

Title	Social thingsにおける名前付けに関する研究
Sub Title	A study on naming for social things
Author	今木, 彩翔(Imaki, Ayaka) 砂原, 秀樹(Sunahara, Hideki)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2014
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2014年度メディアデザイン学 第381号
Genre	Thesis or Dissertation
URL	<a href="https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002014-0381">https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002014-0381</a>

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

修士論文 2014年度（平成26年度）

Social Things における  
名前付けに関する研究

慶應義塾大学大学院  
メディアデザイン研究科

今木 彩翔

本論文は慶應義塾大学大学院メディアデザイン研究科に  
修士(メディアデザイン学)授与の要件として提出した修士論文である。

今木 彩翔

審査委員：

砂原 秀樹 教授 (主指導教員)

稲蔭 正彦 教授 (副指導教員)

岸 博幸 教授 (副指導教員)

修士論文 2014年度（平成26年度）

## Social Things における 名前付けに関する研究

カテゴリー：サイエンス / エンジニアリング

### 論文要旨

家庭用電化製品、自動車などの性能向上やネットワーク機器の小型化などにより、インターネットに様々な「モノ」が接続されるようになる Internet of Things の時代が到来している。「モノ」と人という単一の繋がりだけではなく、「モノ」と「モノ」の連携により、より高度なサービスの実現が試みられている。例えば、Social Things では、照明とテレビが連携することにより、テレビのコンテンツに合わせて照明の明るさなどが変化するということが考えられる。

「モノ」と「モノ」が通信する上で重要なことは、「通信相手を識別すること」である。現状は、各ベンダのルールに基づいた識別子が付与されているため、そのドメインを超えた通信は困難である。そこで、統一的な名前空間で一意的であり、ユーザの可読を補助するシステム側の識別子を提案する。しかしながら、一意な識別子であると、ユーザには解読出来ない複雑な識別子にせざるを得ないため、ユーザフレンドリーな識別子として、ユーザが指示する名前の検討、システムが識別するための識別子との対応付けの検討を行う。そして、システム側の識別子に対するユーザの可読性に関する基礎的な実験を行い、有用性を確認した。

キーワード：

名前解決, コンテキスト, Internet of Thing

慶應義塾大学大学院 メディアデザイン研究科

今木 彩翔

Abstract of Master's Thesis of Academic Year 2014

## A Study on Naming for Social Things

Category: Science / Engineering

### Summary

Due to the rapid evolution of miniaturization of network equipment, home appliances and cars, the age of the Internet of Things(IoT) is fast approaching. A wide variety of real objects are connected to the internet. The various things connected through the network, in the real space, work collaboratively through the network. Social Things provides a new collaborative service which links the networked things with their context.

The most important thing networked things need is to communicate with each other via “ identifying the communication partner”. Since the identifier varies as it is based on the rules implemented by each vendor, communication between services and different devices is difficult. Social Things proposes a unique identifier for each device and service under a unified name space. However, the unique identifier is too complicated to understand by the users. Thus, a user-friendly identifier was also suggested. An experiment was conducted to test the proposed name resolution system between user and system identifier.

### Keywords:

Name Resolution, Context, Internet of Things

Graduate School of Media Design, Keio University

Ayaka Imaki

# 目 次

第 1 章 序論	1
第 2 章 Social Things	4
2.1. 「モノ」とは	4
2.2. Social Things の定義	5
2.3. Internet of Things (IoT)	8
2.3.1 Social Web of Things	9
2.4. 構成要素	12
2.4.1 通信プロトコル	12
2.4.2 資源発見	12
2.4.3 記憶領域	12
2.4.4 コンテキスト	13
2.4.5 ビッグデータ処理	13
2.4.6 名前と識別子	13
第 3 章 関連研究	14
3.1. Internet of Things で用いられている識別子	14
3.1.1 ベンダー固有の識別子	14
3.1.2 標準化された識別子	17
3.1.3 識別子まとめ	20
3.2. 識別子と名前の変換	21
3.2.1 名前解決	21
3.2.2 検索	23
3.2.3 まとめ	26

3.3.	課題と目的	26
<b>第4章</b>	<b>Social Things における名前と識別子の提案</b>	<b>30</b>
4.1.	概要	30
4.2.	名前と識別子	32
4.2.1	システムが識別するための識別子	32
4.2.2	ユーザが指示する名前	32
4.3.	名前と識別子の対応付け	32
<b>第5章</b>	<b>システムが識別するための識別子と属性情報</b>	<b>35</b>
5.1.	システムが識別するための識別子の設計	35
5.1.1	必要な機能と検討	35
5.1.2	要件を満たす識別子の提案	41
5.2.	実装	43
5.2.1	目的	43
5.2.2	開発環境	43
5.2.3	属性情報の抽象化	43
5.2.4	システム構成	45
5.3.	評価	50
5.3.1	評価概要	50
5.3.2	結果	51
5.4.	考察	59
5.4.1	属性情報の取得について	59
5.4.2	名前解決アルゴリズムの計算時間	61
5.5.	まとめ	62
<b>第6章</b>	<b>ユーザが指示するための名前とその補助情報</b>	<b>64</b>
6.1.	設計	64
6.2.	利用イメージ	70
6.3.	考察	73
6.3.1	ユーザ側の名前について	73

6.3.2	補助情報について . . . . .	73
第7章	結論	74
	謝辞	76
	参考文献	77



# 目 次

1.1	各コンセプトの比較図 . . . . .	3
2.1	Internet of Things と Social Things のイメージ図 . . . . .	5
2.2	Social Things イメージ図 . . . . .	6
2.3	Social Web of Things タイムライン図 . . . . .	10
2.4	Social Web of Things ユーザーに属する Things ( Friends ) . . . . .	11
3.1	日立エアコン モバイルコントロール システム構成図 . . . . .	15
3.2	実際に届いたメール ( 実行結果をお知らせ ) . . . . .	16
3.3	i TamaHome 操作画面 . . . . .	17
3.4	hue の電球とブリッジ . . . . .	18
3.5	hue のシステム図 . . . . .	19
3.6	国民 ID とマイナンバー . . . . .	20
3.7	DNS の仕組み . . . . .	22
3.8	nslookup コマンドによる、IP アドレスの問い合わせ結果 . . . . .	23
3.9	「もしかして」の表示例 . . . . .	25
3.10	電話帳の名前解決方法 . . . . .	28
4.1	提案するシステムの流れ . . . . .	31
4.2	Attribute を用いない名前解決のイメージ . . . . .	33
4.3	Attribute を用いた名前解決のイメージ . . . . .	34
5.1	動的 IP アドレスのイメージ . . . . .	37
5.2	Mac book air のシリアルナンバー . . . . .	38
5.3	SONY 5000 のシリアルナンバー . . . . .	38

5.4	MAC アドレスのフォーマット . . . . .	39
5.5	MAC アドレスのイメージ . . . . .	40
5.6	システム側の識別子、Attribute . . . . .	41
5.7	システム構成図 . . . . .	45
5.8	tcpdump コマンドによる出力結果 . . . . .	46
5.9	Ruby/Pcap 拡張ライブラリによる出力結果 . . . . .	47
5.10	IP アドレスの中身 . . . . .	48
5.11	実行結果 機器の名前1 【ニックネーム + 機器名】 . . . . .	52
5.12	実行結果 機器の名前を取得 . . . . .	53
5.13	実行結果 IP アドレスの名前解決 . . . . .	54
5.14	実行結果 行っている通信を表示 . . . . .	55
5.15	ブラウザから IP アドレス「174.129.224.33」へのアクセス結果 . . . . .	56
5.16	実行結果 ポート番号1 . . . . .	56
5.17	実行結果 ポート番号2 . . . . .	57
5.18	一つのパケットに要した時間 . . . . .	58
5.19	実行結果 SYN / ACK 情報 . . . . .	59
5.20	加藤朗教授が作成した distribution map . . . . .	60
5.21	ページの表示にどれぐらい時間がかかると「遅い」と感じるのか。 調査対象 1048 名 . . . . .	61
5.22	Firefox about:config 設定画面 . . . . .	62
6.1	ユーザの指示する名前が明確である場合 . . . . .	68
6.2	Attribute から特定する . . . . .	69
6.3	ユーザとのやりとりの中でインタラクティブに特定する . . . . .	69
6.4	特定のイメージ . . . . .	70
6.5	例) 家の中にある「モノ」の配置 . . . . .	71
6.6	テーブル作成の試行実験 . . . . .	72

# 目 次

5.1	開発環境 . . . . .	43
5.2	属性情報のテーブル . . . . .	44
5.3	テーブルに入る情報 . . . . .	44
5.4	ポート番号 変換 . . . . .	49
5.5	SYN / ACK 変換 . . . . .	50
5.6	動作させた環境 . . . . .	50
5.7	ネットワーク環境 . . . . .	51
5.8	識別子の分類 . . . . .	53
5.9	属性情報の変換 . . . . .	59

# 第1章

## 序

## 論

無線通信機能の向上、小型化、クラウドサービスの発達により、近年ではパソコンやスマートフォンだけではなく、日用品や電化製品 [1]、自動車 [2] など、様々な「モノ」がインターネットをはじめとする通信インフラに接続されるようになってきた。これを Internet of Things と私たちは呼んでいる。例えば、医薬品の分野にも Internet of Things が広がってきている [3]。マイクロチップを入れることによりつくられた「Smart-pill」[4] の誕生により、本来飲むべき薬がきちんと患者の体内で消化されたということをも、インターネットを介して知ることが出来る世界になってきている。

2015年時点で約150億個と言われている Internet of Things であるが、2020年には約500億個 [5] の「モノ」がインターネットに繋がり、そのうちの3分の1はコンピュータやスマートフォン、タブレットやテレビだと言われている。残りの3分の2は従来のインターネットに繋がっていなかった「モノ」が繋がってくる。

「モノ」と人間という単一の繋がりだけではなく、「モノ」と「モノ」がインターネットを介して行う連携により、より高度なサービスを行えることが考えられてきている。その一例として Social Things などの試みが行われている。例えば、人が伝える「照明を消して」という命令に対し、照明とテレビが通信を行うことにより、テレビのコンテンツに合わせて照明の明るさが変化をする。それにより、まるで画面の中にいるかのような空間演出ができ、エンターテインメントの可能性が広がるだろう。また、テレビの近くにある照明は、周囲の明るさを考慮して少し暗めに変化をする。そして、ブラインドが環境光を室内に入るよう調整をしたり、人間同士の関係性を考慮して動作を変える世界になると考える。

それぞれの概念の違いを図 1.1 に示す。インターネットにパソコンが繋がった

ことにより、人間同士が「モノ」を通じて繋がるようになった。そして、「モノ」と「モノ」が繋がり、「モノ」の通信が Internet of Things で行われるようになった。そのやりとりを人間が見れるようになったのが Social Web of Things である。これは、Social Things のレベル0の段階（基礎）である。ネットワークに繋がった「モノ」同士が人間を介すことなく互いに通信をし合い、コミュニティを形成するというのが、一つ進んだ段階の新しいコンセプト Social Things である。「モノ」と「モノ」のやりとりを人間が理解し、関与することが出来る世界である。

たくさんの「モノ」がインターネットに繋がり、相互にコミュニケーションを行う世界が到来しつつある。このような人間と「モノ」との関係を考えて時に、適切な機能を持った基盤が必要となる。例えば、人間と「モノ」がコミュニケーションを行う上で極めて重要となるのが「通信相手を識別する」ことが、基盤に求められる機能の一つである。そのためには、識別子や、解釈が必要となる。しかしながら、現状のシステムでは、各ベンダ毎に「モノ」に対する識別子の付け方が異なることにより、連携をすることの出来る「モノ」の選択肢が限られている。そして、「モノ」のやりとりを人間がのぞき見ても、理解をすることが難しいため関与することが出来る範囲も限られてしまっている。

本研究では、この問題を解決するために、インターネットに繋がった「モノ」に対するベンダに依存しない統一的な名前空間による名前付け、および、ユーザが指示する名前からの変換の際に Attribute（属性情報）から名前解決を行うシステムの提案を行う。

本論文は全7章から構成される。第2章で、インターネットを繋がった「モノ」と「モノ」が連携して、高度なサービスを行う Social Things について述べる。第3章では Social Things を実現するにあたって、構成される技術、要素の既存の取り組みについて述べる。第4章では、ユーザのコンテキストを考慮した名前解決システムの提案、第5章でシステム側の識別子からユーザが理解できる名前への変換について、第6章でユーザが指示する名前と補助情報についての検討を行う。それらを踏まえた結論と、今後の課題と展望を第7章で述べる。

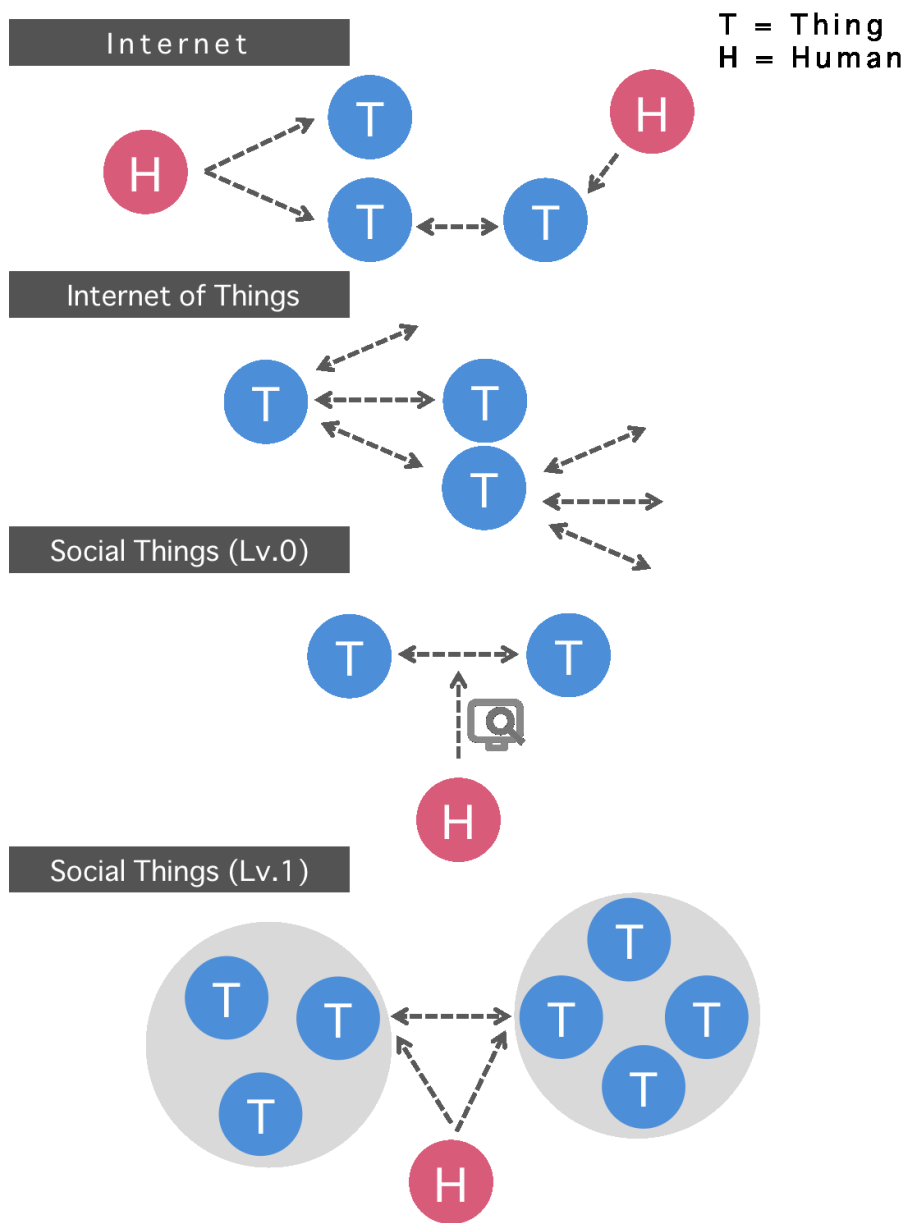


図 1.1: 各コンセプトの比較図

## 第2章

# Social Things

本章では、我々の提案する Social Things の説明を行う。そして、Social Things を実現する上で必要となる基盤技術について示す。まずはじめに、Social Things における「モノ」のはどのようなものであるかの定義付けを行う。

### 2.1. 「モノ」とは

本研究では、「モノ」を以下のように定義する。

“実体があり、かつ、有形であるもの” かつ “継続して(1ヶ月以上)存在する”

- 「モノ」の例
  - － 冷蔵庫
  - － 腕時計
  - － 照明
  - － ぬいぐるみ
- 本論文で「モノ」とみなさない例
  - － 分子
  - － 細胞
  - － 割りばし
  - － お菓子

## 2.2. Social Things の定義

実在するほぼ全ての「モノ」がインターネットに繋がることにより、今までは考えられなかった新しい社会を私たちは過ごすことが出来るようになってきている。例えば、家庭用電気ポットがインターネットに繋がることにより、遠く離れたお年寄りの生存確認が出来る、みまもりほっとライン [6] の誕生など、IoT により見守りの可能性が広がった。

