Vol.2018-MBL-88 No.12
Vol.2018-CDS-23 No.12
2018/8/31

IoT デバイスを活用した放送サービスについての検討

小川展夢[†1]　大亦寿之[†1]　池尾誠哉[†1]　藤井亜里砂[†1]　藤沢寛[†1]

概要：近年，様々な IoT デバイスが開発され，普及に向けた取り組みが行われている．筆者らは，IoT デバイスを放送サービスに活用する仕組みについて検討している．IoT デバイスを活用することで，視聴者の状況や情報の特性に応じて，適切なタイミング・手段で，映像に限らず様々な形式で，情報を提示することが可能になる．しかし，多くの IoT デバイスは，直接放送コンテンツを受信することはできない．そこで本稿では，テレビやスマートフォンなどの端末を介して，IoT デバイスを放送コンテンツと連携して制御するためのシステムアーキテクチャを設計した．また，ユースケースを試作し，システムの性能を評価した．

キーワード：放送通信連携，Hybridcast，Internet of Things (IoT)，Web of Things (WoT)

System Architecture for Broadcasting Service with IoT Devices

HIROMU OGAWA[†1]　TOSHIYUKI OHMATA[†1]　MASAYA IKEO[†1]
ARISA FUJII[†1]　HIROSHI FUJISAWA[†1]

Abstract: Many types of IoT devices have been developed recently. Herein, we consider providing broadcast contents in a new way using diverse IoT devices. We propose a system architecture based on Hybridcast and Web of Things to communicate the metadata of broadcast content among a TV and IoT devices and Control Description to control various IoT devices. We prototyped a 4D movie service and we evaluated our system performance.

Keywords: Broadcast-Broadband System, Hybridcast, Internet of Things (IoT), Web of Things (WoT)

1. はじめに

これまで放送局は，放送コンテンツを製作して配信し，家庭のテレビでユーザーに視聴してもらうことを主要なサービスとしてきた．一方，近年スマートフォンやコミュニケーションロボット，スマート家電，BLE ビーコンなど，多様な IoT デバイスの研究・開発が進められている．筆者らは，これら IoT デバイスを活用することで，放送サービスを，ユーザーのテレビに放送コンテンツを配信することに留まらず，ユーザーに新たな体験を提供するものに拡張できると考えている[1][2]．例えば，動画に IoT デバイスの効果を加えて新たな手法で情報を表現することや，ユーザーがテレビを視聴していないタイミングでもこれまでの視聴行動に関連した情報を提示することなどが可能になる．

関連研究として，Ariyasu ら[3]は，振動するデバイスを動画のシーンの内容に合わせて制御することで，視覚・聴覚に触覚情報を加えた映像表現を行うことについて検討している．また，川上[4]らは，ユーザーとロボットが一緒にテレビを見るという新しい視聴体験を提案している．

これらの研究は，IoT デバイスを活用することで，新たな放送サービスが実現できる可能性を示した．一方，新たな体験の提案に重点が置かれており，現行の放送の仕組みの中で，これらのサービスをどの様に実現するかという点については，十分に検討していない．IoT デバイスを活用して多様なサービスを実現するには，機能やインターフェースの異なる様々な IoT デバイスを自由に活用する仕組みが必要である．また，放送以外の事業者も利用できる仕組みであることが求められる．

そこで筆者らは，放送以外の事業者も含めた様々な事業者が，多様な IoT デバイスを活用して放送サービスを実現するための汎用的なシステムアーキテクチャを設計した．また，サービスを試作し，性能や課題について検証した．

以下，2 章では，IoT デバイスを活用した放送サービスの実現に向けた課題について述べる．3 章では，システムによって実現するユースケースと，そのためのシステム要件を整理する．4 章では，放送通信連携システム Hybridcast と，IoT の仕様の 1 つである Web of Things (WoT)をベースとした，システムのアーキテクチャを提案する．また，IoT デバイスの制御情報の記述方法である Control Description を提案する．5 章では，提案に基づいて試作したサービスについて述べる．6 章では，試作サービスに基づき，提案の特徴や課題などについての考察を述べる．7 章で，今後の展望について述べ，本稿をまとめる．

2. サービス実現に向けた課題

Ariyasu ら[3]や川上ら[4]は，IoT デバイスを活用した放送サービスを提案した．これらの提案は IoT デバイスを活用することで放送サービスが新たな体験を提供できるように

†1 日本放送協会
NHK (Japan Broadcast Corporation)

