Vol.2021-DPS-186 No.73
Vol.2021-CSEC-92 No.73
2021/3/16

非集中型モデルによる動画配信プレーヤー用 個人視聴データ取得・蓄積モジュールの試作

関根大輔[1]　松村欣司[1]　藤井亜里砂[1]

概要：放送やネット配信による動画コンテンツの視聴履歴に関するデータ（以下，視聴データ）を，様々なサービスや体験の質の向上に活用するための技術の検討を進めている．本稿では，ネット配信の視聴データをユーザ自身で取得・蓄積し，管理する手法を提案する．この手法では，Web ブラウザ上の動画プレーヤー内にある HTML5 の video 要素が出力するイベントデータを用いて視聴データを構成する．従って，ユーザは視聴データを，動画を配信するサービス事業者が管理するシステムと独立させて取得と蓄積することが可能である．また，この手法は多くの視聴デバイスやネット動画プレーヤーの実装形態に適用可能である．今回，video 要素からイベントデータを取得し，ユーザ指定の領域へ蓄積するモジュールの試作および動作検証を行い，HTML5 対応ブラウザ上で提案手法が実装可能であることを確認した．

キーワード：視聴データ，パーソナルデータストア，非集中型モデル, HTML5

Prototype of a video streaming player extension for acquiring personal viewing history in decentralized data store model

DAISUKE SEKINE[†1]　KINJI MATSUMURA[†1]
ARISA FUJII[†1]

Abstract: Utilizing personal viewing history of broadcast and Internet media content can improve quality of online services and enhance experiences in daily life. This paper proposes a method in a decentralized data store model to acquire personal viewing history of online streaming video services, in which the user owns and has a control of their data. In this method, by obtaining event data from a video element of HTML5 in a video streaming player on a Web browser, user's viewing history can be stored separately from the content provider's system, and this method has also the potential to support multiple viewing devices and player implementations. A prototype of a video streaming player extension is developed. Implementation of the proposed method and verification test are conducted on software module. We confirm that it is applicable to existing HTML5 compliant web browsers.

Keywords: viewing history, personal data store, decentralized model, HTML5

1. はじめに

ユーザが各種のネットサービスを利用することでサービスログが生まれる．このようなログの中には，ユーザ個人の日常の生活行動や嗜好を表すパーソナルデータが含まれており，これを活用することで生活行動の振り返りや新たな知識の獲得などの生活の質の向上に役立つことが期待できる．また，パーソナルデータが増えるほどユーザの特徴が詳細に抽出できることになり，効果は高まる．しかし，現在パーソナルデータの多くはサービスを提供する事業者が主体となって取得・管理しており，ユーザ自身が様々な用途に利用したり，事業者を横断するサービスで利用することには制限がある．

近年，このような事業者主体のモデルに対して，ユーザのパーソナルデータの所有権やコントロール権を明確にして，ユーザ主体でパーソナルデータを管理するモデルが欧州を中心に検討されている．このモデルでは，Personal Data Store（PDS）と呼ばれるユーザごとに所有するストレージに，ユーザのパーソナルデータを一元的に集約して管理し，必要に応じて事業者のサービスからの利用を許可する．本稿ではこのモデルを非集中型モデルと呼ぶこととする．日本においても，非集中型モデルによるパーソナルデータ活用として[1]が社会実装を始めている．

我々は，パーソナルデータの中でも，ユーザの趣味嗜好と関連が強いと考えられる「いつ，どのコンテンツを見たか」を示す視聴データに着目し，非集中型モデルにおいてユーザが PDS を用いて主体的に視聴データを管理および利活用できる仕組みの研究を進めている[2]．この仕組みでは，図 1 に示すように放送とネットの両方から幅広く視聴データを取得し PDS に蓄積した視聴データを，ユーザの意思に基づき様々な外部サービスと連携させる．視聴データの利活用例には，コンテンツサービスの質の向上やユーザの視聴体験の向上などがあげられる．非集中型モデルの導入により，事業者に対して視聴データ等のパーソナルデー