Internet of Things と Social Things との違いを図 2.1 に示す。

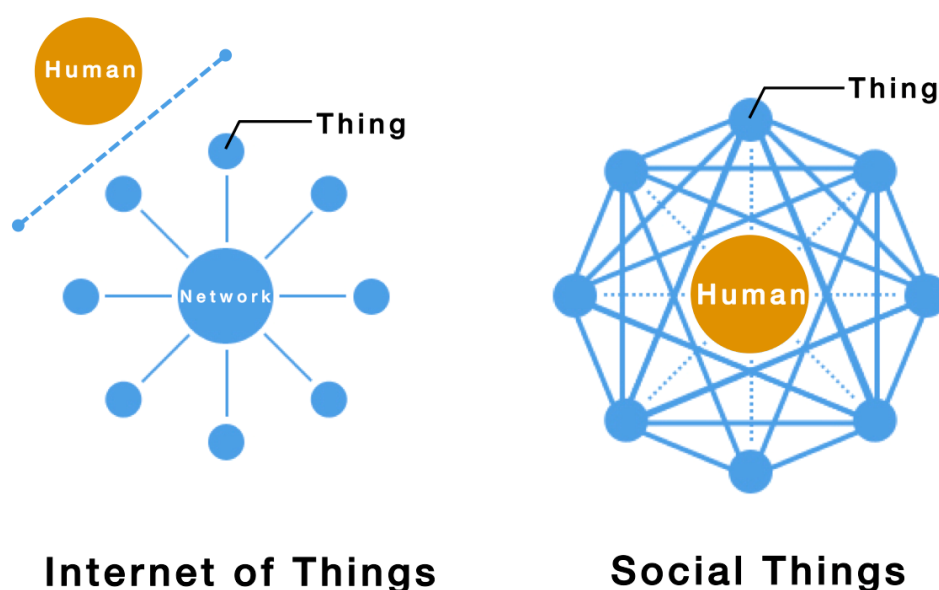


図 2.1: Internet of Things と Social Things のイメージ図

Internet of Things の世界においては、「モノ」と「モノ」が通信をする上で、そのやりとりを人間が理解し関与することが難しく、「モノ」と「モノ」が何かをやっている世界と人間という関係である。一方、Social Things では、「モノ」と「モノ」のやりとりを人間が理解することができ、そのやりとりに対して人間が関与する有機的な関わりが生じる。現状の Internet of Things の先には、このような世界が生じると考える。図 2.2 のように、「モノ」が「モノ」や情報と連携しあうことで彼ら自身の目的を達成し、結果として私たち人間に対して影響を与



えるという世界である。

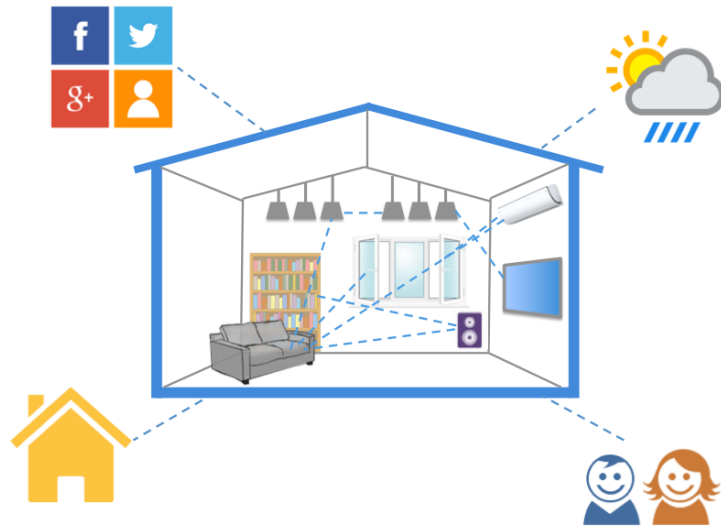


図 2.2: Social Things イメージ図

Social Things では、「モノ」が単にネットワークに繋がり、ON, OFF を人間に伝えるだけの無機的な「モノ」ではなく、個々に意思を持つことによって「モノ」の中でも人間のようなコミュニティが出来上がり、結果として人間に影響を与えることが出来るという可能性に着目している。「モノ」同士の中で起こっていることが複雑化することにより、人間が全て理解することに非常に苦労するようになってきた。その会話に人間が関わり、相互に関わることで、違う社会を創ることができると考えられる。

1. 「モノ」同士がコミュニティ、グループを形成する 2. 目的を達成するために、他の「モノ」や人と協力、協同する 3. 常に人間の支配下にいるわけではないの3つの要素を持つものを Social Things と定義する。

1. 「モノ」同士がコミュニティ、グループを形成する

ひとつの「モノ」がひとつの行動をすることで終わりではなく、「モノ」A が起こした行動が、「モノ」B, C の行動に影響を与えることが出来る。

2. 目的を達成するために、他の「モノ」や人と協力、協同する

彼らはオールラウンドプレイヤーではないため、全てのことをこなすことは

出来ない。個の目的を、他の「モノ」の力を借りることで達成することが出来る。

### 3. 常に人間の支配下にいるわけではない

彼らは人間の指令にただ答えるだけのデバイスではない。彼らが彼らのタスクを行うことにより、結果として人間の役に立ったり、影響をあたえることが出来る。

街全体の「モノ」が意思を持つことにより、街全体で人を見守ることが出来る。例えば、6歳の子どもミキちゃんがおばあちゃんの家で一人で遊びに行く時、ミキちゃんの事を中心に見守ってくれる靴を履いてお母さんと駅まで歩く。そして、電車と靴が反応をして、乗るべき電車へとミキちゃんを自然に導く。おばあちゃんの家最寄り駅に近づくと、靴が反応をしてミキちゃんに下車を知らせる。駅の改札を通ると、改札がおかあさん、おばあちゃんと連絡を取り合う。商店街を歩いておばあちゃんの家に向かう際には、郵便ポストがミキちゃんの顔を伺い、迷子になっていないかを伺ったり、信号がミキちゃんが怖がっている大型犬と出会わないように上手く誘導をしてくれる。もし、郵便ポストがミキちゃんの顔をしっかりと判断することが出来なければ、その先にあるお店の看板にその役割を担ってもらえるかもしれない。そして、その街に長くいる「モノ」たちだからこそ知っている情報、例えば、いつも通っている細い道で工事をしている、作業車が通ると危険な道があったとき、たとえ遠回りであったとしてもその道を避けておばあちゃんの家に向かえるよう、靴と街全体が通信を行う。そして、おばあちゃんの家近くにつくと、靴からおばあちゃんに連絡がいき、家の前でおばあちゃんが迎えてくれる。おばあちゃんとミキちゃんが出会えたら、そのことがおかあさんに連絡がいくといった仕組みである。

例えば、このように Social Things の元では、古き良き日本の下町にあったような商店街においておじちゃん、おばちゃんが近所の子どもを、自然と見守っていたような、生死の0,1を伝えるような無機的な判断ではなく、緩やかな繋がりからの見守り、有機的な判断を実現することができるようになると思う。

## 2.3. Internet of Things (IoT)

人間の介在をなし、「モノ」とインターネットが、自律的にコミュニケーションを行う情報技術をここでは、Internet of Things と呼ぶ。従来のインターネットは、計算機と計算機が繋がり、その後ろに人間がいてコミュニケーションをしているものであった。つまり、ユーザーである人間が主体的に操作をすることで、情報のやりとりが行われてきたのである。Internet of Things の考え方では、このインターネットに多様な「モノ」を接続することにより、よりダイナミックかつ自律的な情報のやりとりが実現されることが期待されている。

Internet of Things の一例として、Nike の事例を挙げる。

- Nike+ FuelBand  
日々の人間の活動を測定し、活動量を Nike Fuel という新しい単位に変換される。活動量が増えると Nike Fuel をより多く貯めることが出来る。
- Nike+ Sports Watch  
GPS が内蔵されているため、距離、ペース、時間、スプリット、高度、ルート燃焼カロリーを記録することが出来る。
- Nike+ GPS App  
走ったルートや距離、ペースを Google マップに示し確認することが出来るアプリケーション。
- Nike+ Basketball, Nike+ Training  
圧力センサや加速度計が内蔵された靴により、ゲームやトレーニング時のあらゆる動きを測定することが可能に。Nike Fuel に加え、垂直跳躍や速さ、カロリーなどが可視化される。

このように、Internet of Things によりリアルタイムな状況把握をし、その情報を自ら発信することが可能になった。そして、トレーニングの記録やフィードバックを個人が容易にふりかえられるようになり、スポーツの新たな可能性が広がったと言えるだろう。Internet of things で、わたしたちの身の回りにある「モノ」

が、自分のエージェントとしてリアルタイムに情報を取得、送信出来る機能を持つようになった。

Internet of Thingsの世界においては、これらのデバイスから取得されたユーザのデータが可視化されふりかえられるようになったが、Social Thingsの世界においては、例えばテニスであれば、ラケットとネットが通信を行い、軌道が低いのを修正するためにわざと通常よりもネットの高さを高くするという行動を起こす事が出来る。

### 2.3.1 Social Web of Things

Internet of Thingsの下で、Social Web と Internet of Things を合わせた、Social Web of Things というコンセプトを Ericsson が提唱している [7]。Internet of Things では、「モノ」が行った行動の最終結果がユーザーに返されるため、ユーザーが何が行われていたのかを理解することが出来ず、メンタル・モデルの中で乖離が生じている。そこで、Social Web of Things では「モノ」が行っている行動を、図 2.3 のように Facebook や Twitter のような一つのタイムライン上で確認することが出来る。

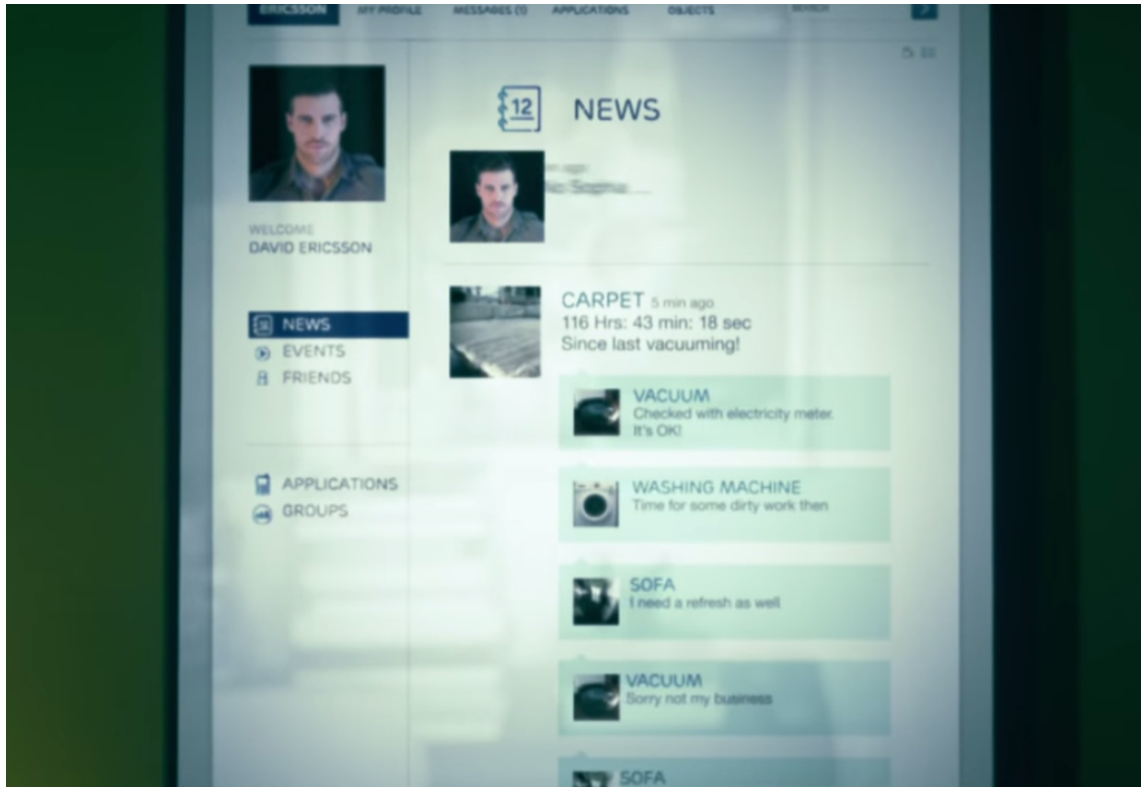


図 2.3: Social Web of Things タイムライン図  
( 出典 : [http://youtu.be/1syj\\_2jfJ4g](http://youtu.be/1syj_2jfJ4g) (2014) )

Social Web of Things のコンセプトビデオの中では、以下のようなストーリーで、家の「モノ」と人間が会話を行っている。David という男性が、会社を出たという情報とともに Sophia という女性をディナーに招くことを、家に対して伝える。すると、家が「Sophia が家に入れるように、鍵を開ける？」と David に対して聞く。David が「Yes」と答えると、家の中にあるものが Sophia を迎え入れるために準備をしだす。例えば、オーブンは、David の到着時に調理をしやすいように省電力モードで稼働しはじめる。そして、カーペットが「しばらく掃除されていない」ということをつぶやく。すると、掃除機が、カーペットの掃除をはじめめる。また、ソファも「そろそろキレイにして欲しい」とつぶやくが、それに対し掃除機は「ソファの掃除はボクの仕事じゃない(ボクには出来ない)」という返事をする。しかし、突然 David から「ディナーの予定が中止になった」

ということが家に対して伝えられる。すると、テレビは、David が観たいと思っているサッカーの試合を、そして音楽プレーヤーは、David が好きな音楽リストをダウンロード、再生し、家全体が落ち込んでいる David を温かく迎え入れてくれる。

これらのやりとりが、タイムライン上で行われることで、Social Web of Things では何が家の中で行われているのかを人間が理解することが出来るようになった。

「モノ」を擬人化し、人間にとっての友達のように扱い、人間と「モノ」のコミュニティを示している。(図 2.4 を参照)そして、Facebook や Twitter のようなタイムラインを用いることによって「モノ」のやりとりの可視化を行っている。しかし、文字によるやりとりだけでは、人と「モノ」、「モノ」と「モノ」の関係性やコンテキストを表現することが出来ない。命令を受動的に実行しているだけで、「モノ」同士のコミュニティからユーザに影響を与えることには至っていない。

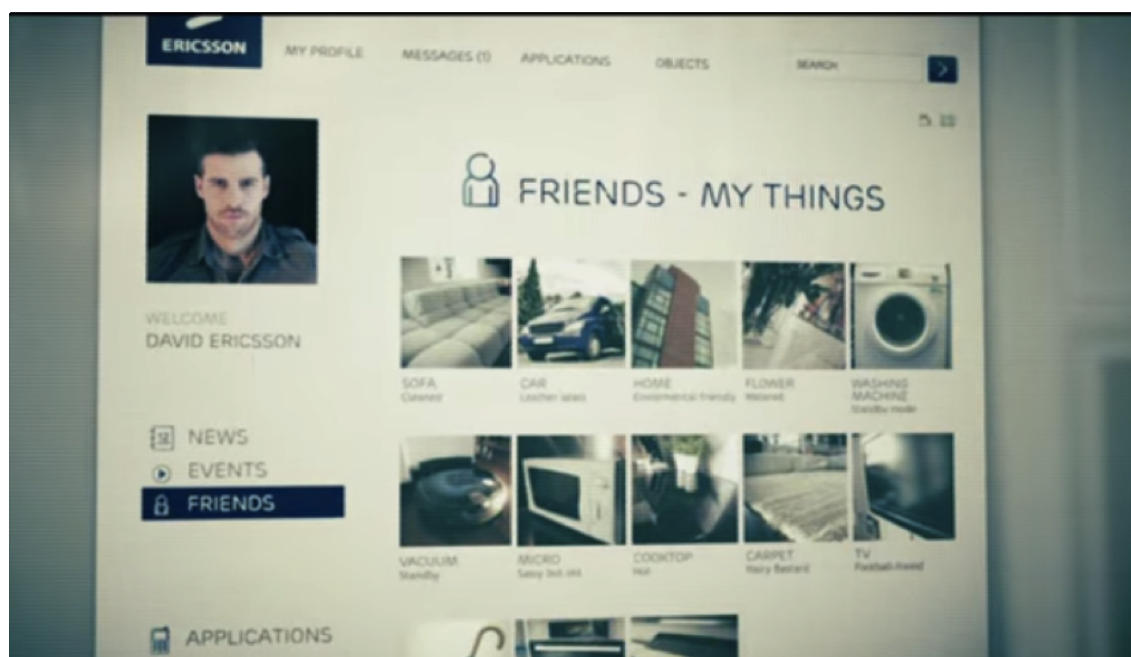


図 2.4: Social Web of Things ユーザーに属する Things (Friends)

