある。

・番組制作者は番組素材と編集データを管理し、作業を行う編集機にこれらのデータや素材を移動する必

要がある。

- ・利用する編集機ごとに性能や機能に差があり、インターフェース、操作方法が異なる。
- ・ノンリニア編集機は、一般的に、レンダリング*¹の時間が長い。
- ・ノンリニア編集機を占有して編集作業を行うので、機器配備を行う際には最大稼働数を見込んで配備しなければならない。
- ・ノンリニア編集機の性能や機能は機器を導入する時点でほぼ固定され、柔軟な変更や更新が難しい。

そこで、ファイルベースシステム導入後の将来の放送局システムとしてフレキシブル制作システムを提案する。フレキシブル制作システムでは、ファイルベ

* 1 タイムライン上の映像や音声を1つの映像・音声ファイルに出力すること。

システムの課題を解決し、効率的なワークフローを実現する。

2. フレキシブル制作システムのアーキテクチャー

ファイルベースシステムが導入され、番組素材や放送番組がすべてファイル化されると、それらのファイルは拡張が可能な分散ストレージ*2に保存されることが考えられる。また、ファイルの処理を容易に行うために、放送局の機器のソフトウェア化が進み、これらがすべて広帯域のネットワークで接続されることが考えられる。ユーザーが特別なハードウェアやソフトウェアを導入することなく利用でき、システム管理者が機器の管理と拡張、更新を容易に行うことのできるシステムが要求されている。

データやソフトウェアのリソースを集約して仮想化し、ユーザーはコンピューターをネットワークに接続するだけで、集約されたリソースの中から必要なものだけを利用する形態をクラウド・コンピューティングと言う。フレキシブル制作システムは放送局のクラウド・コンピューティング化と言うことができる。

提案するフレキシブル制作システムのアーキテクチャーを1図に示す²⁾。ユーザーインターフェースを担うフロントエンドと、編集処理を担うバックエンドで構成される。放送局の編集機能を持つサーバーや番組素材が保存されているサーバーはすべて広帯域のネットワークで接続されている。

フロントエンドにウェブブラウザを用いたインターフェースを採用することで、ユーザーはどこからでも同じインターフェースを用いて編集作業を行うことができる。また、ユーザーはインターフェースをある程度自由に変更でき、それを保存し呼び出すことも可能である。

バックエンドでは、グリッド・コンピューティング技術*3を用いて編集処理を複数のサーバーに分散する。編集機能を共有することでリソースの効率的な利用や冗長系の確保、柔軟な編集機能の追加が可能となる。また、リソースの管理や保守を容易に行うことができるほか、多数のユーザーからのアクセスによる処理の負荷分散が可能となる。フロントエンドでユーザーが操作すると同時に、バックエンドで編集処理をカットごとに並列して行うので、レンダリング時間を短縮することができる。

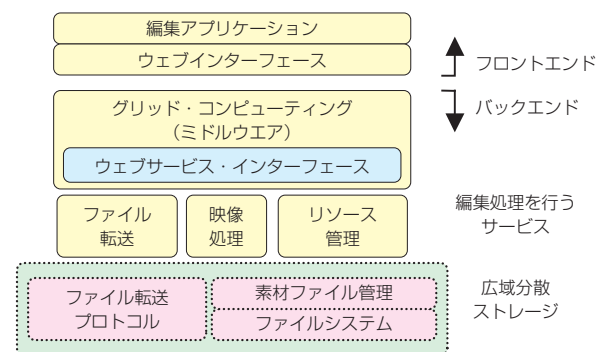
分散したファイルサーバーにすべての番組素材を保存し、仮想的に1つのストレージとして扱う機構を持

つ広域分散ストレージシステムによって、どこからでも同じように番組素材を利用することができる。また、番組素材を共有することでユーザーによる管理の手間と移動やコピーの回数を低減することができる。