1 日本放送協会
NHK (Japan Broadcasting Corporation)

タの提供を前提とするという，サービス利用時におけるユーザの抵抗感の解消につながる．

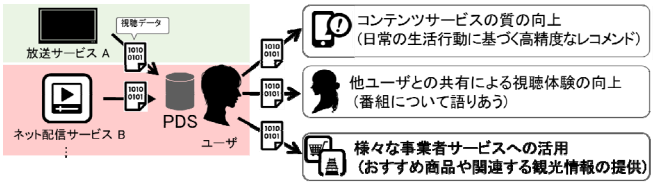

図 1　視聴データ利活用ユースケース

コンテンツ視聴においては伝送路の違いや，テレビやスマートフォン（以下，スマホ）など視聴端末の区別によって視聴する仕組みが異なるため，それぞれ適切な視聴データの取得方法が異なる．これまで我々は[3]で，放送受信機と携帯端末を Wifi 通信で接続する端末連携技術を用いて，放送のコンテンツの視聴データを個人単位で取得する手法を提案した．さらに，ケーブルテレビの専用受信機であるセットトップボックスに対して提案手法を検証する実験を一般家庭の実環境下で実施し，提案手法が放送のコンテンツ視聴時に個人の視聴データを蓄積し，ユーザ自身の視聴傾向を再確認するための実現性の高い方法であることを示した．本稿では，ネット経由で配信されるコンテンツであるネット動画の個人の視聴データをユーザ主体で取得し，PDS に蓄積する手法を提案する．

本稿では，まず 2 章にて，ユーザの視聴データを収集する関連研究について述べる．3 章では，非集中型モデルで視聴データを蓄積する趣旨，現行のネット動画配信サービスにおける視聴データの扱いとの違いを比べる．また，上記をもとに視聴データ蓄積手法の要求条件を整理する．4 章では，3 章で整理した要求条件を満たす視聴データ蓄積手法を提案する．5 章では，提案した手法を実装した視聴データ取得・蓄積モジュールを試作し，その動作検証の結果を紹介する．最後に 6 章で，今後に向けた取り組みを述べ，本稿をまとめる．

2. 関連研究

あるユーザの過去の視聴履歴に基づいたネット動画の推薦などのために，事業者がユーザの視聴データを取得および蓄積する仕組みが開発され，実サービスとして運用されている[4].

また，ユーザの視聴行動を記録する研究として[5][6]がある．[5]は，スマートテレビ，スマホ，スマートウォッチで構成されるロギングシステムを提案し，ユーザのテレビ視聴行動を把握することができることを示した．[6]は，テレビの IR 信号ログ，IP のパケットデータ，Bluetooth 信号センサや照度センサのデータを組み合わせて，家庭内のテレビの視聴行動に費やした時間を推定できることを示した．

以上の先行研究では，[4]は，視聴データを事業者自身のサービスへの活用を目的とするため，事業者が管理するシステムで視聴データを取得する仕組みである．また，[5][6]は，ユーザ側で視聴データを記録する手法を提案しているが，コンテンツの視聴時間を把握するための手法であり，いつ，どのコンテンツを見たかという視聴データの生成を目的としていない．本研究の新規性は，ネット動画視聴時に，ユーザ主体での視聴データの取得と，どのコンテンツを見たかということを特定できる視聴データの取得を両立させることである．

3. 非集中型モデルにおける視聴データ蓄積手法の検討

3.1 非集中型モデルにおける視聴データ活用の特徴

非集中型モデルでは，ユーザが自身のパーソナルデータを取得し，PDS に蓄積する．また，ユーザは自身の判断のもと，PDS 内に蓄積されたパーソナルデータを利用する権限を事業者のサービスに与えることで，パーソナライズされたサービスを享受する．非集中型モデルの特徴は，ユーザが全てのパーソナルデータを管理するため，パーソナルデータの種別と利用するサービスが特定の組み合わせに制限されないことである．そのため，事業者や業種をまたいだデータの連携の可能性が広がることが期待できる．