( 出典 : [http://youtu.be/1syj\\_2jfJ4g](http://youtu.be/1syj_2jfJ4g) (2014) )

また、この Social Web of Things はコンセプト段階なので、実現方法や名前付け

システムについては言及されていない。Social Web of Things で出てくる VACUUM や SOFA, WASHING MACHINE は、タイムライン上で有効、つまり FRIENDS というドメイン内で有効であるため、ドメインを超えた範囲からは制御する出来ない。

## 2.4. 構成要素

Social Things の社会を実現する上での課題を以下に示す。

### 2.4.1 通信プロトコル

Social Things の世界においては、今まで以上に様々な「モノ」がネットワークに繋がることにより、通信に対する負荷が増えるため、ネットワークの性能向上が求められる。

通信プロトコルとして、センサ情報をインターネット・オンライン化することに加えて、アプリケーションソフトウェアやクラウドサービスをベンダの枠を超えて連携することが出来るようになる、IEEE1888 (FIAP) [8]。ネットワーク上にミラーサーバを置き負荷を分散する技術である CDN [9] を用いる。

### 2.4.2 資源発見

たくさんの「モノ」が繋がった中で、それぞれの情報を一から登録していくべきなのか否かの課題が挙げられる。

機器を接続するだけで、コンピュータネットワークに参加することが可能となるプロトコルである UPnP [10]、Bonjour [11] などの技術が存在する。

### 2.4.3 記憶領域

「モノ」と人間、「モノ」と「モノ」の通信のデータを保持する領域が必要となる。

非常に安価なクラウドストレージサービス Amazon S3 ( Amazon Simple Storage Service ) [12] などが提供されている。

#### 2.4.4 コンテキスト

ユーザの使用シーンにおいた適切な判断をすることが求められる。

認知科学の研究が進み、システムが自然言語、仮説生成、根拠に基づく学習を活用するテクノロジーにより、経験を通じて学習し、相関関係を見つけて将来についての仮説を立てる、また成果から学習することができるシステムである、コグニティブコンピューティング [13] [14] が研究されている。主な例として、Watson [15] がある。

#### 2.4.5 ビッグデータ処理

「モノ」と人間に加え、「モノ」と「モノ」の通信も行われるため、一度に大量のデータの処理が求められる。

大規模データを効率的に分散処理・管理するためのソフトウェア基盤である Hadoop [16] などの試みがある。

#### 2.4.6 名前と識別子

「モノ」に対する名前付けと、各「モノ」につけられた識別子について検討をする必要がある。

IPv4 を用いたり、各ベンダの決まりに基づいた識別子があるが、統一された仕組みがあるわけではない。第 3 章で識別子と名前についての関連研究、議論を行う。



## 第3章

# 関連研究

本章では、2.4節で述べた Social Things の社会を実現する上での課題の中で、識別子の部分の関連研究を挙げる。

### 3.1. Internet of Things で用いられている識別子

#### 3.1.1 ベンダー固有の識別子

白くまくん

日立製作所の提供する、スマートフォンからアプリケーションを通じて操作が出来るエアコン [17] である。システム構成図は図 3.1 である。実行結果は、毎回図 3.2 のように、ユーザにメールで送られる。操作、接続方法は以下のように行う。

1. ネットワークに繋いだ無線 LAN 接続アダプタをとエアコンを接続する
2. 無線 LAN 接続アダプタの ID とパスワードをアプリケーションをダウンロードしたスマートフォンに入力する
3. エアコンの設置場所を選択、ニックネームを登録する
4. スマートフォンに、稼働状況などをメールでお知らせ

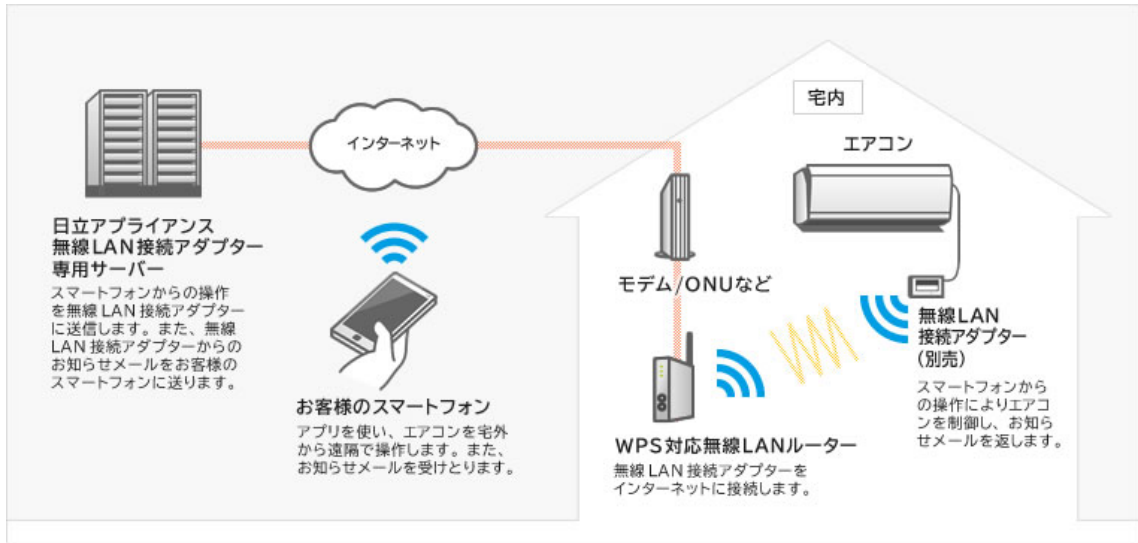


図 3.1: 日立エアコン モバイルコントロール システム構成図

( 出典 : <http://kadenfan.hitachi.co.jp/ra/app/about.htm> )

エアコン自体に識別子が付けられているのではなく、無線 LAN 接続アダプタに識別子が付けられており、それをユーザが入力することで紐付けを行っている。

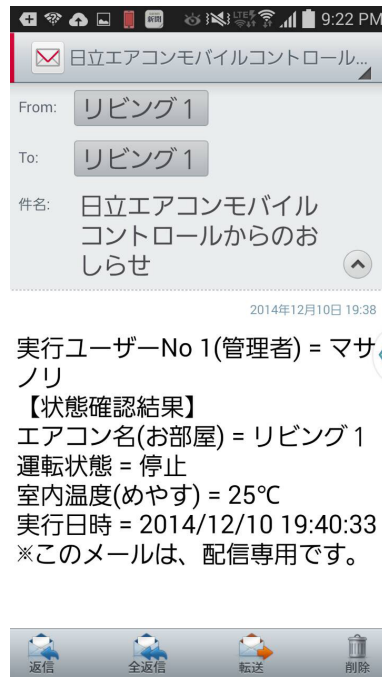


図 3.2: 実際に届いたメール（実行結果をお知らせ）

## i TamaHome

i TamaHome [18] は、ネットワークを介して室内に設置した赤外線リモコンを作動させことにより、ユーザがスマートフォンからテレビやエアコンの制御を出来るようにしたものである。

操作画面は図 3.3 のようになっており、エアコンやカーテンなどの操作がアプリケーション上から行えるようになっている。音声認識、もしくはタッチパネルで操作が行える。スマートフォンで行った操作が、室内に設置した学習リモコン機器を通じて、各機器に命令を送ることが可能となっている [19]。



図 3.3: i TamaHome 操作画面

( 出典 : <http://www.g-mark.org/award/describe/40478> )

### 3.1.2 標準化された識別子

#### hue

Philips 社の提案する、アプリケーションやネットワークを通じて色や ON/OFF が制御出来る電球である。図 3.5 は接続の仕組み図で、操作、接続方法は以下のように行う。

1. 図 3.4 の「ブリッジ」を LAN に有線接続する。



図 3.4: hue の電球とブリッジ

( 出典 : <http://www2.meethue.com/> )

2. アプリケーション [20] をダウンロードしたスマートフォンやタブレットを同一 LAN 環境で立ち上げる。
3. hue ブリッジとペアリングを行う。
4. hue 電球とブリッジが ZigBee Light Link で接続されているので、アプリケーションで示す動作を hue 照明が行う。

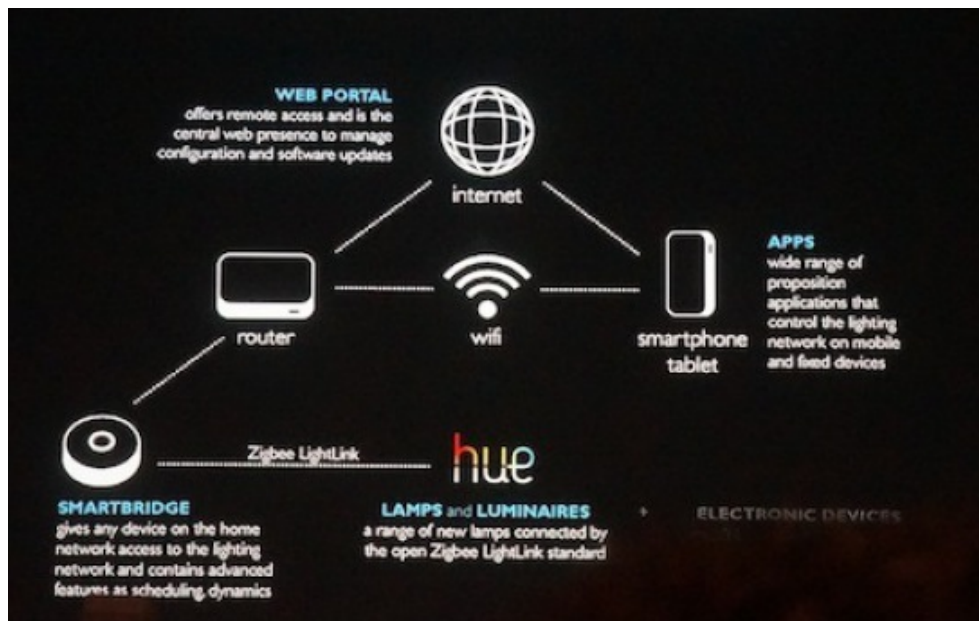


図 3.5: hue のシステム図

( 出典 : <http://japanese.engadget.com/gallery/philips-friends-of-hue/> )

ZigBee Light Link [21] とは、無線通信プロトコルのところで、最大 65,536 個の端末を一つのネットワークに接続可能で、端末同士が通信することが出来るという特徴を持つ、照明を操作することに特化した仕組みのこと。一つの hue ブリッジには最大 50 個の hue 電球を接続でき、hue 照明とブリッジは 30m 離れていても通信できる。hue は各電球の識別に、ZigBee の ID を使用している。

また、50 個の hue 電球を接続した場合は、利用者がそれぞれが何を指すのか (照明 1,2,3... や、ニックネーム、場所など) を把握する必要がある。

## マイナンバー

日本では現在、健康保険被保険者番号、パスポートの番号、基礎年金番号、運転免許証番号、住民票コード、などは、各行政機関が個別に番号を割り振っている。そのため、個人番号の運用による行政の混乱による問題も生じている。2013 年 5 月に成立した「行政手続における特定の個人を識別するための番号の利用等に関する法律 (マイナンバー法)」における、識別子である [22]。国民一人ひと

りに固有の番号を割り当て、それに基づいて、行政サービスを提供する仕組みで、法人にも法人番号が割り振られる。

マイナンバーは、ユーザが利用する番号として、利用者が見える 12 桁の数字（法人は 13 桁）で表され、個人の氏名や住所から特定をしていた「個人」を一つの番号から特定する事が可能となる [23]。それに対し、データ連携のためにコンピュータが処理するための番号を国民 ID と呼んでいる。図 3.6 のように 2 つは使われ方が異なる。

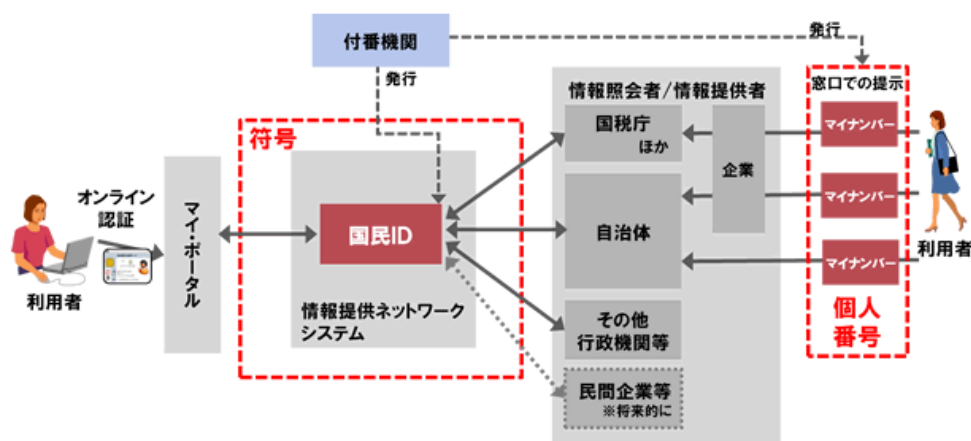


図 3.6: 国民 ID とマイナンバー

( 出典 : <http://cgs-online.hitachi.co.jp/contents/288.1.html> )

### 3.1.3 識別子まとめ

各サービスにおける名前付けは、LAN 接続アダプタが各機器に対して、ベンダーで固有の識別子を割り当て、各ベンダが対象とする連携のために必要な情報が内包されるという形で行われる。ユーザは接続アダプタを経由して、アプリケーションなどで制御が出来る仕組みになっている。しかしながら、各サービス領域を超えたネットワークに繋がる「モノ」たちを他サービスが認識できる形での名前付けは行われていない。

一方、ベンダを超えた通信をすることが可能となっている hue であると、多くの機器が繋がった時に、ユーザ側がどの識別子がどの電球を指すのかを記憶し

ておく必要がある。また、マイナンバー制度では、ユーザが12桁の数字列を覚える事が求められ、ユーザにとって使いやすい形であるとは言えない。

これらから、ベンダに依存しない識別子、かつ、ユーザが用いる識別子は容易なものが必要であると考える。

## 3.2. 識別子と名前の変換

### 3.2.1 名前解決

#### Yellow Page (電話帳)

Yellow Page とは、配信者の情報を電話帳のように集めたサイトのことである。日本版の Yellow Page はタウンページ。

例えば、「慶應義塾大学大学院メディアデザイン研究科」に電話を掛けたいときに、「045-564-2517」という番号を覚えている人はほとんどいないので、ユーザは電話帳を用いる。図 3.10 のように、電話帳は「KMD」に対応する「045-564-2517」という番号を返す。このことを、「ひもづける」、「名前解決をする」という。ユーザにとって分かりやすい言葉(名称、人名)を手がかりに、その名称の電話番号を特定する。

#### FQDN

インターネットの世界においても、名前というものが必要であり、様々な名前の形式が作られてきた。その一例が Domain Name System (DNS) である。DNS とは URL などの FQDN と、コンピュータ同士のやりとりに必要な IP アドレスを関連付ける「名前解決」の機能を提供する [24]。ドメイン名とは、「www.kmd.keio.ac.jp」のようにラベルを「.」で区切る形で付けられたインターネット上での住所にあたるものである。インターネットに繋がられた「モノ」同士が通信をする際には、「131.113.136.195」といった IP アドレスを元に通信相手を特定しているが、この数字をユーザが識別するのは非常に難解である。そこでユーザにとってわかりやすいドメイン名 (www.kmd.keio.ac.jp) に変換し利用されているのである。この、



ドメイン名とIPアドレスを対応付ける仕組みが Domain Name System ( DNS ) である。

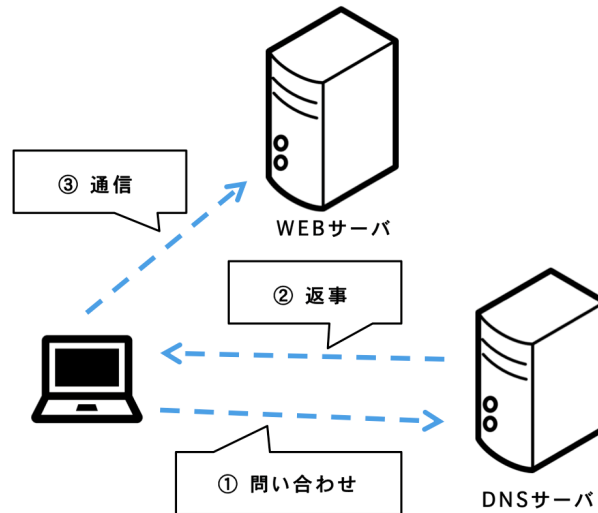


図 3.7: DNS の仕組み

図 3.7 のように

1. 「www.google.com」を IP アドレスで表記し直すと何になるのか問い合わせる。
2. 「www.google.com」に対応する IP アドレスが返ってくる。
  - 「173.194.117.144」
  - 「173.194.117.145」
  - 「173.194.117.146」
  - 「173.194.117.147」
  - 「173.194.117.148」
3. 上記の IP アドレスを元に WEB サーバと通信を行う。

```
Last login: Sun Dec 14 22:40:32 on ttys003
[~] nslookup 22:55:48
> www.google.com
Server:          131.113.138.164
Address:         131.113.138.164#53

Non-authoritative answer:
Name:   www.google.com
Address: 173.194.117.146
Name:   www.google.com
Address: 173.194.117.147
Name:   www.google.com
Address: 173.194.117.148
Name:   www.google.com
Address: 173.194.117.144
Name:   www.google.com
Address: 173.194.117.145
> █
```

