

修士論文 2012年度 (平成24年度)

XMPPによる個人識別に基づく
センサー情報のハンドリング

慶應義塾大学大学院 政策・メディア研究科

佐藤弘崇

XMPP による個人識別に基づく センサー情報のハンドリング

近年ネットワークにつながっているデバイスは増加を続けており、その種類も増加している。また、それぞれのデバイスに搭載されているセンサーやアクチュエータも数、種類共に増加傾向にあり、実世界の情報をより詳細に、迅速に取得することが可能となった。しかし、センサーデバイスはそれぞれ別個のサービスに紐付けられていることが多く、身近になったセンサーデバイスから情報を取得・活用しようとしても、外部から柔軟かつリアルタイムに取得することができない、という問題がある。

本論文ではこの問題の解決のため、デバイスと人間とを結びつけることを中心に考え、個人にとって扱いやすい、ネットワークにつながったデバイスを統合的に扱う拡張可能なアーキテクチャを提案した。具体的にはネットワークに接続されたデバイスを個人と結びつけることによって、取得した情報を個人が所有できるシステムをチャットプロトコルである XMPP に拡張を施したシステムを構築し、動作検証を行った。これによって、デバイスとユーザがエンティティとして同一でフラットな環境と対話するアーキテクチャを提供できる可能性を示した。

キーワード:

1. XMPP, 2. 識別, 3. センサー, 4. インターネット

慶應義塾大学大学院 政策・メディア研究科

佐藤弘崇

Handling sensor data based on personal identification by XMPP
--

In recent years, the number and diversity of devices which are connected to a network has grown. The number of sensors and actuators which are installed on them has also increased. This enables us to know more about what is happening in the real world in real-time. In most cases, however, devices are tied to specific services. This limits how the data can be obtained from the sensor devices in real-time.

This thesis proposes a unified, extensible architecture which centers on association between devices on a network and humans. We verified this architecture's feasibility by constructing a system which enables users to own data from sensor devices or obtain ownership of actuators by associating devices and humans by extending XMPP. This indicates a possibility of making a flat architecture in which devices and users occupy the same level and can talk to each other directly.

Keywords :

1. XMPP, 2. Identification, 3. Sensor,4. Internet

Keio University Graduate School of Media and Governance

Hiroataka Sato

目次

第1章 序論	1
1.1 背景	1
1.2 目的	2
1.3 アプローチ	2
1.4 本論文の構成	3
第2章 関連技術	4
2.1 既存のセンサーネットワークのモデル	4
2.1.1 デバイスの特性	4
2.1.2 通信モデル	4
2.1.3 モビリティ	5
2.1.4 空間	6
2.1.5 時間	6
2.2 データの所有権	7
2.3 センサーデータ共有技術	7
2.3.1 XMPP	7
2.3.2 SOAP	9
2.3.3 REST	10
2.3.4 MQTT	11
2.3.5 センサーネットワーク向けのサービス	11
2.3.6 まとめ	12
第3章 問題定義とアーキテクチャ要件	14
3.1 問題の定義	14
3.1.1 人間の社会とネットワークにつながったデバイスの関係の現状	14
3.1.2 デバイスのネットワークにおけるデータ所有権	15
3.2 デバイス所有権のダイナミックな切替	16
3.3 データの共有	17
3.4 アーキテクチャ要件	17
3.4.1 デバイスを使用する際の権限	18
第4章 提案アーキテクチャ	20
4.1 デバイスの擬人化	20

4.1.1	デバイスのインターフェースを擬人化による利点の例	21
4.2	方針	21
4.3	仕様	24
4.3.1	デバイスの分類	24
4.3.2	デバイスへのユーザーの権限	25
4.3.3	デバイスの種類とデバイスへのユーザーの権限の種類に対応	26
4.3.4	クライアント間の通信	26
4.3.5	データ送信	28
4.3.6	アクション	28
第 5 章	実装	33
5.1	周期起動型イベント向けセンサー	33
5.2	散発型イベント向けセンサー	35
5.2.1	ユーザークライアント	37
5.3	アクチュエータ	39
5.4	XMPP サーバ	40
第 6 章	評価	43
6.1	実現可能性	43
6.1.1	周期起動型イベント向けセンサー	43
6.1.2	散発型イベント向けセンサー	44
6.1.3	アクチュエータ	44
6.2	規模性	45
6.2.1	ユーザーの増加によるセンサー XMPP クライアントからの Stanza の増加	45
6.2.2	デバイスに付属しているセンサーの増加	49
6.3	汎用性	50
6.3.1	考察	51
第 7 章	結論	52
7.1	まとめ	52
7.2	今後の展望と課題	52
謝辞		54
付 録 A	本提案システムにおける Stanza 例一覧	58
A.1	人間とデバイスの結びつけに関する Stanza	58
A.2	権限に関する Stanza	59
A.3	データ送信に関する Stanza	61
A.4	アクションに関する Stanza	61

目 次

2.1	多対一の通信モデル	5
2.2	一对多の通信モデル	5
2.3	多対多の通信モデル	6
2.4	XMPP におけるメッセージの動き	8
2.5	SOAP のメッセージのやりとり	10
2.6	JSON のメッセージのやりとり	11
2.7	cosm 上の放射線線量計	12
3.1	人間の社会とネットワークにつながったデバイスの関係の現状	14
3.2	デジタルサイネージ型の自動販売機	15
3.3	公衆に設置された健康機器の例:体重計	16
4.1	保育者による保育施設に設置されたカメラへのアクセスコントロール	20
4.2	擬人化されたインターフェースによる保育施設に設置されたカメラへのアクセスコントロール	21
4.3	システム概要	23
4.4	ユーザとデバイスの結びつけ	23
4.5	ユーザとデバイスアカウントとの結びつけの破棄	24
5.1	Mini スペクトルメーター TA100	33
5.2	FeliCa リーダー RC-S310/ED2	34
5.3	周期起動型イベント向けセンサークライアントシステム全体図	35
5.4	周期起動型イベント向けセンサークライアント詳細図	36
5.5	センサークライアントとしての MacBook	36
5.6	体組成計 UC-411PBT-C	37
5.7	散発型イベント向けセンサークライアントシステム全体図	38
5.8	散発型イベント向けセンサークライアント詳細図	39
5.9	ユーザークライアントとしての Xserve	39
5.10	ユーザークライアント詳細図	40
5.11	アクチュエータクライアントシステム全体図	41
5.12	アクチュエータクライアント詳細図	41
5.13	XMPP サーバとしての Mac Mini	42
6.1	12/31 8:00-9:00 の空間放射線量	44

6.2	MUC 非使用時のクライアント結びつけ時のユーザ増加に伴う Stanza の増加	46
6.3	MUC 使用時のクライアント結びつけ時のユーザ増加に伴う Stanza の増加	47
6.4	MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較	47
6.5	MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較	48
6.6	サーバー側の MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較	49
6.7	MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較	50
6.8	Stanza のサイズによる処理時間の変化	50

目 次

2.1	XEP-0045 における各ユーザの権限の違い	9
2.2	各プロトコルの比較	13
4.1	デバイスの種類とユーザーの権限の種類に対応	26
4.2	Limited であることの通知	27
4.3	権限一覧要求	28
4.4	権限の一覧応答	28
4.5	ユーザー権限の owner への変更	29
4.6	ユーザー権限の変更要求	29
4.7	計測データの送信	30
4.8	デバイス固有の機能の一覧要求	30
4.9	デバイス固有の機能の一覧の応答	31
4.10	アクチュエーションの実行	31
4.11	アクチュエーションの結果	32
5.1	テクノエーピー Mini スペクトルメーター TA100 詳細	34
5.2	ソニー株式会社 Felica リーダー RC-S310/ED2 詳細	34
5.3	FeliCaID 読み込み実装実行環境	35
5.4	エー・アンド・デイ 体組成計 詳細	37
5.5	体重計連携および XMPP クライアントアプリケーション実行環境	38
5.6	ユーザークライアントプログラム実行環境	38
5.7	Openfire 実行環境	41
6.1	フレンド及びアンフレンドリクエストの成功率	43
6.2	2名による重量の測定と実測値との差異	44
6.3	アクチュエーションの成功率	45
6.4	パブリック XMPP サーバに使用されているサーバ実装	51
6.5	既存 XMPP サーバ・サービスにおける動作検証	51

第1章 序論

1.1 背景

ネットワークにつながったデバイスの増加

近年ネットワークにつながったデバイスは増加を続けている。Cisco社の予測によると、2020年までに500億台のデバイスがインターネットに接続されるとされる[1]。これら台数の増加には旧来のデスクトップ型PCに加えて、近年増加しつつある携帯電話、スマートフォンなどのモバイルデバイス、さらにはセンサーデバイスなどが挙げられる。これらの爆発的な台数の増加はInternet Of Things、あるいはInternet Of Everythingと呼ばれている。ネットワークにつながったセンサーの増加により、大量の実世界のデータがリアルタイムに取得できるようになる世界ができつつあると言える。この世界ではデバイスの数のみならず、デバイスの種類そのものの増加も見られる。具体的にはこれまでインターネットにつながっていなかった家電、車、センサーとしてのモバイル端末が挙げられる。特にネットワークにつながったセンサーの例としては温度計を始め、放射線量計などがあり、その他様々な種類のセンサーが時々刻々とデータをインターネット上に送り続けている。また、これらのセンサーデバイスから取得されたデータを集積し、その情報を利用者に提供するCosm[2]などのクラウドサービスも存在する。同時に、それ自体はインターネットに接続する手段を持たないが、スマートフォンと連携してインターネット上のサービスにデータを送信する種類のデバイスも存在する。Personal Area Network(PAN)用の通信方式の1つであるBluetoothによる接続機能を持つ健康機器が浸透しつつあり、体重計や血圧計、また万歩計などがある。これらはスマートフォンとの紐付け(ペアリング)し、スマートフォン側のアプリケーションによってデータが取得され、そのアプリケーションによってクラウド型のサービスにデータが送信されている。こうしたサービスの代表例としては「gooからだログ」[3]などが挙げられる。また健康機器専用のBluetoothの専用プロファイルであるHDP(Health Device Profile)[4]が存在し、Continua Health Alliance[5]として健康機器の業界団体で標準化が進められている。

人と人がつながるテクノロジーの変遷

人間と人間とのコミュニケーションの形はインターネットの普及と共に変化しつつある。1971年より使われ始めたメールは初期のものは英語しか使えなかったものの、1998年[6]に多言語化がなされ、日本語でのメールができるようになった。このことは後のiモード[7]などの携帯電話端末におけるメールの爆発的な普及の要因となった。また、メール

とは別に、オンラインでのチャットも Multics に搭載された .SAVED[8] から時代の変遷と共に IRC[9] が登場した。IRC は現在グループチャットと呼ばれるものであり、1つのサーバの中のチャットルームにログインし、多人数による会議を楽しむことを可能にした。その後、AOL による AOL Instant Messenger[10] が米国にて爆発的な人気を得た。2005 年には Google によって GoogleTalk[11] と呼ばれるサービスが開始された。このサービスは XMPP[12] と呼ばれる標準化されたチャット技術が用いられ、他のサービスとの相互接続が可能なチャットプロトコルである。また 2007 年には Twitter[13] がサービスインした。従来 SNS と呼ばれたサービスとは一線を画し、1人1人の状態を更新していく（ツイートと呼ばれる）ことができる人と人との緩いつながりを可能とするものである。このサービスを利用しているのは人間だけではない。自発的に、あるいは他のユーザーの発言に対する応答として、自動的に発言を行うプログラム（いわゆる Bot）も存在する。

データの所有権

先述したように、センサーデバイスは数、種類共に増えているが、日常の中で自分のものとして使えるデバイスは多くない。事実として、個人の身の回りには、多くの、本来有用であるはずのネットワークにつながったセンサーが多数存在している。しかしこれらのデータは現在それらセンサー専用のサービスの中に閉じられているため、そのデータを自分のものとして容易に利用するのは困難である。また、センサーによっては自分のものであるとは限らず、設置者が不明のセンサーも多く存在している。また、これらデバイスのデータを公開しようとする、アクセスコントロールが求められるが、こうした機構が個々のサービスにあるとは限らず、障害となっている。

1.2 目的

本論文ではネットワークに接続されたデバイスをこれまでより、人間により近づけることを目的とする。データの取得にコストがかかることから、デバイスを使用する権利は、ある 1つの組織や個人が所有するものではなく、コミュニティの中で共有されるべきである。デバイスを利用する権利、そこから生成されるデータの扱いは、日常生活の中におけるデータの取得のみならず、デバイスの所有者が変更された場合等、様々な場面で問題となる。本論文では、ネットワークに接続されたデバイスと人間との関係を整理し、それらの権利の扱いを容易かつ汎用的にすることを目的とする。

1.3 アプローチ

本研究ではネットワークにつながったデバイスと人間の結びつけを軸にしたシステムを構築する。従来、デバイスの所有権とその利用の権利はデバイスの所有者にある。また、この結びつけは強力であり、取得したデータの二次利用は困難であることが多い。本研究で提案するアーキテクチャではこれをユーザーを識別し権限を与えることによって分離で

きるようにする。これにより、センサーデータの共有がより円滑となり、デバイスの所有者の変更も容易となる。

1.4 本論文の構成

本論文は全 7 章から構成される。第 1 章では本論文の全体の概要及び構成を述べる。第 2 章では本研究の関連技術として、センサーネットワークのモデルとセンサーデータの共有技術について述べる。第 3 章では問題定義とアーキテクチャの要件についてまとめる。第 4 章では提案するアーキテクチャについて述べる。第 5 章では実際に構築した実装について述べる。第 6 章では本システムの評価を行う。第 7 章では本システムのまとめと今後の課題、展望について述べる。

第2章 関連技術

2.1 既存のセンサーネットワークのモデル

センサネットワーク活用調査報告書 [14] によると、センサーネットワークは近年、センサーの増加、高機能化、さらにネットワークそのものの高速・大容量化により、実世界の情報をより詳細に、迅速に取得することが可能となり、現在地球上の様々な場所で運用されている。こうして取得されたデータは環境モニタリングなどの研究、施設管理、物流やマーケティングに利用されているとされる。既存のセンサーネットワークは様々な種類があり、それぞれのシステムにおいて使われている技術も異なっている。この節ではこれらを分類していく。

Mottola[15] によると、既存のセンサーネットワークのアーキテクチャはデバイスの特性、通信モデル、モビリティ、空間、時間によって分類分けすることができる。その分類に基づいて、センサーネットワークに必要な要件を以下にまとめる。

2.1.1 デバイスの特性

デバイスの特性は以下の2つに分類される。

- センサー

センサーネットワークがセンサーのみで構成される場合はこれにあたる。この場合のデータの流れは片方向である。

- センサー+アクチュエータ

ネットワークがセンサーとアクチュエータで構成される場合はこれにあたる。この場合にはデータは双方向に流れる。

2.1.2 通信モデル

通信モデルは以下の3つに分類される。

- 多対一

図2.1に示す多対一のモデルは、1つ以上のセンサーからのセンシングをするのみのネットワークであれば多対一のモデルである。多くのセンサーネットワークがこれに当たる。