今回検討する視聴データの蓄積手法も上述の非集中型モデルの考えに従って設計する．非集中型モデルでは，ユーザが視聴したコンテンツの視聴データを，ユーザの意思のもと取得し，PDS に蓄積する．また，蓄積する視聴データには，タイトルやキーワードなどのコンテンツそのものに関する情報に加え，再生，停止やスキップ操作等のユーザの意図の導出に繋がる情報も含める．その理由は，視聴データの取得時点では，どのサービスでどのような視聴に関する情報を利用するか未確定であるからである．放送やネットで提供されるコンテンツのジャンルや内容は多岐にわたるため，その視聴データはコンテンツ視聴以外の多様な場面で活用できることが想定される．例えば，旅行番組の視聴データとスケジュールデータを連携させることで，視聴したコンテンツで紹介された旅行先のおすすめレストランを推薦するサービスなどが考えられる．このような多様なサービスへの適用性を担保するうえで，ユーザの意図の導出に繋がる情報を持つ視聴データが有効となる．

3.2 ネット動画配信サービスの状況

近年，YouTube[7], Netflix[8], TVer[9], NHK プラス[10]など，様々な事業者がネット動画配信サービスを提供している．これらの事業者は視聴用アプリであるネット動画プレーヤーをスマホ，PC，テレビなどのデバイス向けに提供しており，その実装形態としては各デバイスの OS に依存するネイティブアプリや，OS 非依存な Web ブラウザ上で動作する Web アプリがある．このようにデバイスと実装形態の組み合わせが多数存在するなかで，各事業者はターゲ

ットとするデバイスと実装形態を選択し，それに応じたネット動画プレーヤーを開発し運用している．そのうえで多くの事業者は，提供するネット動画の視聴傾向の分析や，ネット動画のレコメンドなどの自社サービスの改善のために，ネット動画プレーヤーにサービス利用者の視聴データを収集する仕組みを組み込んでいる．

3.3 非集中型モデルにおける視聴データ蓄積手法の要求条件の整理

3.1 で述べた非集中型モデルにおける視聴データ活用の特徴と 3.2 で述べたネット動画配信サービスの状況に基づいて，非集中型モデルでの視聴データ蓄積手法に求められる要求条件を整理する．

3.1 と 3.2 で示したように非集中型モデルで扱う視聴データと，事業者が収集する視聴データは利用目的と蓄積場所が異なる．そのため，非集中型モデルでは，視聴データをネット動画配信サービス提供者の仕組みとは独立して，ユーザが管理する PDS に蓄積できることが必要となる．

また，3.1 で述べたように，多様なサービスへの視聴データの適用可能性を担保するために，ネット動画そのものに関する情報に加え，ユーザの意図の導出に繋がる詳細な視聴データを取得できることが求められる．

さらに，3.2 に示したように，ユーザのネット動画視聴環境はデバイスやネット動画プレーヤーの実装形態の組み合わせによって様々である．ユーザにとっては，各種のネット動画配信サービスで視聴したコンテンツの視聴データをできる限り多く PDS に蓄積することで，視聴データの利活用により享受するサービスや体験の質の向上が期待できる．しかし，ネット動画プレーヤー個別に視聴データを取得する機能を実装すると，開発および管理のコストが大きくなる．そのため，非集中型モデルでの視聴データの蓄積手法には，各種のネット動画配信サービスで，より多くのデバイスと実装形態の組み合わせに適用できることが求められる．

これらの要求条件をまとめると，以下の通りになる．

要求条件 1

視聴データを，ネット動画配信サービス提供者の仕組みとは独立して，ユーザが主体的に PDS に蓄積できること

要求条件 2

ネット動画プレーヤーの再生や停止操作等のユーザの意図の導出に繋がる視聴データを取得できること

要求条件 3

各種のネット動画配信サービスで，より多くのデバイスと実装形態（ネイティブアプリや Web アプリ）の組み合わせに適用できる手法であること

4. 非集中型モデルにおける視聴データ蓄積手法の提案

3.3 で整理した要求条件を満たす視聴データの蓄積手法を提案する．この手法では，3.2 で述べた実装形態のうち，Web アプリとして実装されるネット動画プレーヤーと連携して視聴データの取得および蓄積を行うモジュールを導入する．図 2 に提案する取得･蓄積モジュールの構成を示す．

