

## マルチオーバーレイネットワーク環境における リソース探索言語とその処理系

石芳正<sup>†</sup> 寺西裕一<sup>\*†</sup> 吉田幹<sup>††</sup> 下條真司<sup>\*</sup> 西尾章治郎<sup>†</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科 <sup>††</sup> 株式会社ビーバーール

<sup>\*</sup> 独立行政法人情報通信研究機構

### 概要

ユビキタス環境においては、オーバーレイネットワーク上に分散して存在する様々なリソースを探索できることが必要となる。単一のオーバーレイでは対応可能な探索パターンが制限されるため、複数のオーバーレイネットワークを同時に稼動させるマルチオーバーレイネットワーク環境が要求される。本研究では、マルチオーバーレイネットワーク環境において、分散して存在するリソースに属性による探索を行うための探索言語を定義し、探索言語から探索に適したオーバーレイネットワークを動的に選択することで、効率的な探索を可能とする処理系を提案する。本処理系をPIAX上に実装し、Live E!プロジェクトの気象センサから得られる任意のデータをブラウザ上に表示可能なシステムを実現した。

## A Proposal of Resource Search Language and Its Implementation for a Multi-overlay Network

Yoshimasa Ishi<sup>†</sup>, Yuuichi Teranishi<sup>†\*</sup>, Mikio Yoshida<sup>††</sup>, Shinji Shimojo<sup>\*</sup>, and Shojiro Nishio<sup>†</sup>

<sup>†</sup> Graduate School of Information Science, Osaka University

<sup>††</sup> BBR Inc.

<sup>\*</sup> National Institute of Information and Communication Technology

### Abstract

In a ubiquitous environment, it's necessary to find various resources on a overlay-network to make wide variety of services. But an overlay network supports a limited query pattern. To support various query pattern, we propose to use multiple overlays together. I define a query language for a multi-overlay network to search those distributed resources by its attributes. And I extend the PIAX to support the query language and a multiple overlay handling mechanism. The mechanism selects a suitable overlay dynamically by evaluating the query. Finally, I implemented a application using the mechanism to show any data obtained from Live E! project's weather sensors on a web browser.

### 1 はじめに

ユビキタス環境においては、PDAや携帯電話といった小型アプライアンス、センサや街頭カメラといった小型の環境情報収集機器、家電機器や自動車といった生活に密着した機器などがオープンなネットワークを介して相互に接続される。これらの機器を情報源として扱い、発生する情報を相互利用することでユーザが置かれた状況やユーザプロフィールに応じた高度なサービスが実現可能となる。こうした情報源はデバイス技術やワイヤレスネットワーク技術の高度化に伴い、ますます小型化、廉価化、多様化が進み、環境中には多種多様なデバイスがより高密度に配置されるとともに、それらのデバイスから生じる情報も多様化、増加すると考えられる。

上記状況下において、ユビキタスアプリケーションには広域に分散して存在する大量の情報源やプロ

セッシング資源、ストレージ、出力デバイスなどの各種リソースの中からサービス実現に必要な条件に合致する物を迅速に探索できることが要求される。こうした要求に対処するため、広域に分散したリソースを自律的に相互接続させ、効率的な探索を実現するオーバーレイネットワークの構成法に関する研究開発が数多くなされてきている。

ここで、アプリケーションが要求する探索は、リソースが持つ属性に対する条件指定となる。例えば、属性 'category' が 'storage' を満たすリソース、属性 'location' が指定矩形に含まれるリソース等である。このときアプリケーションとしては、完全一致検索、条件が範囲を持つ範囲検索、AND条件、OR条件を含む複合検索など種々の検索条件を指定したいと考えられる。しかし、通常、オーバーレイネットワークは、対象となる探索はあらかじめ決められおり、特定の探索パターンに特化されている。例えば、LL-

Net[1], Delauney Network[2][3] は 2 次元の地理的條件を, Skip Graphs[4][5] は完全一致または範囲検索を, Bloom Finger Table[6] は文書の全文検索をそれぞれ扱うものである。これらは構造型オーバーレイネットワークであるが, 非構造型のオーバーレイネットワークを特定の目的に特化し効率化する提案も数多くなされている。