Vol.2018-MBL-88 No.12
Vol.2018-CDS-23 No.12
2018/8/31

なる可能性を示した．しかし，現行の放送システムの上で実際にサービスを開発するには，検討すべき課題がある．

1 点目は，多様な IoT デバイスを活用する仕組みについてである．先行研究では，試験的サービスのために特定の IoT デバイスが用いられた．一方，実際にサービスを行うには，スマート家電や BLE ビーコンなど数多ある IoT デバイスの中から，サービス開発者が目的に合ったものを選択し，簡単に活用できることが望ましい．しかし，それぞれの IoT デバイスは，異なるプロトコルやインターフェースを持ち，実装方法が異なる．同じ機能の IoT デバイスでも実装方法が異なる場合もある．サービス開発者が，サービスの対象となり得る IoT デバイスと，それぞれの実装方法をすべて把握し開発を行うということは，現実的に困難である．従って，サービス開発者が，サービスに使用する IoT デバイスを自由に選択し，統一された抽象的な方法で制御する仕組みが必要である．

2 点目は，放送局以外の事業者がサービスを開発する仕組みについてである．今後，様々な機能を持つ IoT デバイスが爆発的に増加することが予想される[5]．それらの能力を活かした魅力あるサービスをユーザーに提供するには，放送事業者だけでサービスを開発するのではなく，様々な事業者が開発に参加できることが望ましい．放送コンテンツの制作者でない放送以外の事業者が放送コンテンツを活用したサービスを開発するには，ある放送コンテンツやコンテンツ中のシーンが，何に表しいているのかという意味的な情報を知る必要がある．しかし現状では，放送以外の事業者が利用可能な放送コンテンツに関する情報は，EPG（電子番組ガイド）の情報等に限定される．従って，放送以外の事業者が放送コンテンツの意味的な情報を取得し利用するための仕組みが必要である．

3. システム要件の整理

2 章で挙げた課題を解決可能な放送サービスシステムの要件をまとめる．まず提案システムにより実現したいユースケースを整理し，続いてシステム要件を整理する．

3.1 システムにより実現するサービスのユースケース

筆者らは，IoT デバイスを活用した放送サービスのユースケースを複数検討し，放送コンテンツの視聴との時間的な関係を基に，以下の 2 グループに分類した．

（1）同期型：主に放送コンテンツの再生に同期して IoT デバイスを制御して実現するサービス．例えば，番組の内容に合わせてロボットが喋ったりスマート LED ライトが光ったりする高臨場感放送サービスなど．

（2）非同期型：放送コンテンツの視聴中以外のタイミングに提供するサービス．例えば，テレビでグルメ番組を観た後，自動車でドライブ中に，走路の近くに関連したお店があることをカーナビがお知らせしてくれるサービスなど．

3.2 システムが満たすべき要件

要件 1：放送コンテンツを起点に IoT デバイスを制御できること

ユーザーの放送コンテンツの視聴をきっかけとして IoT デバイスを同期または非同期に制御する必要がある．多くの場合，IoT デバイスは放送波を直接受信できないため，別の手段で IoT デバイスに視聴に関連した情報を通知する必要がある．

要件 2：テレビがない場所でも放送コンテンツの情報を活用できること

非同期型サービスでは，サービスを提供するシーンが放送コンテンツを視聴している間でなく，場所もテレビの前ではない場合がある．このような状況においても，視聴履歴など放送コンテンツやその視聴に関する情報をサービスが利用できる必要がある．

要件 3：IoT デバイスを簡単に制御できること

IoT デバイスには様々な機能や規格のものが存在するが，サービス開発者にとっては，デバイスごとの実装方法の差異をなるべく意識せずに開発できることが望ましい．

要件 4：サービスが必要な IoT デバイスを探索し制御できること

サービス開発者が，放送コンテンツを視聴するユーザー個々の IoT デバイス環境を事前に知ることは困難である．そのため，サービスが必要な IoT デバイスをその場で探索し，マッチング，接続，制御する仕組みが必要である．

要件 5：放送コンテンツの意味的な内容を放送局以外の事業者のサービスが解釈可能であること

放送コンテンツの制作者以外の多様な事業者がサービスを開発するために，放送コンテンツの意味的な内容をサービスが解釈可能である必要がある．例えば，シーンの内容に合わせて IoT デバイスを制御する同期型サービスであれば，あるシーンにおいて誰が何をしているのか，という情報などである．また，視聴した番組に応じてクーポンを発行する非同期型サービスであれば，ユーザーが何の番組を観たのか，という情報などである．