図 2 において左側の青枠で示す部分は，サービス事業者が提供する Web アプリとして実装されたネット動画プレーヤーである．図では Web アプリとして実装されるネット動画プレーヤーの多くが共通に有する機能として，典型的な実装形態における再生動作に関わる機能部分を図示している．video 要素は，W3C の HTML5 仕様[11]で規定される機能であり，動画再生動作処理を行う．また，video 要素は，ユーザ操作等による状態変化に応じてイベントを出力する機能を有している．このイベントに関する仕様も HTML5 で規定されている．再生制御 UI（User Interface）はユーザからのプレーヤー操作を受け付ける機能を持つ．

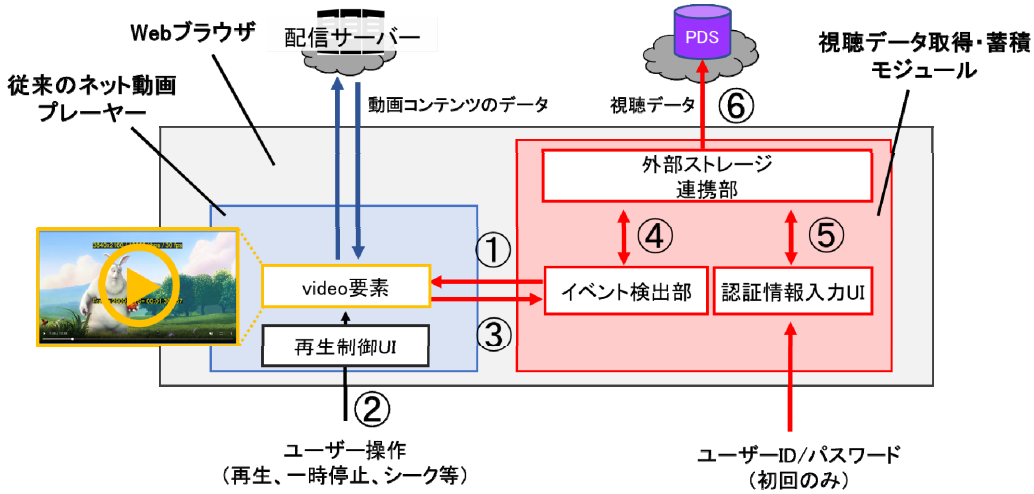

図 2　提案する取得・蓄積モジュールの構成

図 2 の右側の赤枠で示す部分が提案する取得・蓄積モジュールであり， 3 つの機能部から構成され，前述した青枠部分のサービス事業者が提供するモジュールと連携して動作する．それぞれの役割は以下の通りである．

A) イベント検出部
video 要素が出力するイベントを検出し, そのイベントデータを外部ストレージ連携部に転送する．

B) 外部ストレージ連携部
PDS と認証情報や視聴データの送受信を行う．また，PDS への認証作業の頻度を適正化するために，認証状況の管理や認証情報入力 UI の表示制御を行う．

C) 認証情報入力 UI
PDS の URL や，PDS の認証に必要なユーザ ID やパスワードを入力するインターフェースを提供する．

取得･蓄積モジュールの動作シーケンスは次の通りである．

① ネット動画プレーヤーの起動時に，イベント検出部は連携するネット動画プレーヤーの video 要素にイベント監視処理を登録する．イベント監視処理とは，設定したイベントが発生したときに，そのイベントデータをイベント検出部に転送する処理である．

② ユーザは，ネット動画プレーヤーが提供する再生制御 UI を通じて，ネット動画プレーヤーの操作を行う．その操作に応じて，ネット動画プレーヤーは動画再生などの処理を行う．