従来の研究では, P2P システムは一つのオーバーレイネットワークを持つことが前提とされてきた。しかし, 実際には必ずしも一つに限る必要はなく, 実装が可能であれば, オーバーレイネットワークを複数同時に稼働させても構わないはずである。そこで我々の研究プロジェクトでは, P2P エージェントミドルウェア PIAX[7] において, 必要となるオーバーレイネットワークを複数同時に稼働させ, アプリケーションが用途に応じて使い分けことが可能な ‘マルチオーバーレイネットワーク環境’ の研究開発を実施してきた。

マルチオーバーレイネットワーク環境では, それぞれのオーバーレイネットワークが独立して動作するため, 個々が動的に稼働を開始, 終了することが考えられる。例えば PIAX においては, オーバーレイネットワークは動的ローディングにより新たに追加可能となっている。よって, 従来存在しなかった新たなオーバーレイネットワークが稼働を開始する可能性があり, アプリケーション開発者は, どのオーバーレイネットワークが実行時に利用可能であるか知ることができないという問題が生じる。

そこで本研究では, マルチオーバーレイネットワーク環境において, 分散して存在するリソースから条件を満たすものを探索可能とする探索言語を定義し, 適切なオーバーレイネットワークを動的に選択する処理系を提案する。本処理系は PIAX に実装し, 実証アプリケーションの実装等を通じてその有効性を確認している。

## 2 ユビキタス環境におけるクエリパターンとマルチオーバーレイ環境

CASTANET[8][9] や OSNAP[10] などのセンサネットワークやアクチュエータネットワークを介した共通アクセスプロトコルが提案されており, センサやアクチュエータへのアクセスが容易に可能となりつつある。また, デジタル百葉箱を個人や組織が設置運営し, 環境保護対策, 教育等様々な分野での自由な利用を目指す Live E! プロジェクト [11] や, 複数の分散した研究拠点に配備されたセンサデバイス群やセンサデバイス群から得られる情報を研究者間で共有する X-Sensor プロジェクト [12] では, センシングデータアーカイブを公開しており, オープンなデータベースとして利用可能となっている。これらネットワーク上のリソースは識別名や種別名, 物理的な位置といった様々な属性を持つと考えられる。例えば, センサデバイスでは, そのセンサの観測対象や設置されている位置情報, 現時点での最新のセン

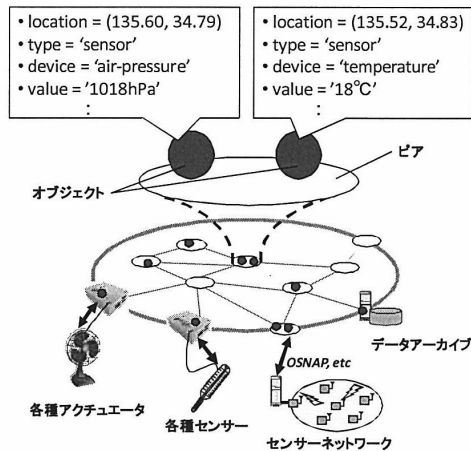


図 1: オブジェクトと属性

シングデータなどが属性として挙げられる。本研究では, ネットワーク上に分散し, アプリケーションからアクセス可能なリソースをオブジェクトとして扱う。また, オブジェクトはネットワーク上のピアに複数存在しうる物とする。各オブジェクトは複数の属性を持つ(図 1)。ユビキタスアプリケーションにおいては, これらの属性をもとにオブジェクトを探索し, 条件を満たす複数のオブジェクトにアクセスする。