4. IoT デバイスを活用した放送サービスシステムのアーキテクチャと Control Description

3 章で示した要件を満たす, IoT デバイスを活用した放送サービスのためのシステムアーキテクチャを提案する．また，IoT デバイス制御のためのデータ記述である Control Description を提案する．

4.1 IoT デバイスを活用した放送サービスの全体概要

図 1 に，提案する放送サービスシステムの全体概要を示す．受信機を除く IoT デバイスの多くは，放送コンテンツを直接受信する機能を持たない．そこで，放送コンテンツを受信する受信機と連携することで，間接的に放送コンテンツとの連携を実現する構成とする．システムは，放送コ

ンテンツを受信・再生する受信機，受信機と直接やり取りする IoT デバイス（以降，直接連携デバイスと呼称する），直接連携デバイスを介して間接的に受信機とやり取りする IoT デバイス（以降，間接連携デバイスと呼称する）で構成される．また，IoT デバイスを制御するための情報である Control Description (CD)と，間接連携デバイスの機能などの情報を表す Thing Description (TD)[6]を利用する．

受信機と直接連携デバイスの連携には，Hybridcast[7]の機能を利用する．Hybridcast とは，欧州の HbbTV[8]等と同様，放送通信連携プラットフォームの 1 つである．Hybridcast 対応受信機は web ブラウザを持ち, HTML5 アプリを実行できる．また HTML5 アプリは，放送波で放送コンテンツと一緒に送られてくるイベントメッセージ(以下，EM）を取得し，EM の内容に応じた処理を実行できる．一方，直接連携端末は，Hybridcast モジュールを組み込んだアプリをインストールすることで，Hybridcast 対応受信機と連携できる．具体的には，同一 LAN 内に存在する受信機を探索，接続し，受信機上で動作する HTML5 アプリとメッセージをやり取りできる. また, 直接連携デバイスは, Control Description (CD)を参照することで, 放送コンテンツを起点に IoT デバイスを制御するための情報を知ることができる.

直接連携デバイスと間接連携デバイスの連携には，Web of Things (WoT)[9]のアーキテクチャを適用する．ここで，WoT について説明する；WoT とは，W3C が標準化を進める IoT 規格である．IoT の分野では，IoT デバイスを利用するための様々な標準化団体が，規格やフレームワークの作成に取り組んでいる. 例えば, 主にコンシューマ向けの IoT デバイスに関する標準化団体である Open Connectivity Foundation (OCF)[10]，モバイル通信関連企業が加盟する Open Mobile Alliance (OMA)[11]，Home Energy Management System (HEMS）の規格である ECHONET[12]等がある．それぞれは，規格に基づいた IoT デバイスの相互運用を容易にするためのプロトコル，インターフェースなどを規定している．一方，規格を跨いだ IoT デバイスの相互運用は，一般に困難であるという課題がある．WoT は，この課題の解決を目指した規格である．WoT は，既にインターネットの分野で広く利用されている web の技術を用いて，様々な規格に準拠した IoT デバイスを統一されたフォーマットで制御するためのインターフェースを提供する．WoT デバイスは，自らの機能や API へのアクセス方法などを，既定のフォーマットに従って記述する．この機能が記述されたものを Thing Description (TD)と呼ぶ[6]．提案システムにおける間接連携デバイスは WoT デバイスである. 直接連携デバイスは，間接連携デバイスの TD を参照することで，間接連携デバイスの機能や制御方法を知り，統一されたインターフェースを用いて制御することができる．

提案システムを構成する各要素は，放送事業者，アプリ事業者，デバイス事業者の 3 者によって提供されるものとする．放送事業者は，放送コンテンツ，受信機上で動作する HTML5 アプリ， IoT デバイスを制御するための情報である CD を提供する．アプリ事業者は，直接連携端末上で動作するデバイスアプリを提供する．デバイス事業者は，間接連携デバイスと，その特徴を示す TD を提供する．アプリ事業者は, CD と TD を参照することで, 放送コンテンツと IoT デバイスを活用したサービスを開発することが可能となる．なお，3 つの事業者の内，いずれか 2 つ，または 3 つ全てが同一であっても構わない．

