

# セマンティックデータにより記述されるデバイスを活用した 通知提示アプリケーション

阿部晋矢<sup>1</sup> 藤井翔子<sup>1</sup> 佐藤辰哉<sup>1</sup> 小松佑人<sup>1</sup> 藤津智<sup>1</sup> 藤沢寛<sup>1</sup>

**概要:** IoT デバイスの普及や、テレビのインターネット接続など、メディアを提示できるスマートデバイス、メディアの種類は増加しつつある。一方で、あるコンテンツを提示する際の、適切なデバイスの選択に関する検討はほとんどされていない。そこで我々はセマンティックデータを活用した Web ベースのデバイス選択手法を検討している。提案手法はコンテンツとして通知を例とし、ユーザーを中心としたコンテンツ、デバイスを力学モデルにより 2 次元空間へ配置する。そして、配置結果から通知を提示するためにユーザー近傍のデバイスを選択する。本稿では、デバイス選択の結果をグラフィカルに表示し、実際に選択したデバイスに通知を提示するアプリケーションの試作結果を紹介する。実験結果より、通知提示アプリケーションの動作を確認した。

**キーワード:** Web TV, セマンティック Web, Internet of Things, Web of Things

## Notification Application with Semantic description about Devices

SHINYA ABE<sup>†1</sup> SHOKO FUJII<sup>†1</sup> TATSUYA SATO<sup>†1</sup> YUTO KOMATSU<sup>†1</sup>  
SATOSHI FUJITSU<sup>†1</sup> HIROSHI FUJISAWA<sup>†1</sup>

**Abstract:** With the proliferation of the Internet of Things and smart devices, content presentation devices and content types are constantly increasing. By contrast, there is little consideration on selecting the appropriate devices for presenting a particular content. Therefore, we are considering about a web-based device selection method based on semantic data from the resource description framework. In our proposed method, we use notifications as examples of content, and a network graph representing users, content, and devices is arranged in a two-dimensional space by our user-centric force-directed algorithm. By selecting device nodes near the user node from the placement results, devices are determined to present content. In this study, we introduce a prototype application that graphically displays the result of device selection and presents a notification to the selected devices. We verified its operation in experiments.

**Keywords:** Web TV, Semantic Web, Internet of Things, Web of Things

### 1. はじめに

近年、家庭におけるスマートスピーカーなどをはじめ、冷蔵庫や電子レンジ、電球など様々なデバイスの IoT 化が進んでいる。従来、放送受信が中心であったテレビもインターネットへ接続し、Integrated broadcasting-broadband (IBB)システムにより複数の端末が連携してコンテンツを提示することもある。端末を連携させてのコンテンツ提示やスマートデバイス普及を背景に、我々は IoT-based Media Framework というコンセプトを提案した[1, 2]。IoT-based Media Framework では、デバイスなどの環境、ユーザーコンテキスト、コンテンツ情報を活用し、ユーザーにコンテンツ体験を提供する。IoT-based Media Framework の課題の 1 つに、コンテンツ提示デバイスの適切な選択があげられる。現状は、例えば家庭にあるデバイスの種類や数は、我々が把握できる範囲であるため、我々はコンテンツを視聴するデバイスを手動で選択している。しかし、今後予想できない種類、数のデバイスでコンテンツを提示する場合、手

動でのデバイス選択は難しく、自動的なデバイス選択が求められる。つまり、あるコンテンツを提示する際に、ユーザーコンテキストとデバイスの環境から、適切なデバイスを自動的に選択することが求められる。

様々なスマートデバイスが増え、我々は日常的に複数のデバイスを利用しているが、コンテンツを提示するデバイスの適切な選択方法についてはほとんど検討されていない。確認できる既存手法として、ユーザーが持つ TV や PC、スマートフォン、タブレット PC の「ディスプレイサイズ」と事前に定めた「レイアウト」の候補の組合せ最適化[3]がある。この手法は特に組合せ最適化の効率化に焦点を当て、その効率化は他の問題にも適用可能である。しかし、ディスプレイ機能のないデバイス、例えばスマートスピーカーは対象とされていない。また、今後どのような種類のデバイスが普及するか予想できない。そのため、事前に組合せ最適化の候補やコスト関数などを用意することは難しい。コンテンツを提示するデバイス選択においては、より柔軟な応用範囲の広い手法が求められる。