### 2.1 クエリパターン

我々の研究グループでは, ユビキタスアプリケーションがオブジェクトを探索する際に要求するクエリパターンとして, 以下の 4 パターンを想定している [7]。

#### 完全一致探索

指定された値, キーワードに完全に一致する属性を持つ情報を探索する。

#### 部分一致探索

指定された値, キーワードに部分的に一致する属性を持つ情報を探索する。

#### 範囲探索(1次元またはN次元)

指定された範囲に含まれる属性を持つ情報を探索する。地理的領域指定の探索はこれに相当する。

#### 複合探索

上記 3 種のクエリパターンを AND もしくは OR により結合した条件による探索。

これらの探索をオーバーレイネットワークの適用により実現することを考えると, 完全一致探索は, 各種 DHT (Distributed Hash Table) や DOLR (Decentralized Object Location and Routing) を用いることで対応可能である。また, Skip Graphs や Multi-key Skip Graph [13] では, 完全一致探索及び一次元範囲探

表 1: Operator

=	完全一致探索
=~	部分一致探索
<	範囲探索
>	
<=	
>=	
inside	範囲内探索
and	論理演算
or	

索が可能となっている。LL-Netも地理的情報に特化した範囲探索といえる。一方、AND や OR のような複合探索に対応するには単一のオーバーレイだけでは実現が困難である。このため、アプリケーションからの探索要求に応えるには、想定されるクエリパターンに応じたオーバーレイを用意し、適切にオーバーレイを切り替えるマルチオーバーレイハンドリングを行うマルチオーバーレイ環境が必要となる。

### 3 リソース探索言語の定義

2章で述べたように、本研究では探索対象となるネットワーク上のリソースを属性を持つオブジェクトとして扱う。属性とは、一般的なプログラミング言語における変数と同様に、値に属性名がついたものとして扱い、“属性名 is 値”の対応を持つこととする。また、オブジェクトは同時に複数の属性を持つことができるが、同名の属性を持つことはできないものとする。

本研究において提案するリソース探索言語の基本定義は以下の通りである。

```
Expression ::=
  Expression operator Expression |
  AttributeName | ObjectValue
```

AttributeName は属性名、ObjectValue は比較対象となる値を表す。operator は表 1 に示す演算子である。

一般的な演算子に加え、範囲探索を容易に実現するために“ある範囲内”を意味する述語として inside を導入した。Object Value には Number, String, Boolean などの基本型の他、矩形範囲を表すオブジェクトに対応する rect(), 一次元の範囲を表すオブジェクトに対応する range(), 位置座標を表現する location() 等の Special function を指定可能とする。

オーバーレイネットワークの特性を検討すると、オーバーレイネットワークではノードの出入りやネットワーク断などが生じるため、ネットワークにおける探索対象を完全に網羅しているとは仮定できず、開世界仮説 (Open World Assumption) に基づくことになる。このため、否定演算 (“not”, “!=”) は言語体系に含めない。

例えば、“location 属性が矩形領域( 135.5, 34.8 ) –

( 135.6, 34.9 ) 内にあり、type 属性が university であるオブジェクト”を対象とする場合は、まず、第一の Expression として、Attribute Name に “location”, Operator に “inside”, Object Value に Special function の rect() を使用する。そして、第二の Expression として、同様に “type”, “=”, “university” を指定する。これら 2 つの Expression を “and” で結合した以下が上記に対応した検索条件となる。

```
location inside rect(135.5, 34.8, 0.1, 0.1)
and type = "university"
```

## 4 PIAX におけるリソース探索処理の実装

### 4.1 PIAX の概要

我々の研究グループでは、オーバーレイネットワークと分散エージェントの機能を統合したプラットフォーム PIAX を研究開発してきた。PIAX は、複数のオーバーレイネットワークを同時に持つことができ、かつ新たなオーバーレイネットワークを、運用中に動的に追加できるマルチオーバーレイ環境を実現している。現在、利用可能なオーバーレイネットワークとして、一般的なフラッディング、DHT, LL-Net, Multi-key Skip Graph, 属性に応じたルーティングを行なう DOLR が実装されている。また、オーバーレイネットワークを用いて発見される複数のエージェントに対して、発見と同時にメッセージを送信し、複数のピアにおける処理の結果を、一度に収集することが可能な発見型メッセージング ( discovery messaging ) 機能を実現している。