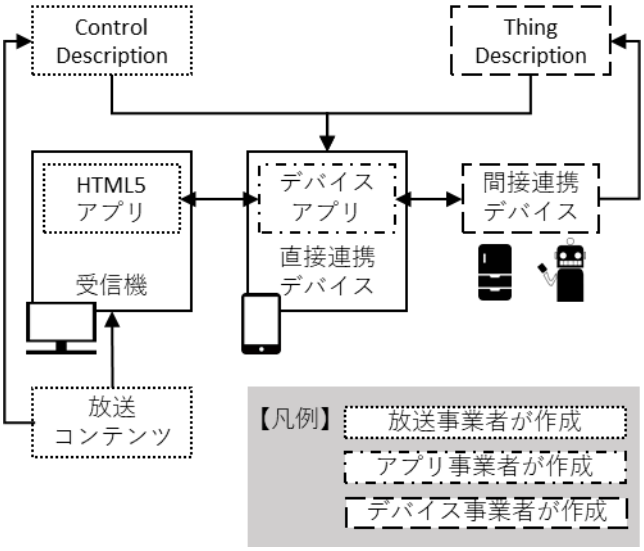

図 1　放送コンテンツと IoT デバイス連携の概念図
Figure 1　Conceptual Diagram.

4.2 提案システムのアーキテクチャ

図 2 に，提案システムのアーキテクチャを示す．以下，それぞれのモジュールについて説明する．

受信機は，受信機機能，ブラウザ，HTML5 アプリから構成される．受信機機能は，放送コンテンツの受信，映像の再生，通信など受信機としての様々な機能を担う．ブラウザは，アプリサーバーから HTML5 アプリを取得し，実行する．HTML5 アプリは，直接連携デバイス上で動作するデバイスアプリとの接続時や EM の受信時など，何らかのイベントをトリガーとして，デバイスアプリにテキストメッセージを送る．このメッセージは，デバイスアプリが放送コンテンツに関連した CD を CD サーバーから取得するための情報（CD の取得先 URL など）を含んでいる．以下，このメッセージを CD 指定メッセージと呼称する．

間接連携デバイスは，HTTP や CoAP などのプロトコルにより外部からアクセス可能な API を持ち，直接連携デバイスからのリクエストを受け付ける．リクエストを受けると，それに基づき処理を実行する．また，ある間接連携デバイスが持つ機能やその機能に対する制御リクエストの方法などの情報は，デバイス事業者が TD として記述し，TD

サーバーに登録する．TD サーバーは，間接連携デバイスの外部（クラウド上など）に存在しても，間接連携デバイス自身がその機能を担ってもよい．直接連携デバイスは，TD サーバーから TD を取得し，間接連携デバイスとの連携手段を知ることができる．

直接連携デバイスは，デバイスアプリ，通信機能から構成される．デバイスアプリは，受信機や間接連携デバイスとの連携を司る．デバイスアプリは，CD 取得機能，コンテンツデバイス連携機能，デバイス管理機能を含む．CD 取得機能は，受信機上で動作する HTML5 アプリから受信した CD 指定メッセージを基に，CD サーバーから CD を取得する．デバイス管理機能は，ネットワーク内に存在する間接連携デバイスを探索する．また，それぞれの TD を取得し，間接連携デバイスを管理する．コンテンツデバイス連携機能は，取得した CD と TD を基に，制御対象として適当な間接連携デバイスと機能を選択し，制御リクエストを送る．通信機能は，制御リクエストをそれぞれのデバイスの規格に準じたプロトコルに変換する Protocol Binding 機能を含む．

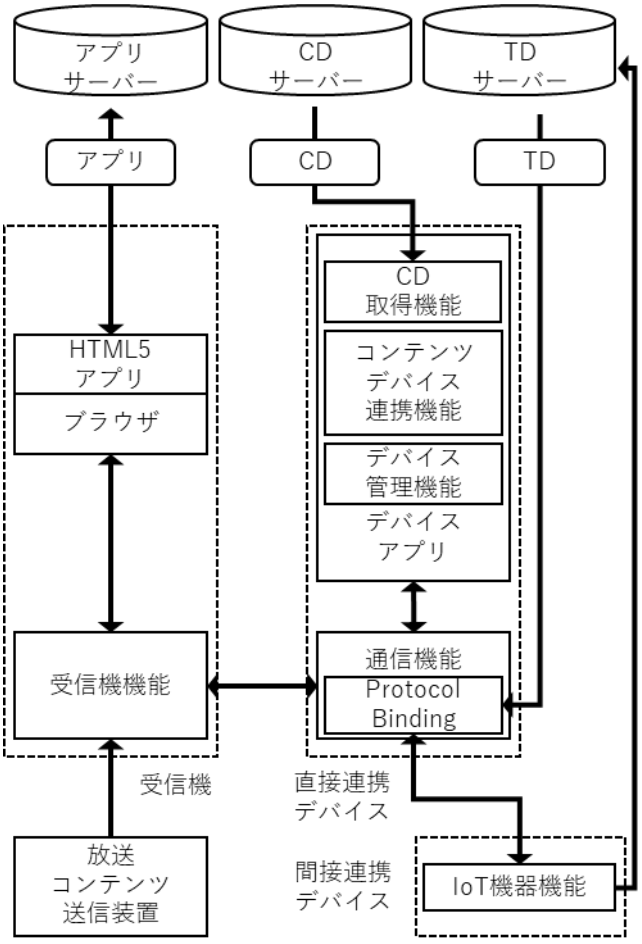

図 2　システムのアーキテクチャ
Figure 2　System Architecture.