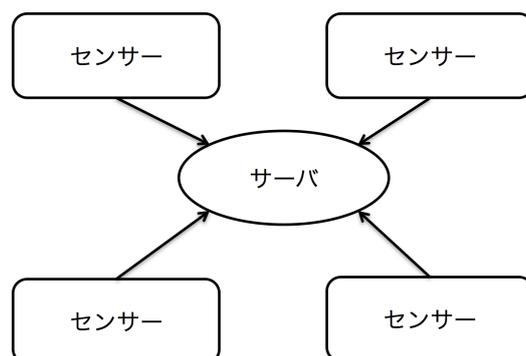


図 2.1: 多対一の通信モデル

- 一対多

図 2.2 に示す一対多のモデルは、多数のデバイスに設定を流しこむなどの必要がある場合には、一対多のモデルを考える必要がある。多対一の次に多い形である。

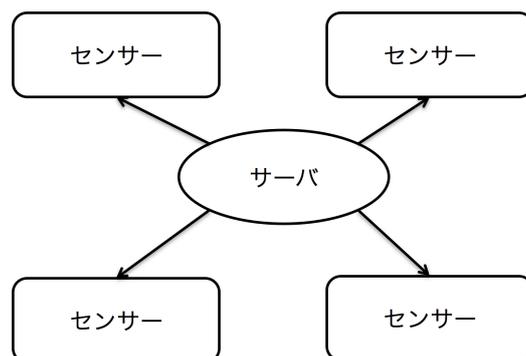


図 2.2: 一対多の通信モデル

- 多対多

図 2.3 に示す多対多のモデルは多数のセンサーと多数のアクチュエータで構成される場合に見られる。1つのセンサーから情報を受け取り、その結果によってアクチュエータが動作する場合などである。

2.1.3 モビリティ

デバイスが固定されているか移動するかについて論じる。

- 固定 (Static)

デバイスおよびデータを回収するシンクノードを設置後固定し、動かさない場合にはこのモデルにあたる。

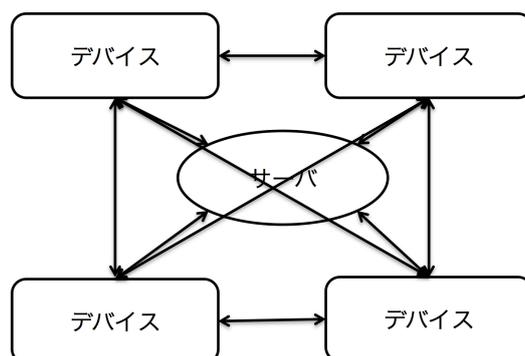


図 2.3: 多対多の通信モデル

- モバイルノード (Mobile Node)
設置後にデバイスを移動させる、あるいはデバイスが常に動いている場合はこのモデルにあたる。
- モバイルシンクノード (Mobile Sink Node)
シンクノードが移動する場合にはこのモデルにあたる。

2.1.4 空間

空間は以下の 2 つに分類される。

- グローバル
デプロイされたセンサーネットワークが対象とする現象がその全体に影響を及ぼすようなものの場合。
- リージョナル
対象とする現象がデプロイされたエリアの一部にしか影響を与えない場合。

2.1.5 時間

時間は以下の 2 つに分類される。

- 周期型
センシングデータを継続的に処理するタイプのもの。
- イベントトリガー型
イベントトリガー型の中で 2 つに大別される。1 つ目はセンシングデータがイベントによって送られてくるもの。2 つ目はデータが条件にマッチする場合にデバイスからデータが送信される場合である。

2.2 データの所有権

前章で述べたように、ネットワークに接続されたセンサーデバイスの数は増加傾向にあり、その所有者とデータは強く結びついている。このつながりの強さはセンサーの管理者にとってはその管理において他の利用者が少ないことによるメリット (運用上におけるメンテナンスの容易さ) があるが、これらのセンサーの利用者、また利用したいと考えている潜在的利用者にとっては、センサーデータへのアクセスの容易さを損なう原因となっている。デバイスの所有者とそこから得られるデータの所有者の分離が可能であれば、ネットワークにつながったデバイスの利活用がより円滑に行われるものと考えられる。

2.3 センサーデータ共有技術

現在、センサーデータを共有するプロトコルには、XMPP、HTTP、6LoWpan などがあり、またプロトコルの上に存在する手法として REST、SOAP、COAP、HTTP-RPC などがある。またこれらの手法におけるデータの表記法として JSON などがある。ここでは、主なものとして、XMPP、SOAP、REST、MQTT について解説し、その特徴を述べ、比較する。

2.3.1 XMPP

XMPP は Extensible Messaging and Presence Protocol の略である。RFC 6120(XMPP Core)[12]、及び 6121(Instant Messaging and Presence)[16] にて定義されており、IETF[17] の XMPP WG にて現在も標準化が進行中である。RFC2778[18] にて示された必要条件を満たすプロトコルの 1 つとして開発されている。XMPP は、開発当時は Jabber と呼ばれる技術であった。Jabber はプレゼンスとメッセージをリアルタイムに送りあうためのプロトコルとして、1998 年に Jeremie Miller によって開発された。その後、このソフトウェアにプログラマーが参加するようになり、Jabber のコミュニティが生まれ、Linux、Macintosh、Windows 向けのオープンソースのクライアントなどが開発された。また、Perl、Java などの言語のライブラリも同時期に開発された。また、1999 年から 2000 年にかけてプロトコルの詳細な設計が行われ 2000 年 5 月の jabberd 1.0[19] のリリースによって一定の終結をみた。XMPP は基本的にサーバー・クライアントモデルのプロトコルである。XMPP のアドレス (JID) はメールアドレスと同じように、[アカウント名]@[ドメイン名] の形となっている。ドメイン名は、FQDN(Fully Qualified Domain Name) となっており、名前解決には HTTP(Hypertext Transfer Protocol) や、SMTP(Simple Mail Transfer Protocol) のように DNS(Domain Name System) が使われる。アカウント名はユーザーがサーバーで JID を取得する際に決まる。また、クライアントが XMPP でサーバーにつながるときには、リソース名を追加して FullJID が必要となる。リソース名は XMPP の通信をログインしている全てのクライアントに配信させずに、特定の機械だけに通信させるための仕組みを内包するために作られた。FullJID は JID の後ろに、/とリソース名

の形で `alice@wonderland.lit/chair` のように追加される。XMPP において、図 2.4 の中の `foo@foo.jp` から `bar@bar.jp` へのメッセージの動きは以下の通りである。

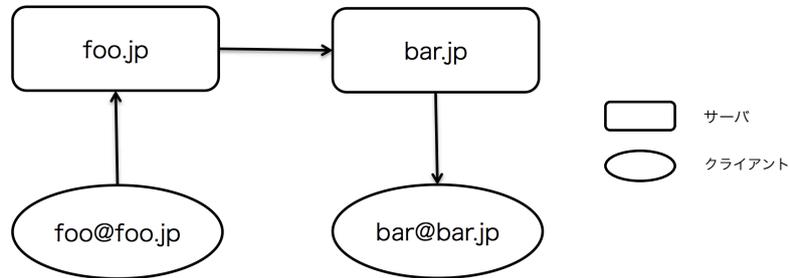


図 2.4: XMPP におけるメッセージの動き

1. `foo@foo.jp` はクライアントを使ってサーバーである `foo.jp` にログイン。
2. `foo@foo.jp` はサーバー `foo.jp` に向かってメッセージを送る。
3. サーバー `foo.jp` はサーバー `bar.jp` に向かってメッセージを送る。
4. サーバー `bar.jp` は `bar@bar.jp` に向かってメッセージを送る。
5. `bar@bar.jp` にメッセージが到着する。

このように、XMPP では電子メールとほぼ同じ要領でメッセージの配信が行われる。

XMPP は XML(Extensible Mark Up Language) のフォーマットを使って情報をやりとりしている。この XML の要素のことを XMPP では Stanza と呼ぶ。XMPP は Stanza のやりとりと言っても過言ではない。例えば図 2.4 に示したメッセージのやりとりの Stanza のサンプルは以下ようになる。

```
<message to='bar@bar.jp' from='foo@foo.jp/laptop' type='chat' xml:lang='en'>
<body>
Hi! How are you?
</body >
</message>
```

また、XMPP は、改ざんと傍受からの確保のために、TLS が使われている。さらにクライアント、サーバ双方において、STARTTLS を実装することが RFC 6120 [12] において求められている。さらに、上記のようなセキュリティ面、あるいはその他の機能の拡張は XSF(XMPP Standards Foundation)[20] により発行された XMPP の拡張の標準を定めた文書である XEP(The XMPP Extension Protocol) において定められる。XSF は XMPP の開発者のコミュニティの名称である。RFC と同様の発行形式をとっており、まずドラフトがコミュニティ内に示され、議論を重ね、標準化が行われ、XEP となって発行されてい

表 2.1: XEP-0045 における各ユーザの権限の違い

権限	なし	訪問者 (Visitor)	参加者 (Participant)	司会 (Moderator)
チャットルームへの参加	×	○	○	○
メッセージの受信	×	○	○	○
参加者のプレゼンスの受信	×	○	○	○
全参加者へのプレゼンスの送信	×	○	○	○
ステータスの変更	×	○	○	○
ニックネームの変更	×	○	○	○
プライベートメッセージの送信	×	○	○	○
参加者の招待	×	○	○	○
全ての参加者への メッセージの送信	×	×	○	○
件名の変更	×	×	○	○
訪問者と参加者の追放	×	×	×	○
発言の許可	×	×	×	○
発言の取り消し	×	×	×	○

る。例として、SASLにおけるクライアント証明書の取り扱いについてはXEP-0257:Client Certificate Management for SASL EXTERNAL[21]によって定められている。

代表的な XEP の 1 つにチャットルームを実現する XEP-0045:Multi-User Chat[22] が挙げられる。チャットルームの仕組みは特殊な JID に送られてきた Stanza をそこに参加している全ての参加者に送る仕組みとなっている。また、チャットルームに参加している参加者はニックネームで識別され、JID は共有されない。なお、チャットルームは参加者によって権限が違い、表 2.1 のようになっている。

2.3.2 SOAP

SOAP はメッセージのやりとりの定義付けを行うプロトコルである。W3C において標準化作業が行われており、現在の SOAP のバージョンは 1.2 である [23]。RPC(Remote Procedure Call) を実現するために用いられ、主に web サービスにおいて使われている。SOAP 自体は相互運用性のために、そのトランスポートプロトコルにとらわれないようになっているが、存在する実装のほぼ全てが HTTP を利用するものとなっている。また、初期の SOAP の仕様においては HTTP による使用が想定されていた。図 2.5 に示すように、SOAP でのメッセージの交換はクライアントは POST メソッドによって、XML をサーバーに送信、そのレスポンスもまた XML で返ってくるようになっている。SOAP は TLS を使った伝送路の暗号化と、証明書を用いたサーバおよびクライアントの証明が可能である。また、HTTP、あるいは XML のライブラリやアプライアンスをそのまま流用する

ことも可能となっている。しかしながら、HTTP のセッションは短いため、保持することが難しい。そのため、双方向の通信を行うことには向いていない。

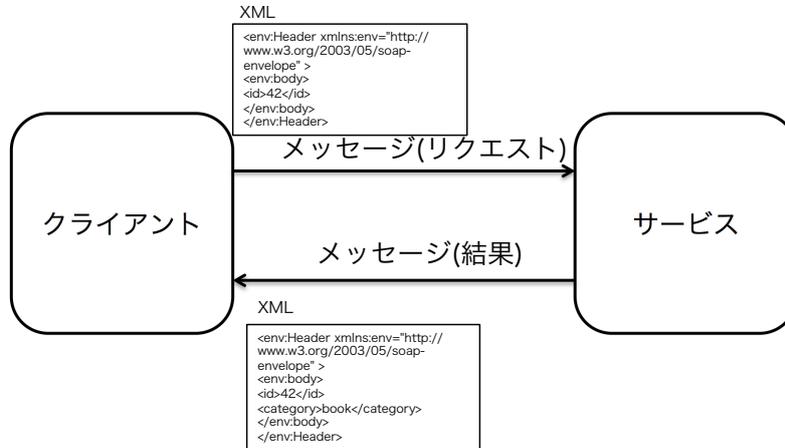


図 2.5: SOAP のメッセージのやりとり

2.3.3 REST

REST とは、Representational State Transfer の略で、HTTP を使って通信を行うものである。REST は本来の意味であればコンテンツの転送方法のアーキテクチャの 1 つであるが、ここでは REST を用いたアプリケーションについて述べる。REST は SOAP とは違い、XML ではなく HTTP のレスポンスを用いてデータをやり取りする。代表的な REST を用いられている例は RSS の配信である。クライアントは GET によってサイトの更新情報を XML で取得する。REST は GET に限らず、POST や REST、また DELETE を用いて URI によって特定したリソースを操る。SOAP と同じく、HTTPS による伝送路の安全性の確保と、証明書によるサーバ・クライアントの認証が可能である。また、ライブラリも豊富に存在する。

JSON

JSON(Java Script Object Notation) は、JavaScript のオブジェクトと同様の表記をするデータフォーマットの 1 つであり、REST と同時に Web アプリケーションに用いられることが多い。XML に比べて軽量でプログラムの中で容易に扱うことが可能であるが、値の 1 つ 1 つが何を示すかについては書かれていないため、JSON のデータそのものだけでは可読性は低くなってしまう。図 2.6 に JSON におけるメッセージのやりとりを示す。

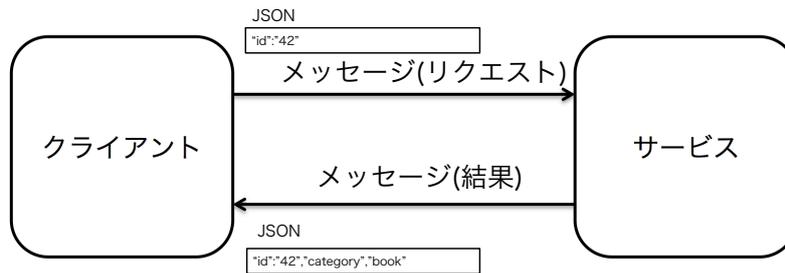


図 2.6: JSON のメッセージのやりとり

2.3.4 MQTT

MQTT(MQ Telemetry Transport)[24]とは、IBMによって開発されたマシン間通信向けの通信プロトコルである。Publish/Subscribe型の通信を行い、超軽量、低帯域向けかつ高遅延向けのネットワークにも耐えるように設計されている。MQTTは主にCEMS(Community Energy Management System)において採用実績があり、エネルギーの使用状況の管理などに用いられている[25]。しかしながら、セキュリティ面においてはMQTTでは定義されておらず、TLSやアプリケーションによる実装が求められている。

2.3.5 センサーネットワーク向けのサービス

cosm

cosmはセンサーやアクチュエーターを共有可能にするwebサービスである。世界中のセンサーの情報がリアルタイムでこのサービスに向かって送信されており、またAPIを利用して、このサービスに集まった情報を再度利用することも可能である。さらにアクチュエーションも可能となっている。また、cosmが受け付けるデータフォーマットは様々であり、XMLの1つの形式であるEEMLの他JSONやCSVも受け付ける。図2.7にcosmで放射線線量計センサーが公開されていることを示す。しかしながら、APIによってデータ受信が一極集中しており、cosmのサービスはたびたび停止している[26]。