### 4.2 クエリ処理のための API 拡張

PIAX では 3 章で述べたオブジェクトを、ピア上のエージェントに対応付け、属性を保持させる。本研究における実装では、属性名とオーバーレイネットワークを対応付けることでオブジェクトのネットワーク上への登録と、探索言語による探索を実現している。これに伴い、属性設定 API, オーバーレイ関連付け API, 発見型メッセージング API を新たに実装している。

#### 4.2.1 属性設定 API

エージェントに属性を設定するためのメソッド定義は以下の通りである。

```
void setAttrib(String name, Object value)
void setAttrib(String name, Object value,
  boolean useOverlay)
void setAttrib(String name, Object value,
  String ovClassName)
```

これらの API はエージェントが持ち、個々のエージェント毎に独立した属性設定を可能としている。引数 name は属性名、value は属性に対応する任意の型を持つ値を指定する。基本的には、属性に対応する

オーバーレイネットワークの選択は自動的に行われるが、明示的にどのオーバーレイネットワークを使うか指定することも可能である。useOverlay は属性をオーバーレイネットワーク上に登録するか否かを真偽値を指定する。ovClassName には属性を登録するオーバーレイネットワークを提供するクラス名を指定する。

#### 4.2.2 オーバレイ関連付け API

属性名とオーバーレイネットワークを対応付けを行うメソッド定義は以下の通りである。

```
void setAttribOverlayAssoc(String name,
String ovClassName)
```

引数 name は属性名、ovClassName には属性に対応するオーバーレイネットワークのクラス名を指定する。ここで対応付けた関係は、図 2 中の overlay table に登録され、発見型メッセージングを行う際に使用する。

#### 4.2.3 発見型メッセージング API

以下に発見型メッセージングを行うためのメソッドを記す。

```
Object[] discoveryCall(String queryCond,
String method, Object... args)
ReturnSet<Object> discoveryCallAsync(
String queryCond, String method,
Object... args)
void discoveryCallOneway(String queryCond,
String method,
```

queryCond に本研究で述べた探索言語によるクエリを文字列で指定する。また、探索により発見されたエージェントから得られる応答の待機処理の違いにより、同期的に応答を得る discoveryCall、非同期で応答を得る discoveryCallAsync、メッセージを送信のみで応答を待たない discoveryCallOneway が用意されている。

#### 4.3 クエリ処理系の実装

図 2 に発見型メッセージングによるクエリ発行時の処理の流れを示す。発見型メッセージング API により、与えられたクエリは Query Parser において解析され、ツリー状に展開した内部形式に変換される。次に Overlay selector がツリーを走査し、オーバーレイネットワークに発行可能な Expression (overlay query と呼ぶ) の候補が抽出される。発行可能か否かの判定には、overlay table に属性名が登録されているか否かで行う。ここで、'overlay query' の候補は複数抽出される可能性があるが、現時点の実装では最初に抽出されたトリプルを 'overlay query' としている。'overlay query' 以外は、オーバーレイネットワークを通じて各ノードに配信され、実行される Expression

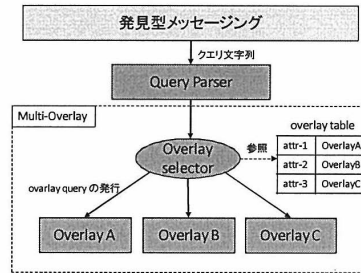


図 2: クエリ発行系の実装

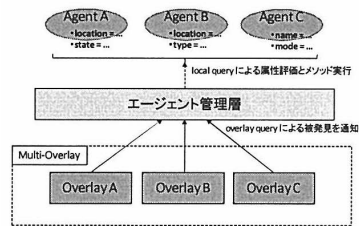


図 3: クエリ受信系の実装

(local query と呼ぶ) となる。'overlay query' に含まれる属性名から、発行先のオーバーレイネットワークを決定し、属性値を探索キーとしてオーバーレイネットワークに探索要求を出す。この探索要求のペイロードには 'local query', メソッド名, 引数を格納し、オーバーレイネットワークを介して 'overlay query' に合致したエージェントを持つピアに配信される。