4.3 Control Description による IoT デバイスの制御

Control Description (CD)は，放送コンテンツを起点に直接連携デバイスや間接連携デバイスを制御するために利用される構造化データである．番組毎に紐づけられる．図 3 に，Control Description のデータモデルを示す．

CD オブジェクトは，複数の Scene オブジェクトを持つ．これは，番組が連続する複数のシーンから成ることに対応している．各 Scene オブジェクトは，間接連携デバイスを制御するための情報である Device Control Description (DCD)オブジェクトと，デバイスアプリがシーンの内容を意味的に理解するための情報である Scene Description (SD)オブジェクトを持つ．DCD オブジェクトは，間接連携デバイスの機能を呼び出すための Interaction Pattern (IP)オブジェクトを持つ．IP クラスは，間接連携デバイスの状態変数を制御するための Property クラスと，動的な機能を制御するための Action クラスをサブクラスに持つ．また IP オブジェクトは，間接連携デバイスの機能呼び出し時に用いる制御値を表す Data Schema (DS)オブジェクトを持つ．

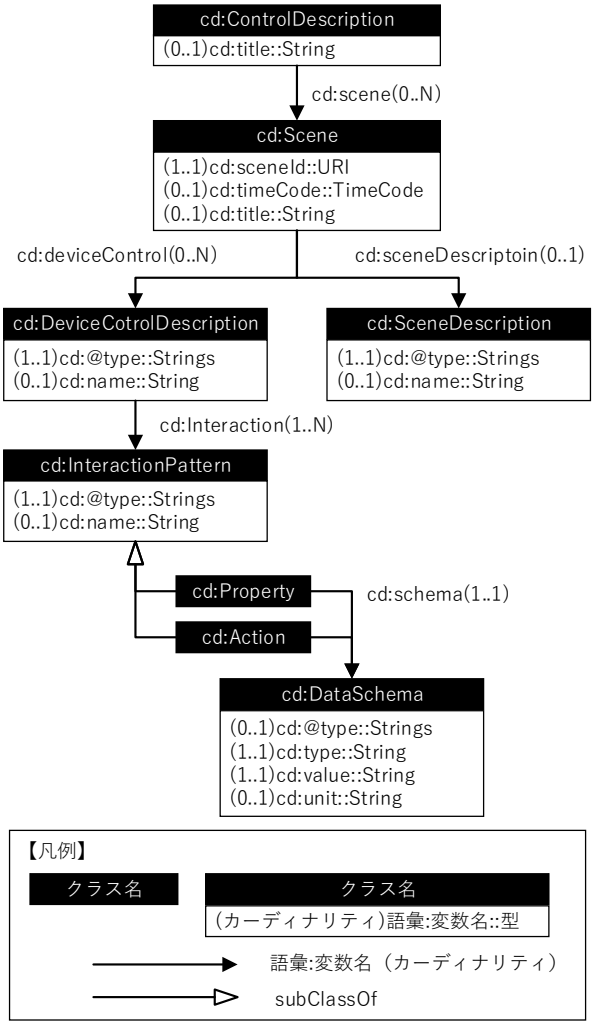

図 3　Control Description のデータモデル
Figure 3　Data Model of Control Description

各クラスの持つ変数の詳細を表 1 に示す.

クラス名	変数名	変数の説明	変数の型
Control Description	title	オブジェクトの名前。このオブジェクトが紐づく番組のタイトルや識別子など。	Stirng
Scene	sceneId	シーンの識別子。	URI
	timeCode	シーンの番組内におけるタイムコード。	TimeCode
	title	シーンの名前。	String
Device Control Description	@type	制御対象とするデバイスの種類。	String配列
	name	制御対象とするデバイスの名称。	String
Interaction Pattern	@type	制御対象とするインタラクションの種類。Propertyかactionのいずれか1つを必ず含む。	String配列
	name	制御対象とするインタラクションの名前。	String
Data Schema	@type	値の種類。	String
	type	値の型。	String
	value	値。	String
	unit	値の単位。	String
Scene Description	@type	表現するシーンの種類。	String
	name	表現するシーンの名前。	String