Twitter

また、センサーネットワークよりも単純に個人ユーザなどがデバイスから得た情報を共有する手段としてTwitter[13]が挙げられる。Twitterはマイクロブローギングと呼ばれる140文字に限定した記事を公開することができるサービスである。軽量のライブラリも存在することからBotの形でネットワークにつながったセンサーの情報がTwitterで公開されている事例も存在する。しかしながらSOAP、RESTと同じく一方向なメッセージングのみとなっていること、またセキュリティ面においてはProtectedにすることも可能であるが、米国において行われた裁判においてはTwitterのデータは機密情報ではないとする判決が出ている[27]。

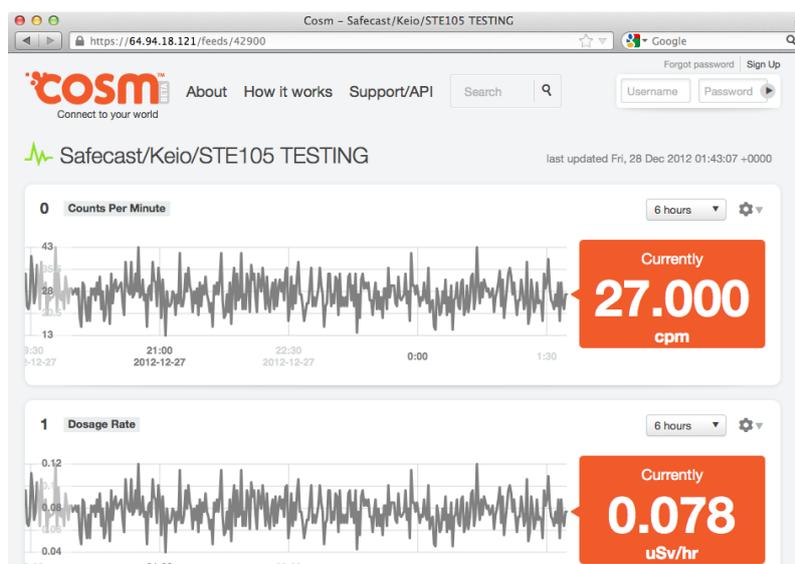


図 2.7: cosm 上の放射線線量計

2.3.6 まとめ

それぞれのプロトコルを利用した場合、またそれらプロトコルを利用したサービスを比較を表 2.2 に示す。また各項の内容についてそれぞれ述べる。

- 双方向性

双方向性については XMPP と MQTT をベースにしたものについては親和性が高い。これは XMPP が人間用のチャットプロトコルとして、MQTT が機械同士の通信に用いられることを考えられて設計されていることに起因する。Twitter については人のメッセージに返信する形であるリプライのみに対応している。デバイス向けの通信であることを考えると 140 文字の制限も不利であると考えられる。

- ライブラリの豊富さ

SOAP Based や REST Based のものや Twitter についてはユーザ向けアプリとして幅広く使われているため、豊富にライブラリが存在する。MQTT については実際に使われるのが企業向けであることが多く、ライブラリの数はさほど多くない。また XMPP については必ずしもライブラリは多いとは言えない状況である。

- セキュリティ

XMPP や SOAP、REST についてはその伝送路を TLS によって担保することが可能である。また、これらプロトコルはメッセージそのものを暗号化することも可能となっている。MQTT については閉じたネットワークにて少ない通信量で使用する事が仮定されているため暗号化などが行われていない。また、Twitter については TLS とアカウントを鍵付きにすることによってそのセキュリティが担保されるが、メッセージそのものを暗号化することができない。

表 2.2: 各プロトコルの比較

	XMPP Based	SOAP Based	REST Based	MQTT Based	Twitter
双方向性	○	×	×	○	△*1
ライブラリ の豊富さ	△	○	○	△	○
セキュリティ	○	○	○	×	△*2
拡張性	○	△	△	△	×

*1:Reply のみ

*2:Protected アカウント

- 拡張性

XMPP についてはその拡張性がプロトコルレベルで担保されている。これらは XEP による MUC への拡張などに現れており、またデータフォーマットそのものの拡張も Stanza にタグを入れることによって可能となっている。SOAP、MQTT 及び REST についてはそのアーキテクチャ上のデータが拡張可能であるが、プロトコルレベルでの拡張は難しい。また Twitter については 140 文字の制限から拡張は難しい。

第3章 問題定義とアーキテクチャ要件

この章では問題定義及びアーキテクチャ要件の整理を行う。

3.1 問題の定義

1.1 節では、身の回りに存在するデバイスの数は増加しているにも関わらず、アクセスコントロールが必ずしもうまくいっていないことから、実際にこれらのセンサーからデータを取得する際には困難があることを挙げた。

3.1.1 人間の社会とネットワークにつながったデバイスの関係の現状

1.1 節で挙げたように多くのセンサーはそれらセンサー固有のサービスの中に閉じており、人間が活動しているインターネット上、及び現実のコミュニティとデバイスのネットワークは別のネットワークとなっている。人間側のコミュニティからデバイスのネットワークにアクセスする時には図3.1に示すように「ゲートウェイ」となる専用のソフトウェアやデバイスを使う必要がある場合がほとんどである。こうした透過性のなさがデバイスと人間の間を隔てている。

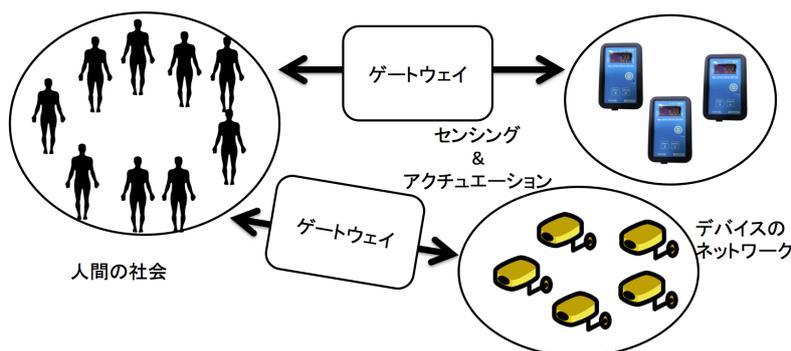


図 3.1: 人間の社会とネットワークにつながったデバイスの関係の現状

3.1.2 デバイスのネットワークにおけるデータ所有権

多くのデバイスのネットワークのアーキテクチャにおいてもデバイスのデータの統合的な取得、あるいは透過的な利用をすることは難しい。これは所有者とデバイスの結びつけを分離することが難しくなっていることに起因する。多くのシステムにおいて管理者のアカウントが1つとなっていて、一時的にアーキテクチャの一部の貸し出しや、オーナーの変更が難しいなどの問題が存在する。例として、公衆に設置されている体重計などの、得られるデータが利用者のパラメータによって変化し、かつそのデータにある一定のアクセスコントロールが求められる場合には、既存の1人の管理者あるいは1つの組織が全てのデータをコントロールするタイプのアーキテクチャは不向きである。公衆に設置された体重計、血圧計などの健康機器の他に、利用者によって取得されるデータが変わるデバイスとしてはデジタルサイネージが挙げられる。例として電子マネーカードを利用しているデジタルサイネージ型の自動販売機 [28] などは、カメラを内蔵することにより、購入者の性別や年齢層を判定しており、時間情報と合わせることでマーケティングデータとして利用されている。近い将来、こうした情報が利用者にフィードバックされ、電子マネーと連携し、レコメンデーションが行われることも想定される。この情報のやりとりの間にはデバイスと人間の結びつけが行われている。また、有事の際に一時的にデジタルサイネージデバイスの所有者を公共機関などに移譲して緊急情報の掲示に利用するケース場合も考えられる。こうした場合にはデバイスの所有者のダイナミックな切替が必要となる。また自転車がネットワークに接続される場合には所有者の切替をすることにより、レンタサイクルとして貸し出すことができる。このように公衆に設置されたネットワークにつながったデバイスにおいてはしばしば以下の2つの要件が必要となる。

- デバイス所有者のダイナミックな切替
- デバイスから取得できるデータの共有



図 3.2: デジタルサイネージ型の自動販売機



図 3.3: 公衆に設置された健康機器の例:体重計

3.2 デバイス所有権のダイナミックな切替

現状においてネットワークにつながったデバイスの所有権を変更し、統合的に扱うことは容易ではない。なぜなら、多くのデバイスがそのそれぞれのシステムに依存し構築されたネットワークの元にあるため、透過的にセンシングやアクチュエーションを行うようにはできていないからである。

以上を実現するためにまず、実世界におけるデバイスの人間による利用とそのデバイスの例を以下に分類した。

1. 1つのデバイスが1人の人間に専有される
例:体重計などの健康機器、アクチュエータなど
2. 1つのデバイスを何人かの人間によって共有される
例:温度計、ガイガーカウンターなどの、ユーザーが変わった場合においてもデータの単位以外の情報が変わらないデバイス
3. 1つのデバイスを別の1つのデバイスが専有し、それが何人かの人間によって共有される。あるいは1人の人間によって専有される
例:デバイスに搭載されたセンサー、グループ化されたセンサー
4. 1つのデバイスを別のいくつかのデバイスが共有し、それが何人かの人間によって共有される。あるいは1人の人間によって専有される

例: ネットワーク機器などからの情報 (SNMP など)

以上から、人間がデバイスを使う際にはかならず専有、あるいは共有が起きていることが分かる。また、利用者によって取得できるデータが変動するかどうかによる違いから、そのデバイスが共有できるものであるのか、専有する必要があるのかが分けられることが分かった。

1 や 3 のようなデバイスの使用に専有が必要になる場合には、そのユーザーは常に 1 人であることが必要である。また、2 や 4 のようなデバイスの場合には、そのデバイスにユーザーが複数同時に存在しても問題ないことが判明した。

また、占有を可能とすることによって健康機器からのデータ、あるいは取得したデータに機密情報を含む場合などにおいてもアクセスコントロールをかけることができる。これにより、プライバシーやセキュリティが確保が実現する。

3.3 データの共有

共有されているデバイスから得られるデータはデバイスの共有と呼ぶよりも、データの共有と呼ぶ方がふさわしい。なぜならば、所有権が誰であった場合においてもユーザーが得られるデータは同一であるからである。しかしながら、デバイスの所有者がユーザーが取得できるデータの種類の制限をかけたい場合は、これにデバイスの所有と同じくアクセスコントロールが可能とされることが求められる。

また、デバイスから取得可能なデータには型の違いが存在する。1.1 節で挙げたように、現在のセンサーは高機能化しており、1 つのセンサーが 1 つの値を出すとは限らずいくつかの複合的な情報を提供する場合が多い。体重計を例としても、体重の他にも身長や、体脂肪率などが取得できる場合がある。実際に様々なデバイスに対応したアーキテクチャを構築する際にはこれら型の違いを包括する拡張性が必須となる。

3.4 アーキテクチャ要件

以上に挙げた問題定義から得られるアーキテクチャのシステム要件は以下の通りである。

- デバイスと人間を結びつけられるアカウント機構が備わっていること
- 時分割によってデバイスの所有者が変化できること
- センシングデータの型の違いを包括する拡張性
- デバイスの所有者が個人に提供するデータを制限できること
- 実装の簡易化するための、標準化されたプロトコルの採用

3.4.1 デバイスを使用する際の権限

デバイスを使用する際に必要な権限は、デバイスが専有されるものか、あるいは共有されるものかによって異なる。この節では、デバイスを共有する際にどのような権限が必要であるかについて論じる。

デバイスの所有者が持つ権限

デバイスの所有者が持ちうる権限は以下が挙げられる。

- デバイスの利用の開始/停止
- データの取得
- データの 2 次利用可否の決定
- アクチュエーション
- 設定変更
- ユーザーのデバイス利用可能時間
- ユーザー間の割り込み優先度

専有型デバイス

1 人の人間がデバイスを専有するタイプのものでは、デバイスの共用利用を行う際に一時的な所有者のレベルでの権限が必要となる。以下に、アクセス権限を並べる。

- 他ユーザーの割り込みの許可
- データの取得
- データの 2 次利用可否の決定
- アクチュエーション
- デバイス設定変更
- 個人設定変更

共有型デバイス

複数の人間で同時に共有して使うことが可能なデバイスにおいては、他ユーザーとのコンフリクトを避けるために、デバイス自体の設定の変更や、データの2次利用可否の決定を行うことができないような設計をする必要がある。

- データの取得
- 個人設定変更
- アクチュエーション

第4章 提案アーキテクチャ

本研究では、ネットワークにつながったデバイスと、人間との結びつけを中心としたアーキテクチャを提案する。

4.1 デバイスの擬人化

3節では、現在の人間とデバイスの関係および現状のデバイスのアーキテクチャにおいて、それぞれにおける透過性のなさを問題として挙げた。本論文では、このような透過性を実現するアーキテクチャとしてデバイスのインテリジェンス性を上げることによる、デバイスのインターフェースの擬人化を提案する。擬人化により、これまでユーザーによる管理を出来る限り自動化し、ユーザーと人間が対等に対話するアーキテクチャを目指す。

例として保育施設に設置されたカメラへのアクセスコントロールを挙げる。通常、このようなカメラへアクセスは図4.1に示すように所有者である保育士によって、子どもとの関係性を元にコントロールされている。この判断は子どもの親、家族などカメラへのアクセス権をどこまで認めるかによって負担が増大していく。

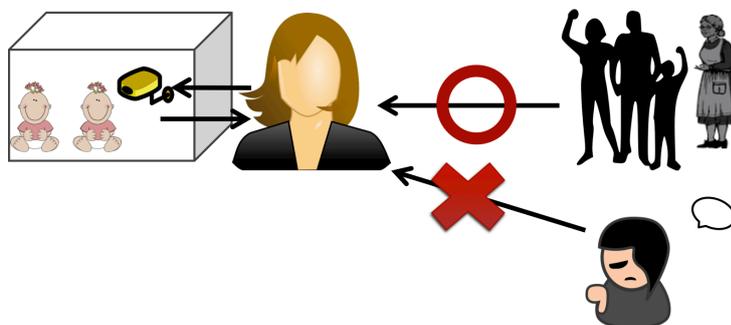


図 4.1: 保育者による保育施設に設置されたカメラへのアクセスコントロール

この問題は、図4.2に示すように、デバイスが擬人化されたインターフェースが個人を識別し、データベース化された子どもとの関係性を元に判断し、データベースがない場合のみその親に判断させるようにすることによって、所有者の負担を減らすことが可能となる。