次に、クエリ受信時の処理の流れについて図 3 に示す。受信されたクエリのペイロードより取り出された 'local query', メソッド名, 引数は発行時の流れを逆にたどり、ピア上の全エージェントを管理するエージェント管理層に通知される。ここで 'local query' の評価を行い、条件を満たす属性を持つエージェント (オブジェクト) を同定する。同定されたエージェントが、メソッド名で指定されたメソッドを持つ場合はそのメソッドに引数を受け渡し、実行する。実行後、得られた返り値は発行元のピアに返送する。以上の動作により、マルチオーバーレイ環境における資源探索が実現される。

#### 4.4 関連研究

オーバーレイネットワークを対象とした検索処理を扱った研究はこれまでにいくつか存在する。文献 [14] では、DHT ベースのファイル共有のための検索処理機構を実現しており、主に JOIN を扱う方法を示している。文献 [15] は、階層化された複数の index を作成して、DHT に格納しておき、データが見つかるまで再帰的にクエリを繰り返す方法を提案している。文献 [16] では、複数の属性を対象とした検索、範囲を持つ条件を持つ検索、JOIN などを DHT (Chord) を拡張して行なう方法が示されている。いずれも複合検索を DHT ベースの一つのオーバーレイネット

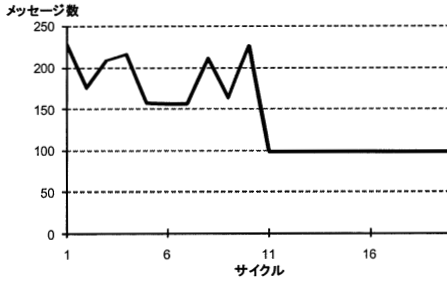


図 4: メッセージ数の推移

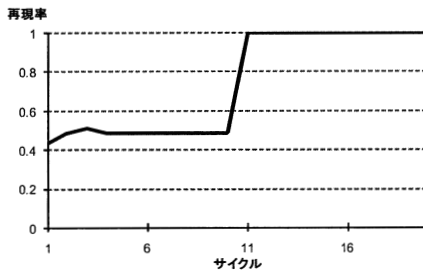


図 5: 再現率の推移

ワークを対象として実現する方法を示したものであり、従来、同時に存在する複数のオーバーレイネットワークを使い分けることを想定した検索処理は見あたらない。

## 5 評価

実装した処理系の有用性及び動作を確認するため、シミュレーション評価を行った。ピア数は 100 とし、各ピア上には探索対象としてエージェントを 1 つずつ配置した。このエージェントには location 属性を付与し、ランダムな位置を設定する。99ピアは LL-Net オーバレイとフラッディングオーバーレイを持ち、残り 1ピアは、まだ LL-Net オーバレイに参加しておらず、フラッディングオーバーレイのみを持つ想定とした。このフラッディングオーバーレイのみを持つピアより、単位時間 1 サイクル毎に 1 回発見型メッセージングを繰り返し行い、発見型メッセージング毎のメッセージ数と再現率を取得した。サイクル 10 において、このピアの location 属性を LL-Net オーバレイに設定し、LL-Net オーバレイに参加させた。

図 4 にメッセージ数の推移を示す。横軸をサイクル、縦軸をメッセージ数とした。図 5 に再現率の推移を示す。横軸をサイクル、縦軸を再現率とした。

サイクル 10 においてメッセージ数が増加しているが、これは発見型メッセージングのコストに LL-Net への参加コストが加わったためである。また、サイクル 10 を境に発見型メッセージングが構造型オーバーレイである LL-Net 上でのクエリ発行に切り替わったため、メッセージ数は約 100 個で安定し、再現率は

0.5 前後から 1.0 に改善している。この切り替えは、システムの再起動など行う必要なく、シームレスに行われる。以上より、クエリを同一に保ちつつ、より効率のよいオーバーレイの切り替えの動作が確認できた。