表 1　各クラスの変数の詳細
Table 1　Class's Detail

Device Control Description (DCD)

DCD は, 間接連携デバイスを制御するための情報である. DCD のデータ構造は, WoT の TD[6]のデータ構造に対応している．図 4 は，JSON-LD[13]で記述された LED ライトの TD の記述例を示している．7 行目の”@type”の値は，このデバイスがスイッチの ON/OFF と明るさに関する制御が可能な照明デバイスであることを示している．また，11 行目以降に，明るさの値の読み書きの可否，制御リクエストの送信先エンドポイントの URI などが示されている. 同様に, 27 行目以降に，スイッチの ON/OFF 制御に関する情報が記載されている．図 5 は，LED ライトの明るさとスイッチを制御するときの DCD を含む CD の記述例を示している. 16 行目以降が，DCD に関する記述である．21 行目の”@type”と 23 行目の”name”の値により，制御対象とする間接連携デバイスを指定する．また，24 行名以降の”interaction”は，対象のデバイスの”Property”や”Action”に対する制御内容を指定する．直接連携デバイス上で動作するデバイスアプリは，CD と TD を比較し，制御対象の間接連携デバイスを決定する．また制御リクエストの内容を決定し，TD からリクエストのエンドポイントを特定する．

```
1  {
2     "@context": [
3        "https://w3c.github.io/wot/w3c-wot-td-
4  context.jsonld",
5        {"iot": "http://iotschema.org/"}
6     ],
7     "@type": ["Thing", "iot:Light",
8  "iot:BinarySwitch"],
9     "name": "LED light",
10    "interaction": [
11       {
12          "@type": ["Property", "iot:brightness"],
13          "name": "brightness",
14          "schema": {
15             "type": "integer",
16             "minimum": 0,
17             "maximum": 100
18          },
19          "writable": true,
20          "observable": true,
21          "form": [{
22             "href":
23  "coaps://mylamp.example.com:5683/status",
24             "mediaType": "application/json"
25          }]
26       },
27       {
28          "@type": ["Action", "iot:Toggle"],
29          "name": "toggle",
30          "form": [{
31             "href":
32  "coaps://mylamp.example.com:5683/toggle",
33             "mediaType": "application/json"
34          }]
35       }
36    ]
37 }
```

ThingDescription.json

図 4　Thing Description の JSON-LD による記述例
Figure 4　Example of Thing Description in JSON-LD

Scene Description (SD)

SD は，直接連携デバイスを制御する際に用いる情報である．直接連携デバイスは，アプリ開発者が開発したデバイスアプリの記述次第で，処理内容を柔軟に決定できる．そこで，放送事業者からはシーンの意味的な情報を表す構造化データである SD を提供し，それを受けて直接連携デバイスがどの様に動作するかについては，デバイスアプリの記述に委ねるものとする．図 5 は，サッカーの試合において，チーム kashima の ogawa 選手がゴールを決めた際の SD を含む CD の記述例である．9 行目の”@type”の値は，このシーンの出来事の種類を表している．また，11 行目以降の”description”の値は，このシーンの出来事を具体的に表

す構造化データである．この構造は，”@type”の値毎に規定される．”@type”の値毎の”description”が取るべきデータ構造については，今後の検討課題とする．

```
{
   "title": "NHK Soccer:",
   "scene": [{
      “sceneId”: “nhk/G/sport/soccer/2018-08-06-1800/abcdef",
      "timeCode": “xx:xx:xx",
      "title": "first goal made by team kashima.",
      "sceneDescription": {
               "@type": "soccerGame:goal",
               "title": "kashima's goal",
               "description": {
                  "player": "ogawa“,
                  …
               }
      },
      “deviceControlDescription":{
         "@context": [
            "https://w3c.github.io/wot/w3c-wot-td-context.jsonld",
            {"iot": "http://iotschema.org/"}
         "@type": ["Thing","iot:Light","iot:BynarySwitch"],
         "name": "LED light",
         "interaction": [{
            "@type": ["Property","brightness"],
            "schema": {
               "type": "Integer",
               "value": 100
            }
         },{
            "@type": ["Action","Toggle"],
            "schema": {
               "@type": ["onoff"],
               "type": "boolean",
               "value": "true"
            }
         }]
      }
   }]
}
```

ControlDescription.json

図 5　Control Description の JSON LD による記述例
Figure 5　Example of Control Description in JSON-LD