図 3.8: nslookup コマンドによる、IP アドレスの問い合わせ結果

このように、インターネット上での通信を行う上で、名前解決は欠かすことが出来ない不可欠なものとなっている。DNS は、IP アドレス ( 131.113.136.195 ) を名前 ( www.kmd.keio.ac.jp ) に変換して、各ホスト ( ネットワークにアクセスしている「モノ」 ) を可読出来る形で認識することが出来る。

しかしながら、「モノ」と「モノ」が自律的な通信を行う必要がある Social Things を行う上ではこの DNS では必要な連携を実現することが出来ない。

### 3.2.2 検索

#### RNS(Real Name System)

Real Names 社 [25] が開発をした、ドメイン入力ではなく、単語や固有名詞などの実名の入力でも Web ページへのアクセスを可能にする技術のことである。DNS は、ユーザにとって覚えることが用意ではなく、入力を一文字でも間違えると目的の Web ページに到達することが出来ないことが多い。

リアルネームの技術を使用することによって、アドレスバーに企業名や商品名のキーワードを入力すると、それに関連付けられた企業などの Web サイトに転送される [26]。

例えば、「慶應義塾大学メディアデザイン研究科」の Web ページへのアクセスが「<http://www.kmd.keio.ac.jp/>」ではなく、「慶應 メディアデザイン」や「KMD」でアクセスが可能になる。

## Google 検索

ユーザが自分が求める Web ページの検索をする際に用いる検索エンジンの一つが、Google 検索 [27] である。検索エンジンの仕組みとして、1. クローリング, 2. インデックス, 3. ランキングの 3 つのステップ [28] がある。

### 1. クローリング

クローラと呼ばれるソフトウェアを使用して、公開されている Web ページの情報を取得する。そして、それを Google のデータベースまで運ぶ。

### 2. インデックス

クローリングでデータベースに運ばれ、それを登録することをインデックスという。インデックスを作成することで、検索対象の正確な場所を把握 [29] する。

### 3. ランキング

インデックスされた Web ページを、キーワードとコンテンツの関連性など 200 項目以上の要素 ( アルゴリズム ) から総合的に分析し、ページの検索順位を決定する。

検索アルゴリズムの例 [30] を以下にいくつか挙げる。

#### ● ナレッジグラフ

コンピュータが現実世界の「モノ」や、「モノとモノの間のつながり」を認識できるように開発された [31] 技術。複数の解釈があると認識した場合には、検索結果に複数の候補を表示する。

- オートコンプリート  
ユーザが検索する可能性のあるものを予測する。
- クエリの解析  
入力された語句の深い意味を理解する。
- 絞り込み  
検索ツールや他のキーワードなど、検索ツールを使い検索を微調整する。
- ユーザのコンテキスト  
地域やウェブ履歴などの要素に基づいて、より関連性の高い検索結果を提供する。
- スペル  
スペルミス特定し、修正したり別の候補を表示したりする。
- 同義語  
同じような意味を持つ単語を認識する。

### 「もしかして」検索

タイプミスと思われるキーワードで検索をしたときに、図 3.9 のように「もしかして」といった形で、検索語の誤りを指摘してくれる機能。



図 3.9: 「もしかして」の表示例

## あいまい検索

基本的な検索エンジンの仕組みは、ユーザが入力した検索語を元に情報を探し表示をする。しかしながら、ユーザが入力する検索語（キーワード）が常に正確ではなく、タイプミスや、表記ミス、変換ミスが生じている。この問題に対応し、ユーザが求める検索結果を表示させるためのクエリ上のプロセスにあいまい検索がある。これにより、正確な検索語でなくとも、適切な結果を表示することが可能となっている。

### 3.2.3 まとめ

名前解決の方法として、現在インターネットで用いられている FQDN は、ユーザにとってある程度認識しやすい名前付けの方法であるが、重複が許されないため長くなりがちであるという課題が挙げられる。ユーザが正確に覚えることのできる長さには限界があるので、わかりやすいルールや、わかりやすい名前を使用することが出来るという事が求められる。

また、ユーザが使用する言葉からシステム言葉に変換する際には、ユーザの間違いや、意図する結果を補助するための仕組みがあることが求められる。

## 3.3. 課題と目的

次章から具体的な本研究の提案をおこなうにあたって、関連研究として現状検討されているシステムを述べた。そのなかで、識別子の部分が、現状のシステムでは、各ベンダ毎に「モノ」に対する識別子の付け方が異なることにより、連携をすることの出来る「モノ」の選択肢が限られている。そのため、「モノ」同士が大きなコミュニティをつくり、連携をし合い動作をすることが難しい。

本研究では、この問題を解決するために、Social Things で必要となる、インターネットに繋がった「モノ」に対するベンダに依存しない統一的な名前空間による名前付けの提案を行う。そして、ユーザが可読可能な形で変換することが求めら

れる。そして、二つ目に、ユーザが扱うことの出来るユーザの指示する名前から、システムが扱う名前に変換するシステムの検討を行う必要がある。

Social Things を実現するためには、インターネットに繋がった「モノ」同士が通信をする必要がある。通信を行う上で一番重要なことは、「通信相手を特定すること」だ。通信相手を特定するためには、相手を識別可能とする必要がある。そして、識別可能とするためには、対象に対し識別子を付与し識別を行う。

また、Social Things は、「モノ」と「モノ」のやりとりが行われている世界に人間が入ったり、「モノ」が人間に対して関与をする世界なので、「モノ」と人間が理解できる識別子を用いる必要がある。しかしながら、人間が理解できる識別子を用いようと考えると、簡単な名前しか使えず（煩雑なものは使用出来ない）、重複が生じてしまう。重複が生じてしまうと、「モノ」側がインターネットを通じてその通信相手を特定することが難しくなってしまう。そこで、「モノ」が扱う識別子と、人間が扱う識別子との対応付け、名前解決が必要となる。

本論文では、Social Things における名前付け、名前解決システムの構築を本論文では目指している。通信相手を識別するための識別子として、以下の2種類の識別子を「モノ」につける。

#### 1. システムが識別するための識別子

- 一意である  
重複があると、通信相手を特定出来ないため、世界中で一意である。
- 「モノ」と「モノ」のやりとりを人がわかる  
設計上、セキュリティの観点から、ネットワーク管理者がフィルタリングをする際に必要な情報（例えばある程度の判断がしやすくなるような、所有者やどんな機器なのか）を人間がわかるためのサポートを考慮する必要がある。

#### 2. ユーザが指示するための名前

- ユーザが名前を表す対象を、具体的にイメージを出来る  
ユーザは、「01:35:55.425417:0.10.1.17.55791」といった整数の羅列では、

識別することができない。ユーザが使用することが出来るためには、意味のある文字列のように、ユーザが可読可能である必要がある。

図 3.10 の Yellow Page の名前解決のように、異なる 2 つの識別子を紐付けることのできるシステムを名前解決システムと呼ぶ。

- 「KMD」  
「KMD」とは、慶應義塾大学大学院メディアデザイン研究科 ( Keio Media Design ) の略称のこと。  
これを本論文ではユーザが指示する名前とする。
- 「045-564-2517」  
「045-564-2517」は、慶應義塾大学大学院メディアデザイン研究科の電話番号。  
これを本論文ではシステムが識別するための識別子とする。

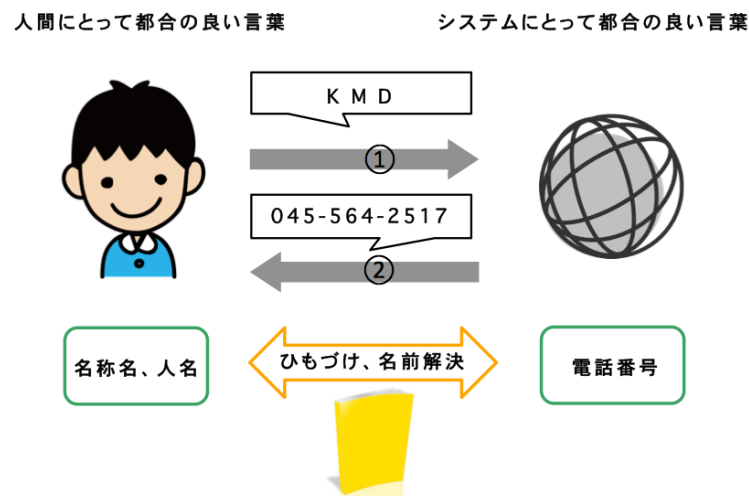


図 3.10: 電話帳の名前解決方法

Social Things では、「照明」という名前一つに関しても、その名前を呼ぶユーザが「会社」にいるのか、「(一人暮らしの)家」にいるのか、「実家」にいるのかにより「照明」という名前が指すオブジェクトは異なる。

一方で、同じ場所で、同じタイミングでAさんとBさんが「おかあさん」という名前を使った場合であっても、近くにいる「おかあさん」ではなく、AさんはAさんの「おかあさん」、BさんはBさんの「おかあさん」を指し示す。

ユーザが指す識別子をシステム側が、ユーザのコンテキストに応じた識別子に翻訳を行うシームレスな動作が求められるようになるが、その機能が現状のInternet of Thingsの基盤では実現することが難しいため、Social Thingsのための翻訳システムの検討を行う。



## 第4章

# Social Thingsにおける名前と識別子の提案

本章において、Social Things を実現する上で必要な基盤システムとして必要となる名前空間および名前付けシステムの提案を行う。第3章で述べたように、「モノ」の名前を制御するための手段は複数あった。しかし、Social Things では人とのインタラクションが必須となり、自然な流れで制御が出来ることが求められるためには現状のシステムでは不十分であるといえる。本研究ではこの問題を解決するために、名前空間および名前付けシステムを提案する。

### 4.1. 概要

実在するほぼ全ての「モノ」がインターネットに繋がることにより、今までは考えられなかった新しい社会を私たちは過ごすことが出来るようになってくるだろう。

例えば、一週間の仕事を終え、帰宅をしたサラリーマンの男性が、テレビで楽しみにしていたサイエンス・フィクションの映画を観る。「照明を暗くして」と指示を出すと、テレビの周りがある照明が暗くなり、部屋の間接照明だけはひっそりと点灯している。このときに暗くなってほしいのは、キッチンや玄関、書斎の照明ではなく、男性が観ているテレビの周辺にある照明である。映画の邪魔をしないために、スマートフォンは、メールマガジン等の重要度の低いお知らせの通知はマナーモードで対応する。映画が終わり、バラエティの番組に切り替わったら照明は明るくなる。日常使用時も、太陽光やテレビが発する明かりに合わせて、

部屋全体が同じ明るさを発するのではなく、各々が周りの環境に合わせて明るさを調整する。また、外の天気を反映して部屋の明るさが変わったり、外で風が吹いていたらエアコンと窓が協力をして部屋の温度を快適に保つ。これらの結果がその家の SNS に投稿され、もしかしたらこれから遊びに来る友人の情報（アレルギーなど）を、あらかじめ家が知っているかもしれない。

このような Social Things の世界のなかで、「モノ」や人がコミュニケーションを行う上で極めて重要となるのが「通信相手を識別する」ための識別子である。同じ「照明」という名前でも、リビングにある照明を指すのか、お父さんの部屋の照明なのかをユーザのシチュエーション、コンテキストに応じて判断することが Social Things の世界では重要となってくる。図 4.1 は、提案するシステムの流れである。ユーザが指示する名前から、キーワードを抽出し、そのキーワードとユーザのシチュエーション、コンテキストから対応するシステムが識別するための識別子への変換を行う。またそれらの要素から絞り出した結果が曖昧な場合であれば、いくつかの候補をユーザ側に返し、ユーザとのやりとりの中で特定をするというインタラクティブな変換を行う。

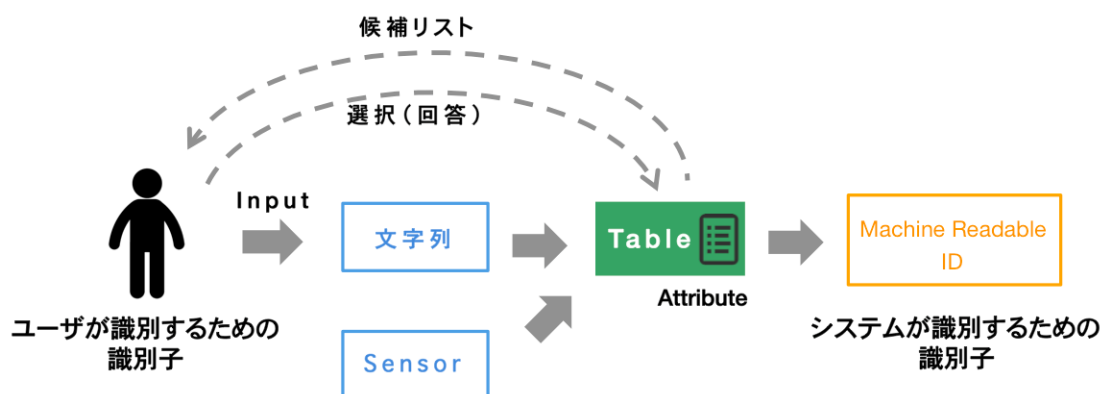


図 4.1: 提案するシステムの流れ

## 4.2. 名前と識別子

Social Things の世界においては、人と「モノ」のインタラクションが大切になってくるため、1. システムが識別するための識別子と 2. ユーザが識別するための名前の 2 つを提案する。それぞれの識別子に求められる要件を述べる。

### 4.2.1 システムが識別するための識別子

1. インターネットに繋がっている「モノ」の中で一意であり、滅多に変わらない
2. 「モノ」同士のやりとりを、人間がある程度理解することができる  
ネットワーク管理者が、フィルタリングをする際にある程度の判断がしやすくなるように設計上検討する必要がある。

### 4.2.2 ユーザが指示する名前

1. ユーザが、名前を表す対象を具体的にイメージを出来る  
扱う人間が覚えられる必要がある
2. 重複が許されている  
一意である識別子であると長くなりがちでユーザにとって使いやすいものではないので、実用性を考慮して重複を許す

## 4.3. 名前と識別子の対応付け

ユーザ側が識別するための名前とシステム側が識別するための識別子の対応付けを図 4.2 に示す。Table が辞書のような役割をし、双方の識別子の紐付けを行う。

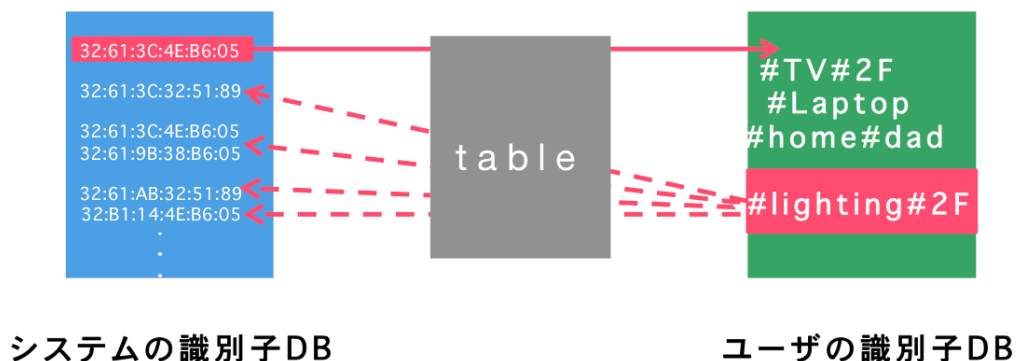


図 4.2: Attribute を用いない名前解決のイメージ

システムが識別するための識別子は一意であるため、そのみでユーザが識別するための識別子を特定することが出来る。しかしながら、「Lighting（照明）」というユーザが識別するための識別子に対して対応するシステムが識別するための識別子は複数あるため、特定することが出来ない。そこで、ユーザが識別するための識別子からシステムが識別するための識別子に変換する際に、図 4.3 のようにユーザの指示する名前をシステムが識別するための識別子に変換する際に、Attribute（属性）を用いることにより、テーブルにある複数の候補から正しいシステムが識別するための識別子の特定をする名前解決をすることが出来ると考える。

Attribute に入る情報については、以下のように定義する。

“ヒトや「モノ」に対して、その状態を規定する情報”