③ video 要素は, ①で登録した監視処理がイベントの発生に応じてイベントデータを出力し，イベント検出部に転送する．

④ イベント検出部は，③で受け取ったイベントデータを外部ストレージ連携部に転送する．

⑤ 外部ストレージ連携部は，PDS のプライベート領域にアクセスするための認証トークンを保持しているか確認する．認証トークンを保持していない場合は認証情報入力 UI をユーザに提示し，PDS の URL およびユーザ ID やパスワード等の認証に必要なデータの入力をユーザに要求する．その入力内容をもとに外部ストレージ連携部は PDS の認証を行う. 認証が成功した場合は，外部ストレージ連携部は以降のアクセスに必要な認証トークンを保持しておく．

⑥ 外部ストレージ連携部は，④で受信したデータを視聴データとして PDS のプライベート領域に保存する．このときプライベート領域にアクセスするための認証トークンは⑤で取得したものを用いる．

本手法では, video 要素が出力するイベントの送信先をネット動画配信サービス事業者の仕組みとは独立する取得・蓄積モジュールに設定する．さらに，その取得したイベントデータを取得・蓄積モジュール内で処理を行い，PDS に蓄積する. したがって, 要求条件 1 を満たすことができる．さらに, video 要素はユーザ操作に応じた状態変化を出力できるため，ユーザの意図の導出につながる詳細な視聴データを取得できる．このことより，要求条件 2 を見たすことができる．また，3.2 で述べたような様々なネット動画視聴環境がある中で，Web アプリとして実装されるネット動画プレーヤーに着目した理由は，Web ブラウザは PC やスマホなどの各種デバイス向けに提供されているからである．そのため，本手法は複数のデバイスと実装形態の組み合わせに適用可能であり，適用範囲が広い．以上より要求条件 3 を満たすことができる．

5. 視聴データ取得･蓄積モジュールの試作と検証

5.1 視聴データ取得・蓄積モジュールの試作と動作検証

4 章で提案した手法の動作検証のために，Web アプリの記述言語である JavaScript を用いた視聴データ取得・蓄積モジュールを試作した．取得・蓄積モジュールが視聴データを蓄積する PDS には, オープンソースプロジェクトである Solid[12] で開発が進められている Personal Online Datastore（POD）を用いた．POD は Web 技術を中心に実装されているため，今回試作した取得・蓄積モジュールとのインターフェースを結合させやすい．さらに，提案手法の実装可能性を検証するために，Web アプリとして実装されたネット動画プレーヤーのサンプルアプリと，試作した取得･蓄積モジュールを組み込みこんだ検証用 Web アプリを作成した．

この検証用 Web アプリを HTML5 に対応する表 1 に示す Web ブラウザ上で動作させた．動画コンテンツや配信方式の条件も表 1 に記す. 図 3 に検証用 Web アプリでの動画再生に応じて，POD に視聴データが書き込まれる様子を示す.

表 1 取得・蓄積モジュールを組み込んだ検証用 Web アプリの動作条件

Web ブラウザ	Google Chrome [13] Firefox [14]
動画コンテンツ	Big Buck Bunny [15]
配信方式	MPEG-DASH