3. フレキシブル制作システムの構成と動作

システムの全体構成を2図に示す。フロントエンドはウェブブラウザを用いた編集インターフェースと編集用ウェブサーバーで構成される。編集用ウェブサーバーは編集記述*4を出力し、バックエンドに編集処理を要求する。バックエンドはグリッド・コンピューティングによって編集処理を管理・実行するサービス群と広域分散ストレージシステムで構成される。編集処理を管理・実行するサービス群は編集記述に従って各サービスに処理を要求する処理クライアント、リソース選択サービス、サービスとサーバーのリソースを管理するMDS (Monitoring and Discovery System)、指定された映像フレームの転送と書き込みを行うファイル転送サービス、指定された映像フレームに映像加工を行う映像処理サービスで構成される。

ユーザーPCのウェブブラウザに提供される編集インターフェースで編集作業が行われると、編集用ウェブサーバーから編集作業に応じた編集記述が出力される。編集用ウェブサーバーが編集記述を出力すると、処理クライアントはその記述を解析し、リソース選択サービスに処理に必要なサービスを問い合わせる(①)。各サービスはMDSにどのような処理を行うサービスで



1図 アーキテクチャー

- * 2 複数のファイルサーバーをネットワークで接続し、仮想的に1つのストレージとして扱えるもの。
- * 3 ネットワーク上のさまざまなコンピューターリソースを統合して、仮想的に1つのコンピューターリソースとして扱う仕組み。
- * 4 編集作業の内容を番組素材名やタイムコードなどとともに記述したデータ。

あるのかという情報と負荷に関する情報をあらかじめ登録しているので、リソース選択サービスはMDSに登録されている情報を参照して(②)、処理を行うサービスを決定し、サーバーのアドレスを処理クライアントに通知する(③)。処理クライアントが通知されたサービスに対してファイル転送または映像編集の処理要求を行い、その処理が実行される(④)。リソース選択サービスを用いることで、サーバーの負荷などに合わせてサービスを割り当てることが可能となる。

4. システムの実装と各部の説明

システムで扱う番組素材をすべてMXF (Material Exchange Format)³⁾*5を用いてファイル化する。フォーマットはOP-Atom⁶⁾*6または単一コンテンツを記録したOP1a⁷⁾*7とする。フレームラッピング⁸⁾*8を用いてMXFファイル化することで、映像フレーム単位で番組素材にアクセスすることが可能となる。

- * 5 SMPTEで規格化された番組素材ファイルの交換を目的とした伝送用ファイルフォーマット。
- * 6 番組素材の映像や音声の複数のトラックを別々のファイルに保存するデータ構造。MXFではデータ構造の複雑さをレベル分けしたものをOP (オペレーショナル・パターン) と言う。
- * 7 番組素材の映像や音声の複数のトラックを1つのファイルに保存するデータ構造。
- * 8 番組素材のデータをフレーム単位で区切って保存する方式。
- * 9 ウェブブラウザ上で動的な表示をしたり対話的な動作をさせたりするのに適したプログラミング言語。

4.1 編集インターフェース

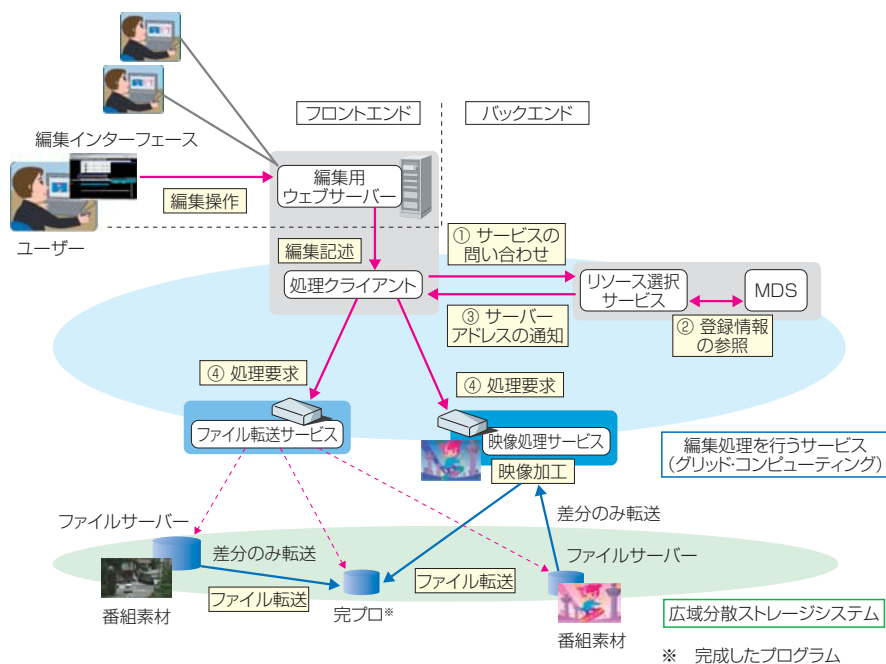
編集インターフェースにはウェブブラウザを用い、ノンリニア編集機のようにタイムライン上に素材のカットを時系列に並べて編集作業を行う。HTML (Hypertext Markup Language) とJavaScript⁹⁾だけを用いて実装しており、画面遷移を行わずに非同期処理を行うことができるので、バックエンドの処理を待たずに次の操作に移ることができる。