そこで我々は情報提示へ向けた Web ベースかつ力学モデルによるデバイス選択手法を提案した[4]。提案手法はセマンティック Web[5]の思想をベースにデバイス同士やコ

<sup>1</sup> 日本放送協会 放送技術研究所  
NHK Science & Technology Research Laboratories

コンテンツ，ユーザーの関係を Resource Description Framework(RDF) [6]により管理する，つまり，それらの関係性は RDF のトリプルにより有向グラフにより表される．適切なデバイスを選択するため，RDF グラフを整理し，デバイス，ユーザー，コンテンツの関係を空間上にプロットする．提案手法によるコンテンツの提示例として「通知」を取り上げ，ユーザーを中心に，通知コンテンツ，デバイスを Fruchterman-Reingold アルゴリズム[7]により 2 次元空間へプロットし，ユーザー近傍のデバイスを選択する．本稿では，提案手法によるデバイス選択の結果をグラフィカルに表示し，実際に選択したデバイスに通知を提示するアプリケーションの試作結果を紹介する．実験結果より，通知提示アプリケーションの動作を確認した．

## 2. 関連研究

インターネットの普及に伴い，放送と通信の連携へ向けた IBB システムの標準規格が提案されてきた．加えて，スマートフォンなどのスマートデバイスの増加に伴い，TV とスマートデバイスを連携させる端末連携が提案，標準化されてきた．日本の Hybridcast[8]，欧州の HbbTV 2.0[9]，米国の ATSC 3.0[10]が端末連携の標準の例として挙げられる．これらの端末連携のしくみを発展させ，より人間中心にコンテンツを提示するコンセプトとして，我々は IoT-based Media Framework [1]を提案した．IoT-based Media Framework では，hypertext transfer protocol (HTTP) や WebSocket などの Web 技術をベースにおく W3C Web of Things [11]を意識している．W3C Web of Things で提唱される Servient として受信機を扱う，Common receiver architecture [2]も我々は提案した．我々は普段，コンテンツを視聴するためにリビングで TV を利用したり，ベッドでスマートフォンを利用したり，適切なデバイスを手動で選択している．今後，世の中に，また部屋の中にスマートデバイスが増えた場合，状況に合わせたデバイスを手動で選択するのは現実的でない．我々の知る唯一のデバイス選択手法として，ユーザーが持つ TV や PC，スマートフォン，タブレット PC の「ディスプレイサイズ」と事前に定めた「レイアウト」の候補からコンテンツを提示するデバイスを最適化する手法が提案された[3]．マルチデバイス適応[3]は特に組合せ最適化の効率化に焦点を当て，その効率化は他の問題にも適用可能である．しかし，ディスプレイ機能のないコンテンツ提示デバイスは増えており，今後もスマート化された家電製品など我々の想定しないデバイスが増え続けると予想できる．そういったデバイスへ向けたコンテンツのレイアウトの予想は難しく，コスト関数を事前に設定することも難しい．そのため，コンテンツを提示するデバイス選択においては，より柔軟な応用範囲の広い手法が求められる．