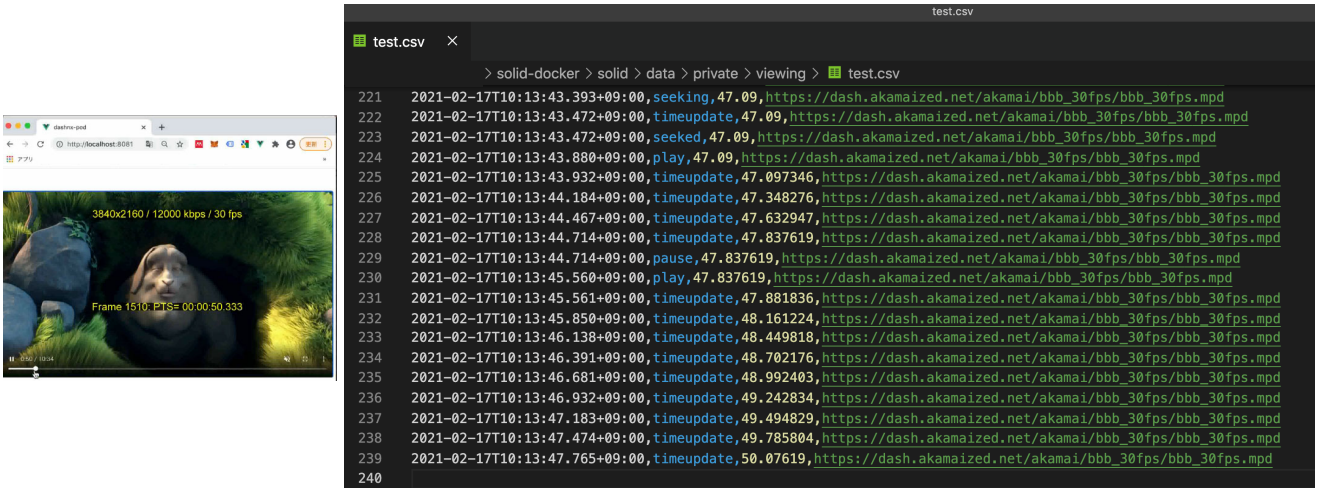

図 3 ネット動画プレーヤーのサンプルアプリ（左）と POD に書き込まれる視聴データの様子（右）

今回試作したモジュールのイベント検出部が取得する video 要素のイベント種別を表 2 に示す. きめ細かな視聴データを取得するために，標準化されている video 要素のイベント種別の中からユーザ操作と関連が強いものを選択した．これらのイベントが発生するごとに視聴データを取得し，POD に蓄積させた．POD に蓄積する視聴データの項目を表 3 に示す．これらは，図 3 の右図が示す各 1 行のテキストデータが表す項目に対応する．

表 2 取得・蓄積モジュールで取得する
video 要素のイベント種別

イベント名	イベント発生条件
play	動画の再生が開始した
pause	動画の再生が一時停止した
seeking	動画のシーク操作が開始した
seeked	動画のシーク操作が完了した
ended	動画の末尾に達し，再生が停止した
timeupdate	動画の再生位置が更新された

表 3　POD に蓄積する視聴データの項目

項目	例
取得時刻	2020-12-10T10:15:00.000+09:00
イベント名	play
動画再生位置（秒）	15.0
動画ファイル URL	https://sample.com/test.mpd

5.2 結果と考察

動作検証の結果，表 2 に示した全てのイベントの発生に応じて，試作したモジュールからユーザの視聴データを POD に蓄積できることを確認した．これにより，4.3 で提案した手法が HTML5 対応ブラウザで実装可能であることを確認した．

一方，この手法ではイベントごとに視聴データが生成されるため，ネット動画プレーヤーへの操作が多い場合や，視聴時間が長くなると，POD 内の視聴データ量が増大しやすいことが明らかになった．データ量削減のため，視聴データの記述方法や記録方法の検討が必要である．また今後は，事業者サービスに視聴データの利用を許可する際に，許可する視聴データの範囲をユーザが容易に判断可能にするために，POD 内部の視聴データの可視化方法や検索方法の検討も必要になると考えられる．

今回，取得・蓄積モジュールと Web アプリとして実装されたネット動画プレーヤーのサンプルアプリを一つの検証用 Web アプリに組み込んだ．しかし，既存のネット動画配信サービスのネット動画プレーヤーのプログラムは事業者が管理しているため，Web ブラウザ上で動作しているネット動画プレーヤーと独自のモジュールを，ユーザが一つの Web アプリに統合することは難しい．今後，Web ブラウザを介して，事業者が提供するネット動画配信プレーヤーと取得・蓄積モジュールを連携させる方法などを検討する．