編集用ウェブサーバーは編集インターフェース上のタイムライン全体を編集記述として保存し、編集インターフェースで操作が行われる度に編集作業を編集記述として出力する。

通常のノンリニア編集機の場合には、編集作業の最後にレンダリングを行い出力ファイルを生成するが、このシステムでは、インターフェース上で操作が行われる度に、逐次、処理をバックエンドで並列して実行するので、レンダリングの待ち時間を短縮することができる。また、編集中の番組であっても編集記述として保存されているので、これを呼び出すことで別のウェブブラウザからでも編集作業を継続することができる。3図は編集用インターフェースの例、4図は編集記述の例である。

4.2 グリッド・コンピューティングによる編集処理を行うサービス

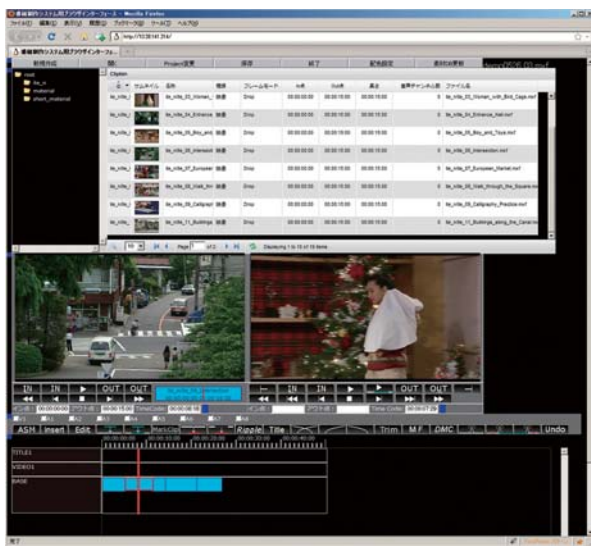
グリッド・コンピューティングとは、先にも述べたように、ネットワーク上のさまざまなコンピューター



2図 システムの全体構成

リソースを統合して、仮想的に1つのコンピューターリソースとして扱う仕組みである。グリッド・コンピューティングは分散コンピューティングの基本機能であるサーバーの管理や、ユーザー認証によるプロセスの実行権限などのセキュリティー機能を備え、サービスの追加や拡張、スケーラビリティーを確保する機能を有する。

処理クライアントと編集処理を行うための各サービスをグリッド・コンピューティングの標準的なライブラリーであるGlobus Toolkit⁴⁾を用いて実装した。



3図 編集インターフェースの例

Globus Toolkitが提供するMDSは各サーバーの負荷状態やOSの情報を収集するためのサービスであるが、本システムでは処理内容によってサーバーを選択する必要があるため、サーバーの持つサービスの処理内容情報を収集する機能を追加した。

ファイル転送サービスはファイルサーバーと映像処理サービスの間の映像フレーム転送を管理するサービスである。処理クライアントから指定された映像フレームだけを番組素材が保存されているファイルサーバーから完プロ（完成したプログラム）を保存するファイルサーバーに転送する。映像処理が必要な場合には、指定された映像処理サービスを経て転送する。番組素材ファイル全体の転送を行わないので、バックエンドの処理全体を高速に行うことができる。