4.4 放送 IoT デバイス連携のシーケンス例

放送コンテンツを起点に IoT デバイスを活用したサービスを実行するまでのシーケンスは，サービスの内容によって様々である．ここでは 3.1 節に示した「高臨場感放送サービス」を例として，受信機が放送コンテンツを受信・再生中に，EM の受信をトリガーとして，直接連携デバイスを介して間接連携デバイスに制御リクエストを送る場合のシーケンスを述べる．前提として，直接連携デバイス上で動作するデバイスアプリが，同一ネットワーク内に存在する受信機と間接連携デバイスの状態を把握しているものとする．始めに，放送信号送出装置が受信機に対して放送コンテンツを送信する．放送コンテンツには Application Information Table (AIT)が含まれる．AIT は，受信機が実行する HTML5 アプリの取得先 URL 等の情報が記述されている．受信機は AIT の情報を基に，アプリケーションサーバーから HTML5 アプリを取得・実行する．続いて，放送信号送出装置は，受信機に EM を送る．受信機上の HTML5 アプリは，EM の受信をきっかけに，直接連携端末上で動作するデバイスアプリに CD 指定メッセージを送信する．デバイスアプリは CD 指定メッセージの情報を基に，CD サーバーから CD を取得する．デバイスアプリは，CD の SD の記述に基づき，直接連携デバイスの動作を決定する．また，CD の DCD の記述に基づき，適当な間接連携デバイスを選択し，制御リクエストを送る．間接連携デバイスは，デバイスアプリからのリクエストに基づき動作する．

5. 同期型サービスの試作

4 章で提案したアーキテクチャに基づきサービスを試作し，システムが正常に動作することを確認した．以下，試作サービスについて述べる．

5.1 試作サービスの概要

本稿では，2.1 節で示したユースケースの内，同期型サービスに分類される「高臨場感放送サービス」を試作した．図 6 にイメージを示す．サッカーの番組進行に合わせて，小型ロボット（Sota，ヴィストン株式会社製）やスマート LED ライト（HUE，PHILIPS 社製）が動作して，番組の雰囲気を演出するサービスである．

5.2 IoTivity フレームワークを用いたシステムの実装

図 7 に，試作したサービスのシステム図を示す．受信機は，Hybridcast 対応機であり，放送コンテンツを受信・再生する．スマートフォンは，受信機と連携する直接連携デバイスとして機能する．Raspberry Pi は，スマートフォンと連携する間接連携デバイスとして機能する．小型ロボットとスマート LED ライトは Raspberry Pi からの HTTP による制御リクエストを受けて動作し，間接連携デバイスである Raspberry Pi の機能の実行部として機能する．

受信機と直接連携デバイスの連携は，Hybridcast モジュールを組み込んだデバイスアプリを開発し，Hybridcast の機能により実現した．また，直接連携デバイスと間接連携デバイスの連携は，デバイスアプリと Raspberry Pi それぞれに，IoTivity[14]フレームワークのモジュールを組み込んで利用することで実現した．IoTivity は，WoT がインターフェースの共通化の対象の 1 つとする OCF[10]の仕様に基づくフレームワークである．規格に準拠したデバイス同士の機器発見，相互接続などの機能を提供する．WoT インターフェースから Iotivity の通信プロトコルへと処理を変換する Protocol Binding の機能を果たすモジュールは自作し

た．

本試作サービスは，4.4 節で示したシーケンスに従って実行した．即ち，番組中のイベントシーン（試合開始シーン，得点シーンなど）毎に受信機に EM を送信し，それをきっかけに HTML5 アプリからデバイスアプリ，デバイスアプリから間接連携デバイスへとメッセージを送信して，間接連携デバイスを動作させた．

図 6　試作サービス
Figure 6　Trial Service.

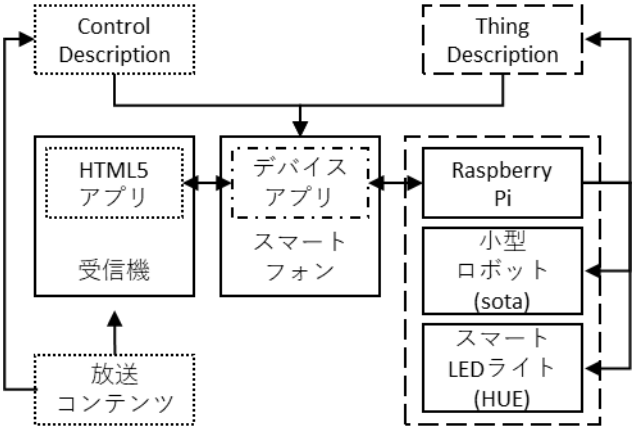

図 7　試作サービスの構成
Figure 7　System of Trial Service.