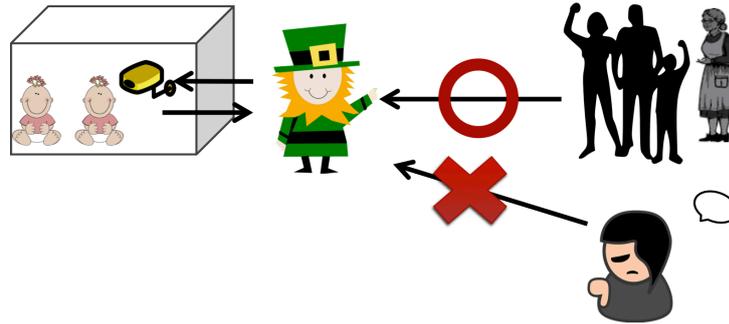


図 4.2: 擬人化されたインターフェースによる保育施設に設置されたカメラへのアクセスコントロール

4.1.1 デバイスのインターフェースを擬人化による利点の例

デバイスのインターフェースのみが擬人化されるため、デバイスそのものは単機能なものであったとしても、インターフェースをインテリジェントにすることのみで、人間同士の会話の中にデバイスを介在させることができるようになる。

- 擬人化されたインターフェースを持つ気温センサー

気温センサーが擬人化されることにより、最低気温や最高気温をユーザーに伝えるだけでなく前の日との差異からユーザーへ知らせることが可能となる。

- 擬人化されたインターフェースを持つ冷蔵庫

冷蔵庫が擬人化されることにより、家族の食事を決める会話の中において冷蔵庫による食品の在庫状況とレシピ情報によるサジェスションが行われることが可能となる。

また、ユーザーあるいはデバイスをグループにすることにより、人間とデバイスの1対1の通信だけでなく、1対多、多対1、多対多による対話を行うことが可能となる。これにより、多数のデバイスからのセンシング、アクション、コミュニケーションが可能となる。

4.2 方針

3.4 節に挙げられた要件を満たすシステムの方針を決定していく。

- 実装の簡易化のための、標準化されたプロトコルの採用

2.3.6 項にて、公衆に設置されたネットワークに接続されたデバイス向けのアーキテクチャを構築する際には XMPP が適切であることがわかった。以上より XMPP を使ったシステムの設計をする。

- デバイスと人間を結びつけることができるアカウント機構

XMPP はアカウントベースでメッセージのやりとりを行うため、この機構がすでに備わっている。センサーと人間を 1 つのエンティティとして捉え、フラットなネットワークを構築することによって、デバイスと人間に実空間に似たつながりを持たせるを持たせることができると考えられる。

- 時分割によってデバイスの所有者が変化できること

管理者として、デバイスと人間のアカウントを結びつけることができ、さらにそのアカウントが変更可能にしておく。

- センシングデータの型の違いを包括する拡張性

デバイスがセンシングデータを人間に送信すると同時に、そのデバイスの仕様を示すファイルの場所を送信し、それを人間側のクライアントで読み込むことで、拡張性を担保することができる。また、この方式を使うことで、デバイスが様々な種類のセンサーを搭載している場合においても対応できる。

- デバイスの所有者が人間に提供するデータを制限できること

アカウントによってデバイスが人間に提供するデータを、デバイスの所有者が制限できるようにしなければならない。デバイスの所有者と一時的な所有者との違いを区別する設計とする必要がある。

システム概要を図 4.3 に示す。ユーザーとデバイスはそれぞれクライアントを持ち、クライアントはサーバへと接続されている。ユーザーとデバイスはフラットにエンティティの形で存在し、デバイスはクライアントによって擬人化される。

図 4.4 にユーザーとデバイスの結びつけの遷移を示す。

1. ユーザーとデバイスが別個に存在する。

ユーザーとデバイスの間に結びつきはなくインタラクションはない。

2. ユーザーとデバイスの結びつけの開始

デバイスからのデータを取得したいユーザーは自分とデバイスを結びつけるためにフレンドリクエストをデバイスアカウントに申請する。

3. ユーザーとデバイスの結びつけの完了

結びつけが完了すると、ユーザーとデバイスは相互にやり取りが行えるようになる。

4. センサークライアントからユーザークライアントへのデータの送信

デバイスは取得したデータをユーザークライアントに向かってメッセージの形で送信する。

図 4.5 にユーザーとデバイスアカウントの結びつけの破棄の遷移を示す。

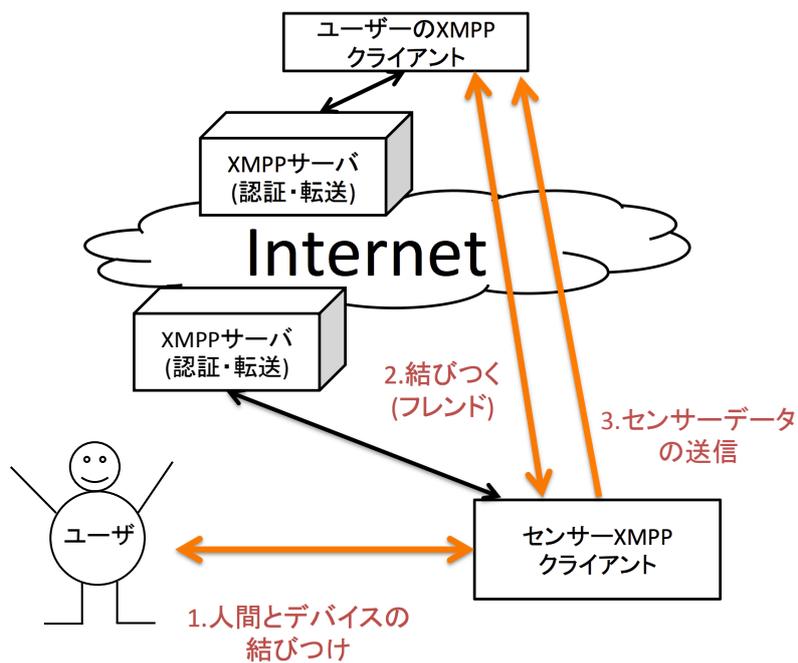


図 4.3: システム概要

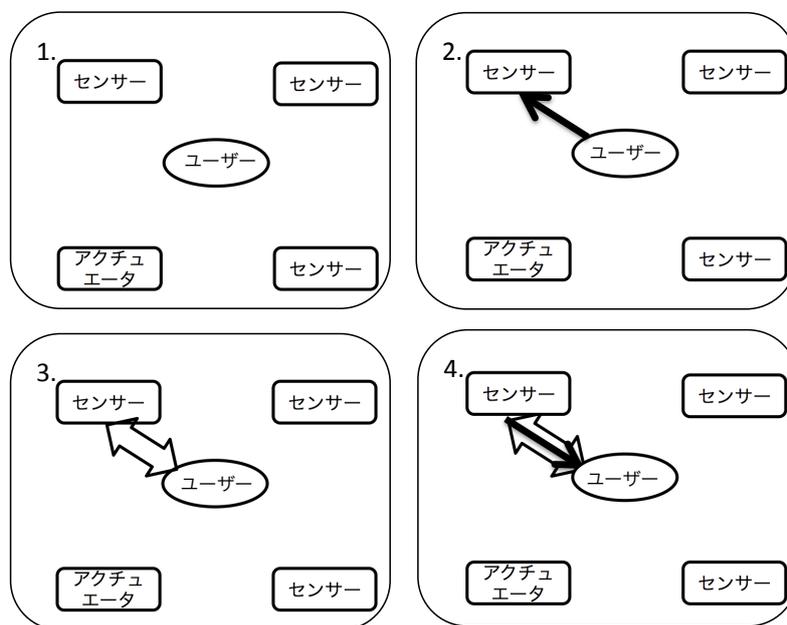


図 4.4: ユーザとデバイスの結びつけ

1. ユーザーとデバイスが結びついている
デバイスからユーザーに向かってデータが送られている。

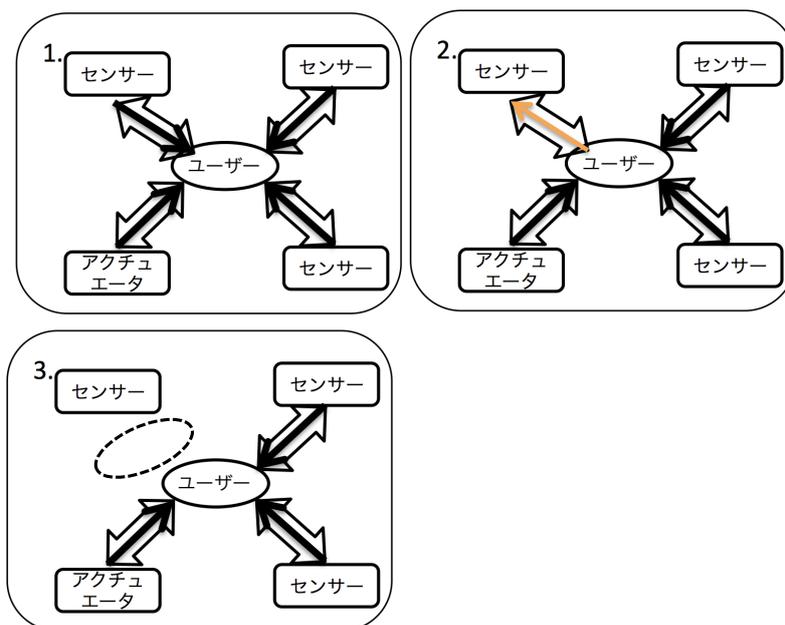


図 4.5: ユーザとデバイスアカウントとの結びつけの破棄

2. ユーザーアカウントからデバイスへのリムーブの送信
ユーザーがデバイスからのデータを受信しないことを決めた場合、リムーブを行う。
3. 結びつけの破棄
ユーザーとデバイスとの結びつけは破棄される。

4.3 仕様

4.2 節にて決定した方針をもとに仕様を策定していく。

4.3.1 デバイスの分類

次に世の中に存在するデバイスを4つに分類する。

- 周期起動型イベント向けセンサー
- 散発型イベント向けセンサー
- アクチュエータ
- 複合型デバイス

周期起動型イベント向けセンサーとは続けて起こる事象に対してある一定の間隔で数値が送信されるデバイスのことを指す。散発型イベント向けセンサーとは断続的に起こるイベント発生時のみに数値が送信されるデバイスのことを指す。アクチュエータは他のデバイスあるいは人間から発せられた命令を実行するデバイスである。アクチュエータは上記 2 つの情報を送信するデバイスに付随する場合もある。

周期起動型イベント向けセンサー

継続的なイベントをある一定の時間で区切り、ある一定の間隔でデータを送り続けるセンサデバイスを指す。温度計、ガイガーカウンターなどでは温度そのものは継続的な現象であるものの、実際のセンシングにおいてはデジタルである以上、サンプリングが行われるため、周期的になる。

散発型イベント向けセンサー

散発型イベント向けセンサーとはイベントの発生時にのみデータを送信するデバイスである。例としては体重計が挙げられる。体重計は人間あるいはものがその上に乗った時のみ動作するからである。

アクチュエータ

アクチュエータとは他のデバイスあるいは人間から発せられた命令を実行するデバイスのことである。例としては Web カメラなどがあげられる。

複合型デバイス

実世界におけるデバイスは、周期起動型イベント向けセンサーと散発型イベント向けセンサーのような組み合わせを持った複合型デバイスがほとんどである。また、センサーとアクチュエータを同時に持ち合わせている場合も存在する。なぜなら、電源の on/off は本論文においては周期起動型イベントに該当するため、多くの散発型イベント向けセンサーは同時に周期起動型イベント向けセンサーでもあるからである。

4.3.2 デバイスへのユーザーの権限

3.4.1 項で示したように、専有型のものとは共有型のデバイスにおいてはそのユーザーが持ちうる権限が異なる。本節では権限によるユーザーの分類を行う。

- Owner

Owner とはデバイスの所有者である。デバイスの所有者はデバイスに関する全ての権限を持つ。

表 4.1: デバイスの種類とユーザーの権限の種類の対応

	周期起動型イベント向けデバイス	散発型イベントデバイス
専有型	心拍計、スクリーンなど Moderator	体重計、体温計、など Moderator
共有型	気温計など Subscriber	ドアの開閉、照明の on/off、 ライブカメラなど Subscriber

- Moderator

Moderator はデバイスの排他的利用権を持つユーザーである。専有型のデバイス向けのユーザーの種別として用いられる。3.4.1 項に示した通りの権限を持ち、デバイスを専有している間は、他ユーザーの割り込みの可否を判断する。

- Subscriber

Subscriber はデバイスのデータを取得できるユーザーである。共有型デバイス向けのユーザーの種別として用いられる。3.4.1 項に示した通りの権限を持ち、デバイスを使用中はデバイスのデータを取得することが可能である。

- Limited

Limited とは各 User に Owner の要求に応じて付加されるフラグであり、Moderator や Subscriber の権限を制限するものである。各センサークライアントはユーザーからの利用要求があるたびに、ユーザーデータベースに問い合わせを行い、Limited フラグがついている場合には結びつけ後のユーザーの権限を制限する。

4.3.3 デバイスの種類とデバイスへのユーザーの権限の種類の対応

表 4.1 に、デバイスの種類とユーザーの権限の種類の対応を示す。専有型のデバイスには Moderator、共有型のデバイスには Subscriber が用いられる。

4.3.4 クライアント間の通信

この節ではクライアント間の通信について述べる。

なお、仕様の中で 2 人の人間のアカウントと 2 つのセンサークライアント向けアカウントと 1 つのアクチュエータ向けアカウントが登場する。

- alice@domain.com

表 4.2: Limited であることの通知

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
flag	limited	

- bob@domain.com

上記2つのアカウントは人間のアカウントであり、Stanza は実際に人間がクライアントを操作した結果送受信されるものである。

- exclusive@domain.com

exclusive@domain.com は専有型デバイスに用いられているセンサークライアント向けアカウントである。

- share@domain.com

share@domain.com は共有型デバイスに用いられているセンサークライアント向けアカウントである。

- actuator@domain.com

actuator@domain.com はアクチュエータに用いられているアクチュエータ向けアカウントである。

人間とデバイスの結びつけに関するメッセージ

RFC 6120(XMPP Core)[12]、及び 6121(Instant Messaging and Presence)[16] にて定義されているものに準拠する。

権限に関するメッセージ

あるユーザーが Limit であることの通知は表 4.2 のフォーマットによって表現される。ユーザーによるデバイスへの自分の権限の一覧の要求は表 4.3 のフォーマットによって送信される。

権限一覧応答は表 4.4 のフォーマットによって表現される。

あるユーザーの権限を owner に変更するメッセージは表 4.5 のフォーマットによって表現される。

表 4.3: 権限一覧要求

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
request		
	type	privilege

表 4.4: 権限の一覧応答

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
response		
privilege		
item		
	type	data
	value	acquire
	customset	allow
	actuation	allow

ユーザー権限の変更要求は表 4.6 のフォーマットによって表現される。

4.3.5 データ送信

計測データの送信は表 4.7 のフォーマットによって表現される。

4.3.6 アクチュエーション

デバイス固有の機能の一覧の要求は表 4.8 のフォーマットによって表現される。

デバイス固有の機能の一覧の応答は表 4.9 に示すフォーマットによって表現される。

表 4.5: ユーザー権限の owner への変更

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
set		
	type	privilege
	user	owner に変更される JID
	value	owner

表 4.6: ユーザー権限の変更要求

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
set		
	type	privilege
	user	権限が変更される JID
	value	limited
privilege		
item		
	type	acquire
	customset	allow
	actuation	allow