映像処理サービスは内部に映像加工を行うプロセスを持ち、処理クライアントで指定されたパラメーターに従って、転送された映像フレームの映像加工を行う。どのような映像加工を行うかはサービスによって異なるが、処理クライアントはリソース選択サービスに通知されたサービスを用いればよい。

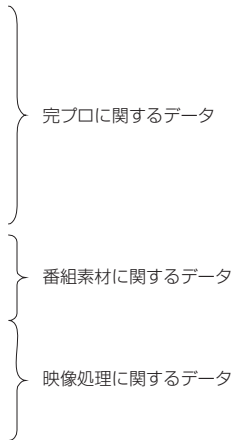
4.3 番組素材ファイルの管理

ファイルベースシステムの普及に伴って増加する膨大な番組素材ファイルを保存し、共有して利用できるようにするためには、複数のファイルサーバーを仮想的な1つのストレージとして扱うことのできる分散ストレージシステムが有効である。分散ストレージシ

```

<?xml version="1.0" encoding="utf8"?>
<project>
  <project_id>100</project_id>
  <history_no from="6" to="7"/>
  <project_file_name>giken.mxf</project_file_name>
  <frame_mode>0</frame_mode>
  <start_timecode>0</start_timecode>
  <total_frame_num>360</total_frame_num>
  <edit new_frames="110" target_in="250" target_out="250">
    <track id="1" type="0">
      <cut id="9" type="0">
        <base_in>250</base_in>
        <base_out>360</base_out>
        <source type="video">
          <file>river.mxf</file>
          <in>171</in>
          <out>288</out>
        </source>
        <processor type="color">
          <parameter name="hue">10</parameter>
          <parameter name="lum">20</parameter>
          <parameter name="sat">10</parameter>
        </processor>
      </cut>
    </track>
  </edit>
</project>

```



4図 編集記述の例

テムで番組素材ファイルを管理する方式として、構造型P2P (Peer-to-Peer)^{*10}の管理方式がある。P2Pによるファイル管理方式には中央集権的な管理機構が無いので、管理コストが低減でき、障害耐性や拡張性が優れている。しかし、一般に、P2Pに参加しているサーバーの管理テーブルを分散して管理しているので、システム障害が生じたときに障害個所の特定が迅速にできないという問題がある。

DHT (Distributed Hash Table : 分散ハッシュテーブル)^{*11}を用いた構造型P2Pの1方式としてOneHop⁵⁾が提案されている。OneHopでは、オーバーレイネットワーク^{*12}に参加するすべてのサーバーが、参加しているすべてのサーバーのテーブルを持っているので、参加しているサーバーを常時どこからでも把握でき、障害時の対応が比較的容易である。また、ファイルの検索は各サーバーが保持するテーブルを使って完了するので、高速な検索が可能である。しかし、物理ネットワークの構造を考慮していないので、広域な分散ストレージシステムを構成すると、局間ネットワークに多くの管理パケットが流れることになる。そこで、P2P

- *10 決められたルールに従ってファイルの管理や検索を行い、仮想的なネットワークを構築するP2P。
- *11 ハッシュ値 (ファイルごとに異なる値で、そのファイルを代表する値) を用いてデータの検索を行うハッシュテーブルを複数のノード (クライアント機能とサーバー機能を併せて持つPCや通信端末のこと) で分散管理する手法。
- *12 P2Pアプリケーションが物理的なネットワーク上に構築する仮想的なネットワーク。
- *13 インターネットにおける標準の転送プロトコルの1つ。受信側が送信側にデータ受信を確認する機能や再送を求める機能を持たない通信方式で、高速な転送や1対他の通信で使用される。
- *14 インターネットにおける標準の転送プロトコルの1つ。再送によるエラー訂正機能などを持ち、信頼性の高いデータ転送ができる。