## 6 センサデータ共有アプリケーションの実装

本処理系を実装した PIAX を用いて、Live E! プロジェクトが運営するデジタル百葉箱が発信するデータをエージェントに蓄積し、探索オーバーレイとして LL-Net を用いて探索・共有するアプリケーションを実装した。本アプリケーションでは、分散して配備された気象センサ群が発するリアルタイムなデータ及び各ノードに蓄積された過去のセンサデータをブラウザ上の地図を用いて俯瞰参照することができる。

本アプリケーションは、デジタル百葉箱が設置された拠点ごとにピアを置くことを想定しており、デジタル百葉箱から SOAP で送られるセンシングデータを受け取る。各ピアには Web インタフェースと SOAP インタフェースを実現するための Servlet コンテナ (Jetty) と、デジタル百葉箱から送られるセンシングデータを格納する組み込み DB (Apache Derby) を内蔵した。さらに、デジタル百葉箱の位置情報を管理し、ブラウザ、他ピアからの要求に応じて、必要な情報を応答するため Sensor Agent を実装した (図 6)。

ユーザは Web ブラウザで各拠点のピア、またはローカルに立ち上げたピアに接続することで、共有されているセンサデータを図 7 に示すように参照することができる。画面上には地図、データ取得時刻を調整するスライド、センサ種別選択ボタン、温度制限スライド、風速制限スライドが配置されている。地図上には地図の範囲内に含まれる百葉箱のアイコンがセンサの設置場所に合わせて表示され、アイコンをクリックすることで、個々の百葉箱の詳細な情報がポップアップ表示される。このアイコンの表示に際して、表示されている地図の範囲内に含まれるセンサの情報を得るため、表示されている地図の範囲、温度制限スライドの位置、風速制限スライドの位置から以下のクエリを生成し、発見型メッセージングを行う。

```
location inside rect(x, y, width, height)
and X < limit
( X : Temperature もしくは WindSpeed )
```

発見された Sensor Agent は自身が管理するローカル DB を参照し、パラメータに応じたセンサの ID、緯度経度、センサ種別、センシングデータ、データ取得時刻を要求元に返す。この動作は、定期的なアップデート、またはユーザ操作による地図表示位置の変更、センサ種別の変更時にも行われ、ほぼリアルタイムでの情報表示を実現している。

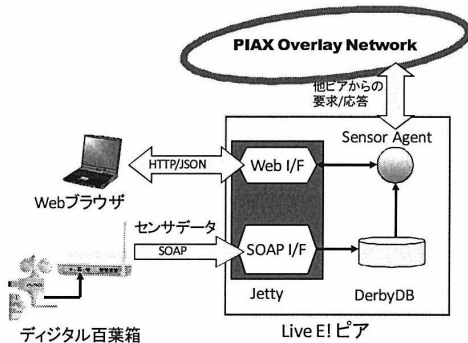


図 6: Live E! ピアの構成

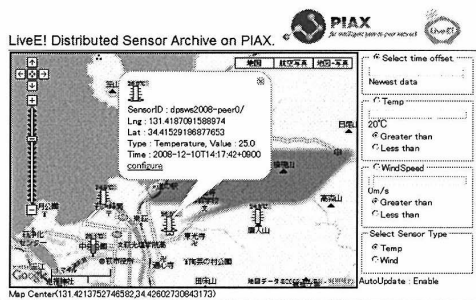


図 7: ブラウザ上の表示

## 7 まとめ

本研究では、マルチオーバーレイネットワーク環境において、分散して存在するリソースのなかから条件を満たすものを探索可能とする探索言語を提案し、その処理系を PIAX に拡張する形で実装した。適切なオーバーレイネットワークを用いた探索の動作を、実装した処理系を用いたシミュレーションにより確認した。また、Live E! プロジェクトの気象センサを題材にしたセンサ共有環境を構築し、その動作を確認した。今後は、センサデータ共有環境などへの適用を通じ、運用評価を行うほか、適合オーバーレイの選択手法について検討を進める予定である。

## 謝辞

本研究の一部は、平成 20 年度総務省委託研究「ユビキタスサービスプラットフォーム技術の研究開発」による成果である。ここに記して謝意を表す。

## 参考文献

[1] 金子雄, 春本要, 福村真哉, 下條真司, 西尾章治郎. ユビキタス環境における端末の位置情報に基づく P2P ネットワーク. 情報処理学会論文誌. データベース, Vol. 46, No. 18, pp. 1-15, December 2005.