例えば、「Lighting」、「2F」という情報が与えられた時に、実家を指すのか、一人暮らしの家を指すのかを示す情報があれば、特定をすることが出来ると考える。

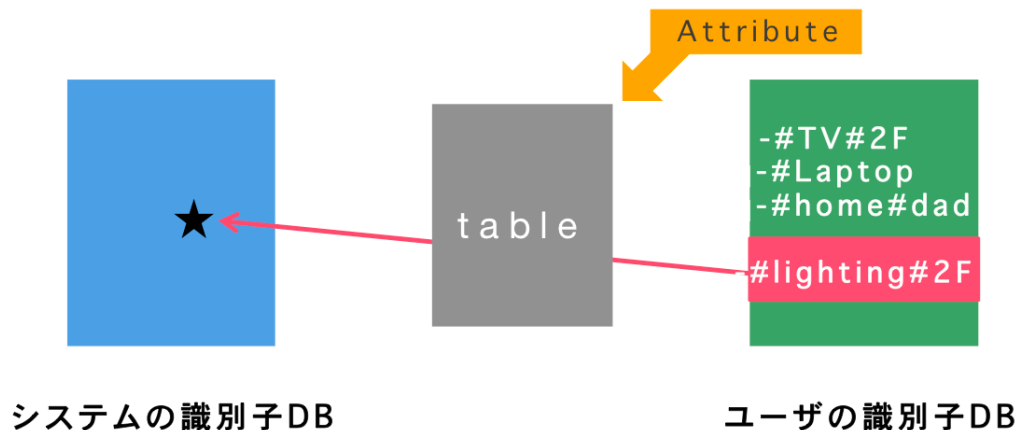


図 4.3: Attribute を用いた名前解決のイメージ

ユーザ側が用いる識別子、名前が煩雑であると、実世界において使いやすいものであるとは言えないので、ユーザ側の識別子はシンプルにする。名前解決の際に、ユーザに付随する情報 (Attribute) をユーザの指示する名前に付与することにより、システム側の識別子を特定する。

ユーザが指し示す「Lighting」に対して、対応されるものは「Lighting」は複数あり、「2F」という情報からも、その「2F」はユーザの神奈川の家なのか、両親が住む京都の実家の2Fなのか、東京の職場の2Fなのかを判断することが出来ない。ユーザの背景情報によって、異なることを指すため、ユーザにはっきりとした説明を求めるか、その他の情報で判断をする必要がある。そこでユーザが指示する名前に加えて、例えば今のユーザの位置情報や、ユーザの状態を結びつけることによって、京都の実家にいるから、京都の家の2Fの照明ということが特定することが出来ると考える。

システムが識別するための識別子から、ユーザの分かる名前への変換を第5章で、ユーザの名前からシステムが識別するための名前への変換を第6章で述べる。

## 第5章

# システムが識別するための識別子と 属性情報

本章では、4章で述べた提案する2つのシステムのうち、システムが識別するための識別子と属性情報について議論を行う。設計、提案する全体像の中の一部に関する実装と評価、そして結果からの考察を述べる。

### 5.1. システムが識別するための識別子の設計

#### 5.1.1 必要な機能と検討

システムが識別するための識別子の検討を行う。現在インターネットに繋がるモノにつけられている識別子として、IP アドレス、製造番号、MAC アドレスを検討していく。それぞれについて、以下のうち 1. 2. の部分を満たす識別子を検討する。

1. インターネットに繋がっているモノの中で一意である
2. 滅多に変わらない
3. 「モノ」同士のやりとりを、人間がある程度理解することができる

#### 1. IP アドレス

現在主流である IPv4 が 32bit であるため枯渇するという問題点が挙げられている。これを解決するために 128bit の IPv6 が普及しつつある。IPv6 に

ついて、セキュリティやIPv6-IPv4 フォールバックなどいくつか問題点があるが、本論文では言及を行わないとする。

IP アドレスには、グローバルIP アドレスとプライベートIP アドレスの二種類がある。一つ目がインターネットに接続されたモノに対して、相手と通信するために一意に割り当てられるIP アドレスで、「グローバルIP アドレス」と呼ぶ。IP アドレスの枯渇問題が生じ、全ての端末がインターネットからアクセス出来る必要はないという考え方より、一定の範囲のIP アドレスを会社や家庭などの組織で一意に割り当てられるIP アドレス「プライベートIP アドレス」として付与する仕組みが出来た。基本的には、プライベートIP アドレスとグローバルIP アドレスを相互変換し、インターネットに接続される。

このIP アドレスは、上述したように、インターネットに接続される機器に対して「一意に」割り当てられる。しかしながら、IP アドレスには動的IP アドレスという仕組みがある。この仕組みにより、ISP (Internet Services Provider) が保有しているIP アドレスのうち、接続されていない空いているIP アドレスが、利用者に順次割り当てられる。図5.1のようにインターネットなどのネットワークに接続をする都度異なる識別番号、が割り当てられてしまう。つまり、自分が使っていたIP アドレスをネットワークから切断後、他人に付与されるということが生じてしまう。

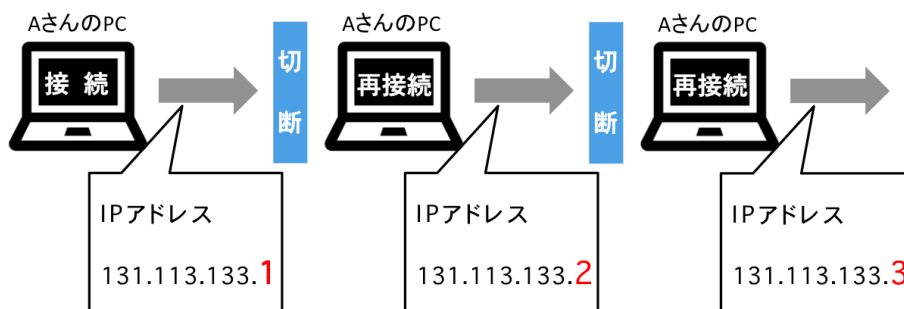


図 5.1: 動的 IP アドレスのイメージ

よって、IP アドレスは、1. 一意である という要件を満たすが、2. 滅多に変わらない という要件を満たさないため、適さない。

## 2. 製造番号

製造番号 (シリアルナンバー) は、様々なモノが出荷時に付けられている番号と定義する。図 5.2 や図 5.3 のように、各製品ごとに一意に特定することの出来る番号が割り当てられている。



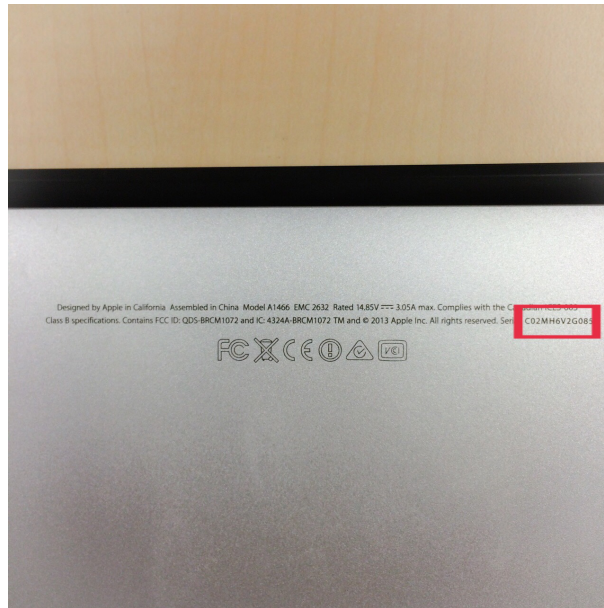


図 5.2: Mac book air のシリアルナンバー



図 5.3: SONY 5000 のシリアルナンバー

しかしながら、製品によって製造番号の付け方が異なることによる管理の難しさが挙げられる。また、異なる製品で、同じ製造番号がつけられること

もあるため、1. 一意である という要件を満たさないため、適さないと考えられる。

### 3. MAC アドレス

MAC アドレス (Media Access Control address) は、LAN 規格であるイーサネットにおいて、ネットワーク上で機器を識別するためにつけられる番号。

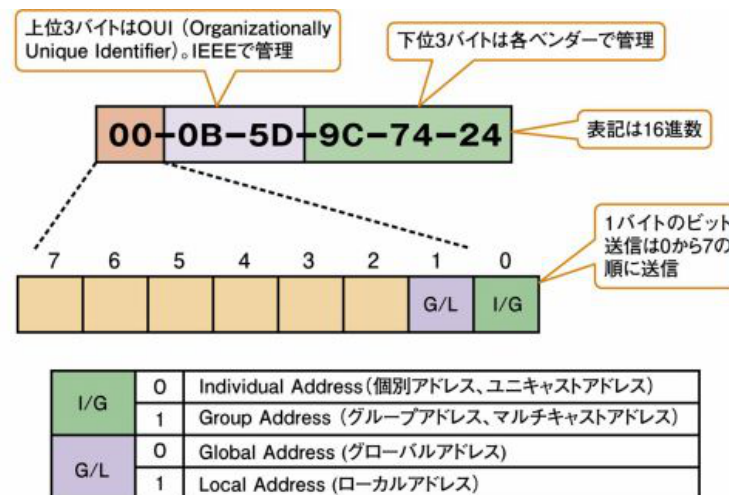


図 5.4: MAC アドレスのフォーマット

( 出典 : <http://ascii.jp/elem/000/000/417/417556/> )

図 5.4 のように、16 進数で書かれた 48bit の番号で、前半 24bit (3byte) の、IEEE が管理・割り当てをしている各メーカーごとに固有な番号と、後半 24bit (3byte) のメーカーが独自に割り当てる番号の組み合わせにより表現されている。原則として、世界中に同じ番号はない (一意である)。

IP アドレスと異なり、機器を別のネットワークにつなぎかえたとしても、原則 MAC アドレスは変わらない。図 5.5 を参照。

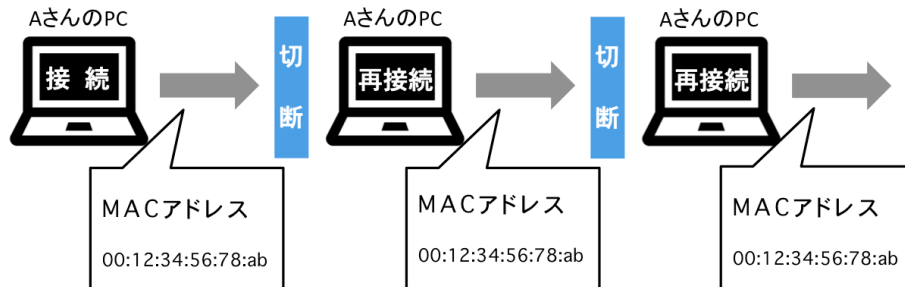


図 5.5: MAC アドレスのイメージ

MAC アドレスには、以下のルールが設けられている。

- 多くは 48 bit のコードから生成される。
- メーカーが出荷する段階で書き込む。  
(原則、ユーザがその値を変更することが出来ない。)
- 同じ MAC アドレスを持つ機器は存在しない。
- メーカーごとに固有な番号が組み込まれているため、メーカーを識別することが出来る。

MAC アドレスの個数は、先頭にある IG ビット/GL ビット分を除いた、2 の 46 乗 = 70,368,744,177,664 70 兆個 である。Internet of Things が進んだ 2020 年時点では、500 億個のモノが繋がるということが予想されているが、当面枯渇問題については考えなくて良い。

## 5.1.2 要件を満たす識別子の提案

### 識別子

IPv6 も条件を満たすことが出来る識別子であると言えるが、現状で使用されている 1. 一意である また、2. 滅多に変わらないの要件を満たす MAC アドレスを、本論文では用いることにする。

### Attribute

そして、図 5.6 のように。MAC アドレスの後ろに、それが何であるのかを示す Attribute (属性) をラベルとしてつけたものを、システムが識別するための識別子とすることにより、3. 「モノ」同士のやりとりを、人間がある程度理解することができるようになる。

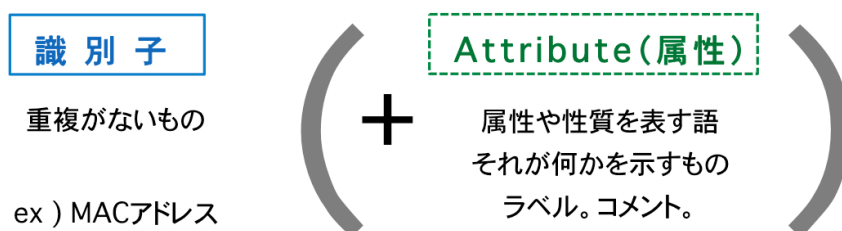


図 5.6: システム側の識別子、Attribute

Attribute の例としては、以下の様なものが考えられる。

- ロケーション
  - － 絶対名

\* 住所

例) 神奈川県横浜市港北区日吉4丁目1-1

4-1-1, Hiyoshi, Kohoku-ku Yokohama-shi, Kanagawa, Japan

\* 緯度経度(高度)

例) 35.552703,139.650877

北緯 [North] 35 ° 33 10 東経 [East] 139 ° 39 3

– 話者依存の相対名

例) リビング、トイレ、KMD、電車、実家、学校

● デバイス情報

– デバイス名

例) Macbook, VAIO, AQUOS

– デバイスのカテゴリ

例) パソコン、時計、冷蔵庫、電子レンジ

● 所有者

– 所有者が明確

例) 個人(今木彩翔)、会社

– 所有者が不明確

例) 公共物、建物内のもの、地域のもの

● ブランド名、ベンダ名

例) Apple, SONY, CASIO

● モデル

例) 2013 モデル、クリスマス限定モデル

## 5.2. 実装

### 5.2.1 目的

ユーザの指示する名前の補助を行う際や、システム同士の会話を人間がのぞき見て理解をするためには、その「モノ」がどんなものであるのかを読み取る必要がある。本項では、提案する全体像の中の一部である、Attribute (属性) の部分をシステム言語から、人間が理解することの出来る識別子への変換することについての実装を行う。

### 5.2.2 開発環境

実装環境は、パケットデータをパケットデータベースにインポートする際の実装言語として Ruby を使用した。ライブラリは Pcap を使用した。なお、事前に LinuxPC に Ruby をインストールし、Ruby/Pcap 拡張ライブラリのインストールを行った。以下、実装環境を表 5.1 にまとめて示す。

表 5.1: 開発環境

項目	種類
OS	Mac
開発言語	Ruby
ライブラリ	Pcap

### 5.2.3 属性情報の抽象化

表 5.2 のように、属性情報の抽象化を行った。左から右に名前解決を行う。

表 5.2: 属性情報のテーブル

テーブル (対応表)	パケットキャプチャ	実空間	名前
------------	-----------	-----	----

今回は、テーブルに入れるものとして以下の3点を使用し、それに対応するパケットキャプチャ上の情報を表 5.3 に示す。

- Location
- Device
- Owner

表 5.3: テーブルに入る情報

テーブル (対応表)	パケットキャプチャ	実空間	名前
Location	ネットワークサブネット	A	a
Device	ホスト名	B	b
Owner	MAC アドレス	C	c

Location は、パケットキャプチャではネットワークサブネットを、Device はホスト名、Owner は MAC アドレスを対応させる。その後、実空間の結果を名前と対応付けを行い、属性 (Attribute) として、テーブル (対応表) = 名前 とする。

例) #Location = a, #Device = b, #Owner = c

#は区切り記号として用い、順番についての決まりは設けない。一つ一つの#が属性情報として存在をする。

## 5.2.4 システム構成

本システムの構成は図 6.1 のとおりである。

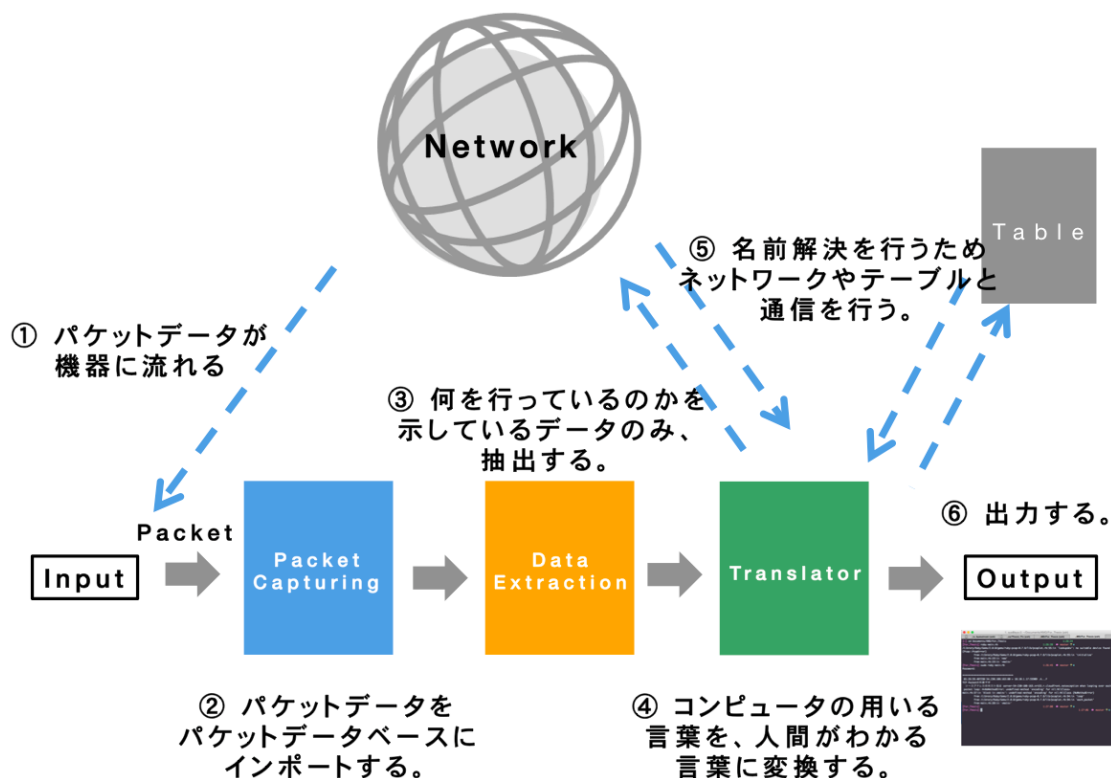


図 5.7: システム構成図

まず、パケットデータが kmd-hiyoshi のアクセスポイントを通じて各機器に流れる。そのパケットデータベースを取得する。取得したパケットデータの中で、何を行っているのかがわかるデータのみを抽出する。抽出したデータは、コンピュータ言語で書かれているので、それを人間が可読出来る言葉に変換を行う。その際に、ホスト名など名前解決を行うためネットワークや、名前の対応表 (Table) と通信を行う。その後、変換した結果をユーザーが可読出来る形式で出力する。