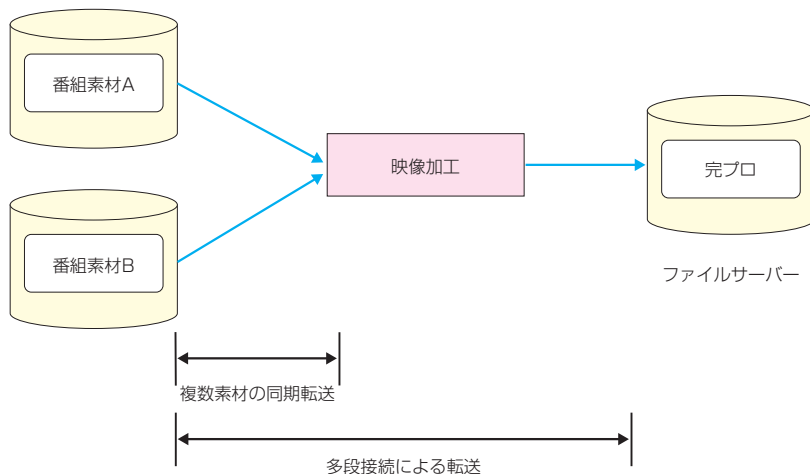
のノード管理に物理的なネットワークの位置を考慮させ、局間ネットワークに流れるパケットの量を少なくするノード管理方式を開発した。また、各ファイルサーバーには保存するファイル数に比例して管理するファイル数を割り当てるファイル管理方式を開発した⁶⁾。更に、現在のコンピュータで一般的な階層ディレクトリーによるファイルアクセスを可能とするために、分散ディレクトリー管理手法を提案した⁷⁾。分散ディレクトリー管理手法を用いることで、ユーザーは互いに共通のディレクトリー名でどこからでもファイルへアクセスすることができる。なお、分散ディレクトリー管理方式はファイル管理方式と同様に中央集権的な管理機構が無い構造とした。

4.4 転送プロトコル

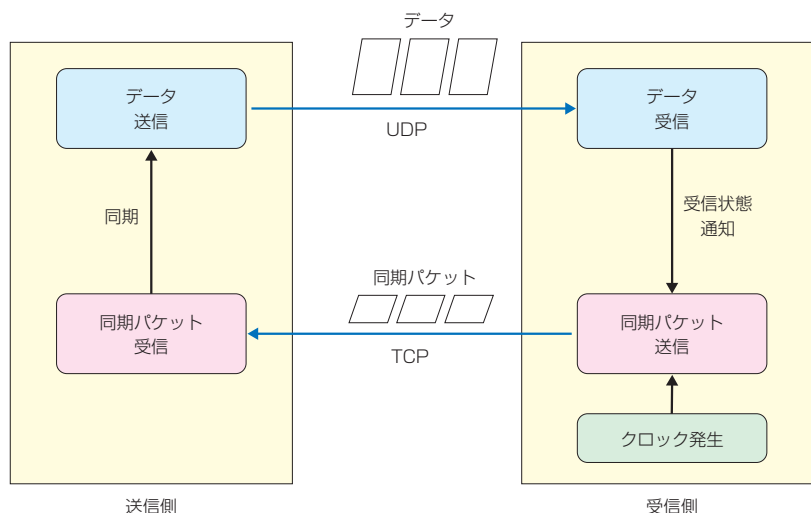
映像加工を行う場合、ファイルサーバーは転送の要求に従ってサーバー間で編集に必要な映像フレームの転送を行う。異なるサーバーに保存されている番組素材ファイルを加工するためには、複数の素材間で同期をとり、映像加工を含めた多段接続での転送が必要である (5図)。

UDP (User Datagram Protocol)^{*13}を用いたファイル転送にはエラー耐性が無く、同期伝送や多段接続の機構も無い。TCP (Transmission Control Protocol)^{*14}を用いたファイル転送では、ふくそう制御機構が原因となり、長距離転送でその帯域を満たすだけの性能が出ない。そこで、上記要件を満足するファイル転送プロトコルとしてFMFTP (Flow Media File Transfer Protocol) を提案した⁸⁾。

FMFTPでの送受信部の構成を6図に示す。受信側からTCPを用いて送信を要求するデータのブロック番号



5図 システムに必要な転送



6図 FMFTP送受信部の構成

を記した同期パケットをクロックの間隔で送出する。送信側は同期パケットに同期して要求されたブロックのデータパケットをUDPで送信する。受信側でクロックの間隔を変化させて同期パケットの送出間隔を制御して転送速度の制御を行う。データの送信にはUDPを用いているので長距離での高速転送が可能である。また、同期パケットをスルーして送信することで多段接続の転送が可能である。