セマンティック Web[5]は，Web ページに代表される情報

をコンピュータが理解し利用するため，情報同士の関係を意味のあるつながりにより表現する仕組みである．Web ページのつながりを継続的に表すため拡張性高く，機械可読なメタデータを記述可能なデータモデルである RDF[6]などの技術仕様が標準化されている．セマンティック Web の考え方は Web ページに留まらず浸透し続けている．W3C Web of Things でも RDF を意識した Thing Description でデバイスのデータを表している．W3C Web of Things では，RDF による Thing Description 記述のため，Internet of Things を意識したオントロジーを整理している．一方，現状 W3C Web of Things ではコンテンツを扱う例は少なく，提示デバイスの自動選択は扱っていない．Web ページをはじめ，近年は TV 番組もコンテンツとして，RDF によりコンテンツのメタデータを表す試みが進んでいる[12, 13]．商業的な理由としては検索エンジンで検索結果の上位となるため，schema.org[14]などのオントロジーを用いてデータを RDF で表すことが多い．このように，デバイスやコンテンツを RDF とオントロジーにより表すことは増えているが，RDF データからの情報抽出は発展途上である．RDF データからの情報抽出の代表例は SPARQL[15]によるデータ抽出である．SPARQL によるデータ抽出を効率化するオントロジーの提案も多いが，一方で必ずしも柔軟性の高い検索，情報抽出はできない．RDF データを対象とした柔軟性の高い情報抽出へ向け，RDF データをベクトル化する RDF2vec[16]が提案された．RDF2vec は Web ページの関係性などをベクトル化し，ベクトルから情報抽出を行う．大規模な RDF データ向けに検証を行っている一方，家庭向けのスマートデバイスについてコンテンツ提示候補を選択する際に利用できるか分からない．

これらの状況から，我々はコンテンツ，デバイスが RDF データとして表されることを前提に，より利用しやすいベクトル化を行い，柔軟性高くデバイスを選択することを提案した[4]．本稿では，提案手法によるデバイス選択結果をグラフィカルに提示し，実際に選択デバイスに通知を提示するアプリケーションの試作結果を紹介する．

## 3. 情報提示へ向けた Web ベースおよび力学モデルによるデバイス選択

提案手法は，デバイス，コンテンツ，ユーザーの関係を RDF により表す．RDF ではデータを主語，述語，目的語のトリプルで表し，その構造は有向グラフで表される．関係の表されたグラフを空間へプロットすることで，関係性をベクトル化する．関係の表されたグラフの元の構造が複雑で直接空間へプロットできない場合，プロット可能な形へ抽出および加工する．抽出および加工したグラフを空間へプロットしたことで，ベクトル化された関係をもとにコンテンツを提示するデバイスを選択する．元の RDF グラフに十分にデバイスやユーザー，コンテンツの関係が含まれて

いれば、各要素の関係を反映したベクトルが得られるため、例えば距離の近いデバイスを選択することでユーザーの状況に合ったデバイスを選択できる可能性がある。

本稿では、もっとも単純なコンテンツの例として、警報や速報ニュースなどの受信通知を出すことを目的にデバイスを選択する手法を紹介する。デバイスによって受信内容を表示する等も可能だが、単に通知を受信したことのみ提示可能なデバイスもある。そこで本稿では、単に通知を受信したことを通知することを対象とし、内容によらない通知自体を通知コンテンツとする。図 1 に 2 次元空間へ提案手法を適用し、通知コンテンツを提示するデバイスを選択するアルゴリズムを示す。提案手法はグラフ変換、空間配置、機器選択の 3 つのフェーズからなる。グラフ変換では、元の RDF グラフの構造が複雑で直接空間へプロットできないため、プロット可能な形へ抽出および加工する。空間配置では、抽出および加工したグラフを Fruchterman-Reingold アルゴリズムで 2 次元空間へ配置しベクトル化する。機器選択では、2 次元にベクトル化されたユーザー、デバイス、コンテンツから、ユーザーを中心に近傍のデバイスを選択することで、通知コンテンツを提示するデバイスを選択する。以降は、図 2 の通知コンテンツ、デバイス、ユーザーの状況を例にアルゴリズムの各手順について紹介する。ユーザーはリビングルームと寝室のある空間におり、現在はリビングルームにいる。デバイスは TV、IoT エアコン、スマートフォン、各部屋に IoT 電球がある。通知コンテンツは #AAA とし、通知コンテンツを放送やネットワークから直接受信可能なデバイスは、TV、IoT エアコン、スマートフォンである。各部屋の IoT 電球はスマートフォンを経由して、通知を受信し、提示可能である。通知コンテンツ #AAA は温度、音、光に対する伝わりやすさである ServiceTypeParam をメタデータとして持ち、順に 0, 1, 1 の値であるとする。上記状況を表した RDF データの Turtle 記述を図 3 に示す。