## パケットキャプチャ

kmd-hiyoshi のアクセスポイントに接続している機器の通信データをパケットデータベースにインポートするプログラムを Ruby で記述した。その際に、Ruby で Packet Capture Library を使うための拡張ライブラリである Ruby/Pcap 拡張ライブラリを用いた。

Ruby/Pcap 拡張ライブラリ (図 5.9) を用いても、tcpdump コマンド (図 5.8) と同程度の結果を出力することが出来た。

```
[For_Thesis] sudo tcpdump
tcpdump: data link type PKTAP
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on pktap, link-type PKTAP (Packet Tap), capture size 65535 bytes
19:55:51.252004 IP6 fe80::218d:ecec:895c:9821.57268 > ff02::1:3:llmnr: UDP, length 22
19:55:51.252007 IP6 fe80::218d:ecec:895c:9821.57771 > ff02::1:3:llmnr: UDP, length 22
19:55:51.252009 IP host196.kmd.keio.ac.jp.57771 > 224.0.0.252.llmnr: UDP, length 22
19:55:51.252028 IP host196.kmd.keio.ac.jp.57268 > 224.0.0.252.llmnr: UDP, length 22
19:55:51.461088 IP host196.kmd.keio.ac.jp.57772 > 239.255.255.250.upnp-discovery: UDP, length 624
19:55:51.461303 IP6 fe80::218d:ecec:895c:9821.57773 > ff02::c:upnp-discovery: UDP, length 624
19:55:51.461662 IP host196.kmd.keio.ac.jp.57772 > 239.255.255.250.upnp-discovery: UDP, length 624
19:55:51.565552 IP6 fe80::218d:ecec:895c:9821.57773 > ff02::c:upnp-discovery: UDP, length 624
19:55:51.565558 IP6 fe80::218d:ecec:895c:9821.54404 > ff02::1:3:llmnr: UDP, length 22
19:55:51.565581 IP host196.kmd.keio.ac.jp.54404 > 224.0.0.252.llmnr: UDP, length 22
19:55:51.565583 IP6 fe80::218d:ecec:895c:9821.63961 > ff02::1:3:llmnr: UDP, length 22
19:55:51.565775 IP host196.kmd.keio.ac.jp.63961 > 224.0.0.252.llmnr: UDP, length 22
19:55:51.672442 IP6 fe80::32f7:dff:fecl:bc40 > ff02::1: ICMP6, router advertisement, length 64
19:55:51.747575 IP6 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935 > tg-in-x75.1e100.net.https: Flags [P.], seq 1180808898:1180809372, ack 2859841191, win 4096, options [nop,nop,TS val 652103748 ecr 1979058900], length 474
19:55:51.747721 IP6 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935 > tg-in-x75.1e100.net.https: Flags [P.], seq 474:515, ack 1180809372, win 4096, options [nop,nop,TS val 652103748 ecr 1979058900], length 41
19:55:51.785452 IP6 tg-in-x75.1e100.net.https > 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935: Flags [P.], seq 474, win 435, options [nop,nop,TS val 1979119328 ecr 652103748], length 0
19:55:51.785455 IP6 tg-in-x75.1e100.net.https > 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935: Flags [P.], seq 474, win 435, options [nop,nop,TS val 1979119328 ecr 652103748], length 0
19:55:51.785747 IP6 tg-in-x75.1e100.net.https > 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935: Flags [P.], seq 1:42, ack 515, win 435, options [nop,nop,TS val 1979119329 ecr 652103748], length 41
19:55:51.785781 IP6 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935 > tg-in-x75.1e100.net.https: Flags [P.], seq 42, win 4094, options [nop,nop,TS val 652103785 ecr 1979119329], length 0
19:55:51.786980 IP6 tg-in-x75.1e100.net.https > 2001:200:167:2ec1:2cf0:6d9a:329a:2ee9.52935: Flags [P.], seq 42:102, ack 515, win 435, options [nop,nop,TS val 1979119330 ecr 652103748], length 60
```

図 5.8: tcpdump コマンドによる出力結果

```
[For_Thesis] sudo ruby ruby_pcap.rb 19:55:13 master
Password:
19:55:20.336344 131.113.136.131:58794 > 224.0.0.1:8612 len 24 sum 27868
19:55:21.067639 131.113.137.237:5353 > 224.0.0.251:5353 len 112 sum 28322
19:55:21.067643 Some packet
19:55:21.485484 Some packet
19:55:21.485487 Some packet
19:55:21.485488 131.113.136.196:54016 > 224.0.0.252:5355 len 35 sum 32101
19:55:21.485489 131.113.136.196:50888 > 224.0.0.252:5355 len 38 sum 42628
19:55:21.485528 Some packet
19:55:21.485529 131.113.136.196:53769 > 224.0.0.252:5355 len 41 sum 32385
19:55:21.485530 Some packet
19:55:21.485724 Some packet
19:55:21.485726 131.113.136.196:58351 > 224.0.0.252:5355 len 38 sum 16421
19:55:21.486078 131.113.136.196:49663 > 224.0.0.252:5355 len 35 sum 13950
19:55:21.486082 Some packet
19:55:21.486083 131.113.136.196:57583 > 224.0.0.252:5355 len 41 sum 21995
19:55:21.903185 Some packet
19:55:21.903188 Some packet
19:55:21.903189 131.113.136.196:50888 > 224.0.0.252:5355 len 38 sum 42628
19:55:21.903190 131.113.136.196:54016 > 224.0.0.252:5355 len 35 sum 32101
19:55:21.903223 Some packet
19:55:21.903224 Some packet
19:55:21.903461 Some packet
19:55:21.903464 131.113.136.196:49663 > 224.0.0.252:5355 len 35 sum 13950
19:55:21.903492 131.113.136.196:57583 > 224.0.0.252:5355 len 41 sum 21995
19:55:21.903716 Some packet
19:55:21.903719 131.113.136.196:58351 > 224.0.0.252:5355 len 38 sum 16421
19:55:21.904058 131.113.136.196:53769 > 224.0.0.252:5355 len 41 sum 32385
19:55:21.904062 131.113.137.49:50865 > 239.255.255.250:1900 len 105 sum 63866
19:55:21.904062 131.113.136.196:62446 > 239.255.255.250:1900 len 141 sum 24927
```

図 5.9: Ruby/Pcap 拡張ライブラリによる出力結果

## データ抽出

インポートしてきたデータのうち、その機器がなにを行っているのかがわかるデータの抽出を行った。今回は、以下のデータを抽出した。

- IP アドレス ( src, dst)

IP パケットの中身は図 5.10 のようになっている。

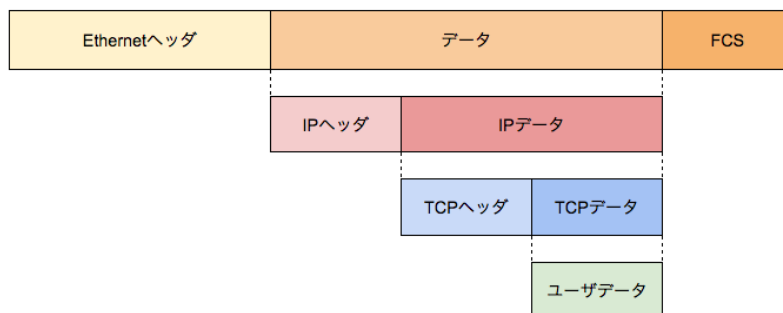


図 5.10: IP アドレスの中身

IP アドレスの送信元と送信先を取得することにより、何が、どこのネットワークに繋がるのかを探索することが出来る。

- ポート番号 (src, dst)

TCP/IP 通信において、コンピュータが通信に使用するプログラムを IP アドレスのみではわからないため、識別するためにポート番号を用いる。このポート番号を調べることで、どんな通信を行っているのかを把握することが出来る。

ポート番号は 0 番 ~ 65535 番までであるが、今回は代表的なポート番号を表 5.4 の通り変換を行った。

表 5.4: ポート番号 変換

変換前	変換後
22	SSH:SSH 通信をしています
25	SMTP:メールを送っています
53	DNS:名前を検索しています
80	HTTP:WEB を徘徊しています
110	POP : メールを受信しています
143	IMAP:メールを保存します
443	HTTPS:ヒミツの通信をしています
5353	mDNS:名前を決めています
その他	その他の宛先です

- 時間

各パケットデータ毎に時刻情報を取得した。

- SYN / ACK の表示

TCP 通信においては、通信許可要求スリーウェイハンドシェイクを行っている [32]。その通信を可視化するために、表 5.5 のように変換をするプログラムを記述した。

表 5.5: SYN / ACK 変換

変換前	変換後
SYN	接続してもいい？
ACK	接続していいよ！

### 5.3. 評価

本項では、第 5.2 項で述べたシステムの実行結果、評価を行う。

#### 5.3.1 評価概要

本評価は、本研究で提案した名前解決システムの有用性評価することを目的とする。今回の実験の際に使用した環境を表 5.6、5.7 に、記載する。

表 5.6: 動作させた環境

項目	名前
OS	Mac OS (Yosemite)
プロセッサ	1.4 GHz Intel Core i5
メモリ	4 GB 1600 MHz DDR3
ネットワーク	IEEE 802.11ac

表 5.7: ネットワーク環境

項目	名前
ネットワークに 接続された「モノ」	PC スマートフォン タブレット プリンタ

### 5.3.2 結果

#### ユーザ側の識別子

今回実装したシステムにおいては、ユーザ側の識別子を、各機器に名付けている名前とした。パケットデータの中から、ユーザの識別子を取得して表示を行った。実行結果は図 5.11 である。

ネットワークにアクセスする被験者に対し、識別子の付け方に制約を課さずに行った。

その結果、以下の様な名前の付け方が現れた、使用言語も日本語英語などが混在した。

- ニックネーム

例) iriesusumutar, ichigo

- 氏名

例) 入江晋太

- 苗字だけ

例) 野尻

- 名前だけ

例) Yukari, シュー

- 機器自体の名前も付随

上記 4 点の識別子に、機器の名称が付随したもの

例) HUI の iPhone, jimi'sMac

- 可読不可能な名前

例) 599178202, qbbcijmsqzvbe, PNJB

```

15:24:18.885968 131.113.137.121:5353 > 224.0.0.251:5353 len 103 sum 10392
  ソースアドレスのホスト名は host377.kmd.keio.ac.jp
  デスティネーションアドレスのホスト名は
224.0.0.251
UDP Packetの中身です:⓪
                                HUIのiPhone
                                デスティネーションポート番号は、5353
ソースポート番号は、5353
mDNS:名前を決めています
*****

```

図 5.11: 実行結果 機器の名前 1 【ニックネーム + 機器名】

また、5 分間の中で流れてきたユーザ側の識別子の分類を表 5.8 示す。同じ識別子であれば、カウントを行っていない。

表 5.8: 識別子の分類

項目	個数
文字フルネーム	1
ニックネーム	1
苗字だけ	
名前だけ	1
機器自体の名前も付随	
可読不可能な名前	
有効数計	30

本実験においては、図 5.12 のように、「野尻の Mac Book」という言葉を抽出することが出来た。これは、ユーザにとって可読可能な識別子であると言える。

```
*****
15:24:14.708080 131.113.136.94:5353 > 224.0.0.251:5353 len 132 sum 15739
  ソースアドレスのホスト名は host094.kmd.keio.ac.jp
  ディスティネーションアドレスのホスト名は
224.0.0.251
UDP Packetの中身です:野尻 の MacBookアスティネーションポート番号は、5353
ソースポート番号は、5353
mDNS:名前を決めています
```

図 5.12: 実行結果 機器の名前を取得

しかしながら、本実験においては可読可能な文字として取得できる数が少なかった。また、本実験においては名前付けのルールを設けずにやったため、言語や表示名の不統一が目立った。これは、ルールや人の制約を設けずに、アクセスポイントに流れるデータを取得したため、パソコンやスマートフォン、タブレット、



プリンタしか繋がっていなかった、また、実験を行った際にネットワークに接続をしていた人数が少なかったことも挙げられる。

一部取得することが出来た結果に関しては、ユーザが可読可能な識別子であると言えるため、たくさんの人がネットワーク通信を行ったり、「モノ」が繋がった上で、統一した表記方法の下行うと、今回の結果よりも有用な結果が取得することが出来ると考える。

### ホスト名の名前解決

本実験では、各ホストがどんな「モノ」であり、何を行っているのかを示すことを目的とした。図 5.13 は、IP アドレス「131.113.136.81」を、ユーザが可読可能な名前であるホスト名「hacking-studio-printer-mono.kmd.keio.ac.jp」に変換する名前解決を行った。

```
224.0.0.251
UDP Packetの中身です:_rfb_tcplocデスティネーションポート番号は、5353
ソースポート番号は、5353
mDNS:名前を決めています

*****
15:24:17.215085 Some packet

*****
15:24:17.215087 131.113.136.81:5353 > 224.0.0.251:5353 len 48 sum 40677
  ソースアドレスのホスト名は hacking-studio-printer-mono.kmd.keio.ac.jp
  デスティネーションアドレスのホスト名は
224.0.0.251
UDP Packetの中身です:
                                     _googlecast_デスティネーションポート番号は、5353
ソースポート番号は、5353
mDNS:名前を決めています

*****
15:24:17.215254 Some packet
```

図 5.13: 実行結果 IP アドレスの名前解決

これは、Hacking Studio<sup>1</sup> にあるプリンターがデータを送っている事がわかる。

図5.14は、KMDのネットワークから、IPアドレス(174.129.224.33)へHTTPS通信を行っているのを示している。通信先のポート番号443(システム側の識別子)とHTTPS通信(ユーザー側の識別子)の名前解決を行っている。

```
*****
15:24:13.761493 131.113.136.195:61159 > 174.129.224.33:443 .A...
TCP Packetの中身です
  ソースアドレスのホスト名は host195.kmd.keio.ac.jpその他の宛先です
  ディスティネーションアドレスのホスト名は ec2-174-129-224-33.compute-1.amazonaws.com

HTTPS:ヒミツの通信をしています データ部分の長さは、(0)
接続していいよ！
Window Sizeは、WIN 4094
1418797453.761493 66(0): *を文字列に変換 131.113.136.195: ソースポート番号は、 61159 ->
174.129.224.33: デスティネーションポート番号は、 443: TCPフラグフィールドの値を文字列として返しま
ず、.A....
(DF)

*****
15:24:13.977139 Some packet
```

図 5.14: 実行結果 行っている通信を表示

また、「174.129.224.33」は、図 5.15 のように、Dropbox [33] の IP アドレスである。Dropbox は、Amazon AWS を使用している [34] ため、パケットデータのアクセス先が「ec2-174-129-224-33.compute-1.amazonaws.com」と表示されていることがわかった。