また，今回取得した視聴データが表すコンテンツ情報はネット動画の URL であり，タイトルなどのコンテンツ情報を表現できていない．視聴データを様々なサービスへ活用することを想定すると，POD 内の視聴データとコンテンツ情報を表すメタデータを関連付ける方法を検討する必要もある．

6. おわりに

本稿では，ユーザが PDS を用いて主体的にパーソナルデータを管理・活用する非集中型モデルにおいて，Web アプリとして実装されたネット動画プレーヤーの視聴データを蓄積する手法を提案した．この手法では，Web アプリとして実装されたネット動画プレーヤー内の HTML の video 要素と連携する視聴データ取得・蓄積モジュールを用いて，各ユーザが管理する PDS に視聴データを蓄積する．また，提案手法の実装可能性を検証するために，Web アプリの記述言語である JavaScript を用いて，視聴データ取得・蓄積モジュールを試作した．試作したモジュールとネット動画プレーヤーのサンプルアプリを組み合わせた検証用 Web

アプリを作成して動作確認を行った結果， HTML5 対応ブラウザ上で提案した視聴データ蓄積手法が実装可能であることを確認した．

今後，5 章で述べた提案手法の課題解決のほか，活用先のサービスから放送およびネットの視聴データを統合的に扱えるようにするために，各取得手法の改良に加え，視聴データの記述方式の統一化などに取り組んでいく．

参考文献

[1] 橋田浩一: 分散 PDS と情報銀行：集めないビッグデータによる生活と産業の全体最適化, 情報管理, vol. 60, no.4, pp. 251-260 (2017).
[2] Taguchi, S. Yamamura, C. Ohmata, H. Sekine, D. Kajita, K. and Fujii, A.: Sharing Same Elements in User Viewing History Data Securely Through Private Set Intersection Under User-centric Data Control, *Proceedings of the 2020 ACM International Conference on Interactive Media Experiences* (*IMX'20*), pp.138-142 (2020).
[3] 山村千草, 大亦寿之, 田口周平, 関根大輔, 藤沢寛, 藤井亜里砂: 家庭環境下でのスマホ連携によるテレビ視聴関連行動データの収集と分析, マルチメディア，分散，協調とモバイルシンポジウム論文集, vol. 2020, no.1, 1H-4, pp.155-160 (2020).
[4] System Architectures for Personalization and Recommendation, https://netflixtechblog.com/system-architectures-for-personalization-and-recommendation-e081aa94b5d8 （参照 2021-02-17）.
[5] Seo, J. Lim, H. Oh, C. Yun, C. Suh, B and Lee, J.: A System Designed to Collect Users' TV-Watching Data Using a Smart TV, Smartphones, and Smart Watches, *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video* (*TVX'16*), pp.147-153 (2016).
[6] Kim, M. Kim, J. Han, S. and Lee, J.: A Data-driven Approach to Explore Television Viewing in the Household Environment, *Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video* (*TVX'18*), pp.89-100 (2018).
[7] YouTube, https://www.youtube.com/, （参照 2021-02-17）.
[8] Netflix, https://www.netflix.com/jp/, （参照 2021-02-17）.
[9] TVer, https://tver.jp/, （参照 2021-02-17）.
[10] NHK プラス, https://plus.nhk.jp/, （参照 2021-02-17）.
[11] HTML 5.2, https://www.w3.org/TR/html52/, （参照 2021-02-17）.
[12] Mansour, E. Sambra, V. A. Hawke, S. Zereba, M. Capadisli, S. Ghanem, A. Aboulnaga, A. and Berners-Lee, T.: A Demonstration of the Solid Platform for Social Web Applications, *Proceedings of the 25th International Conference Companion on World Wide Web* (*WWW'16 Companion*), pp.223-226 (2016).
[13] Google Chrome, https://www.google.com/intl/ja_jp/chrome/, （参照 2021-02-17）.
[14] Firefox, https://www.mozilla.org/en-US/firefox/, （参照 2021-02-17）.
[15] Big Buck Bunny, https://peach.blender.org/, （参照 2021-02-17）.