アクチュエーションの実行は表 6.3 のフォーマットによって表現される。
 アクチュエーションの結果は表 4.11 のフォーマットによって表現される。

表 4.7: 計測データの送信

要素	属性	属性値	値
message			
	type	chat	
	to	送信先 JID	
	from	送信元 JID	
device	urn:xmpp:device		
data			
dose			
	unit	単位	
			値
time			
			[YYYY:MM:DD HH:MM:SS タイムゾーン]

表 4.8: デバイス固有の機能の一覧要求

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
request		
	type	uniqueness

表 4.9: デバイス固有の機能の一覧の応答

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
response		
uniqueness		
device		
	type	sensor
	name	機能の名前
element		
	value	要素名
attribute		
	value	属性名
	variable	属性値として取りうる文字列
	meaning	属性の指す意味

表 4.10: アクチュエーションの実行

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
request		
take	type	privilege
	デバイス固有属性名	属性値

表 4.11: アクチュエーションの結果

要素	属性	属性値
message		
	type	chat
	to	送信先 JID
	from	送信元 JID
device	urn:xmpp:device	
request		
data		
take		
	type	privilege
	result	URL

第5章 実装

5.1 周期起動型イベント向けセンサー

本節では周期起動型イベント向けセンサーとして、放射線量計を使った実装について述べる。放射線を計測するセンサーとして、図 5.1 に示すテクノエーピーの Mini スペクトルメーター TA100 を用いた。表 5.1 にその詳細を示す。



図 5.1: Mini スペクトルメーター TA100

また、個人識別を FeliCa カードによって行った。FeliCa カードとしては慶應義塾大学の学生証、Suica[29]、PASMO[30] を用いた。FeliCa のリーダーとして、図 5.2 に示す、SONY 非接触 IC カードリーダー/ライター PaSoRi RC-S310 を用いた。FeliCa リーダーの詳細については表 5.2 に示す。

図 5.3 に本センサークライアントシステムの全体を示す。線量を知りたいと思ったユーザー (人間) は自分の FeliCa カードを FeliCa リーダーにタッチする。すると、ユーザーとセンサーの結びつけが行われ、ユーザーの XMPP アカウントに線量データが送られ続けるようになる。

図 5.4 にシステムの詳細を示す。センサークライアントプログラムは図 5.5 に示す MacBook 上に構築した仮想マシン上で動作している。FeliCa の IDm を読み込む実装の実行環境については表 5.3 に示す。このプログラムでは常に FeliCa の個別識別子である IDm

表 5.1: テクノエーピー Mini スペクトルメーター TA100 詳細

製品名	Mini スペクトルメーター TA100
メーカー名	株式会社テクノエーピー
製品種類	ガンマ線核種線量率計
通信方式	USB シリアル通信
測定線種	γ 線、X 線 (20keV \sim 1.5MeV)
線量率範囲	0.01 μ Sv/h \sim 10mSv/h
線量率時定数	3 秒、10 秒、30 秒、60 秒、90 秒、AUTO



図 5.2: FeliCa リーダー RC-S310/ED2

の読み取りが polling によって行われている。ユーザーの FeliCa カードをリーダーをかざすとその IDm を読み取る。次に取得した IDm をあらかじめ用意してある IDm と JID の対照データベースに問い合わせる。ユーザの JID を受け取り次第、そのユーザーのフレンドリクエストをセンサーの JID からユーザの JID に送信する。ユーザ側がこれを許可すると、クライアントプログラムは線量計に現在の放射線量を問い合わせ、その結果をユーザに XMPP のメッセージとして送信する。

表 5.2: ソニー株式会社 Felica リーダー RC-S310/ED2 詳細

製品名	SONY 非接触 IC カードリーダー/ライター PaSoRi RC-S310
メーカー名	株式会社ソニー
通信方式	USB シリアル通信
サイズ	高さ 63mm \times 幅 90mm \times 奥行 17mm

表 5.3: FeliCaID 読み込み実装実行環境

OS	Ubuntu 10.04
ハードウェア	MacBook 上の仮想マシン
Felica リーダ	RC-S310/ED2
使用言語	C 言語
データベース	MySQL Server

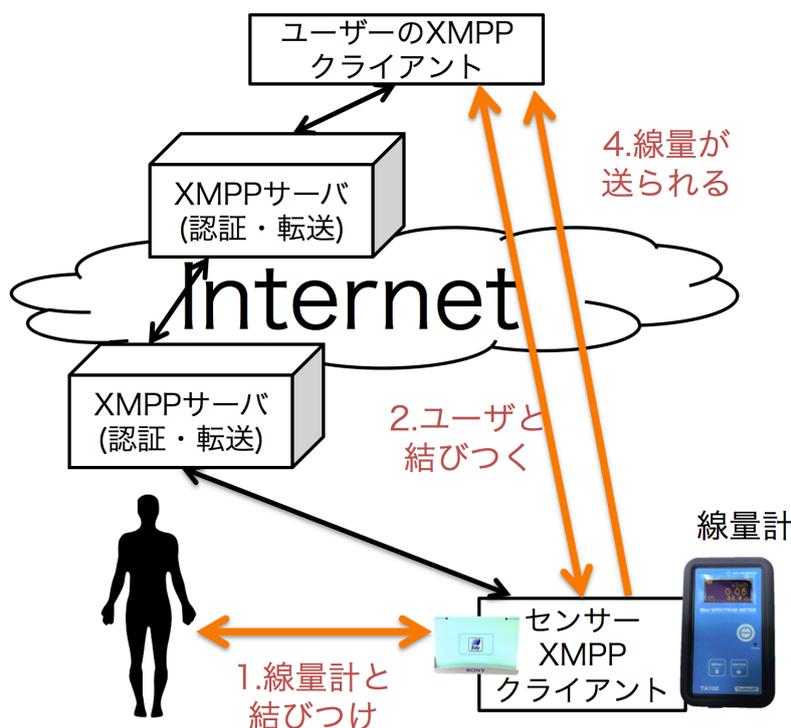


図 5.3: 周期起動型イベント向けセンサークライアントシステム全体図

5.2 散発型イベント向けセンサー

本節では散発型イベント向けセンサーとして Bluetooth で接続される体重計を使った実装について述べる。体重を計測するセンサーとして図 5.6 に示すエー・アンド・デイの体組成計 UC-411PBT-C を用いた。表 5.4 が体組成計の詳細である。本体組成計は Bluetooth によってそのデータが送信される。データの送受信に使われるプロファイルは HDP(Health Device Profile)[4] によるものである。

図 5.7 に散発型イベント向けセンサークライアントシステムの全体を示す。ユーザーは、あらかじめ登録した FeliCa カードを FeliCa リーダにかざす。これによって、ユーザーと体組成計が結びつく。結びつけ完了後、ユーザーは体組成計に乗って、データを取得す

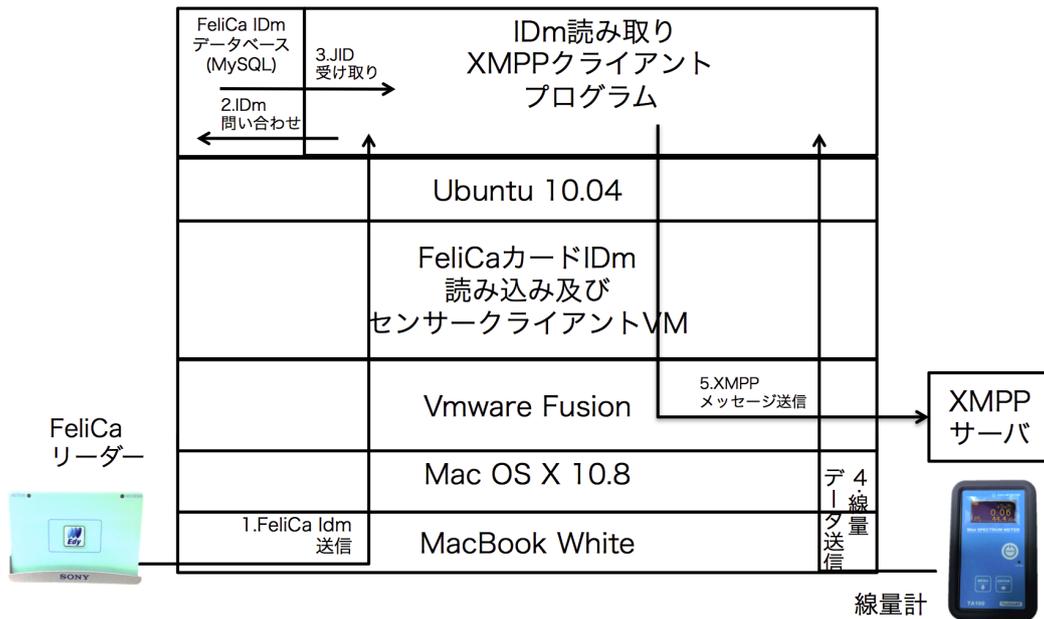


図 5.4: 周期起動型イベント向けセンサークライアント詳細図



図 5.5: センサークライアントとしての MacBook

る。最後にセンサークライアントからユーザクライアントに XMPP のメッセージによってデータが送信される。

図 5.8 にシステムの詳細を示す。センサークライアントとしては周期起動型イベント向けセンサー実装と同じく図 5.5 に示す MacBook を使用し、その上に 2 台の仮想マシンを構築した。1 台目の仮想マシンは周期起動型イベント向けセンサーと同様に FeliCa カードを読み込むプログラムが動作している。よって、実行環境についても表 5.3 に示す通り同一である。この仮想マシンはカードリーダーより読み込んだ IDm と対応する JID を 2 台目の仮想マシンに受け渡すことを目的としている。このため、周期起動型イベント向けセンサー実装の読み取りプログラムを Apache サーバ上のテキストファイルに JID を出力す



図 5.6: 体組成計 UC-411PBT-C

表 5.4: エー・アンド・デイ 体組成計 詳細

製品名	体組成計 UC-411PBT-C
メーカー名	エー・アンド・デイ
通信方式	Bluetooth 通信 Ver2.1 Class1
Bluetooth プロファイル	Health Device Profile(HDP)
通信内容	個人番号、測定値（体重、体脂肪率、基礎代謝、筋肉量、水分量）、測定日時

るように変更した。2台目の仮想マシンは、表5.5に示す通り、スマートフォン向けOSであるAndroidのx86版のものを構築した。この仮想マシンはユーザが体組成計に乗って、データがBluetoothによって送信されると、表5.5に示す実行環境において、MacBookのBluetoothアダプタよりデータを体重計連携アプリケーションによって取得する。受け取ったデータはXMPPクライアントアプリケーションに受け渡される。このアプリケーションは1台目の仮想マシンのApache Webサーバにアクセスし、ユーザのJIDを取得する。次に、このJIDに向かってXMPPによってメッセージを送信する。

5.2.1 ユーザークライアント

ユーザ側のクライアントプログラムはセンサークライアントからメッセージを受信するものである。実装の実行環境は図5.9に示すXserve上に構築した仮想マシン上で動作する。また実行環境については表5.6に示す。

ユーザークライアントはセンサークライアントからメッセージを受け取り後、Stanzaの

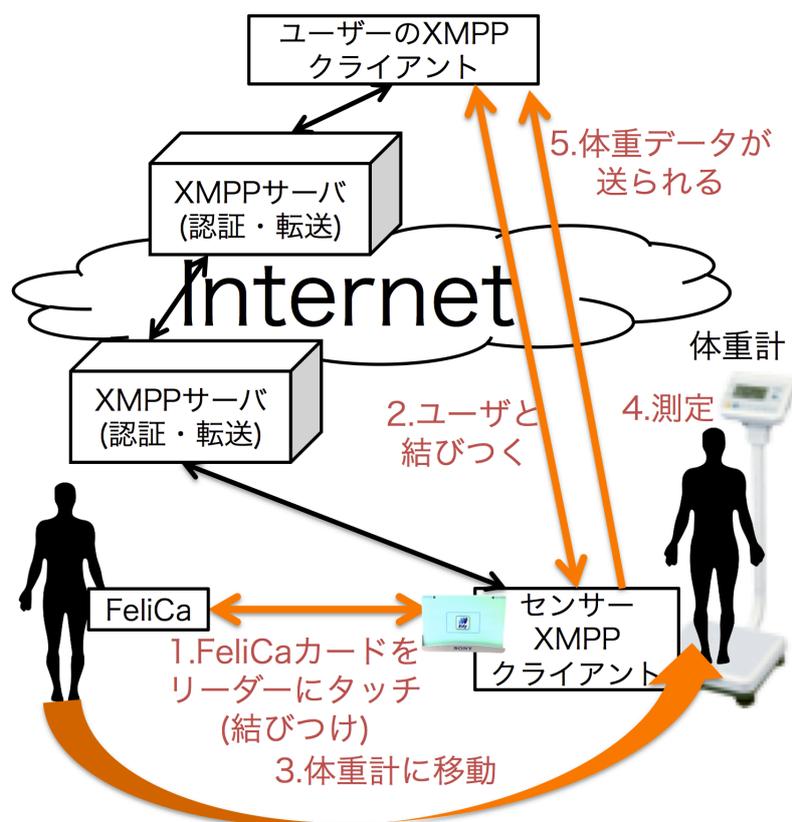


図 5.7: 散発型イベント向けセンサークライアントシステム全体図

表 5.5: 体重計連携および XMPP クライアントアプリケーション実行環境

OS	Android-x86 4.0.3
ハードウェア	MacBook 上の仮想マシン
Bluetooth アダプタ	MacBook 内蔵
使用言語	Java

表 5.6: ユーザークライアントプログラム実行環境

OS	Ubuntu 10.04
ハードウェア	Xserve 上の仮想マシン
使用言語	C 言語
データベース	MySQL Server

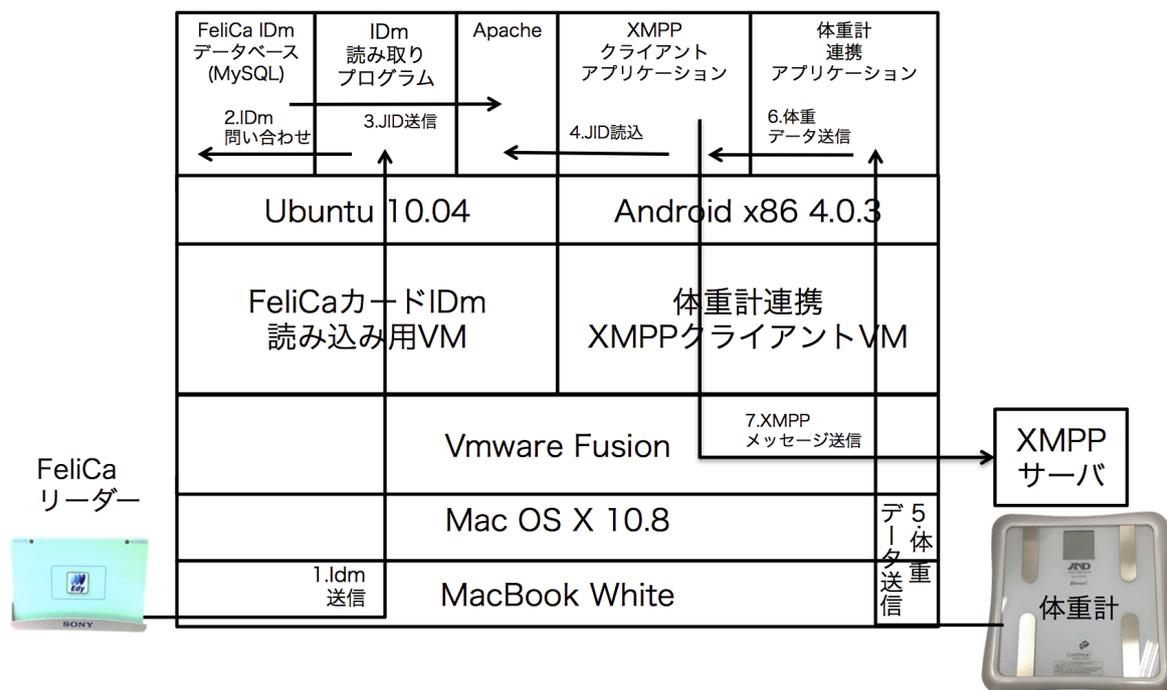


図 5.8: 散発型イベント向けセンサークライアント詳細図



図 5.9: ユーザークライアントとしての Xserve

パースを行い、データベースに保存、ユーザーは Web インターフェースを使ってその結果を閲覧することができるようになっている。図 5.10 に受け取ったメッセージの処理についての詳細を示す。