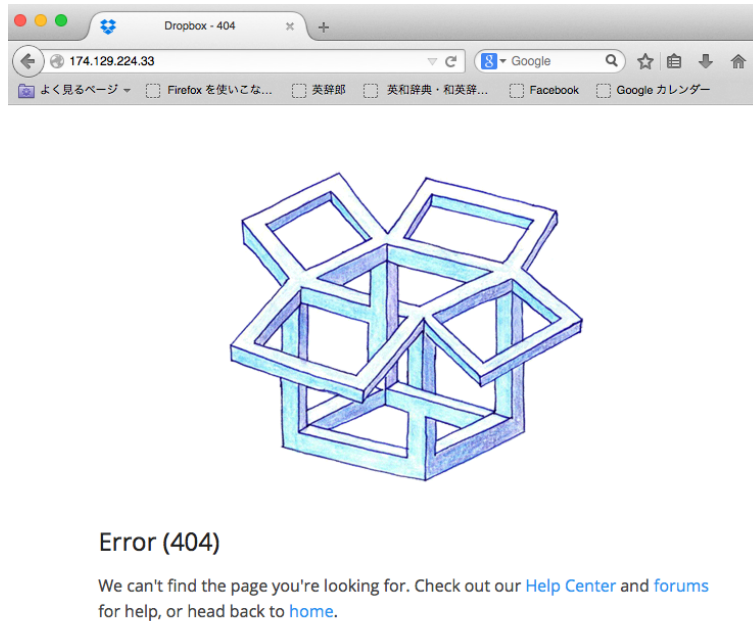


図 5.15: ブラウザから IP アドレス「174.129.224.33」へのアクセス結果

## ポート番号

図 5.16 のように、表 5.4 の変換を行うことが出来た。しかしながら、取り上げたポート番号の数が一部（表 5.4）だったため、図 5.17 のように何を行っているのかがわからない出力「その他の宛先です」が多数見られた。

```
*****
15:24:13.761161 174.129.224.33:443 > 131.113.136.195:61159 .AP...
TCP Packetの中身です
  ソースアドレスのホスト名は ec2-174-129-224-33.compute-1.amazonaws.com
HTTPS:ヒミツの通信をしています
  ディスティネーションアドレスのホスト名は host195.kmd.keio.ac.jpその他の宛先です
データ部分の長さは、(37)

接続していいよ！

Window Sizeは、WIN 70
```

図 5.16: 実行結果 ポート番号 1

```
*****  
15:24:14.081638 131.113.136.87:57910 > 224.0.0.252:5355  
  ソースアドレスのホスト名は host087.kmd.keio.ac.jp  
  デスティネーションアドレスのホスト名は  
224.0.0.252  
UDP Packetの中身ですfwpadデスティネーションポート番号は、5355  
ソースポート番号は、57910  
その他の宛先です  
  
*****
```

図 5.17: 実行結果 ポート番号 2

## 処理時間

一つのパケットがどのくらいの時間を要しているのかを調べるため、20 個のパケットに要した時間を図 5.18 に示す。また、平均時間は、0.223 秒となった。

	処理開始	処理終了	かかった時間
1	21:56:06.554	21:56:06.764	00:00:00.210
2	21:56:06.764	21:56:06.765	00:00:00.001
3	21:56:07.181	21:56:07.495	00:00:00.314
4	21:56:07.495	21:56:07.808	00:00:00.313
5	21:56:07.808	21:56:07.912	00:00:00.104
6	21:56:07.912	21:56:08.957	00:00:01.045
7	21:56:14.284	21:56:14.597	00:00:00.313
8	21:56:14.597	21:56:14.910	00:00:00.313
9	21:56:17.103	21:56:17.730	00:00:00.627
10	21:56:21.386	21:56:21.595	00:00:00.209
11	21:56:21.595	21:56:21.699	00:00:00.104
12	21:56:21.699	21:56:21.805	00:00:00.106
13	21:56:22.012	21:56:22.117	00:00:00.105
14	21:56:22.683	21:56:22.785	00:00:00.102
15	21:56:22.785	21:56:23.058	00:00:00.273
16	21:56:24.206	21:56:24.207	00:00:00.001
17	21:56:25.355	21:56:25.459	00:00:00.104
18	21:56:25.459	21:56:25.564	00:00:00.105
19	21:56:25.668	21:56:25.773	00:00:00.105
20	21:56:25.810	21:56:25.811	00:00:00.001

図 5.18: 一つのパケットに要した時間

### SYN / ACK の表示

SYN / ACK 通信データを取得したが、図 5.19 のように ACK についてしか検出することは出来なかった。

```

*****
15:24:13.684664 209.85.228.10:443 > 131.113.136.195:60357 .A....
TCP Packetの前身です
   ソースアドレスのホスト名は 209.85.228.10HTTPS: ヒミツの通信をしています
   ディスティネーションアドレスのホスト名は host195.kmd.keio.ac.jpその他の宛先です データ部分の長さは、
(1392)
接続していいよ！
Window Sizeは、WIN 1392
1418797453.684664 1458(1392): *を文字列に変換 209.85.228.10: ソースポート番号は、 443 ->
131.113.136.195: デスティネーションポート番号は、 60357: TCPフラグフィールドの値を文字列として返しま
す、.A....
(DF)
*****

```

図 5.19: 実行結果 SYN / ACK 情報

## 5.4. 考察

### 5.4.1 属性情報の取得について

今回行った実装では、表 5.9 のようなテーブルを作成した。実空間を名前に変換する際に、もう一つ対応表が必要となる。

表 5.9: 属性情報の変換

テーブル (対応表)	パケットキャプチャ	実空間	名前
Location	ネットワークサブネット	3f-south	Hacking Studio
Device	ホスト名	hacking- studio-printer- mono.kmd.keio.ac.jp	Printer
Owner	MAC アドレス	64:76:ba:93:63:7c	AkiraKato

例えば、Owner 部分については、慶應義塾大学メディアデザイン研究科では、校内のネットワークに繋げるために、MAC アドレスの登録をする必要がある。MAC アドレスと学生、教授の対応付けの表があるとこのように機能する。Location 部分については、慶應義塾大学メディアデザイン研究科が入っている協生館に 63 箇所 (1F:4 箇所, 2F:12 箇所, 3F:24 箇所, 4F:10 箇所, 5F:2 箇所, 6F:9 箇所, 7F:2 箇所)

設置されており、どこのアクセスポイントに接続されているかにより、その「モノ」がどこにいるのかを把握することが出来る。図 5.20 は、その「モノ」がどの場所に接続されているのかを示したものである。People 部分には、登録している人の大学メールアドレスの@以前が表示されている。

As of 1/23 16:40:08

Note: the information may not be accurate due to the following reasons:

- Information is updated in every 10 minutes.
- Information is based on WiFi access, and association may last extra minutes after the machine is shutdown or removed.
- Newly registered machine may not appear for a day or two.
- System may misinterpret laptop/cellphone left as the owner is still in Kyoseikan.

Area	People
Hiyoshi Campus	n.kokic3-racovet.xiao-stratmanu
Kyoseikan 2F-North	m.shirasaki@uia-p.sugita
Kyoseikan 2F-South	rlkonanqa
Kyoseikan 3F-North	hamamita@hiroso.i.nataku.i.wamoteyuta.jy.lacostardi.raiki.1118.mitsuya@maranishi.rdm.sogio noda.sakayan.taisaku.mina.tomoyuki28777@uim.vt.jpplus.yukishoji
Kyoseikan 3F-South	Sk-i-mi.yukita.angelia@oki-tera@staylor.charith@chibiko@disentle@niachina@api.h.bayakawa. haagou12@p.de.kodharu@hiroso3323@hiroso@lekuta.maria.julayan@kai.kam@yama.kurino@waraki kay@hori@koiko@kovin@an.kuni@v27@marcos@nartan@masato-p@nikke@satani@naitaki@natsuki21@ mi@anqun@nishi@yama@riya.kahir@pohiang@rdm.sogio@ryan.ad@91@samuel@san@shoukou@k@tamoki tama@uia@toyama@xurion
Kyoseikan 4F	janaki@k@natsuyama@makosacke@uim@toyama
Kyoseikan 6F-South	3@osode@u@arima@ayaka.j@hachibetsu79@uim.s.0009@k@naitomi@kato@koma@megumi@fujikawa@e- dei@sei@system@tachi@yamamura@yukari@924

図 5.20: 加藤朗教授が作成した distribution map

このような対応付けの表があれば、Location や Owner、Device などの属性情報を取得することが出来ると考えられる。

## 5.4.2 名前解決アルゴリズムの計算時間

名前解決アルゴリズムの計算時間が、どのくらいの時間以内に来る必要があるのかを検討するにあたり、以下の2点を参考にする。

- 人間がインターネット接続時に、ストレスを感じる待ち時間
- システムが、Timeout 処理を行う待ち時間

これらの時間から、どのくらいの時間であれば条件を満たすのかを検討する。

### 1. 人間がインターネット接続時に、ストレスを感じる待ち時間

図 5.21 はページの表示にどのくらい時間がかかると、「遅い」と感じるのか [35] を表したグラフである。

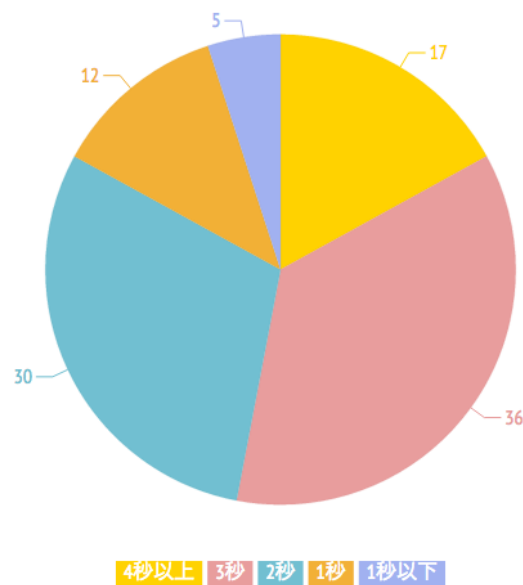


図 5.21: ページの表示にどれくらい時間がかかると「遅い」と感じるのか。調査対象 1048 名  
( [www.akamai.com/2seconds](http://www.akamai.com/2seconds) のデータを元に作成 )



このグラフから、2秒で約半数の47%、3秒で83%の人が「遅い」と感じている事が読み取れる。よって、インターネット接続時に、ストレスを感じる待ち時間は「2秒」であるとする。

## 2. システムが、Timeout 処理を行う待ち時間

- ブラウザがタイムアウト処理を行う時間は、Firefox のデフォルト値が 300 秒 (=5 分) となっている。(図 5.22)

network.ftp.idleConnectionTimeout	初期設定値	300	整数値
network.http.response.timeout	初期設定値	300	整数値
network.http.pipelining.read-timeout	初期設定値	30000	整数値

図 5.22: Firefox about:config 設定画面

- サーバ側で動作する Java アプリケーションを作成する Servlet [36] のデフォルトのセッション有効期限は 1800 秒 (30 分)。

以上 a,b より、人間が関わってくるシステムに関する名前解決アルゴリズムの計算時間は約 2 秒以内であれば妥当であると言えると考えられる。

今回の実装については、平均 0.223 秒であるため、上記条件を満たすと言える。

しかしながら、今回は特定の「モノ」が一つの相手と通信する速度であったため、様々な「モノ」が相互に通信する際にはより多くの時間がかかると考えられる。

## 5.5. まとめ

システムの言葉を、人間が理解することが出来る言葉へ変換する実装を行った。今回はネットワーク上に繋がった「モノ」がパソコンやスマートフォンであったが、人間が理解できる言葉へ変換することが出来たと考えられる。また、ネットワークに繋がる「モノ」に対して、ネットワークに流れる情報から属性情報 (Attribute) 取得することが有効そうであることを確認した。しかしながら、本実験においては可読可能な文字として取得できる数が少なく、表記方法の不統一が目立っ

た。今後は、ユーザ側の識別子の表現に関するルールをつけた上で、ネットワークに様々な「モノ」を繋ぎ実験を行うこと、そして対応付けのテーブルを充実させることが求められる。

## 第6章

# ユーザが指示するための名前とその 補助情報

本章では、ユーザが指示する名前だけではユーザが意図する結果を表示することが難しい場合が考えられるので、意図する結果を導き出すことのできる補助情報の検討を行う。

### 6.1. 設計

ユーザが指示する名前と補助情報に求められる機能は以下の2つである。

1. ユーザが名前を表す対象を、具体的にイメージを出来る
2. 重複が許されている

今回、名詞に Attribute を付与したものをユーザが指示するための識別子として提案する。Attribute の部分に入るものとして以下のものが考えられる。

- デバイス名
  - 照明、テレビ、携帯、時計
  - LED、電気、iPhone、カシオ
- ロケーション
  - 神奈川県横浜市港北区日吉4丁目1-1

- 4-1-1, Hiyoshi, Kohoku-ku Yokohama-shi, Kanagawa, Japan
- 35.552703,139.650877
- 北緯 [North] 35 ° 33 10 東経 [East] 139 ° 39 3
- リビング、トイレ、KMD、電車、実家、学校

- 天候

- 晴れ、雨、曇
- 暑い、寒い
- 23 、湿度 40%

- 時刻

- 2014/12/21 14:25
- 朝、日中、夕方、夜、深夜

- 使用者

- 個人名：今木彩翔
- 所属：会社や学校

- 所有者

- 個人：今木彩翔、会社
- 公共物、建物内のもの、地域のもの

- ブランド名、ベンダ名

- Apple, SONY, CASIO
- MAC アドレスの最初の 24 ビット (3 バイト)

- 色

- blue, red, yellow

- #0000FF, #FF0000, #FFFF00

- 大きさ

- 40cm, 1m
- 身長くらい、手くらい

- モデル

- 2013 モデル、クリスマス限定モデル

- ニックネーム

- CHIBI、ステファニー、ayapod

上記をカテゴリにわけ、以下の Attribute をユーザが指示する名前の修飾語として提案する。

1. 状況に関する情報

- ロケーション

- 絶対名

- \* 住所

- 文字列: 神奈川県横浜市港北区日吉4丁目1-1 4-1-1, Hiyoshi, Kohoku-ku Yokohama-shi, Kanagawa, Japan

- \* 緯度経度 (高度)

- 数値: 35.552703, 139.650877

- 北緯 [North] 35 ° 33 10 東経 [East] 139 ° 39 3

- 話者依存の相対名

- 例) リビング、トイレ、KMD、電車、実家、学校

- 天候

- 数え上げ型: 晴れ、雨、曇
- 相対値: 感情: 暑い、寒い

- 数値：23 、湿度 40%

## 2. ユーザに関する情報

- 時刻

- 絶対値：数値：2014/12/21 14:25
- 相対値：数え上げ型：朝、日中、夕方、夜、深夜

- 使用者

- 個人名：今木彩翔
- 所属：会社や学校

## 3. 「モノ」に関する情報

- デバイス名

- 文字列：照明、テレビ、携帯、時計
- 文字列：呼び名：LED、電気、iPhone、カシオ

- 所有者

- 個人：今木彩翔、会社
- 公共物、建物内のもの、地域のもの

- ブランド名、ベンダ名

- 文字列：Apple, SONY, CASIO
- 数値：MAC アドレスの最初の 24 ビット (3 バイト)

- 色

- 文字列：blue, red, yellow
- 数値：#0000FF, #FF0000, #FFFF00

- 大きさ

- 絶対値 (数値): 40cm, 1m
- 相対値：身長くらい、手くらい

- モデル
  - 文字列：2013 モデル、クリスマス限定モデル
- ニックネーム
  - 文字列：CHIBI、ステファニー、ayapod

これらの修飾語と、一般名詞のユーザが指示する名前から、システムが識別するための識別子へ変換する名前解決の手法としては以下の3つが考えられる。

1. 図 6.1：ユーザの指し示す名前がはっきりしたものであれば、対応するシステムが識別するための識別子に変換。  
例)「慶應義塾大学 協生館 3F の 01 教室の照明全部を消して」など、はっきりした説明から一意に特定する。

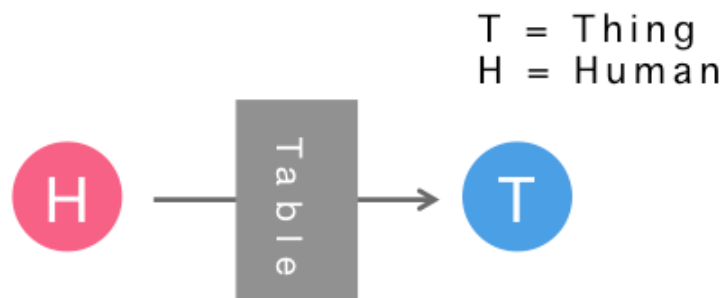


図 6.1: ユーザの指示する名前が明確である場合

2. 図 6.2：はっきりした名前でなければ、Location, Owner, Device といった Attribute から対応するシステムが識別するための識別子に変換。  
例) Attribute 情報を補助情報として付与することにより、テーブルの中から対応するものを絞り込み特定する。