### 3.1 Graph transformation

図 3 で表される RDF データは構造が複雑で、そのままでは 2 次元空間へ配置できない。グラフ変換では、入力 of RDF データを配置可能なよう整理する。グラフ変換は 3 つの処理からなる。

まず、コンテンツを利用可能(available\_service)なデバイスを抽出する。ここでは、#tv, #smartphone, #air\_conditioner が抽出される。抽出されたデバイスはコンテンツと重みが 1 のエッジで接続する。SPARQL クエリを図 4 に示す。

続いて、ユーザーとデバイスの関係を抽出する。ここでは、ユーザーと同じ場所(schema:location)であるデバイスについて、serviceType からユーザーとの関係を重み付きエッジで表す。結果として、#air\_conditioner は重み 0, #tv は重み 2, #iot-light は重み 1 のエッジでユーザーと接続される。SPARQL クエリを図 5 に示す。

最後にユーザーとの間にエッジがあるが、コンテンツとの間にエッジがないデバイスについて、今回は #iot-light1 の関係を抽出する。ここでは #iot-light1 を制御可能な (hub\_connect), コンテンツを利用可能なデバイスである #smartphone が抽出される。その後、デバイス同士をユーザーとの間の重み、コンテンツとの重みの合計の重みで接続する。今回は  $2(=1+1)$  の重みで、#iot-light1 と #smartphone を接続する。SPARQL クエリを図 6 に示す。

最終的な抽出、加工されたグラフを図 7 に示す。元の RDF データ(図 3)は、ユーザーとコンテンツ、デバイスのみに整理され、エッジは関係性を表す重みをもつ。

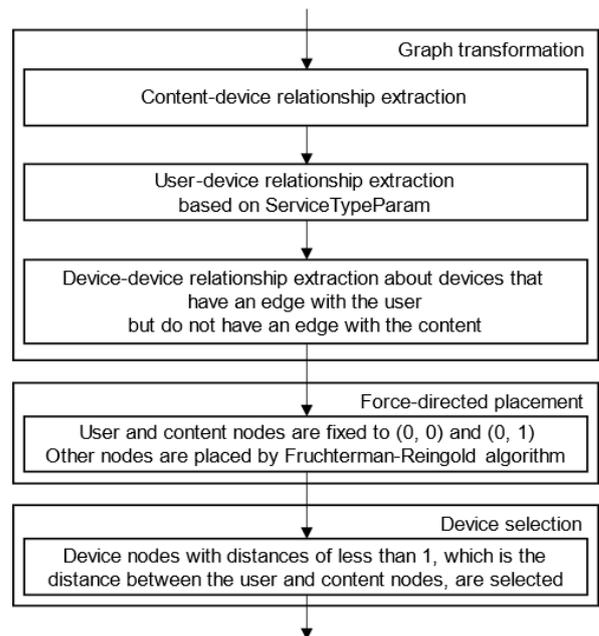


図 1 2 次元空間へのプロットによる Web ベースおよび力学モデルによるデバイス選択アルゴリズム

Figure 1 Web-based force-directed adaptive device selection algorithm in two-dimensional space.