なお、FMFTPのファイル転送プロトコルはITU-Tで標準化を行い、ITU-T Rec. J285として勧告された⁹⁾。

4.5 ファイルシステム

転送された映像フレームをファイルサーバーの完プロファイルに挿入または上書きする必要がある。また、完プロファイルはファイル転送サービスの要求に従って映像フレーム単位で削除する必要がある。しかし、既存のファイルシステムには部分的なデータの挿入や削除の機能は無いので、ファイルすべてを書き換える必要があり、時間を要する。そこで、ファイルシステムが管理するブロック単位で挿入・削除が可能なファイルシステムを開発した。開発したファイルシステムでは、ファイルのデータを管理するエクステンツリー^{*15}を直接操作できるので、実データを書き換えることなくファイル内の部分的なデータの挿入・削除と、ファイル間のデータ移動を高速にできる。すなわち、MXFファイルを映像フレーム単位で高速に挿入・削除できるシステムを実現した¹⁰⁾。なお、このファイルシ

テムはLinux OSで用いられるext4ファイルシステム(Fourth Extended File System)を拡張することで実装した。

5. あとがき

ファイルベースシステムを更に発展させた次世代番組制作システムとして提案したフレキシブル制作システムについて述べた。フレキシブル制作システムのアーキテクチャーを提案し、システムの実装を行った。また、編集インターフェース、グリッド・コンピューティングを用いた編集処理のための各サービス、分散ストレージシステムの管理方式、放送素材用ファイル転送プロトコル、映像フレーム単位で挿入・削除ができるファイルシステムについて述べた。本システムを用いることで、将来的に増え続ける膨大な番組素材や編集データの利用と保存管理が可能となり、制作機能や性能を柔軟に追加、増強、更新することやリソースを有効に利用することが可能になる。また、ユーザーには場所や操作性に拘束されないユーザーごとの作業環境を提供することができる。今後、本システムの実用化と普及を目指し、クラウド時代の制作システムの1つの標準仕様とするために、サービスや編集記述のオープンソース化を行う予定である。

*15 エクステンツリーとはファイルシステムのディスク上の物理アドレスとファイル内の論理アドレスを管理する単位であり、ファイルのデータは複数のエクステンツリーによるツリー構造で管理される。

参考文献

- 1) 石田：“テープレス化の技術動向,” 映情学誌, Vol.63, No.1, pp.30-35 (2009)
- 2) 竹内, 金子, 黄, 和泉：“グリッド・コンピューティング技術を用いた番組編集システムの検討,” 信学ソ大, B-6-20 (2009)
- 3) SMPTE 377M, “Material Exchange Format (MXF) File Format Specification”
- 4) The Globus Toolkit, <http://www.globus.org/toolkit/>
- 5) A. Gupta, B. Liskov and R. Rodrigues：“One Hop Lookups for Peer-to-Peer Overlays,” 9th Workshop on Hot Topics in Operating Systems (HotOS-IX) (2003)
- 6) 金子, 竹内, 南, 和泉：“OneHop-P2P拡張方式の実装方法と性能評価,” 信学技報, NS2008-52, pp.57-62 (2008)
- 7) 金子, 黄, 竹内, 和泉：“構造型P2Pを使った分散ファイルシステムにおける分散ディレクトリ管理手法,” FIT2009, L-033 (2009)
- 8) 黄, 金子, 竹内, 和泉：“長距離・広帯域IPネットワークにおける優先度に応じたファイル転送プロトコルの提案,” 映情学冬大, 3-11 (2009)
- 9) ITU-T Rec. J.285, “Architecture for synchronized programme transfer with pull operation over IP-based networks” (2007)
- 10) 金子, 南, 黄, 竹内, 和泉：“挿入削除機能付きファイルシステムによるMXFファイル編集の性能評価,” 映情学年次大, 10-4 (2009)



たけうちしんや
竹内真也

1999年入局。大阪放送局を経て、2002年から放送技術研究所にて、ITとネットワークを利用した放送局システムの研究に従事。現在、放送技術研究所次世代プラットフォーム研究部に所属。