5.3 アクチュエータ

本節では Web カメラを使ったアクチュエータクライアントについて述べる。Web カメラとしては、他の実装と同じく図 5.5 に示す MacBook の内蔵カメラを用いた。図 5.11 に本アクチュエータクライアントシステムの全体を示す。システムの実装例として web カメラを挙げる。実際の web カメラの動作の様子は以下の通りである。

1. ユーザーがデバイス XMPP クライアントに向かってリクエストを受信する。
2. デバイス XMPP クライアントはそのユーザーがデバイスを使用してよいかについて、NG のユーザーのリスト、OK のユーザーのリストに問い合わせを行う。

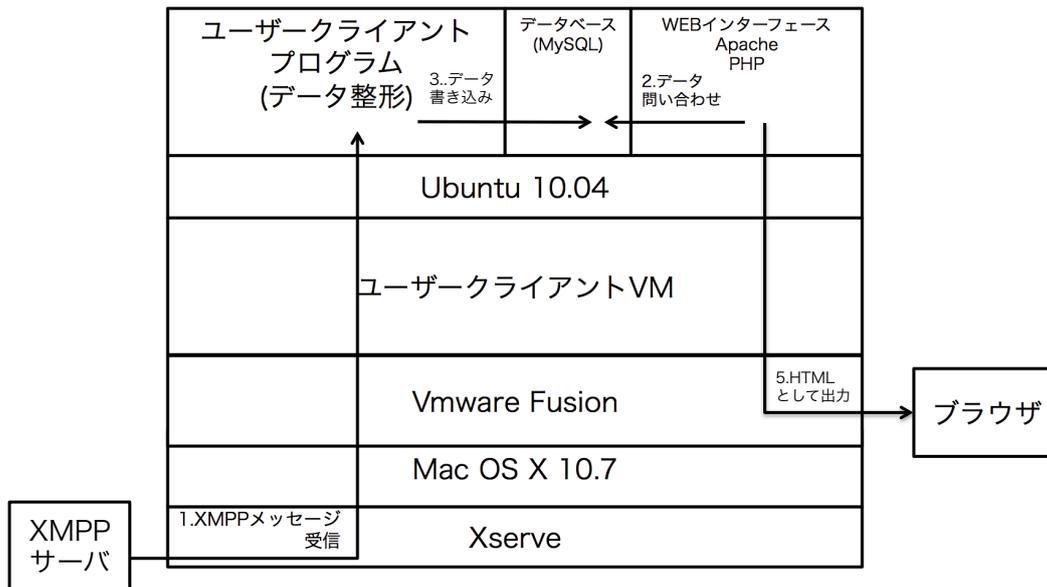


図 5.10: ユーザークライアント詳細図

3. ユーザーのアカウントがNGのリストに入っていた場合には、ユーザーへアクセスが禁止されていることを伝える。ユーザーのアカウントOKのリスト、NGのリストのどちらにも存在しない場合には、デバイスの所有者に判断を仰ぐメッセージを送信する。OKのリストに入っていた場合には、4に進む。
4. 画像の取得を行う。
5. データをアクチュエータ XMPP クライアントに受け渡す。
6. アクチュエータ XMPP クライアントはデータを web サーバにアップロードし、URL を取得する。URL は一時的なものとする。
7. 画像の URL の入ったメッセージをユーザーに送信する。

図5.12 にシステムの詳細を示す。アクチュエーションの実際としては、MacBook に接続された Web カメラを外部コマンドと連携することによって、撮影をし、画像ファイルを Web サーバにアップロードし、当該ファイルの URL を含む XMPP メッセージをユーザーに送信することによって実現される。

5.4 XMPPサーバ

本システムで利用した XMPP サーバーは図5.13 に示す Mac Mini 上に構築した Openfire[31] である。表 5.7 にサーバーの実行環境を示す。

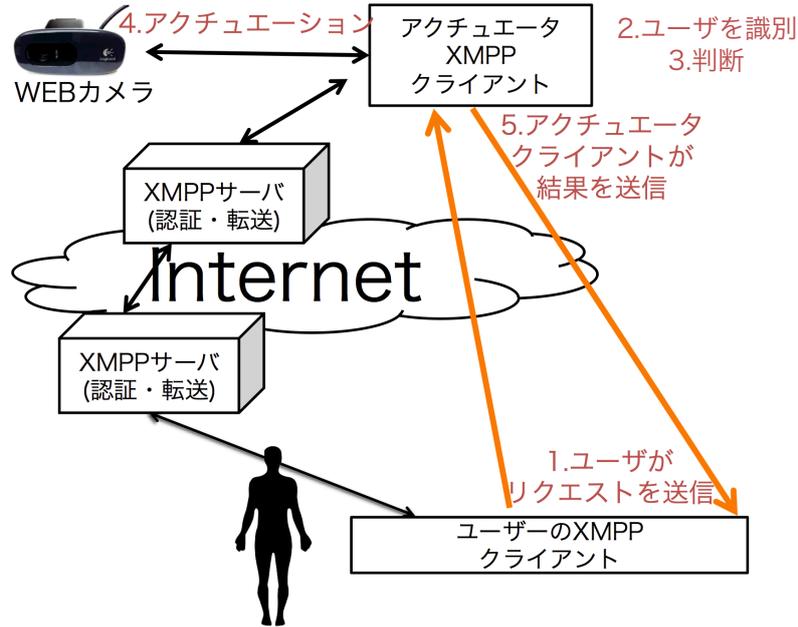


図 5.11: アクチュエータクライアントシステム全体図

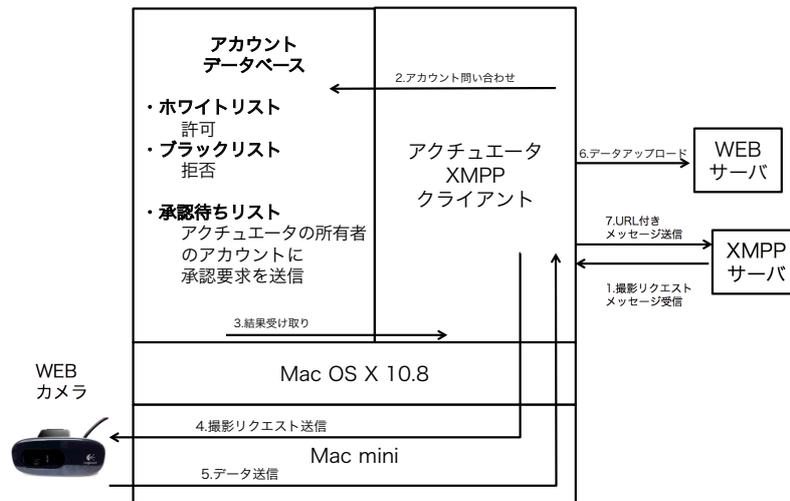


図 5.12: アクチュエータクライアント詳細図

表 5.7: Openfire 実行環境

OS	Ubuntu 10.04
ハードウェア	Mac Mini
Openfire	Version 3.7.0
Java	Version 1.6.0-24 Sun Microsystems Inc. OpenJDK Zero VM



図 5.13: XMPP サーバとしての Mac Mini

第6章 評価

第5章において述べた周期起動型、散発型イベント向けセンサーシステム、及びアクチュエータシステムの実装について、実現可能性、規模性、汎用性の観点から評価を行う。

6.1 実現可能性

各実装についてその実現可能性について述べる。

6.1.1 周期起動型イベント向けセンサー

周期起動型イベント向けセンサーとしての放射線線量計実装を実際に2012年12月31日から2013年1月5日まで継続して動作させた。2つの線量計を用意し、それぞれについてセンサー XMPP クライアントを動作させた。

- フレンドリクエストへの応答

実際に運用するにあたり、ユーザに見立てた JID より、断続的にフレンドリクエストおよびアンフレンドリクエストを送信し、それに対して正しく応答するかどうかについて検証を行った。表 6.1 にその成功率について示す。

- 継続的なデータの読取りの可用性

連続的に1つのユーザー XMPP クライアントに向けてデータを送り続け、その動作検証を行った。図 6.1 に結果を記す。

表 6.1: フレンド及びアンフレンドリクエストの成功率

	12/30	12/31	1/1	1/2	1/3	1/4	1/5	成功率
センサー A	○	○	○	○	○	○	○	100 %
センサー B	○	○	○	○	○	○	○	100 %

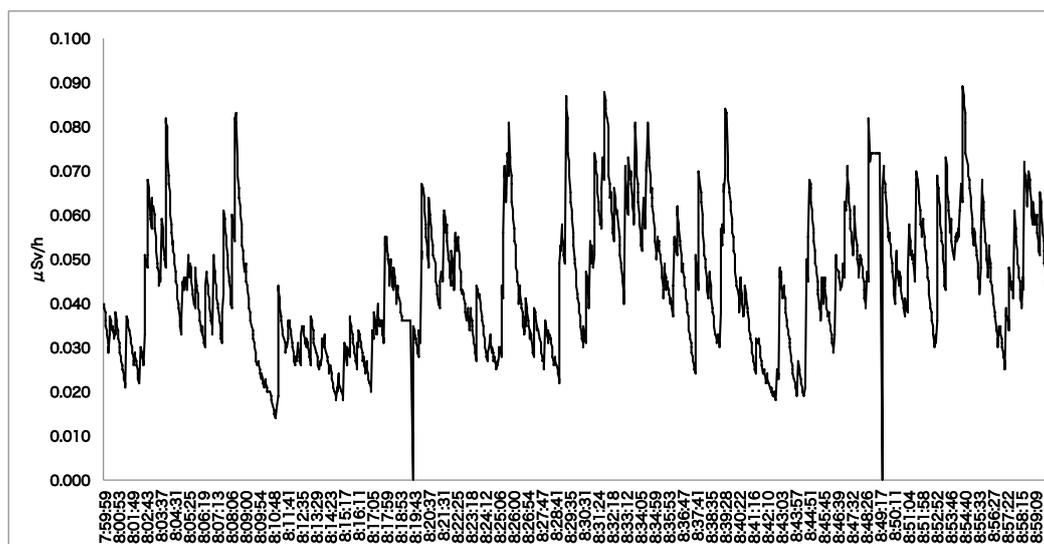


図 6.1: 12/31 8:00-9:00 の空間放射線量

表 6.2: 2 名による重量の測定と実測値との差異

	1回目	2回目	3回目	4回目
被験者 A	88.8	98.3	90.2	92.95
被験者 A(実測値)	88.8	98.3	90.2	92.95
被験者 A の 実測値との違い	0	0	0	0
被験者 B	51.7	53.2	61.15	55.55
被験者 B(実測値)	51.7	53.2	61.15	55.55
被験者 B の 実測値との違い	0	0	0	0

6.1.2 散発型イベント向けセンサー

散発型イベント向けセンサーとしての体組成計実装について動作検証を行った。2名で交代で8回ずつ荷物を持つことによって重量を変えながら、正確なデータが送信されていることについて検証を行った。表 6.2 にその結果を記す。

6.1.3 アクチュエータ

実際に運用するにあたり、断続的にアクチュエーションをリクエストして、それに対して正しく応答するかどうかについて検証を行った。表 6.3 にその成功率について示す。

表 6.3: アクチュエーションの成功率

	1 回目	2 回目	3 回目	4 回目	5 回目	6 回目	7 回目	成功率
アクチュエータ A	○	○	○	○	○	○	○	100 %
アクチュエータ B	○	○	○	○	○	○	○	100 %

6.2 規模性

ユーザー、あるいはデバイスに付属しているセンサーが増加した場合におけるデバイス側のセンサー XMPP クライアントの規模性について述べる。

6.2.1 ユーザーの増加によるセンサー XMPP クライアントからの Stanza の増加

1 つのセンサー XMPP クライアントに結びついているユーザー数が増加すると 1 回のデータ送信につき、メッセージ Stanza の数が増加する。Stanza が増加が著しいと、遅延が発生する原因となる。ユーザにメッセージを送信する方法には、各ユーザに向けてデータを送信する方法と、MUC(チャットルーム)によって一度に複数のユーザーに向けてデータを送信する 2 種類の送信方法が存在する。MUC を利用することにより、擬似的な Multicast が可能となるため、Stanza の増加を抑えられることが期待される。デバイスと人間の結びつけ時と、メッセージの送信時の、MUC を使用した場合と使用しなかった場合のクライアント、及びサーバにおける流れる Stanza の数の比較を行った。クライアント側については受信 Stanza 数、送信 Stanza 数、総 Stanza 数を、サーバについては扱う Stanza の総数について比較した。

クライアント側の Stanza の数の変化

クライアント側における MUC を使用した場合と MUC を使用しなかった場合の結びつけ時とメッセージ送信時における、ユーザ増加に伴う Stanza の数を比較した。