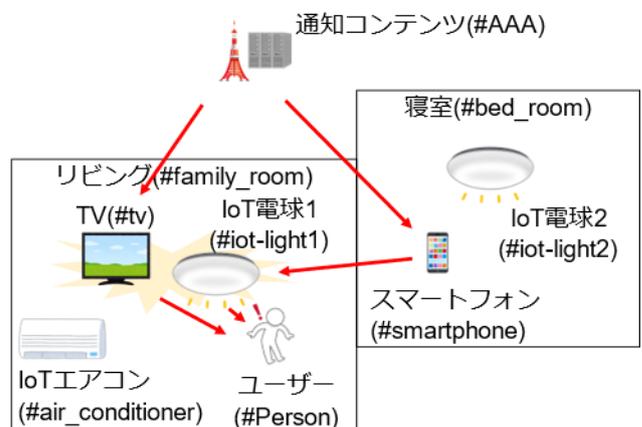


図 2 デバイスとコンテンツ、ユーザーの関係の例  
 Figure 2 Example of device, content, and user status.

```
@base <http://example.org/> .
@prefix schema: <https://schema.org/> .
<#AAA>
  schema:name "AAA" ;
  a schema:Organization .
<#Person>
  schema:name "person" ;
  schema:location <#family_room> ;
  a schema:Person .
<#tv>
  schema:location <#family_room> ;
  schema:serviceType <light> ;
  schema:serviceType <sound> ;
  <available_service> <#AAA> ;
  a schema:Thing .
<#smartphone>
  schema:location <#bed_room> ;
  schema:serviceType <light> ;
  schema:serviceType <sound> ;
  <hub_connect> <#iot-light1> ;
  <hub_connect> <#iot-light2> ;
  <available_service> <#AAA> ;
  a schema:Thing .
<#iot-light1>
  schema:location <#family_room> ;
  schema:serviceType <light> ;
  a schema:Thing .
<#iot-light2>
  schema:location <#bed_room> ;
  schema:serviceType <light> ;
  a schema:Thing .
<#airconditioner>
  schema:location <#family_room> ;
  schema:serviceType <temperature> ;
  <available_service> <#AAA> ;
  a schema:Thing .
<light> schema:value 1 .
<sound> schema:value 1 .
<temperature> schema:value 0 .
```

図 3 入力となる Turtle によるデバイス、コンテンツ、ユーザーを表す RDF データ

Figure 3 Turtle-based input RDF data of the example situation.

```
SELECT ?service_device
WHERE {
  ?service_device
  <http://example.org/available_service>
  <http://example.org/#AAA> .
}
```

図 4 コンテンツとデバイスの関係を抽出する SPARQL クエリ

Figure 4 SPARQL query that extracts the relationship between content and devices.

```
SELECT ?person_device ?type
WHERE {
  <http://example.org/#Person>
  schema:location ?location .
  ?person_device schema:location ?location .
  ?person_device schema:serviceType ?type .
}
```

図 5 ユーザーとデバイス間の関係からデバイスと ServiceType を抽出する SPARQL クエリ

Figure 5 SPARQL query to extract devices and ServiceTypes from user-device relationships.

```
SELECT ?device_device
WHERE {
  ?device_device
  <http://example.org/hub_connect> <device URI> .
  ?device_device
  <http://example.org/available_service>
  <http://example.org/#AAA> .
}
```

図 6 中継となるデバイスを抽出する SPARQL クエリ

Figure 6 SPARQL query to extract relaying devices.

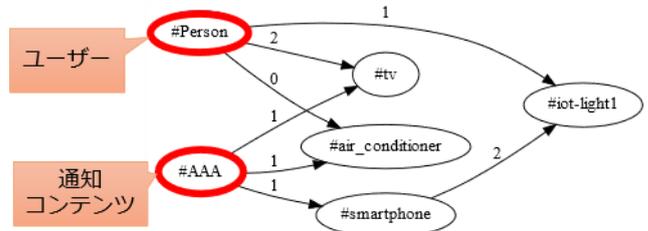


図 7 Graph transformation により得られるグラフ

Figure 7 Graph output by graph transformation.