6. 考察

6.1 Scene Description が必要な場合

試作サービスでは，Device Control Description (DCD)により間接連携デバイスを制御した．デバイスアプリは DCD と Thing Description (TD)を比較し，適当な間接連携デバイスをマッチングし，制御リクエストを送信する．これらの機能を共通モジュール化して，デバイスアプリがこのモジュールを利用できるようにすれば，デバイスアプリは簡単に間接連携デバイスを制御できるようになる．一方，DCD は放送事業者がサービスの意図に応じて記述するが，デバイスアプリの開発者がユーザーの特性等に応じて独自のサービスを追加したい場合は，Scene Description の情報が必要になる．例えば，ユーザーの好みの選手が得点した際に特別な演出を加える，というサービスを行う場合，どのチームのどの選手が得点した，という情報を Scene Description から取得する必要がある．

6.2 今後の課題

今後の課題として，間接連携デバイスの制御の優先順位付けが挙げられる．以下に列挙する状況では，1 つの間接連携デバイスに複数の制御リクエストが重複して送られる可能性があるため，予め制御の優先順位を決めておくなど，何らかの工夫が必要である．（1）ある間接連携デバイスにリクエスト可能な直接連携デバイスが，同一ネットワーク内に複数ある場合（2）ある間接連携デバイスに対する制御情報が，Device Control Description と直接連携デバイス上で実行するアプリの両方にある場合．

また，Scene Description の記述方法についても検討していく必要がある．番組のジャンル毎に考えると，サッカーや野球などのスポーツ番組は，番組中に生じるイベントやその際に記述すべき事柄が比較的予想・限定しやすいと考えられる．従って，まずはこのジャンルの番組から検討していきたい．

7. まとめ

本稿では，放送以外の事業者を含む様々な事業者が，様々な IoT デバイスを活用した放送サービスを実現するためのシステムについて検討した．そして，Hybridcast と WoT をベースとしたシステムアーキテクチャと，IoT デバイスの制御情報である Control Description を提案した．また，同期型サービスの代表的な例である高臨場感放送サービスを試作し，システムの特徴や今後の課題について検討した．

今後は，他の同期型サービスや非同期型のサービスについても実装し，検証を進めていく．また，サービスのユーザー評価も行っていく．さらに，放送局が検討する番組のインターネット同時配信等の動向も踏まえ，放送コンテンツに限定せず通信による配信も含めたコンテンツ起点の IoT デバイス連携アーキテクチャへの拡張についても検討していく．

謝辞　研究内容についてご助言・ご協力いただいた皆様に，謹んで感謝の意を表する．

参考文献

[1]小川展夢，池尾誠哉，大亦寿之，藤沢寛．放送コンテンツを基点とした IoT 機器連携動作のためのアーキテクチャの検討．情報処理学会第 79 回全国大会. 2017, 7C-02
[2]小川展夢，大亦寿之，山村千草，藤井亜里砂，藤沢寛. IoT 機器の使用による放送局のデータ・コンテンツの活用機会拡大に向けた検討．情報処理学会第 80 回全国大会. 2018, 7D-04
[3]K. Ariyasu, H. Kawakita, T. Handa, H. Kaneko. Tactile sensibility presentaiton service for Smart TV, Proc. IEEE 3rd Global Conference on Consumer Electronics. 2014, p.236-237
[4]川上晧平，村上洋平．テレビと連携する IoT の一検討．第 53 回民放技術報告会予稿集．2016, p.146-147

[5]"総務省 平成 29 年版情報通信白書". http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/html/nc133100.html, （参照 2018-08-06）
[6]"Web of Things (WoT) Thing Description". https://www.w3.org/TR/wot-thing-description/, （参照 2018-08-06）
[7]"IPTV フォーラム： ハイブリッドキャスト". http://www.iptvforum.jp/hybridcast/, （参照 2018-08-06）
[8]"HbbTV". https://www.hbbtv.org/, （参照 2018-08-06）
[9]"W3C: Web of Things at W3C". https://www.w3.org/WoT/, （参照 2018-08-06）
[10]"OPEN CONNECTIVITY FOUNDATION". https://openconnectivity.org/, （参照 2018-08-06）
[11]"oma SpecWorks". https://www.omaspecworks.org/, （参照 2018-08-06）
[12]"ECHONET". https://echonet.jp/, （参照 2018-08-06）
[13]"JSON for Linking Data". https://json-ld.org/, （参照 2018-08-06）
[14]"Iotivity". https://iotivity.org/, （参照 2018-08-06）