1. 結びつけに伴って発生する Stanza の比較

- MUC 非使用時の結びつけに伴って発生する Stanza

MUC 非使用時の結びつけに伴って発生する Stanza は以下の 2 種類である。

- ユーザーからのフレンドリクエスト受信
- リクエストの応答送信

したがって、受信 Stanza 数、送信 Stanza 数共にユーザ数と同値である。図 6.2 に増加の様子をグラフにしたものを示す。

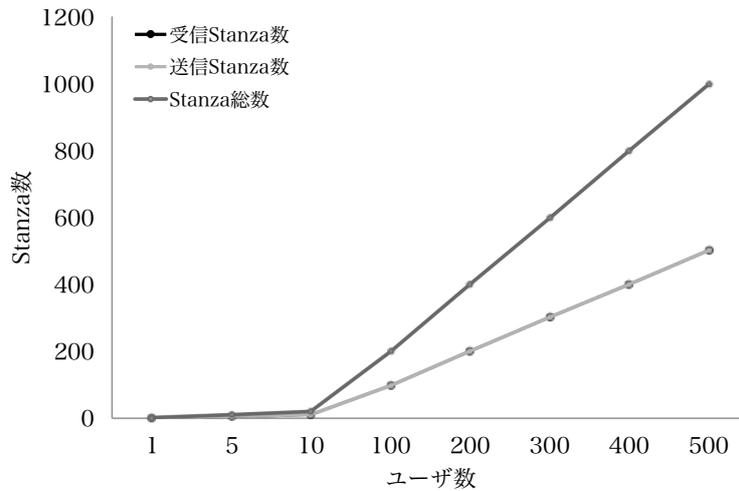


図 6.2: MUC 非使用時のクライアント結びつけ時のユーザ増加に伴う Stanza の増加

- MUC 使用時の結びつけに伴って発生する Stanza

MUC 使用時におけるクライアントとユーザーの結びつけ時における Stanza は、

- クライアントのチャットルームへの参加要求送信
- それに対するサーバからの応答受信
- 各ユーザがチャットルームに参加するごとにクライアントへの参加の通知受信

の 3 種類である。受信 Stanza 数については「クライアント数 \times 1+1」となり、送信 Stanza 数については 1 となる。この時のグラフは図 6.3 に示すように線形となる。

- MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較

図 6.4 に MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 総数の比較を示す。MUC の使用によって Stanza の増加を約 1/2 に抑えられていることがわかる。また、その差もユーザー数の増加によって大きく異なるため、本提案アーキテクチャにおいて、クライアントの増加が著しい場合には MUC の使用が望ましい。

2. メッセージ送信に伴って発生する Stanza の比較

メッセージ送信に伴って発生する Stanza は MUC 使用時の場合、チャットルームへのメッセージの送信のみとなる。したがって、MUC 使用時にはユーザーの増加に伴う Stanza 数の増加は起こらない。

MUC 非使用時の場合については全ユーザーへのメッセージの送信となる。こちらはユーザーの増加に伴って Stanza 数の増加が起こる。図 6.5 に増加の様子を示す。

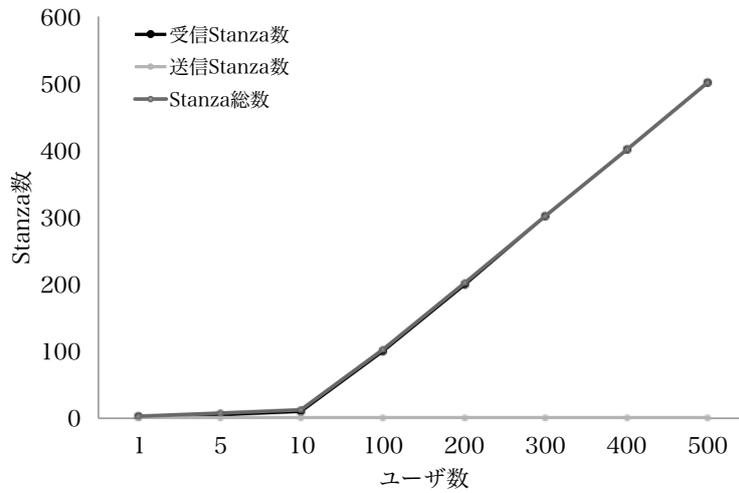


図 6.3: MUC 使用時のクライアント結びつけ時のユーザ増加に伴う Stanza の増加

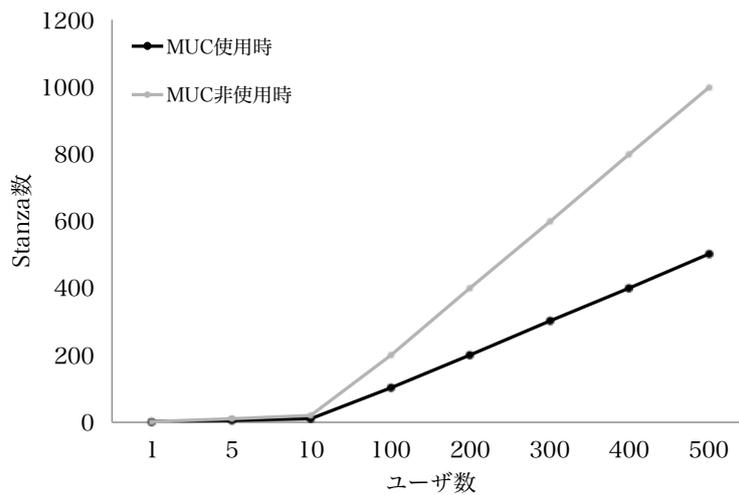


図 6.4: MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較

サーバー側の Stanza 数の変化

サーバー側における MUC を使用した場合と MUC を使用しなかった場合の結びつけ時とメッセージ送信時における、ユーザ増加に伴う Stanza の数を比較した。

1. 結びつけに伴って発生する Stanza の比較

- MUC 非使用時

MUC 非使用時におけるユーザーとクライアントとの結びつけに伴ってサーバーで発生する Stanza は以下の通りである。

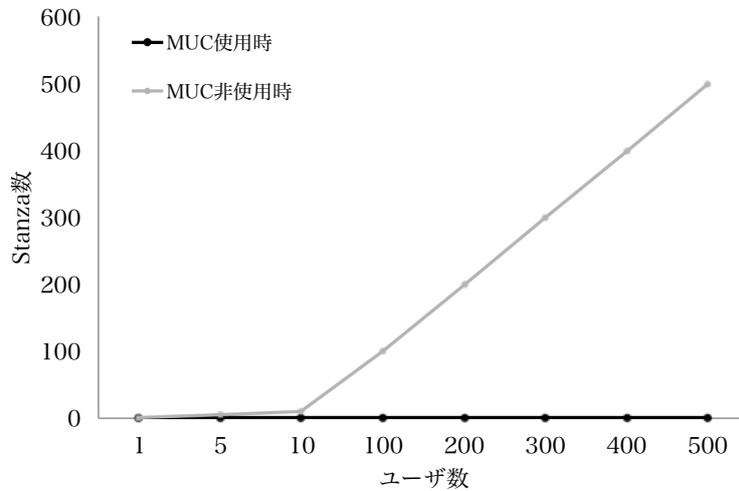


図 6.5: MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較

- (a) ユーザーからのフレンドリクエスト受信
- (b) クライアントへのリクエスト送信
- (c) クライアントからのリクエストの応答受信
- (d) ユーザーへのリクエストの応答送信
- MUC 使用時 MUC 使用時におけるユーザーとクライアントとの結びつけに伴って発生する Stanza は以下の通りである。
 - (a) クライアントからのチャットルームへの参加リクエスト受信
 - (b) クライアントへのチャットルームへの参加応答送信
 - (c) ユーザーからのチャットルームへの参加リクエスト受信
 - (d) ユーザーへのチャットルームへの参加応答送信
 - (e) チャットルームの参加者リストの更新に伴う通知送信

MUC 使用時の発生する Stanza の総量はユーザー数を n とした場合、

$$\sum_{i=1}^n a_i + n \quad (6.1)$$

と表される。

- 比較

図 6.6 に、サーバー側の MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較を示す。

2. メッセージ送信に伴って発生する Stanza の比較

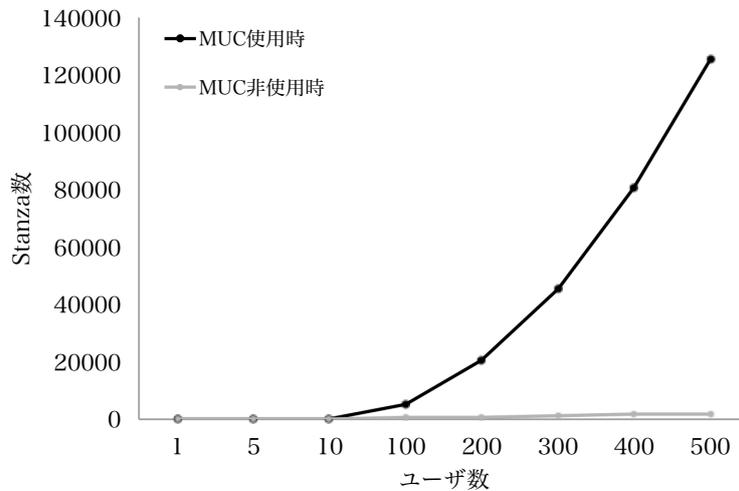


図 6.6: サーバー側の MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較

- MUC 非使用時

MUC 非使用時におけるクライアントからのメッセージ送信に伴ってサーバーで発生する Stanza は以下の通りである。

- クライアントからのメッセージ受信
- ユーザーへのメッセージ送信

- MUC 使用時

MUC 使用時におけるユーザーとクライアントとの結びつけに伴って発生する Stanza は以下の通りである。

- クライアントからのチャットルームへのメッセージ送信
- チャットルームからユーザーへのメッセージ送信

図 6.7 に示す通り、MUC を使用することによってクライアントから各ユーザーに向かってメッセージを送ることがなくなったため約 1/2 の Stanza 数で各ユーザーに向けてメッセージを送信することができた。

6.2.2 デバイスに付属しているセンサーの増加

1つのデバイスに付属しているセンサーの増加が起こると、センサー XMPP クライアントからのデータ送信時、あるいはユーザー XMPP クライアントにおけるデータ受信時においてメッセージ Stanza のサイズが変化する。図 6.8 にメッセージ Stanza が変化した場合のユーザ XMPP クライアント内に組み込まれている読み込み用プログラムの処理時間の変化について計測した結果を示す。

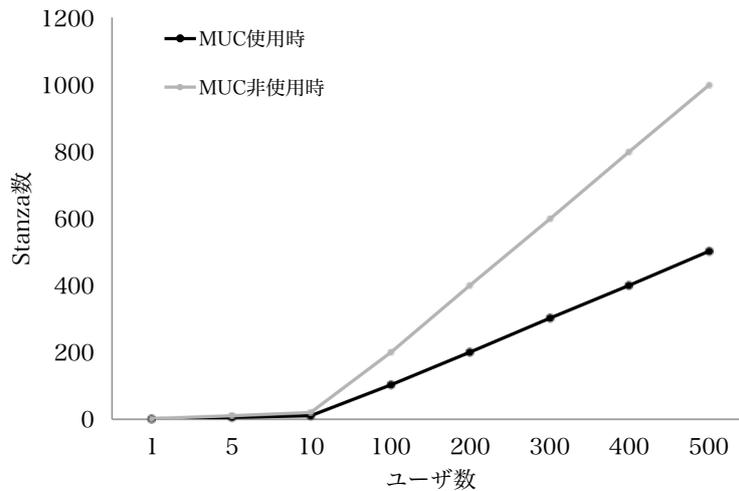


図 6.7: MUC 使用時と MUC 非使用時の結びつけに伴って発生する Stanza 数の比較

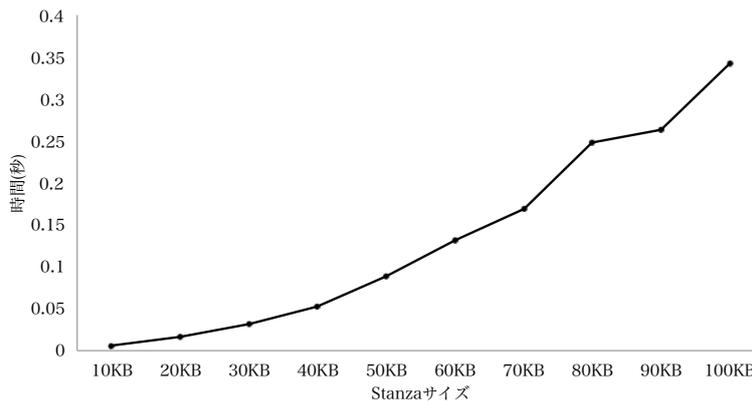


図 6.8: Stanza のサイズによる処理時間の変化

6.3 汎用性

今回の実装を動かすにあたって、XMPP のサーバ実装である Openfire[31] を使用した。パブリックに運用されている XMPP サーバには上記の他に ejabberd[32] や、Tigase[33] などがある。今回は xmpp.net[34] に公開されている XMPP サーバのリストより、3つの異なるサーバ実装を運用しているサービスを選んで周期起動型イベント向けセンサーの動作検証を行った。表 6.5 に、動作検証に使ったサーバの一覧を示す。

検証は各サービスに 2 つずつアカウントを作成し、3 つの異なるサービス間において、放射線量のデータが正しく転送されるかを検証した。表 6.5 にその結果を示す。

縦軸が送信側、横軸が受信側のサービスを示す。ejabberd を用いている xmpp.jp のアカウントを使用してデータが送信される場合を除いて正しく動作することが確認された。

表 6.4: パブリック XMPP サーバに使用されているサーバ実装

サービス名	サーバ実装名
jabber.de	openfire
xmpp.jp	ejabberd
tigase.im	tigase

表 6.5: 既存 XMPP サーバ・サービスにおける動作検証

送信 / 受信	jabber.de	xmpp.jp	tigase.im
jabber.de	○	○	○
xmpp.jp	×	×	×
tigase.im	○	○	○

以上から、メッセージを送信する場合の型については厳しく、メッセージを受信する場合の型については緩くなっている傾向があることがわかった。

6.3.1 考察

本章では、本アーキテクチャの実現可能性、規模性、汎用性について検証をおこなった。

実現可能性としては周期起動型イベント向けセンサー、散発型イベント向けセンサー、アクチュエータについてその正しい動作が確認された。これは、ネットワークにつながったデバイスが人間を中心としたアーキテクチャと親和性が高いことを示し、XMPP の拡張性がネットワークにつながったデバイスの多様性と適合することを示した。

規模性としては、1つ目にユーザーの増加によるセンサー XMPP クライアントからの Stanza の増加は、MUC を使用することによって低く抑えられることが判明した。これは、本アーキテクチャを実際に運用する際に、ユーザの増減が頻繁に起こる場合には MUC を使用し、チャットルームのユーザーの増減に伴うユーザークライアントへの通知送信の負荷に耐えられるサーバを用いる必要があることを示す。2つ目のデバイスに付属しているセンサーの増加に伴う Stanza の肥大化による処理時間は、指数関数的に増加することが判明した。したがって、XMPP にネットワークにつながったデバイスを参加させる際にはユーザー側のクライアントの性能を高くするか、センサークライアントから発せられる Stanza を少なくする必要がある。また、比較的リソースの限られたアクチュエータにメッセージを送信する際にもなるべく小さくする必要があることが推測される。

汎用性としては、3つのサービスを使って動作検証を行った。これにより、サーバ側で独自拡張した Stanza を受け入れるものと受け入れないものがあることが判明した。これを解決するには、XMPP の機械向け拡張の XEP を標準化することによって今後サーバ側の対応がなされていくのではないかと期待される。