### 3.2 Force-directed placement

前のグラフ変換の処理により、配置可能なようにグラフは整理された。そこで空間配置では、グラフ変換で得られたグラフをもとに、2次元空間に配置する。配置には、ネットワークグラフの可視化で一般的に利用される Fruchterman-Reingold アルゴリズムを用いる。まず、ユーザーを原点(0, 0)に配置する。次に、コンテンツを(0, 1)に配置する。そして、デバイスについてはランダムな配置とし、ユーザーとコンテンツの位置を固定して Fruchterman-Reingold アルゴリズムにより配置する。図 7 のグラフをもとに配置した結果を図 8 に示す。ユーザー、コンテンツ、デバイスの関係をもとに、通知を伝えるうえでより良いと考えられるデバイスは重みのあるエッジでコンテンツやユーザーとつながっているため、その関係を反映した配置結果となっている。なお、図 8 には次の処理の Device Selection の処理で用いる選択範囲も描画している。

### 3.3 Device selection

空間配置によりベクトル化された結果をもとにコンテンツを提示するデバイスを選択する。本稿では、配置結果をもとにユーザーから距離が 1 以内のデバイスを選択する。図 8 のユーザーを中心とした半径 1 の円に含まれる #tv, #smartphone, #iot-light1 が選択される。

結果として、図 2 の通り TV から直接、およびスマートフォンを経由し IoT ライトから通知が提示される。一方で、情報を伝えにくい IoT エアコンは選択されず、ユーザーの状況に適応したデバイスが選択できる。

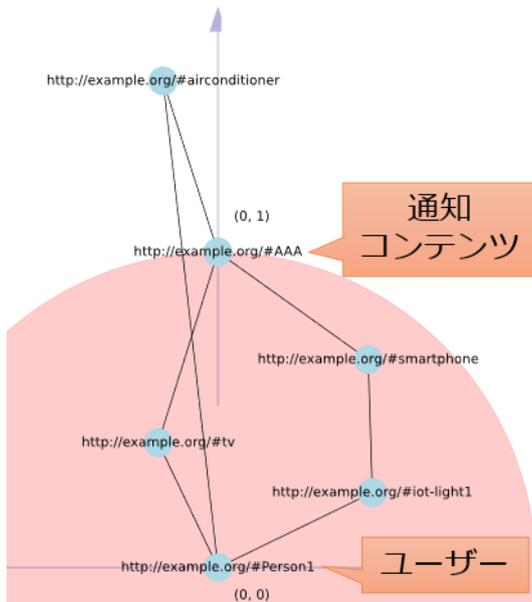


図 8 Force-directed placement と Device selection の結果  
 Figure 8 Results of force-directed placement and device selection.

#### 4. 通知提示アプリケーションの試作検証

提案手法によるデバイス選択結果の検証へ向け、通知提示アプリケーションを試作した。通知提示アプリケーションは以下の2種類から構成される。

- ・通知受信アプリケーション: 各デバイス上で動作するアプリケーション。REST APIによる通知コンテンツを受信し、デバイス自身が通知を提示する場合、画面上にメッセージを表示する。自身が経由デバイスで、次に通知を伝える先がある場合、相手のデバイスに対応した REST APIによるメッセージを送信する。
- ・選択結果提示アプリケーション: 家庭内のいずれかのデバイスもしくはクラウド上で動作するアプリケーション。デバイスの選択状況をグラフィカルに表示し、選択結果を変更できる。また、通知提示実行を押下することで、通知受信アプリケーションへ REST APIによるメッセージを送信する。

図 2 の状況に対応するデバイスと各アプリケーションの構成を図 9 に示す。選択結果提示アプリケーションはクラウド上を想定し、図 2 には登場しない PC 上に用意することとした。IoT 電球 1, IoT 電球 2 は直接通知受信アプリケーションをインストールできないため、スマートフォンを介して通知を提示することとした。各デバイスは同一の LAN (Local Area Network) に接続しており、またインターネットと接続しインターネットサービスも受信できる。

通知受信アプリケーションは Android OS 上で直接動作するネイティブアプリとして実装し、Android TV™ と Android スマートフォンへインストールした。IoT エアコン

については所望のエアコンを用意できなかったため、別の Android スマートフォンをエアコンに見立てることとした。通知受信アプリケーションは HTTP による REST API を用いて通知を受信、もしくは経由して IoT 電球へ通知内容を送信する。通知内容の宛先が自身である場合、画面上に通知を受信したことを提示する。一方、経由デバイスとして IoT 電球へ通知を送信する場合、IoT 電球を点滅させる内容の POST メッセージを IoT 電球へ送信する。