[2] F. Araujo and L. Rodrigues. GeoPeer: A Location-Aware Peer-to-Peer System. Proc. The 3rd IEEE International Symposium on Network Computing and Applications (NCA ' 04), pp. 39-46, 2004.

[3] 大西真晶, 源元佑太, 江口隆之, 加藤宏章, 西出亮, 上島紳一. ノード位置を用いた P2P モデルのためのドロネー図の自律分散生成アルゴリズム. 情報処理学会論文誌: データベース, Vol. 47, pp. 51-64, March 2006.

[4] James Aspnes and Gauri Shah. Skip graphs. In *Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 384-393, Baltimore, MD, USA, 12-14 January 2003.

[5] James Aspnes and Gauri Shah. Skip graphs. *ACM Transactions on Algorithms*, Vol. 3, No. 4, p. 37, November 2007.

[6] 佐藤一帆, 松本倫子, 吉田紀彦. 複数キーワード検索に対応した分散ハッシュ型 P2P ネットワーク. In *Proceedings of FIT2007*, 2007.

[7] 吉田幹, 奥田剛, 寺西裕一, 春本要, 下條真司. マルチオーバーレイと分散エージェントの機構を統合した P2P プラットフォーム PIAX. 情報処理学会論文誌, Vol. 49, No. 1, pp. 402-413, January 2008.

[8] 小澤政博, 川原圭博, 川西直, 森川博之. B-7-169 REST アーキテクチャスタイルに基づくコンテキストウェアサービス連携フレームワークの設計. 電子情報通信学会総合大会講演論文集, Vol. 2007, p. 259. 社団法人電子情報通信学会, March 2007.

[9] 川西直, 小澤政博, 川原圭博, 森川博之. B-7-170 コンテキストウェアサービス連携フレームワークを用いたアプリケーションの作成. 電子情報通信学会総合大会講演論文集, Vol. 2007, p. 260. 社団法人電子情報通信学会, March 2007.

[10] 長谷川晃朗, 横堀充, 川西直, 寺西裕一, 小花貞夫. 画像も扱える異種センサネットワーク相互接続プロトコル OSNAP Vi の提案. 電子情報通信学会総合大会講演論文集, Vol. 2008, p. 97. 社団法人電子情報通信学会, March 2008.

[11] Live E! ~活きた地球の環境情報~. available at <http://www.live-e.org/>.

[12] 神崎映光, 原隆浩, 若宮直紀, 下條真司. 複数拠点統合型センサネットワークテストベッド X-Sensor の設計と実装. 情報処理学会マルチメディア通信と分散ワークショップ論文集, Vol. 2007, pp. 117-121, November 2007.

[13] 小西佑治, 吉田幹, 竹内亨, 寺西裕一, 春本要, 下條真司. 単一ノードに複数キーを保持可能とする SkipGraph 拡張. 情報処理学会論文誌, Vol. 49, No. 9, pp. 3223-3233, September 2008.

[14] Matthew Harren, Joseph M. Hellerstein, Ryan Huebsch, Boon Thau Loo, Scott Shenker, and Ion Stoica. Complex Queries in DHT-based peer-to-peer networks. Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS ' 02), pp. 242-259, March 2002.

[15] P.A. Felber, E.W. Biersack, L. Garcés-Erice, K.W. Ross, and G. Urvoy-Keller. Data Indexing and Querying in DHT Peer-to-Peer Networks. In *Proceedings of ICDCS*, 2004.

[16] Peter Triantafyllou and Theoni. Pitoura. Towards a Unifying Framework for Complex Query Processing over Structured Peer-to-Peer Data Networks. *LECTURE NOTES IN COMPUTER SCIENCE*, Vol. 2944/2004, .