第7章 結論

本章では、本論文をまとめ、今後の展望について述べる。

7.1 まとめ

本論文では、人とデバイスの結びつけを軸とした拡張可能なシステムを XMPP をベースとして構築できることを確認した。はじめに、第1章では、背景と目的及びアプローチについて述べた。第2章では、既存のセンサーネットワークのモデルを整理し、データの所有権の所在がデバイスの所有者ではなく、データを取得する人にあるべきであることを確認し、センサーデータの共有技術についての比較を行った。次に、第3章ではデバイスの所有者のダイナミックな切替と、デバイスから取得できるデータの共有が実現できる必要があることを示した。また、第4章ではデバイスの所有者のダイナミックな切替においてデバイスの種類によってその権限に違いがあることを明らかにした。以上の条件をもとに、ユーザとデバイスがフラットであり、これらの間において結びつけに伴う所有権の切替が可能なアーキテクチャを提案した。第5章では、提案アーキテクチャが実現可能であることを示すために、周期起動型イベント向けセンサーとして放射線量計、散発型イベント向けセンサーとして体重計、アクチュエータとして Web カメラを用いて実装を行った。最後に、第6章では、このアーキテクチャの実現可能性の検証、および提案アーキテクチャの規模性、汎用性についてそれぞれ評価を行い、実際に運用を行う際に考慮すべき条件を考察した。以上より、ネットワークに接続されたデバイスの所有権を制御することによる、人間を中心としたネットワークにつながった擬人化されたデバイスとコミュニケーションが可能なアーキテクチャの有用性を示した。

7.2 今後の展望と課題

本論文では、XMPP をベースとしたシステムの構築を行った。公共に設置されたネットワークにつながったデバイスは様々であり、センサーだけでなく、今後、デジタルサイネージなどこれまででは想定されなかったデバイスがネットワークに接続されていていく。デバイスと個人の結びつけの形は世界中で様々な試みがなされており、現在それぞれのデバイスにとって最も適切なコミュニケーションの形が模索されている。今回は XMPP による実装を行ったが、HTTP が SPDY を元とした双方向性をもったものに発展していこうとしている現状を考えると、将来においては全く異なった設計のプロトコルによって、個人との結びつけをもとにした、環境と対話するインターフェースシステムが構築され

ることも想定される。また、このシステムを Machine-to-Machine の環境に適用した場合には、本来チャット向けのプロトコルとして設計されたプロトコルであるが故に、リクエスト-レスポンスの形でなく、クライアント同士が相互にデータ送信を行い始めてしまうため、データ転送が止まらなくなって、暴走するようなことが起こらない設計をする必要がある。また、XMPP をセンサーネットワーク向けに拡張する標準化活動が 2013 年 1 月現在活発になりつつあり、2014 年中に標準化される可能性がある。現在世界中の研究者、企業の人々が議論を行なっている最中であり、これらとの整合性も取る必要があると考えられる。

謝辞

本論文の作成にあたり、主査である慶應義塾大学村井純博士、副査である植原啓介博士、斉藤賢爾博士のご指導、ご助言に感謝いたします。特に斉藤博士には、大学院への入学以前より指導していただきました。重ねて感謝申し上げます。

また、ご助言を賜りました、中村修博士、吉藤英明博士、Rodney D. Van Meter III 博士、三次仁博士、楠本博之博士、鈴木茂哉博士、Achmad Husni Thamrin 博士、武田圭史博士、朝枝仁博士、片岡広太郎博士を含む全ての Mobile and Ubiquitous Internet プロジェクトに所属する教員のみなさまと、XMPP に触れるきっかけを与えてくださった共愛学園前橋国際大学小柏伸夫博士に感謝いたします。

そして、研究を進めるにあたり、様々なご助言と励ましをいただきました徳田・村井・楠本・中村・高汐・バンミーター・植原・三次・中澤・武田合同研究室のみなさまに感謝申し上げます。特に実装を共に行った慶應義塾大学政策・メディア研究科勝治宏基氏および三部剛義氏に感謝いたします。偶然の共同開発とは言え、2人の協力がなければ実装の完了を果たすことは不可能でした。重ね重ね感謝いたします。また、ことあるごとに研究へのご助言をいただきました、空閑洋平氏、堀場勝広氏、松谷健史氏、岡田耕司氏、六田佳祐氏、上原雄貴氏、永山翔太氏、波多野敏明氏、重松邦彦氏、土井 裕介博士に感謝いたします。また、三部氏の他に共に励まし合いながら修士論文の執筆を行った宮崎圭太氏、米村茂氏に感謝いたします。執筆および学事手続きにあたり、大変お世話になりました。研究にご協力いただいた碓井利宣氏、山岸祐大氏、横石雄大氏、木本 瑞希氏、上野 幸杜氏、三條場 直希氏、渋谷拓也氏、小澤みゆき氏、鴻野弘明氏、中島明日香氏、水谷伊織氏、佐藤友紀氏、山本知典氏、関根 冬輝氏、徳差雄太氏、出水厚輝氏に感謝いたします。

大学院への進学へのきっかけを与えていただき、XMPP WG BoF にてご助言をいただいた、WIDE プロジェクトのみなさまに感謝いたします。

センサークライアントである体重計のアイデアが浮かぶきっかけとなった、天山湯治郷に感謝いたします。同時に、食事に困る一人暮らしの私に良質の麺類の食事を与えて下さった、ニューオリンズ様、常勝軒様、溜璃様、辰巳屋様に感謝致します。

最後に、私を育て、支えていただいた母貴邦子をはじめとする家族に感謝し、謝辞いたします。

参考文献

- [1] Dave Evans. The Internet of Things How the Next Evolution of the Internet The Internet of Things How the Next Evolution of the Internet Is Changing Everything. (April), 2011.
- [2] Cosm. Cosm. <https://cosm.com/>, 12 2012.
- [3] エヌ・ティ・ティ レゾナント株式会社. goo からだログ. <http://karada.goo.ne.jp/>, 12 2012.
- [4] Bluetooth Sig. HDP Implementation Guidance Whitepaper. (December), 2009.
- [5] Continua Health Alliance. Continua health alliance. <http://www.continuaalliance.org>, 12 2012.
- [6] H. Alvestrand. IETF Policy on Character Sets and Languages. RFC 2277 (Best Current Practice), January 1998.
- [7] NTT DOCOMO INC. iモード. <http://www.nttdocomo.co.jp/service/imode/>, 12 2012.
- [8] Marty Lyons. Terminals to texting Messaging from computers to mobile devices Origins of time-sharing MIT and Project MAC. 2006.
- [9] J. Oikarinen and D. Reed. Internet Relay Chat Protocol. RFC 1459 (Experimental), May 1993. Updated by RFCs 2810, 2811, 2812, 2813.
- [10] AOL Inc. Aol instant messenger. <http://blog.aim.com/>, 12 2012.
- [11] Google Inc. Our history in depth — company — google:. <http://www.google.com/about/company/history/>, 12 2012.
- [12] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120 (Proposed Standard), March 2011.
- [13] Twitter Inc. Twitter. <http://twitter.com>, 12 2012.
- [14] 次世代電子商取引推進協議会. センサネットワーク活用調査報告書. Technical report, 2008.

- [15] Luca Mottola. Programming Wireless Sensor Networks : Fundamental Concepts and State of the Art. *ACM Computing Surveys*, V(3):1–51, 2011.
- [16] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 6121 (Proposed Standard), March 2011.
- [17] Internet Engineering Task Force. Internet engineering task force. <http://www.ietf.org>, 12 2012.
- [18] M. Day, J. Rosenberg, and H. Sugano. A Model for Presence and Instant Messaging. RFC 2778 (Informational), February 2000.
- [19] Matthias Wimmer. jabberd14 xmpp/jabber server daemon:. <http://jabberd.org/>, 12 2012.
- [20] XMPP Standards Foundation. Xmpp standards foundation. <http://www.xmpp.org>, 12 2012.
- [21] Thijs Alkemade Dirk Meyer. *XEP-0257: Client Certificate Management for SASL EXTERNAL*. XMPP Standards Foundation, 1999-2012. <http://www.xmpp.org/extensions/xep-0257.html>.
- [22] Peter Saint-Andre. *XEP-0045: Multi-User Chat*. XMPP Standards Foundation, 1999-2012. <http://www.xmpp.org/extensions/xep-0045.html>.
- [23] W3C. Soap specification. <http://www.w3.org/TR/soap/>, 12 2012.
- [24] 日本 IBM. Ibm it で都市を、より賢くスマートに—ソリューション開発 - japan:. http://www-06.ibm.com/innovation/jp/smarterplanet/cities/cities_topic_201203/ohtaka.html, 12 2012.
- [25] MQTT.org. Mqtt: Mq telemetry transport. <http://mqtt.org>, 12 2012.
- [26] Current Cost Support Forum. Current cost support forum • view topic - bridge frozen/pachube server issues:. <http://currentcost.com/support/viewtopic.php?f=8&t=40>, 12 2012.
- [27] Forbes.com LLC. Court rules against wikileaks: Twitter data isn't secret, government orders are - forbes:. <http://www.forbes.com/sites/andygreenberg/2011/11/10/court-rules-against-wikileaks-twitter-data-isnt-secret-government-orders-are/>, 12 2012.
- [28] Yokota Yoshitake, Muto Kenji, and Goto Tomohiro. Digital Signage Vending Machine ” JX34 ”. 85(5):374–378, 2012.

- [29] 東日本鉄道株式会社. Suica. <http://www.jreast.co.jp/suica/index.html>, 12 2012.
- [30] 株式会社パスモ. Pasma. <http://www.pasmo.co.jp/>, 12 2012.
- [31] Ignite Realtime. Ignite realtime: Openfire server. <http://www.igniterealtime.org/projects/openfire/>, 12 2012.
- [32] Process One. ejabberd - scalable, distributed, fault-tolerant instant messaging - processone:. <http://www.process-one.net/en/ejabberd/>, 1 2013.
- [33] Inc. Tigase. Tigase.org — open source and free xmpp/jabber software:. <http://www.tigase.org/>, 1 2013.
- [34] XMPP Standards Foundataion. xmpp.net. <http://xmpp.net/>, 1 2013.

付録A 本提案システムにおける Stanza例一覧

本提案システムにおいてクライアント間でのコミュニケーションに使われる XMPP の Stanza の例の一覧を以下に記す。

なお、Stanza には 2 人の人間のアカウントと 2 つのセンサークライアント向けアカウントと 1 つのアクチュエータ向けアカウントが登場する。

- alice@domain.com

- bob@domain.com

上記 2 つのアカウントは人間のアカウントであり、Stanza は実際に人間がクライアントを操作した結果送受信されるものである。

- exclusive@domain.com

exclusive@domain.com は専有型デバイスに用いられているセンサークライアント向けアカウントである。

- share@domain.com

share@domain.com は共有型デバイスに用いられているセンサークライアント向けアカウントである。

- actuator@domain.com

actuator@domain.com はアクチュエータに用いられているアクチュエータ向けアカウントである。

なお、Stanza の中に含まれる id の値は一意であることが必須である。

A.1 人間とデバイスの結びつけに関する Stanza

Alice から Share へのフレンド要求

```
<presence type='subscribe' to='share@domain.com' >  
<nick>s</nick>  
</presence>
```

Share から Alice へのフレンド要求の承認

```
<presence type='subscribed' to='alice@domain.com' >  
</presence>
```

Alice による Share のリムーブ

```
<iq type='set' id='remove'>  
<query xmlns='jabber:iq:roster'>  
<item jid='share@domain.com' subscription='remove' />  
</query>  
</iq>
```

A.2 権限に関する Stanza

Alice が LimitedFlag を持つことの Share からの通知

```
<message xmlns='jabber:client' type='chat'  
id='notify' to='alice@domain.com'  
from='share@domain.com/camera'>  
<device xmlns='urn:xmpp:device' />  
<flag = 'limited' />  
</message>
```

Alice によるセンサークライアント権限の一覧の要求

```
<message xmlns='jabber:client' type='chat'  
id='request' to='share@domain.com'  
from='alice@domain.com/mobile'>  
<device xmlns='urn:xmpp:device' />  
<request type='privilege' />  
</message>
```

share による権限一覧の結果の送信

```
<message xmlns='jabber:client' type='chat'
id='response' to='alice@domain.com'
from='share@domain.com/camera'>
<device xmlns='urn:xmpp:device' />
<response>
<privilege>
<item type='data' value='acquire' />
<item type='customset' value='allow' />
<item type='actuation' value='allow' />
</privilege>
</response>
</message>
```

bob による alice の share におけるユーザー権限を owner への変更要求

```
<message xmlns='jabber:client' type='chat'
id='response' to='share@domain.com'
from='bob@domain.com/mobile'>
<device xmlns='urn:xmpp:device' />
<set type='privilege' user='alice@domain.com'
value='owner' />
</message>
```

bob による alice の share におけるユーザー権限の変更要求

```
<message xmlns='jabber:client' type='chat'
id='response' to='share@domain.com'
from='bob@domain.com/mobile'>
<device xmlns='urn:xmpp:device' />
<set type='privilege' user='alice@domain.com'
value='limited'>
<privilege>
<item type='data' value='acquire'>
<item type='customset' value='allow'>
<item type='actuation' value='deny'>
</privilege>
</message>
```

A.3 データ送信に関する Stanza

share から alice への計測データの送信

```
<message xmlns='jabber:client' type='chat'
id='submit' to='alice@domain.com'
from='share@domain.com/mobile'>
<device xmlns='urn:xmpp:device' />
<data type='privilege' user='alice@domain.com'
value='limited'>
<privilege>
<item type='data' value='acquire'>
<item type='customset' value='allow'>
<item type='actuation' value='deny'>
</privilege>
</data>
</message>
```

A.4 アクチュエーションに関する Stanza

alice から share へのデバイス特定タグ一覧要求

```
<message xmlns='jabber:client' type='chat'
id='request' to='share@domain.com'
from='alice@domain.com/mobile'>
<device xmlns='urn:xmpp:device' />
<dose unit='μ Sv/h'>XXX</dose>
<time>YYYY:MM:DD HH:MM:SS TMZ</time>
</data>
</message>
```

share によるタグ一覧の送信

```
<message xmlns='jabber:client' type='chat'
id='response' to='alice@domain.com'
from='share@domain.com/camera'>
<device xmlns='urn:xmpp:device' />
<response>
<uniqueness>
<device type='sensor' name='camera'>
<element value='take'>
<attribute value='time' variable='now'
mean='take picture immedeately' >
</attribute>
</element>
</device>
</uniqueness>
</response>
</message>
```

alice から share へのアクチュエーションの実行

```
<message xmlns='jabber:client' type='chat'
id='response' to='share@domain.com'
from='alice@domain.com/mobile'>
<device xmlns='urn:xmpp:device' />
<request>
<take get='now' />
</request>
</message>
```

share から alice へのアクチュエーション結果の送信

```
<message xmlns='jabber:client' type='chat'
id='response' to='alice@domain.com' from='share@domain.com/camera'>
<device xmlns='urn:xmpp:device' />
<response>
<data>
<take result='http://domain.com/~camera/120111112233.jpg' />
</data>
</response>
</message>
```

変更履歴

本論文の変更履歴は次の通りである。

第1版 2013年1月10日
慶應義塾大学湘南藤沢メディアセンター収蔵版

第2版 2013年2月7日
慶應義塾大学湘南藤沢メディアセンター収蔵改訂版

第3版 2013年2月12日
本版