選択結果提示アプリケーションは python により Web アプリケーションとしてソフトウェア実装した。フレームワークとして Flask[17]を用い、ネットワークグラフの描画には Cytoscape.js[18]を用いた。選択結果を提示した画面を図 10 に示す。提案手法の配置結果を受信し、グラフィカルに選択結果を表示可能である。また、選択結果を確認した後、各デバイスにあたるノードを移動させることで、選択結果を変更することも可能である。通知提示実行ボタンを押下すると選択範囲である円の中に含まれるノードに対応するデバイスについて通知を提示する。以上について試作、動作確認した全体システムの概観写真を図 11 に示す。

#### 5. おわりに

我々はセマンティック Web の思想をベースに RDF データからデバイスを選択する Web ベースおよび力学モデルによるデバイス選択を提案した。本稿では、提案手法のデバイス選択結果をグラフィカルに表示し、選択デバイスをグラフィカルに変更可能で、最終的に選択デバイスへ通知を提示するアプリケーションを試作。動作を確認した。

今後の課題として、例として用いた状況だけでなく、より多くの状況・環境での検証があげられる。また、試作アプリケーションにより可能になった、デバイス選択結果をグラフィカルに表示することや、選択結果の正しさについて、ユーザー評価実験によって検証することがあげられる。

#### 参考文献

- [1] Endo, H., Sato, T., Ogawa, H., Abe, S., Fujitsu, S., Matsumura, K. and Fujisawa, H.: IoT-based Media Framework for Public Service Media: Expansion of Current Digital Broadcasting Systems(2021).
- [2] Abe, S., Takiguchi, T., Endo, H., Ikeo, M. and Fujisawa, H.: Web-Based Architecture for Broadcast and Broadband Content with Device Collaboration in a Local Network(2021).
- [3] Domínguez, A., Flórez, J., Lafuente, A., Masneri, S., Tamayo, I. and Zorrilla, M.: A Model for User Interface Adaptation of Multi-Device Media Services, IEEE Transactions on Broadcasting, pp.1-13(2021).
- [4] 阿部晋矢, 佐藤辰哉, 佐藤壮一, 小松佑人, 藤沢寛, 藤津智: 情報提示へ向けた Web ベースおよび力学モデルによるデバイス選択手法, 情報科学技術フォーラム講演論文集(2021).
- [5] Berners-Lee, T., Hendler, J. and Lassila, O.: The semantic web, Scientific american, Vol.284, No.5, pp.34-43(2001).
- [6] RDF 1.1 Primer, <https://www.w3.org/TR/rdf11-primer/>, (参照 2022-08-01).
- [7] Fruchterman, T. M. and Reingold, E. M.: Graph drawing by force - directed placement, Software: Practice and experience, Vol.21, No.11, pp.1129-1164(1991).

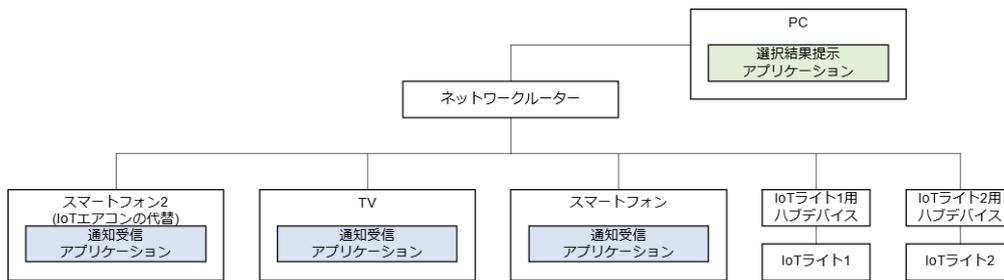


図 9 デバイスとアプリケーションの構成

Figure 9 Device and application configuration diagram.