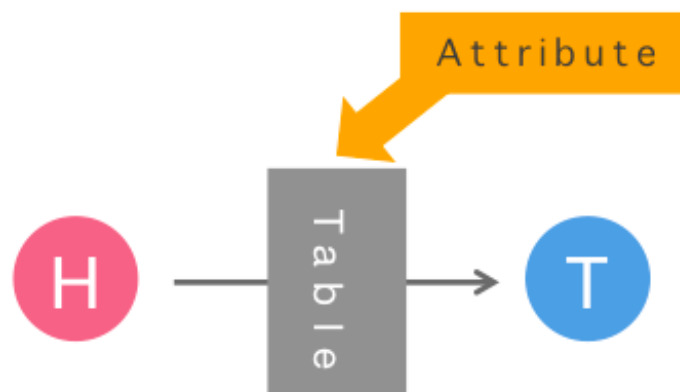


図 6.2: Attribute から特定する

3. 図 6.3 : いくつか候補が出て来てしまう曖昧な場合であれば、可能性の高い順番からユーザに返し、やりとりの中で特定を行うインタラクティブな変換。

例) いくつかの候補の中から「もしかしてこの照明ですか?」ということユーザに返し、ユーザとのやりとりの中から特定をする。

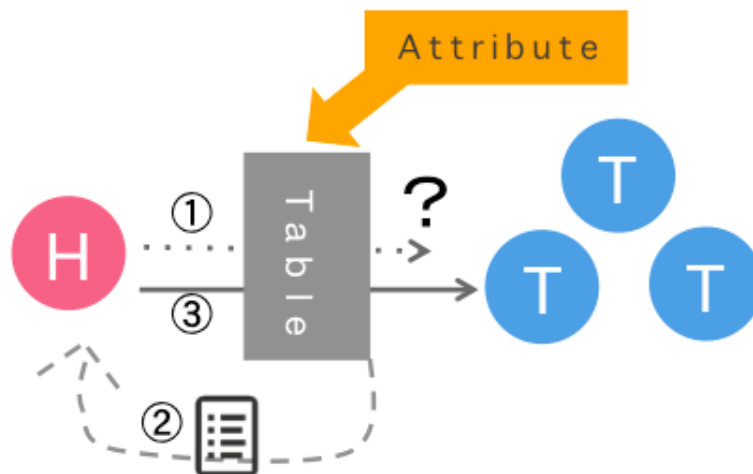


図 6.3: ユーザとのやりとりの中でインタラクティブに特定する



また、6.1節で述べたロケーションの中にある、絶対名と話者依存の相対名2つの表記方法を結びつけるテーブルをそれぞれに用意する必要がある。

## 6.2. 利用イメージ

図 6.4 に、特定が行われる際の利用イメージを示す。

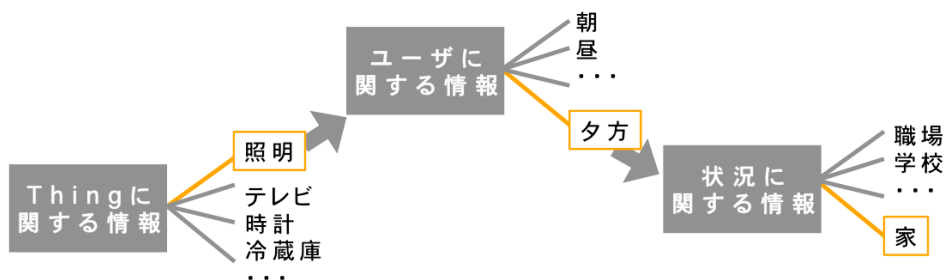


図 6.4: 特定のイメージ

例えば、ユーザが指示する名前のなかから「モノ」に関する情報であるデバイスの情報を参照する。そして、照明に紐づく情報のテーブルを呼び出す。その中で、ユーザのコンテキストを考慮して、テーブルを引っ張ってきて、システムが識別するための識別子への変換を行う。

図 6.5 におけるユーザが指示する名前の補助情報のイメージを図 6.6 に示す。

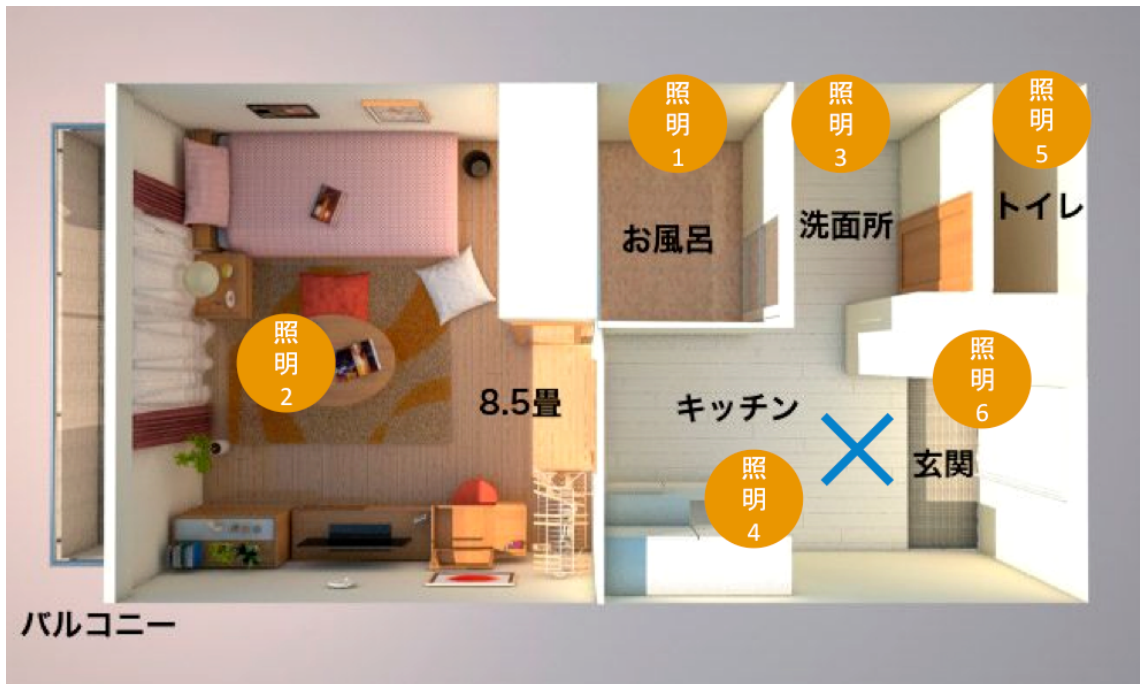


図 6.5: 例) 家の中にある「モノ」の配置

ユーザが図中の×印にいるときに、「照明をつけて」といった際に、システム側は、「照明」に当てはまる候補を6つ出してくる。そして、その中からユーザの位置情報や、利用履歴（使用頻度）を元に、ユーザの近くにある候補から順にユーザに対して「もしかしてこの照明ですか？」と返す。そして、そのやりとりの中でユーザの指示する「照明」の特定を行う。

Device	照明1	照明2	照明3	照明4	照明5	照明6
Location	35.552703,139.650877	35.552703,139.750878	35.552703,149.650879	35.452703,139.650880	33.552703,139.650881	33.552703,139.650882
Owner	Ayakalmaki	Ayakalmaki	Ayakalmaki	Ayakalmaki	Ayakalmaki	Ayakalmaki
Manufacturers	Panasonic	Panasonic	Panasonic	Panasonic	Panasonic	Panasonic
Date	20141101	20130129	20141101	20141010	20141101	20141101
Color	White	White	White	White	White	White
Size	20cm	1m	20cm	30cm	20cm	20cm
Model	2014model	2013	2014	standard	standard	standard
Machine Readable	7C:1B:3D:FD:65:9C	4B:1B:3D:F4:65:9C	64:76:ba:93:63:7c	F1:D2:0C:AA:03:00	7C:1B:3D:FD:65:9C	64:76:ba:93:63:7c

図 6.6: テーブル作成の試行実験

## 6.3. 考察

### 6.3.1 ユーザ側の名前について

5.2 項で行った実装の結果、ユーザ側が各「モノ」につける名前を、他のユーザが可読することが可能であると言える。しかしながら、今回の環境にはネットワークに照明や冷蔵庫、窓などが繋がっていないため、それらの「モノ」につけられる識別子が可読可能であるかは検証できていない。また、今回は機器が限られてはいたが、表 5.8 のように可読不可能なものがほとんどだったため、ユーザ側に対し言語やルールの制約が一切ない状態で自由に名前をつけられるようにすると、機能することが難しいと考えられるので、言語や表記手法のルールを付ける必要がある。

### 6.3.2 補助情報について

- ユーザが指示する名前をシステムが識別するための識別子に変換する際の補助情報の検討を行った。その中で、同じ情報に対しても、客観的に判断することが出来る表現方法（絶対値）と、客観的な判断では人によりズレてしまう表現方法（相対値）が見られた。ユーザが意図する結果を返すためには、絶対値と相対値を結びつけるためのテーブルを別途用意する必要がある。また、誰にでも適用可能なよう平均化したテーブルにするのか、ユーザ毎にテーブルを作成をするのかを検討する必要がある。
- 試行実験では、室内にある複数の同一名称の「モノ」に対して、位置情報や行動履歴を用いることにより、特定を出来るよう絞り込みを行うということ考えた。しかしながら、実際現状のセンサから室内の位置情報を正確に読み取ることが難しいため、性能向上や各々の「モノ」自身が相手の位置情報を取得する機能の実装が求められる。

## 第7章

# 結 論

本研究では、Social Things における基礎基盤となる通信相手を特定するための名前と識別子、そして名前解決についての提案を行った。インターネットに「モノ」が繋がることにより、「モノ」と人という単一の繋がりだけではなく、人を介さない「モノ」と「モノ」のコミュニケーションが誕生し、より高度なサービスの実現が試みられている。Internet of Things のコンセプトは、「モノ」と「モノ」が「なにか」を行っている世界であり、Social Things では、その「なにか」をのぞき見たり、人間の言葉でコミュニケーションをすることが求められる。相互にコミュニケーションを行う上で重要なことは、「通信相手を識別すること」である。しかしながら現状は、各ベンダのルールに基づいた識別子が付与されているため、そのドメインを超えた通信は困難である。また、ユーザに難解な識別子を覚えるということの難しさも問題点として挙げられる。

本研究ではそれらの課題を解決するために、ユーザが指示する名前とシステム側の識別子、そしてそれらの名前解決についての提案を行った。そして、システム側の識別子とユーザ側の識別子の対応付けに関する基礎的な実験を行い、有用性を確認した。

今後 Social Things の世界を実現するにあたり、提案するシステムモデルの実施を行い、「モノ」との対応付け、ユーザによる実証実験による精度の向上が求められる。

## 今後の課題と展望

### 名前解決の精度向上

今後の基盤システムにおける識別子の課題として、ユーザが指示する「モノ」の特定に対する制度の向上が挙げられる。そのためには、ユーザのコンテキストを高度に理解する必要が求められる。シチュエーションセンサなどから、ユーザの環境情報を読み取る精度の向上、「モノ」自体が自身の情報を発信して、受け取る精度の向上がまず挙げられる。また、ユーザがどの「モノ」を指し示すのかを判断するにあたり、ユーザの利用履歴や使用頻度を各ユーザ毎に保持する必要がある。

### データの保持

各ユーザの生活圏が変わる度に、テーブルが生成されるので、それを保持するための領域、そしてコストの検討も挙げられる。また、例えば「暑い、寒い、小さい、大きい」などユーザの主観的な感情と、システムが判断できる数値への対応付けが求められる。ユーザ毎の客観情報を保持するのか、平均化した情報を用意に対応させるのかの検討も求められる。

様々な「モノ」が人を介さずとも相互にデータの共有が行われることによる、ビッグデータの蓄積により、コンテキストを「モノ」が判断することができるようになると思う。「モノ」と人がより有機的な繋がりを持つことで広がる世界をつくるために、識別子だけでなくセキュリティやネットワーク、記憶領域などが一体となり共有、管理、分析方法を検討していくことが必要である。

# 謝 辞

本研究の指導教員であり、幅広い知見からの確な指導、そして入学前から暖かい励ましやご指摘をしていただきました慶應義塾大学大学院メディアデザイン研究科の砂原秀樹教授に心から感謝いたします。

研究指導や多大なるご助言をいただきました慶應義塾大学大学院メディアデザイン研究科の稲蔭正彦教授に心から感謝いたします。

いつも温かいご助言をいただきました慶應義塾大学大学院メディアデザイン研究科の岸博幸教授に心から感謝いたします。

研究の方向性について悩んでいる時に、様々な助言やとろけるご指導をいただきました慶應義塾大学大学院メディアデザイン研究科の加藤朗教授に心から感謝いたします。

慶應義塾大学大学院メディアデザイン研究科の古川 享教授に心から感謝いたします。大変豊富な心躍る経験談や、最新のガジェット話、そして貴重な機会を与えてくださり、ときめく瞬間がたくさんありました。

世界各地から数多くの励ましや助言を賜りました慶應義塾大学大学院メディアデザイン研究科の山内正人特任助教授に心から感謝いたします。

そして、時には支え合い、共に研究を進めて来た同期である小川景子さん、北本亜由美さん、國友美希さん、豊島有紀さん、野尻梢さんをはじめとする、Network Media Project の皆様に心から感謝いたします。

本研究のテーマである Social Things のチームメンバー、スポンサーである Ericsson 社に心から感謝いたします。

最後に、遠くから暖かくいつも見守り、味方でいてくれる母、最終的に大学院で学ぶ応援をしてくれた父、大切な家族に深く感謝いたします。ありがとう。

# 参 考 文 献

- [1] Panasonic. Panasonic スマート家電. <http://panasonic.jp/pss/ap/>.
- [2] Tesla Motors. Tesla motors. <http://www.teslamotors.com/>.
- [3] 日経コンピュータ. すべてわかる IoT 大全. 日経コンピュータ, 2014.
- [4] Michael Chorost. The networked pill. <http://www.technologyreview.com/biomedicine/20434/?a=f>, MIT Technology Review, mar 2008.
- [5] Dave Evans. The internet of things how the next evolution of the internet is changing everything. Technical report, Cisco Internet Business Solutions Group, apr 2011.
- [6] 象印マホービン株式会社. みまもりほっとライン i-pot. <http://www.live-e.org/>.
- [7] Ericsson. User experience lab blog. <http://www.ericsson.com/uxblog/>.
- [8] 落合秀也, 江崎浩. インプレス標準教科書シリーズスマートグリッド対応 IEEE1888 プロトコル教科書. インプレスジャパン, 2012.
- [9] CDN とは. <http://www.kobelcosys.co.jp/column/itwords/18/>.
- [10] UPnP DLNA. [http://www.foobar2000.org/components/view/foo\\_upnp](http://www.foobar2000.org/components/view/foo_upnp).
- [11] <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Introduction.html>.



- [12] Amazon S3. Amazon s3. <https://aws.amazon.com/jp/s3/>.
- [13] IBM. <http://www-06.ibm.com/systems/jp/power/solutions/watson/>.
- [14] 松田卓也. 2045年問題 コンピュータが人類を超える日. 廣済堂出版, 2012.
- [15] Watson. Watson. <http://www-03.ibm.com/systems/power/solutions/watson/index.html>.
- [16] 田澤孝之. はじめての Hadoop 分散データ処理の基本から実践まで. 技術評論社, 2012.
- [17] 日立製作所 - 白くまくん. <http://kadenfan.hitachi.co.jp/ra/shirokuma/>.
- [18] i tamahome. <http://www.tamahome.jp/i-tamahome>.
- [19] タマホーム株式会社 プレスリリース. <http://www.tamahome.jp/qkeioebqx4/wp-content/uploads/2013/03/news20130325.pdf>.
- [20] Apple Store Philips Hue. <https://itunes.apple.com/jp/app/philips-hue/id557206189?mt=8>.
- [21] NXP Laboratories UK. Zigbee light link user guide. [http://www.nxp.com/documents/user\\_manual/JN-UG-3091.pdf](http://www.nxp.com/documents/user_manual/JN-UG-3091.pdf).
- [22] 内閣官房 マイナンバー制度. <http://www.cas.go.jp/jp/seisaku/bangoseido/>.
- [23] Cyber government online. [http://cgs-online.hitachi.co.jp/contents/288\\_1.html/](http://cgs-online.hitachi.co.jp/contents/288_1.html/).
- [24] 田村奈央. Dnsの本質を知るならまず「ルート」から始めよ (特集インターネットの根幹 dns ルートサーバーの秘密). 日経 network, pp. 55-57, 2013.
- [25] IT用語辞典バイナリ. Rnsとは. <https://realnames.com/>.

- [26] RealNames. Realnames. <http://www.sophia-it.com/content/RNS>.
- [27] Google. [www.google.com](http://www.google.com).
- [28] Google. <https://www.google.com/insidesearch/howsearchworks/thestory/>.
- [29] Google. <https://www.google.co.jp/intl/ja/insidesearch/howsearchworks/crawling-indexing.html>.
- [30] Google. <https://www.google.co.jp/intl/ja/insidesearch/howsearchworks/algorithms.html>.
- [31] Google Japan Blog. <http://googlejapan.blogspot.jp/2012/12/blog-post.html>.
- [32] aico, 株式会社ディレクターズ, 村井純. 小悪魔女子大生のサーバエンジニア日記ーインターネットやサーバのしくみが楽しくわかる. 技術評論社, 2011.
- [33] Dropbox. <https://www.dropbox.com/>.
- [34] Arik Hesseldahl. Look out box and dropbox here comes amazon 's zocalo. In *Re/code*, oct 2014.
- [35] Akamai Technologies Inc. ecommerce web site performance today an updated look at consumer reaction to a poor online shopping experience. Technical report, Akamai, 2009.
- [36] Servlet. <http://www.javadrive.jp/servlet/>.