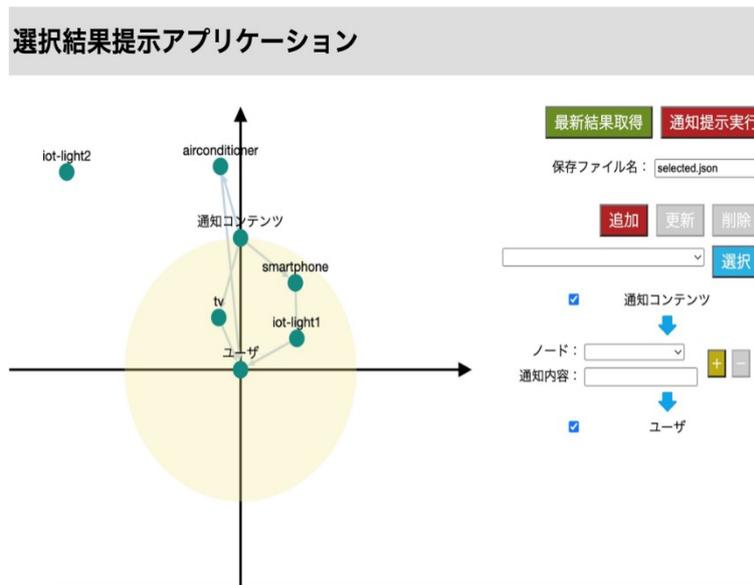


図 10 選択結果提示アプリケーション画面

Figure 10 Screen of notification and device selection application.

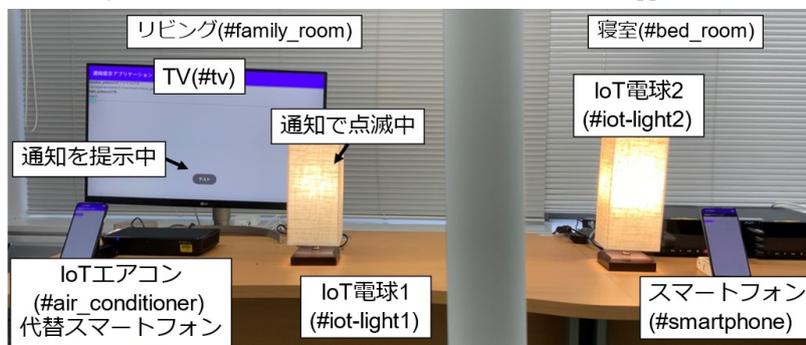


図 11 試作システム概観

Figure 11 Overview of the prototype system.

[8] IPTVFJ: Integrated Broadcast-Broadband System Specification, IPTVFJ STD-0010 Version 2.2(2018).

[9] ETSI and EBU: Hybrid Broadcast Broadband TV, European Telecommunications Standards Institute, European Broadcasting Union(2018).

[10] ATSC: ATSC Standard: ATSC 3.0 Interactive Content, Advanced Television Systems Committee, Washington, DC(2020).

[11] W3C Web of things, <https://www.w3.org/WoT/>, (Accessed 2020-08-31).

[12] Urakawa, M., Miyazaki, M., Yamada, I., Fujisawa, H. and Nakagawa, T.: A study about integrating video contents with web services based on the RDF, International Journal of Space-Based and Situated Computing, Vol.6, No.2, pp.65-73(2016).

[13] Urakawa, M. and Fujisawa, H.: Data Structuring for Launching Web Services Triggered by Media Content(2017).

[14] schema.org, <https://schema.org/>, (参照 2022-08-01).

[15] SPARQL 1.1 Query Language, <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/>, (参照 2022-08-01).

[16] Ristoski, P., Rosati, J., Di Noia, T., De Leone, R. and Paulheim, H.: RDF2Vec: RDF graph embeddings and their applications, Semantic Web, Vol.10, No.4, pp.721-752(2019).

[17] flask, <https://flask.palletsprojects.com/>, (参照 2022-08-01).

[18] Franz, M., Lopes, C. T., Huck, G., Dong, Y., Sumer, O. and Bader, G. D.: Cytoscape.js: a graph theory library for visualisation and analysis, Bioinformatics, Vol.32, No.2, pp.309-311(2015).